the Garbage Collection Bibliography

Richard Jones
Computing Laboratory
University of Kent at Canterbury

April 4, 2017

[Aarts and others, 1991] Aarts et al., editors. *PARLE'91 Parallel Architectures and Languages Europe*, volume 505 of *Lecture Notes in Computer Science*, Eindhoven, The Netherlands, June 1991. Springer-Verlag.

[Abdullah and Edemenang, 1993] Saleh E. Abdullah and E.J.A. Edemenang. A comparative study of dynamic memory management techniques. *Advances in Modelling and Analysis*, 15(2):17–31, 1993.

[Abdullah, 1992] Saleh E. Abdullah. Managing computer memory: Dynamic allocation and deallocation strategies. In *Proceedings of Second Conference on Information Technology and its Applications*, pages 25–40, Leicester, December 1992.

[Abdullah, 1994] Saleh E. Abdullah. Recycling garbage. In *Proceedings of Third Conference on Information Technology and its Applications*, pages 192–197, Leicester, April 1994.

[Abdullahi and Ringwood, 1996a] Saleh E. Abdullahi and Graem A. Ringwood. Empirical studies of distributed garbage collection: Parts i, ii and iii. Technical report, Queen Mary and Westfield College, University of London, 1996.

[Abdullahi and Ringwood, 1996b] Saleh E. Abdullahi and Graem A. Ringwood. Garbage collecting the Internet. Technical report, Queen Mary and Westfield College, University of London, 1996? Draft version of [Abdullahi and Ringwood, 1998].

[Abdullahi and Ringwood, 1998] Saleh E. Abdullahi and Graem A. Ringwood. Garbage collecting the Internet: a survey of distributed garbage collection. *ACM Computing Surveys*, 30(3):330–373, September 1998.

[Abdullahi *et al.*, 1992] Saleh E. Abdullahi, Eliot E. Miranda, and Graem A. Ringwood. Distributed garbage collection. In Bekkers and Cohen [Bekkers and Cohen1992].

[Abdullahi, 1995] Saleh E. Abdullahi. *Empirical Studies of Distributed Garbage Collection*. PhD thesis, Queen Mary and Westfield College, December 1995.

[Abelson *et al.*, 1996] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, second edition, 1996.

[Abraham and Patel, 1987] Santosh Abraham and J. Patel. Parallel garbage collection on a virtual memory system. In *International Conference on Parallel Processing*, pages 243–246, University Park, Pennsylvania, USA, August 1987. Pennsylvania State University Press. Also technical report CSRD 620, University of Illinois at Urbana-Champaign, Center for Supercomputing Research and Development.

[Abramowich, 1967] John Abramowich. Storage allocation in a certain iterative process. *Communications of the ACM*, 10(6):368–370, June 1967.

[Abuaiadh *et al.*, 2004] Diab Abuaiadh, Yoav Ossia, Erez Petrank, and Uri Silbershtein. An efficient parallel heap compaction algorithm. In OOPSLA 2004 [OOPSLA 20042004], pages 224–236.

[Acar *et al.*, 2006] Umut A. Acar, Guy E. Blelloch, M. Blume, and K. Tangwongsan. An experimental analysis of self-adjusting computation. In Schwartzbach and Ball [Schwartzbach and Ball2006], pages 96–107.

[Accetta *et al.*, 1986] M. Accetta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian, and M. Young. Mach: A new kernel foundation for Unix development. In *Proceedings of Usenix Summer Conference*. USENIX Association, July 1986.

[Aditya *et al.*, 1994] Shail Aditya, Christine Flood, and James Hicks. Garbage collection for strongly-typed languages using run-time type reconstruction. In LFP 1994 [LFP 19941994], pages 12–23.

[Adjih, 1996] Cédric Adjih. Mesure et caractérisation d'applications réparties. Master's thesis, Université Paris Sud, 1996.

[Adl-Tabatabai *et al.*, 2004] Ali-Reza Adl-Tabatabai, Richard L. Hudson, Mauricio J. Serrano, and Sreenivas Subramoney. Prefetch injection based on hardware monitoring and object metadata. In Pugh and Chambers [Pugh and Chambers2004], pages 267–276.

[Adl-Tabatai *et al.*, 1998] Ali-Reza Adl-Tabatai, Michal Cierniak, Guei-Yuan Leuh, Vihesh M. Parikh, and James M. Stichnoth. Fast effective code generation in a Just-In-Time Java compiler. In PLDI 1998 [PLDI 19981998], pages 280–290.

[Adve and Gharachorloo, 1995] Sarita V. Adve and Kourosh Gharachorloo. Shared memory consistency models: A tutorial. WRL Research Report 95/7, Digital Western Research Laboratory, September 1995.

[Adve and Gharachorloo, 1996] Sarita V. Adve and Kourosh Gharachorloo. Shared memory consistency models: A tutorial. *IEEE Computer*, 29(12):66–76, December 1996.

[Aerts, 1981] J. P. H. Aerts. Implementing SASL without garbage collection. EUT Report 81–WSK–05, Eindhoven University of Technology, 1981.

[Aftandilian and Guyer, 2008] Edward Aftandilian and Samuel Guyer. GC assertions: Using the garbage collector to check heap properties. In *MSPC 2008: ACM SIGPLAN Workshop on Memory Systems Performance and Correctness*, Seattle, WA, March 2008.

[Aftandilian and Guyer, 2009] Edward Aftandilian and Samuel Guyer. GC assertions: Using the garbage collector to check heap properties. In PLDI 2009 [PLDI 20092009], pages 235–244.

[Agaram *et al.*, 2006] Kartik Agaram, Steve Keckler, Calvin Lin, and Kathryn McKinley. Decomposing memory performance: Data structures and phases. In Petrank and Moss [Petrank and Moss2006], pages 95–103.

[Agarwal *et al.*, 1988] Anant Agarwal, Mark Horowitz, and John Hennessy. Cache performance of operating systems and multiprogramming workloads. *ACM Transactions on Computer Systems*, 6(4):393–431, November 1988.

[Agarwal, 1987] Anant Agarwal. *Analysis of Cache Performance for Operating Systems and Multiprogramming*. PhD thesis, Stanford University, Palo Alto, CA, May 1987. Available as Technical Report CSL-TR-87-332.

[Agesen and Detlefs, 1997] Ole Agesen and David Detlefs. Finding references in Java stacks. In Dickman and Wilson [Dickman and Wilson1997].

[Agesen and Garthwaite, 2000] Ole Agesen and Alex Garthwaite. Efficient object sampling via weak references. In Chambers and Hosking [Chambers and Hosking2000], pages 121–136.

[Agesen *et al.*, 1998] Ole Agesen, David Detlefs, and J. Eliot B. Moss. Garbage collection and local variable type-precision and liveness in Java virtual machines. In PLDI 1998 [PLDI 19981998], pages 269–279.

[Agesen, 1998] Ole Agesen. GC points in a threaded environment. Technical Report SMLI TR-98-70, Sun Microsystems Laboratories, Palo Alto, CA, 1998.

[Agesen, 1999] Ole Agesen. Space and time-efficient hashing of garbage-collected objects. *Theory and Practice of Object Systems*, 5(2):119–124, 1999.

[Aggarwal, 2002] Aneesh Aggarwal. Software caching *vs.* prefetching. In Boehm and Detlefs [Boehm and Detlefs2002], pages 157–162.

[Aggoun and Beldiceanu, 1990] A. Aggoun and N. Beldiceanu. Time stamps techniques for the trailed data in constraint logic programming systems. In *Séminaire de Programmation Logique de Trégastel, CNET, France*, pages 487–509, 1990.

[Agha, 1986] G. Agha. *Actors: A Model of Concurrent Computation in Distributed Systems*. MIT Press, 1986.

[Aho *et al.*, 1986] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques and Tools*. Addison-Wesley, 1986.

[Aho *et al.*, 1988] Alfred V. Aho, Brian W. Kernighan, and Peter J. Weinberger. *The AWK Programming Language*. Addison-Wesley, 1988.

[Aigner *et al.*, 2010] Martin Aigner, Andreas Haas, Christoph Kirsch, Hannes Payer, Andreas Schönegger, and Ana Sokolova. Short-term memory for self-collecting mutators. Technical Report 2010–03, University of Salzburg, April 2010.

[Aiken *et al.*, 1995a] Alex Aiken, Manuel Fähndrich, and Raph Levien. Better static memory management: Improving region-based analysis of higher-order languages. Technical report, University of California at Berkeley, 1995.

[Aiken *et al.*, 1995b] Alex Aiken, Manuel Fähndrich, and Raph Levien. Better static memory management: Improving region-based analysis of higher-order languages. In PLDI 1995 [PLDI 19951995], pages 174–185.

[Aït-Kaci, 1991] Hassan Aït-Kaci. The WAM: A (real) tutorial. In *Warren's Abstract Machine: A Tutorial Reconstruction*. MIT Press, 1991. Also Technical report 5, DEC Paris Research Laboratory, 1990.

[AIX, version 32] *Subroutines Overview*, General Programming Concepts, AIX version 3.2 edition, version 3.2.

[Akyürek and Salem, 1995] Sedat Akyürek and Kenneth Salem. Adaptive block rearrangement. *ACM Transactions on Computer Systems*, 13(2):95–121, May 1995.

[Albano and Morrison, 1992] Antonio Albano and Ronald Morrison, editors. *Proceedings of the Fifth International Workshop on Persistent Object Systems (September, 1992)*, Workshops in Computing, San Miniato, Italy, 1992. Springer.

[Albert *et al.*, 2007] Elvira Albert, Samir Genaim, and Miguel Gómez-Zamalloa. Heap space analysis for Java bytecode. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 105–116.

[Albert *et al.*, 2009] Elvira Albert, Samir Genaim, and Miguel Gómez-Zamalloa Gil. Live heap space analysis for languages with garbage collection. In Kolodner and Steele [Kolodner and Steele2009], pages 129–138.

[Albert *et al.*, 2010] Elvira Albert, Samir Genaim, and Miguel Gómez-Zamalloa. Parametric inference of memory requirements for garbage collected languages. In jan Vitek and Lea [jan Vitek and Lea2010], pages 121–130.

[Ali, 1998] K.A.M. Ali. A simple generational real-time garbage collection scheme. *Computing Paradigms and Computational Intelligence (New Generation Computing)*, 16(2), 1998.

[Allard and Hawkinson, 1991] J. R. Allard and L. B. Hawkinson. Real-time programming in Common Lisp. *Communications of the ACM*, 34(9):64–69, 1991.

[Allison, 1989] L. Allison. Circular programs and self-referential structures. *Software Practice and Experience*, 19(2):99–109, 1989.

[Almes *et al.*, 1983] Guy Almes, A. Borning, and E. Messinger. Implementing a Smalltalk-80 system on the Intel 432. In Krasner [Krasner1983], pages 175–187.

[Almes, 1980] Guy T. Almes. *Garbage collection in an Object-Oriented System*. PhD thesis, Carnegie Mellon University, 1980.

[Alon *et al.*, 1987] Noga Alon, Amnon Barak, and Udi Mander. On disseminating information reliably without broadcasting. In *Seventh International Conference on Distributed Computing Systems ICDCS97*, Berlin, September 1987. IEEE Press.

[Alonso and Appel, 1990] R. Alonso and Andrew W. Appel. Advisor for flexible working sets. In *Proceedings of the 1990 ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems. Boulder, May 22–25*, pages 153–162. ACM Press, 1990.

[Alpern *et al.*, 1999] Bowen Alpern, C. R. Attanasio, Anthony Cocchi, Derek Lieber, Stephen Smith, Ton Ngo, John J. Barton, Susan Flynn Hummel, Janice C. Sheperd, and Mark Mergen. Implementing Jalapeño in Java. In OOPSLA 1999 [OOPSLA 19991999], pages 314–324.

[Alpern *et al.*, 2000] Bowen Alpern, Dick Attanasio, John J. Barton, M. G. Burke, Perry Cheng, J.-D. Choi, Anthony Cocchi, Stephen J. Fink, David Grove, Michael Hind, Susan Flynn Hummel, D. Lieber, V. Litvinov, Mark Mergen, Ton Ngo, J. R. Russell, Vivek Sarkar, Manuel J. Serrano, Janice Shepherd, S. Smith, V. C. Sreedhar, H. Srinivasan, and J. Whaley. The Jalapeño virtual machine. *IBM Systems Journal*, 39(1), February 2000.

[Alpern *et al.*, 2002] Bowen Alpern, Maria Butrico, Anthony Cocchi, Julian Dolby, Stephen Fink, David Grove, and Ton Ngo. Experiences porting the Jikes RVM to Linux/IA32. In JVM 2002 [JVM 20022002].

[Amadio *et al.*, 2004] Roberto Amadio, Solange Coupet-Grimal, Silvano Dal Zilio, and Line Jakubiec. A functional scenerio for bytecode verification of space bounds. In SPACE 2004 [SPACE 20042004].

[Amamiya *et al.*, 1983] M. Amamiya, R. Hasegawa, and H. Mikami. List processing with a data flow machine. In *Proceedings of RIMS Symposia on Software Science and Engineering, 1980–1982*, volume 147 of *Lecture Notes in Computer Science*, pages 165–190, Kyoto, 1983. Springer-Verlag.

[Amsaleg *et al.*, 1994] Laurent Amsaleg, Michael Franklin, and Olivier Gruber. Efficient incremental garbage collection for workstation/server database systems. Technical Report CS–TR–3370, University of Maryland, November 1994. Also University of Maryland Institute for Advanced Computer Studies report UMIACS-TR-94-121.

[Amsaleg *et al.*, 1995a] Laurent Amsaleg, Paulo Ferreira, Michael Franklin, and Marc Shapiro. Evaluating garbage collectors for large persistent stores. In *OOPSLA'95 Workshop on Object Database Behavior, Benchmarks, and Performance*, Austin, TX, October 1995.

[Amsaleg *et al.*, 1995b] Laurent Amsaleg, Michael Franklin, and Olivier Gruber. Efficient incremental garbage collection for client–server object database systems. In *Twenty-first International Conference on Very Large Databases (VLDB95)*, Zurich, Switzerland, September 1995.

[Amsaleg *et al.*, 1999] Laurent Amsaleg, Michael Franklin, and Olivier Gruber. Garbage collection for a client-server persistent object store. *ACM Transactions on Computer Systems*, 17(3):153–201, 1999.

[Amsterdam, 2000] Jonathan Amsterdam. Use object pools to sidestep garbage collection. *Java Report*, pages 120–119 (really), September 2000.

[Anderson *et al.*, 1987] Wayne Anderson, William Galway, Robert Kessler, Herbert Melenk, and Winfried Neun. The implementation and optimisation of Portable Standard Lisp for the Cray. In *20th Annual Hawaii International Conference on Science Systems*, January 1987.

[Anderson *et al.*, 1991] Thomas E. Anderson, Henry M. Levy, Brian N. Bershad, and Edward D. Lazowska. The interaction of architecture and operating systems design. In ASPLOS 1991 [ASPLOS 19911991], pages 108–120.

[Anderson *et al.*, 2004] T. Anderson, M. Eng, N. Glew, B. Lewis, V. Menon, and J Stichnoth. Experience integrating a new compiler and garbage collector into Rotor. In *.NET Technologies'2004*, 2004.

[Anderson, 2010] Todd A. Anderson. Optimizations in a private nursery-based garbage collector. In jan Vitek and Lea [jan Vitek and Lea2010], pages 21–30.

[Andre, 1986] David L. Andre. Paging in Lisp programs. Master's thesis, University of Maryland, College Park, Maryland, 1986.

[Andreasson *et al.*, 2002] Eva Andreasson, Frank Hoffmann, and Olof Lindholm. To collect or not to collect? machine learning for memory management. In JVM 2002 [JVM 20022002].

[ANSI-C, 1989] American National Standards Institute. *American National Standard for Information Systems: Programming Language C*, December 1989.

[ANSI-C++, 1995] ANSI document X3J16/95–0087, ISO document WG21/N0618. *Draft Proposed International Standard for Information SSystems: Programming Language C++*, April 1995.

[Aonix,] The PERC virtual machine. http://www.aonix.com.

[Appel and Bendiksen, 1988] Andrew W. Appel and Aage Bendiksen. Vectorized garbage collection. Technical Report CS-TR-169-88, Department of Computer Science, Princeton University, July 1988.

[Appel and Bendiksen, 1989] Andrew W. Appel and Aage Bendiksen. Vectorized garbage collection. *The Journal of Supercomputing*, 3:151–160, 1989.

[Appel and Gonçalves, 1993] Andrew W. Appel and Marcelo J. R. Gonçalves. Hash-consing garbage collection. Technical Report CS-TR-412-93, Department of Computer Science, Princeton University, February 1993.

[Appel and Hanson, 1988] Andrew W. Appel and David R. Hanson. Copying garbage collection in the presence of ambiguous references. Technical Report CS-TR-162-88, Department of Computer Science, Princeton University, 1988.

[Appel and Li, 1991] Andrew W. Appel and Kai Li. Virtual memory primitives for user programs. In ASPLOS 1991 [ASPLOS 19911991], pages 96–107.

[Appel and Shao, 1994] Andrew W. Appel and Zhong Shao. An empirical and analytic study of stack vs. heap cost for languages with closures. Technical Report CS–TR–450–94, Department of Computer Science, Princeton University, March 1994.

[Appel and Shao, 1996] Andrew W. Appel and Zhong Shao. Empirical and analytic study of stack versus heap cost for languages with closures. *Journal of Functional Programming*, 6(1):47–74, January 1996.

[Appel *et al.*, 1988a] Andrew W. Appel, Bruce F. Duba, and David B. MacQueen. Profiling in the presence of optimization and garbage collection. Technical Report CS-TR-197-88, Department of Computer Science, Princeton University, November 1988.

[Appel *et al.*, 1988b] Andrew W. Appel, John R. Ellis, and Kai Li. Real-time concurrent collection on stock multiprocessors. *ACM SIGPLAN Notices*, 23(7):11–20, 1988.

[Appel, 1987] Andrew W. Appel. Garbage collection can be faster than stack allocation. *Information Processing Letters*, 25(4):275–279, 1987.

[Appel, 1988] Andrew W. Appel. Allocation without locking. Technical Report CS-TR-182-88, Department of Computer Science, Princeton University, September 1988.

[Appel, 1989a] Andrew W. Appel. Allocation without locking. *Software Practice and Experience*, 19(7), 1989. Short Communication.

[Appel, 1989b] Andrew W. Appel. Runtime tags aren't necessary. *Lisp and Symbolic Computation*, 2:153–162, 1989.

[Appel, 1989c] Andrew W. Appel. Simple generational garbage collection and fast allocation. *Software Practice and Experience*, 19(2):171–183, 1989.

[Appel, 1990] Andrew W. Appel. A runtime system. *Lisp and Symbolic Computation*, 3:343–380, 1990.

[Appel, 1991] Andrew W. Appel. Garbage collection. In Peter Lee, editor, *Topics in Advanced Language Implementation*, pages 89–100. MIT Press, 1991.

[Appel, 1992a] Andrew W. Appel. *The runtime system*, chapter 16, pages 205–214. Cambridge University Press, 1992.

[Appel, 1992b] Andrew W. Appel. Tutorial: Compilers and runtime systems for languages with garbage collection. In PLDI 1992 [PLDI 19921992].

[Appel, 1994] Andrew W. Appel. Emulating write-allocate on a no-write-allocate cache. Technical Report TR-459-94, Department of Computer Science, Princeton University, June 1994.

[Appel, 1997a] Andrew W. Appel. A better analytical model for the strong generational hypothesis, November 1997. Cited by [Stefanović, 1999].

[Appel, 1997b] Andrew W. Appel. *Modern Compiler Implementation in C: Basic Techniques*. Cambridge University Press, 1997.

[Appel, 1997c] Andrew W. Appel. *Modern Compiler Implementation in Java: Basic Techniques*. Cambridge University Press, 1997.

[Appel, 1997d] Andrew W. Appel. *Modern Compiler Implementation in ML: Basic Techniques*. Cambridge University Press, 1997.

[Appel, 1998a] Andrew W. Appel. *Modern Compiler Implementation in C*. Cambridge University Press, 1998.

[Appel, 1998b] Andrew W. Appel. *Modern Compiler Implementation in Java*. Cambridge University Press, 1998.

[Appel, 1998c] Andrew W. Appel. *Modern Compiler Implementation in ML*. Cambridge University Press, 1998.

[Appleby *et al.*, 1988] Karen Appleby, Mats Carlsson, Seif Haridi, and Dan Sahlin. Garbage collection for Prolog based on WAM. *Communications of the ACM*, 31(6):719–741, 1988.

[Armbruster *et al.*, 2007] Austin Armbruster, Jason Baker, Antonio Cunei, Chapman Flack, David Holmes, Filip Pizlo, Edward Pla, Marek Prochazka, and Jan Vitek. A real-time Java virtual machine with applications in avionics. *ACM Transactions on Embedded Computer Systems*, 7(1), 2007. Supersedes [Baker *et al.*, 2006].

[Armstrong and Virding, 1995] Joe Armstrong and Robert Virding. One-pass real-time generational mark-sweep garbage collection. In Baker [Baker1995a].

[Armstrong *et al.*, 1993] Joe Armstrong, Robert Virding, and Mike Williams. *Concurrent Programming in ERLANG*. Prentice-Hall, 1993.

[Arnborg, 1972] Stefan Arnborg. Storage administration in a virtual memory simulation system. *BIT*, 12(2):125–141, 1972.

[Arnborg, 1974] Stefan Arnborg. Optimal memory management in a system with garbage collection. *BIT*, 14(4):375–381, 1974.

[ASPLOS 1982, 1982] *Proceedings of the First International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGARCH Computer Architecture News 10(2), Palo Alto, CA, USA, March 1982.

[ASPLOS 1987, 1987] *Proceedings of the Second International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGARCH Computer Architecture News 15(5), Palo Alto, CA, USA, October 1987.

[ASPLOS 1989, 1989] *Proceedings of the Third International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGARCH Computer Architecture News 17(2), Boston, MA, USA, April 1989.

[ASPLOS 1991, 1991] *Proceedings of the Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGARCH Computer Architecture News 19(2), Santa Clara, CA, USA, April 1991.

[ASPLOS 1992, 1992] *Proceedings of the Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGPLAN Notices 27(9), Boston, MA, USA, October 1992.

[ASPLOS 1994, 1994] *Proceedings of the Sixth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGPLAN Notices 29(11), San Jose, CA, USA, October 1994.

[ASPLOS 1996, 1996] *Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGPLAN Notices 31(9), Cambridge, MA, USA, October 1996.

[ASPLOS 1998, 1998] *Proceedings of the Eighth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGPLAN Notices 33(11), San Jose, CA, USA, October 1998.

[ASPLOS 2000, 2000] *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGPLAN Notices 35(11), Cambridge, MA, USA, November 2000.

[ASPLOS 2002, 2002] *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGPLAN Notices 37(10), San Jose, CA, USA, October 2002.

[Assenmacher *et al.*, 1993] Holger Assenmacher, Thomas Breitbach, Peter Buhler, Volker Hübsch, and Reinhard Schwarz. PANDA — supporting distributed programming in C++. In Nierstras [Nierstras1993], pages 361–383.

[Atkey, 2004] Robert Atkey. A calculus for resource relationships. In SPACE 2004 [SPACE 20042004].

[Atkins and Nackman, 1988] Martin C. Atkins and Lee R. Nackman. The active deallocation of objects in object-oriented systems. *Software Practice and Experience*, 18(11):1073–1089, 1988.

[Atkins, 1989] Martin Atkins. *Implementation Techniques for Object-Oriented Systems*. PhD thesis, University of York, June 1989.

[Atkinson and Morrison, 1985] Malcolm P. Atkinson and Ronald Morrison. Procedures as persistent data objects. *ACM Transactions on Programming Languages and Systems*, 7(4):539–559, October 1985.

[Atkinson *et al.*, 1983] Malcolm P. Atkinson, P.J. Bailey, K.J. Chisholm, W.P. Cockshott, and Ron Morrison. An approach to persistent programming. *Computer Journal*, 26(4):360–365, December 1983.

[Atkinson *et al.*, 1988] Malcolm P. Atkinson, Peter Buneman, and Ronald Morrison, editors. *Proceedings of the First International Workshop on Persistent Object Systems (August, 1985)*, Data Types and Persistence, Appin, Scotland, 1988. Springer-Verlag.

[Atkinson *et al.*, 1989] Russ Atkinson, Alan Demers, Carl Hauser, Christian Jacobi, Peter Kessler, and Mark Weiser. Experiences creating a portable Cedar. In PLDI 1989 [PLDI 19891989], pages 322–329.

[Atkinson *et al.*, 1995] Malcolm P. Atkinson, David Maier, and Véronique Benzaken, editors. *Proceedings of the Sixth International Workshop on Persistent Object Systems (September, 1994)*, Workshops in Computing, Tarascon, Provence, France, 1995. Springer and British Computer Society.

[Attanasio *et al.*, 2001] Clement R. Attanasio, David F. Bacon, Anthony Cocchi, and Stephen Smith. A comparative evaluation of parallel garbage collectors. In *Fourteenth Annual Workshop on Languages and Compilers for Parallel Computing*, volume 2624 of *Lecture Notes in Computer Science*, Cumberland Falls, KT, August 2001. Springer-Verlag.

[Attardi and Flagella, 1984] Giuseppe Attardi and Tito Flagella. A customisable memory management framework. In ECOOP 1984 [ECOOP 19841984], pages 320–343.

[Attardi and Flagella, 1994] Giuseppe Attardi and Tito Flagella. A customisable memory management framework. Technical Report TR-94-010, International Computer Science Institute, Berkeley, 1994. Also Proceedings of the USENIX C++ Conference, Cambridge, MA, 1994.

[Attardi and Flagella, 1996] Giuseppe Attardi and Tito Flagella. Memory management in the PoSSo solver. *Journal of Symbolic Computation*, 21(3):293–311, 1996.

[Attardi *et al.*, 1995] Giuseppe Attardi, Tito Flagella, and Pietro Iglio. Performance tuning in a customizable collector. In Baker [Baker1995a].

[Attardi *et al.*, 1998] Giuseppe Attardi, Tito Flagella, and Pietro Iglio. A customisable memory management framework for C++. *Software Practice and Experience*, 28(11):1143–1183, November 1998.

[Auerbach *et al.*, 2007a] Joshua Auerbach, David F. Bacon, Bob Blainey, Perry Cheng, Michael Dawson, Mike Fulton, David Grove, Darren Hart, and Mark Stoodley. Design and implementation of a comprehensive real-time Java virtual machine. In *Proceedings of the 7th ACM & IEEE International Conference on Embedded Software*, pages 249–258, Salzburg, Austria, 2007.

[Auerbach *et al.*, 2007b] Joshua Auerbach, David F. Bacon, Florian Bömers, and Perry Cheng. Real-time music synthesis in Java using the Metronome garbage collector. In *Proceedings of the International Computer Music Conference*, Lecture Notes in Computer Science, Copenhagen, Denmark, 2007.

[Auerbach *et al.*, 2007c] Joshua Auerbach, David F. Bacon, Daniel T. Iercan, Christopher M. Kirsch, V.T. Rajan, Harald Röck, , and Rainer Trummer. Java takes flight: Time-portable real-time programming with Exotasks. In *Proceedings of ACM Conference on Languages, Compilers, and Tools for Embedded Systems*, pages 31–62, 2007.

[Auerbach *et al.*, 2008] Joshua Auerbach, David F. Bacon, Perry Cheng, David Grove, Ben Biron, Charlie Gracie, Bill McCloskey, Aleksandar Micic, and Ryan Sciampacone. Tax-and-spend: Democratic scheduling for real-time garbage collection. In *Proceedings of the 7th ACM International Conference on Embedded software*, pages 245–254, Atlanta, GA, 2008.

[Augenstein and Tenenbaum, 1986] Moshe J. Augenstein and Aaron M. Tenenbaum. *Data Structures using Pascal*. Prentice-Hall, Englewood Cliffs, N. J., second edition, 1986.

[Augusteijn, 1987] Lex Augusteijn. Garbage collection in a distributed environment. In de Bakker et al. [de Bakker *et al.*1987], pages 75–93.

[Augustsson, 1984] Lennart Augustsson. A compiler for lazy ML. In Steele [Steele1984], pages 218–227.

[Austin *et al.*, 1994] Todd M. Austin, Scott E. Breachand, and Gurindar S. Sohi. Efficient detection of all pointer and array access errors. In PLDI 1994 [PLDI 19941994], pages 290–301.

[Axford, 1990] Thomas H. Axford. Reference counting of cyclic graphs for functional programs. *Computer Journal*, 33(5):466–470, 1990.

[Azagury *et al.*, 1998] Alain Azagury, Elliot K. Kolodner, Erez Petrank, and Zvi Yehudai. Combining card marking with remembered sets: How to save scanning time. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 10–19.

[Azagury *et al.*, 1999] Alain Azagury, Elliot K. Kolodner, and Erez Petrank. A note on the implementation of replication-based garbage collection for multithreaded applications and multiprocessor environments. *Parallel Processing Letters*, 1999.

[Azatchi and Petrank, 2003] Hezi Azatchi and Erez Petrank. Integrating generations with advanced reference counting garbage collectors. In *Proceedings of the 12th International Conference on Compiler Construction, CC 2003*, volume 2622 of *Lecture Notes in Computer Science*, pages 185–199, Warsaw, Poland, May 2003. Springer.

[Azatchi and Petrank, 2006] Hezi Azatchi and Erez Petrank. Integrating generations with advanced reference counting garbage collectors. *Concurrency and Computation: Practice and Experience*, 18(9):959–995, 2006.

[Azatchi *et al.*, 2003] Hezi Azatchi, Yossi Levanoni, Harel Paz, and Erez Petrank. An on-the-fly mark and sweep garbage collector based on sliding view. In OOPSLA 2003 [OOPSLA 20032003].

[Azimi *et al.*, 2007] Reza Azimi, Livio Soares, Michael Stumm, Thomas Walsh, and Angela Demke Brown. PATH: Page access tracking to improve memory management. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 31–42.

[Azul, 2008] Azul. Pauseless garbage collection. White paper AWP–005–020, Azul Systems Inc., July 2008.

[Azul, 2010] Azul. Comparison of virtual memory manipulation metrics. White paper, Azul Systems Inc., 2010.

[Babaoglu and Ferrari, 1983] Ozalp Babaoglu and Domenico Ferrari. Two-level replacement decisions in paging stores. *IEEE Transactions on Computers*, C-32(12):1151–1159, December 1983.

[Babaoglu and Marzullo, 1993] Ozalp Babaoglu and Keith Marzullo. Consistent global states of distributed systems: Fundamental concepts and mechanisms. In S. Mullender, editor, *Distributed Systems*, pages 55–96. Addison-Wesley, 1993.

[Babaoglu and Marzullo, 1996] Özalp Babaoglu and Keith Marzullo, editors. *Proceedings or the Tenth International Workshop on Distributed Algorithms*, volume 1151 of *Lecture Notes in Computer Science*, Bologna, Italy, October 1996. Springer.

[Back *et al.*, 1983] R. J. R. Back, Heikki Mannila, and Kari-Jouko Räihä. Derivation of efficient DAG marking algorithms. In POPL 1983 [POPL 19831983], pages 20–27.

[Bacon and Diwan, 2004] David F. Bacon and Amer Diwan, editors. *Proceedings of the Fourth International Symposium on Memory Management*, Vancouver, Canada, October 2004. ACM Press.

[Bacon and Rajan, 2001] David F. Bacon and V.T. Rajan. Concurrent cycle collection in reference counted systems. In Knudsen [Knudsen2001].

[Bacon and Sweeney, 1996] David F. Bacon and Peter F. Sweeney. Fast static analysis of C++ virtual function calls. In OOPSLA 1996 [OOPSLA 19961996], pages 324–34.

[Bacon *et al.*, 2001] David F. Bacon, Clement R. Attanasio, Han Bok Lee, V. T. Rajan, and Stephen E. Smith. Java without the coffee breaks: A nonintrusive multiprocessor garbage collector. In PLDI 2001 [PLDI 20012001], pages 92–103.

[Bacon *et al.*, 2002] David F. Bacon, Stephen Fink, and David Grove. Space- and time-efficient implementation of the Java object model. In ECOOP 2002 [ECOOP 20022002], pages 111–132.

[Bacon *et al.*, 2003a] David F. Bacon, Perry Cheng, and V.T. Rajan. Controlling fragmentation and space consumption in the Metronome, a real-time garbage collector for Java. In LCTES 2003 [LCTES 20032003], pages 81–92.

[Bacon *et al.*, 2003b] David F. Bacon, Perry Cheng, and V.T. Rajan. The Metronome, a simpler approach to garbage collection in real-time systems. In *Proceedings of the OTM 2003 Workshops*, pages 466–478, 2003.

[Bacon *et al.*, 2003c] David F. Bacon, Perry Cheng, and V.T. Rajan. A real-time garbage collector with low overhead and consistent utilization. In POPL 2003 [POPL 20032003], pages 285–298.

[Bacon *et al.*, 2004a] David F. Bacon, Perry Cheng, and David Grove. Garbage collection for embedded systems. In *International Conference on Embedded Software (EMSOFT'04)*, pages 125–136. ACM Press, 2004.

[Bacon *et al.*, 2004b] David F. Bacon, Perry Cheng, and V. T. Rajan. A unified theory of garbage collection. In OOPSLA 2004 [OOPSLA 20042004], pages 50–68.

[Bacon *et al.*, 2005] David F. Bacon, Perry Cheng, David Grove, and Martin T. Vechev. Syncopation: Generational real-time garbage collection in the Metronome. In LCTES 2005 [LCTES 20052005], pages 183–192.

[Bacon *et al.*, 2006] David F. Bacon, Perry Cheng, Daniel Frampton, David Grove Matthias Hauswirth, and V.T. Rajan. On-line visualization and analysis of real-time systems with TuningFork. In *Proceedings of the Fifteenth International Conference on Compiler Construction (CC'06)*, volume 3923 of *Lecture Notes in Computer Science*, pages 96–100, Vienna, Austria, March 2006.

[Bacon, 2005] David F. Bacon. Real-time garbage collection. *ACM Queue*, 5(1):40–49, February 2005.

[Baden, 1982] Scott B. Baden. High performance reclamation in an object-based memory system. Master's thesis, Computer Science Division, Department of EECS, University of California, Berkeley, June 1982.

[Baden, 1983] Scott B. Baden. Low-overhead storage reclamation in the Smalltalk-80 virtual machine. In Krasner [Krasner1983], pages 331–342.

[Baden, 1984] Scott B. Baden. High performance storage reclamation in an object-based memory system. CSD 84-167, University of California, Berkeley, 1984.

[Baecker, 1970] H. D. Baecker. Implementing the Algol–68 heap. *BIT*, 10(4):405–414, 1970.

[Baecker, 1972] H. D. Baecker. Garbage collection for virtual memory computer systems. *Communications of the ACM*, 15(11):981–986, November 1972.

[Baecker, 1973] H. D. Baecker. Aspects of reference locality in list structures in virtual memory. *Software Practice and Experience*, 3(3):245–254, 1973.

[Baecker, 1975] H. D. Baecker. Areas and record classes. *Computer Journal*, 18(3):223–226, August 1975.

[Baer and Fries, 1977] Jean-Loup Baer and M. Fries. On the efficiency of some list marking algorithms. In B. Gilchrist, editor, *Information Processing 77, Toronto*, pages 751–6. North-Holland, August 1977.

[Baer and Sager, 1976] Jean-Loup Baer and Gary R. Sager. Dynamic improvement of locality in virtual memory systems. *IEEE Transactions on Software Engineering*, SE-2(1):54–62, March 1976.

[Bagherzadeh *et al.*, 1991] Nader Bagherzadeh, S-l. Heng, and C-l. Wu. A parallel asynchronous garbage collection algorithm for distributed systems. *IEEE Transactions on Knowledge and Data Engineering*, 3(1):100–107, March 1991.

[Bagherzadeh, 1987] Nader Bagherzadeh. *Distributed Resource Management: Garbage Collection*. PhD thesis, University of Texas at Austin, 1987.

[Baker and Hewitt, 1977a] Henry G. Baker and Carl E. Hewitt. The incremental garbage collection of processes. AI memo 454, MIT Press, December 1977.

[Baker and Hewitt, 1977b] Henry G. Baker and Carl E. Hewitt. The incremental garbage collection of processes. *ACM SIGPLAN Notices*, 12(8):55–59, August 1977.

[Baker *et al.*, 1985] Brenda Baker, E. G. Coffman, and D. E. Willard. Algorithms for resolving conflicts in dynamic storage allocation. *Journal of the ACM*, 32(2):327–343, April 1985.

[Baker *et al.*, 2006] Jason Baker, Antonio Cunei, Chapman Flack, Filip Pizlo, Marek Prochazka, Jan Vitek, Austin Armbuster, Edward Pla, and David Holmes. Real-time Java in avionics applications. In *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2006.

[Baker *et al.*, 2007] Jason Baker, Antonio Cunei, Filip Pizlo, and Jan Vitek. Accurate garbage collection in uncooperative environments with lazy pointer stacks. In *Proceedings of the International Conference on Compiler Construction (CC07)*, 2007.

[Baker *et al.*, 2009] Jason Baker, Antonio Cunei, Tomas Kalibera, Filip Pizlo, and Jan Vitek. Accurate garbage collection in uncooperative environments revisited. *Concurrency and Computation: Practice and Experience*, 2009. Supersedes [Baker *et al.*, 2007].

[Baker-Finch, 1992] C. A. Baker-Finch. Relevance and contraction: A logical basis for strictness and sharing analysis, 1992. Submitted to Journal of Functional Programming.

[Baker, 1978a] Henry G. Baker. Actor systems for real-time computation. Technical Report MIT Rep. TR–197, Laboratory for Computer Science, March 1978.

[Baker, 1978b] Henry G. Baker. List processing in real-time on a serial computer. *Communications of the ACM*, 21(4):280–94, 1978. Also AI Laboratory Working Paper 139, 1977.

[Baker, 1979] Henry G. Baker. Optimizing allocation and garbage collection of spaces in MaclLisp. In Winston and Brown, editors, *Artificial Intelligence: An MIT Perspective*. MIT Press, 1979.

[Baker, 1980] Henry G. Baker. The paging behavior of the Cheney list copying algorithm. Technical report, University of Rochester Computer Science Department, 1980.

[Baker, 1989] Henry G. Baker. Garbage collection in Ada. Ada-9X Revision Request 643, Ada Joint Program Office, October 1989.

[Baker, 1990] Henry G. Baker. Unify and conquer (garbage, updating, aliasing, ...) in functional languages. In LFP 1990 [LFP 19901990], pages 218–226.

[Baker, 1991] Henry G. Baker. Cache-conscious copying collection. In Wilson and Hayes [Wilson and Hayes1991a].

[Baker, 1992a] Henry G. Baker. The buried binding and dead binding problems of Lisp 1.5: Sources of incomparability in garbage collector measurements. *Lisp Pointers*, 4(2):11–19, April 1992.

[Baker, 1992b] Henry G. Baker. CONS should not CONS its arguments, or a lazy alloc is a smart alloc. *ACM SIGPLAN Notices*, 27(3), March 1992.

[Baker, 1992c] Henry G. Baker. Lively linear Lisp — 'look Ma, no garbage!'. *ACM SIGPLAN Notices*, 27(9):89–98, August 1992.

[Baker, 1992d] Henry G. Baker. Thermodynamics of garbage collection. In Bekkers and Cohen [Bekkers and Cohen1992].

[Baker, 1992e] Henry G. Baker. The Treadmill, real-time garbage collection without motion sickness. *ACM SIGPLAN Notices*, 27(3):66–70, March 1992.

[Baker, 1993a] Henry G. Baker. The boyer benchmark meets linear logic. *Lisp Pointers*, 6(4):3–10, October 1993.

[Baker, 1993b] Henry G. Baker. 'Infant mortality' and generational garbage collection. *ACM SIGPLAN Notices*, 28(4), April 1993.

[Baker, 1993c] Henry G. Baker. Safe and leak-proof resource management using Ada83 limited types. *ACM Ada Leters*, 13(5):32–42, September 1993.

[Baker, 1994a] Henry G. Baker. CONS should not CONS its arguments, part ii: Cheney on the M.T.A. comp.lang.scheme.c newsgroup, February 1994.

[Baker, 1994b] Henry G. Baker. Minimising reference count updating with deferred and anchored pointers for functional data structures. *ACM SIGPLAN Notices*, 29(9), September 1994.

[Baker, 1995a] Henry G. Baker, editor. *Proceedings of the International Workshop on Memory Management*, volume 986 of *Lecture Notes in Computer Science*, Kinross, Scotland, 27–29September 1995. Springer.

[Baker, 1995b] Henry G. Baker. Use-once variables and linear objects — storage management, reflection and multi-threading. *ACM SIGPLAN Notices*, 30(1), 1995.

[Bakewell and Runciman, 2000] Adam Bakewell and Colin Runciman. A model for comparing the space usage of lazy evaluators. In *Second International Conference on Principles and Practice of Declarative Programming*, Montreal, September 2000.

[Bakewell, 2001] Adam Bakewell. Looking for leaks. In SPACE 2001 [SPACE 20012001].

[Bal and Tanenbaum, 1991] Henri E. Bal and Andrew S. Tanenbaum. Distributed programming with shared data. *Computer Languages*, 16(2):129–146, 1991.

[Bal *et al.*, 1992] Henri E. Bal, M. Frans Kaashoek, and Andrew S. Tanenbaum. Orca: A language for parallel programming of distributed systems. *ACM Transactions on Software Engineering*, 18(3):190–205, 1992.

[Balakrishnan *et al.*, 2005] Saisanthosh Balakrishnan, Ravi Rajwar, Mike Upton, and Konrad Lai. The impact of performance asymmetry in emerging multicore architectures. In *32nd AnnualInternational Symposium on Computer Architecture(ISCA'05)*, pages 506–517, 2005.

[Ball and Larus, 1992] Thomas Ball and James Larus. Optimal profiling and tracing of programs. In POPL 1992 [POPL 19921992], pages 59–70.

[Ballard and Shirron, 1983] Stoney Ballard and Stephen Shirron. The design and implementation of VAX/Smalltalk-80. In Krasner [Krasner1983], pages 127–150.

[Banach, 1994] R Banach. Term graph rewriting and garbage collection using opfibrations. *Theoretical Computer Science*, 131(1):29–94, August 1994.

[Banerjee *et al.*, 1999] A. Banerjee, N. Heintze, and J.G. Riecke. Region analysis and the polymorphic lambda calculus. In *Proceedings of the Fourteenth Annual IEEE Symposium on Logic in Computer Science*, pages 88–97, Trento, Italy, July 1999. IEEE Press.

[Barabash and Petrank, 2010] Katherine Barabash and Erez Petrank. Tracing garbage collection on highly parallel platforms. In jan Vitek and Lea [jan Vitek and Lea2010], pages 1–10.

[Barabash *et al.*, 2001] Katherine Barabash, N. Buchbinder, Tamar Domani, Elliot Kolodner, Yoav Ossia, S.S. Pinter, J. Shepherd, R.Sivan, and V. Umansky. Mostly accurate stack scanning. In JVM 2001 [JVM 20012001], pages 153–170.

[Barabash *et al.*, 2003] Katherine Barabash, Yoav Ossia, and Erez Petrank. Mostly concurrent garbage collection revisited. In OOPSLA 2003 [OOPSLA 20032003].

[Barabash *et al.*, 2005] Katherine Barabash, Ori Ben-Yitzhak, Irit Goft, Elliot K. Kolodner, Victor Leikehman, Yoav Ossia, Avi Owshanko, and Erez Petrank. A parallel, incremental, mostly concurrent garbage collector for servers. *ACM Transactions on Programming Languages and Systems*, 27(6):1097–1146, November 2005.

[Barach *et al.*, 1982] David R. Barach, David H. Taenzer, and Robert E. Wells. A technique for finding storage allocation errors in C-language programs. *ACM SIGPLAN Notices*, 17(5):16–23, May 1982.

[Barbacci, 1971] M. Barbacci. A LISP processor for C.ai. Memo CMU-CS-71-103, Carnegie Mellon University, 1971.

[Barklund and Millroth, 1986a] Jonas Barklund and Høakan Millroth. Garbage cut. Technical Report 38, Uppsala University, 1986.

[Barklund and Millroth, 1986b] Jonas Barklund and Høakan Millroth. Garbage cut for garbage collection of iterative Prolog programs. In *Proceedings — 1986 Symposium on Logic Programming. Salt Lake City, Sept 22–25*, pages 276–283. IEEE Press, 1986.

[Barklund, 1987] Jonas Barklund. A garbage collection algorithm for Tricia. Technical Report 37B, Uppsala University, 1987.

[Barnes *et al.*, 1997] Nick Barnes, Richard Brooksby, David Jones, Gavin Matthews, Pekka P. Pirinen, Nick Dalton, and P. Tucker Withington. A proposal for a standard memory management interface. In Dickman and Wilson [Dickman and Wilson1997].

[Barnett, 1979] Jeff A. Barnett. Garbage collection versus swapping. *ACM SIGOPS Operating Systems Review*, 13(3), 1979.

[Barrett and Zorn, 1993a] David A. Barrett and Benjamin Zorn. Garbage collection using a dynamic threatening boundary. Computer Science Technical Report CU-CS-659-93, University of Colorado, July 1993.

[Barrett and Zorn, 1993b] David A. Barrett and Benjamin G. Zorn. Using lifetime predictors to improve memory allocation performance. In PLDI 1993 [PLDI 19931993], pages 187–196.

[Barrett and Zorn, 1995] David A. Barrett and Benjamin Zorn. Garbage collection using a dynamic threatening boundary. In PLDI 1995 [PLDI 19951995], pages 301–314.

[Barroso *et al.*, 1998] L.A. Barroso, K. Gharachorloo, and E. Bugnion. Memory system characterization of commercial workloads. In *25th Annual International Symposium on Computer Architecture*, pages 3–14, 1998.

[Barth, 1977] Jeffrey M. Barth. Shifting garbage collection overhead to compile time. *Communications of the ACM*, 20(7):513–518, July 1977.

[Bartlett, 1988] Joel F. Bartlett. Compacting garbage collection with ambiguous roots. Technical Report 88/2, DEC Western Research Laboratory, Palo Alto, CA, February 1988. Also in Lisp Pointers 1, 6 (April–June 1988), 2–12.

[Bartlett, 1989a] Joel F. Bartlett. Mostly-Copying garbage collection picks up generations and C++. Technical note, DEC Western Research Laboratory, Palo Alto, CA, October 1989. Sources available in ftp://ftp.digital.com/pub/DEC/CCgc.

[Bartlett, 1989b] Joel F. Bartlett. SCHEME->C: a portable Scheme-to-C compiler. Technical report, DEC Western Research Laboratory, Palo Alto, CA, January 1989.

[Bartlett, 1990] Joel F. Bartlett. A generational, compacting collector for C++. In Jul and Juul [Jul and Juul1990].

[Bastani *et al.*, 1988] F. B. Bastani, S. S. Iyengar, and I. L. Yen. Concurrent maintenance of data-structures in a distributed environment. *Computer Journal*, 31(2):165–174, 1988.

[Bates *et al.*, 1982] Raymond L. Bates, David Dyer, and Johannes A. G. M. Koomen. Implementation of Interlisp on VAX. In LFP 1982 [LFP 19821982], pages 81–87.

[Batson and Brundage, 1977] Alan P. Batson and R. E. Brundage. Segment sizes and lifetimes in ALGOL 60 programs. *Communications of the ACM*, 20(1):36–44, January 1977.

[Batson, 1976] Alan Batson. Program behavior at the symbolic level. *IEEE Computer*, pages 21–26, November 1976.

[Bauer and Wössner, 1982] F. L. Bauer and H. Wössner. *Algorithmic Language and Program Development*. Springer-Verlag, 1982.

[Baumgartner and Wah, 1991] K. M. Baumgartner and B. W. Wah. Computer scheduling algorithms — past, present, and future. *Information Sciences*, pages 319–345, September 1991.

[Bawden *et al.*, 1977] A. Bawden, Richard Greenblatt, J. Holloway, T. Knight, David A. Moon, and D. Weinreb. Lisp machine progress report. Technical Report Memo 444, A.I. Lab, MIT, Cambridge, MA, August 1977.

[Baylor *et al.*, 2000] S.J. Baylor, M. Devarakonda, S. Fink, E. Gluzberg, M. Kalantar, P. Muttineni, E. Barsness, S. Munroe, R. Arora, and R. Dimpsey. Java server benchmarks. *IBM Systems Journal*, 39(1), 2000.

[Bays, 1977] C. Bays. A comparison of next-fit, first-fit and best-fit. *Communications of the ACM*, 20(3):191–192, March 1977.

[Beaudoing and Queinnec, 1991] Barbara Beaudoing and Christian Queinnec. Mark-DURING-Sweep: A real-time garbage collector. In Aarts et al. [Aarts and others1991].

[Beaudoing, 1991] Barbara Beaudoing. *Recycler-en-Marquant: Un Algorithme de Gestion de Mémoire en Temps Réel, Étude et Implantation*. PhD thesis, Université de Paris VI, 1991.

[Becerra *et al.*, 2003] Yolande Becerra, Toni Cortes, Jordi Garcia, and Nacho Navarro. Evaluating the importance of virtual memory for Java. In *Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software*. IEEE Press, 2003.

[Beck, 1982] Leland L. Beck. A dynamic storage allocation technique based on memory residence time. *Communications of the ACM*, 25(10):714–724, October 1982.

[Beckerle and Ekanadham, 1986] Michael J. Beckerle and Kattamuri Ekanadham. Distributed garbage collection with no global synchronisation. Research Report RC 11667 (#52377), IBM Corp., January 1986.

[Beebee, Jr. and Rinard, 2001] William S. Beebee, Jr. and Martin Rinard. An implementation of scoped memory for real-time Java. In Henzinger and Kirsch [Henzinger and Kirsch2001], pages 289–305.

[Beemster, 1990] Marcel Beemster. Back-end aspects of a portable POOL–X implementation. In Pierre America, editor, *Parallel Database Systems (PRISMA) Workshop Proceedings*, volume 503 of *Lecture Notes in Computer Science*, pages 193–228. Springer-Verlag, 1990.

[Beg and van Beek, 2010] Mirza Beg and Peter van Beek. A graph theoretic approach to cache-conscious placement of data for direct mapped caches. In jan Vitek and Lea [jan Vitek and Lea2010], pages 113–120.

[Bekkers and Cohen, 1992] Yves Bekkers and Jacques Cohen, editors. *Proceedings of the International Workshop on Memory Management*, volume 637 of *Lecture Notes in Computer Science*, St Malo, France, 17–19 September 1992. Springer.

[Bekkers and Ungaro, 1991] Yves Bekkers and L. Ungaro. Implementing parallel garbage collection for Prolog. In A. Voronkov, editor, *Russian Conference on Logic Programming*, volume 592 of *Lecture Notes in Computer Science*, 1991.

[Bekkers *et al.*, 1983] Yves Bekkers, B. Canet, Olivier Ridoux, and L. Ungaro. A short note on garbage collection in Prolog interpreters. *Logic Programming Newsletter*, 5, 1983.

[Bekkers *et al.*, 1984] Yves Bekkers, B. Canet, Olivier Ridoux, and L. Ungaro. A memory management machine for Prolog interpreters. In Tärnlund [Tärnlund1984], pages 343–351.

[Bekkers *et al.*, 1985] Yves Bekkers, B. Canet, Olivier Ridoux, and L. Ungaro. A memory management machine for Prolog. *Informatique–85, Symposium Soviéto-Français, Tallin*, pages 111–117, 1985.

[Bekkers *et al.*, 1986] Yves Bekkers, B. Canet, Olivier Ridoux, and L. Ungaro. MALI: A memory with a real-time garbage collector for implementing logic programming languages. In *3rd Symposium on Logic Programming*. IEEE Press, 1986.

[Bekkers *et al.*, 1992] Yves Bekkers, Olivier Ridoux, and L. Ungaro. A survey on memory management for logic programming. In Bekkers and Cohen [Bekkers and Cohen1992].

[Belotsky, 2003] George Belotsky. C++ memory management: From fear to triumph. O'Reilly linuxdevcenter.com, July 2003.

[Ben-Ari, 1982] Mordechai Ben-Ari. On-the-fly garbage collection: New algorithms inspired by program proofs. In M. Nielsen and E. M. Schmidt, editors, *Automata, languages and programming. Ninth colloquium*, pages 14–22, Aarhus, Denmark, July 12–16 1982. Springer-Verlag.

[Ben-Ari, 1984] Mordechai Ben-Ari. Algorithms for on-the-fly garbage collection. *ACM Transactions on Programming Languages and Systems*, 6(3):333–344, July 1984.

[Ben-Yitzhak *et al.*, 2002] Ori Ben-Yitzhak, Irit Goft, Elliot Kolodner, Kean Kuiper, and Victor Leikehman. An algorithm for parallel incremental compaction. In Boehm and Detlefs [Boehm and Detlefs2002], pages 100–105.

[Benes, 1981] V. E. Benes. Models and problems of dynamic storage allocation. In *Applied Probability and Computer Science — the Interface*. Institute of Management Science and Operations Research Society of America, January 1981.

[Bengtsson and Magnusson, 1990] Mats Bengtsson and Boris Magnusson. Real-time compacting garbage collection. In Jul and Juul [Jul and Juul1990].

[Bengtsson, 1990] Mats Bengtsson. Real-time compacting garbage collection algorithms. Licentiate thesis, Department of Computer Science, Lund University, 1990.

[Bennet *et al.*, 1990] J. Bennet, J. Carter, and W. Zwaenepoel. Munin: Distributed shared memory based on type-specific memory coherence. In PPOPP 1990 [PPOPP 19901990], pages 168–176.

[Bennett, 1987] J. K. Bennett. The design and implementation of distributed Smalltalk. In OOPSLA 1987 [OOPSLA 19871987], pages 318–330.

[Benson, 1997] Peter Benson. The memory manager for the Aurora Java Virtual Machine testbed. In Dickman and Wilson [Dickman and Wilson1997].

[Benton and Torp-Smith, 2006] Nick Benton and Noah Torp-Smith. Abstracting allocation: The New new thing. In SPACE 2006 [SPACE 20062006], pages 108–110.

[Berger and Blumofe, 1999] Emery D. Berger and Robert D. Blumofe. Hoard: A fast, scalable, and memory-efficient allocator for shared-memory multiprocessors. Technical Report UTCS TR99-22, University of Texas at Austin, November 1999.

[Berger *et al.*, 2000] Emery Berger, Kathryn McKinley, Robert Blumofe, and Paul Wilson. Hoard: A scalable memory allocator for multithreaded applications. In ASPLOS 2000 [ASPLOS 20002000], pages 117–128.

[Berger *et al.*, 2001] Emery D. Berger, Benjamin G. Zorn, and Kathryn S. McKinley. Composing high-performance memory allocators. In PLDI 2001 [PLDI 20012001], pages 114–124.

[Berger *et al.*, 2002] Emery D. Berger, Benjamin G. Zorn, and Kathryn S. McKinley. Reconsidering custom memory allocation. In OOPSLA 2002 [OOPSLA 20022002].

[Bergstein, 1988] Steven H. Bergstein. Best-case caching in a symbolic multiprocessor. Bachelor's thesis, Massachusetts Institute of Technology EECS Department, Cambridge, MA, February 1988.

[Berkeley and Bobrow, 1974] E. C. Berkeley and Daniel G. Bobrow, editors. *The Programming Language LISP: Its Operation and Applications*. Information International, Inc., Cambridge, MA, fourth edition, 1974.

[Berlea *et al.*, 2000] Alexandru Berlea, Sorin Cotofana, Irina Athanasiu, John Glossner, and Stamatis Vassiliadis. Garbage collection for the Delft Java processor. In *Proceedings of 18th IASTED International Conference, Applied Informatics*, Innsbruck, 2000.

[Berry *et al.*, 1978] D. M. Berry, L. M. Chirica, J. B. Johnston, D. F. Martin, and Sorkin A. Time required for garbage collection in retention block-structured languages. *Journal of Computer Information Science*, 7(1):361–404, 1978.

[Berry, 2002] Robert F. Berry. The business importance of Java garbage collection, June 2002. Invited talk.

[Bertziss, 1965] A. T. Bertziss. A note on the storage of strings. *Communications of the ACM*, 8(8):512–513, August 1965.

[Berztiss, 1975] A. T. Berztiss. *Data Structures Theory and Practice*. Academic Press, second edition, 1975.

[Besson *et al.*, 2007] Frédéric Besson, Thomas Jensen, and Tiphaine Turpin. Computing stack maps with interfaces. Research report PI 1879, INRIA, Université Rennes I, Institut National des Sciences Appliquées de Rennes, 2007.

[Besson *et al.*, 2008] Frédéric Besson, Thomas Jensen, and Tiphaine Turpin. Computing stack maps with interfaces. In ECOOP 2008 [ECOOP 20082008].

[Betteridge, 1973] Terry Betteridge. An analytical storage allocation model. *Acta Informatica*, 3:101–122, 1973.

[Betteridge, 1982] Terry Betteridge. *An Algebraic Analysis of Storage Fragmentation*. UMI Research Press, Ann Arbor, Michigan, 1982.

[Bevan, 1987] David I. Bevan. Distributed garbage collection using reference counting. In de Bakker et al. [de Bakker *et al.*1987], pages 176–187.

[Bevan, 1988] David I. Bevan. Efficient reference counting solution to the distributed garbage collection problem. In *PARLE: Conference onParallel Architectures and Languages Europe*, pages 179–192, Eindhoven, June 1988. Also in Parallel Computing 9(2) Jan 1988–1989. p 179–192.

[Bevemyr and Lindgren, 1994] Johan Bevemyr and Thomas Lindgren. A simple and efficient copying garbage collector for Prolog. In *PLILP94 International Symposium on Programming Language Implementation and Logic Programming*, pages 88–101, 1994.

[Bevemyr, 1995] Johan Bevemyr. A generational parallel copying garbage collector for shared memory Prolog. Technical Report 117, Uppsala University, October 1995.

[Beyer and Buneman, 1979] Eric Beyer and Peter Buneman. A space efficient dynamic allocation algorithm for queuing messages. *ACM Transactions on Programming Languages and Systems*, 1(2):287–294, October 1979.

[Bézivin *et al.*, 1987] J. Bézivin, J.-M. Hullot, P. Cointe, and Henry Lieberman, editors. *Proceedings of 1987 European Conference on Object-Oriented Programming, ECOOP87*, volume 276 of *Lecture Notes in Computer Science*. Springer-Verlag, June 1987.

[Bhatia *et al.*, 2006] Sapan Bhatia, Charles Consel, and Julia Lawall. Memory-manager/scheduler co-design: Optimizing event-driven servers to improve cache behavior. In Petrank and Moss [Petrank and Moss2006], pages 104–114.

[Biagioni *et al.*, 1994] Edoardo Biagioni, Robert Harper, Peter Lee, and Brian Milnes. Signatures for a network protocol stack: A systems application of Standard ML. In LFP 1994 [LFP 19941994].

[Bielak and Sarkis, 1999] Richard Bielak and Jean-Pierre Sarkis. Implementing a distributed garbage collector for OO databases. In *TOOLS USA '99 Technology of Object-Oriented Languages and Systems*, Santa Barbara, CA, August 1999.

[Bigler *et al.*, 1985] B. M. Bigler, S. J. Allan, and Rod R. Oldehoeft. Parallel dynamic storage allocation. In *1985 International Conference on Parallel Processing*, pages 272–275, 1985.

[Biliris, 1992] Alexandros Biliris. An efficient database storage structure for large dynamic objects. In *Proceedings, IEEE Data Engineering Conference*, pages 301–308, Phoenix, Arizona, February 1992. IEEE Press.

[Bingham *et al.*, 1993] Tim Bingham, Nancy Hobbs, and Dave Husson. Experiences developing and using an object-oriented library for program manipulation. In OOPSLA 1993 [OOPSLA 19931993].

[Birkedal *et al.*, 1996] Lars Birkedal, Mads Tofte, and Magnus Vejlstrup. From region inference to von Neumann machines via region representation inference. In POPL 1996 [POPL 19961996].

[Birkedal *et al.*, 2004] Lars Birkedal, Noah Torpe-Smith, and John C. Reynolds. Local reasoning about a copying garbage collector. In POPL 2004 [POPL 20042004], pages 220–231.

[Birman, 1987] K. Birman. Exploiting virtual synchrony in distributed systems. *ACM SIGOPS Operating Systems Review*, 21(5):123–138, November 1987.

[Birrell and Needham, 1978] Andrew D. Birrell and Roger M. Needham. An asynchronous garbage collector for the CAP filing system. *ACM SIGOPS Operating Systems Review*, 12(2):31–33, April 1978.

[Birrell *et al.*, 1993] Andrew Birrell, David Evers, Greg Nelson, Susan Owicki, and Edward Wobber. Distributed garbage collection for network objects. Technical Report 116, DEC Systems Research Center, 130 Lytton Avenue, Palo Alto, CA 94301, December 1993.

[Birrell *et al.*, 1994a] Andrew Birrell, Greg Nelson, Susan Owicki, and Edward Wobber. Network objects. Technical Report 115, DEC Systems Research Center, Palo Alto, CA, February 1994.

[Birrell *et al.*, 1994b] Andrew Birrell, Greg Nelson, Susan Owicki, and Edward Wobber. Network objects. In *Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles*, pages 217–230, Asheville, NC, 1994. ACM Press.

[Birrell *et al.*, 1995] Andrew Birrell, Greg Nelson, Susan Owicki, and Edward Wobber. Network objects. *Software Practice and Experience*, 25(4):87–130, December 1995. Also appeared as SRC Research Report 115.

[Bishop, 1975] Peter B. Bishop. Garbage collection in a very large address space. Working paper 111, AI Laboratory, MIT, Cambridge, MA, September 1975.

[Bishop, 1977] Peter B. Bishop. *Computer Systems with a Very Large Address Space and Garbage Collection*. PhD thesis, MIT Laboratory for Computer Science, May 1977. Technical report MIT/LCS/TR–178.

[Björnerstedt, 1989] Anders Björnerstedt. Secondary storage garbage collection for decentralized object-based systems. In D. Tsichritzis, editor, *Object Oriented Development*, pages 277–319. Centre Universitaire d'Informatique, University of Geneva, July 1989.

[Björnerstedt, 1990] Anders Björnerstedt. *Secondary Storage Garbage Collection for Decentralized Object-Based Systems*. PhD thesis, Royal Institute of Technology and Stockholm University, Sweden, June 1990. Technical Report 77.

[Bjornsson and Shrira, 2002] Magnus Bjornsson and Liuba Shrira. BuddyCache: High performance object storage for collaborative strong-consistency applications in a WAN. In OOPSLA 2002 [OOPSLA 20022002].

[Black *et al.*, 1986] Andrew Black, Norman Hutchinson, Eric Jul, and Henry Levy. Object structure in the Emerald system. In OOPSLA 1986 [OOPSLA 19861986], pages 78–86.

[Black *et al.*, 1987] Andrew Black, Norman Hutchinson, Eric Jul, Henry Levy, and Larry Carter. Distribution and abstract types in Emerald. *ACM Transactions on Software Engineering*, 13(1):65–76, January 1987.

[Black, 2005] Andrew Black, editor. *Proceedings of 19th European Conference on Object-Oriented Programming, ECOOP 2005*, Lecture Notes in Computer Science, Glasgow, July 2005. Springer-Verlag.

[Blackburn and Hosking, 2004] Stephen M. Blackburn and Tony Hosking. Barriers: Friend or foe? In Bacon and Diwan [Bacon and Diwan2004], pages 143–151.

[Blackburn and McKinley, 2002] Stephen M. Blackburn and Kathryn S. McKinley. In or out? putting write barriers in their place. In Boehm and Detlefs [Boehm and Detlefs2002], pages 175–184.

[Blackburn and McKinley, 2003] Stephen M. Blackburn and Kathryn S. McKinley. Ulterior reference counting: Fast garbage collection without a long wait. In OOPSLA 2003 [OOPSLA 20032003].

[Blackburn and McKinley, 2008] Stephen Blackburn and Kathryn McKinley. Immix garbage collection: Mutator locality, fast collection, and space efficiency. In Gupta and Amarasinghe [Gupta and Amarasinghe2008], pages 22–32.

[Blackburn *et al.*, 2001] Stephen M. Blackburn, Sharad Singhai, Matthew Hertz, Kathryn S. McKinley, and J. Eliot B. Moss. Pretenuring for Java. In OOPSLA 2001 [OOPSLA 20012001], pages 342–352.

[Blackburn *et al.*, 2002] Stephen M. Blackburn, Richard E. Jones, Kathryn S. McKinley, and J. Eliot B. Moss. Beltway: Getting around garbage collection gridlock. In PLDI 2002 [PLDI 20022002], pages 153–164.

[Blackburn *et al.*, 2003] Stephen M. Blackburn, Perry Cheng, and Kathryn S. McKinley. A garbage collection design and bakeoff in JMTk: An extensible Java memory management toolkit. Technical Report TR–CS–03–02, Australian National University, February 2003.

[Blackburn *et al.*, 2004a] Stephen M. Blackburn, Perry Cheng, and Kathryn S. McKinley. Myths and realities: The performance impact of garbage collection. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, ACM SIGMETRICS Performance Evaluation Review 32(1), pages 25–36, New York, NY, June 2004.

[Blackburn *et al.*, 2004b] Stephen M. Blackburn, Perry Cheng, and Kathryn S. McKinley. Oil and water? High performance garbage collection in Java with MMTk. In ICSE 2004 [ICSE 20042004], pages 137–146.

[Blackburn *et al.*, 2006a] Stephen Blackburn, Robin Garner, Chris Hoffman, Asjad M. Khan, Kathryn S. McKinley, Rotem Bentzur, Amer Diwan, Daniel Feinberg, Samuel Z. Guyer, Martin Hirzel, Antony Hosking, Maria Jump, Han Lee, J. Eliot B. Moss, Aashish Phansalkar, Darko Stefanović, Thomas VanDrunen, Daniel von Dincklage, and Ben Wiederman. The DaCapo benchmarks: Java benchmarking development and analysis (extended version). Technical report, The DaCapo Group, 2006.

[Blackburn *et al.*, 2006b] Stephen Blackburn, Robin Garner, Kathryn S. McKinley, Amer Diwan, Samuel Z. Guyer, Antony Hosking, J. Eliot B. Moss, Darko Stefanović, et al. The DaCapo benchmarks: Java benchmarking development and analysis. In OOPSLA 2006 [OOPSLA 20062006], pages 169–190.

[Blackburn *et al.*, 2007] Stephen M. Blackburn, Matthew Hertz, Kathryn S. Mckinley, J. Eliot B. Moss, and Ting Yang. Profile-based pretenuring. *ACM Transactions on Programming Languages and Systems*, 29(1):1–57, 2007.

[Blackburn *et al.*, 2008] S. M. Blackburn, K. S. McKinley, R. Garner, C. Hoffmann, A. M. Khan, R. Bentzur, A. Diwan, D. Feinberg, D. Frampton, S. Z. Guyer, M. Hirzel, A. Hosking, M. Jump, H. Lee, J. E. B. Moss, A. Phansalkar, D. Stefanović, T. VanDrunen, D. von Dincklage, and B. Wiedermann. Wake up and smell the coffee: Evaluation methodology for the 21st century. *Communications of the ACM*, 51(8):83–89, 2008.

[Blanchet, 1996] Bruno Blanchet. Garbage collection statique. Dea report, INRIA, Rocquencourt, September 1996.

[Blanchet, 1998] Bruno Blanchet. Escape analysis: Correctness proof, implementation and experimental results. In POPL 1998 [POPL 19981998], pages 25–37.

[Blanchet, 1999] Bruno Blanchet. Escape analysis for object oriented languages: Application to Java. In OOPSLA 1999 [OOPSLA 19991999], pages 20–34.

[Blanchet, 2003] Bruno Blanchet. Escape analysis for Java: Theory and practice. *ACM Transactions on Programming Languages and Systems*, 25(6):712–775, November 2003.

[Blau, 1983] Ricki Blau. Paging on an object-oriented personal computer for Smalltalk. In *ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Minneapolis*. ACM Press, August 1983. Also appears as Technical Report UCB/CSD 83/125, University of California at Berkeley, Computer Science Division (EECS).

[Blelloch and Cheng, 1999] Guy E. Blelloch and Perry Cheng. On bounding time and space for multiprocessor garbage collection. In PLDI 1999 [PLDI 19991999], pages 104–117.

[Blelloch *et al.*, 2001] Guy E. Blelloch, Perry Cheng, and Phillip B. Gibbons. Room synchronizations. In *Proceedings of the Thirteenth ACM Symposium on Parallel Algorithms and Architectures*, pages 122–133, Heraklion, Crete, Greece, July 2001.

[Blelloch *et al.*, 2003] Guy E. Blelloch, Perry Cheng, and Phillip B. Gibbons. Scalable room synchronizations. *Theory of Computing Systems*, 36(5):397–430, September 2003.

[Blondel *et al.*, 1997] Xavier Blondel, Paulo Ferreira, and Marc Shapiro. PerDIS PPF case study: Fitting a distributed garbage collection algorithm to a persistent distributed store architecture. In Dickman and Wilson [Dickman and Wilson1997].

[Blondel *et al.*, 1998] Xavier Blondel, Paulo Ferreira, and Marc Shapiro. Implementing garbage collection in the PerDiS system. Submitted for publication, 1998.

[Blondel, 1999] Xavier Blondel. Report on the scalability of garbage collection. Technical report, Esprit, 1999. Deliverable TC.1.3-B, PerDiS project.

[Blondel, 2000] Xavier Blondel. *Gestion de Méta-donneés de la Mémoire dans un Environment Réparti Persistant Transactionnel à Grande Échelle: l'Exemple de PerDiS*. PhD thesis, Conservatoire National des Arts et Métiers, September 2000.

[Bobrow and Clark, 1979] Daniel G. Bobrow and Douglas W. Clark. Compact encodings of list structure. *ACM Transactions on Programming Languages and Systems*, 1(2):266–286, October 1979.

[Bobrow and Murphy, 1967] Daniel G. Bobrow and Daniel L. Murphy. Structure of a LISP system using two-level storage. *Communications of the ACM*, 10(3):155–159, March 1967.

[Bobrow and Murphy, 1968] Daniel G. Bobrow and Daniel L. Murphy. A note on the efficiency of a LISP computation in a paged machine. *Communications of the ACM*, 11(8):558–560, August 1968.

[Bobrow and Raphael, 1964] Daniel G. Bobrow and Bertram Raphael. A comparison of list-processing computer languages. *Communications of the ACM*, 7(4):231–240, April 1964.

[Bobrow, 1968] Daniel G. Bobrow. Storage management in Lisp. *Symbol manipulation languages and techniques*, 1968.

[Bobrow, 1975] Daniel G. Bobrow. A note on hash linking. *Communications of the ACM*, 18(7):413–15, July 1975.

[Bobrow, 1980] Daniel G. Bobrow. Managing re-entrant structures using reference counts. *ACM Transactions on Programming Languages and Systems*, 2(3):269–273, July 1980.

[Bode *et al.*, 1993] Arndt Bode, Mike Reeve, and Gottfried Wolf, editors. *PARLE'93 Parallel Architectures and Languages Europe*, volume 694 of *Lecture Notes in Computer Science*, Munich, June 1993. Springer-Verlag.

[Boehm and Adve, 2008] Hans-Juergen Boehm and Sarita V. Adve. Foundations of the C++ concurrency memory model. In Gupta and Amarasinghe [Gupta and Amarasinghe2008], pages 68–78.

[Boehm and Chase, 1992] Hans-Juergen Boehm and David R. Chase. A proposal for garbage-collector-safe C compilation. *Journal of C Language Translation*, pages 126–141, 1992.

[Boehm and Demers, 1985] Hans-Juergen Boehm and Alan Demers. Implementing Russell. Technical Report COMP TR85-25, Rice University, 1985.

[Boehm and Detlefs, 2002] Hans-J. Boehm and David Detlefs, editors. *Proceedings of the Third International Symposium on Memory Management*, ACM SIGPLAN Notices 38(2 supplement), Berlin, Germany, June 2002.

[Boehm and Grove, 2006] Jans-J. Boehm and David Grove, editors. *Proceedings of the Second ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, Ottawa, Canada, June 2006.

[Boehm and Hederman, 1988] Hans-Juergen Boehm and Lucy Hederman. Storage allocation optimization in a compiler for Russell. Submitted for publication, July 1988.

[Boehm and Shao, 1993] Hans-Juergen Boehm and Zhong Shao. Inferring type maps during garbage collection. In Moss et al. [Moss *et al.*1993].

[Boehm and Spertus, 2009] Hans-Juergen Boehm and Mike Spertus. Garbage collection in the next C++ standard. In Kolodner and Steele [Kolodner and Steele2009], pages 30–38.

[Boehm and Weiser, 1988] Hans-Juergen Boehm and Mark Weiser. Garbage collection in an uncooperative environment. *Software Practice and Experience*, 18(9):807–820, 1988.

[Boehm *et al.*, 1991a] Hans-Juergen Boehm, Alan J. Demers, and Scott Shenker. Mostly parallel garbage collection. In PLDI 1991 [PLDI 19911991], pages 157–164.

[Boehm *et al.*, 1991b] Hans-Juergen Boehm, Eliot Moss, Joel Bartlett, and David R. Chase. Panel discussion: Conservative vs. accurate garbage collection. In Wilson and Hayes [Wilson and Hayes1991a]. Summary appears in Wilson and Hayes' OOPSLA'91 GC workshop report.

[Boehm, 1991a] Hans-Juergen Boehm. Hardware and operating system support for conservative garbage collection. In Cabrera et al. [Cabrera *et al.*1991], pages 61–67.

[Boehm, 1991b] Hans-Juergen Boehm. Simple GC-safe compilation. In Wilson and Hayes [Wilson and Hayes1991a].

[Boehm, 1993] Hans-Juergen Boehm. Space efficient conservative garbage collection. In PLDI 1993 [PLDI 19931993], pages 197–206.

[Boehm, 1995] Hans-Juergen Boehm. Dynamic memory allocation and garbage collection. *Computers in Physics*, 9(3):297–303, May/June 1995.

[Boehm, 1996] Hans-Juergen Boehm. Simple garbage-collector safety. In PLDI 1996 [PLDI 19961996], pages 89–98.

[Boehm, 2000a] Hans-Juergen Boehm. Fast multiprocessor memory allocation and garbage collection. Technical Report HPL-2000-165, HP Laboratories, Palo Alto, December 2000.

[Boehm, 2000b] Hans-Juergen Boehm. Reducing garbage collector cache misses. In Chambers and Hosking [Chambers and Hosking2000], pages 59–64.

[Boehm, 2002] Hans-Juergen Boehm. Bounding space usage of conservative garbage collectors. In POPL 2002 [POPL 20022002].

[Boehm, 2003] Hans-Juergen Boehm. Destructors, finalizers, and synchronization. In POPL 2003 [POPL 20032003].

[Boehm, 2004] Hans-Juergen Boehm. The space cost of lazy reference counting. In POPL 2004 [POPL 20042004], pages 210–219.

[Bogda and Hölzle, 1999] Jeff Bogda and Urs Hölzle. Removing unnecessary synchronization in Java. In OOPSLA 1999 [OOPSLA 19991999], pages 35–46.

[Boizumault, 1986] P. Boizumault. A general model to implement *dif* and *freeze*. In Schapiro [Schapiro1986].

[Bond and McKinley, 2006] Michael Bond and Kathryn McKinley. Bell: Bit-encoding online memory leak detection. In Shen and Martonosi [Shen and Martonosi2006], pages 61–72.

[Bond and McKinley, 2008] Michael Bond and Kathryn McKinley. Tolerating memory leaks. In OOPSLA 2008 [OOPSLA 20082008].

[Boquist, 1999] U. Boquist. *Code Optimisation Techniques for Lazy Functional Languages*. PhD thesis, Chalmers University of Technology, Gothenburg, Sweden, April 1999.

[Borg *et al.*, 2006] Andrew Borg, Andy Wellings, Christopher Gill, and Ron K. Cytron. Real-time memory management: Life and times. In *18th Euromicro Conference on Real-Time Systems (ECRTS'06)*, pages 237–250, Dresden, Germany, July 2006.

[Borman, 2002a] Sam Borman. Sensible sanitation — understanding the IBM Java garbage collector, part 1: Object allocation. *IBM developerWorks*, August 2002.

[Borman, 2002b] Sam Borman. Sensible sanitation — understanding the IBM Java garbage collector, part 2: Garbage collection. *IBM developerWorks*, August 2002.

[Borman, 2002c] Sam Borman. Sensible sanitation — understanding the IBM Java garbage collector, part 3: `verbosegc` and command-line. *IBM developerWorks*, September 2002.

[Bornat *et al.*, 2004] Richard Bornat, Cristiano Calcagno, and Peter O'Hearn. Local reasoning, separation, and aliasing. In SPACE 2004 [SPACE 20042004].

[Bowman *et al.*, 1993] Howard Bowman, John Derrick, and Richard E. Jones. Modelling garbage collection algorithms. In *International Workshop on Concurrency in Computational Logic, City University, London, 13 December 1993*, December 1993.

[Boyapati *et al.*, 2003] Chandrasekhar Boyapati, Alexandru Salcianu, William Beebee, Jr., and Martin Rinard. Ownership types for safe region-based memory management in Real-Time Java. In PLDI 2003 [PLDI 20032003], pages 324–337.

[Boyer and Moore, 1972] R. S. Boyer and J. S. Moore. The sharing of structure in theorem-proving programs. In B. Meltzer and Donald Michie, editors, *Machine Intelligence*, pages 101–116. Edinburgh University Press, 1972.

[Boysen and Shah, 1993] P. Boysen and P. Shah. Reducing object storage requirements in a multiuser environment. *Software Practice and Experience*, 23(2):243–253, March 1993.

[Bozman *et al.*, 1984] G. Bozman, W. Buco, T. P. Daly, and W. H. Tetzlaff. Analysis of free storage algorithms — revisited. *IBM Systems Journal*, 23(1):44–64, 1984.

[Bozman, 1984] Gerald Bozman. The software lookaside buffer reduces search overhead with linked lists. *Communications of the ACM*, 27(3):222–227, March 1984.

[Braberman *et al.*, 2008] Víctor Braberman, Federico Fernández, Diego Garbervetsky, and Sergio Yovine. Parametric prediction of heap memory requirements. In Jones and Blackburn [Jones and Blackburn2008], pages 141–150.

[Branquart and Lewi, 1971] P. Branquart and J. Lewi. A scheme of storage allocation and garbage collection for Algol–68. In Peck [Peck1971], pages 198–238.

[Brecht *et al.*, 2001] Tim Brecht, Eshrat Arjomandi, Chang Li, and Hang Pham. Controlling garbage collection and heap growth to reduce the execution time of Java applications. In OOPSLA 2001 [OOPSLA 20012001].

[Brecht *et al.*, 2006] Tim Brecht, Eshrat Arjomandi, Chang Li, and Hang Pham. Controlling garbage collection and heap growth to reduce the execution time of Java applications. *ACM Transactions on Programming Languages and Systems*, 28(5), September 2006.

[Brega and Rivera, 2000] Roberto Brega and Gabrio Rivera. Dynamic memory management with garbage collection for embedded applications. In *Proc. of the USENIX Workshop on Industrial Experiences with Systems Software (WIESS 2000)*, San Diego, CA, October 2000.

[Brent, 1989] R. P. Brent. Efficient implementation of the first-fit strategy for dynamic storage allocation. *ACM Transactions on Programming Languages and Systems*, 11(3):388–403, July 1989.

[Brisset, 1992] P. Brisset. *Compilation de λProlog*. PhD thesis, Université de Rennes, 1992.

[Britton, 1975] Dianne Ellen Britton. Heap storage management for the programming language Pascal. Master's thesis, University of Arizona, 1975.

[Broberg *et al.*,] Magnus Broberg, Daniel Häggander, Per Lidén, and Lars Lundberg. Improving the performance of multiprocessor memory management in Java. *Java Report*. To appear.

[Brodie-Tyrrell *et al.*, 2004] W. Brodie-Tyrrell, H. Detmold, K.E. Falkner, and David S. Munro. Garbage collection for storage-oriented clusters. In V. Estivill-Castro, editor, *Proceedings of Australian Computer Science Conference*, volume 26 of CRPIT, pages 99–108. Australian Computer Society, 2004.

[Bromley, 1980] A. G. Bromley. Memory fragmentation in buddy methods for dynamic storage allocation. *Acta Informatica*, 14(2):107–117, August 1980.

[Brooks *et al.*, 1982] Rodney A. Brooks, Richard P. Gabriel, and Guy L. Steele. S–1 Common Lisp implementation. In LFP 1982 [LFP 19821982], pages 108–113.

[Brooks *et al.*, 1983] Rodney A. Brooks, Richard P. Gabriel, and Guy L. Steele. LISP-in-LISP: High performance and portability. In *International Joint Conference on Artifical Intelligence*, volume 2, pages 845–849. IJCAI, 1983.

[Brooks, 1984] Rodney A. Brooks. Trading data space for reduced time and code space in real-time garbage collection on stock hardware. In Steele [Steele1984], pages 256–262.

[Brooksby, 2002] Richard Brooksby. The Memory Pool System: Thirty person-years of memory management development goes open source. Technical report, Ravenbrook Limited, January 2002.

[Brownbridge, 1984] David R. Brownbridge. *Recursive Structures in Computer Systems*. PhD thesis, University of Newcastle upon Tyne, September 1984.

[Brownbridge, 1985] David R. Brownbridge. Cyclic reference counting for combinator machines. In Jouannaud [Jouannaud1985].

[Broy and Pepper, 1982] Manfred Broy and Peter Pepper. Combining algebraic and algorithmic reasoning: An approach to the Schorr–Waite algorithm. *ACM Transactions on Programming Languages and Systems*, 4(3):362–381, July 1982.

[Bruha, 1987] Ivan Bruha. Representation of structures and garbage collection in McMaster POPLOG. Technical Report 88-01, McMaster University, Department of Computer Science and Systems,Canada, 1987.

[Brus *et al.*, 1987] T. Brus, M. J. C. D. van Eekelen, M. J. Plasmeijer, and H. P. Barendregt. Clean: A language for functional graph rewriting. In Kahn [Kahn1987], pages 364–384.

[Bruynooghe and Janssens, 1988] Maurice Bruynooghe and G. Janssens. An instance of abstract interpretation integrating type and mode inferencing. In *5th International Conference and Symposium on Logic Programming*, pages 669–683. MIT Press, 1988.

[Bruynooghe, 1980] Maurice Bruynooghe. Garbage collection in Prolog implementations. *Logic Programming*, pages 83–98, 1980. Also in Workshop on Logic Programming, Debrecen, Hungary, 1980.

[Bruynooghe, 1982] Maurice Bruynooghe. A note on garbage collection in Prolog interpreters. In ICLP 1982 [ICLP 19821982], pages 52–55.

[Bruynooghe, 1984] Maurice Bruynooghe. Garbage collection in Prolog implementations. In J. A. Campbell, editor, *Implementations of Prolog*, pages 259–267. Ellis-Horwood, 1984.

[Bruynooghe, 1986] Maurice Bruynooghe. Compile-time garbage collection. Report CW43, Katholieke Universiteit of Leuven, 1986.

[Bruynooghe, 1987] Maurice Bruynooghe. Compile-time garbage collection or How to transform programs in an assignment-free language into code with assignments. In L. G. L. T. Meertens, editor, *Program specification and transformation. The IFIP TC2/WG 2.1 Working Conference, Bad Tolz, Germany*, pages 113–129. North-Holland, Amsterdam, April 15–17, 1986 1987.

[Budimlic *et al.*, 2009] Zoran Budimlic, Aparna M. Chandramowlishwaran, Kathleen Knobe, Geoff N. Lowney, Vivek Sarkar, and Leo Treggiari. Declarative aspects of memory management in the Concurrent Collections parallel programming model. In *DAMP 2009: Workshop on Declarative Aspects of Multicore Programming*, Savannah, GA, January 2009.

[Burdy, 2001] L. Burdy. B vs. Coq to prove a garbage collector. In R. J. Boulton and P. B. Jackson, editors, *Fourteenth International Conference on Theorem Proving in Higher Order Logics: Supplemental Proceedings*, pages 85–97, September 2001. Report EDI–INF–RR–0046, Division of Informatics, University of Edinburgh.

[Burgess *et al.*, 1999] Peter Burgess, Nigel Perry, and Robert Pointon. The concurrent Massey Hope+C functional language system. Report, Massey University, 1999. Available on request from Nigel Perry.

[Burnett, 1987] T. D. Burnett. Parallel reduction architecture. In *Highly Parallel Computers, Proceedings of the IFIP WG 10.3 Working Conference for Numerical and Signal Processing Applications. Sophia Antipolis, France*, pages 41–57, Amsterdam, March 24–26 1987. North-Holland.

[Burton and Simpson, 2000] F. Warren Burton and David J. Simpson. Memory requirements for parallel programs. *Parallel Computing*, 26(13—14):1739—1763, 2000.

[Burton, 1976] F. Warren Burton. A buddy system variation for disk storage allocation. *Communications of the ACM*, 19(7):416–417, July 1976.

[Burton, 2000] Joshua W. Burton. Garbage collection on the run. *Dr. Dobb's Journal*, 311:46–53, April 2000.

[Butler, 1986] Margaret H. Butler. Storage reclamation for object oriented database systems: a summary of the expected costs. In *Proceedings — 1986 International Workshop on Object-Oriented Database Systems. Pacific Grove, Ca, USA, Sept 23–26*, pages 210–211. IEEE Press, 1986.

[Butler, 1987] Margaret H. Butler. Storage reclamation in object oriented database systems. In *Proceedings of the ACM SIGMOD, San Francisco*, pages 410–425, May 1987.

[Butters, 2007] Albin M. Butters. Total cost of ownership: A comparison of C/C++ and Java. Technical report, Evans Data Corporation, June 2007.

[Buytaert *et al.*, 2004] Dries Buytaert, Kris Venstermans, Lieven Eeckhout, and Koen De Bosschere. Garbage collection scheduling. In *Program Acceleration through Application and Architecture Driven Code Transformations: Symposium Proceedings*, pages 47–49, 2004.

[Buytaert *et al.*, 2005] Dries Buytaert, Kris Venstermans, Lieven Eeckhout, and Koen De Bosschere. Garbage collection hints. In *HiPEAC 2005 International Conference on High Performance Embedded Architectures and Compilers*, Barcelona, November 2005.

[Cabrera *et al.*, 1991] Luis-Felipe Cabrera, Vincent Russo, and Marc Shapiro, editors. *International Workshop on Object Orientation in Operating Systems*, Palo Alto, CA, October 1991. IEEE Press.

[Cabrera *et al.*, 1992] Luis-Felipe Cabrera, Vince Russo, and Marc Shapiro, editors. *International Workshop on Object Orientation in Operating Systems*, Paris, September 1992. IEEE Press.

[Cahill *et al.*, 1993] Vinny Cahill, Seán Baker, Chris Horn, and Gradimir Stavovic. The Amadeus GRT — generic runtime support for distributed persistent programming. In OOPSLA 1993 [OOPSLA 19931993]. Technical report TCD–CS–93–37.

[Cai and Wellings, 2003] H. Cai and A. J. Wellings. Towards a high-integrity real-time Java virtual machine. In *On the Move to Meaningfull Internet Systems 2003: Workshop on Java Technologies for Real-Time and Embedded Systems*, volume LNCS 2889, pages 319–334. Springer, 2003.

[Calcagno *et al.*, 2003] Cristiano Calcagno, Peter O'Hearn, and Richard Bornat. Program logic and equivalence in the presence of garbage collection. *Theoretical Computer Science*, 298(3), 2003.

[Calcagno, 2001] Cristiano Calcagno. Program logics in the presence of garbage collection. In SPACE 2001 [SPACE 20012001].

[Calder *et al.*, 1994] Brad Calder, Dirk Grunwald, and Benjamin Zorn. Quantifying behavioral differences between C and C++ programs. *Journal of Programming Languages*, 2(4):313–351, 1994.

[Calder *et al.*, 1998] Brad Calder, Chandra Krintz, S. John, and T. Austin. Cache-conscious data placement. In ASPLOS 1998 [ASPLOS 19981998], pages 139–149.

[Campbell, 1971] John A. Campbell. A note on an optimal-fit method for dynamic allocation of storage. *Computer Journal*, 14(1):7–9, February 1971.

[Campbell, 1974] John A. Campbell. Optimal use of storage in a simple model of garbage collection. *Information Processing Letters*, 3(2):374, November 1974.

[Campin and Atkinson, 1986] J. Campin and Malcolm Atkinson. A persistent store garbage collector with statistical facilities. Technical report, Universities of Glasgow and St Andrews, 1986.

[Campos and Hanson, 1993] Alvaro E. Campos and David R. Hanson. Garbage collection in EZ. In R. Baeza-Yates, editor, *Proceedings of Thirteenth International Conference on Computer Sceince*, La Serena, Chile, 1993. Plenum Press.

[Campos, 1993] Alvaro E. Campos. *Distributed, Garbage-Collected, Persistent, Virtual Address Spaces*. PhD dissertation, Princeton University, Department of Computer Science, June 1993.

[Cann and Oldehoeft, 1988] D. C. Cann and Rod R. Oldehoeft. Reference count and copy elimination for parallel applicative computing. Technical Report CS–88–129, Department of Computer Science, Colorado State University, Fort Collins, CO, 1988.

[Cann *et al.*, 1992] D. C. Cann, J. T. Feo, A. D. W. Bohoem, and Rod R. Oldehoeft. *SISAL Reference Manual: Language Version 2.0*, 1992.

[Cannarozzi *et al.*, 2000] Dante Cannarozzi, Michael P. Plezbert, and Ron Cytron. Contaminated garbage collection. In PLDI 2000 [PLDI 20002000], pages 264–273.

[Caplinger, 1988] Michael Caplinger. A memory allocator with garbage collection for C. In *USENIX Winter Conference*, pages 323–323. USENIX Association, 1988.

[Cardelli *et al.*, 1988] Luca Cardelli, James Donahue, Lucille Glassman, Mick Jordan, Bill Kalsow, and Greg Nelson. Modula-3 report (revised). Research Report PRC–131, DEC Systems Research Center and Olivetti Research Center, 1988.

[Cardelli *et al.*, 1992] Luca Cardelli, James Donahue, Lucille Glassman, Mick Jordan, Bill Kalsow, and Greg Nelson. Modula-3 language definition. *ACM SIGPLAN Notices*, 27(8):15–42, August 1992.

[Cardelli, 1984] Luca Cardelli. Compiling a functional language. In Steele [Steele1984], pages 208–217.

[Cardelli, 1991] Luca Cardelli. Typeful programming. In E.J. Neuhold and M. Paul, editors, *Formal Description of Programming Concepts*. Springer-Verlag, 1991. Revised 1 January, 1993.

[Carlini and Rendina, 1992] Giuliano Carlini and Susan Rendina. Garbage collection for C programs. *Dr. Dobb's Journal*, 17(11), November 1992.

[Carlsson *et al.*, 1990] Svante Carlsson, Christer Mattsson, and Mats Bengtsson. A fast expected-time compacting garbage collection algorithm. In Jul and Juul [Jul and Juul1990].

[Carlsson *et al.*, 1991] Svante Carlsson, Christer Mattsson, Mats Bengtsson, and Patricio Poblete. A new compacting garbage collection algorithm with a good average-case performance. In *STACS-91*, 1991.

[Carlsson, 1987] M. Carlsson. Freeze, indexing and other implementation issues in the WAM. In Lassez [Lassez1987], pages 40–58.

[Carrick and Cooper, 1987] Raymund Carrick and Richard Cooper, editors. *Proceedings of the Second International Workshop on Persistent Object Systems*, Appin, Scotland, August 1987. Universities of Glasgow and St Andrews.

[Carter, 1989] A. M. Carter. Cascade: Hardware for high/variable precision arithmetic. In *Proceedings: 9th Symposium on Computer Arithmetic (ARITH9). Santa Monica, Sept 6–8*, pages 184–191. IEEE Press, 1989.

[Case, 1991] Brian Case. PA–RISC provides rich instruction set within RISC framework. *Microprocessor Report*, 5(6), April 1991.

[Caudill and Wirfs-Brock, 1986] Patrick J. Caudill and Allen Wirfs-Brock. A third-generation Smalltalk-80 implementation. In OOPSLA 1986 [OOPSLA 19861986], pages 119–130.

[CC 2005, 2005] *Proceedings of the 14th International Conference on Compiler Construction*, Edinburgh, April 2005. Springer-Verlag.

[CenterLine, 1992] CenterLine Software, Cambridge, MA. *CodeCenter, The Programming Environment*, 1992.

[CGO 2003, 2003] *1st IEEE / ACM International Symposium on Code Generation and Optimization (CGO 2003), 23-26 March 2003, San Francisco, CA, USA*, San Francisco, CA, March 2003. IEEE Computer Society.

[Chaiken *et al.*, 1990] David Chaiken, Craig Fields, Kiyoshi Kurihara, and Anant Agarwal. Directory-based cache coherence in large-scale multiprocessors. *IEEE Computer*, 23(6):49–58, June 1990.

[Chailloux *et al.*, 1984] Jérome Chailloux, Matthieu Devin, and Jean-Marie Hullot. Le_Lisp: A portable and efficient Lisp system. In Steele [Steele1984], pages 113–122.

[Chailloux, 1991] Emmanuel Chailloux. *Compilation des Langages Fonctionnels: CeML un Traducteur ML vers C*. PhD thesis, Université de Paris VII, November 1991.

[Chailloux, 1992a] Emmanuel Chailloux. A conservative garbage collector with ambiguous roots, for static type checking languages. In Bekkers and Cohen [Bekkers and Cohen1992].

[Chailloux, 1992b] Emmanuel Chailloux. An efficient way of compiling ML to C. In David MacQueen, editor, *ACM SIGPLAN Workshop on ML and its Applications, San Francisco*. ACM Press, June 1992.

[Chambers and Hosking, 2000] Craig Chambers and Antony L. Hosking, editors. *Proceedings of the Second International Symposium on Memory Management*, ACM SIGPLAN Notices 36(1), Minneapolis, MN, October 2000.

[Chambers and Ungar, 1989] Craig Chambers and David M. Ungar. Customization: Optimizing compiler technology for Self, a dynamically-typed object-oriented language. In *Proceedings of SIGPLAN '89*, pages 146–160. ACM Press, 1989.

[Chambers *et al.*, 1989] Craig Chambers, David M. Ungar, and Elgin Lee. An efficient implementation of SELF, a dynamically-typed object-oriented language based on prototypes. In OOPSLA 1989 [OOPSLA 19891989], pages 48–70.

[Chambers *et al.*, 1991] Craig Chambers, David M. Ungar, and Frank Jackson. An efficient implementation of SELF, a dynamically-typed object-oriented language based on prototypes. *Lisp and Symbolic Computation*, 4:243–281, 1991.

[Chambers, 1992] Craig Chambers. *The Design and Implementation of the SELF Compiler, an Optimizing Compiler for an Objected-Oriented Programming Language*. PhD thesis, Stanford University, March 1992.

[Chambers, 1993] Craig Chambers. Cost of garbage collection in the SELF system. In Moss et al. [Moss *et al.*1993].

[Chang and Daugherty, 2000] J. Morris Chang and C.H. Daugherty. An efficient data structure for dynamic memory management. *Journal of Systems and Software*, 54(3):219–226, November 2000.

[Chang and Gehringer, 1993a] J. Morris Chang and Edward F. Gehringer. Evaluation of an object-caching coprocessor design for object-oriented systems. In *Proceedings of IEEE International Conference on Computer Design*. IEEE Press, October 1993.

[Chang and Gehringer, 1993b] J. Morris Chang and Edward F. Gehringer. Performance of object caching for object-oriented systems. In *Proceedings of International Conference on Very Large Scale Integration, VLSI'93, Grenoble, France*, September 1993.

[Chang and Gehringer, 1996] J. Morris Chang and Edward F. Gehringer. A high-performance memory allocator for object-oriented systems. *IEEE Transactions on Computers*, pages 357–366, March 1996.

[Chang and Katz, 1989] Ellis E. Chang and Randy H. Katz. Exploiting inheritance and structure semantics for effective clustering and buffering in an object-oriented DBMS. In SIGMOD 1989 [SIGMOD 19891989], pages 348–357.

[Chang and Kuo, 2002] Li-Pin Chang and Tei-Wei Kuo. A real-time garbage collection mechanism for flash-memory storage systems in embedded systems. In RTCSA 2002 [RTCSA 20022002].

[Chang and Lee, 1998] J. Morris Chang and Woo Hyong Lee. A study on memory allocations in C++. In *Proceedings of 14th International Conference on Advanced Science and Technology (ICAST'98)*, pages 53–62, Naperville, IL, April 1998.

[Chang and Maxemchuk, 1984] J. Chang and N. F. Maxemchuk. Reliable broadcast protocols. *ACM Transactions on Computer Systems*, 2:251–273, August 1984.

[Chang and Wellings, 2005] Yang Chang and Andy Wellings. Integrating hybrid garbage collection with dual priority scheduling. In RTCSA 2005 [RTCSA 20052005], pages 185–188.

[Chang and Wellings, 2006a] Yang Chang and Andy Wellings. Low memory overhead real-time garbage collection for Java. In *Proceedings of the 4th International Workshop on Java Technologies for Real-time and Embedded Systems*. University of York, October 2006.

[Chang and Wellings, 2006b] Yang Chang and Andy J. Wellings. Hard real-time hybrid garbage collection with low memory requirements. In *Proceedings of the 27th IEEE Real-Time Systems Symposium*, pages 77–86, December 2006.

[Chang *et al.*, 1999a] J. Morris Chang, Woo Hyong Lee, and Yusuf Hasan. Measuring dynamic memory invocations in object-oriented programs. In *Proceedings of 18th IEEE International Performance Conference on Computers and Communications*, pages 268–274, Phoenix, AZ, February 1999.

[Chang *et al.*, 1999b] J. Morris Chang, Witiwas Srisa-an, and Chia-Tien Dan Lo. DMMX (dynamic memory management extensions): An introduction. In *Workshop notes of ICCD workshop on Hardware Support for Objects and Microarchitectures for Java*, pages 11–14, Austin, TX, October 1999.

[Chang *et al.*, 1999c] J. Morris Chang, Witiwas Srisa-an, and Chia-Tien Dan Lo. OMeX: Object management extensions for embedded systems. In *The Second International Workshop on Compiler and Architecture Support for Embedded Systems (CASES'99)*, Washington, DC, October 1999.

[Chang *et al.*, 2000a] J. Morris Chang, Yusuf Hasan, and Woo Hyong Lee. High-performance memory allocator for memory intensive applications. In *Proceedings of Fourth IEEE International Conference on High Performance Computing in Asia-Pacific Region*, pages 6–12, Beijing, China, May 2000.

[Chang *et al.*, 2000b] J. Morris Chang, Chia-Tien Dan Lo, and Edward F. Gehringer. Hardware support for dynamic memory management. In *Workshop notes ofInternational Symposium on Computer Architecture(ISCA) workshop on Solving the Memory Wall Problem*, Vancouver, June 2000.

[Chang *et al.*, 2000c] J. Morris Chang, Witiwas Srisa-an, and C.D. Lo. Hardware support for concurrent garbage collection in SMP systems. In *Proceedings of Fourth IEEE International Conference on High Performance Computing in Asia-Pacific Region*, pages 513–517, Beijing, China, May 2000.

[Chang *et al.*, 2000d] J. Morris Chang, Witiwas Srisa-an, and Chia-Tien Dan Lo. Architectural support for dynamic memory management. In *Proceedings of IEEE International Conference on Computer Design*, pages 99–104, Austin, TX, September 2000.

[Chang *et al.*, 2000e] J. Morris Chang, Witiwas Srisa-an, and Chia-Tien Dan Lo. Architectural support for dynamic memory management. In *Proceedings of IEEE International Conference on Computer Design*, pages 99–104, Austin, TX, September 2000.

[Chang *et al.*, 2001] J. Morris Chang, W.H. Lee, and Witiwas Srisa-an. A study of the allocation behavior of C++ programs. *Journal of Systems and Software*, 2001. accepted for publication, Fall 2001.

[Chang *et al.*, 2002] J. Morris Chang, Witawas Srisa-an, Chia-Tien Dan Lo, and Edward F. Gehringer. DMMX: Dynamic memory management extensions. *Journal of Systems and Software*, 63(3), September 2002.

[Chang *et al.*, 2004] Li-Pin Chang, Tei-Wei Kuo, and Shi-Wu Lo. Real-time garbage collection for flash-memory storage systems of real-time embedded systems. *ACM Transactions on Embedded Computer Systems*, 3(4):837–863, November 2004.

[Chang, 1997] J. Morris Chang. An optimized two-dimensional buddy system for dynamic resource allocation. *Journal of High Performance Computing*, 4(1):47–55, December 1997.

[Chang, 2005] Yang Chang. Integrating hybrid garbage collection with dual priority scheduling. Technical Report YCS388(2005), University of York, 2005.

[Chang, 2006] Yang Chang. Hard real-time hybrid garbage collection with low memory requirement. Technical report, University of York, 2006.

[Chang, 2007] Yang Chang. *Garbage Collection for Flexible Hard Real-time Systems*. PhD thesis, University of York, 2007.

[Chansler, 1986] R. J. Chansler. *Efficient Use of Systems with Many Processors*. Number 6 in Computer science: computer architecture and design. University of Michigan Press, Ann Arbor, MI, 1986.

[Chase *et al.*, 1990] David R. Chase, Wegman, and Zadeck. Analysis of pointers and structures. *ACM SIGPLAN Notices*, 25(6), 1990.

[Chase *et al.*, 1992a] Jeffrey S. Chase, Henry M. Levy, Miche Baker-Harvey, and Edward D. Lazowska. How to use a 64-bit virtual address space. Technical Report 92-03-02, University of Washington, Seattle, Washington, February 1992.

[Chase *et al.*, 1992b] Jeffrey S. Chase, Henry M. Levy, Edward D. Lazowska, and Miche Baker-Harvey. Lightweight shared objects in a 64-bit operating system. In OOPSLA 1992 [OOPSLA 19921992].

[Chase, 1987] David R. Chase. *Garbage Collection and Other Optimizations*. PhD thesis, Rice University, August 1987.

[Chase, 1988] David R. Chase. Safety considerations for storage allocation optimizations. *ACM SIGPLAN Notices*, 23(7):1–10, 1988.

[Chawla, 2003] Sumit Chawla. Fine-tuning Java garbage collection performance: How to detect and troubleshoot garbage collection issues with the IBM Java virtual machine. *IBM developerWorks*, January 2003.

[Cheadle *et al.*, 2000] Andrew M. Cheadle, Anthony J. Field, Simon Marlow, Simon L. Peyton Jones, and R.L While. Non-stop Haskell. In ICFP 2000 [ICFP 20002000], pages 257–267.

[Cheadle *et al.*, 2004] Andrew M. Cheadle, Anthony J. Field, Simon Marlow, Simon L. Peyton Jones, and Lyndon While. Exploring the barrier to entry — incremental generational garbage collection for Haskell. In Bacon and Diwan [Bacon and Diwan2004], pages 163–174.

[Cheadle *et al.*, 2006] Andrew Cheadle, Tony Field, John Ayres, Neil Dunn, Richard Hayden, and Johan Nystrom-Persson. Visualising dynamic memory allocators. In Petrank and Moss [Petrank and Moss2006], pages 115–125.

[Cheadle *et al.*, 2008] A. M. Cheadle, A. J. Field, and J. Nyström-Persson. A method specialisation and virtualised execution environment for Java. In Gregg et al. [Gregg *et al.*2008], pages 51–60.

[Chen *et al.*, 2002a] Guangyu Chen, R. Shetty, Mahmut T. Kandemir, Narayanan Vijaykrishnan, Mary Jane Irwin, and Mario Wolczko. Influence of garbage collection on memory system energy. *ACM Transactions on Embedded Computer Systems*, 1(1), November 2002.

[Chen *et al.*, 2002b] Gungyu Chen, Mahmut T. Kandemir, Narayanan Vijaykrishnan, Mary Jane Irwin, and Mario Wolczko. Adaptive garbage collection for battery-operated environments. In JVM 2002 [JVM 20022002], pages 1–12.

[Chen *et al.*, 2002c] Gungyu Chen, R. Shetty, Mahmut T. Kandemir, Narayanan Vijaykrishnan, Mary Jane Irwin, and Mario Wolczko. Tuning garbage collection for reducing memory system energy in an embedded Java environment. *ACM Transactions on Embedded Computer Systems*, 1(1):6–26, November 2002.

[Chen *et al.*, 2002d] Gungyu Chen, R. Shetty, Mahmut T. Kandemir, Narayanan Vijaykrishnan, Mary Jane Irwin, and Mario Wolczko. Tuning garbage collection in an embedded Java environment. In *Proceedings of the Eighth International Symposium on High-Performance Computer Architecture (HPCA'02)*, pages 92–, Boston, MA, February 2002. IEEE Computer Society.

[Chen *et al.*, 2003a] D. Chen, A. Messer, D. Milojicic, and S. Dwarkadas. Garbage collector assisted memory offloading for memory constrained devices. In *Fifth IEEE Workshop on Mobile Computing Systems and Applications*. IEEE Press, 2003.

[Chen *et al.*, 2003b] Guangyu Chen, Mahmut Kandemir, Naraya Vijaykrishnan, Mary Jane Irwin, Bernd Mathiske, and Mario Wolczko. Heap compression for memory-constrained Java environment. In OOPSLA 2003 [OOPSLA 20032003].

[Chen *et al.*, 2004] Guangyu Chen, Mahmut Kandemir, Narayanan Vijaykrishnan, and Mary Jane Irwin. Field level analysis for heap space optimization in embedded Java environments. In Bacon and Diwan [Bacon and Diwan2004], pages 131–142.

[Chen *et al.*, 2005] Guangyu Chen, Mahmut Kandemir, and Mary J. Irwin. Exploiting frequent field values in Java objects for reducing heap memory requirements. In Hind and Vitek [Hind and Vitek2005], pages 68–78.

[Chen *et al.*, 2006] Wen-Ke Chen, Sanjay Bhansali, Trishul M. Chilimbi, Xiaofeng Gao, and Weihaw Chuang. Profile-guided proactive garbage collection for locality optimization. In Schwartzbach and Ball [Schwartzbach and Ball2006], pages 332–340.

[Cheney, 1970] C. J. Cheney. A non-recursive list compacting algorithm. *Communications of the ACM*, 13(11):677–8, November 1970.

[Cheng and Blelloch, 2001] Perry Cheng and Guy Blelloch. A parallel, real-time garbage collector. In PLDI 2001 [PLDI 20012001], pages 125–136.

[Cheng *et al.*, 1998] Perry Cheng, Robert Harper, and Peter Lee. Generational stack collection and profile-driven pretenuring. In PLDI 1998 [PLDI 19981998], pages 162–173.

[Cheng, 2001] Perry Cheng. *Scalable Real-Time Parallel Garbage Collection for Symmetric Multiprocessors*. PhD thesis, Carnegie Mellon University, 2001.

[Cheong, 1992] Fah-Chun Cheong. Almost tag-free garbage collection for strongly-typed object-oriented languages. Technical Report CSE-TR-126-92, University of Michigan, 1992.

[Cher *et al.*, 2004] Chen-Yong Cher, Antony L. Hosking, and T.N. Vijaykumar. Software prefetching for mark-sweep garbage collection: Hardware analysis and software redesign. In Mukherjee and McKinley [Mukherjee and McKinley2004], pages 199–210.

[Cherem and Rugina, 2004] Sigmund Cherem and Radu Rugina. Region analysis and transformation for Java programs. In Bacon and Diwan [Bacon and Diwan2004], pages 85–96.

[Cherem and Rugina, 2006] Sigmund Cherem and Radu Rugina. Compile-time deallocation of individual objects. In Petrank and Moss [Petrank and Moss2006], pages 138–149.

[Cherem and Rugina, 2007] Sigmund Cherem and Radu Rugina. Uniqueness inference for compile-time object deallocation. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 117–128.

[Cherem *et al.*, 2007] Sigmund Cherem, Lonnie Princehouse, and Radu Rugina. Practical memory leak detection using guarded value-flow analysis. In Ferrante and McKinley [Ferrante and McKinley2007], pages 480–491.

[Chevalier *et al.*, 2002] K. Chevalier, J. Kodumal, and X. Jiang. Memory subsystem optimization for functional languages: A case study. Technical report, Computer Science Division, University of California, Berkeley, May 2002.

[Chicha and Watt, 2006] Yannis Chicha and Stephn M. Watt. A localized tracing scheme applied to garbage collection. In Naoki Kobayashi, editor, *Proceedings of the 4th Asian Symposium on Programming Languages and Systems (APLAS)*, number 4279 in Lecture Notes in Computer Science, pages 323–349, Sydney, Australia, November 2006.

[Chicha, 2002] Yannis Chicha. *Practical Aspects of Interacting Garbage Collectors*. PhD thesis, University of Western Ontario, 2002.

[Chihaia and Gross, 2004] I. Chihaia and T. Gross. An analytical model for software-only main memory compression. In *Proceedings of the 3rd Worksop on Memory Issues (WMP'04)*, pages 77–113, New York, NY, 2004. ACM Press.

[Chikayama and Kimura, 1987] T. Chikayama and Y. Kimura. Multiple reference management in Flat GHC. In *4th International Conference on Logic Programming*, pages 276–293, 1987.

[Chilimbi and Larus, 1998] Trishul M. Chilimbi and James R. Larus. Using generational garbage collection to implement cache-conscious data placement. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 37–48.

[Chilimbi *et al.*, 1999a] Trishul M. Chilimbi, Bob Davidson, and James R. Larus. Cache-conscious structure definition. In PLDI 1999 [PLDI 19991999], pages 13–24.

[Chilimbi *et al.*, 1999b] Trishul M. Chilimbi, Mark D. Hill, and James R. Larus. Cache-conscious structure layout. In PLDI 1999 [PLDI 19991999], pages 1–12.

[Chilimbi *et al.*, 2000] Trishul Chilimbi, Richard E. Jones, and Benjamin Zorn. Designing a trace format for heap allocation events. In Chambers and Hosking [Chambers and Hosking2000], pages 35–49.

[Chilimbi, 2001] Trishul M. Chilimbi. Efficient representations and abstractions for quantifying and exploiting data reference locality. In PLDI 2001 [PLDI 20012001], pages 191–202.

[Chin *et al.*, 2003] Wei-Ngan Chin, Florin Craciun, Shengchao Qin, and Martin Rinard. Region inference for an object-oriented language. Technical report, National University of Singapore, November 2003.

[Chin *et al.*, 2004] Wei-Ngan Chin, Florin Craciun, Shengchao Qin, and Martin C. Rinard. Region inference for an object-oriented language. In Pugh and Chambers [Pugh and Chambers2004], pages 243–254.

[Chin *et al.*, 2008] Wei-Ngan Chin, Huu Hai Nguyen, Corneliu Popeea, and Shengchao Qin. Analysing memory resource bounds for bytecode programs. In Jones and Blackburn [Jones and Blackburn2008], pages 151–160.

[Chinta, 1992] Ramakrishna Chinta. Hardware-assisted garbage collection for the Icon programming language. Technical Report 92–39, Iowa State University, December 1992.

[Chirimar *et al.*, 1992] Jawahar Chirimar, Carl A. Gunter, and Jon G. Riecke. Proving memory managemeont invariants for a language based on linear logic. In LFP 1992 [LFP 19921992], pages 129–150.

[Chirimar *et al.*, 1996] Jawahar Chirimar, Carl A. Gunter, and Jon G. Riecke. Reference counting as a computational interpretation of linear logic. *Journal of Functional Programming*, 6(2):195–244, March 1996.

[Chiueh, 1991] Tzi-cker Chiueh. An architectural technique for cache-level garbage collection. In Hughes [Hughes1991a], pages 520–537.

[Cho *et al.*, 2007] Hyeonjoong Cho, Chewoo Na, Binoy Ravindran, and E. Douglas Jensen. On scheduling garbage collector in dynamic real-time systems with statistical timing assurances. *Real-Time Systems*, 36(1–2):23–46, 2007.

[Cho *et al.*, 2009] Hyeonjoong Cho, Binoy Ravindran, and Chewoo Na. Garbage collector scheduling in dynamic, multiprocessor real-time systems. *IEEE Transactions on Parallel and Distributed Systems*, 20(6):845–856, June 2009.

[Choi *et al.*, 1999] Jong-Deok Choi, Manish Gupta, Mauricio Serrano, Vugranam C. Sreedhar, and Samuel Midkiff. Escape analysis for Java. In OOPSLA 1999 [OOPSLA 19991999], pages 1–19.

[Choi *et al.*, 2003] Jong-Deok Choi, Manish Gupta, Mauricio Serrano, Vugranam C. Sreedhar, and Samuel Midkiff. Stack allocation and synchronization options for Java using escape analysis. *ACM Transactions on Programming Languages and Systems*, 25(6):876–910, November 2003.

[Choi *et al.*, 2005] Hyung-Kyu Choi, Yoo C. Chung, and Soo-Mook Moon. Java memory allocation with lazy worst fit for small objects. *Computer Journal*, 48:437–442, 2005.

[Christiansen and Velschrow, 1998] M. V. Christiansen and P. Velschrow. Region-based memory management in Java. Master's thesis, Department of Computer Science (DIKU), University of Copenhagen, May 1998.

[Christie *et al.*, 2010] Dave Christie, Jae-Woong Chung, Stephan Diestelhorst, Michael Hohmuth, Martin Pohlack, Christof Fetzer, Martin Nowack, Torvald Riegel, Pascal Felber, Patrick Marlier, and Etienne Riviere. Evaluation of AMD's advanced synchronization facility within a complete transactional memory stack. In *EuroSys*, pages 27–40, 2010.

[Christopher, 1984] T. W. Christopher. Reference count garbage collection. *Software Practice and Experience*, 14(6):503–507, June 1984.

[Chung and Moon, 2000] Yoo C. Chung and Soo-Mook Moon. Memory allocation with lazy fits. In Chambers and Hosking [Chambers and Hosking2000], pages 65–70.

[Chung *et al.*, 2000] Yoo C. Chung, Soo-Mook Moon, Kemal Ebcioglu, and Dan Sahlin. Reducing sweep time for a nearly empty heap. In *27th Annual ACM SIGPLAN-SIGACTACM Symposium on Principles of Programming Languages(POPL '00)*, Boston, MA, January 2000. ACM Press.

[Chung *et al.*, 2005] Yoo C. Chung, Soo-Mook Moon, Kemal Ebciolu, and Dan Sahlin. Selective sweeping. *Software Practice and Experience*, 35(1):15–26, 2005.

[Chung *et al.*, 2010] Jaewoong Chung, Luke Yen, Stephan Diestelhorst, Martin Pohlack, Michael Hohmuth, Dan Grossman, and David Christie. ASF: AMD64 extension for lock-free data structures and transactional memory. In *Proceedings of the 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, December 2010.

[Ciepielewski and Haridi, 1983] A. Ciepielewski and Seif Haridi. Storage models for Or-parallel execution of logic programs. Technical Report TRITA–CS–8301, Royal Institute of Technology, Stockholm, Sweden, 1983.

[Cierniak *et al.*, 2003] Michal Cierniak, Marsha Eng, Neal Glew, Brian Lewis, and James Stichnoth. The Open Runtime Platform: A flexible high-performance managed runtime environment. *Intel Technology Journal*, 7(1):5–18, 2003.

[Clark and Green, 1977] Douglas W. Clark and C. Cordell Green. An empirical study of list structure in Lisp. *Communications of the ACM*, 20(2):78–86, February 1977.

[Clark and Green, 1978] Douglas W. Clark and C. Cordell Green. A note on shared list structure in Lisp. *Information Processing Letters*, 7(6):312–314, October 1978.

[Clark, 1975] Douglas W. Clark. A fast algorithm for copying binary trees. *Information Processing Letters*, 9(3):62–63, December 1975.

[Clark, 1976] Douglas W. Clark. An efficient list moving algorithm using constant workspace. *Communications of the ACM*, 19(6):352–354, June 1976.

[Clark, 1978] Douglas W. Clark. A fast algorithm for copying list structures. *Communications of the ACM*, 21(5):351–357, May 1978.

[Clark, 1979] Douglas W. Clark. Measurements of dynamic list structure in Lisp. *ACM Transactions on Software Engineering*, 5(1):51–59, January 1979.

[Clarke and Mason, 1996] Charles L. A. Clarke and Dave Mason. Compacting garbage collection can be fast and simple. *Software Practice and Experience*, 26(2):177–194, February 1996.

[Clarke, 2001] David Clarke. On deleting aggregate objects. In SPACE 2001 [SPACE 20012001].

[Click *et al.*, 2005] Cliff Click, Gil Tene, and Michael Wolf. The Pauseless GC algorithm. In Hind and Vitek [Hind and Vitek2005], pages 46–56.

[Click, 2003] Cliff Click. Performance myths exposed. Talk at JavaOne 2003, 2003.

[Clinger and Hansen, 1997] William D. Clinger and Lars T. Hansen. Generational garbage collection and the radioactive decay model. In PLDI 1997 [PLDI 19971997], pages 97–108.

[Clinger and Klock, 2009] William D. Clinger and Felix S. Klock. Scalable garbage collection with guaranteed MMU. In *Scheme and Functional Programming 2009*, Boston, MA, August 2009.

[Clinger and Rojas, 2006] William D. Clinger and Fabio V. Rojas. Linear combinations of radioactive decay models for generational garbage collection. In *Science of Computer Programming* [Jones2006], pages 184–203.

[Clinger *et al.*, 1988] William D. Clinger, Anne Hartheimer, and Erik Ost. Implementation strategies for continuations. In LFP 1988 [LFP 19881988], pages 124–131.

[Cockshott *et al.*, 1984] W. Cockshott, Malcolm Atkinson, K. Chisholm, P. Bailey, and Ron Morrison. Persistent object management system. *Software Practice and Experience*, 14(1):49–71, January 1984.

[Codewright's Toolworks, 1993] Codewright's Toolworks, San Pedro, CA. *Alloc-GC: The Garbage Collecting Replacement for malloc()*, 1993.

[Coffman, Jr. and Leighton, 1989] E. G. Coffman, Jr. and F. T. Leighton. A provably efficient algorithm for dynamic storage allocation. *Journal of Computer and System Sciences*, 38(1):2–35, February 1989.

[Coffman, Jr. *et al.*, 1985] E. G. Coffman, Jr., T. T. Kadota, and L. A. Shepp. On the asymptotic optimality of first-fit storage allocation. *IEEE Transactions on Software Engineering*, SE-11(2):235–239, February 1985.

[Coffman, Jr., 1983] E. G. Coffman, Jr. An introduction to combinatorial models of dynamic storage allocation. *SIAM Review*, 25(3):311–325, July 1983.

[Cohen and Nicolau, 1983] Jacques Cohen and Alexandru Nicolau. Comparison of compacting algorithms for garbage collection. *ACM Transactions on Programming Languages and Systems*, 5(4):532–553, 1983.

[Cohen and Trilling, 1967] Jacques Cohen and Laurent Trilling. Remarks on garbage collection using a two level storage. *BIT*, 7(1):22–30, 1967.

[Cohen and Zuckerman, 1972] Jacques Cohen and C. Zuckerman. Evalquote in simple Fortran: A tutorial on interpreting Lisp. *BIT*, 12(3):299–317, 1972.

[Cohen *et al.*, 2006] M. Cohen, S. Kooi, and Witiwas Srisa-an. Clustering the heap in multi-threaded applications for improved garbage collection. In *Proceedings of the ACM Genetic and Evolutionary Computation Conference (GECCO)*, pages 1901–1908, Seattle, WA, July 2006.

[Cohen, 1967] Jacques Cohen. A use of fast and slow memories in list-processing languages. *Communications of the ACM*, 10(2):82–86, February 1967.

[Cohen, 1981] Jacques Cohen. Garbage collection of linked data structures. *Computing Surveys*, 13(3):341–367, September 1981.

[Cohn and Singh, 1997] David Cohn and Satinder Singh. Predicting lifetimes in dynamically allocated memory. In M. Mozer et al., editors, *Advances in Neural Information Processing Systems 9*, 1997.

[Collins, 1960] George E. Collins. A method for overlapping and erasure of lists. *Communications of the ACM*, 3(12):655–657, December 1960.

[Collins, 1961] George E. Collins. Experience in automatic storage allocation. *Communications of the ACM*, 4(10):436–440, October 1961.

[Collins, 1965] George E. Collins. REFCO III, a reference count list processing system for the IBM 7094. Research Report RC-1436, IBM Corp., May 1965.

[Collinson and Pym, 2006] Matthew Collinson and David Pym. Bunching for region and location models. In SPACE 2006 [SPACE 20062006], pages 14–22.

[Colnet *et al.*, 1998a] Dominique Colnet, Philippe Coucaud, and Olivier Zendra. Compiler support to customize the mark and sweep algorithm. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 154–165.

[Colnet *et al.*, 1998b] Dominique Colnet, Olivier Zendra, and Philippe Coucaud. Using type inference to customize the garbage collector in an object-oriented language. the SmallEiffel compiler. Rapport de recherche, INRIA Lorraine, 1998.

[Colvin, 1995] G. Colvin. Smart pointer for C++ garbage collection. *C/C++ Users Journal*, 12(12), December 1995.

[Comfort, 1964] W. T. Comfort. Multiword list items. *Communications of the ACM*, 7(6), June 1964.

[Connor and Nettles, 1997] Richard C. H. Connor and Scott Nettles, editors. *Proceedings of the SeventhInternational Workshop on Persistent Object Systems (May, 1996)*, Persistent Object Systems: Principles and Practice, Cape May, NJ, USA, 1997. Morgan Kaufmann.

[Conrad, 1974] W. R. Conrad. A compactifying garbage collector for ECL's non-homogenous heap. Research Report 2–74, Center for Research in Computing Technology, Harvard, February 1974.

[Cook *et al.*, 1993a] Jonathan E. Cook, Alexander L. Wolf, and Benjamin G. Zorn. The design of a simulation system for persistent object storage management. Computer Science Technical Report CU-CS-647-93, University of Colorado, Campus Box 430, Boulder, CO 80309, March 1993.

[Cook *et al.*, 1993b] Jonathan E. Cook, Alexander L. Wolf, and Benjamin G. Zorn. The performance of partitioned garbage collection in object databases. Computer Science Technical Report CU-CS-653-93, University of Colorado, Campus Box 430, Boulder, CO 80309, June 1993.

[Cook *et al.*, 1994] Jonathan E. Cook, Alexander L. Wolf, and Benjamin G. Zorn. Partition selection policies in object databases garbage collection. In Richard T. Snodgrass and Marianne Winslett, editors, *Proceedings of ACM SIGMOD International Conference on the Management of Data*, volume 23(2), pages 317–382, Minneapolis, May 1994. ACM Press.

[Cook *et al.*, 1996] Jonathan E. Cook, Artur Klauser, Alexander L. Wolf, and Benjamin G. Zorn. Semi-automatic, self-adaptive control of garbage collection rates in object databases. In H. V. Jagadish and Inderpal Singh Mumick, editors, *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 377–388, Montreal, Quebec, Canada, June 1996. *SIGMOD Record* 25(2), June 1996.

[Cook *et al.*, 1998] Jonathan E. Cook, Alexander L. Wolf, and Benjamin G. Zorn. A highly effective partition selection policy for object database garbage collection. *IEEE Transactions on Knowledge and Data Engineering*, 10(1):153–172, 1998.

[Cooper and Harvey, 1998] Keith D. Cooper and Timothy J. Harvey. Compiler-controlled memory. In ASPLOS 1998 [ASPLOS 19981998], pages 2–11.

[Cooper *et al.*, 1992] Eric Cooper, Scott Nettles, and Indira Subramanian. Improving the performance of SML garbage collection using application-specific virtual memory management. In LFP 1992 [LFP 19921992], pages 43–52.

[Coplien, 1992] James Coplien. *Advanced C++ Programming Styles and Idioms*. Addison-Wesley, 1992.

[Corporaal *et al.*, 1988] H. Corporaal, T. Veldman, and A. J. van de Goor. Reference weight-based garbage collection for distributed systems. In *Proceedings of the SION Conference on Computing Science in the Netherlands*, Utrecht, November 1988.

[Corporaal *et al.*, 1990] H. Corporaal, T. Veldman, and A. J. van de Goor. Efficient, reference weight-based garbage collection method for distributed systems. In *PARBASE-90: International Conference on Databases, Parallel Architectures, and Their Applications*, pages 463–465, Miami Beach, 7–9 March 1990. IEEE Press.

[Corporaal, 1989] H. Corporaal. Garbage collection in distributed systems. Internal report, Technical University, Delft, 1989.

[Corry, 2006] Erik Corry. Optimistic stack allocation for Java-like languages. In Petrank and Moss [Petrank and Moss2006], pages 162–173.

[Cortemanche, 1986] Anthony J. Cortemanche. MultiTrash, a parallel garbage collector for MultiScheme. Bachelor's thesis, MIT Press, January 1986.

[Courts, 1988] Robert Courts. Improving locality of reference in a garbage-collecting memory management-system. *Communications of the ACM*, 31(9):1128–1138, 1988.

[Couvert *et al.*, 1989] André Couvert, Aomar Maddi, and René Pedrono. Partage d'objets dans les systèmes distribués. principes des ramasse-miettes. Rapport de recherche 963, INRIA, January 1989.

[Craciunas *et al.*, 2008] Silviu S. Craciunas, Christoph M. Kirsch, Hannes Payer, Ana Sokolova, Horst Stadler, , and Robert Staudinger. A compacting real-time memory management system. In *Proceedings of USENIX'08*, pages 349–362, 2008.

[Crammond and Lindholm, 1995] Jim Crammond and Tim Lindholm. Memory management in Quintus Prolog. In Kluwer, editor, *Implementations of Logic Programming Systems, Budapest*, 1995.

[Crammond, 1988] Jim Crammond. A garbage collection algorithm for shared memory parallel processors. *International Journal Of Parallel Programming*, 17(6):497–522, 1988.

[Cranston and Thomas, 1975] B. Cranston and R. Thomas. A simplified recombination scheme for the Fibonacci buddy system. *Communications of the ACM*, 18(6):331–332, July 1975.

[Crary *et al.*, 1998] Karl Crary, David Walker, and Greg Morrisett. Typed memory management in a calculus of capabilities. Technical report, Cornell University, July 1998.

[Crary *et al.*, 1999] Karl Crary, David Walker, and Greg Morrisett. Typed memory management in a calculus of capabilities. In POPL 1999 [POPL 19991999].

[Crawley, 1991] S. C. Crawley. Local and global distributed garbage collection. In Wilson and Hayes [Wilson and Hayes1991a].

[Cridlig, 1992] Regis Cridlig. An optimising ML to C compiler. In David MacQueen, editor, *ACM SIGPLAN Workshop on ML and its Applications*, San Francisco, June 1992. ACM Press.

[Critchlow, 1992] Terence J. Critchlow. A distributed garbage collection algorithm. Master's thesis, Department of Computer Science, University of Utah, August 1992. University of Utah Technical report CSTD–92–011.

[Curial *et al.*, 2008] Stephen Curial, Peng Zhao, Jose Nelson Amaral, Yaoqing Gao, Shimin Cui, Raul Silvera, and Roch Archambault. Memory pooling assisted data splitting (MPADS). In Jones and Blackburn [Jones and Blackburn2008], pages 101–110.

[Daconta, 1993] Michael C. Daconta. *C Pointers and Dynamic Memory Management*. QED Publishing, 1993.

[Daconta, 1995] Michael C. Daconta. *C++ Pointers and Dynamic Memory Management*. Wiley, 1995.

[Dahl and Nygaard, 1966] O. J. Dahl and K. Nygaard. Simula — an Algol-based simulation language. *Communications of the ACM*, 9:671–678, 1966.

[Dahl, 1963] O. J. Dahl. The SIMULA storage allocation scheme. NCC Document 62, Norsk Regnesentral, November 1963.

[Danvy and Jean, 1986] Olivier Danvy and Danielle Jean. Langage d'impleémentation pour langages applicatifs: contribution á l'étude d'une réalisation informatique. Technical Report 86–23, LITP, France, January 1986. Also Bigre+Globule 48, 3e journées d'étude sur les Langages Orientés Objet.

[Danvy, 1986a] Olivier Danvy. Agir avec LILA: le manuel de référence. Technical Report 86–40, LITP, France, May 1986.

[Danvy, 1986b] Olivier Danvy. LILA: a virtual machine for functional and declarative languages. Technical Report 86–38, LITP, France, May 1986. Workshop on Future Directions in Computer Science and Software.

[Danvy, 1987] Olivier Danvy. Memory allocation and higher-order functions. *ACM SIGPLAN Notices*, 22(7):241–252, June 1987.

[Daugherty and Chang, 1998] C. H. Daugherty and J. Morris Chang. Common list method: A simple, efficient allocator implementation. In *Proceedings of Sixth Annual High-Performance Computing Symposium*, pages 180–185, Boston, MA, April 1998.

[Davies, 1984] D. Julian M. Davies. Memory occupancy patterns in garbage collection systems. *Communications of the ACM*, 27(8):819–825, August 1984.

[Dawson, 1982] Jeffrey L. Dawson. Improved effectiveness from a real-time LISP garbage collector. In LFP 1982 [LFP 19821982], pages 159–167.

[Day and Zimmermann, 1983] J. D. Day and H. Zimmermann. The OSI Reference Model. In *Proceedings of the IEEE*, volume 71, pages 1334–1340. IEEE Press, December 1983.

[Day *et al.*, 1994] M. Day, Barbara Liskov, Umesh Maheshwari, and A. Myers. References to remote mobile objects in Thor. *ACM Letters on Programming Languages and Systems*, 2(1–4), March 1994.

[Daylight *et al.*, 2004] Edgar G. Daylight, Bart Demoen, and Francky Catthor. Formally specifying dynamic data structures for embedded software design: An initial approach. In SPACE 2004 [SPACE 20042004].

[Daynès and Atkinson, 1997] Laurent Daynès and Malcolm P. Atkinson. Main-memory management to support orthogonal persistence for Java. In *Proceedings of the Second International Workshop on Persistence and Java (PJW2)*, Half Moon Bay, CA, USA, August 1997. To be published.

[de Bakker *et al.*, 1987] Jacobus W. de Bakker, L. Nijman, and Philip C. Treleaven, editors. *PARLE'87 Parallel Architectures and Languages Europe*, volume 258/259 of *Lecture Notes in Computer Science*, Eindhoven, The Netherlands, June 1987. Springer-Verlag.

[de Boer and Klop, 1992] F. S. de Boer and J. W. Klop. Asynchronous communication in process algebra: Extended abstract. Technical Report CS-R9206, Stichting Mathematisch Centrum, January 1992.

[de Pauw and Sevitski, 1999] W. de Pauw and G. Sevitski. Visualizing reference patterns for solving memory leaks in Java. In ECOOP 1999 [ECOOP 19991999], pages 116–134.

[de Pauw and Sevitski, 2000] W. de Pauw and G. Sevitski. Visualizing reference patterns for solving memory leaks in Java. *Concurrency and Computation: Practice and Experience*, 12:1431–1454, 2000.

[de Pauw *et al.*, 1993] W. de Pauw, R. Helm, D. Kimelman, and J. Vlissides. Visualizing the behaviour of object-oriented systems. In OOPSLA 1993 [OOPSLA 19931993], pages 326–337.

[de Pauw *et al.*, 1997] W. de Pauw, D. Kimelman, and J. Vlissides. Visualizing object-oriented software execution. In John T. Stasko, John B. Domingue, Marc H. Brown, and Blaine A. Price, editors, *Software Visualization*. MIT Press, 1997.

[de Pauw *et al.*, 2001] W. de Pauw, N. Mitchell, M. Robillard, G. Sevitski, and H. Srinivasan. Drive-by analysis of running programs. In *Proceedings for Workshop on Software Visualization, International Conference on Software Engineering*, Toronto, May 2001.

[Dearle and others, 1992] Alan Dearle et al. An examination of operating system support for persistent object system s. In *25th Hawaii International Conference on Systems Sciences, vol. 1*, pages 779–789, 1992.

[Dearle *et al.*, 1991] Alan Dearle, Gail M. Shaw, and Stanley B. Zdonik, editors. *Proceedings of the Fourth International Workshop on Persistent Object Systems (September, 1990)*, Implementing Persistent Object Bases: Principles and Practice, Martha's Vineyard, MA, USA, 1991. Morgan Kaufman.

[Dearle *et al.*, 1992] Alan Dearle, Rex di Bona, James Farrow, Frans Henskens, Anders Lindstrom, John Rosenberg, and Francis Vaughan. Grasshopper — a persistent operating system for conventional hardware. In Cabrera et al. [Cabrera *et al.*1992], pages 81–85.

[Deb, 1984] Ashoke Deb. An efficient garbage collector for graph machines. Technical Report CS/E-84-003, Oregon Graduate Center, 1984.

[Deb, 1987] Ashoke Deb. Parallel garbage collection in a parallel virtual memory environment. In J. H. Fasel and R. M. Keller, editors, *Graph Reduction: Proceedings of a Workshop at Santa Fe, New Mexico*, volume 279 of *Lecture Notes in Computer Science*, pages 252–264, New York, NY, 1987. Springer-Verlag.

[Delacour, 1992] V. Delacour. Allocation regions and implementation contracts. In Bekkers and Cohen [Bekkers and Cohen1992].

[Dellar, 1980] C. N. R. Dellar. Removing backing store administration from the CAP operating system. *ACM SIGOPS Operating Systems Review*, 14(4):41–49, 1980.

[Demers *et al.*, 1990] Alan Demers, Mark Weiser, Barry Hayes, Hans Boehm, Daniel G. Bobrow, and Scott Shenker. Combining generational and conservative garbage collection: Framework and implementations. In POPL 1990 [POPL 19901990], pages 261–269.

[Demoen and Sagonas, 1998] Bart Demoen and Konstantinos Sagonas. Memory management for Prolog with tabling. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 97–106.

[Demoen *et al.*, 1996] Bart Demoen, Geert Engels, and Paul Tarau. Segment preserving copying garbage collection for WAM-based Prolog. In Jim Hightower, editor, *ACM Symposium on Applied Computing*, Philadelphia, February 1996. ACM Press. Programming languages track.

[Demoen *et al.*, 2002] Bart Demoen, Phuong-Lan Nguyen, and Ruben Vandeginste. Copying garbage collection for the WAM: To mark or not to mark? In Stuckey [Stuckey2002].

[Demoen, 2002] Bart Demoen. A different look at garbage collection for the WAM. In Stuckey [Stuckey2002].

[Denning and Scwartz, 1972] P. J. Denning and Scwartz. Properties of the working-set model. *Communications of the ACM*, 15(3):191–198, March 1972.

[Denning, 1968a] P. J. Denning. Thrashing: Its causes and prevention. In *Proceedings of AFIPS 1968 Fall Joint Computer Conference*, volume 33, pages 915–922, June 1968.

[Denning, 1968b] P. J. Denning. The working set model for program behaviour. *Communications of the ACM*, 11:323–333, 1968.

[Denning, 1970] P. J. Denning. Virtual memory. *ACM Computing Surveys*, 2(3):153–190, September 1970.

[Denning, 1980] Peter Denning. Working sets past and present. *IEEE Transactions on Software Engineering*, SE-6(1):64–84, January 1980.

[Derbyshire, 1987] Margaret H. Derbyshire. Garbage collection on the IRM: Report number 6. Departmental Research Report FS/MU/MHD/004–87, University of Manchester, Department of Computer Science, 1987.

[Derbyshire, 1990] Margaret H. Derbyshire. Mark scan garbage collection on a distributed architecture. *Lisp and Symbolic Computation*, 3(2):135 – 170, April 1990.

[Dershowitz, 1980] N. Dershowitz. The Schorr–Waite marking algorithm revisited. *Information Processing Letters*, 11(3):141–143, November 1980.

[Deters and Cytron, 2002] Morgan Deters and Ron Cytron. Automated discovery of scoped memory regions for real-time Java. In Boehm and Detlefs [Boehm and Detlefs2002], pages 25–35.

[Deters *et al.*, 2004] Morgan Deters, Nicholas A. Leidenfrost, M Matthew P. Hampton, James C. Brodman, and Ron Cytron. Automated reference-counted object recycling for real-time Java. In *Real-Time and Embedded Technology and Applications Symposium, RTAS 2004*, pages 424–433. IEEE Press, 2004.

[Detlefs and Kalsow, 1995] Dave Detlefs and Bill Kalsow. Debugging storage management problems in garbage-collected environments. In *USENIX Conference on Object-Oriented Technologies*. USENIX Association, June 1995.

[Detlefs *et al.*, 1993] David L. Detlefs, Al Dosser, and Benjamin Zorn. Memory allocation costs in large C and C++ programs. Computer Science Technical Report CU-CS-665-93, Digital Equipment Corporation and University of Colorado, 130 Lytton Avenue, Palo Alto, CA 94301 and Campus Box 430, Boulder, CO 80309, August 1993.

[Detlefs *et al.*, 1994] David Detlefs, Al Dosser, and Benjamin Zorn. Memory allocation costs in large C and C++ programs. *Software Practice and Experience*, 24(6), 1994.

[Detlefs *et al.*, 2001] David L. Detlefs, Paul A. Martin, Mark Moir, and Guy L. Steele. Lock-free reference counting. In *Proceedings of 20th ACM Symposium on Distributed Computing*, August 2001.

[Detlefs *et al.*, 2002a] David Detlefs, William D. Clinger, Matthias Jacob, and Ross Knippel. Concurrent remembered set refinement in generational garbage collection. In JVM 2002 [JVM 20022002].

[Detlefs *et al.*, 2002b] David L. Detlefs, Paul A. Martin, Mark Moir, and Guy L. Steele. Lock-free reference counting. *Distributed Computing*, 15:255–271, 2002.

[Detlefs *et al.*, 2004] David Detlefs, Christine Flood, Steven Heller, and Tony Printezis. Garbage-first garbage collection. In Bacon and Diwan [Bacon and Diwan2004], pages 37–48.

[Detlefs, 1990a] David L. Detlefs. Concurrent, atomic garbage collection. In Jul and Juul [Jul and Juul1990].

[Detlefs, 1990b] David L. Detlefs. Concurrent garbage collection for C++. Technical Report CMU–CS–90–119, Carnegie Mellon University, Pittsburgh, PA, May 1990.

[Detlefs, 1991a] David L. Detlefs. *Concurrent, Atomic Garbage Collection*. PhD thesis, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213, November 1991.

[Detlefs, 1991b] David L. Detlefs. Concurrent garbage collection for C++. In Peter Lee, editor, *Topics in Advanced Language Implementation*. MIT Press, 1991.

[Detlefs, 1992] David L. Detlefs. Garbage collection and runtime typing as a C++ library. In *USENIX C++ Conference*, Portland, Oregon, August 1992. USENIX Association.

[Detlefs, 1993] David L. Detlefs. Empirical evidence for using garbage collection in C and C++ programs. In Moss et al. [Moss *et al.*1993].

[Detlefs, 2004a] David Detlefs. Automatic inference of reference-count invariants. In SPACE 2004 [SPACE 20042004].

[Detlefs, 2004b] David Detlefs. A hard look at hard real-time garbage collection. In ISORC 2004 [ISORC 20042004], pages 23–32. Invited paper.

[DeTreville, 1990a] John DeTreville. Experience with concurrent garbage collectors for Modula-2+. Technical Report 64, DEC Systems Research Center, Palo Alto, CA, August 1990.

[DeTreville, 1990b] John DeTreville. Experience with garbage collection for Modula-2+ in the Topaz environment. In Jul and Juul [Jul and Juul1990].

[DeTreville, 1990c] John DeTreville. Heap usage in the Topaz environment. Technical Report 63, DEC Systems Research Center, Palo Alto, CA, August 1990.

[Deutsch and Bobrow, 1976] L. Peter Deutsch and Daniel G. Bobrow. An efficient incremental automatic garbage collector. *Communications of the ACM*, 19(9):522–526, September 1976.

[Deutsch and Schiffman, 1984] Peter L. Deutsch and A. M. Schiffman. Efficient implementation of the Smalltalk-80 system. In POPL 1984 [POPL 19841984], pages 297–302.

[Deutsch, 1973] L. Peter Deutsch. A LISP machine with very compact programs. In *International Joint Conference on Artificial Intelligence*, pages 697–703, Stanford, CA, 1973.

[Deutsch, 1983] L. Peter Deutsch. The Dorado Smalltalk-80 implementation: Hardware architecture's impact on software architecture. In Krasner [Krasner1983], pages 113–125.

[Deutsch, 1990] A. Deutsch. On determining lifetime and aliasing of dynamically allocated data in higher-order functional specifications. In POPL 1990 [POPL 19901990], pages 157 – 168.

[Deutsch, 1994] Alain Deutsch. Interprocedural may-alias analysis for pointers: Beyond k-limiting. In PLDI 1994 [PLDI 19941994], pages 230–241.

[Dewar and McCann, 1977] Robert B. K. Dewar and A. P. McCann. MACRO SPITBOL — a SNOBOL4 compiler. *Software Practice and Experience*, 7(1):95–113, 1977.

[Dewar *et al.*, 1982] Robert B. K. Dewar, Micha Sharir, and Elia Weixelbaum. Transformational derivation of a garbage collection algorithm. *ACM Transactions on Programming Languages and Systems*, 4(4):650–667, October 1982.

[Dhurjati *et al.*, 2003] Dinakar Dhurjati, Sumant Kowshik, Vikram Adve, and Chris Lattner. Memory safety without runtime checks or garbage collection. In LCTES 2003 [LCTES 20032003], pages 69–80.

[di Santo *et al.*, 1981] M. di Santo, L. Nigro, and W. Russo. On the efficient implementation of retention block-structured languages. *Int. J. Comput. Inf. Sci*, 10(1):39–54, February 1981.

[Díaz *et al*., 1994] Manuel Díaz, E. Pimentel, and José M. Troya. DROL: a distributed and real-time object-oriented logic environment. *Computer Journal*, 37:407–421, 1994.

[Díaz *et al*., 1997] Manuel Díaz, Bartolomé Rubio, and José M. Troya. DRL: a distributed real-time logic language. *Journal of Computing Languages, special issue on Extensions of Logic Programming*, 1997.

[Dice and Garthwaite, 2002] Dave Dice and Alex Garthwaite. Mostly lock-free malloc. In Boehm and Detlefs [Boehm and Detlefs2002], pages 163–174.

[Dickman and Wilson, 1997] Peter Dickman and Paul R. Wilson, editors. *OOPSLA Workshop on Garbage Collection and Memory Management*, October 1997.

[Dickman, 1991] Peter Dickman. Effective load balancing in a distributed object-support operating system. In Cabrera et al. [Cabrera *et al*.1991].

[Dickman, 1992a] Peter Dickman. *Distributed Object Management in a Non-Small Graph of Autonomous Networ ks With Few Failures*. PhD thesis, University of Cambridge, September 1992.

[Dickman, 1992b] Peter Dickman. Optimising weighted reference counts for scalable fault-tolerant distributed object-support systems. Unpublished note, 1992.

[Dickman, 1996a] Peter Dickman. Efficient, incremental, distributed orphan detection and actor garbage collection. In preparation, 1996.

[Dickman, 1996b] Peter Dickman. Incremental, distributed orphan detection and actor garbage collection using graph partitioning and Euler cycles. In Babaoglu and Marzullo [Babaoglu and Marzullo1996], pages 141–158.

[Dickman, 2000] Peter Dickman. Diffusion tree redirection for indirect reference counting. In Chambers and Hosking [Chambers and Hosking2000], pages 167–177.

[Dieckmann and Hölzle, 1999] Sylvia Dieckmann and Urs Hölzle. A study of the allocation behaviour of the SPECjvm98 Java benchmarks. In ECOOP 1999 [ECOOP 19991999], pages 92–115.

[Dieckmann and Hölzle, 2001] Sylvia Dieckmann and Urs Hölzle. The allocation behaviour of the SPECjvm98 Java benchmarks. In Rudolf Eigenman, editor, *Performance Evaluation and Benchmarking with Realistic Applications*, chapter 3, pages 77–108. MIT Press, 2001.

[Dijkstra and Scholten, 1980] Edsgar W. Dijkstra and C. S. Scholten. Termination detection for diffusing computations. *Information Processing Letters*, 11:1–4, August 1980.

[Dijkstra *et al.*, 1976] Edsgar W. Dijkstra, Leslie Lamport, A. J. Martin, C. S. Scholten, and E. F. M. Steffens. On-the-fly garbage collection: An exercise in cooperation. In *Language Hierarchies and Interfaces: International Summer School*, volume 46 of *Lecture Notes in Computer Science*, pages 43–56. Springer-Verlag, Marktoberdorf, Germany, 1976.

[Dijkstra *et al.*, 1978] Edsgar W. Dijkstra, Leslie Lamport, A. J. Martin, C. S. Scholten, and E. F. M. Steffens. On-the-fly garbage collection: An exercise in cooperation. *Communications of the ACM*, 21(11):965–975, November 1978.

[Dijkstra, 1975] Edsgar W. Dijkstra. Notes on a real-time garbage collection system. From a conversation with D. E. Knuth (private collection of D. E. Knuth), 1975.

[Dijkstra, 1976] Edsgar W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, Englewood Cliffs, N. J., 1976.

[Dillig *et al.*, 2008] Isil Dillig, Thomas Dillig, Eran Yahav, and Satish Chandra. The CLOSER: Automating resource management in Java. In Jones and Blackburn [Jones and Blackburn2008], pages 1–10.

[Dimpsey *et al.*, 2000] Robert Dimpsey, Rajiv Arora, and Kean Kuiper. Java server performance: A case study of building efficient, scalable JVMs. *IBM Systems Journal*, 39(1):151–174, 2000.

[Ding and Kennedy, 1999] Chen Ding and Ken Kennedy. Improving cache performance of dynamic applications through data and computation reorganization at run time. In PLDI 1999 [PLDI 19991999], pages 229–241.

[Ding and Li, 2002] Yuping Ding and Xining Li. Cache performance of chronological garbage collection. In *IEEE Canadian Conference on Electrical and Computer Engineering, 1998*, 2002.

[Ding *et al.*, 2005] Chen Ding, Chengliang Zhang, Xipeng Shen, and Mitsunori Ogihara. Gated memory control for memory monitoring, leak detection and garbage collection. In *MSP '05: Proceedings of the 2005 Workshop on Memory System Performance*, pages 62–67, Chicago, IL, 2005. ACM Press.

[Diwan *et al.*, 1992] Amer Diwan, J. Eliot B. Moss, and Richard L. Hudson. Compiler support for garbage collection in a statically typed language. In PLDI 1992 [PLDI 19921992], pages 273–282.

[Diwan *et al.*, 1993] Amer Diwan, David Tarditi, and J. Eliot B. Moss. Memory subsystem performance of programs with intensive heap allocation. Technical Report CMU-CS-93-227, Computer Science Department, Carnegie-Mellon University, December 1993. Also appears as Fox Memorandum CMU-CS-FOX-93-07.

[Diwan *et al.*, 1994] Amer Diwan, David Tarditi, and J. Eliot B. Moss. Memory subsystem performance of programs using copying garbage collection. In POPL 1994 [POPL 19941994].

[Diwan *et al.*, 1995] Amer Diwan, David Tarditi, and J. Eliot B. Moss. Memory subsystem performance of programs with intensive heap allocation. *ACM Transactions on Computer Systems*, 13(4):244–273, August 1995.

[Diwan *et al.*, 2002] Amer Diwan, Han Lee, Dirk Grunwald, and Keith Farkas. Energy consumption and garbage collection in low-powered computing. Technical Report CU–CS–930–02, University of Colorado, Boulder, 2002.

[Diwan, 1991] Amer Diwan. Stack tracing in a statically typed language. In Wilson and Hayes [Wilson and Hayes1991a].

[Doherty *et al.*, 2004] Simon Doherty, David L. Detlefs, Lindsay Groves, Christine H. Flood, Victor Luchangco, Paul A. Martin, Mark Moir, Nir Shavit, and Guy L. Steele, Jr. DCAS is not a silver bullet for nonblocking algorithm design. In *Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 216–224, Barcelona, Spain, 2004.

[Dolby and Chien, 1998] Julian Dolby and Andrew A. Chien. An evaluation of automatic object inline allocation techniques. In OOPSLA 1998 [OOPSLA 19981998].

[Dolby and Chien, 2000] Julian Dolby and Andrew A. Chien. An automatic object inlining optimization and its evaluation. In PLDI 2000 [PLDI 20002000], pages 345–357.

[Dolby, 1997] Julian Dolby. Automatic inline allocation of objects. In PLDI 1997 [PLDI 19971997], pages 7–17.

[Doligez and Gonthier, 1994] Damien Doligez and Georges Gonthier. Portable, unobtrusive garbage collection for multiprocessor systems. In POPL 1994 [POPL 19941994].

[Doligez and Leroy, 1993] Damien Doligez and Xavier Leroy. A concurrent generational garbage collector for a multi-threaded implementation of ML. In POPL 1993 [POPL 19931993], pages 113–123.

[Dolstra *et al.*, 2004] Eelco Dolstra, Eelco Visser, and Meijn de Jonge. Imposing a memory management discipline on software deployment. In ICSE 2004 [ICSE 20042004].

[Domani *et al.*, 2000a] Tamar Domani, Elliot Kolodner, and Erez Petrank. A generational on-the-fly garbage collector for Java. Technical Report 88.385, IBM Haifa Research Laboratory, 2000. Fuller version of [Domani *et al.*, 2000c].

[Domani *et al.*, 2000b] Tamar Domani, Elliot K. Kolodner, Ethan Lewis, Elliot E. Salant, Katherine Barabash, Itai Lahan, Erez Petrank, Igor Yanover, and Yossi Levanoni. Implementing an on-the-fly garbage collector for Java. In Chambers and Hosking [Chambers and Hosking2000], pages 155–166.

[Domani *et al.*, 2000c] Tamar Domani, Elliot K. Kolodner, and Erez Petrank. A generational on-the-fly garbage collector for Java. In PLDI 2000 [PLDI 20002000], pages 274–284.

[Domani *et al.*, 2002] Tamar Domani, Elliot K. Kolodner, Ethan Lewis, Erez Petrank, and Dafna Sheinwald. Thread-local heaps for Java. In Boehm and Detlefs [Boehm and Detlefs2002], pages 76–87.

[Donahue *et al.*, 2001] Steven M. Donahue, Matthew P. Hampton, Morgan Deters, Jonathan M. Nye, Ron K. Cytron, and Krishna M. Kavi. Storage allocation for real-time, embedded systems. In Henzinger and Kirsch [Henzinger and Kirsch2001], pages 131–147.

[Donnelly *et al.*, 2006] Kevin Donnelly, Joe Hallett, and Assaf Kfoury. Formal semantics of weak references. In Petrank and Moss [Petrank and Moss2006], pages 126–137.

[Dor *et al.*, 1998] Nurit Dor, Michael Rodeh, and Mooly Sagiv. Detecting memory errors via static pointer analysis. In PASTE98 [PASTE981998].

[Dorfman and Neuberger,] Len Dorfman and Marc J. Neuberger. *C++ Memory Management*. McGraw-Hill. Out of print.

[Dorochevsky and Véron, 1992] M. Dorochevsky and A. Véron. Binding techniques and garbage collection for OR-parallel CLP systems. In PLILP92 [PLILP921992], pages 39–53.

[Dorochevsky *et al.*, 1991] M. Dorochevsky, K. Schuerman, A. Véron, and J. Xu. Contraint handling, garbage collection and execution models in ElipSys. In A. Beaumont and G. Gupta, editors, *ICLP'91 Workshop on Parallel Execution of Logic Programs*, volume 569 of *Lecture Notes in Computer Science*, pages 17–28, 1991.

[Douglass, 2002] Bruce Powel Douglass. *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*, chapter 6. Addison-Wesley, 2002.

[Douglis *et al.*, 1991] Fred Douglis, M. Frans Kaashoek ad John K. Ousterhout, and Andrew S. Tanenbaum. A comparison of two distributed operating systems : Amoeba and Sprite. *Computing Systems*, 4(4):353–384, September 1991.

[Douglis, 1993] Fred Douglis. The compression cache: Using on-line compression to extend physical memory. In *1993 Winter USENIX Conference*, pages 519–529, San Diego, CA, January 1993. USENIX Association.

[Drezner and Barak, 1986] Z. Drezner and A. Barak. An asynchronous algorithm for scattering information between the active nodes of a multi-computer system. *Journal of Parallel and Distributed Computing*, 3(3):344–351, September 1986.

[Dubé *et al.*, 1996] Danny Dubé, Marc Feeley, and Manuel Serrano. Un GC temps réel semi-compactant. *Journées Francophones des Langages Applicatifs*, pages 165–181, January 1996.

[Duimovich, 1990] John Duimovich. Garbage collection in a multiprocessor Smalltalk system. Master's thesis, Carleton University, Canada, 1990.

[Durdanovic, 1991] Igor Durdanovic. A fast garbage collection algorithm for WAM-based PROLOG. In *Proceedings of 4th Workshop on Computer Science Logic, Heidelberg*, number 533 in Lecture Notes in Computer Science, pages 110–127, October 1991.

[Durieux *et al.*, 1984] Jean-Louis Durieux, Danielle Jean, coise Carré Fran and Patrick Sallé. Langage d'implémentation pour logique et acteurs. *Bigre+Globule*, November 1984. 2e journées d'étude sur les Langages Orientés Objet.

[Dwyer, 1973] B. Dwyer. Simple algorithms for traversing a tree without an auxiliary stack. *Inf Process. Lett.*, 2(5):143–145, December 1973.

[Dybvig *et al.*, 1993] R. Kent Dybvig, Carl Bruggeman, and David Eby. Guardians in a generation-based garbage collector. In PLDI 1993 [PLDI 19931993], pages 207–216.

[Dybvig *et al.*, 1994] R. Kent Dybvig, David Eby, and Carl Bruggeman. Don't stop the BIBOP: Flexible and efficient storage management for dynamically-typed languages. Technical Report 400, Indiana University Computer Science Department, March 1994.

[Dykstra *et al.*, 2002] L. Dykstra, Witiwas Srisa-an, and J. Morris Chang. An analysis of the garbage collector performance in Sun's HotSpot JVM. In *Proceedings of Twenty-First IEEE International Performance Computing and Communications Conference (IPCCC)*, pages 335–339, Phoenix, AZ, April 2002.

[Eckart and Leblanc, 1987] J. Dana Eckart and Richard J. Leblanc. Distributed garbage collection. *ACM SIGPLAN Notices*, 22(7):264–273, 1987.

[Eckart, 1987] J. Dana Eckart. *Garbage Collection for Functional Languages in a Distributed System*. PhD thesis, Georgia Institute of Technology, 1988, 1987.

[ECOOP 1984, 1984] *Proceedings of 1984 European Conference on Object-Oriented Programming, ECOOP84*, 1984.

[ECOOP 1999, 1999] *Proceedings of 13th European Conference on Object-Oriented Programming, ECOOP99*, Lisbon, July 1999.

[ECOOP 2002, 2002] *Proceedings of 16th European Conference on Object-Oriented Programming, ECOOP 2002*, Lecture Notes in Computer Science. Springer-Verlag, 2002.

[ECOOP 2004, 2004] *Proceedings of 18th European Conference on Object-Oriented Programming, ECOOP 2004*, Lecture Notes in Computer Science, Oslo, June 2004. Springer-Verlag.

[ECOOP 2007, 2007] *Proceedings of 2st European Conference on Object-Oriented Programming, ECOOP 2007*, Lecture Notes in Computer Science, Berlin, July 2007. Springer-Verlag.

[ECOOP 2008, 2008] *Proceedings of 22nd European Conference on Object-Oriented Programming, ECOOP 2008*, Lecture Notes in Computer Science, Paphos, Cyprus, July 2008. Springer-Verlag.

[Edelson and Pohl, 1990] Daniel R. Edelson and Ira Pohl. The case for garbage collection in C++. In Jul and Juul [Jul and Juul1990]. Also University of California Santa Cruz technical report UCSC-CRL-90-37.

[Edelson and Pohl, 1991] Daniel R. Edelson and Ira Pohl. A copying collector for C++. In *Usenix C++ Conference Proceedings*, pages 85–102. USENIX Association, 1991.

[Edelson, 1990] Daniel R. Edelson. Dynamic storage reclamation in C++. Master's thesis, University of California at Santa Cruz, June 1990.

[Edelson, 1992a] Daniel R. Edelson. A mark-and-sweep collector for C++. In POPL 1992 [POPL 19921992].

[Edelson, 1992b] Daniel R. Edelson. Precompiling C++ for garbage collection. In Bekkers and Cohen [Bekkers and Cohen1992].

[Edelson, 1992c] Daniel R. Edelson. Smart pointers: They're smart, but they're not pointers. In *USENIX C++ Conference*. USENIX Association, 1992.

[Edelson, 1993a] Daniel R. Edelson. Comparing two garbage collectors for C++. Technical Report UCSC-CRL-93-20, University of California, Santa Cruz, January 1993.

[Edelson, 1993b] Daniel Ross Edelson. *Type-Specific Storage Management*. PhD thesis, University of California, Santa Cruz, May 1993.

[Edwards, Date unknown] Daniel J. Edwards. Lisp II garbage collector. AI Memo 19, MIT AI Laboratory, Date unknown.

[Effinger-Dean *et al.*, 2006a] Laura Effinger-Dean, Chris Erickson, Melissa O Neill, and Darren Strash. Extending garbage collection to complex data structures. In SPACE 2006 [SPACE 20062006], pages 91–97.

[Effinger-Dean *et al.*, 2006b] Laura Effinger-Dean, Chris Erickson, Melissa O Neill, and Darren Strash. Garbage collection for trailer arrays. In SPACE 2006 [SPACE 20062006], pages 83–90.

[Eggers and Larus, 2008] Susan J. Eggers and James R. Larus, editors. *Proceedings of the Thirteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGPLAN Notices 43(3), Seattle, WA, USA, March 2008.

[Egudo, 1992] R. Egudo. An analysis of a garbage collection operation. *International journal of mathematical education in science and technology*, 23(1):89–96, January 1992.

[Ehn, 1989a] L. Ehn. A contribution to the increase of efficiency of on-the-fly garbage collection. *Computers And Artificial Intelligence*, 8(1):83–91, 1989.

[Ehn, 1989b] L. Ehn. Performance analysis of on-the-fly garbage collection systems. *Computers And Artificial Intelligence*, 8(2):141–152, 1989.

[El Desokey *et al.*, 2006] Ali Ebrahim El Desokey, Aida Abd El Gawad, Amany Sarhan, and Seham Moawed. Improving the performance of the deferrable server based garbage collection scheduling strategy. In *ITI 4th International Conference on Information and Communications Technology*, Cairo, Egypt, 2006.

[El-Habbash *et al.*, 1990] Ahmed El-Habbash, Chris Horn, and Neville Harris. Garbage collection in an object oriented, distributed, persistent environment. In Jul and Juul [Jul and Juul1990].

[Ellis and Detlefs, 1993] John R. Ellis and David L. Detlefs. Safe, efficient garbage collection for C++. Technical report, Xerox PARC, Palo Alto, CA, 1993.

[Ellis and Olson, 1988] C. S. Ellis and T. J. Olson. Algorithms for parallel memory allocation. *International Journal of Parallel Programming*, 17(4):303–345, 1988.

[Ellis and Stroustrup, 1990] Margaret A. Ellis and Bjarne Stroustrup. *The Annotated C++ Reference Manual*. Addison-Wesley, 1990.

[Ellis *et al.*, 1988] John R. Ellis, Kai Li, and Andrew W. Appel. Real-time concurrent collection on stock multiprocessors. Technical Report DEC–SRC–TR–25, DEC Systems Research Center, Palo Alto, CA, February 1988.

[Ellis, 1993] John R. Ellis. Put up or shut up. In Moss et al. [Moss *et al.*1993].

[Ellis, 1995] John R. Ellis. Tutorial: Is safe C++ an oxymoron? In PLDI 1995 [PLDI 19951995].

[Elsman and Hallenberg, 1995] Martin Elsman and Niels Hallenberg. An optimizing back-end for the ML Kit using a stack of regions. Student Project 95–7–8, Department of Computer Science (DIKU), University of Copenhagen, July 1995.

[Elsman, 2001] Martin Elsman. A stack machine for region based programs. In SPACE 2001 [SPACE 20012001].

[Elsman, 2003] Martin Elsman. Garbage collection safety for region-based memory management. In Shao and Lee [Shao and Lee2003], pages 123–134.

[Elson, 1975] M. Elson. *Data Structures*. Science Research Associates, 1975.

[Endo *et al.*, 1997] Toshio Endo, Kenjiro Taura, and Akinori Yonezawa. A scalable mark-sweep garbage collector on large-scale shared-memory machines. In *Proceedings of High Performance Computing and Networking (SC'97)*, 1997.

[Endo *et al.*, 2002] Toshio Endo, Kenjiro Taura, and Akinori Yonezawa. Reducing pause time of conservative collectors. In Boehm and Detlefs [Boehm and Detlefs2002], pages 12–24.

[Endo, 1998] Toshio Endo. A scalable mark-sweep garbage collector on large-scale shared-memory machines. Master's thesis, University of Tokyo, February 1998.

[Endo, 2001] Toshio Endo. *Scalable Dynamic Memory Management Module on Shared Memory Multiprocessors*. PhD thesis, University of Tokyo, September 2001.

[Engelstad and Vandendorpe, 1991] Steven L. Engelstad and James E. Vandendorpe. Automatic storage management for systems with real time constraints. In Wilson and Hayes [Wilson and Hayes1991a].

[Ennals and Brewer, 2007] David Gay Rob Ennals and Eric Brewer. Safe manual memory management. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 2–14.

[Ershov, 1958] A. P. Ershov. On programming of arithmetic operations. *Communications of the ACM*, 1(8):3–6, August 1958.

[ESOP 2003, 2003] *12th European Symposium on Programming (ESOP 2003)*. Elsevier, 2003. Also published in Science of Computer Programming, 50(1–3), March 2004.

[Etienne, 2004] Lozes Etienne. Separation logic preserves the expressive power of classical logic. In SPACE 2004 [SPACE 20042004].

[Evans and Dickman, 1997] Huw Evans and Peter Dickman. Garbage collection and memory management. In OOPSLA 1997 [OOPSLA 19971997], pages 138–143. Addendum to proceedings.

[Evans, 1996] David Evans. Static detection of dynamic memory errors. In PLDI 1996 [PLDI 19961996], pages 44–53.

[Even and Shiloach, 1981] Shimon Even and Yossi Shiloach. An on-line edge-deletion problem. *Journal of the ACM*, 28(1):1–4, January 1981.

[Explorer, 1987] *Explorer (tm) System Software Design Notes*, June 1987. Texas Instruments part number 2243208–0001*A.

[Fabri, 1979] Janet Fabri. Automatic storage optimization. In SIGPLAN 1979 [SIGPLAN 19791979], pages 83–91.

[Faes *et al.*, 2005] Philippe Faes, Mark Christiaens, Dries Buytaert, and Dirk Stroobandt. FPGA-aware garbage collection in Java. In T. Rissa, S. Wilton, and P. Leong, editors, *2005 International Conference on Field Programmable Logic and Applications (FPL)*, pages 675–680, Tampere, Finland, 1 2005. IEEE.

[Falcone and Stinger, 1983] J. R. Falcone and J. R. Stinger. The Smalltalk-80 implementation at Hewlett–Packard. In Krasner [Krasner1983], pages 79–112.

[Farkas *et al.*, 2000] Keith I. Farkas, Jason Flinn, Godmar Back, Dirk Grunwald, and Jennifer Anderson. Quantifying the energy consumption of a pocket computer and a Java virtual machine. In *Proceedings of ACM SIGMETRICS'00 International Conference on Measurement and Modelling of Computer Systems*, 2000.

[Farrens and Park, 1991] Matthew Farrens and Arvin Park. Dynamic base register caching: A technique for reducing address bus width. In *Proc. 18th Annual Int'l Symposium on Computer Architecture*, pages 128–137, May 1991. Toronto, Canada.

[Feeley and Miller, 1990] Marc Feeley and James S. Miller. A parallel virtual machine for efficient Scheme compilation. In LFP 1990 [LFP 19901990], pages 119–130.

[Feeley, 1993] Marc Feeley. Polling efficiently on stock hardware. In Hughes [Hughes1993], pages 179–187.

[Feizabadi and Black, 2005] Shahrooz Feizabadi and Godmar Black. Java garbage collection scheduling in utility accrual scheduling environments. In JTRES 2005 [JTRES 20052005].

[Feizabadi and Black, 2007] Shahrooz Feizabadi and Godmar Black. Garbage collection-aware sheduling utility accrual scheduling environments. *Real-Time Systems*, 36(1–2), July 2007.

[Fenichel and Yochelson, 1969] Robert R. Fenichel and Jerome C. Yochelson. A Lisp garbage collector for virtual memory computer systems. *Communications of the ACM*, 12(11):611–612, November 1969.

[Fenichel, 1971a] Robert R. Fenichel. Comment on cheney's list-compaction algorithm. *Communications of the ACM*, 14(9):603–604, September 1971.

[Fenichel, 1971b] Robert R. Fenichel. List tracing in systems allowing multiple cell types. *Communications of the ACM*, 14(8):522–526, August 1971.

[Fenton and Payne, 1974] J. S. Fenton and D. W. Payne. Dynamic storage allocation of arbitrary sized segments. In *Proceedings of IFIPS*, pages 344–348, 1974.

[Ferguson, 1976] H. R. P. Ferguson. On a generalization of the Fibonacci numbers useful in memory allocation schema. *The Fibonacci Quarterly*, 14(3):233–243, October 1976.

[Fernandez and Hanson, 1992] Mary F. Fernandez and David R. Hanson. Garbage collection alternatives for Icon. *Software Practice and Experience*, 22(8):659–672, August 1992.

[Ferrante and McKinley, 2007] Jeanne Ferrante and Kathryn S. McKinley, editors. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 42(6), San Diego, CA, USA, June 2007.

[Ferrari, 1974] Domenico Ferrari. Improving locality by critical working sets. *Communications of the ACM*, 17(11):614–620, November 1974.

[Ferreira and Shapiro, 1994a] Paulo Ferreira and Marc Shapiro. Distributed shared memory consistency and garbage collection. In *Workshop Inter-PRS*, Paris, December 1994.

[Ferreira and Shapiro, 1994b] Paulo Ferreira and Marc Shapiro. Garbage collection and DSM consistency. In *First Symposium on Operating Systems Design and Implementation*, pages 229–241, Monterey, CA, November 1994. ACM Press.

[Ferreira and Shapiro, 1994c] Paulo Ferreira and Marc Shapiro. Garbage collection of persistent objects in distributed shared memory. In *Workshop Franco-Isralienne*, St Malo, France, September 1994.

[Ferreira and Shapiro, 1995a] Paulo Ferreira and Marc Shapiro. Garbage collection in the Larchant persistent distributed shared store. In *Fifth Workshop on Future Trends in Distributed Computing Systems*, Cheju Island, Korea, August 1995.

[Ferreira and Shapiro, 1995b] Paulo Ferreira and Marc Shapiro. Garbage collection of persistent objects in distributed shared memory. In Atkinson et al. [Atkinson *et al.*1995], pages 176–191.

[Ferreira and Shapiro, 1996a] Paulo Ferreira and Marc Shapiro. Asynchronous distributed garbage collection in the Larchant cached shared store. Available from Marc Shapiro, May 1996.

[Ferreira and Shapiro, 1996b] Paulo Ferreira and Marc Shapiro. Larchant: Persistence by reachability in distributed shared memory through garbage collection. In *Sixteenth International Conference on Distributed Computer Systems*, Hong Kong, May 1996.

[Ferreira and Shapiro, 1998] Paulo Ferreira and Marc Shapiro. Modelling a distributed cached store for garbage collection: the algorithm and its correctness proof. In Jul [Jul1998].

[Ferreira and Veiga, 2005] Paulo Ferreira and Luis Veiga. Garbage collection curriculum. MSDN Academic Alliance Curriculum Repository, object ID 6812, July 2005.

[Ferreira and Veiga, 2006] Paulo Ferreira and Luis Veiga. Mobile middleware – seamless service access via resource replication. In Paolo Bellavista and Antonio Corradi, editors, *The Handbook of Mobile Middleware*. Auerbach Publications, Taylor and Francis-CRC Press, October 2006.

[Ferreira *et al.*, 1998] Paulo Ferreira, Marc Shapiro, Xavier Blondel, Olivier Fambon, Jo ao Garcia, Sytse Kloosterman, Nicolas Richer, Marcus Roberts, Fadi Sandakly, George Colouris, Jean Dollimore, Paulo Guedes, Daniel Hagimont, and Sacha Krakowiak. PerDiS: Design, implementation and use of a PERsistent DIstributed Store. Technical Report QMW TR 752, CSTB ILC/98-1392, INRIA RR 3525, INESC RT/5/98, QMW, CSTB INRIA and INESC, October 1998.

[Ferreira *et al.*, 2000] Paulo Ferreira, Marc Shapiro, Xavier Blondel, Olivier Fambon, Jo ao Garcia, Sytse Kloosterman, Nicolas Richer, Marcus Roberts, Fadi Sandakly, George Colouris, Jean Dollimore, Paulo Guedes, Daniel Hagimont, and Sacha Krakowiak. PerDiS: Design, implementation and use of a PERsistent DIstributed Store. In Krakowiak and Shrivastava [Krakowiak and Shrivastava2000], pages 427–452.

[Ferreira *et al.*, 2003] Paulo Ferreira, Luis Veiga, and Carlos Ribeiro. OBIWAN: Design and implementation of a middleware platform. *IEEE Transactions on Parallel and Distributed Systems*, 14(11):1086–1099, November 2003.

[Ferreira, 1990] Paulo Ferreira. Storage reclamation. In Jul and Juul [Jul and Juul1990].

[Ferreira, 1991a] Paulo Ferreira. Garbage collection in C++. In Wilson and Hayes [Wilson and Hayes1991a].

[Ferreira, 1991b] Paulo Ferreira. Garbage collection in C++. In *Workshop on Extensions to C++*, Lisbon, June 1991.

[Ferreira, 1991c] Paulo Ferreira. Reclaiming storage in an object-oriented platform supporting extended C++ and Objective-C applications. In Cabrera et al. [Cabrera *et al.*1991].

[Ferreira, 1992] Paulo Ferreira. Reciclagem automática de memória num sistema orientado a objectos. Master's thesis, Technical University of Lisbon, June 1992.

[Ferreira, 1996] Paulo Ferreira. *Larchant: garbage collection in a cached distributed shared store with persistence by reachability*. PhD thesis, Université Paris VI, Pierre et Marie Curie, May 1996.

[Field and Harrison, 1988] Anthony J. Field and Peter G. Harrison. *Functional Programming*. Addison-Wesley, 1988.

[Fink and Qian, 2003] Stephen J. Fink and Feng Qian. Design, implementation and evaluation of adaptive recompilation with on-stack replacement. In CGO 2003 [CGO 20032003], pages 241–252.

[Fisher, 1974] David A. Fisher. Bounded workspace garbage collection in an address order preserving list processing environment. *Information Processing Letters*, 3(1):25–32, July 1974.

[Fisher, 1975] David A. Fisher. Copying cyclic list structure in linear time using bounded workspace. *Communications of the ACM*, 18(5):251–252, May 1975.

[Fitch and Norman, 1978] John P. Fitch and Arthur C. Norman. A note on compacting garbage collection. *Computer Journal*, 21(1):31–34, February 1978.

[Fitzgerald and Tarditi, 2000] Robert Fitzgerald and David Tarditi. The case for profile-directed selection of garbage collectors. In Chambers and Hosking [Chambers and Hosking2000].

[Fiuczynski *et al.*, 2010] Marc E. Fiuczynski, Emery Berger, and Andrew Warfield, editors. *Proceedings of the Sixth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, Pittsburgh, PA, USA, March 2010.

[Flanagan and Nikhil, 1996] Cormac Flanagan and Rishiyur S. Nikhil. *pHluid*: the design of a parallel functional language implementation on workstations. In ICFP 1996 [ICFP 19961996], pages 169–179.

[Fleisch, 1989] B. D. Fleisch. Mirage: A coherent distributed shared memory design. In *Proceedings of 12th ACM Symposium on Operating Systems Principles*, pages 211–213. ACM Press, December 1989.

[Flood *et al.*, 2001] Christine Flood, Dave Detlefs, Nir Shavit, and Catherine Zhang. Parallel garbage collection for shared memory multiprocessors. In JVM 2001 [JVM 20012001].

[Fluet and Wang, 2004] Matthew Fluet and Daniel Wang. Implementation and performance evaluation of a safe runtime system in Cyclone. In SPACE 2004 [SPACE 20042004].

[Fluet, 2004] Matthew Fluet. Monadic regions. In SPACE 2004 [SPACE 20042004].

[Foderaro and Fateman, 1981] John K. Foderaro and Richard J. Fateman. Characterization of VAX Macsyma. In *1981 ACM Symposium on Symbolic and Algebraic Computation*, pages 14–19, Berkeley, CA, 1981. ACM Press.

[Foderaro *et al.*, 1985] John K. Foderaro, Keith Sklower, Kevin Layer, et al. *Franz Lisp Reference Manual*. Franz Inc., 1985.

[Ford, 1988] R. Ford. Concurrent algorithms for real-time memory management. *IEEE Software*, pages 10–23, September 1988.

[Forin *et al.*, 1989] A. Forin, J. Barrera, M. Young, and R. Rashid. Design, implementation, and performance evaluation of a distributed shared memory server for Mach. In *Proceedings of the 1989 Winter USENIX Conference*. USENIX Association, January 1989.

[Foster and Winsborough, 1991] Ian Foster and William Winsborough. Copy avoidance through compile-time analysis and local reuse. In *Proceedings of International Logic Programming Symposium*, pages 455–469, 1991.

[Foster, 1968] J. M. Foster. *List Processing*. Elsevier Computer Monographs. Elsevier-North Holland, New York, 1968.

[Foster, 1985] Mark H. Foster. Design of a list-structure memory using parallel garbage collection. Master's thesis, Oregon Graduate Center, 1985.

[Foster, 1988] Ian Foster. An asynchronous parallel garbage collector for a single-assignment language. Technical report, Imperial College, London, May 1988.

[Foster, 1989a] Ian Foster. Copy avoidance through local reuse. Technical Report MCS–P99–0989, Argonne National Laboratory, 1989.

[Foster, 1989b] Ian Foster. A multicomputer garbage collector for a single-assignment language. *International Journal of Parallel Programming*, 18(3):181–203, 1989.

[Fotheringham, 1961] J. A. Fotheringham. Dynamic storage allocation in the Atlas computer including an automatic use of the backing store. *Communications of the ACM*, 4:435, 1961.

[FPCA 1989, 1989] *Record of the 1989 Conference on Functional Programming and Computer Architecture*, Imperial College, London, August 1989. ACM Press.

[FPCA 1995, 1995] *Record of the 1995 Conference on Functional Programming and Computer Architecture*, June 1995.

[Fradet, 1994] Pascal Fradet. Collecting more garbage. In LFP 1994 [LFP 19941994], pages 24–33.

[Frampton *et al.*, 2007] Daniel Frampton, David F. Bacon, Perry Cheng, and David Grove. Generational real-time garbage collection: A three-part invention for young objects. In ECOOP 2007 [ECOOP 20072007].

[Frampton *et al.*, 2009a] Daniel Frampton, Stephen M. Blackburn, Perry Cheng, Robin J. Garner, D. Grove, J. Eliot B. Moss, and S. I. Salishev. Demystifying magic: High-level low-level programming. In Hosking et al. [Hosking *et al.*2009], pages 81–90.

[Frampton *et al.*, 2009b] Daniel Frampton, Stephen M. Blackburn, Luke N. Quinane, and John N. Zigman. Efficient concurrent mark-sweep cycle collection. Technical Report TR-CS-09-02, The Australian National University, October 2009.

[Frampton, 2003] Daniel Frampton. An investigation into automatic dynamic memory management strategies using compacting collection. Honours thesis, 2003.

[Frampton, 2010] Daniel Frampton. *Garbage Collection and the Case for High-level Low-level Programming*. PhD thesis, Australian National University, June 2010.

[Francez and Kozen, 1984] Nissim Francez and Dexter Kozen. Generalized fair termination. In POPL 1984 [POPL 19841984], pages 46–53.

[Francez, 1978] Nissim Francez. An application of a method for analysis of cyclic programs. *ACM Transactions on Software Engineering*, 4(5):371–377, September 1978.

[Francez, 1980] Nissim Francez. Distributed termination. *ACM Transactions on Programming Languages and Systems*, 2(1):42–55, January 1980.

[Franklin *et al.*, 1989] Michael Franklin, G. Copeland, and G. Weikum. What's different about garbage collection for persistent programming languages. Technical Report ACA-ST-062-89, Microelectronics and Computer Technology Corporation, 1989.

[Franz, 1988, 1988] Franz Inc. *Allegro CL User Guide*, release 3.0 (beta) edition, April 1988.

[Franz, 1992, 1992] Franz Inc. *Allegro CL User Guide, Version 4.1*, revision 2 edition, March 1992.

[Freiburger *et al.*, 1975] W. F. Freiburger, U. Grenander, and P. D. Sampson. Patterns in program references. *IBM Journal of Research and Development*, 19(3):230–243, May 1975.

[Friedman and Wise, 1976] Daniel P. Friedman and David S. Wise. Garbage collecting a heap which included a scatter table. *Information Processing Letters*, 5(6):161–164, December 1976.

[Friedman and Wise, 1978] Daniel P. Friedman and David S. Wise. Aspects of applicative programming for parallel processing. *IEEE Transactions on Computers*, 27(4):289–296, April 1978.

[Friedman and Wise, 1979] Daniel P. Friedman and David S. Wise. Reference counting can manage the circular environments of mutual recursion. *Information Processing Letters*, 8(1):41–45, January 1979.

[Fu and Hauser, 2005a] Wei Fu and Carl Hauser. Modeling real-time garbage collection cost. In RTCSA 2007 [RTCSA 20072005].

[Fu and Hauser, 2005b] Wei Fu and Carl Hauser. A real-time garbage collection framework for embedded systems. In *Proceedings of the 2005 workshop on Software and compilers for embedded systems*, pages 20–26, Dallas, TX, 2005.

[Fuchs, 1995] Matthew Fuchs. Garbage collection on an open network. In Baker [Baker1995a].

[Furusou *et al.*, 1991] Shinichi Furusou, Satoshi Matsuoka, and Akinori Yonezawa. Parallel conservative garbage collection with fast allocation. In Wilson and Hayes [Wilson and Hayes1991a].

[Gabriel and Mansinter, 1982] Richard P. Gabriel and L. M. Mansinter. Performance of Lisp systems. In LFP 1982 [LFP 19821982], pages 123–142.

[Gabriel, 1985] Richard P. Gabriel. *Performance and Evaluation of Lisp Systems*. MIT Press Series in Computer Science. MIT Press, Cambridge, MA, 1985.

[Gai and Mezzalama, 1985] S. Gai and M. Mezzalama. Dynamic storage allocation: Experiments using the C language. *Software Practice and Experience*, 15(7):693–704, July 1985.

[Ganesan, 1994] Ravichandran Ganesan. Local variable allocation for accurate garbage collection of C++. Master's thesis, Iowa State University, July 1994. Technical Report ISUTR 94–12.

[Gannon *et al.*, 1988] D. Gannon, W. Jalby, and K. Gallivan. Strategies for cache and local memory management by global program transformation. *Journal of Parallel and Distributed Computing*, 5:587–616, 1988.

[Gao and Nilsen, 1994] Hong Gao and Kelvin Nilsen. Reliable general purpose dynamic memory management for real-time systems. Technical Report TR94–09, Iowa State University, July 1994.

[Garey *et al.*, 1972] M. R. Garey, R. L. Graham, and Jeffrey D. Ullman. Worst-case analysis of memory allocation algorithms. In *Fourth Annual ACM Symposium on the Theory of Computing*. ACM Press, 1972.

[Garner *et al.*, 2007] Robin Garner, Stephen M. Blackburn, and Daniel Frampton. Effective prefetch for mark-sweep garbage collection. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 43–54.

[Garnett and Needham, 1980] N. H. Garnett and Roger M. Needham. An asynchronous garbage collector for the Cambridge file server. *ACM SIGOPS Operating Systems Review*, 14(4):36–40, 1980.

[Garthwaite and Nettles, 1997] Alex Garthwaite and Scott Nettles. Concurrent collection for the Java Development Kit. In Dickman and Wilson [Dickman and Wilson1997].

[Garthwaite and Nettles, 1998] Alex Garthwaite and Scott Nettles. TJava: a transactional Java. In *IEEE International Conference on Computer Languages*. IEEE Press, 1998.

[Garthwaite *et al.*, 2005] Alex Garthwaite, Dave Dice, and Derek White. Supporting per-processor local-allocation buffers using lightweight user-level preemption notification. In Hind and Vitek [Hind and Vitek2005], pages 24–34.

[Garthwaite, 2001] Alex Garthwaite. Memory management = partitioning + alpha-renaming. In SPACE 2001 [SPACE 20012001].

[Garthwaite, 2005] Alex Garthwaite. *Making the Trains Run On Time*. PhD thesis, University of Pennsylvania, 2005.

[Gay and Aiken, 1998] David Gay and Alexander Aiken. Memory management with explicit regions. In PLDI 1998 [PLDI 19981998], pages 313–323.

[Gay and Aiken, 2001] David Gay and Alexander Aiken. Language support for regions. In PLDI 2001 [PLDI 20012001], pages 70–80.

[Gay and Steensgaard, 1998] David Gay and Bjarne Steensgaard. Stack allocating objects in Java. Technical report, Microsoft Research, October 1998.

[Gay and Steensgaard, 2000] David Gay and Bjarne Steensgaard. Fast escape analysis and stack allocation for object-based programs. In *International Conference on Compiler Construction(CC'2000)*, volume 1781 of *Lecture Notes in Computer Science*. Springer-Verlag, 2000.

[Gay, 2001] David Gay. A type system for reference-counted regions. In SPACE 2001 [SPACE 20012001].

[Gee *et al.*, 1993] Jeffrey D. Gee, Mark D. Hill, Dionisios N. Pnevmatikatos, and Alan J. Smith. Cache performance of the SPEC92 benchmark suite. *IEEE Micro*, 13(4):17–27, 1993.

[Gehringer and Chang, 1993] Edward F. Gehringer and Ellis Chang. Hardware-assisted memory management. In Moss et al. [Moss *et al*.1993].

[Gelenbe, 1971] E. Gelenbe. The two-thirds rule for dynamic storage allocation under equilibrium. *Information Processing Letters*, 1:59–60, 1971.

[Gelernter *et al.*, 1960] H. Gelernter, J. R. Hansen, and C. L. Gerberich. A Fortran-compiled list processing language. *Journal of the ACM*, 7(2):87–101, April 1960.

[Georges *et al.*, 2004] Andy Georges, Dries Buytaert, Lieven Eeckhout, and Koen De Bosschere. Method-level phase behavior in Java workloads. In OOPSLA 2004 [OOPSLA 20042004], pages 270–287.

[Georges *et al.*, 2007] Andy Georges, Dries Buytaert, and Lieven Eeckhout. Statistically rigorous Java performance evaluation. In OOPSLA 2007 [OOPSLA 20072007], pages 57–76.

[Georges *et al.*, 2008] Andy Georges, Lieven Eeckhout, and Dries Buytaert. Java performance evaluation through rigorous replay compilation. In OOPSLA 2008 [OOPSLA 20082008], pages 367–384.

[Gerhart, 1979] S. L. Gerhart. A derivation oriented proof of Schorr–Waite marking algorithm. *Lecture Notes in Computer Science*, 69:472–492, 1979.

[Gheorghioiu *et al.*, 2003] Ovidiu Gheorghioiu, Alexandru Salcianu, and Martin Rinard. Interprocedural compatibility analysis for static object preallocation. In POPL 2003 [POPL 20032003].

[Ghesquiere *et al.*, 2003a] T. Ghesquiere, J.-D. Choi, and K. De Bosschere. Accurate replay of memory management in Java. In *Fourth FTW PhD Symposium*, Gent, Belgium, December 2003.

[Ghesquiere *et al.*, 2003b] T. Ghesquiere, J.-D. Choi, and K. De Bosschere. Memory management replay in DejaVu. In *Program Acceleration through Application and Architecture driven Code Transformations: Symposium Proceedings*, pages 113–115, Edegem, Belgium, September 2003.

[Ghiya *et al.*, 2001] Rakesh Ghiya, Daniel M. Lavery, and David C. Sehr. On the importance of points-to analysis and other memory disambiguation methods for C programs. In PLDI 2001 [PLDI 20012001], pages 47–58.

[Ghosh, 1994] Kaushik Ghosh. Reconfigurable garbage collection of data structures in a speculative real-time system. Technical Report GIT-CC-94-57, Georgia Institute of Technology, 1994.

[Gibbs and Coady, 2005] Celina Gibbs and Yvonne Coady. Aspects of memory management. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS05) - Track 9*, page 275.2, 2005.

[Gillam,] Richard Gillam. An introduction to garbage collection. http://oss.software.ibm.com/icu/docs/papers/cpp_report/an_introduction_to_garbage_collection_part_i.html.

[Ginter, 1991] Andrew Ginter. Cooperative garbage collection using smart pointers in the C++ programming language. Master's thesis, University of Calgary, December 1991. Technical report 91/451/45.

[Girard, 1987] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Glaser and Hayes, 1986] Hugh W. Glaser and S. Hayes. Another implementation technique for applicative languages. In *Proceedings of ESOP'86 — European Symposium on Programming*, volume 213 of *Lecture Notes in Computer Science*, pages 70–81, Saarbruecken, March 1986. Springer-Verlag.

[Glaser and Thompson, 1987] Hugh W. Glaser and P. Thompson. Lazy garbage collection. *Software Practice and Experience*, 17(1):1–4, January 1987.

[Glaser *et al.*, 1989] Hugh W. Glaser, Michael Reeve, and S. Wright. An analysis of reference count garbage collection schemes for declarative languages. Technical report, Department of Computing, Imperial College, London, 1989.

[Glaser, 1987] Hugh W. Glaser. On minimal overhead reference count garbage collection in distributed systems. Technical report, Department of Computing, Imperial College, London, 1987.

[Goa and Nielsen, 1995] H. Goa and Kelvin Nielsen. The real-time behaviour of dynamic memory management in C++. In *Proceedings of the Real-Time Techniques and Applications Symposium*, Chicago, IL, May 1995.

[Goetz, 2003a] Brian Goetz. Java theory and practice: A brief history of garbage collection. http://www-106.ibm.com/developerworks/java/library/j-jtp10283/, October 2003. First of a series of three articles.

[Goetz, 2003b] Brian Goetz. Java theory and practice: Garbage collection in the 1.4.1 JVM. http://www-106.ibm.com/developerworks/java/library/j-jtp11253/, November 2003. Second of a series of three articles.

[Goetz, 2004] Brian Goetz. Java theory and practice: Garbage collection and performance. http://www-106.ibm.com/developerworks/java/library/j-jtp01274/, January 2004. Last of a series of three articles.

[Goguen *et al.*, 1998] Healfdene Goguen, Richard Brooksby, and Rod Burstall. An abstract formulation of memory management, December 1998. draft.

[Goguen *et al.*, 2000] Healfdene Goguen, Richard Brooksby, and Rod M. Burstall. Memory management: An abstract formulation of incremental tracing. In *Types for Proofs and Programs, International Workshop TYPES'99*, pages 148–161. Springer, 2000.

[Goh *et al.*, 2006] O. Goh, Yann-Hang Lee, Z. Kaakani, and E. Rachlin. Integrated scheduling with garbage collection for real-time embedded applications in CLI. In ISORC 2006 [ISORC 20062006], page 8.

[Goldberg and Gloger, 1992] Benjamin Goldberg and Michael Gloger. Polymorphic type reconstruction for garbage collection without tags. In LFP 1992 [LFP 19921992], pages 53–65.

[Goldberg and Hassinger, 1974] Ron P. Goldberg and R. Hassinger. The double paging anomaly. In *AFIPS National Computer Conference*, pages 195–199, May 1974.

[Goldberg and Robson, 1983] Adele Goldberg and D. Robson. *Smalltalk-80: The Language and its Implementation*. Addison-Wesley, 1983.

[Goldberg, 1989] Benjamin Goldberg. Generational reference counting: A reduced-communication distributed storage reclamation scheme. In PLDI 1989 [PLDI 19891989], pages 313–320.

[Goldberg, 1991] Benjamin Goldberg. Tag-free garbage collection for strongly typed programming languages. *ACM SIGPLAN Notices*, 26(6):165–176, 1991.

[Goldberg, 1992] Benjamin Goldberg. Incremental garbage collection without tags. In *Proceedings ESOP92 — European Symposium on Programming*, 1992.

[Golding, 1992] Richard A. Golding. *Weak-Consistency Group Communication and Membership*. PhD thesis, University of California at Santa Cruz, December 1992. UCSC- CRL-92-52.

[Goldman and Gabriel, 1988] Ron Goldman and Richard P. Gabriel. Preliminary results with the initial implementation of Qlisp. In LFP 1988 [LFP 19881988], pages 143–152.

[Gonçalves and Appel, 1994] Marcelo J. R. Gonçalves and Andrew W. Appel. Cache performance of fast-allocating programs. Technical Report CS–TR–482–94, Department of Computer Science, Princeton University, December 1994.

[Gonçalves and Appel, 1995] Marcelo J. R. Gonçalves and Andrew W. Appel. Cache performance of fast-allocating programs. In FPCA 1995 [FPCA 19951995].

[Gonçalves, 1995] Marcelo J. R. Gonçalves. *Cache Performance of Programs with Intensive Heap Allocation and Generational Garbage Collection*. PhD thesis, Department of Computer Science, Princeton University, May 1995.

[Gonthier, 1996] Georges Gonthier. Verifying the safety of a practical concurrent garbage collector. In R. Alur and T. Henzinger, editors, *Computer Aided Verification CAV'96*, Lecture Notes in Computer Science, New Brunswick, NJ, 1996. Springer-Verlag.

[Gorman and Healy, 2008] Mel Gorman and Patrick Healy. Supporting superpage allocation without additional hardware support. In Jones and Blackburn [Jones and Blackburn2008], pages 41–50.

[Gosling *et al.*, 1997] James Gosling, Bill Joy, and Guy Steele. *The Java Language Specification*. The Java Series. Addison-Wesley, 1997.

[Gosling *et al.*, 2005] James Gosling, Bill Joy, Guy Steele, and Gilad Bracha. *The Java Language Specification*. Addison-Wesley, third edition edition, May 2005.

[Gotlieb and Gotlieb, 1978] C. C. Gotlieb and L. R. Gotlieb. *Data Types and Structures*. Prentice-Hall, 1978.

[Goto *et al.*, 1979] E. Goto, I. Tetsuo, K. Hiraki, M. Susuki, and N. Inada. FLATS, a machine for numerical, symbolic and associative computing. In *6th Annual Symposium on Computer Architecture*, pages 102–110, April 1979.

[Goto *et al.*, 1988] Atsuhiro Goto, Y. Kimura, T. Nakagawa, and T. Chikayama. Lazy reference counting: An incremental garbage collection method for parallel inference machines. In ICLP 1988 [ICLP 19881988], pages 1241–1256. Also ICOT Technical Report TR-354, 1988.

[Goto, 1974] Eiichi Goto. Monocopy and associative algorithms in an extended LISP. Technical Report Technical Report 74-03, Information Science Laboratories, Faculty of Science, University of Tokyo, 1974.

[Gottlieb and Wilson, 1982] A. Gottlieb and J. Wilson. Parallelizing the usual buddy algorithm. Technical Report System Software Note 37, Courant Institute, New York University, 1982.

[Gourhant *et al.*, 1992] Y. Gourhant, S. Louboutin, V. Cahill, A. Condon, G. Starovic, and B. Tangney. Dynamic clustering in an object-oriented distributed system. In *Proceedings of OLDA-II (Objects in Large Distributed Applications)*, Ottawa, Canada, October 1992.

[Goyer, 1971] P. Goyer. A garbage collector to be implemented on a CDC 3100. In Peck [Peck1971], pages 303–317.

[Gray, 1987] Stanley M. Gray. Garbage collection in a parallel processing environment. Master's thesis, East Texas State University, 1987.

[Greenblat, 1974] Richard Greenblat. The LISP machine. Working Paper 79, MIT AI Laboratory, November 1974.

[Greenblatt, 1984] Richard Greenblatt. The LISP machine. In D. R. Barstow, H. E. Shrobe, and E. Sandewall, editors, *Interactive Programming Environments*. McGraw-Hill, 1984.

[Gregg *et al.*, 2008] David Gregg, Vikram Adve, and Brian Bershad, editors. *Proceedings of the Fourth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, Seattle, WA, USA, March 2008.

[Gries, 1977a] David Gries. An exercise in proving parallel programs correct. *Communications of the ACM*, 20(12):921–930, December 1977.

[Gries, 1977b] David Gries. On believing programs to be correct. *Communications of the ACM*, 20(1):49–50, January 1977.

[Gries, 1979] David Gries. The Schorr–Waite graph marking algorithm. *Acta Informatica*, 11(3):223–232, 1979.

[Griffin *et al.*, 2005a] Paul Griffin, Witawas Srisa-An, and J. Morris Chang. An energy efficient garbage collector for Java embedded devices. In LCTES 2005 [LCTES 20052005], pages 230–238.

[Griffin *et al.*, 2005b] Paul Griffin, Witawas Srisa-An, and J. Morris Chang. On designing a low-power garbage collector for Java embedded devices: A case study. In *Proceedings of the 2005 ACM symposium on Applied Computing*, pages 868–873, Santa Fe, NM, 2005.

[Grimsrud, 1989] Knut S. Grimsrud. Multiple prefetch adaptive disk caching with strategic data layout. Master's thesis, Brigham Young University, December 1989.

[Griswold and Griswold, 1983] Ralph E. Griswold and Madge T. Griswold. *The Icon Programming Language*. Prentice-Hall, 1983.

[Griswold and Griswold, 1986] Ralph E. Griswold and Madge T. Griswold. *The Implementation of the Icon Programming Language*. Princeton University Press, 1986. Out of print.

[Griswold, 1972] R. E. Griswold. The macro implementation of Snobol 4, 1972.

[Grit and Page, 1981] Dale H. Grit and Rex L. Page. Deleting irrelevant tasks in an expression-oriented multiprocessor system. *ACM Transactions on Programming Languages and Systems*, 3(1):49–59, January 1981.

[Grossman *et al.*, 2002] Dan Grossman, Greg Morrisett, Trevor Jim, Michael Hicks, Yanling Wang, and James Cheney. Region-based memory management in Cyclone. In PLDI 2002 [PLDI 20022002], pages 282–293.

[Grossman, 2007] Dan Grossman. The transactional memory / garbage collection analogy. In OOPSLA 2007 [OOPSLA 20072007], pages 695–706. Essay session.

[Grouleff, 1999] Morten Grouleff. A concurrent garbage collector for BETA. Master's thesis, University of Aarhus, December 1999.

[Gruian and Salcic, 2005] Flavius Gruian and Zoran Salcic. Designing a concurrent hardware garbage collector for small embedded systems. In Thambipillai Srikanthan, Jingling Xue, and Chip-Hong Chang, editors, *Advances in Computer Systems Architecture*, volume 3740 of *Lecture Notes in Computer Science*, pages 281–294. Springer-Verlag, 2005.

[Grunwald and Zorn, 1992] Dirk Grunwald and Benjamin Zorn. CUSTOMALLOC: Efficient synthesized memory allocators. Computer Science Technical Report CU-CS-602-92, University of Colorado, Campus Box 430, Boulder, CO 80309, July 1992.

[Grunwald and Zorn, 1993] Dirk Grunwald and Benjamin Zorn. Customalloc: Efficient, synthesised memory allocators. *Software Practice and Experience*, 23:851–869, 1993.

[Grunwald *et al.*, 1993] Dirk Grunwald, Benjamin G. Zorn, and Robert Henderson. Improving the cache locality of memory allocation. In PLDI 1993 [PLDI 19931993], pages 177–186.

[Grzegorczyk *et al.*, 2007] Chris Grzegorczyk, Sunil Soman, Chandra Krintz, and Rich Wolski. Isla Vista heap sizing: Using feedback to avoid paging. In *CGO '07: Proceedings of the International Symposium on Code Generation and Optimization*, pages 325–340. IEEE Computer Society, 2007.

[Gu *et al.*, 2009] Xiaoming Gu, Ian Christopher, Tongxin Bai, Chengliang Zhang, and Chen Ding. A component model of spatial locality. In Kolodner and Steele [Kolodner and Steele2009], pages 99–108.

[Guan *et al.*, 2009] Xiaohua Guan, Witawas Srisa-an, and Chenghuan Jia. Investigating the effects of using different nursery sizing policies on performance. In Kolodner and Steele [Kolodner and Steele2009], pages 59–68.

[Gudeman, 1993] David Gudeman. Representing type information in dynamically-typed languages. Technical Report TR93-27, University of Arizona, Department of Computer Science, Tucson, Arizona, 1993.

[Guggilla, 1994] Satish Kumar Guggilla. Generational garbage collection of C++ targeted to SPARC architectures. Master's thesis, Iowa State University, July 1994. Technical report ISUTR 94-11.

[Gupta and Amarasinghe, 2008] Rajiv Gupta and Saman P. Amarasinghe, editors. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 43(6), Tucson, AZ, USA, June 2008.

[Gupta and Fuchs, 1988] Aloke Gupta and W. K. Fuchs. Reliable garbage collection in distributed object oriented systems. In *Proceedings of the Twelfth Annual International Computer Software Applications Conference (COMPSAC 88)*, pages 324–328, Chicago, October 1988. IEEE Press.

[Gupta and Fuchs, 1993] Aloke Gupta and W. K. Fuchs. Garbage collection in a distributed object-oriented system. *IEEE Transactions on Knowledge and Data Engineering*, 5(2), April 1993.

[Gupta, 1990] Aloke Gupta. Low overhead garbage collection in a distributed object-oriented system. Master's thesis, University of Illinois at Urbana-Champaign, 1990.

[Gupta, 2003] Alka Gupta. GC portal. http://java.sun.com/developer/technicalArticles/Programming/GCPortal/, July 2003.

[Guttman *et al.*, 1995] J.D. Guttman, J.D. Ramsdel, and V. Swarup. The VLISP verified Scheme system. *Lisp and Symbolic Computation*, 8(1/2):33–110, March 1995.

[Guyer and McKinley, 2004] Samuel Guyer and Kathryn McKinley. Finding your cronies: Static analysis for dynamic object colocation. In OOPSLA 2004 [OOPSLA 20042004], pages 237–250.

[Guyer *et al.*, 2006] Samuel Z. Guyer, Kathryn S. McKinley, and Daniel Frampton. Free-Me: A static analysis for automatic individual object reclamation. In Schwartzbach and Ball [Schwartzbach and Ball2006], pages 364–375.

[Guzmán and Hudak, 1990] J. C. Guzmán and Paul Hudak. Single threaded polymorphic lambda calculus. In *Fifth IEEE Symposium on Logic in Computer Science*. IEEE Press, 1990.

[GWFP 1993, 1993] *Sixth Annual Glasgow Workshop on Functional Programming*, Workshops in Computer Science. Springer-Verlag, 1993.

[Haddon and Waite, 1967] B. K. Haddon and W. M. Waite. A compaction procedure for variable length storage elements. *Computer Journal*, 10:162–165, August 1967.

[Hagan, 1996] Tom Hagan. Not just a stopgap. *Information Week*, January 1996.

[Häggander and Lundberg, 1998] Daniel Häggander and Lars Lundberg. Optimizing dynamic memory management in a multithreaded application executing on a multiprocessor. In *Proceedings of ICPP'98 27th International Conference on Parallel Processing*, Minneapolis, MN, August 1998.

[Häggander and Lundberg, 1999] Daniel Häggander and Lars Lundberg. Memory allocation prevented telecommunication application to be parallelized for better database utilization. In *Proceedings of PART'99 6th International Australasian Conference on Parallel and Real-Time Systems*, Melbourne, November 1999.

[Häggander and Lundberg, 2000] Daniel Häggander and Lars Lundberg. Attacking the dynamic memory problem for SMPs. In *Proceedings of PDCS'2000 13th International Conference on Parallel and Distributed Computing System*, 2000.

[Häggander *et al.*, 2001] Daniel Häggander, Per Liden, and Lars Lundberg. A method for automatic optimization of dynamic memory management in C++. In *Proceedings of ICPP'01 30th International Conference on Parallel Processing*, Valencia, Spain, September 2001.

[Hall *et al.*, 1988] Cordelia Hall, R. John M. Hughes, and John T. O'Donnell, editors. *Glasgow Workshop on Functional Programming*. Glasgow University Computer Science Report 89/R4, February 1988.

[Hallenberg *et al.*, 2002] Niels Hallenberg, Martin Elsman, and Mads Tofte. Combining region inference and garbage collection. In PLDI 2002 [PLDI 20022002], pages 141–152.

[Hallenberg, 1996] Niels Hallenberg. A region profiler for a Standard ML compiler based on region inference. Student Project 96–5–7, Department of Computer Science (DIKU), University of Copenhagen, June 1996.

[Hallenberg, 1999] N. Hallenberg. Combining garbage collection and region inference in the ML Kit. Master's thesis, Department of Computer Science (DIKU), University of Copenhagen, June 1999.

[Halpern *et al.*, 1984] Joseph Y. Halpern, Albert R. Meyer, and B. A. Trakhtenbrot. The semantics of local storage, or what makes the free-list free? In POPL 1984 [POPL 19841984], pages 245–257.

[Halstead, 1978] Robert H. Halstead. Multiple-processor implementations of message passing systems. Technical Report TR–198, MIT Laboratory for Computer Science, April 1978.

[Halstead, 1984] Robert H. Halstead. Implementation of Multilisp: Lisp on a multiprocessor. In Steele [Steele1984].

[Halstead, 1985] Robert H. Halstead. Multilisp: A language for concurrent symbolic computation. *ACM Transactions on Programming Languages and Systems*, 7(4):501–538, October 1985.

[Hamid and Crowe, 1992] T. Hamid and M.K. Crowe. Garbage collection in large scale distributed object stores. In *Objects in Large Distributed Applications (OLDA II) — OOPSLA'92*, 1992.

[Hamilton and Jones, 1990] G. W. Hamilton and Simon B. Jones. Compile-time garbage collection by necessity analysis. Technical Report 67, Department of Computer Science and Mathematics, University of Stirling, 1990.

[Hamilton and Jones, 1991] G. W. Hamilton and Simon B. Jones. Compile-time garbage collection by necessity analysis. In Peyton Jones et al. [Peyton Jones *et al.*1991], pages 66–70.

[Hamilton, 1993] G. W. Hamilton. *Compile-Time Optimisation of Store Usage in Lazy Funtional Programs*. PhD thesis, University of Stirling, 1993.

[Hamilton, 1995] G. W. Hamilton. Compile-time garbage collection for lazy functional languages. In Baker [Baker1995a].

[Hamilton, 1997] Craig Hamilton. Measuring the performance of disk garbage collectors: Garbage collecting persistent Java stores. Master's thesis, University of Glasgow, 1997.

[Hammer and Acar, 2008] Matthew Hammer and Umut Acar. Memory management for self-adjusting computation. In Jones and Blackburn [Jones and Blackburn2008], pages 51–60.

[Hammond *et al.*, 1994] Kevin Hammond, Geoff L. Burn, and D. B. Howe. Spiking your caches. Glasgow University, 1994.

[Hampton, 2003] Matthew Hampton. Using contaminated garbage collection and reference counting garbage collection to provide automatic reclamation for real-time systems. Master's thesis, Washington University, 2003.

[Han *et al.*, 2006] Longzhe Han, Yeonseung Ryu, and Keunsoo Yim. CATA: A garbage collection scheme for flash memory file systems. In *Ubiquitous Intelligence and Computing*, volume 4159 of *Lecture Notes in Computer Science*, 2006.

[Hannon, 1995] J. Hannon. A type-based analysis for stack allocation in functional languages. In *Proceedings of Second International Static Analysis Sumposium (SAS'95)*, volume 983 of *Lecture Notes in Computer Science*, pages 172–188. Springer, 1995.

[Hansen and Clinger, 2002] Lars Thomas Hansen and William D. Clinger. An experimental study of renewal-older-first garbage collection. In *Proceedings of the 2002 ACM SIGPLANInternational Conference on Functional Programming(ICFP02)*, ACM SIGPLAN Notices 37(9), pages 247–258, Pittsburgh, PA, 2002. ACM Press.

[Hansen, 1969] Wilfred J. Hansen. Compact list representation: Definition, garbage collection, and system implementation. *Communications of the ACM*, 12(9):499–507, September 1969.

[Hansen, 1992] Lars Thomas Hansen. The impact of programming style on the performance of Scheme programs. Master's thesis, University of Oregon, August 1992.

[Hansen, 2000] Lars Thomas Hansen. *Older-first Garbage Collection in Practice*. PhD thesis, North-eastern University, November 2000.

[Hanson, 1977] David R. Hanson. Storage management for an implementation of Snobol 4. *Software Practice and Experience*, 7(2):179–192, 1977.

[Hanson, 1990] David R. Hanson. Fast allocation and deallocation of memory based on object lifetimes. *Software Practice and Experience*, 20(1):5–12, January 1990.

[Harland and Beloff, 1987] David M. Harland and Brune Beloff. OBJEKT — a persistent object store with an integrated garbage collector. *ACM SIGPLAN Notices*, 22(4):70–79, 1987.

[Harms, 1989] Douglas E. Harms. Efficient initialization and finalization of data structures: Why and how. Technical Report OSU-CISRC-3/89-TR11, Ohio State University, Computer and Information Science Research Center, February 1989.

[Harris, 1998a] Warren Harris. Applet lifecycle in Netscape Communicator. Technical Report TN-JAVA-05-9707, Netscape Communications Corporation, 1998.

[Harris, 1998b] Warren Harris. The design and implementation of the Sport Model garbage collector. incomplete draft, January19 1998.

[Harris, 1999] Timothy Harris. Early storage reclamation in a tracing garbage collector. *ACM SIGPLAN Notices*, 34(4):46–53, April 1999.

[Harris, 2000] Timothy Harris. Dynamic adaptive pre-tenuring. In Chambers and Hosking [Chambers and Hosking2000], pages 127–136.

[Harris, 2006] Tim Harris. Leaky regions: Linking reclamation hints to program structure. Technical Report MSR-TR-2006-84, Microsoft Research, June 2006.

[Hart and Evans, 1974] Timothy P. Hart and Thomas G. Evans. Notes on implementing LISP for the M–460 computer. In Berkeley and Bobrow [Berkeley and Bobrow1974], pages 191–203.

[Hartel *et al.*, 1994] Pieter H. Hartel, Marc Feeley, Martin Alt, Lennart Augustsson, Peter Baumann, Marcel Beemster, Emmanuel Chailloux, Christine H. Flood, Wolfgang Grieskamp, John H. G. van Groningen, Kevin Hammond, BogumiłHausman, Melody Y. Ivory, Peter Lee, Xavier Leroy, Sandra Loosemore, Niklas Röjemo, Manuel Serrano, Jean-Pierre Talpin, Jon Thackray, Pierre Weis, and Peter Wentworth. Pseudoknot: A float-intensive benchmark for functional compilers. In J. R. W. Glauert, editor, *6th Implementation of Functional Languages*, pages 13.1–13.34. School of Information Systems, University of East Anglia, Norwich, UK, September 1994.

[Hartel *et al.*, 1996] Pieter H. Hartel, Marc Feeley, Martin Alt, Lennart Augustsson, Peter Baumann, Marcel Beemster, Emmanuel Chailloux, Christine H. Flood, Wolfgang Grieskamp, John H. G. van Groningen, Kevin Hammond, BogumiłHausman, Melody Y. Ivory, Richard Jones, Peter Lee, Xavier Leroy, Rafael Lins, Sandra Loosemore, Niklas Röjemo, Manuel Serrano, Jean-Pierre Talpin, Jon Thackray, Stephen P. Thomas, Pierre Weis, and Peter Wentworth. Benchmarking implementations of functional languages with 'Pseudoknot', a float-intensive benchmark. *Journal of Functional Programming*, 6(4), 1996.

[Hartel, 1988] Pieter H. Hartel. *Performance Analysis of Storage Management in Combinator Graph Reduction*. PhD thesis, Department of Computer Systems, University of Amsterdam, Amsterdam, 1988.

[Hartel, 1990] Pieter H. Hartel. A comparison of 3 garbage collection algorithms. *Structured Programming*, 11(3):117–127, 1990.

[Hasan and Chang, 2003] Yusuf Hasan and J. Morris Chang. A hybrid allocator. In *Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS-03)*, Austin, TX, March 2003.

[Hastings and Joyce, 1992] Reed Hastings and Bob Joyce. Fast detection of memory leaks and access errors. In *Proceedings of the Winter '92 USENIX Conference*, pages 125–136. USENIX Association, 1992.

[HAT,] Sun Microsystems. *Java Heap Analysis Tool (HAT)*. http://java.sun.com/people/billf/heap/.

[Hattori *et al.*, 1987] A. Hattori, H. Masuzawa, and H. Hayashi. AI machine. *Fujitsu Scientific and Technical Journal*, 23(4):369–378, 1987.

[Haug, 1999] Scott Haug. Automatic storage optimization via garbage collection. Master's thesis, Washington University, St Louis, 1999.

[Hauswirth and Chilimbi, 2004] Mattias Hauswirth and Trishul M. Chilimbi. Low-overhead memory leak detection using adaptive statistical profiling. In Mukherjee and McKinley [Mukherjee and McKinley2004], pages 156–164.

[Havelund and Shankar, 1997] Klaus Havelund and N. Shankar. A mechanized refinement proof for a garbage collector. Technical report, Aalborg University, 1997. Submitted to Formal Aspects of Computing.

[Havelund, 1999] Klaus Havelund. Mechanical verification of a garbage collector. In *FMPPTA'99*, 1999.

[Hawblitzel and Petrank, 2009] Chris Hawblitzel and Erez Petrank. Automated verification of practical garbage collectors. In *Conference Record of the Thirty-Sixth AnnualACM Symposium on Principles of Programming Languages*, pages 441–453, Savannah, GA, 2009.

[Hawblitzel *et al.*, 2004] Chris Hawblitzel, Edward Wei, Heng Huang, Eric Krupski, and Lea Wittie. Low-level linear memory management. In SPACE 2004 [SPACE 20042004].

[Hawblitzel *et al.*, 2007] Chris Hawblitzel, Heng Huang, Lea Wittie, and Juan Chen. A garbage-collecting typed assembly language. In *ACM SIGPLAN Workshop on Types in Language Design and Implementation*, Nice, France, January 2007.

[Hawblitzel, 2006] Chris Hawblitzel. Linear types for aliased resources. In SPACE 2006 [SPACE 20062006], pages 105–107.

[Hayashi *et al.*,] H. Hayashi, A. Hattori, and H. Akimoto. ALPHA: High-performance Lisp machine equipped with a new stack structure and real-time garbage collection system. Draft report, Fujitsu Laboratories.

[Hayes, 1990] Barry Hayes. Open systems require conservative garbage collectors. In Jul and Juul [Jul and Juul1990].

[Hayes, 1991] Barry Hayes. Using key object opportunism to collect old objects. In OOPSLA 1991 [OOPSLA 19911991], pages 33–46.

[Hayes, 1992] Barry Hayes. Finalization in the collector interface. In Bekkers and Cohen [Bekkers and Cohen1992].

[Hayes, 1993] Barry Hayes. *Key Objects in Garbage Collection*. PhD thesis, Stanford University, March 1993.

[Hayes, 1997] Barry Hayes. Ephemerons: A new finalization mechanism. In OOPSLA 1997 [OOPSLA 19971997], pages 176–183.

[Hazelwood *et al.*, 2009] Kim Hazelwood, Greg Lueck, and Robert Cohn. Scalable support for multithreaded applications on dynamic binary instrumentation systems. In Kolodner and Steele [Kolodner and Steele2009], pages 20–29.

[Heck and Wise, 1992] Brian C. Heck and David S. Wise. An implementation of an applicative file system. In Bekkers and Cohen [Bekkers and Cohen1992], pages 248–263.

[Hederman, 1988] Lucy Hederman. Compile-time garbage collection using reference count analysis. Master's thesis, Rice University, August 1988. Also Rice University Technical Report TR88–75 but, according to Rice University's technical report list, this report is no longer available for distribution.

[Heeb and Pfister, 1991] Beat Heeb and Cuno Pfister. Oberon technical notes: An integrated heap allocator/garbage collector. ETH Technical Report 156, ETH Eidgenössische Technische Hochschule Zürich, March 1991.

[Heil and Smith, 2000] Timothy Heil and James E. Smith. Concurrent garbage collection using hardware assisted profiling. In Chambers and Hosking [Chambers and Hosking2000], pages 80–93.

[Heintze and Tardieu, 2001a] Nevin Heintze and Olivier Tardieu. Demand-driven pointer analysis. In PLDI 2001 [PLDI 20012001], pages 24–34.

[Heintze and Tardieu, 2001b] Nevin Heintze and Olivier Tardieu. Ultra-fast aliasing analysis using CLA: A million lines of C code in a second. In PLDI 2001 [PLDI 20012001], pages 254–263.

[Hellyer *et al.*, 2010] Laurence Hellyer, Richard Jones, and Antony L. Hosking. The locality of concurrent write barriers. In jan Vitek and Lea [jan Vitek and Lea2010], pages 83–92.

[Helsen, 2001] Simon Helsen. Syntactic type soundness for the imperative region calculus. In SPACE 2001 [SPACE 20012001].

[Henderson *et al.*, 1995] Fergus Henderson, Thomas Conway, and Zoltan Somogyi. Compiling logic programs to C using GNU C as a portable assembler. In *ILPS'95 Post-conference Workshop on Sequential Implementation Technologies for Logic Programming*, pages 1–15, Portland, Or, 1995.

[Henderson, 2002] Fergus Henderson. Accurate garbage collection in an uncooperative environment. In Boehm and Detlefs [Boehm and Detlefs2002], pages 150–156.

[Hendren, 1990] Laurie J. Hendren. *Parallelizing Programs with Recursive Data Structures*. PhD thesis, Cornell University, April 1990.

[Heng, 1988] Seng-Lai Heng. Performance evaluation of numerous garbage collections by real-time simulation. Master's thesis, University of Texasat Austin, 1988.

[Henglein *et al.*, 2001] Fritz Henglein, Henning Makholm, and H. Niss. A direct approach to control-flow sensitive region-based memory management. In *International Conference on Principles and Practice of Declarative Programming (PPDP)*, pages 175–186, 2001.

[Hennessey, 1993] Wade Hennessey. Real-time garbage collection in a multimedia programming language. In Moss et al. [Moss *et al.*1993].

[Hennessy and Patterson, 1996] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufman, second edition, 1996.

[Henning, 1998] Michi Henning. Binding, migration and scalability in CORBA. *Communications of the ACM*, 41(10):62–71, October 1998.

[Henriksson, 1994] Roger Henriksson. Scheduling real-time garbage collection. In *Proceedings of NWPER'94*, Lund, Sweden, 1994.

[Henriksson, 1996a] Roger Henriksson. Adaptive scheduling of incremental copying garbage collection for interactive applications. Technical Report 96–174, Lund University, Sweden, 1996.

[Henriksson, 1996b] Roger Henriksson. Scheduling real-time garbage collection. Licentiate thesis, Department of Computer Science, Lund University, 1996. Lund technical report LU-CS-TR:96-161.

[Henriksson, 1997] Roger Henriksson. Predictable automatic memory management for embedded systems. In Dickman and Wilson [Dickman and Wilson1997].

[Henriksson, 1998] Roger Henriksson. *Scheduling Garbage Collection in Embedded Systems*. PhD thesis, Lund Institute of Technology, July 1998.

[Henzinger and Kirsch, 2001] T.A. Henzinger and C.M. Kirsch, editors. *First International Workshop on Embedded Software (EMSOFT)*, volume 2211 of *Lecture Notes in Computer Science*, Tahoe City, CA, 2001. Springer.

[Herlihy and Moss, 1990] Maurice Herlihy and J. Eliot B Moss. Non-blocking garbage collection for multiprocessors. Technical Report CRL 90/9, DEC Cambridge Research Laboratory, 1990.

[Herlihy and Moss, 1991] Maurice P. Herlihy and J. Eliot B. Moss. Lock-free garbage collection for multiprocessors. In *Parallel Algorithms and Architectures*, pages 229–236. ACM Press, July 1991.

[Herlihy and Moss, 1992] Maurice Herlihy and J. Eliot B Moss. Lock-free garbage collection for multiprocessors. *IEEE Transactions on Parallel and Distributed Systems*, 3(3), May 1992.

[Herlihy and Moss, 1993] Maurice P. Herlihy and J. Eliot B. Moss. Transactional memory: Architectural support for lock-free data structures. In ISCA 1993 [ISCA 19931993], pages 289–300.

[Herlihy and Shavit, 2008] Maurice Herlihy and Nir Shavit. *The Art of Multiprocessor Programming*. Morgan Kaufman, April 2008.

[Herlihy and Wing, 1990] Maurice Herlihy and Jeannette M. Wing. Linearizability: A correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems*, 12(3):463–492, 1990.

[Herlihy *et al.*, 2002a] Maurice P. Herlihy, Victor Luchangco, and Mark Moir. The repeat offender problem: A mechanism for supporting dynamic-sized lock-free data structures. In *Proceedings of the 16th International Symposium on Distributed Computing*, pages 339–353, October 2002.

[Herlihy *et al.*, 2002b] Maurice P. Herlihy, Paul Martin, Victor Luchangco, and Mark Moir. Dynamic-sized and lock-free data structures. Technical Report TR–2002–110, Sun Microsystems Laboratories, June 2002.

[Hertz and Berger, 2004] Matthew Hertz and Emery Berger. Automatic vs. explicit memory management: Settling the performance debate. Technical Report CS TR-04-17, University of Massachusetts, 2004.

[Hertz and Berger, 2005] Matthew Hertz and Emery Berger. Quantifying the performance of garbage collection vs. explicit memory management. In OOPSLA 2005 [OOPSLA 20052005].

[Hertz *et al.*, 2002a] Matthew Hertz, Steve M. Blackburn, K. S. McKinley, J. Eliot B. Moss, and Darko Stefanović. Error-free garbage collection traces: How to cheat and not get caught. In *Proceedings of the International Conference on Measurements and Modeling of Computer Systems*, Marina Del Rey, CA, June 2002.

[Hertz *et al.*, 2002b] Matthew Hertz, N. Immerman, and J. Eliot B. Moss. Framework for analyzing garbage collection. In *2nd IFIP Theoretical Computer Science Congress*, pages 230–241. Kluwer, 2002.

[Hertz *et al.*, 2004] Matthew Hertz, Yi Feng, and Emery Berger. Page-level cooperative garbage collection. Technical Report CS TR-04-16, University of Massachusetts, 2004.

[Hertz *et al.*, 2005] Matthew Hertz, Yi Feng, and Emery D. Berger. Garbage collection without paging. In Sarkar and Hall [Sarkar and Hall2005], pages 143–153.

[Hertz *et al.*, 2006] Matthew Hertz, Stephen M. Blackburn, J. Eliot B. Moss, Kathryn McKinley, and Darko Stefanović. Generating object lifetime traces with Merlin. *ACM Transactions on Programming Languages and Systems*, 28(3):476–516, May 2006.

[Hertz *et al.*, 2009] Matthew Hertz, Jonathan Bard, Stephen Kane, Elizabeth Keudel, Tongxin Bai, Kirk Kelsey, and Chen Ding. Waste not, want not — resource-based garbage collection in a shared environment. Technical Report TR–951, The University of Rochester, December 2009.

[Hertz, 2006] Matthew Hertz. *Quantifying and Improving the Performance of Garbage Collection*. PhD thesis, University of Massachusetts, September 2006.

[Hesselink and Groote, 2001] Wim H. Hesselink and Jan Friso Groote. Wait-free concurrent memory management by create and read until deletion (CaRuD). *Distributed Computing*, 14(1):31–39, 2001.

[Hewitt, 1977] Carl Hewitt. Viewing control structures as patterns of passing messages. *Journal of Artificial Intelligence*, 8(3):323–364, June 1977.

[Heymann, 1991] J. Heymann. A comprehensive analytical model for garbage collection algorithms. *ACM SIGPLAN Notices*, 26(8):50–59, 1991.

[Hibino, 1980] Y. Hibino. A practical parallel garbage collection algorithm and its implementation. In *7th Annual Symposium on Computer Architecture*, pages 113–120, May 1980.

[Hickey and Cohen, 1984] Tim Hickey and Jacques Cohen. Performance analysis of on-the-fly garbage collection. *Communications of the ACM*, 27(11):1143–1154, November 1984.

[Hicks *et al.*, 1997] Michael W. Hicks, Jonathan T. Moore, and Scott M. Nettles. The measured cost of copying garbage collection mechanisms. In ICFP 1997 [ICFP 19971997], pages 292–305.

[Hicks *et al.*, 1998] Michael Hicks, Luke Hornof, Jonathan T. Moore, and Scott Nettles. A study of Large Object Spaces. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 138–145.

[Hicks *et al.*, 2004a] Michael Hicks, Dan Grossman, and Trevor Jim. Combining garbage collection and safe manual memory management. In SPACE 2004 [SPACE 20042004].

[Hicks *et al.*, 2004b] Michael Hicks, Greg Morrisett, Dan Grossman, and Trevor Jim. Experience with safe manual memory-management in Cyclone. In Bacon and Diwan [Bacon and Diwan2004], pages 73–84.

[Hicks, 1993] James Hicks. Experiences with compiler-directed storage reclamation. In Hughes [Hughes1993].

[Hieb *et al.*, 1990] R. Hieb, R. K. Dybvig, and C. Bruggeman. Representing control in the presence of first-class continuations. *ACM SIGPLAN Notices*, 25(6):66–77, 1990.

[Hiep, 1991] Van Nguyen Hiep. *Compilation et Environment d'Execution d'un Langage à Base d'Objects*. PhD thesis, Institut National Polytechnique de Grenoble, February 1991.

[Higuera and Issarny, 2005] M. Teresa Higuera and Valerie Issarny. Improving the memory management performance of RTSJ. *Concurrency and Computation: Practice and Experience*, 2005.

[Higuera *et al.*, 2002] M. Teresa Higuera, Valerie Issarny, Michel Banatre, Gilbert Cabillic, Jean-Philippe Lesot, and Frederic Parain. Memory management for real-time Java: an efficient solution using hardware support. *Real-Time Systems Journal*, 2002.

[Higuera-Toledano and Issarny, 2001] M. Teresa Higuera-Toledano and Valerie Issarny. Analyzing the performance of memory management in RTSJ. In ISORC 2001 [ISORC 20012001].

[Higuera-Toledano *et al.*, 2004] M. Teresa Higuera-Toledano, Valerie Issarny, Michel Banatre, Gilbert Cabillic, Jean-Philippe Lesot, and Frederic Parain. Memory management for Real-Time Java: an efficient solution using hardware support. *Real-Time Systems Journal*, 26(1):63–87, 2004.

[Higuera-Toledano, 2006a] M. Teresa Higuera-Toledano. Analyzing the memory management semantic and requirements of the Real-Time specification of Java. In ISORC 2006 [ISORC 20062006], pages 419–423.

[Higuera-Toledano, 2006b] M. Teresa Higuera-Toledano. Improving the scoped memory region garbage collector of real-time Java. In Olivier Zendra, editor, *Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS'2006)*, pages 1–14, Nantes, France, July 2006.

[Higuera-Toledano, 2006c] M. Teresa Higuera-Toledano. The indeterministic behaviour of scoped memory in Real-Time Java. In *4th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-06)*, pages 656–664. IEEE Press, 2006.

[Higuera-Toledano, 2006d] M. Teresa Higuera-Toledano. Towards an analysis of garbage collection techniques for embedded real-time Java systems. In *12th international Conference on Embedded and Real-Time Computing Systems and Applications*, pages 97–100. IEEE Press, 2006.

[Higuera-Toledano, 2007a] M. Teresa Higuera-Toledano. Allowing cycle's references among scoped memory areas in the Real-Time Specification for Java. In ISORC 2006 [ISORC 20062007], pages 110–114.

[Higuera-Toledano, 2007b] M. Teresa Higuera-Toledano. Name-based write barriers in real-time Java. In *IEEE international Conference on Computer and Information Technology (CIT-07)*, pages 781–786. IEEE Press, 2007.

[Higuera-Toledano, 2008] M. Teresa Higuera-Toledano. Allowing cycle references by introducing controlled violations of the assignment rules in real-time Java. In ISORC 2008 [ISORC 20082008], pages 463–467.

[Higuera, 2003] M. Teresa Higuera. Memory management design to the concurrent execution of RTSJ applications. In JTRES 2003 [JTRES 20032003], pages 479–489.

[Hill and Smith, 1989] Mark D. Hill and Alan Jay Smith. Evaluating associativity in CPU caches. *IEEE Transactions on Computers*, 38(12):1612–1629, December 1989.

[Hill, 1987] Mark D. Hill. *Aspects of Cache Memory and Instruction Buffer Performance*. PhD thesis, University of California, Berkeley, November 1987. Also UCB/CSD Technical report 87/381.

[Hill, 1988] Mark D. Hill. A case for direct-mapped caches. *IEEE Computer*, 21(12):25–40, December 1988.

[Hind and Vitek, 2005] Michael Hind and Jan Vitek, editors. *Proceedings of the First ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, Chicago, IL, USA, June 2005.

[Hind, 2001] Michael Hind. Pointer analysis: Haven't we solved this problem yet? In *2001 ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering (PASTE'01)*, Snowbird, UT, June 2001.

[Hinds, 1975] J. A. Hinds. An algorithm for locating adjacent storage blocks in the buddy system. *Communications of the ACM*, 18(4):221–222, April 1975.

[Hirschberg, 1973] D. S. Hirschberg. A class of dynamic memory allocation algorithms. *Communications of the ACM*, 16(10):615–618, October 1973.

[Hirzel and Diwan, 2000] Martin Hirzel and Amer Diwan. On the type accuracy of garbage collection. In Chambers and Hosking [Chambers and Hosking2000], pages 1–11.

[Hirzel *et al.*, 2001] Martin Hirzel, Amer Diwan, and Antony L. Hosking. On the usefulness of liveness for garbage collection and leak detection. In Knudsen [Knudsen2001].

[Hirzel *et al.*, 2002a] Martin Hirzel, Amer Diwan, and Johannes Henkel. On the usefulness of type and liveness for garbage collection and leak detection. *ACM Transactions on Programming Languages and Systems*, 24(6):593–624, November 2002.

[Hirzel *et al.*, 2002b] Martin Hirzel, Johannes Henkel, Amer Diwan, and Michael Hind. Understanding the connectivity of heap objects. In Boehm and Detlefs [Boehm and Detlefs2002], pages 36–49.

[Hirzel *et al.*, 2003a] Martin Hirzel, Amer Diwan, and Matthew Hertz. Connectivity-based garbage collection. In OOPSLA 2003 [OOPSLA 20032003].

[Hirzel *et al.*, 2003b] Martin Hirzel, Harold N. Gabow, and Amer Diwan. Choosing a set of partitions to collect in a connectivity-based garbage collector. Technical Report CU–CS–958–03, University of Colorado, August 2003.

[Hirzel *et al.*, 2004] Martin Hirzel, Amer Diwan, and Michael Hind. Pointer analysis in the presence of dynamic class loading. In ECOOP 2004 [ECOOP 20042004].

[Hirzel *et al.*, 2007] Martin Hirzel, Daniel von Dincklage, Amer Diwan, , and Michael Hind. Fast online pointer analysis. *ACM Transactions on Programming Languages and Systems*, April 2007.

[Hirzel, 2000] Martin Hirzel. Effectiveness of garbage collection and explicit deallocation. Master's thesis, University of Colorado, 2000.

[Hirzel, 2004] Martin Hirzel. *Connectivity-Based Garbage Collection*. PhD thesis, University of Colorado at Boulder, July 2004.

[Hirzel, 2007] Martin Hirzel. Data layouts for object-oriented programs. In *International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2007. This paper plus a 2-page appendix is available as IBM Research Report RC24218, Watson.

[Hoare, 1969] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12, October 1969.

[Hoare, 1974] C. A. R. Hoare. Optimisation of store size for garbage collection. *Information Processing Letters*, 2(6):165–166, April 1974.

[Hoare, 2009] C. A. R. Hoare. Null references: The billion dollar mistake. In *Proceedings of QCon*, Historically Bad Ideas, London, UK, March 2009.

[Hofman and Jost, 2003] Martin Hofman and Steffen Jost. Static prediction of heap usage for first-order functional programs. In POPL 2003 [POPL 20032003].

[Hofmann, 2001] Martin Hofmann. A type system for controlling heap space and its translation to JavaCard. In SPACE 2001 [SPACE 20012001].

[Hogen and Loogen, 1993] Guido Hogen and Rita Loogen. A new stack technique for the management of runtime structures in distributed implementations. Aachener Informatik-Berichte 93-3, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Germany, 1993.

[Hogen and Loogen, 1994a] Guido Hogen and Rita Loogen. Efficient organization of control structures in distributed implementations. In Peter A. Fritzson, editor, *Compiler Construction*, volume 786 of *Lecture Notes in Computer Science*, pages 98–112. Springer-Verlag, 1994.

[Hogen and Loogen, 1994b] Guido Hogen and Rita Loogen. Parallel functional implementations: Graphbased vs. stackbased reductions. Technical report, RWTH Aachen, ghogen@zesu.informatik.rwth-aachen.de, lllgen@informatik.uni-marburg.de, 1994.

[Holloway *et al.*, 1980] Jack Holloway, Guy L. Steele, Gerald Jay Sussman, and Alan Bell. The SCHEME–79 chip. AI Memo 559, MIT AI Laboratory, January 1980.

[Holmström, 1983] S. Holmström. A simple and efficient way to handle large datastructures in applicative languages. In *Joint SERC/Chalmers Workshop on Declarative Programming, University College, London*, 1983.

[Holt, 1961] Anatol W. Holt. Program organization and record keeping for dynamic storage allocaton. *Communications of the ACM*, 4(10), October 1961.

[Hölzle and Ungar, 1995] Urs Hölzle and David Ungar. Do object-oriented languages need special hardware support? In Nierstras [Nierstras1995].

[Hölzle, 1991] Urs Hölzle. The myth of high object creation rates. In Wilson and Hayes [Wilson and Hayes1991a].

[Hölzle, 1993] Urs Hölzle. A fast write barrier for generational garbage collectors. In Moss et al. [Moss *et al.*1993].

[Horowitz and Sahni, 1977] E. Horowitz and S. Sahni. *Fundamentals of Data Structures*. Computer Science Press, Woodland Hills, CA, 1977.

[Horspool and Huberman, 1987] R. Nigel Horspool and Ronald M. Huberman. Analysis and development of demand prepaging policies. *Journal of Systems and Software*, 7:183–194, 1987.

[Horwitz *et al.*, 1989] Susan Horwitz, Phil Pfeiffer, and Thomas W. Reps. Dependence analysis for pointer variables. In PLDI 1989 [PLDI 19891989], pages 28–40.

[Hosking and Chen, 1999] Antony L. Hosking and Jiawan Chen. Mostly-copying reachability-based orthogonal persistence. In OOPSLA 1999 [OOPSLA 19991999], pages 382–398.

[Hosking and Hudson, 1993] Antony L. Hosking and Richard L. Hudson. Remembered sets can also play cards. In Moss et al. [Moss *et al.*1993].

[Hosking and Moss, 1993a] Antony L. Hosking and J. Eliot B. Moss. Object fault handling for persistent programming languages: A performance evaluation. In OOPSLA 1993 [OOPSLA 19931993].

[Hosking and Moss, 1993b] Antony L. Hosking and J. Eliot B. Moss. Protection traps and alternatives for memory management of an object-oriented language. In SOSP 1993 [SOSP 19931993], pages 106–119.

[Hosking and Moss, 1995] Antony L. Hosking and J. Eliot B. Moss. Lightweight write detection and checkpointing for fine-grained persistence. Technical Report 95-084, Purdue University, 1995.

[Hosking and Novianto, 1997] Antony L. Hosking and Aria P. Novianto. Reachability-based orthogonal persistence for C, C++ and other intransigents. In Dickman and Wilson [Dickman and Wilson1997].

[Hosking *et al.*, 1992] Antony L. Hosking, J. Eliot B. Moss, and Darko Stefanović. A comparative performance evaluation of write barrier implementations. In OOPSLA 1992 [OOPSLA 19921992], pages 92–109.

[Hosking *et al.*, 1999] Antony L. Hosking, Nathaniel Nystrom, Quintin Cutts, and Kumar Brahnmath. Optimizing the read and write barrier for orthogonal persistence. In Morrison et al. [Morrison *et al.*1999].

[Hosking *et al.*, 2009] Antony L. Hosking, David Bacon, and Orran Krieger, editors. *Proceedings of the Fifth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, Washington, DC, USA, March 2009.

[Hosking, 1991] Antony L. Hosking. Main memory management for persistence. In Wilson and Hayes [Wilson and Hayes1991a].

[Hosking, 2006] Antony L Hosking. Portable, mostly-concurrent, mostly-copying garbage collection for multi-processors. In Petrank and Moss [Petrank and Moss2006], pages 40–51.

[Hosoya and Yonezawa, 1998] Haruo Hosoya and Akinori Yonezawa. Garbage collection via dynamic type inference — a formal treatment. In *Proceedings of the Second Workshop on Types in Compilation*, pages 215–239, 1998.

[Hu *et al.*, 2003] Y. Charlie Hu, Weimin Yu, Alan Cox, Dan Wallach, and Willy Zwaenepoel. Runtime support for distributed sharing in safe languages. *ACM Transactions on Computer Systems*, 21:1–35, February 2003.

[Huang *et al.*, 2003] Xianlong Huang, J. Eliot B. Moss, Kathryn S. McKinley, Stephen M. Blackburn, and D. Burger. Dynamic SimpleScalar: Simulating Java virtual machines. Technical Report TR–03–03, University of Texas at Austin, February 2003.

[Huang *et al.*, 2004a] Wei Huang, Y. Qian, Witiwas Srisa-an, and J. Morris Chang. Object allocation and memory contention study of Java multithreaded application. In *Proceedings of IEEE International Performance Computing and Communications Conference (IPCCC 2004)*, Phoenix, AZ, April 2004.

[Huang *et al.*, 2004b] Wei Huang, Witiwas Srisa-an, and J. Morris Chang. Adaptive pretenuring for generational garbage collection. In *Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS-04)*, pages 133–140, Austin, TX, March 2004.

[Huang *et al.*, 2004c] Xianlong Huang, Stephen M. Blackburn, Kathryn S. McKinley, J. Eliot B. Moss, Z. Wang, and Perry Cheng. The garbage collection advantage: Improving program locality. In OOPSLA 2004 [OOPSLA 20042004], pages 69–80.

[Huang *et al.*, 2006] Xianglong Huang, Stephen M Blackburn, David Grove, and Kathryn S McKinley. Fast and efficient partial code reordering: Taking advantage of dynamic recompilation. In Petrank and Moss [Petrank and Moss2006], pages 184–192.

[Hudak and Bloss, 1985] Paul Hudak and Adrienne Bloss. The aggregate update problem in functional programming systems. In POPL 1984 [POPL 19841984], pages 300–314.

[Hudak and Keller, 1982] Paul R. Hudak and R. M. Keller. Garbage collection and task deletion in distributed applicative processing systems. In LFP 1982 [LFP 19821982], pages 168–178.

[Hudak *et al.*, 1992] Paul Hudak, Simon L. Peyton Jones, and Phillip Wadler. Report on the programming language Haskell, a non-strict purely functional language (version 1.2). *ACM SIGPLAN Notices*, 27(5), May 1992.

[Hudak, 1981] Paul R. Hudak. Call-graph reclamation: an alternative storage reclamation scheme. AMPS Technical Memorandum 4, University of Utah, August 1981.

[Hudak, 1982] Paul R. Hudak. *Object and Task Reclamation in Distributed Applicative Processing Systems*. PhD thesis, University of Utah, Salt Lake City, Utah, 1982.

[Hudak, 1983a] Paul R. Hudak. Distributed graph marking. Departmental Research Report 268, University of Yale, 1983.

[Hudak, 1983b] Paul R. Hudak. Distributed task and memory management. In *ACM Symposioum on the Principles of Distributed Computing*, pages 277–89. ACM Press, August 1983.

[Hudak, 1986] Paul R. Hudak. A semantic model of reference counting and its abstraction (detailed summary). In LFP 1986 [LFP 19861986], pages 351–363.

[Hudak, 1987] Paul R. Hudak. A semantic model of reference counting and its abstraction. In Samson Abramsky and Chris Hankin, editors, *Abstract Interpretation of Declarative Languages*, pages 45–62. Ellis Horward, 1987.

[Hudson and Diwan, 1990] Richard L. Hudson and Amer Diwan. Adaptive garbage collection for Modula-3 and Smalltalk. In Jul and Juul [Jul and Juul1990].

[Hudson and Hannah, 1992] S. Hudson and J. M. Hannah. Structured knowledge manipulation system for real-time engineering applications. *IEE Proceedings, Part E: Computers and Digital Techniques*, 139(1):59–63, January 1992.

[Hudson and Moss, 1992] Richard L. Hudson and J. Eliot B. Moss. Incremental garbage collection for mature objects. In Bekkers and Cohen [Bekkers and Cohen1992].

[Hudson and Moss, 2001] Richard L. Hudson and J. Eliot B. Moss. Sapphire: Copying GC without stopping the world. In *Joint ACM Java Grande — ISCOPE 2001 Conference*, Stanford University, CA, June 2001.

[Hudson and Moss, 2003] Richard L. Hudson and J. Eliot B. Moss. Sapphire: Copying garbage collection without stopping the world. *Concurrency and Computation: Practice and Experience*, 15(3–5):223–261, 2003.

[Hudson *et al.*, 1991] Richard L. Hudson, J. Eliot B. Moss, Amer Diwan, and Christopher F. Weight. A language-independent garbage collector toolkit. Technical Report COINS 91-47, University of Massachusetts at Amherst, Department of Computer and Information Science, September 1991.

[Hudson *et al.*, 1997a] Richard L. Hudson, Ron Morrison, J. Eliot B. Moss, and David S. Munro. Garbage collecting the world: One car at a time. In OOPSLA 1997 [OOPSLA 19971997].

[Hudson *et al.*, 1997b] Richard L. Hudson, Ron Morrison, J. Eliot B. Moss, and David S. Munro. Training distributed garbage: The DMOS collector. Technical report, University of St Andrews, 1997.

[Hudson *et al.*, 1998] Richard L. Hudson, Ron Morrison, J. Eliot B. Moss, and David S. Munro. Where have all the pointers gone? In *Proceedings of 21st Australasian Computer Science Conference*, pages 107–119, Perth, 1998.

[Hudson *et al.*, 2000] Richard L. Hudson, J. Eliot B. Moss, Sreenivas Subramoney, and Weldon Washburn. Cycles to recycle: Garbage collection on the IA-64. In Chambers and Hosking [Chambers and Hosking2000], pages 101–110.

[Hudson *et al.*, 2006] Richard L. Hudson, Bratin Saha, Ali-Reza Adl-Tabatabai, and Benjamin Hertzberg. McRT-malloc - a scalable transactional memory allocator. In Petrank and Moss [Petrank and Moss2006], pages 74–83.

[Hudson, 1991] Richard L. Hudson. Finalization in a garbage collected world. In Wilson and Hayes [Wilson and Hayes1991a].

[Huelsbergen and Larus, 1993] Lorenz Huelsbergen and James R. Larus. A concurrent copying garbage collector for languages that distinguish (im)mutable data. In PPOPP 1993 [PPOPP 19931993], pages 73–82.

[Huelsbergen and Winterbottom, 1998] Lorenz Huelsbergen and Phil Winterbottom. Very concurrent mark-&-sweep garbage collection without fine-grain synchronization. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 166–175.

[Hughes, 1982] R. John M. Hughes. A semi-incremental garbage collection algorithm. *Software Practice and Experience*, 12(11):1081–1084, November 1982.

[Hughes, 1983] R. John M. Hughes. Reference counting with circular structures in virtual memory applicative systems. Internal paper, Programming Research Group, Oxford, 1983.

[Hughes, 1985] R. John M. Hughes. A distributed garbage collection algorithm. In Jouannaud [Jouannaud1985], pages 256–272.

[Hughes, 1987] R. John M. Hughes. Managing reduction graphs with reference counts. Departmental Research Report CSC/87/R2, University of Glasgow, March 1987.

[Hughes, 1991a] R. John M. Hughes, editor. *Record of the 1991 Conference on Functional Programming and Computer Architecture*, volume 523 of *Lecture Notes in Computer Science*, Cambridge, MA, August 1991. Springer-Verlag.

[Hughes, 1991b] Simon Hughes. *Static Analysis of Store Use in Functional Programs*. PhD thesis, Imperial College, University of London, 1991.

[Hughes, 1992] Simon Hughes. Compile-time garbage collection for higher-order functional languages. *Journal of Logic and Computation*, 2(4):483–509, August 1992. Special Issue on Abstract Interpretation.

[Hughes, 1993] R. John M. Hughes, editor. *Record of the 1993 Conference on Functional Programming and Computer Architecture*, Copenhagen, June 1993. ACM Press.

[Humphries *et al.*, 1997] Thorna O. Humphries, Alexander L. Wolf, and Benjamin G. Zorn. A framework for storage management evaluation in persistent object systems. In Dickman and Wilson [Dickman and Wilson1997].

[Humphries *et al.*, 2000] Thorna O. Humphries, Artur W. Klauser, Alexander L. Wolf, and Benjamin G. Zorn. POSSE trace format, version 1.0. Technical Report CU-CS-897-00, University of Colorado, Boulder, CO, January 2000.

[Hunter and Krishnamurthi, 2003] Rob Hunter and Shriram Krishnamurthi. A model of garbage collection for OO languages. In *Tenth International Workshop on Foundations of Object-Oriented Languages (FOOL10)*, 2003.

[Hutchinson *et al.*, 1987] Norman Hutchinson, R. K. Raj, Andrew P. Black, Henry M. Levy, and Eric Jul. The Emerald programming language report. Technical Report 87–10–07, University of Washington, October 1987.

[Hutchinson, 1987] Norman Hutchinson. *Emerald: An Object-Based Language for Distributed Programming*. PhD thesis, University of Washington, January 1987.

[IBM Corp., 2003] JVM garbage collection and storage allocation techniques. Technical report, IBM Corp., November 2003. Details of the IBM Developer Kit and Runtime Environment, version 1.4.1.

[ICFP 1996, 1996] *Proceedings of First International Conference on Functional Programming*, Philadelphia, PA, May 1996. ACM Press.

[ICFP 1997, 1997] *Proceedings of Second International Conference on Functional Programming*, Amsterdam, June 1997. ACM Press.

[ICFP 1998, 1998] *Proceedings of International Conference on Functional Programming*, Baltimore, MA, September 1998. ACM Press.

[ICFP 1999, 1999] *Proceedings of International Conference on Functional Programming*, Paris, September 1999. ACM Press.

[ICFP 2000, 2000] *Proceedings of International Conference on Functional Programming*, Montreal, September 2000. ACM Press.

[ICFP 2010, 2010] *Proceedings of International Conference on Functional Programming*, Baltimore, MD, USA, September 2010. ACM Press.

[Ichisuki and Morita, 1994] Yuuji Ichisuki and Masao Morita. A shared-memory parallel extension of KLIC and its garbage collection. In *Proceedings of FGCS'94 Workshop on Parallel Logic Programming*, pages 113–126, 1994.

[Ichisuki and Yonezawa, 1990a] Yuuji Ichisuki and Akinori Yonezawa. Distributed garbage collection using group reference counting. In Jul and Juul [Jul and Juul1990].

[Ichisuki and Yonezawa, 1990b] Yuuji Ichisuki and Akinori Yonezawa. Distributed garbage collection using group reference counting. Technical Report 90–014, University of Tokyo, 1990.

[ICLP 1982, 1982] *Proceedings of First International Conference on Logic Programming*, 1982.

[ICLP 1988, 1988] *Proceedings of Fifth International Conference on Logic Programming*, 1988.

[ICSE 2004, 2004] *Proceedings of the 26th International Conference on Software Engineering*, Edinburgh, May 2004.

[Iliadis, 2010] Ilias Iliadis. Performance of the greedy garbage-collection scheme in flash- based solid-state drives. Research Report RZ 3769 (# 99779), IBM Research -: Zurich, March 2010.

[Iliffe and Jodeit, 1962] J. K. Iliffe and J. G. Jodeit. A dynamic storage allocation scheme. *Computer Journal*, 5(3):200–209, October 1962.

[Ilsøe and Pedersen, 2001] Peer Møller Ilsøe and Simon Hem Pedersen. Garbage collection in a Beta virtual machine with the Train Algorithm. Master's thesis, Aalborg University, June 2001.

[Imai and Tick, 1991a] Akira Imai and Evan Tick. Evaluation of parallel copying garbage collection on a shared-memory multiprocessor. ICOT technical report TR-650, Institute for New Generation Computer Technology, May 1991.

[Imai and Tick, 1991b] Akira Imai and Evan Tick. A shared-memory multiprocess or [sic] garbage collector and its evaluation for committed-choice logic programs. ICOT technical report TR-653, Institute for New Generation Computer Technology, June 1991.

[Imai and Tick, 1993] Akira Imai and Evan Tick. Evaluation of parallel copying garbage collection on a shared-memory multiprocessor. *Transactions on Parallel and Distributed Systems*, 4(9):1030–1040, 1993.

[Imai *et al.*, 1990] Akira Imai, Keiji Hirata, and Kazuo Taki. A parallel copying garbage collection for kl1 on a shared memory multiprocessor. ICOT technical memorandum TM-0967, Institute for New Generation Computer Technology, November 1990.

[inmos, 1988] inmos. *The occam 2 Reference Manual*. Prentice-Hall, 1988.

[Inoue and Torii, 1991] Katsuro Inoue and K. Torii. Implementation and analysis of compile-time garbage collection. *New Generation Computing*, 10(1):101–119, 1991.

[Inoue *et al.*, 1988] Katsuro Inoue, Hiroyuki Seki, and Hikaru Yagi. Analysis of functional programs to detect run-time garbage cells. *ACM Transactions on Programming Languages and Systems*, 10(4):555–578, October 1988.

[Inoue *et al.*, 2003] H. Inoue, Darko Stefanović, and S. Forrest. Object lifetime prediction in Java. Technical Report TR–CS–2003–28, University of New Mexico, May 2003.

[Inoue *et al.*, 2009] Hiroshi Inoue, Hideaki Komatsu, and Toshio Nakatani. A study of memory management for web-based applications on multicore processors. In PLDI 2009 [PLDI 20092009].

[Iosif and Sisto, 2000] R. Iosif and R. Sisto. Using garbage collection in model checking. In *Proceedings of 7th International SPIN Workshop on Model Checking of Software*, volume 1885 of *Lecture Notes in Computer Science*, pages 20–33, Stanford, CA, September 2000.

[Ireland, 1989] E. Ireland. Writing interactive and file-processing functional programs. Master's thesis, Victoria University of Wellington, March 1989.

[ISCA 1985, 1985] *12th Annual International Symposium on Computer Architecture*. IEEE Press, June 1985.

[ISCA 1990, 1990] *17th Annual International Symposium on Computer Architecture*, Seattle, Washington, May 1990. IEEE Press.

[ISCA 1991, 1991] *18th Annual International Symposium on Computer Architecture*, Toronto, Canada, May 1991. ACM Press.

[ISCA 1992, 1992] *19th Annual International Symposium on Computer Architecture*, Gold Coast, Australia, May 1992. ACM Press.

[ISCA 1993, 1993] *20th Annual International Symposium on Computer Architecture*, San Diego, CA, May 1993. IEEE Press.

[ISCA 1994, 1994] *21st Annual International Symposium on Computer Architecture*, Chicago, Illinois, April 1994. IEEE Press.

[ISCA 2009, 2009] *36th AnnualInternational Symposium on Computer Architecture*, Austin, Texas, June 2009.

[Ishii, 1983] Carol Y. Ishii. A comparative study of garbage collection and compaction algorithms. Master's thesis, Emory University, 1983.

[Isoda *et al.*, 1971] S. Isoda, E. Goto, and I. Kimura. An efficient bit table technique for dynamic storage allocation of 2^n-word blocks. *Communications of the ACM*, 14(9):589–592, September 1971.

[ISORC 2001, 2001] *Proceedings of the 5th International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'01)*, Washington, DC, April 2001. IEEE Press.

[ISORC 2004, 2004] *Proceedings of the 7th International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'04)*, Vienna, May 2004. IEEE Press.

[ISORC 2006, 2006] *Proceedings of the 9th International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'06)*, Gyeongju, Korea, April 2006. IEEE Press.

[ISORC 2006, 2007] *Proceedings of the 10th International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'07)*. IEEE Press, 2007.

[ISORC 2008, 2008] *Proceedings of the 11th International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'08)*, Orlando, Florida, 2008. IEEE Press.

[Ito and Asai, 1997] Takayasu Ito and Toshihiro Asai. Timed-GC for a real-time Lisp system. In *Workshop on Languages, Compilers, and Tools for Real-Time Systems (LCT-RTS97)*, Las Vegas, Nevada, June 1997. ACM Press.

[IWOOOS 1993, 1993] *International Workshop on Object Orientation in Operating Systems*, 1993.

[Iyengar, 1992] Arun Iyengar. *Dynamic Storage Allocation on a Multiprocessor*. PhD thesis, MIT Laboratory for Computer Science, 1992. Technical Report MIT/LCS/TR-560.

[Iyengar, 1993] Arun K. Iyengar. Parallel dynamic storage allocation algorithms. In *Fifth IEEE Symposium on Parallel and Distributed Processing*. IEEE Press, 1993.

[Jackson, 1991] Frank Jackson. Garbage collection bugs that I have known. In Wilson and Hayes [Wilson and Hayes1991a].

[Jackson, 1998] Paul B. Jackson. Verifying a garbage collection algorithm. In *Proceedings of 11th International Conference on Theorem Proving in Higher Order Logics TPHOLs'98*, volume 1479 of *Lecture Notes in Computer Science*, pages 225–244, Canberra, September 1998. Springer-Verlag.

[Jacobs and Langen, 1989] D. Jacobs and A. Langen. Accurate and efficient approximation of variable aliasing in logic programs. In *North American Conference on Logic Programming*, pages 154–165, 1989.

[Jagadish *et al.*, 1994] H. V. Jagadish, Daniel Lieuwen, Rajeev Rastogi, and Avi Silbersch atz. Dali: A high performance main memory storage manager. In *20th International Conference on Very Large Data Bases*, 1994.

[jan Vitek and Lea, 2010] jan Vitek and Doug Lea, editors. *Proceedings of the Ninth International Symposium on Memory Management*, Toronto, Canada, June 2010. ACM Press.

[Janssens and Bruynooghe, 1992] G. Janssens and Maurice Bruynooghe. Deriving descriptions of possible values of program variables by means of abstract interpretation. *Journal of Logic Programming*, 13(2, 3):205–258, July 1992.

[Janssens, 1986] D. Janssens. Message passing and graph transformations: a model of Actor computation. *Microprocessing and Microprogramming*, 18(1–5):307–318, 1986.

[Janssens, 1990] G. Janssens. *Deriving Run-Time Properties of Logic Programs by Means of Abstract Interpretation*. PhD thesis, Katholieke University of Leuven, Belgium, 1990.

[Java RMI, 1996] Sun Microsystems. *Java Remote Method Invocation Specification*, November 1996.

[Jazayeri and Pozefsky, 1981] Medhi Jazayeri and Diane Pozefsky. Space-efficient storage management in an attribute grammar evaluator. *ACM Transactions on Programming Languages and Systems*, 3(4):388–404, October 1981.

[Jefferson and others, 1987] D. R. Jefferson et al. Distributed simulation and the Time Warp operating system. In *11th ACM Symposium on Operating Systems Principles*. ACM Press, November 1987.

[Jefferson, 1985] D. R. Jefferson. Virtual time. *ACM Transactions on Programming Languages and Systems*, 7(3):404–425, July 1985.

[Jensen and Mogensen, 1990] Thomas P. Jensen and Torben Mogensen. A backwards analysis for compile-time garbage collection. In Neil D. Jones, editor, *ESOP'90 3rd European Symposium on Programming, Copenhagen, Denmark, May 1990. (Lecture Notes in Computer Science, vol. 432)*, pages 227–239. Springer-Verlag, 1990.

[Jensen, 1990] Thomas P. Jensen. Context analysis of functional programs. Master's thesis, Department of Computer Science (DIKU), University of Copenhagen, January 1990.

[Jeschke, 1995] Eric R. Jeschke. *An Architecture for Parallel Symbolic Processing based on Suspending Construction*. PhD thesis, Indiana University, April 1995.

[Jew, 1987] Yanni K. Jew. Distributed garbage collection. Master's thesis, Carleton University, Canada, 1987.

[Jinsight,] *Jinsight*. Visualisation tools for Java.

[Joao *et al.*, 2009] José A. Joao, Onur Mutlu, and Yale N. Patt. Flexible reference-counting-based hardware acceleration for garbage collection. In ISCA 2009 [ISCA 20092009], pages 418–428.

[Johansson *et al.*, 2002] Erik Johansson, Konstantinos Sagonas, and Jesper Wilhelmsson. Memory architectures for concurrent lanugages using message passing. In Boehm and Detlefs [Boehm and Detlefs2002], pages 88–99.

[Johnson and Davis, 1992] Theodore Johnson and Tim Davis. Space efficient parallel buddy memory management, 1992.

[Johnson and Ha, 1994] Eric E. Johnson and Jiheng Ha. PDATS: Lossless address space compression for reducing file size and access time. In *Proceedings of 1994 IEEE International Phoenix Comference on Computers and Communication*, April 1994.

[Johnson and Sasha, 1992] Theodore Johnson and D. Sasha. Parallel buddy memory management. *Parallel Processing Letters*, 2(4):391–398, 1992.

[Johnson, 1985] Steven D. Johnson. Storage allocation for list processing. Technical Report 168, Indiana University, March 1985.

[Johnson, 1988] Douglas Johnson. Trap architectures for Lisp systems. Technical Report UCB/CSD/88/470, University of California, Berkeley, November 1988.

[Johnson, 1991a] Douglas Johnson. The case for a read barrier. *ACM SIGPLAN Notices*, 26(4):279–287, 1991.

[Johnson, 1991b] Douglas Johnson. Comparing two garbage collectors. In Wilson and Hayes [Wilson and Hayes1991a].

[Johnson, 1991c] Theodore Johnson. A concurrent fast fit memory manager. Technical Report 91-009, University of Florida, 1991.

[Johnson, 1992] Ralph E. Johnson. Reducing the latency of a real-time garbage collector. *Letters on Programming Languages and Systems*, 1(1):46–58, March 1992.

[Johnson, 1994] Eric E. Johnson. PDATS II: Improved compression of traces. In *1999 IEEE International Performance, Computing and Communications Conference*, February 1994.

[Johnsson, 1987] Thomas Johnsson. *Compiling Lazy Functional Languages*. PhD thesis, Chalmers University of Technology, 1987.

[Johnstone and Wilson, 1997] Mark S. Johnstone and Paul R. Wilson. The memory fragmentation problem: Solved? In Dickman and Wilson [Dickman and Wilson1997].

[Johnstone and Wilson, 1998] Mark S. Johnstone and Paul R. Wilson. The memory fragmentation problem: Solved? In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 26–36.

[Johnstone, 1997] Mark S. Johnstone. *Non-Compacting Memory Allocation and Real-Time Garbage Collection*. PhD thesis, University of Texas at Austin, December 1997.

[Joisha, 2006] Pramod Joisha. Compiler optimizations for nondeferred reference-counting garbage collection. In Petrank and Moss [Petrank and Moss2006], pages 150–161.

[Joisha, 2007] Pramod Joisha. Overlooking roots: A framework for making nondeferred reference-counting garbage collection fast. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 141–158.

[Joisha, 2008] Pramod Joisha. A principled approach to nondeferred reference-counting garbage collection. In Gregg et al. [Gregg *et al.*2008].

[Jokinen, 1989] M. O. Jokinen. Customizable garbage collectors. *Information Processing Letters*, 30(3):115–118, February 1989.

[Jones,] Richard Jones. The garbage collection page. The definitive on-line resource for garbage collection material.

[Jones and Blackburn, 2008] Richard Jones and Steve Blackburn, editors. *Proceedings of the Seventh International Symposium on Memory Management*, Tucson, AZ, USA, June 2008. ACM Press.

[Jones and Kelly,] Richard Jones and Paul Kelly. Bounds checking for C. http://www-ala.doc.ic.ac.uk/ pjhk/BoundsChecking.html.

[Jones and King, 2004] Richard E. Jones and Andy C. King. Collecting the garbage without blocking the traffic. Technical Report 18–04, Computing Laboratory, University of Kent, September 2004. This report summarises [King, 2004].

[Jones and King, 2005] Richard E. Jones and Andy C. King. A fast analysis for thread-local garbage collection with dynamic class loading. In *Fifth IEEE International Workshop on Source Code Analysis and Manipulation*, pages 129–138, Budapest, September 2005. This is a shorter version of [Jones and King, 2004].

[Jones and le Métayer, 1988] Simon B. Jones and D. le Métayer. Optimisation of storage management in functional languages by static analysis of programs. In Hall et al. [Hall *et al.*1988], pages 87–100.

[Jones and le Métayer, 1989] Simon B. Jones and D. le Métayer. Compile-time garbage collection by sharing analysis. In FPCA 1989 [FPCA 19891989], pages 54–74.

[Jones and Lins, 1992] Richard E. Jones and Rafael D. Lins. Cyclic weighted reference counting without delay. Technical Report 28–92, Computing Laboratory, The University of Kent at Canterbury, December 1992.

[Jones and Lins, 1993] Richard E. Jones and Rafael D. Lins. Cyclic weighted reference counting without delay. In Bode et al. [Bode *et al.*1993], pages 712–715.

[Jones and Muchnick, 1981] Neil D. Jones and Steven S. Muchnick. Flow analysis and optimization of LISP-like structures. In *Program Flow Analysis: Theory and Applications*, pages 102–131. Prentice-Hall, 1981.

[Jones and Ryder, 2006] Richard Jones and Chris Ryder. Garbage collection should be lifetime aware. In Olivier Zendra, editor, *Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS'2006)*, page 8, Nantes, France, July 2006.

[Jones and Ryder, 2008] Richard Jones and Chris Ryder. A study of Java object demographics. In Jones and Blackburn [Jones and Blackburn2008], pages 121–130.

[Jones and Tyas, 1993] Simon B. Jones and Andrew S. Tyas. The implementer's dilemma: A mathematical model of compile-time garbage collection. In GWFP 1993 [GWFP 19931993], pages 139–144.

[Jones and White, 1991] Simon B. Jones and M. White. Is compile time garbage collection worth the effort. In Peyton Jones et al. [Peyton Jones *et al.*1991], pages 172–176.

[Jones *et al.*, 1999] Simon Peyton Jones, Norman Ramsey, and Fermin Reig. C–: a portable assembly language that supports garbage collection. In *International Conference on Principles and Practice of Declarative Programming*, September 1999.

[Jones *et al.*, 2000] Simon Peyton Jones, Simon Marlow, and Conal Elliott. Stretching the storage manager: Weak pointers and stable names in Haskell. In *Eleventh International Workshop on the Implementation of Functional Languages*, volume 1868 of *Lecture Notes in Computer Science*, pages 37–58. Springer-Verlag, 2000.

[Jones, 1992] Richard E. Jones. Tail recursion without space leaks. *Journal of Functional Programming*, 2(1):73–79, January 1992.

[Jones, 1995] Simon B. Jones. An experiment in compile time garbage collection. Technical Report 84, Programming Methodology Group, Göteborg University and Chalmers University of Technology, January 1995.

[Jones, 1996] Richard E. Jones. *Garbage Collection: Algorithms for Automatic Dynamic Memory Management*. Wiley, Chichester, July 1996. With a chapter on Distributed Garbage Collection by R. Lins.

[Jones, 1996 2009] Richard Jones. The garbage collection bibliography. http://www.cs.ukc.ac.uk/people/staff/rej/gcbib/gcbib.html, 1996–2009.

[Jones, 2006] Richard Jones. Five perspectives on modern memory management: Systems, hardware and theory. *Science of Computer Programming*, 62(2):95–204, October 2006.

[Jones, 2007] Richard Jones. Dynamic memory management: Challenges for today and tomorrow. In *International Lisp Conference*, pages 115–124, Cambridge, April 2007. Association of Lisp Users. Invited presentation.

[Jonker, 1992] J. E. Jonker. On-the-fly garbage collection for several mutators. *Distributed computing*, 5(4):187–200, April 1992.

[Jonkers, 1979] H. B. M. Jonkers. A fast garbage compaction algorithm. *Information Processing Letters*, 9(1):26–30, July 1979.

[Jonkers, 1983] H. B. M. Jonkers. *Abstraction, Specification and Implementation Techniques: With an Application to Garbage Collection*. Mathematical Centre, Amsterdam, the Netherlands, 1983.

[Jordan, 1978] M. J. Jordan. Slp: A paged processor for compact lists. *Software Practice and Experience*, 8(3):285–301, 1978.

[Jost, 2004] Steffen Jost. lfd_infer: An implementation of a static inference on heap space usage. In SPACE 2004 [SPACE 20042004].

[Jouannaud, 1985] Jean-Pierre Jouannaud, editor. *Record of the 1985 Conference on Functional Programming and Computer Architecture*, volume 201 of *Lecture Notes in Computer Science*, Nancy, France, September 1985. Springer-Verlag.

[Jouppi, 1990] Norman P. Jouppi. Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers. In *17th Annual International Symposium on Computer Architecture*, pages 346–373, May 1990.

[Jouppi, 1993] Norman P. Jouppi. Cache write policies and performance. In ISCA 1993 [ISCA 19931993], pages 191–201.

[Joy *et al.*, 2000] Bill Joy, Guy Steele, James Gosling, and Gilad Bracha. *The Java Language Specification*. Addison-Wesley, second edition edition, June 2000.

[JPDA,] Sun Microsystems. *Java Platform Debugger Architecture (JPDA)*. http://java.sun.com/j2se/1.3/docs/guide/jpda/.

[JProbe,] Sitraka Inc. *The JProbe Profiler*. http://www.jprobe.com.

[JRockit, 2008] BEA Systems Inc. *Understanding Memory Management*, 2008. The BEA JRockit JVM R27.6.

[JTRES 2003, 2003] *International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, volume 2889 of *Lecture Notes in Computer Science*. Springer, 2003.

[JTRES 2005, 2005] *Proceedings of the 3rd International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, San Diego, CA, 2005.

[JTRES 2007, 2007] *Proceedings of the 5th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, Vienna, Austria, September 2007. ACM Press.

[JTRES 2008, 2008] *Proceedings of the 6th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, Santa Clara, California, September 2008. ACM Press.

[JTRES 2009, 2009] *Proceedings of the 7th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, September 2009.

[Juillerat *et al.*, 2007] Nicolas Juillerat, Stefan Mller Arisona, and Simon Schubiger-Banz. Real-time, low latency audio processing in Java. In *International Computer Music Conference, ICMC 2007*, Copenhagen, Denmark, August 2007.

[Jul and Juul, 1990] Eric Jul and Niels-Christian Juul, editors. *OOPSLA/ECOOP Workshop on Garbage Collection in Object-Oriented Systems*, Ottawa, Canada, October 1990.

[Jul *et al.*, 1987] Eric Jul, Henry Levy, Norman Hutchinson, and Andrew Black. Fine-grained mobility in the Emerald system. In *Eleventh ACM Symposium on Operating Systems Principle*, pages 105–106. ACM Press, December 1987. See also [Jul *et al.*, 1988].

[Jul *et al.*, 1988] Eric Jul, Henry Levy, Norman Hutchinson, and Andrew Black. Fine-grained mobility in the Emerald system. *ACM Transactions on Computer Systems*, 6(1):109–133, January 1988.

[Jul, 1988] Eric Jul. *Object Mobility in a Distributed Object-Oriented System*. PhD thesis, Department of Computer Science, University of Washington, Seattle, Washington, December 1988. Technical Report 88-12-6. Also DIKU Report (Blue Series) 89/1, Department of Computer Science, University of Copenhagen, Denmark.

[Jul, 1998] Eric Jul, editor. *Proceedings of 12th European Conference on Object-Oriented Programming, ECOOP98*, volume 1445 of *Lecture Notes in Computer Science*, Brussels, July 1998. Springer-Verlag.

[Jula and Rauchwerger, 2009] Alin Jula and Lawrence Rauchwerger. Two memory allocators that use hints to improve locality. In Kolodner and Steele [Kolodner and Steele2009], pages 109–118.

[Julien, 1985] Danielle Julien. *Etude et Réalisation de la Machine Virtuelle LILA Adaptée à l'Écriture d'Interprètes*. PhD thesis, Université Paul Sabatier, Toulouse, France, May 1985.

[Jump and McKinley, 2007] Maria Jump and Kathryn S. McKinley. Cork: Dynamic memory leak detection for garbage-collected languages. In *Conference Record of the Thirty-Fourth Annual ACM Symposium on Principles of Programming Languages*, Nice, France, January 2007.

[Jump and McKinley, 2009] Maria Jump and Kathryn S. McKinley. Dynamic shape analysis via degree metrics. In Kolodner and Steele [Kolodner and Steele2009], pages 119–128.

[Jump and McKinley, 2010] Maria Jump and Kathryn S. McKinley. Detecting memory leaks in managed languages with cork. *Software Practice and Experience*, 40(1):1–22, January 2010.

[Jump *et al.*, 2004] Maria Jump, Stephen M. Blackburn, and Kathryn S. McKinley. Dynamic object sampling for pretenuring. In Bacon and Diwan [Bacon and Diwan2004], pages 152–162.

[Jung and Yi, 2008] Yungbum Jung and Kwangkeun Yi. Practical memory leak detector based on parameterized procedural summaries. In Jones and Blackburn [Jones and Blackburn2008], pages 131–140.

[Juul and Jul, 1992] Neils-Christian Juul and Eric Jul. Comprehensive and robust garbage collection in a distributed system. In Bekkers and Cohen [Bekkers and Cohen1992].

[Juul, 1990a] Nils Christian Juul. A distributed, faulting garbage collector for Emerald. In Jul and Juul [Jul and Juul1990].

[Juul, 1990b] Nils Christian Juul. Report on the 1990 workshop on garbage collection in object-oriented systems. In OOPSLA 1990 [OOPSLA 19901990]. Addendum.

[Juul, 1992] Niels Christian Juul. *Comprehensive, Concurrent, and Robust Garbage Collection in the Distributed, Object-Based System, Emerald*. PhD thesis, Department of Computer Science (DIKU), University of Copenhagen, 1992.

[JVM 2001, 2001] *Proceedings of the First Java Virtual Machine Research and Technology Symposium*, Monterey, CA, USA, April 2001. USENIX.

[JVM 2002, 2002] *Proceedings of the Second Java Virtual Machine Research and Technology Symposium*, San Francisco, CA, USA, August 2002. USENIX.

[JVM 2004, 2004] *Proceedings of the Third Java Virtual Machine Research and Technology Symposium*, San Jose, CA, USA, May 2004. USENIX.

[JVMPI,] Sun Microsystems. *Java Virtual Machine Profiling Interface (JVMPI)*. http://java.sun.com/j2se/1.3/docs/guide/jvmpi/.

[Kaashoek *et al.*, 1989] M. Frans Kaashoek, Andrew Tanenbaum, S. Hummel, and Henri E. Bal. An efficient reliable broadcast protocol. *ACM SIGOPS Operating Systems Review*, 23(4):5–19, October 1989.

[Kaburlasos, 1992] Nikos Kaburlasos. Hardware support for garbage collection in the C programming language. Master's thesis, University of Texas at Austin, 1992.

[Kaehler and Krasner, 1983] Ted Kaehler and Glenn Krasner. LOOM — large object-oriented memory for Smalltalk-80 systems. In Krasner [Krasner1983], pages 251–271.

[Kaehler, 1981] Ted Kaehler. Virtual memory for an object-oriented language. *Byte*, 6(8):378–387, August 1981.

[Kaehler, 1986] Ted Kaehler. Virtual memory on a narrow machine for an object-oriented language. In OOPSLA 1986 [OOPSLA 19861986], pages 87–106.

[Kafura *et al.*, 1990] Dennis Kafura, Doug Washabaugh, and Jeff Nelson. Garbage collection of actors. In OOPSLA 1990 [OOPSLA 19901990], pages 126–134.

[Kafura *et al.*, 1995] Dennis Kafura, Manibrata Mukherji, and Doug Washabaugh. Concurrent and distributed garbage collection of active objects. *IEEE Transactions on Parallel and Distributed Systems*, 6(4), April 1995.

[Kagedal and Debray, 1997] Andreas Kagedal and Saumya K. Debray. A practical approach to structure reuse of arrays in single assignment languages. In *International Conference on Logic Programming*, pages 18–32, 1997.

[Kagimasa *et al.*, 1991] T. Kagimasa, K. Takahashi, and S. Yoshizumi. Adaptive storage management for very large virtual/real storage systems. In ISCA 1991 [ISCA 19911991], pages 372–379.

[Kahn, 1987] Gilles Kahn, editor. *Record of the 1987 Conference on Functional Programming and Computer Architecture*, volume 274 of *Lecture Notes in Computer Science*, Portland, Oregon, September 1987. Springer-Verlag.

[Kain, 1969] Y. Kain. Block structures, indirect addressing, and garbage collection. *Communications of the ACM*, 12(7):395–398, July 1969.

[Kakkad *et al.*, 1998] Sheetal V. Kakkad, Mark S. Johnstone, and Paul R. Wilson. Portable run-time type description for conventional compilers. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 146–153.

[Kakuta *et al.*, 1986] K. Kakuta, H. Nakamura, and S. Iida. Parallel reference counting algorithm. *Information Processing Letters*, 23(1):33–37, 1986.

[Kal,] Kaleida Labs. *ScriptX Architectural Overview*.

[Kalagnanam and Kodi, 2003] Aruna Kalagnanam and Sripathi Kodi. Mash that trash — incremental compaction in the IBM JDK garbage collector: How to minimize pause times and free the heap from dark matter. *IBM developerWorks*, June 2003.

[Kalibera *et al.*, 2009a] T. Kalibera, J. Hagelberg, F. Pizlo, A. Plsek, B. Titzer, and J. Vitek. CDx: A family of real-time Java benchmarks. In JTRES 2009 [JTRES 20092009], pages 41–50.

[Kalibera *et al.*, 2009b] T. Kalibera, M. Prochazka, F. Pizlo, M. Decky, V. Vitek, and M. Zulianello. Real-time Java in space: Potential benefits and open challenges. In *Proceedings of Data Systems in Aerospace (DASIA 2009)*, January 2009.

[Kalibera *et al.*, 2009c] Tomas Kalibera, Filip Pizlo, Antony L. Hosking, and Jan Vitek. Scheduling hard real-time garbage collection. In *Proceedings of the 30th IEEE Real-Time Systems Symposium (RTSS)*, pages 81–92, December 2009.

[Kalibera, 2009] Tomas Kalibera. Replicating real-time garbage collector for Java. In JTRES 2009 [JTRES 20092009].

[Kalsow and Muller, 1989] Bill Kalsow and Eric Muller. *SRC Modula-3, Version 1.2*, December 1989.

[Kamada *et al.*, 1993] Tomio Kamada, Satoshi Matsuoka, and Akinori Yonezawa. Efficient parallel global garbage collection on massively parallel computers. In Moss et al. [Moss *et al.*1993].

[Kamada *et al.*, 1994] Tomio Kamada, Satoshi Matsuoka, and Akinori Yonezawa. Efficient parallel global garbage collection on massively parallel computers. In G.M. Johnson, editor, *Supercomputing'94*, pages 79–88. IEEE Press, 1994.

[Kandu *et al.*, 1987] B. Kandu, S. Heng, C. Wu, and Nader Bagherzadeh. Network simulation of synchronous garbage collection algorithm. In P. Roth, editor, *Simulation of computer networks1987 Symposium on the simulation of computer networks, Colorado Springs, Aug. 4–7, 1987*, pages 215–222. IEEE Press, 1987.

[Kaplan *et al.*, 2002] Scott Kaplan, Lyle McGeoch, and Megan Cole. Adaptive caching for demand prepaging. In Boehm and Detlefs [Boehm and Detlefs2002], pages 114–126.

[Karkare *et al.*, 2006] Amey Karkare, Amitabha Sanyal, and Uday Khedker. Effectiveness of garbage collection in MIT/GNU Scheme. arXiv.org cs.PL/0611093, November 2006.

[Katz, 1986] M. J. Katz. ParaTran: A transparent, transaction based runtime mechanism for parallel execution of Scheme. Master's thesis, MIT Press, June 1986.

[Kaufman, 1984] Arie Kaufman. Tailored-list and recombination-delaying buddy systems. *ACM Transactions on Programming Languages and Systems*, 6(4):118–125, 1984.

[Kawachiya *et al.*, 2008] Kiyokuni Kawachiya, Kazunori Ogata, and Tamiya Onodera. Analysis and reduction of memory inefficiencies in Java strings. In OOPSLA 2008 [OOPSLA 20082008].

[Kawakami and Gurd, 1986] K. Kawakami and J. R. Gurd. Scalable dataflow structure store. In *13th Annual International Symposium on Computer Architecture — Conference Proceedings. Tokyo, 1986 Jun 2–5*. IEEE Press, 1986.

[Kempf, 2001] William E. Kempf. A garbage collection framework for C++. The Vode Project website, January 2001.

[Kennedy and Syme, 2004] Andrew Kennedy and Don Syme. Combining generics, pre-compilation and sharing between software-based processes. In SPACE 2004 [SPACE 20042004].

[Kennedy, 1991] Brian Kennedy. The features of the object oriented abstract type hierarchy (OATH). In *Proceedings of the Usenix C++ Conference*, pages 41–50. Usenix Association, April 1991.

[Kenny and Lin, 1991] K. B. Kenny and K. Lin. Flexible real-time systems using the Flex language. *IEEE Transactions on Computers*, pages 70–78, May 1991.

[Kermany and Petrank, 2006] Haim Kermany and Erez Petrank. The Compressor: Concurrent, incremental and parallel compaction. In Schwartzbach and Ball [Schwartzbach and Ball2006], pages 354–363.

[Kermarrec *et al.*, 1995] A. M. Kermarrec, G. Cabillic, A. Gefflaut, C. Morin, and Isabbelle Puaut. A recoverable distributed shared memory integrating coherence and recoverability. In *Twenty-Fifth Annual International Symposium on Fault-Tolerant Computing*, 1995.

[Kero *et al.*, 2007] Martin Kero, Johan Nordlander, and Per Lundgren. A correct and useful incremental copying garbage collector. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 129–140.

[Kessler and Livny, 1989] Richard E. Kessler and M. Livny. An analysis of distributed shared memory algorithms. In *Proceedings of the 9th International Conference on Distributed Computing Systems*, June 1989.

[Kessler *et al.*, 1989] Richard Kessler, Richard Jooss, Alvin Lebeck, and Mark D. Hill. Inexpensive implementations of set-associativity. In *16th Annual International Symposium on Computer Architecture*, pages 131–139, Jerusalem, Israel, June 1989. IEEE-ACM.

[Kessler *et al.*, 1992] Richard Kessler, H. Carr, L. Stoller, and M. Swanson. Implementing Concurrent Scheme for the Mayfly distributed parallel processing system. *Lisp and Symbolic Computation*, 5(1):73–93, May 1992.

[Keyngnaert, 2001] Peter Keyngnaert. Conflict graph based allocation of static objects to memory banks. In SPACE 2001 [SPACE 20012001].

[Khedker *et al.*, 2006] Uday Khedker, Amitabha Sanyal, and Amey Karkare. Heap reference analysis using access graphs. ACM Computing Research Repository, August 2006.

[Khedker *et al.*, 2008] Uday Khedker, Amitabha Sanyal, and Amey Karkare. Heap reference analysis using access graphs. *ACM Transactions on Programming Languages and Systems*, 30(1), 2008.

[Kim and Hsu, 2000] Jin-Soo Kim and Yarsun Hsu. Memory system behavior of Java programs: Methodology and analysis. In *Proceedings of SIGMETRICS International Conference on Measurements and Modeling of Computer Systems*, pages 264–274. ACM Press, July 2000.

[Kim and Shin, 2004] Taehyoun Kim and Heonshik Shin. Scheduling-aware real-time garbage collection using dual aperiodic servers. In *Real-Time and Embedded Computing Systems and Applications*, volume 2968 of *Lecture Notes in Computer Science*, pages 1–17, 2004.

[Kim *et al.*, 1998] Jin-Soo Kim, Xiaohan Qin, and Yarsun Hsu. Memory characterization of a parallel data mining workload. In *Proc. Workload Characterization: Methodology and Case Studies*. IEEE Press, November 1998.

[Kim *et al.*, 1999] Taehyoun Kim, Naehyuck Chang, Namyun Kim, and Heonshik Shin. Scheduling garbage collector for embedded real-time systems. In LCTES 1999 [LCTES 19991999], pages 55–64.

[Kim *et al.*, 2000] Taehyoun Kim, Naehyuck Chang, and Heonshik Shin. Bounding worst case garbage collection time for embedded real-time systems. In *Proceedings of the Sixth IEEE Real Time Technology and Applications Symposium (RTAS 2000)*, 2000.

[Kim *et al.*, 2001] Taehyoun Kim, Naehyuck Chang, and Heonshik Shin. Joint scheduling of garbage collector and hard real-time tasks for embedded applications. *Journal of Systems and Software*, 58(3):247–260, September 2001.

[Kimura *et al.*, 1987] Y. Kimura, K. Nishida, N. Miyauchi, and T. Chikayama. Realtime GC by multiple reference bit in KL1. In *Proceedings of the Data Flow Workshop*, pages 215–222, October 1987. In Japanese.

[King, 2002] Andy C. King. Removing GC synchronisation. In OOPSLA 2002 [OOPSLA 20022002], pages 112–113 (Companion).

[King, 2004] Andy C. King. *Removing Garbage Collector Synchronisation*. PhD thesis, Computing Laboratory, The University of Kent at Canterbury, 2004.

[Kingdon *et al.*, 1991] H. Kingdon, David R. Lester, and Geoffrey L. Burn. The HDG-machine: A highly distributed graph reducer for a transputer network. *Computer Journal*, 34:290–301, September 1991.

[Kirby *et al.*, 2001] Graham N. C. Kirby, Alan Dearle, and Dag I. K. Sjberg, editors. *Proceedings of the Ninth International Workshop on Persistent Object Systems (September, 2000)*, volume 2135 of *Lecture Notes in Computer Science*, Lillehammer, Norway, 2001. Springer.

[Kistler and Franz, 1998] Thomas Kistler and Michael Franz. Automated layout of data members for type-safe languages. Technical Report 98–22, University of California, Irvine, May 1998.

[Kistler and Franz., 1999] Thomas Kistler and Michael Franz. The case for dynamic optimization: Improving memory-hierarchy performance by continuously adapting the internal storage layout of heap objects at run-time. Technical Report 99–21, University of California, Irvine, May 1999.

[Kistler and Franz, 2000] Thomas Kistler and Michael Franz. Automated data-member layout of heap objects to improve memory-hierarchy performance. *ACM Transactions on Programming Languages and Systems*, 22(3):490–505, May 2000.

[Kjelso and Jones, 1995] Morten Kjelso and Simon Jones. Memory management in flash-memory disks with data compression. In Baker [Baker1995a].

[Klaiber and Levy, 1991] Alexander C. Klaiber and Henry M. Levy. An architecture for software-controlled data prefetching. In *18th Annual International Symposium on Computer Architecture*, pages 43–53, Toronto, Canada, May 1991. Association for Computing Machinery.

[Kliot *et al.*, 2009] Gabriel Kliot, Erez Petrank, and Bjarne Steensgaard. A lock-free, concurrent, and incremental stack scanning for garbage collectors. In Hosking et al. [Hosking *et al.*2009].

[Kluk, 1989] Mark G. Kluk. A study of garbage collection schemes for list processors. Master's thesis, Lehigh University, 1989.

[Kluźniak, 1988] F. Kluźniak. Compile-time garbage collection for ground Prolog. In *5th International Conference and Symposium on Logic Programming*, pages 1490–1505, 1988.

[Knight, 1974] Tom Knight. CONS. Working Paper 80, MIT AI Laboratory, November 1974.

[Knowlton, 1965] Kenneth C. Knowlton. A fast storage allocator. *Communications of the ACM*, 8(10):623–625, October 1965.

[Knudsen, 2001] Jørgen Lindskov Knudsen, editor. *Proceedings of 15th European Conference on Object-Oriented Programming, ECOOP 2001*, volume 2072 of *Lecture Notes in Computer Science*, Budapest, June 2001. Springer-Verlag.

[Knuth, 1973a] Donald E. Knuth. *The Art of Computer Programming*, volume I: Fundamental Algorithms. Addison-Wesley, second edition, 1973.

[Knuth, 1973b] Donald E. Knuth. *Lists and Garbage Collection*, chapter 2, pages 408–423. Volume I: Fundamental Algorithms of Knuth Volume 1 [Knuth1973a], second edition, 1973.

[Ko *et al.*, 2008] Sohyang Ko, Seonsoo Jun, Kiyong Kim, and Yeonseung Ry. Study on garbage collection schemes for flash-based linux swap system. In *Advanced Software Engineering and Its Applications, ASEA 2008*, pages 13–16, December 2008.

[Koch *et al.*, 1991] B. Koch, T. Schunke, A. Dearle, F. Vaughan, C. Marlin, R. Fazekerley, and C. Barter. Cache coherence and storage management in a persistent object system. In Dearle et al. [Dearle *et al.*1991], pages 99–109.

[Kogan and Schuster, 1997a] Dmitry Kogan and Assaf Schuster. Collecting garbage pages in a distributed shared memory system. In *Proceedings of Fifth European Symposium on Algorithms*, pages 308–325, Graz, September 1997.

[Kogan and Schuster, 1997b] Dmitry Kogan and Assaf Schuster. Remote reference counting: Distributed garbage collection with reduced memory and communication overhead. In *Proceedings of Fifth European Symposium on Algorithms*, pages 308–325, Graz, September 1997.

[Koide and Noshita, 1993] Hiroshi Koide and K. Noshita. On the copying garbage collection which preserves the genetic order. *Transaction of Information Processing (IPSJ)*, 34(11):2395–2400, November 1993. In Japanese.

[Koide, 1993] Hiroshi Koide. Hybrid garbage collection. Master's thesis, University of Electro-Communications, Tokyo, 1993.

[Kölling and Rosenberg, 1996] Michael Kölling and John Rosenberg. Blue — a language for teaching object-oriented programming. In *Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education*, pages 190–194, March 1996.

[Kolodner and Petrank, 1999] Elliot K. Kolodner and Erez Petrank. Parallel copying garbage collection using delayed allocation. Technical Report 88.384, IBM Haifa Research Lab., November 1999.

[Kolodner and Steele, 2009] Hillel Kolodner and Guy Steele, editors. *Proceedings of the Eighth International Symposium on Memory Management*, Dublin, Ireland, June 2009. ACM Press.

[Kolodner and Weihl, 1992] Elliot K. Kolodner and W. E. Weihl. Atomic incremental garbage collection. In Bekkers and Cohen [Bekkers and Cohen1992].

[Kolodner and Weihl, 1993] Elliot K. Kolodner and William E. Weihl. Atomic incremental garbage collection and recovery for large stable heap. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of 1993 ASM SIGMOD International Conference on the Management of Data*, pages 177–186, Washington, DC, May 1993. Also MIT/LCS/TR-534, February, 1992.

[Kolodner *et al.*, 1989] Elliot K. Kolodner, Barbara Liskov, and W. Weihl. Atomic garbage collection: Managing a stable heap. *SIGMOD Record*, 18(2):15–25, June 1989. Proceedings of 1989 ACM SIGMOD International Conference on Management of Data.

[Kolodner, 1987] Elliot K. Kolodner. Recovery using virtual memory. Technical Memo MIT/LCS/TM–404, MIT Laboratory for Computer Science, July 1987.

[Kolodner, 1991] Elliot K. Kolodner. Atomic incremental garbage collection and recovery for large stable heap, implementing persistent object bases: Principles and practice. In Dearle et al. [Dearle *et al.*1991].

[Kolodner, 1992] Elliot K. Kolodner. *Atomic Incremental Garbage Collection and Recovery for a Large Stable Heap*. PhD thesis, MIT Press, 1992. MIT/LCS/TR 534.

[Koopman *et al.*, 1989] Philip J. Koopman, Jr., Peter Lee, and Daniel P. Siewiorek. Cache performance of combinator graph reduction. In PLDI 1989 [PLDI 19891989], pages 110–119.

[Koopman *et al.*, 1992] Philip J. Koopman, Peter Lee, and Daniel P. Siewiorek. Cache behavior of combinator graph reduction. *ACM Transactions on Programming Languages and Systems*, 14(2):265–297, April 1992.

[Kordale and Ahamad, 1993] R. Kordale and Mustaque Ahamad. A scalable cyclic garbage detection algorithm for distributed systems. In Moss et al. [Moss *et al.*1993].

[Kordale *et al.*, 1992] R. Kordale, John Shilling, and Mustaque Ahamad. Garbage collection in distributed shared memory systems. Technical Report TR GIT-CC-92/45, Georgia Institute of Technology, 1992.

[Kordale *et al.*, 1993] R. Kordale, Mustaque Ahamad, and John Shilling. Distributed/concurrent garbage collection in distributed shared memory systems. In IWOOOS 1993 [IWOOOS 19931993].

[Korn and Vo, 1985] David G. Korn and Kiem-Phong Vo. In search of a better malloc. In *Proceedings of the Summer 1985 USENIX Conference*, pages 489–506. USENIX Association, 1985.

[Kowaltowski, 1979] T. Kowaltowski. Data structures and correctness of programs. *Journal of the ACM*, 26(2):283–301, April 1979.

[Krakowiak and Shrivastava, 2000] S. Krakowiak and S.K. Shrivastava, editors. *Recent Advances in Distributed Systems*, volume 1752 of *Lecture Notes in Computer Science*, February 2000.

[Krall, 2006] Andreas Krall. Static verification of global heap references in Java native libraries. In SPACE 2006 [SPACE 20062006], pages 98–100.

[Kranz *et al.*, 1986] David A. Kranz, Richard Kelsey, Jonathan Rees, Paul Hudak, James Philbin, and Norman Adams. ORBIT: An optimizing compiler for Scheme. In SIGPLAN 1986 [SIGPLAN 19861986], pages 219–233.

[Kranz *et al.*, 1989] David A. Kranz, Robert H. Halstead, and Eric Mohr. Mul-T: A high-performance parallel Lisp. In PLDI 1989 [PLDI 19891989], pages 81–90.

[Kranz, 1988] David A. Kranz. *ORBIT: An Optimizing Compiler For Scheme*. PhD thesis, Yale University, February 1988.

[Krasner, 1983] Glenn Krasner, editor. *Smalltalk-80: Bits of History, Words of Advice*. Addison-Wesley, 1983.

[Kriegel, 1993] E. Ulrich Kriegel. A conservative garbage collector for an EuLisp to ASM/C compiler. In Moss et al. [Moss *et al.*1993].

[Krieger and Stumm, 1990] O. Krieger and Michael Stumm. An optimistic approach for consistent replicated data for multicomputers. In *Proc. HICCSS*, 1990.

[Krintz *et al.*, 2007] Chandra Krintz, Steven Hand, and David Tarditi, editors. *Proceedings of the Third ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, San Diego, CA, USA, June 2007.

[Krishnaswami, 2006] Neelakantan Krishnaswami. Separation logic for a higher-order typed language. In SPACE 2006 [SPACE 20062006], pages 73–82.

[Krogdahl, 1973] S. Krogdahl. A dynamic storage allocation problem. *Information Processing Letters*, 2:96–99, 1973.

[Krueger, 1989] Steven Krueger. Vlsi-appropriate garbage collection support. In Jose G Delgado-Frias and Will R. Moore, editors, *VLSI for artificial intelligence*, The Kluwer international series in engineering and computer science. Kluwer Academic Publishers, 1989.

[Küchlin and Nevin, 1991a] Wolfgang Küchlin and Nicholas J. Nevin. On multi-threaded list-processing and garbage collection. Technical Report OSU-CISRC-3/91-TR11, Ohio State University, March 1991.

[Küchlin and Nevin, 1991b] Wolfgang Küchlin and Nicholas J. Nevin. On multi-threaded list-processing and garbage collection. In *Proceedings of Third IEEE Symposium on Parallel and Distributed Processing*, pages 894–897. IEEE Press, December 1991.

[Küchlin, 1991] Wolfgang Küchlin. A space-efficient parallel garbage collection algorithm. In *Proceedings of Fifth ACM International Conference on Supercomputing*, pages 40–46. ACM Press, June 1991.

[Kumar and Li, 2002] Sanjeev Kumar and Kai Li. Dynamic memory management for programmable devices. In Boehm and Detlefs [Boehm and Detlefs2002], pages 139–149.

[Kung and Song, 1977a] H. T. Kung and S. W. Song. An efficient parallel garbage collection system and its correctness proof. In *IEEE Symposium on Foundations of Computer Science*, pages 120–131. IEEE Press, 1977.

[Kung and Song, 1977b] H. T. Kung and S. W. Song. An efficient parallel garbage collection system and its correctness proof. Department of computer science report, Carnegie Mellon University, September 1977.

[Kung and Song, 1977c] H. T. Kung and S. W. Song. Performance analysis of a parallel garbage collection system. Department of computer science report, Carnegie Mellon University, August 1977.

[Kung, 1983] Delphine T. H Kung. Garbage collection of linked data structures: An example in a network oriented database management system. Master's thesis, Rochester Institute of Technology, 1983.

[Kurihara *et al.*, 1990] Satoshi Kurihara, Mikio Inari, Norihisa Doi, Kazuki Yasumatsu, and Takemi Yamazaki. SPiCE collector : The run-time garbage collector for Smalltalk-80 programs translated into C. In Jul and Juul [Jul and Juul1990].

[Kurokawa, 1975] T. Kurokawa. New marking algorithms for garbage collection. In *Proceedings of the 2nd USA–Japan Computer Conference*, pages 585–584, 1975.

[Kurokawa, 1981] T. Kurokawa. A new fast and safe marking algorithm. *Software Practice and Experience*, 11:671–682, 1981.

[Kuse and Kamimura, 1991] Kazushi Kuse and Tsutomu Kamimura. Generational garbage collection for C-based object-oriented languages. In Wilson and Hayes [Wilson and Hayes1991a].

[Kwon and Koh, 2007] Ohhoon Kwon and K Koh. Swap-aware garbage collection for NAND flash memory based embedded systems. In *7th IEEE International Conference on Computer and Information Technology, CIT 2007*, pages 787–792, November 2007.

[Kwon and Wellings, 2004] J. Kwon and A. Wellings. Memory management based on method invocation in RTSJ. In Meersman et al. [Meersman *et al.*2004], pages 333–345.

[Kwon *et al.*, 2003] J. Kwon, A. Wellings, and S. King. Predictable memory utilization in the Ravenscar-Java profile. In Puschner et al. [Puschner *et al.*2003], pages 267–274.

[Kwon *et al.*, 2007a] Ohhoon Kwon, Jaewoo Lee, and Kern Koh. EF-greedy: A novel garbage collection policy for flash memory based embedded systems. In *Computational Science, ICCS 2007*, volume 4490 of *Lecture Notes in Computer Science*, pages 913–920, 2007.

[Kwon *et al.*, 2007b] Ohhoon Kwon, Yeonseung Ryu, and Kern Koh. An efficient garbage collection policy for flash memory based swap systems. In *Proceedings of the 2007 International Conference on computational Science and its Applications , ICCSA'07*, pages 213–223, Kuala Lumpur, Malaysia, 2007. Springer-Verlag.

[Kyrylkov and Stefanović, 2005] S. Kyrylkov and Darko Stefanović. A study of garbage collection with a large address space for server applications. Technical Report TR–CS–2005–1, University of New Mexico, February 2005.

[Kyrylkov *et al.*, 2004] S. Kyrylkov, Darko Stefanović, and J. Eliot B. Moss. Design and implementation of a 64-bit PowerPC port of Jikes RVM 2.0.3. In *Second Workshop on Managed Runtime Environments (MRE'04)*, 2004.

[Kyrylkov, 2003] S. Kyrylkov. Jikes Research Virtual Machine — design and implementation of a 64-bit PowerPC port. Master's thesis, University of New Mexico, 2003.

[Kyrylkov, 2005] S. Kyrylkov. 64-bit computing and JVM performance. *Dr. Dobb's Journal*, 30(370):24–27, March 2005.

[Ladin and Liskov, 1992] Rivka Ladin and Barbara Liskov. Garbage collection of a distributed heap. In *International Conference on Distributed Computing Systems*, Yokohama, June 1992.

[Ladin, 1989] Rivka Ladin. *A Method for Constructing Highly Available Services and a Technique for Distributed Garbage Collection*. PhD thesis, MIT Press, 1989.

[Lai *et al.*, 2007] Chunrong Lai, Volosyuk Ivan, and Xiao-Feng Li. Behavior characterization and performance study on compacting garbage collectors with Apache Harmony. In *The Tenth Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW-10)*, Phoenix, AZ, February 2007. Held with HPCA-13.

[Lakhamraju *et al.*, 2000] M.K. Lakhamraju, R. Rastoji, S. Seshadri, and S. Sundarshan. On-line reorganization in object databases. *SIGMOD Record*, 28, May 2000. Proceedings of ACM SIGMOD International Conference on the Management of Data, Dallas, Texas.

[Lam *et al.*, 1991] Monica S. Lam, Edward E. Rothberg, and Michael E. Wolf. The cache performance and optimizations of blocked algorithms. In ASPLOS 1991 [ASPLOS 19911991], pages 63–74.

[Lam *et al.*, 1992] Michael S. Lam, Paul R. Wilson, and Thomas G. Moher. Object type directed garbage collection to improve locality. In Bekkers and Cohen [Bekkers and Cohen1992].

[Lam, 1992] Michael S. Lam. *Improving Locality via Garbage Collection*. PhD thesis, University of Illinois at Chicago, 1992.

[Lamb *et al.*, 1991] Charles Lamb, Gordon Landis, Jack Orenstein, and Dan Weinreb. The ObjectStore database system. *Communications of the ACM*, 34(10):50–63, October 1991.

[Lamport, 1976a] Leslie Lamport. Garbage collection with multiple processes: an exercise in cooperation. Technical Report CA-7602-2511, Computer Associates, Wakefield, MA, August 1976.

[Lamport, 1976b] Leslie Lamport. Garbage collection with multiple processes: an exercise in parallelism. In *Proceedings of the 1976 International Conference on Parallel Processing*, pages 50–54, 1976.

[Lamport, 1977] Leslie Lamport. Concurrent reading and writing. *Communications of the ACM*, 20(11):806–811, November 1977.

[Lamport, 1979] Leslie Lamport. A new approach to proving the correctness of multiprocess programs. *ACM Transactions on Programming Languages and Systems*, 1(1):84–97, July 1979.

[Lamport, 1991] Leslie Lamport. The temporal logic of actions. Research Report 79, DEC Systems Research Center, Palo Alto, CA, 1991.

[Lampson, 1983] Butler W. Lampson. A description of the Cedar language: A Cedar language reference manual. Technical Report CLS–83–15, Xerox PARC, Palo Alto, CA, 1983.

[Lang and Dupont, 1987] Bernard Lang and Francis Dupont. Incremental incrementally compacting garbage collection. In SIGPLAN 1987 [SIGPLAN 19871987], pages 253–263.

[Lang and Wegbreit, 1972] Bernard Lang and B. Wegbreit. Fast compactification. Technical Report 25–72, Harvard University, Cambridge, MA, November 1972.

[Lang *et al.*, 1992] Bernard Lang, Christian Quenniac, and José Piquer. Garbage collecting the world. In POPL 1992 [POPL 19921992], pages 39–50.

[Langendoen and Vree, 1991] Koen G. Langendoen and Willem G. Vree. FRATS: A parallel reduction strategy for shared memory. In M. Wirsing and J. Maluszynski, editors, *3rd international Symposium on Programming Language Implementation ans Logic Programming, Passau, Germany*, volume 528 of *Lecture Notes in Computer Science*, pages 99–110. Springer-Verlag, August 1991.

[Langendoen *et al.*, 1992] Koen G. Langendoen, Henk Muller, and Willem G. Vree. Memory management for parallel tasks in shared memory. In Bekkers and Cohen [Bekkers and Cohen1992].

[Larose and Feeley, 1998] Martin Larose and Marc Feeley. A compacting incremental collector and its performance in a production quality compiler. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 1–9.

[Larson, 1977] Richard G. Larson. Minimizing garbage collection as a function of region size. *SIAM Journal of Computing*, 6(4):663–668, December 1977.

[Larus and Hilfinger, 1988] James R. Larus and Paul N. Hilfinger. Detecting conflicts between structure accesses. In PLDI 1988 [PLDI 19881988], pages 21–34.

[Lassez, 1987] J. L. Lassez, editor. *Proceedings of Fourth International Conference on Logic Programming*, Melbourne, 1987.

[Lattner and Adve, 2003] Chris Lattner and Vikram Adve. Automatic pool allocation for disjoint data structures. In MSP 2002 [MSP 20022003].

[Lattner and Adve, 2005] Chris Lattner and Vikram S. Adve. Automatic pool allocation: Improving performance by controlling data structure layout in the heap. In Sarkar and Hall [Sarkar and Hall2005], pages 129–142.

[Layer and Richardson, 1991] D. K. Layer and C. Richardson. Lisp systems in the 1990s. *Communications of the ACM*, 34(9):48–57, 1991.

[LCT-RTS 1994, 1994] *Proceedings of the PLDI Workshop on Language, Compiler, and Tool Support for Real-Time Systems*, Orlando, FL, USA, June 1994.

[LCT-RTS 1995, 1995] *Proceedings of the ACM SIGPLAN Workshop on Languages, Compilers, and Tools for Real-Time Systems*, ACM SIGPLAN Notices 30(11), La Jolla, CA, USA, June 1995.

[LCTES 1999, 1999] *Proceedings of the ACM SIGPLAN Workshop onLanguages, Compilers, and Tools for Embedded Systems*, ACM SIGPLAN Notices 34(7), Atlanta, GA, USA, May 1999.

[LCTES 2000, 2000] *Proceedings of the ACM SIGPLAN Workshop onLanguages, Compilers, and Tools for Embedded Systems*, volume 1985 of *Lecture Notes in Computer Science*, Vancouver, Canada, June 2000. Springer.

[LCTES 2001, 2001] *Proceedings of the ACM SIGPLAN Workshop onLanguages, Compilers, and Tools for Embedded Systems*, ACM SIGPLAN Notices 36(8), Snowbird, UT, USA, June 2001.

[LCTES 2003, 2003] *Proceedings of the ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, ACM SIGPLAN Notices 38(7), San Diego, CA, USA, June 2003.

[LCTES 2004, 2004] *Proceedings of the ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, ACM SIGPLAN Notices 39(7), Washington, DC, USA, June 2004.

[LCTES 2005, 2005] *Proceedings of the ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, ACM SIGPLAN Notices 40(7), Chicago, IL, USA, June 2005.

[LCTES 2006, 2006] *Proceedings of the ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, ACM SIGPLAN Notices 41(7), Ottawa, Canada, June 2006.

[LCTES 2007, 2007] *Proceedings of the ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, ACM SIGPLAN Notices 42(7), San Diego, CA, USA, June 2007.

[LCTES 2008, 2008] *Proceedings of the ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, ACM SIGPLAN Notices 43(7), Tucson, AZ, USA, June 2008.

[LCTES/SCOPES 2002, 2002] *Proceedings of the joint conference on Languages, Compilers, and Tools for Embedded Systems: software and compilers for embedded systems*, ACM SIGPLAN Notices 37(7), Berlin, Germany, June 2002.

[Le Fessant *et al.*, 1997] Fabrice Le Fessant, Ian Piumarta, and Marc Shapiro. A detection algorithm for distributed cycles of garbage. In Dickman and Wilson [Dickman and Wilson1997].

[Le Fessant *et al.*, 1998] Fabrice Le Fessant, Ian Piumarta, and Marc Shapiro. An implementation for complete, asynchronous, distributed, garbage collection. In PLDI 1998 [PLDI 19981998], pages 152–161.

[Le Fessant, 1999] Fabrice Le Fessant. Detection of free distributed cycles in large-scale networks. http://beaune.inria.fr:1976/files/00040.ps.gz, January 1999.

[Le Fessant, 2001] Fabrice Le Fessant. Detecting distributed cycles of garbage in large-scale systems. In *Principles of Distributed Computing (PODC)*, Rhodes Island, August 2001.

[Le Huitouze, 1988] Serge Le Huitouze. *Mise en Oeuvre de PrologII/MALI*. PhD thesis, Université de Rennes I, 1988.

[Le Huitouze, 1990a] S. Le Huitouze. A new data structure for implementing extensions to Prolog. In P. Deransart and J. Maluszyński, editors, *International Workshop on Programming Languages Implementation and Logic Programming*, volume 456 of *Lecture Notes in Computer Science*, 1990.

[Le Huitouze, 1990b] Serge Le Huitouze. A new data structure for implementing extensions to Prolog. In *International Symposium on Programming Language Implementati on and Logic Programming*, pages 136–150, Linköping, Sweden, 1990.

[Le Sergent and Barthomieu, 1992] Thierry Le Sergent and Bernard Barthomieu. Incremental multithreaded garbage collection on virtually shared memory architectures. In Bekkers and Cohen [Bekkers and Cohen1992].

[Lea, 1993] Doug Lea. The GNU C++ library. *The C++ Report*, 1993.

[Lea, 1997] Doug Lea. A memory allocator. http://gee.cs.oswego.edu/dl/html/malloc.html, 1997.

[Lee and Barkley, 1989] T. Paul Lee and R. E. Barkley. Design and evaluation of a watermark-based lazy buddy system. *Performance Evaluation Review*, 17(1), May 1989.

[Lee and Chang, 2002] Woo Hyong Lee and J. Morris Chang. A study of dynamic memory management in C++ programs. *International Journal of Computer Languages*, 28:237–272, 2002.

[Lee and Chang, 2003a] Woo Hyong Lee and J. Morris Chang. A garbage collection policy based on empirical behavior. *International Journal of Information Sciences*, 2003. To appear.

[Lee and Chang, 2003b] Woo Hyong Lee and J. Morris Chang. An integrated dynamic memory tracing tool for C++ programs. *International Journal of Information Sciences*, 151:27–49, May 2003.

[Lee and Yi, 2004a] Oukseh Lee and Kwangkeun Yi. Experiments on the effectiveness of an automatic insertion of memory reuses into ML-like programs. In SPACE 2004 [SPACE 20042004].

[Lee and Yi, 2004b] Oukseh Lee and Kwangkeun Yi. Experiments on the effectiveness of an automatic insertion of memory reuses into ML-like programs. In Bacon and Diwan [Bacon and Diwan2004], pages 97–107.

[Lee and Zorn, 1997] Han Bok Lee and Benjamin G. Zorn. Bytecode instrumentation as an aid in understanding the behaviour of Java persistent stores. In Dickman and Wilson [Dickman and Wilson1997].

[Lee *et al.*, 1979] S. Lee, W. P. De Roever, and S. Gerhart. The evolution of list copying algorithms. In *6th ACM Symposium on Principles of Programming Languages*, pages 53–56, San Antonio, Texas, January 1979. ACM Press.

[Lee *et al.*, 1987] S. Lee, Heonshik Shin, and Miroslaw Malek. Parallel garbage collection with associative tag. In *Proceedings of International Conference on Computers and Applications June 1987*, Biejing, June 1987.

[Lee *et al.*, 2000a] Woo Hyong Lee, J. Morris Chang, and Yusuf Hasan. Dynamic memory measuring tool for C++ programs. In *Proceedings of The Third IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET 2000)*, Richardson, TX, March 2000.

[Lee *et al.*, 2000b] Woo Hyong Lee, J. Morris Chang, and Yusuf Hasan. Evaluation of a high-performance object reuse dynamic memory allocation policy for C++ programs. In *Proceedings of Fourth IEEE International Conference on High Performance Computing in Asia-Pacific Region*, pages 386–391, Beijing, China, May 2000.

[Lee, 1980] K. P. Lee. A linear algorithm for copying binary trees using bounded workspace. *Communications of the ACM*, 23(3):159–162, March 1980.

[Lee, 1988] Elgin Hoe-Sing Lee. Object storage and inheritance for SELF, a prototype-based object-oriented programming language. Engineer's thesis, Stanford University, Palo Alto, CA, December 1988.

[Lermen and Maurer, 1986] C.-W. Lermen and Dieter Maurer. A protocol for distributed reference counting. In LFP 1986 [LFP 19861986], pages 343–350.

[Leslie, 1975] Lamport Leslie. On-the-fly garbage collection: Once more with rigor. Technical Report CA-7508-1611, Computer Associates, Wakefield, MA, August 1975.

[Lester, 1989] David Lester. An efficient distributed garbage collector algorithm. In Odijik et al. [Odijik *et al.*1989].

[Lester, 1992] David Lester. Distributed garbage collection of cyclic structures. In *4th International Workshop on the Parallel Implementation of Functional Languages*, Aachen, September 1992. Available from Herbert Kuchen, Lehrstuhl Informatik II, RWTH Aachen, Ahornstr. 55, W-51000 Aachen. Also Glasgow Functional Programming Workshop 1993.

[Leung and Ting, 1997] Ho-Fung Leung and Hing-Fung Ting. An optimal algorithm for global termination detection in shared-memory asynchronous multiprocessor systems. *IEEE Transactions on Parallel and Distributed Systems*, 8(5):538–543, May 1997.

[Levanoni and Petrank, 1999] Yossi Levanoni and Erez Petrank. A scalable reference counting garbage collector. Technical Report CS–0967, Technion — Israel Institute of Technology, Haifa, Israel, November 1999.

[Levanoni and Petrank, 2001] Yossi Levanoni and Erez Petrank. An on-the-fly reference counting garbage collector for Java. In OOPSLA 2001 [OOPSLA 20012001].

[Levanoni and Petrank, 2006] Yossi Levanoni and Erez Petrank. An on-the-fly reference counting garbage collector for Java. *ACM Transactions on Programming Languages and Systems*, 28(1), January 2006.

[Levelt *et al.*, 1992] Willem G. Levelt, M. Frans Kaashoek, Henri E. Bal, and Andrew Tanenbaum. A comparison of two paradigms for distributed shared memory. *Software Practice and Experience*, 22(11):985–1010, November 1992.

[Leverett and Hibbard, 1982] B. W. Leverett and P. G. Hibbard. An adaptive system for dynamic storage allocation. *Software Practice and Experience*, 12(6):543–556, June 1982.

[Lewis *et al.*, 1974] T. G. Lewis, B. J. Smith, and M. Z. Smith. Dynamic memory allocation systems for minimizing internal fragmentation. In *Proceedings of ACM Annual Conference*, pages 725–728. ACM Press, November 1974.

[Lewis *et al.*, 1998] Bill Lewis, Dan LaLiberte, Richard Stallman, and the GNU Manual Group. *GNU Emacs Lisp Reference Manual*. GNU Project, May 1998.

[Lextrait, 1992] Vincent Lextrait. *Generation de Serveurs de Vues*. PhD thesis, University of Nice, lextrait@hotmail.com, December 1992. (In french).

[LFP 1982, 1982] *Proceedings of the ACM Symposium on Symposium on Lisp and Functional Programming*, Pittsburgh, PA, USA, August 1982.

[LFP 1986, 1986] *Proceedings of the ACM Conference on Symposium on Lisp and Functional Programming*, Cambridge, MA, USA, August 1986.

[LFP 1988, 1988] *Proceedings of the ACM Conference on Symposium on Lisp and Functional Programming*, Snowbird, UT, USA, July 1988. ACM Press.

[LFP 1990, 1990] *Proceedings of the ACM Conference on Symposium on Lisp and Functional Programming*, Nice, France, June 1990.

[LFP 1992, 1992] *Proceedings of the ACM Conference on Symposium on Lisp and Functional Programming*, San Francisco, CA, USA, June 1992.

[LFP 1994, 1994] *Proceedings of the ACM Conference on Symposium on Lisp and Functional Programming*, Orlando, FL, USA, June 1994.

[Lhoták and Hendren, 2002] Ondrej Lhoták and Laurie Hendren. Run-time evaluation of opportunites for object inlining in Java. In *Joint ACM Java Grande - ISCOPE Conference*, pages 175–184, Seattle, WA, 2002. ACM Press.

[Li and Hudak, 1985] Kai Li and Paul Hudak. A new list compaction method. Research Report 362, Yale University, February 1985.

[Li and Hudak, 1986] Kai Li and Paul Hudak. A new list compaction method. *Software Practice and Experience*, 16(2):145–163, February 1986.

[Li and Hudak, 1989] Kai Li and Paul Hudak. Memory coherence in shared virtual memory systems. *ACM Transactions on Computer Systems*, 7(4):321–359, November 1989.

[Li, 1986] Kai Li. *Shared Virtual Memory on Loosely Coupled Multiprocessors*. PhD thesis, Yale University, 1986.

[Li, 1990] Kai Li. Real-time concurrent collection in user mode. In Jul and Juul [Jul and Juul1990].

[Li, 2009] Xiao-Feng Li. Tick: Concurrent gc in Apache Harmony, 2009. Slides on Harmony's Tick GCs.

[Lieberman and Hewitt, 1981] Henry Lieberman and Carl E. Hewitt. A real-time garbage collector based on the lifetimes of objects. AI Memo 569a, MIT, April 1981.

[Lieberman and Hewitt, 1983] Henry Lieberman and Carl E. Hewitt. A real-time garbage collector based on the lifetimes of objects. *Communications of the ACM*, 26(6):419–429, 1983. Also report TM–184, Laboratory for Computer Science, MIT, Cambridge, MA, July 1980 and AI Lab Memo 569, 1981.

[Liekweg, 2006] Florian Liekweg. Compiler-directed automatic memory management. In SPACE 2006 [SPACE 20062006], pages 23–34.

[Lim *et al.*, 1998] Tian F. Lim, Przemyslaw Pardyak, and Brian N. Bershad. A memory-efficient real-time non-copying garbage collector. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 118–129.

[Lin and Chen, 2000] Chi-Min Lin and Tien-Fu Chen. Dynamic memory management for real-time embedded Java chips. In RTCSA 2000 [RTCSA 20002000].

[Lin and Hou, 2006] Chin-Yang Lin and Ting-Wei Hou. A lightweight cyclic reference counting algorithm. In *Proceedings of the International Conference on Grid and Pervasive Computing*, number 3947 in Lecture Notes in Computer Science, pages 246–359. Springer-Verlag, 2006.

[Lin and Hou, 2007] Chin-Yang Lin and Ting-Wei Hou. A simple and efficient algorithm for cycle collection. *ACM SIGPLAN Notices*, 42(3):7–13, March 2007.

[Lin, 1992] Sheng-Lien Lin. Performance evaluation of a generation scavenging algorithm, 1992.

[Lincoln and Mitchell, 1992] Patrick Lincoln and John Mitchell. Operational aspects of linear lambda calculus. In *Proceedings, Seventh Annual IEEE Symposium on Logic in Computer Science*, pages 235–246, Santa Cruz, California, June 1992. IEEE Press.

[Lindholm and O'Keefe, 1987] T. G. Lindholm and R. A. O'Keefe. Efficient implementation of a defensible semantics for dynamic Prolog code. In Lassez [Lassez1987].

[Lindholm and Yellin, 1999] Tim Lindholm and Frank Yellin. *Java Virtual Machine Specification*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[Lindstrom and Soffa, 1981] Gary Lindstrom and Mary Lou Soffa. Referencing and retention in block-structured coroutines. *ACM Transactions on Programming Languages and Systems*, 3(3):263–292, July 1981.

[Lindstrom *et al.*, 1994] Anders Lindstrom, Alan Dearle, Rex di Bona, J. Matthew Farrow, Frans Henskens, John Rosenberg, and Francis Vaughan. A model for user-level memory management in a persisitent distributed environment. In Gopal Gupta, editor, *Proceedings of the Seventeenth Annual Computer Science Conference, ACSC–17, Part B*, pages 343–354, Christchurch, New Zealnad, January 1994.

[Lindstrom, 1973] Gary Lindstrom. Scaning list structures without stacks or tag bits. *Information Processing Letters*, 2(2):47–51, June 1973.

[Lindstrom, 1974] Gary Lindstrom. Copying list structures using bounded workspace. *Communications of the ACM*, 17(4):199–202, April 1974.

[Lins and Dehne, 1994] Rafael D. Lins and Frank Dehne. Distributed reference counting. In *Proceedings of the 1994 Canada–France Conference on Parallel Computing*, number 805 in Lecture Notes in Computer Science, pages 95–100. Springer-Verlag, May 1994.

[Lins and Jones, 1993] Rafael D. Lins and Richard E. Jones. Cyclic weighted reference counting. In K. Boyanov, editor, *Procedings of WP & DP'93 Workshop on Parallel and Distributed Processing*, pages 369–382, Sofia, Bulgaria, May 1993. North Holland. Also Computing Laboratory Technical Report 95, University of Kent, December 1991.

[Lins and Vasques, 1991] Rafael D. Lins and Márcio A. Vasques. A comparative study of algorithms for cyclic reference counting. Technical Report 92, Computing Laboratory, The University of Kent at Canterbury, August 1991.

[Lins *et al.*, 2007] R.D. Lins, F. Heron de Carvalho Junior, and Z.D. Lins. Cyclic reference counting with permanent objects. *Journal of Universal Computer Science*, 13(6):830–838, 2007.

[Lins, 1991] Rafael D. Lins. A shared memory architecture for parallel cyclic reference counting. *Microprocessing and Microprogramming*, 32:53–58, September 1991.

[Lins, 1992a] Rafael D. Lins. Cyclic reference counting with lazy mark-scan. *Information Processing Letters*, 44(4):215–220, 1992. Also Computing Laboratory Technical Report 75, University of Kent, July 1990.

[Lins, 1992b] Rafael D Lins. Generational cyclic reference counting. Technical Report 22-92, Computing Laboratory, University of Kent, September 1992.

[Lins, 1992c] Rafael D. Lins. A multi-processor shared memory architecture for parallel cyclic reference counting. *Microprocessing and Microprogramming*, 35:563–568, September 1992.

[Lins, 2002] Rafael D. Lins. An efficient algorithm for cyclic reference counting. *Information Processing Letters*, 83:145–150, 2002.

[Lins, 2005] R. Lins. A new multiprocessor architecture for parallel lazy cyclic reference counting. In *Proceedings of the 17th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'05)*. IEEE Press, 2005.

[Liskov and Ladin, 1986] Barbara Liskov and Rivka Ladin. Highly available distributed services and fault-tolerant distributed garbage collection. In J. Halpern, editor, *Proceedings of the Fifth Annual ACM Symposium on the Principles on Distributed Computing*, pages 29–39, Calgary, August 1986. ACM Press.

[Llames, 1991] Rene Lim Llames. *Performance Analysis of Garbage Collection and Dynamic Reordering in a Lisp System*. PhD thesis, University of Illinois at Urbana-Champaign, 1991.

[Lo and Chang, 2001] Chia-Tien Dan Lo and J. Morris Chang. A multithreaded concurrent generational garbage collector for Java. In OOPSLA 2001 [OOPSLA 20012001], pages 7–9. Doctoral Symposium.

[Lo *et al.*, 1998] Chia-Tien Dan Lo, Witiwas Srisa-an, and J. Morris Chang. Boundary analysis for buddy systems. In *Proceedings of 1998 International Computer Symposium (Computer Architecture Track)*, pages 96–103, Tainan, Taiwan, December 1998.

[Lo *et al.*, 2000a] Chia-Tien Dan Lo, Witiwas Srisa-an, and J. Morris Chang. Page replacement performance in garbage collection systems. In *Proceedings of 13th International Conference on Parallel and Distributed Computing Systems*, pages 374–379, Las Vegas, NA, August 2000.

[Lo *et al.*, 2000b] Chia-Tien Dan Lo, Witiwas Srisa-an, and J. Morris Chang. A quantitative simulator for dynamic memory managers. In *Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software*, pages 64–69, Austin, TX, April 2000. IEEE Press.

[Lo *et al.*, 2001] Chia-Tien Dan Lo, Witiwas Srisa-an, and J. Morris Chang. A study of page replacement performance in garbage collection heap. *Journal of Systems and Software*, 58(3):235–245, September 2001.

[Lo *et al.*, 2002a] Chia-Tien Dan Lo, J. Morris Chang, Ophir Frieder, and David Grossman. The object behaviour of Java object-oriented database management systems. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'02)*. IEEE Press, 2002.

[Lo *et al.*, 2002b] Chia-Tien Dan Lo, Witiwas Srisa-an, and J. Morris Chang. A high performance garbage collector for Java. In *Proceeding of the 2002 International Computer Symposium, ICS 2002*, National Dong Hwa University, Hualien, Taiwan, December 2002.

[Lo *et al.*, 2002c] Chia-Tien Dan Lo, Witiwas Srisa-an, and J. Morris Chang. A multithreaded concurrent garbage collector which parallelizes the new instruction in Java. In *International Parallel and Distributed Processing Symposium*, pages 59–64, Fort Lauderdale, FL, April 2002.

[Lo *et al.*, 2002d] Chia-Tien Dan Lo, Witiwas Srisa-an, and J. Morris Chang. A performance comparison between stop-the-world and multithreaded concurrent garbage collection for Java. In *21st IEEE International Performance, Computing, And Communications Conference (IPCCC 2002)*, pages 301–308, Phoenix, AZ, April 2002.

[Lo *et al.*, 2003] Chia-Tien Dan Lo, Witiwas Srisa-an, and J. Morris Chang. Who is collecting your Java garbage? *IEEE IT Professional*, 5(2):44–50, April 2003.

[Lo *et al.*, 2004] Chia-Tien Dan Lo, Witiwas Srisa-an, and J. Morris Chang. The design and analysis of a quantitative simulator for dynamic memory management. *Journal of Systems and Software*, 72(3):443–453, August 2004.

[Lo *et al.*, 2005] Chia-Tien Dan Lo, Witiwas Srisa-an, and J. Morris Chang. Security issues in garbage collection. *CrossTalk: The Journal of Defense Software Engineering*, October 2005.

[Loidl, 2000] H-W. Loidl. Investigating the memory management in a parallel graph reducer. In M. Mohnen and P. Koopman, editors, *Proceedings of the 12th International Workshop on Implementation of Functional Languages*, number AIB-00-7 in Aachener Informatik Berichte, pages 185–200. RWTH Aachen, 2000.

[Lomet, 1975] D. B. Lomet. Scheme for invalidating references to freed storage. *IBM Journal of Research and Development*, pages 26–35, January 1975.

[Louboutin and Cahill, 1995a] Sylvain Louboutin and Vinny Cahill. A lazy log-keeping mechanism for comprehensive global garbage detection on Amadeus. In *OOIS (Object-Oriented Information Systems) '95*, pages 118–132, London, December 1995. Springer-Verlag. Technical report TCD–CS–95–11.

[Louboutin and Cahill, 1995b] Sylvain Louboutin and Vinny Cahill. Lazy, per cluster log-keeping mechanism for global garbage detection on Amadeus. Technical Report TCD-CS-95-13, Department of Computer Science, Trinity College, May 1995.

[Louboutin and Cahill, 1995c] Sylvain Louboutin and Vinny Cahill. On comprehensive global garbage detection. In *Proceeding of the European Research Seminar on Advances in Distributed Systems (ERSADS '95)*, pages 208–213, Alpes d'Huez, France, April 1995. INRIA/IMAG. Also technical report TCD-CS-95-11, Department of Computer Science, Trinity College Dublin.

[Louboutin and Cahill, 1997] Sylvain R.Y. Louboutin and Vinny Cahill. Comprehensive distributed garbage collection by tracking causal dependencies of relevant mutator events. In *Proceedings of ICDCS'97 International Conference on Distributed Computing Systems*. IEEE Press, 1997.

[Louboutin, 1998] Sylvain R.Y. Louboutin. *A Reactive Approach to Comprehensive Global Garbage Detection*. PhD thesis, Trinity College, Dublin, 1998.

[Lowry and Munro, 2002] M.C. Lowry and David S. Munro. Safe and complete distributed garbage with the Train algorithm. In *Proceedings of the 2002 International Conference on Parallel and Distributed Systems (ICPADS'2002)*, pages 651–658, Taiwan, December 2002.

[Lowry, 2004] M.C. Lowry. *A New Approach to the Train Algorithm for Distributed Garbage Collection*. PhD thesis, University of Adelaide, December 2004.

[Luc, 1997] Lucent Technologies Inc. *The Limbo Programming Language*, 1997.

[Luk and Mowry, 1996] Chi-Keung Luk and Todd C. Mowry. Compiler-based prefetching for recursive data structures. In ASPLOS 1996 [ASPLOS 19961996], pages 222–233.

[Lynch, 1990] N. A. Lynch. Multivalued possibilities mappings. Technical Memo MIT/LCS/TM–422, MIT Laboratory for Computer Science, August 1990.

[Lyon, 1988] G. Lyon. Tagless marking that is linear over subtrees. *Information Processing Letters*, 27(1):23–28, 1988.

[MacLachlan, 1991] Robert A. MacLachlan. A system model of memory management. In Wilson and Hayes [Wilson and Hayes1991a].

[Madany *et al.*, 1992] Peter W. Madany, Nayeem Islam, Panos Kougiouris, and Roy H. Campbell. Reification and reflection in C++: An operating systems perspective. Technical Report UIUCDCS–R–92–1736, Department of Computer Science, University of Illinois at Urbana-Champaign, March 1992.

[Maebe *et al.*, 2004] Jonas Maebe, Michiel Ronsse, and Koen De Bosschere. The missing leak. In *Program Acceleration through Application and Architecture driven Code Transformations: Symposium Proceedings*, pages 75–77, Edegem, Belgium, 2004.

[Maeda and Ishikawa, 1997] Munenori Maeda and Yutaka Ishikawa. GLEANER-7: A hybrid distributed GC algorithm. In Dickman and Wilson [Dickman and Wilson1997].

[Maeda and Yonezawa, 2006] Toshiyuki Maeda and Akinori Yonezawa. Writing practical memory management code with a strictly typed assembly language. In SPACE 2006 [SPACE 20062006], pages 35–46.

[Maeda *et al.*, 1993] Munenori Maeda, Hiroki Konaka, Yutaka Ishikawa, Takashi Tomokiyo, and Atsushi Hori. An incremental, weighted, cyclic reference counting for object-based languages. RWCP Technical Report P–93–001, Tsukuba Research Center, 1993.

[Maeda *et al.*, 1995] Munenori Maeda, Hiroki Konaka, Yutaka Ishikawa, Takashi Tomokiyo, Atsushi Hori, and Jorg Nolte. On-the-fly global garbage collection based on partly mark-sweep. In Baker [Baker1995a].

[Maeder, 1992] Roman E. Maeder. A provably correct reference count scheme for a symbolic computation system. In unpublished form, cited by Edelson, 1992.

[Maekawa *et al.*, 1987] Mamoru Maekawa, Masataka Ohta, and Kentaro Shimizu. Garbage collection for multimedia processing. Technical Report 87-01, University of Tokyo, 1987.

[Magill *et al.*, 2006] Stephen Magill, Aleksandar Nanevski, Edmund Clarke, and Peter Lee. Inferring invariants in separation logic for imperative list-processing programs. In SPACE 2006 [SPACE 20062006], pages 47–60.

[Magnusson and Henriksson, 1995a] Boris Magnusson and Roger Henriksson. Garbage collection for control systems. In Baker [Baker1995a].

[Magnusson and Henriksson, 1995b] Boris Magnusson and Roger Henriksson. Garbage collection for hard real-time systems. Technical Report 95–153, Lund University, Sweden, 1995.

[Maher, 1961] R. J. Maher. Problems of storage allocation in a multiprocessor multiprogrammed system. *Communications of the ACM*, 4(10):421–422, October 1961.

[Maheshwari and Liskov, 1995] Umesh Maheshwari and Barbara Liskov. Collecting cyclic distributed garbage by controlled migration. In *Proceedings of the 14th ACM Symposium on Principles of Distributed Computing (PODC'95)*, 1995. Later appeared in Distributed Computing, Springer Verlag, 1996.

[Maheshwari and Liskov, 1997a] Umesh Maheshwari and Barbara Liskov. Collecting cyclic distributed garbage by back tracing. In *Proceedings of PODC'97 Principles of Distributed Computing*, pages 239–248, Santa Barbara, CA, 1997. ACM Press.

[Maheshwari and Liskov, 1997b] Umesh Maheshwari and Barbara Liskov. Partitioned garbage collection of a large object store. Technical Report MIT/LCS/TR–699, MIT Press, 1997. This report contains an obsolete proposal. See [Maheshwari and Liskov, 1997c].

[Maheshwari and Liskov, 1997c] Umesh Maheshwari and Barbara Liskov. Partitioned garbage collection of a large object store. In *Proceedings of SIGMOD'97*, 1997.

[Maheshwari, 1992] Umesh Maheshwari. Distributed garbage collection in a client–server transaction system. Master's thesis, Department of Electrical Engineering and Computer Science, Massachuset ts Institute of Technology, 1992.

[Maheshwari, 1993a] Umesh Maheshwari. Distributed garbage collection in a client–server persistent object system. In Moss et al. [Moss *et al.*1993].

[Maheshwari, 1993b] Umesh Maheshwari. Distributed garbage collection in a client–server, transactional, persistent object system. Technical Report MIT/LCS/TR–574, MIT Press, February 1993.

[Maheshwari, 1994] Umesh Maheshwari. Fault-tolerant distributed garbage collection in a client-server object-oriented database. In *Third International Conference on Parallel and Distributed Information Systems, Austin*, September 1994.

[Maheshwari, 1997] Umesh Maheshwari. *Garbage Collection in a Large, Distributed, Object Store*. PhD thesis, MIT Laboratory for Computer Science, September 1997. Technical Report MIT/LCS/TR-727.

[Maisonneuve *et al.*, 1992] Julien Maisonneuve, Marc Shapiro, and Pierre Collet. Implementing references as chains of links. In Cabrera et al. [Cabrera *et al.*1992].

[Maisonneuve, 1996] Julien Maisonneuve. *Hobbes: un modèle de liaison de références réparties*. PhD thesis, Université Paris 6, Pierre et Marie Curie, October 1996.

[Makholm and Niss, 2001] Henning Makholm and Henning Niss. Towards a more flexible region type system. In SPACE 2001 [SPACE 20012001].

[Makholm, 2000] Henning Makholm. A region-based memory manager for Prolog. In Chambers and Hosking [Chambers and Hosking2000], pages 25–34.

[Makpangou *et al.*, 1992] Mesaac Makpangou, Yvon Gourhant, Jean-Pierre Le Narzul, and Marc Shapiro. Structuring distributed applications as fragmented objects. Technical Report Rapport de Recherche INRIA 1404, INRIA, nelly@sor.inria.fr, anonymous FTP nuri.inria.fr [128.93.1.26], February 1992.

[Mancini and Shrivastava, 1991] Luigi V. Mancini and S. K. Shrivastava. Fault-tolerant reference counting for garbage collection in distributed systems. *Computer Journal*, 34(6):503–513, December 1991.

[Mancini *et al.*, 1991] Luigi V. Mancini, Vittoria Rotella, and Simonetta Venosa. Copying garbage collection for distributed object stores. In *Proceedings ot the Tenth Symposium on Reliable Distributed Systems, Pisa*, September 1991.

[Mann *et al.*, 2005] Tobias Mann, Morgan Deters, Rob Legrand, and Ron. K. Cytron. Static determination of allocation rates to support real-time garbage collection. In LCTES 2005 [LCTES 20052005], pages 193–2002.

[Manson *et al.*, 2005] Jeremy Manson, William Pugh, and Sarita V. Adve. The Java memory model. In POPL 2005 [POPL 20052005], pages 378–391.

[Maranget, 1991] L. Maranget. GAML: A parallel implementation of lazy ML. In Hughes [Hughes1991a], pages 102–123.

[Margolin *et al.*, 1971] B. H. Margolin, R. P. Parmelee, and M. Schatzoff. Analysis of free-storage algorithms. *IBM Systems Journal*, 10(4):283–304, 1971.

[Marion *et al.*, 2007] Sebastien Marion, Richard Jones, and Chris Ryder. Decrypting the Java gene pool: Predicting objects' lifetimes with micro-patterns. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 67–78.

[Markatos and Chronaki, 1994] Evangelos P. Markatos and Catherine E. Chronaki. The use of reference counters in update and invalidate based coherent memory. In PARLE94 [PARLE941994].

[Marlow *et al.*, 2008] Simon Marlow, Tim Harris, Roshan James, and Simon Peyton Jones. Parallel generational-copying garbage collection with a block-structured heap. In Jones and Blackburn [Jones and Blackburn2008], pages 11–20.

[Marlow *et al.*, 2009] Simon Marlow, Simon Peyton Jones, and Satnam Singh. Runtime support for multicore Haskell. In *Proceedings of the 14th ACM SIGPLANInternational Conference on Functional Programming*, pages 65–78, 2009.

[Marques and Guedes, 1989] José Alves Marques and Paulo Guedes. Extending the operating system to support an object-oriented environment. In OOPSLA 1989 [OOPSLA 19891989], pages 113–122.

[Marques *et al.*, 1990] José Alves Marques, Paulo Guedes, Pedro Sousa, Paulo Ferreira, José Monge, André Zúquete, and Manuel Sequeira. IK implementation report. Technical Report INESC–TR–0013, INESC, Portugal, July 1990. Extended description of system described in [Ferreira, 1990].

[Marquet and Grimaud, 2007] Kevin Marquet and Gilles Grimaud. A DSL approach for object memory management of small devices. In PPPJ 2007 [PPPJ 20072007], pages 155–164.

[Marquez *et al.*, 2000] A. Marquez, J. N. Zigman, and S. M Blackburn. Fast portable orthogonally persistent Java. *Software Practice and Experience*, 30(4):449–479, 2000.

[Marron *et al.*, 2009] Mark Marron, Deepak Kapur, and Manuel Hermenegildo. Identification of logically related heap regions. In Kolodner and Steele [Kolodner and Steele2009], pages 89–98.

[Marshall, 1971] S. Marshall. An Algol–68 garbage collector. In Peck [Peck1971], pages 239–243.

[Marti *et al.*, 2006] Nicolas Marti, Reynald Affeldt, and Akinori Yonezawa. Verification of the heap manager of an operating system using separation logic. In SPACE 2006 [SPACE 20062006], pages 61–72.

[Martin, 1982] Johannes J. Martin. An efficient garbage compaction algorithm. *Communications of the ACM*, 25(8):571–581, August 1982.

[Martinez *et al.*, 1990] A. D. Martinez, R. Wachenchauzer, and Rafael D. Lins. Cyclic reference counting with local mark-scan. *Information Processing Letters*, 34:31–35, 1990.

[Masmano *et al.*, 2006] Miguel Masmano, Ismael Ripoll, and Alfons Crespo. A comparison of memory allocators for real-time applications. In *Proceedings of the Fourth International Workshop on Java Technologies for Real-time and Embedded Systems*, pages 68–76, 2006.

[Mateu, 1992] L. Mateu. Efficient implementation for coroutines. In Bekkers and Cohen [Bekkers and Cohen1992].

[Matocha and Camp, 1998] Jeff Matocha and Tracy Camp. A taxonomy of distributed termination detection algorithms. *Journal of Systems and Software*, 43(3):207–221, November 1998.

[Matsui *et al.*, 1987] Shogo Matsui, Yoshinobu Kato, Shinsuke Teramura, Tomoyuki Tanaka, Nobuyuki Mohri, Atsushi Maeda, and Masakazu Nakanishi. SYNAPSE — a multimicroprocessor Lisp machine with parallel garbage collector. *Lecture Notes in Computer Science*, 269:131–137, 1987.

[Matsui *et al.*, 1995] Shogo Matsui, Yoshio Tanaka, Atsushi Maeda, and Masakazu Nakanishi. Complementary garbage collector. In Baker [Baker1995a].

[Matsuoka *et al.*, 1991] Satoshi Matsuoka, Shin'ichi Furuso, and Akinori Yonezawa. A fast parallel conservative garbage collector for concurrent object-oriented systems. In Cabrera et al. [Cabrera *et al.*1991], pages 87–93.

[Mattern, 1987] Friedmann Mattern. Algorithms for distributed termination detection. *Distributed Computing*, 2:161–175, 1987.

[Mattern, 1989a] Friedmann Mattern. Global quiescence detection based on credit distribution and recovery. *Information Processing Letters*, 30(4):195–200, 1989.

[Mattern, 1989b] Friedmann Mattern. Virtual time and global states of distributed systems. In M. Cosnard et al., editors, *International Workshop on Parallel and Distributed Algorithms*, pages 215–226, Amsterdam, 1989. Elsevier Science Publishers.

[Matthews and Le Sergent, 1995] David C. J. Matthews and Thierry Le Sergent. LEMMA: A distributed shared memory with global and local garbage coll ection. In Baker [Baker1995a].

[Mattson *et al.*, 1970] R. L. Mattson, J. Gecsei, D. R. Slutz, and I. L. Traiger. Evaluation techniques for storage hierarchies. *IBM Systems Journal*, 9:78–117, 1970.

[Mazur *et al.*, 2000] Nancy Mazur, Gerda Janssens, and Maurice Bruynooghe. A module based analysis for memory reuse in Mercury. In *Computational Logic*, pages 1255–1269, 2000.

[Mazur *et al.*, 2001] Nancy Mazur, Peter Ross, Gerda Janssens, and Maurice Bruynooghe. Practical aspects for a working compile time garbage collection system for Mercury. In *International Conference on Logic Programming*, pages 105–119, 2001.

[Mazur, 2001] Nancy Mazur. Practical structure reuse for Mercury. In SPACE 2001 [SPACE 20012001].

[McBeth, 1963] J. Harold McBeth. On the reference counter method. *Communications of the ACM*, 6(9):575, September 1963.

[McCarthy and Minsky, 1959] John McCarthy and Marvin Minsky. Artificial intelligence, quarterly progress report no. 53. Technical report, Research Laboratory of Electronics at MIT, April 1959.

[McCarthy and others, 1962] John McCarthy et al. *Lisp 1.5 Programmer's Manual*, 1962.

[McCarthy, 1960] John McCarthy. Recursive functions of symbolic expressions and their computation by machine, Part I. *Communications of the ACM*, 3(4):184–195, April 1960.

[McCarthy, 1978] John McCarthy. History of LISP. In Richard L. Wexelblat, editor, *History of Programming Languages I*, pages 173–185. ACM Press, 1978.

[McCloskey *et al.*, 2008] Bill McCloskey, David F. Bacon, Perry Cheng, and David Grove. Staccato: A parallel and concurrent real-time compacting garbage collector for multiprocessors. IBM Research Report RC24505, IBM Research, 2008.

[McCreight *et al.*, 2007] Andrew McCreight, Zhong Shao, Chunxiao Lin, and Long Li. A general framework for certifying garbage collectors and their mutators. In Ferrante and McKinley [Ferrante and McKinley2007], pages 468–479.

[McCreight *et al.*, 2010] Andrew McCreight, Tim Chevalier, and Andrew Tolmach. A certified framework for compiling and executing garbage-collected languages. In ICFP 2010 [ICFP 20102010], pages 273–284.

[McCullough, 1983] P. L. McCullough. Implementing the Smalltalk-80 system: the Tektronix experience. In Krasner [Krasner1983], pages 59–78.

[McDowell, 1998] Charles E. McDowell. Reducing garbage in Java. *ACM SIGPLAN Notices*, 33(9):84–86, September 1998.

[McEntee, 1987] T. J. McEntee. Overview of garbage collection in symbolic computing. *LISP Pointers*, 1(3):8–16, August–September 1987.

[McGachey and Hosking, 2006] Phil McGachey and Antony L Hosking. Reducing generational copy reserve overhead with fallback compaction. In Petrank and Moss [Petrank and Moss2006], pages 17–28.

[McGaughey, 1995] Mike McGaughey. Bounded-space tagless garbage collection for first order polymorphic languages. In *Proceedings of the Eighteenth Australian Computer Science Conference (ACSC '95)*, Australian Computer Science Communications 17(1), pages 380–388, Glenelg, South Australia, January 1995. Also appears as: Technical report 94/208, Department of Computer Science, Monash University.

[McIlroy and Sventek, 2010] Ross McIlroy and Joe Sventek. Hera-jvm: a runtime system for heterogeneous multi-core architectures. In OOPSLA 2010 [OOPSLA 20102010], pages 205–222.

[McIlroy *et al.*, 2008] Ross McIlroy, Peter Dickman, and Joe Sventek. Efficient dynamic heap allocation of scratch-pad memory. In Jones and Blackburn [Jones and Blackburn2008], pages 31–40.

[McIlroy, 1976] M. Douglas McIlroy. Mass-produced software components. In J. M. Buxton, Peter Naur, and Brian Randell, editors, *Software Engineering Concepts and Techniques (1968 NATO Conference of Software Engineering)*, pages 88–98, 1976.

[McIlroy, 1982] M. Douglas McIlroy. The number of states of a dynamic storage allocation system. *Computer Journal*, 25(3):388–392, August 1982.

[McIver and King, 1994] William J. McIver and Roger King. Self-adaptive, on-line reclustering of complex object data. In *SIGMOD94*, pages 407–418, 1994.

[McKenney and Slingwine, 1993] Paul E. McKenney and Jack Slingwine. Efficient kernel memory allocation on shared-memory multiprocessors. In *USENIX 1993 Winter Technical Conference*. USENIX Association, 1993.

[McKenney and Slingwine, 1998] Paul E. McKenney and Jack Slingwine. Read-copy update: Using execution history to solve concurrency problems. In *Proceedings of the 10th IASTED International Conference on Parallel and Distributed Computing and Systems*, October 1998.

[McKusick and Karels, 1988] Marshall K. McKusick and Michael J. Karels. Design of a general-purpose memory allocator for the 4.3BSD UNIX kernel. In *Proceedings of the Summer 1988 USENIX Conference*. USENIX Association, June 1988.

[McNally and Davie, 1991] D. J. McNally and Anthony J. T. Davie. 2 models for integrating persistence and lazy functional languages. *ACM SIGPLAN Notices*, 26(5):43–52, 1991.

[Meersman *et al.*, 2004] R. Meersman, Z. Tari, and A. Corsaro, editors. *Proceedings of the 2nd International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES)*, volume 3292 of *Lecture Notes in Computer Science*, Heidelburg, Germany, October 2004. Springer.

[Meira, 1985] Silvio R. de L. Meira. *On the Efficiency of Applicative Algorithms*. PhD thesis, Computing Laboratory, The University of Kent at Canterbury, March 1985.

[Mellish, 1980] C. S. Mellish. An alternative to structure-sharing in the implementation of a Prolog interpreter. In *Workshop on Logic Programming, Debrecen, Hungary*, 1980.

[Meloan, 1999] Steve Meloan. The Java HotSpot performance engine: An in-depth look. Article on Sun's Java Developer Connection site, 1999.

[Mendelson *et al.*, 1993] Abraham Mendelson, Dominique Thiebaut, and Dhiraj K. Pradhan. Modeling live and dead lines in cache memory systems. *IEEE Transactions on Computers*, 42(1):1–14, January 1993.

[Menezes and Wood, 1997] Ronaldo Menezes and Alan Wood. Garbage collection in open distributed tuple space systems. In *15th Brazilian Computer Networks Symposium — SBRC '97*, pages 525–543, May 1997.

[Menezes and Wood, 1998a] Ronaldo Menezes and Alan Wood. Ligia: A Java based Linda-like runtime system with garbage collection of tuple spaces. Technical Report YCS 304 (1998), University of York, 1998.

[Menezes and Wood, 1998b] Ronaldo Menezes and Alan Wood. Using tuple monitoring and process registration on the implementation of garbage collection in open Linda-like systems. In *Proceedings of the Tenth IASTED International Conference: PDCS'98*, pages 490–495, Las Vegas, October 1998. ASTED/Acta Press.

[Menezes, 1998] Ronaldo Menezes. Ligia: Incorporating garbage collection in a Java based Linda-like run-time system. In *Proceedings of the 2nd Workshop on Distributed Systems (WOSID'98)*, pages 81–88, Curitiba, Parana, Brazil, 1998.

[Merrall and Padget, 1992] S. C. Merrall and J. A. Padget. Collections and garbage collection. In Bekkers and Cohen [Bekkers and Cohen1992].

[Méry, 1995] Dominique Méry. Refining solutions of the on the fly garbage collection from formal specifications, November 1995.

[Metropolis *et al.*, 1980] N. Metropolis, J. Howlett, and Gian-Carlo Rota, editors. *A History of Computing in the Twentieth Century*. Academic Press, 1980.

[Meyer, 1988] Bertrand Meyer. *Object-oriented Software Construction*. Prentice-Hall, 1988.

[Meyer, 1996] Bertrand Meyer. The ISE Eiffel garbage collection mechanism: An overview. Technical report, ISE Inc., 1996.

[Meyer, 2003] Matthias Meyer. A novel processor architecture with exact tag-free pointers. In *2nd Workshop on Application Specific Processors*, pages 96–103, San Diego, CA, 2003.

[Meyer, 2004] Matthias Meyer. A novel processor architecture with exact tag-free pointers. *IEEE Micro*, 24(3):46–55, May–June 2004.

[Meyer, 2005] Matthias Meyer. An on-chip garbage collection coprocessor for embedded real-time systems. In *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 517–524, Hong Kong, China, August 2005.

[Meyer, 2006] Matthias Meyer. A true hardware read barrier. In Petrank and Moss [Petrank and Moss2006], pages 3–16.

[Meyers and Casseres, 1983] R. Meyers and D. Casseres. An MC68000-based Smalltalk-80 system. In Krasner [Krasner1983], pages 175–188.

[Michael and Scott, 1995] Maged M. Michael and M.L. Scott. Correction of a memory management method for lock-free data structures. Technical report, University of Rochester, 1995.

[Michael, 2002a] Maged M. Michael. Safe memory reclamation for dynamic lock-free objects using atomic reads and writes. Research Report RC22317, IBM Corp., Thomas J Watson Research Center, Yorktown Heights, NY, January 2002.

[Michael, 2002b] Maged M. Michael. Safe memory reclamation for dynamic lock-free objects using atomic reads and writes. In *The 21st Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 21–30, July 2002.

[Michael, 2004a] Maged M. Michael. Hazard pointers: Safe memory reclamation for lock-free objects. *IEEE Transactions on Parallel and Distributed Systems*, 15(6):491–504, June 2004.

[Michael, 2004b] Maged M. Michael. Scalable lock-free dynamic memory allocation. In Pugh and Chambers [Pugh and Chambers2004], pages 35–46.

[Microsoft RNI, 1997, 1997] *Raw Native Interface*, 1997. Microsoft's Raw Native Interface for Java.

[Mijajlovic and Torp-Smith, 2004] Ivana Mijajlovic and Noah Torp-Smith. Refinement in separation context. In SPACE 2004 [SPACE 20042004].

[Mikheev and Fedoseev, 2001] V.V. Mikheev and S.A. Fedoseev. Compiler-cooperative memory management in Java. In *Proceedings of the Andrei Ershov Fourth International Conference "Perspectives of System Informatics"*, volume 2244 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.

[Miller and Epstein, 1990] James S. Miller and B. Epstein. Garbage collection in MultiScheme. In *US/Japan Workshop on Parallel Lisp, LNCS 441*, pages 138–160, June 1990.

[Miller and Rozas, 1994] James S. Miller and Guillermo J. Rozas. Garbage collection is fast, but a stack is faster. Technical Report AIM-1462, MIT AI Laboratory, March 1994.

[Miller, 1987] James S. Miller. *MultiScheme: A Parallel Processing System Based on MIT Scheme*. PhD thesis, MIT Press, 1987. Also Technical Report MIT/LCS/402.

[Miller, 1988] B. P. Miller. The frequency of dynamic pointer references in 'C' programs. Technical Report TR 759, University of Wisconsin, Madison, 1988.

[Miller, 1996] Justin Miller. Clean up: C++ garbage collection. *BYTE*, pages 157–158, January 1996.

[Mills Strout *et al.*, 1998] Michelle Mills Strout, Larry Carter, Jeanne Ferrante, and Beth Simon. Schedule-independent storage mapping for loops. In ASPLOS 1998 [ASPLOS 19981998], pages 24–33.

[Milner *et al.*, 1990] Robin Milner, Mads Tofte, and Robert Harper. *The Definition of Standard ML*. MIT Press, 1990.

[Minsky, 1963] Marvin L. Minsky. A Lisp garbage collector algorithm using serial secondary storage. Technical Report Memo 58 (rev.), Project MAC, MIT, Cambridge, MA, December 1963.

[Miranda, 1977] Eliot E. Miranda. Brouhaha — a portable Smalltalk interpreter. *ACM SIGPLAN Notices*, 22(12):354–365, 1977.

[Miranda, 1987] Eliot Miranda. BrouHaHa — a portable Smalltalk interpreter. In OOPSLA 1987 [OOPSLA 19871987], pages 354–365.

[Misra and Chandy, 1982] Jayadev Misra and K. M. Chandy. Termination detection of diffusing computations in communicating sequential processes. *ACM Transactions on Programming Languages and Systems*, 4(1):37–43, January 1982.

[Misra, 1986] Jayadev Misra. Axioms for memory access in asynchronous hardware systems. *ACM Transactions on Programming Languages and Systems*, 8(1):142–153, January 1986.

[Mitchell and Sevitsky, 2003] Nick Mitchell and Gary Sevitsky. LeakBot: An automated and lightweight tool for diagnosing memory leaks in large java applications. In *Proceedings of European Conference on Object-Oriented Programming, ECOOP 2003*, pages 351–377, Darmstadt, Germany, June 2003.

[Mitchell *et al.*, 2009] Nick Mitchell, Edith Schonberg, and Gary Sevitsky. Making sense of large heaps. In *Proceedings of 23rd European Conference on Object-Oriented Programming (ECOOP'09)*, Genova, Italy, July 2009. Springer-Verlag.

[Mittal *et al.*, 1986] S. Mittal, Daniel Bobrow, and K. Kahn. Virtual copies: At the boundary between classes and instances. In OOPSLA 1986 [OOPSLA 19861986], pages 159–166.

[Miyauchi *et al.*, 1987] Nobuhito Miyauchi, A. Matsumoto, Y. Kimura, and A. Goto. Multiple reference management by MRB — GC characteristics on KL1 emulator. In *35th Meeting of Information Processing Society*, September 1987. In Japanese.

[Miyauchi *et al.*, 1989] Nobuhito Miyauchi, Yasuharu Kawada, and Katsuto Nakajima. Tracing garbage collection for KL1 on the Multi-PSI/V2 system. ICOT technical report TR-469, Institute for New Generation Computer Technology, March 1989.

[Mogul and Borg, 1991] Jeffrey C. Mogul and Anita Borg. The effect of context switches on cache performance. In ASPLOS 1991 [ASPLOS 19911991], pages 75–84.

[Mohamed-Ali and Haridi, 1986] Kharyi A. Mohamed-Ali and Seif Haridi. Global garbage collection for distributed heap storage-systems. *International Journal Of Parallel Programming*, 15(5):339–387, 1986.

[Mohamed-Ali, 1984] Khayri A. Mohamed-Ali. *Object Oriented Storage Management and Garbage Collection in Distributed Processing Systems*. PhD thesis, Royal Institute of Technology, Stockholm, December 1984.

[Mohamed-Ali, 1989] Kharyi A. Mohamed-Ali. Garbage collection for Or-parallel Prolog based on WAM. In *Proceedings of the Gigalips Workshop, Stockholm*. SICS, April 1989.

[Mohnen, 1995a] Markus Mohnen. Efficient compile-time garbage collection for arbitrary data structures. Technical Report 95–08, University of Aachen, May 1995. Also in Seventh International Symposium on Programming Languages, Implementations, Logics and Programs, PLILP95.

[Mohnen, 1995b] Markus Mohnen. Efficient compile-time garbage collection for arbitrary data structures. In *Seventh International Symposium on Programming Languages, Implementations, Logics and Programs, PLILP95*, pages 241–258, 1995. Also see [Mohnen, 1995a].

[Moller, 2001] Anders Moller. Verification of data type implementations using graph types and monadic second-order logic. In SPACE 2001 [SPACE 20012001].

[Monnier *et al.*, 2001] Stefan Monnier, Bratin Saha, and Zhong Shao. Principled scavenging. In PLDI 2001 [PLDI 20012001], pages 81–91.

[Monnier, 2001] Stefan Monnier. Principled scavenging. In SPACE 2001 [SPACE 20012001].

[Monnier, 2004] Stefan Monnier. Typed regions. In SPACE 2004 [SPACE 20042004].

[Moon, 1974] David A. Moon. *MACLisp reference manual*. Project MAC, MIT, Cambridge, MA, April 1974.

[Moon, 1984] David A. Moon. Garbage collection in a large LISP system. In Steele [Steele1984], pages 235–245.

[Moon, 1985] David A. Moon. Architecture of the Symbolics 3600. In *Proceedings of the 12th Annual International Symposium on Computer Architecture*, pages 76–83, Boston, MA, June 1985.

[Moon, 1990] David A. Moon. *Symbolics Architecture*, chapter 3. Wiley, 1990.

[Moon, 1991] David A. Moon. Genera retrospective. In Cabrera et al. [Cabrera *et al.*1991]. Order Number 2265.

[Moon, 1992] David A. Moon. Copying garbage collection is harmful. In Bekkers and Cohen [Bekkers and Cohen1992].

[Moore *et al.*, 1997] Jonathan Moore, Mike Hicks, and Scott Nettles. Oscar: A GC testbed. In Dickman and Wilson [Dickman and Wilson1997].

[Morad *et al.*, 2009] Ronny Morad, Martin Hirzel, Elliot K. Kolodner, and Mooly Sagiv. Efficient memory management for long-lived objects. IBM Research Report RC24794, IBM T.J. Watson Research Center, May 2009.

[Moreau and Duprat, 1999] Luc Moreau and Jean Duprat. A construction of distributed reference counting: the constructive proof in Coq, February 1999.

[Moreau and Duprat, 2001] Luc Moreau and Jean Duprat. A construction of distributed reference counting. *Acta Informatica*, pages 563–595, 2001.

[Moreau and Zendra, 2002] P.-E. Moreau and Olivier Zendra. GC^2: A generational conservative garbage collector for the ATerm library. Rapport de Recherche RR–4547, INRIA Lorraine, September 2002.

[Moreau and Zendra, 2004] P.-E. Moreau and Olivier Zendra. GC^2: A generational conservative garbage collector for the ATerm library. *Journal of Logic and Algebraic Programming (JLAP)*, 59(1–2), April 2004.

[Moreau *et al.*, 1997] Luc Moreau, David DeRoure, and Ian Foster. NeXeme: a distributed Scheme based on Nexus. In *Third International Europar Conference (EURO-PAR'97)*, volume 1300 of *Lecture Notes in Computer Science*, pages 581–590, Passau, Germany, August 1997. Springer-Verlag.

[Moreau *et al.*, 2003] Luc Moreau, Peter Dickman, and Richard Jones. Birrell's distributed reference listing revisited. Technical Report 8–03, University of Kent, July 2003.

[Moreau *et al.*, 2005] Luc Moreau, Peter Dickman, and Richard Jones. Birrell's distributed reference listing revisited. *ACM Transactions on Programming Languages and Systems*, 27(6):1344–1395, 2005.

[Moreau, 1997] Luc Moreau. A distributed garbage collector with diffusion tree reorganisation and object mobility. Technical Report M97/2, University of Southampton, October 1997.

[Moreau, 1998a] Luc Moreau. A distributed garbage collector with diffusion tree reorganisation and mobile objects. In ICFP 1998 [ICFP 19981998], pages 204–215.

[Moreau, 1998b] Luc Moreau. Hierarchical distributed reference counting. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 57–67.

[Moreau, 1999] Luc Moreau. Implementation and performance evaluation of a distributed garbage collection algorithm. In Takayasu Ito and Taiichi Yuasa, editors, *Parallel and Distributed Computing for Symbolic and Irregular Applications, PDCSIA'99*, pages 221–241, Sendai, Japan, July 1999. World Scientific Publishing.

[Moreau, 2001] Luc Moreau. Tree rerooting in distributed garbage collection: Implementation and performance evaluation. *Higher-Order and Symbolic Computation*, 14(4), 2001.

[Morris, 1978] F. Lockwood Morris. A time- and space-efficient garbage compaction algorithm. *Communications of the ACM*, 21(8):662–5, 1978.

[Morris, 1979] F. Lockwood Morris. On a comparison of garbage collection techniques. *Communications of the ACM*, 22(10):571, October 1979.

[Morris, 1982] F. Lockwood Morris. Another compacting garbage collector. *Information Processing Letters*, 15(4):139–142, October 1982.

[Morrisett and Harper, 1997] Greg Morrisett and Robert Harper. Semantics of memory management for polymorphic languages. In A. Gordon and A. Pitts, editors, *Higher Order Operational Techniques in Semantics*, pages 175–226. Newton Institute, Cambridge University Press, 1997.

[Morrisett and Sagiv, 2007] Greg Morrisett and Mooly Sagiv, editors. *Proceedings of the Sixth International Symposium on Memory Management*, Montréal, Canada, October 2007. ACM Press.

[Morrisett *et al.*, 1995a] J. Gregory Morrisett, Mattias Felleisen, and Robert Harper. Abstract models of memory management. Technical Report CMU–CS–95–110, Carnegie Mellon University, January 1995. Also published as Fox memorandum CMU–CS–FOX–95–01.

[Morrisett *et al.*, 1995b] J. Gregory Morrisett, Mattias Felleisen, and Robert Harper. Abstract models of memory management. In FPCA 1995 [FPCA 19951995].

[Morrisett, 1995] J. Gregory Morrisett. *Compiling with Types*. PhD thesis, Carnegie Mellon University, December 1995. Published as CMU Technical Report CMU-CS-95-226.

[Morrisett, 2001] Greg Morrisett. Next generation low-level languages. In SPACE 2001 [SPACE 20012001]. Invited talk.

[Morrison *et al.*, 1999] Ronald Morrison, Mick J. Jordan, and Malcolm P. Atkinson, editors. *Proceedings of the EighthInternational Workshop on Persistent Object Systems (August, 1998)*, Advances in Persistent Object Systems, Tiburon, CA, USA, 1999. Morgan Kaufmann.

[Moss and Sinofsky, 1988] J. Eliot B. Moss and S. Sinofsky. Managing persistent data with Mneme: Designing a reliable, shared object interface. In *Advances in Object-oriented Database Systems*, volume 334 of *Lecture Notes in Computer Science*, pages 298–316. Springer-Verlag, 1988.

[Moss *et al.*, 1993] Eliot Moss, Paul R. Wilson, and Benjamin Zorn, editors. *OOPSLA Workshop on Garbage Collection in Object-Oriented Systems*, October 1993.

[Moss *et al.*, 1997] J. Eliot B. Moss, David S. Munro, and Richard L. Hudson. PMOS: A complete and coarse-grained incremental garbage collector for persistent object stores. In Connor and Nettles [Connor and Nettles1997], pages 140–150.

[Moss, 1989a] J. Eliot B. Moss. Addressing large distributed collections of persistent objects: The Mneme project's approach. In *Second International Workshop on Database Programming Languages*, pages 269–285, Glenedon Beach, OR, June 1989. Also available as Technical Report 89-68, University of Massachusetts Department of Computer and Information Science, Amherst, MA, 1989.

[Moss, 1989b] J. Eliot B. Moss. The Mneme persistent object store. COINS Technical Report 89–107, University of Massachusetts, Department of Computer and Information Science, 1989.

[Moss, 1990a] J. Eliot B. Moss. Abstract data types in stack based languages. Technical Memo MIT/LCS/TR–190, MIT Laboratory for Computer Science, August 1990.

[Moss, 1990b] J. Eliot B. Moss. Garbage collecting persistent object stores. In Jul and Juul [Jul and Juul1990]. Also in SIGPLAN Notices 23(1):45–52, January 1991.

[Moss, 1990c] J. Eliot B. Moss. Working with objects: To swizzle or not to swizzle? Technical Report 90–38, University of Massachusetts, Amherst, MA, May 1990.

[Moss, 1991] J. Eliot B. Moss. The UMass language independent garbage collector toolkit. In Wilson and Hayes [Wilson and Hayes1991a].

[Moss, 1992] J. Eliot B. Moss. Working with persistent objects: To swizzle or not to swizzle? *IEEE Transactions on Software Engineering*, 18(8):657–673, August 1992.

[Mowry *et al.*, 1992] Todd C. Mowry, Monica S. Lam, and Anoop Gupta. Design and evaluation of a compiler algorithm for prefetching. In ASPLOS 1992 [ASPLOS 19921992], pages 62–73.

[MSP 2002, 2003] *Proceedings of the Workshop on Memory System Performance (June, 2002)*, ACM SIGPLAN Notices 38(2 supplement), Berlin, Germany, February 2003.

[Mukherjee and McKinley, 2004] Shubu Mukherjee and Kathryn S. McKinley, editors. *Proceedings of the Eleventh International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGPLAN Notices 39(11), Boston, MA, USA, October 2004.

[Mulkers *et al.*, 1990a] Anne Mulkers, William Winsborough, and Maurice Bruynooghe. Analysis of shared data structures for compile-time garbage collection in logic programs. In *7th International Conference on Logic Programming*, pages 747–762. MIT Press, 1990.

[Mulkers *et al.*, 1990b] Anne Mulkers, William Winsborough, and Maurice Bruynooghe. Analysis of shared data structures for compile-time garbage collection in logic programs. Report CW117, Katholieke Universitiet of Leuven, Belgium, 1990. Extended version.

[Mulkers *et al.*, 1992] Anne Mulkers, William Winsborough, and Maurice Bruynooghe. Static analysis of logic programs to detect run-time garbage cells. In *International Conference on Computer Systems and Software Engineering*, pages 526–531. IEEE Press, 1992.

[Mulkers *et al.*, 1993] Anne Mulkers, William Winsborough, and Maurice Bruynooghe. A live-structure data-flow analysis for Prolog. Theory Reort CW167, Katholieke Universitiet of Leuven, Belgium, 1993.

[Mulkers *et al.*, 1994] Anne Mulkers, William Winsborough, and Maurice Bruynooghe. Live-structure dataflow analysis for Prolog. *ACM Transactions on Programming Languages and Systems*, 16(2), March 1994.

[Mulkers, 1991] Anne Mulkers. *Deriving Live Data Structures in Logic Programs by Means of Abstract Interpretation*. PhD thesis, Katholieke Universitiet of Leuven, Belgium, 1991.

[Mulkers, 1993] Anne Mulkers. *Live Data Structures in Logic Programs*. Number 675 in Lecture Notes in Computer Science. Springer-Verlag, 1993.

[Muller *et al.*, 1992] H. L. Muller, K. G. Langendoen, and L. O. Hertzberger. MiG: Simulating parallel functional programs on hierarchical cache architectures. Technical Report CS–92–04, Department of Computer Science, University of Amsterdam, June 1992.

[Müller, 1976] Klaus A. G. Müller. *On the Feasibility of Concurrent Garbage Collection*. PhD thesis, Tech. Hogeschool Delft, March 1976.

[Munro and Brown, 2001] David S. Munro and Alfred L. Brown. Evaluating partition selection policies using the PMOS garbage collector. In Kirby et al. [Kirby *et al.*2001], pages 104–115.

[Munro *et al.*, 1999] David Munro, Alfred Brown, Ron Morrison, and J. Eliot B. Moss. Incremental garbage collection of a persistent object store using PMOS. In Ron Morrison, Mick Jordan, and Malcolm Atkinson, editors, *Advances in Persistent Object Systems*, pages 78–91. Morgan Kaufman, 1999.

[Munsin and Lilius, 2002] Henrik Munsin and Johan Lilius. Compile-time garbage collection using escape analysis. In *Proceedings of the Nordic Workshop on Software Development Tools and Techniques NWPER 2002*, 2002.

[Murtagh, 1984] Thomas P. Murtagh. A less dynamic memory allocation scheme for Algol-like languages. In POPL 1984 [POPL 19841984], pages 283–289.

[Murtagh, 1991] Thomas P. Murtagh. An improved storage management scheme for block structured languages. *ACM Transactions on Programming Languages and Systems*, 13(3):372–398, July 1991.

[Muthu Kumar and Janakiram, 2006] R.M. Muthu Kumar and D. Janakiram. Yama: a scalable generational garbage collector for Java in multiprocessor systems. *IEEE Transactions on Parallel and Distributed Systems*, 17(2):148–159, 2006.

[Muthukumar and Janakiram, 2004] R.M. Muthukumar and D. Janakiram. Yama: a scalable generational garbage collector for Java in multiprocessor systems. Technical Report DOS-CSE-2004-14, Distributed and Object Systems Lab, Indian Institute of Technology, Madras, 2004.

[Mycroft, 2001] Alan Mycroft. Statically allocated systems. In SPACE 2001 [SPACE 20012001].

[Mytkowicz *et al.*, 2008] Todd Mytkowicz, Amer Diwan, Matthias Hauswirth, and Peter F. Sweeney. Producing wrong data without doing anything obviously wrong! In Soffa [Soffa2008], pages 265–276.

[Naeem and Lhoták, 2009] Nomair A. Naeem and Ondrej Lhoták. Efficient alias set analysis using SSA form. In Kolodner and Steele [Kolodner and Steele2009], pages 79–88.

[Naganuma *et al.*, 1988] Jiro Naganuma, Takeshi Ogura, Shin-ichiro Yamada, and Takashi Kimura. High-speed CAM-based architecture for a Prolog machine (ASCA). *IEEE Transactions on Computers*, 37(11):1375–1383, 11 1988.

[Nagarajan *et al.*, 2009] Vijay Nagarajan, Dennis Jeffrey, and Rajiv Gupta. Self-recovery in server programs. In Kolodner and Steele [Kolodner and Steele2009], pages 49–58.

[Nakajima, 1988a] Katsuto Nakajima. Piling GC: Efficient garbage collection for Al languages. ICOT technical report TR-354, Institute for New Generation Computer Technology, 1988.

[Nakajima, 1988b] Katsuto Nakajima. Piling GC: Efficient garbage collection for Al languages. In *IFIP WG 10.3 Working Conference on Parallel Processing*, pages 210–204. North Holland, 1988.

[Nakhli *et al.*, 2006] C. Nakhli, C. Rippert, G. Salagnac, and S. Yovine. Efficient region-based memory management for resource-limited real-time embedded systems. In Olivier Zendra, editor, *Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS'2006)*, page 8, Nantes, France, July 2006.

[Nandivada and Detlefs, 2005] V. Krishna Nandivada and David Detlefs. Compile-time concurrent marking write barrier removal. In *Proceedings of 3rd IEEE/ACM International Symposium on Code Generation and Oprimization (CGO 2005)*, pages 37–48, San Jose, CA, March 2005.

[Neely, 1996] Michael S. Neely. An analysis of the effects of memory allocation policy on storage fragmentation. Master's thesis, University of Texas at Austin, 1996.

[Neirynck *et al.*, 1987] Anne Neirynck, Prakash Panangaden, and Alan J. Demers. Computation of aliases and support sets. In POPL 1987 [POPL 19871987], pages 274–283.

[Neirynck, 1988] Anne Neirynck. *Static Analysis of Aliasing and Side Effects in Higher-Order Languages*. PhD thesis, Cornell University, January 1988.

[Nelson, 1983] Greg Nelson. Verifying reachability invariants of linked structures. In POPL 1983 [POPL 19831983], pages 38–47.

[Nelson, 1989] Jeffrey E. Nelson. Automatic, incremental, on-the-fly garbage collection of actors. Master's thesis, Virginia Polytechnic Institute and State University, 1989.

[Nethercote and Fitzhardinge, 2004] Nicholas Nethercote and Jeremy Fitzhardinge. Bounds-checking entire programs without recompiling. In SPACE 2004 [SPACE 20042004].

[Nettles and O'Toole, 1993a] Scott Nettles and James O'Toole. Real-time replication-based garbage collection. In PLDI 1993 [PLDI 19931993], pages 217–226.

[Nettles and O'Toole, 1993b] Scott M. Nettles and James W. O'Toole. Implementing orthogonal persistence: A simple optimization based on replicating collection. In IWOOOS 1993 [IWOOOS 19931993].

[Nettles and O'Toole, 1997] Scott M. Nettles and James W. O'Toole. A rollback technique for implementing persistence by reachability. In Connor and Nettles [Connor and Nettles1997].

[Nettles *et al.*, 1992] Scott M. Nettles, James W. O'Toole, David Pierce, and Nicholas Haines. Replication-based incremental copying collection. In Bekkers and Cohen [Bekkers and Cohen1992].

[Nettles *et al.*, 1993] Scott M. Nettles, James W. O'Toole, and David Gifford. Concurrent garbage collection of persistent heaps. Technical Report MIT/LCS/TR–569 and CMU–CS–93–137, Computer Science Department, Carnegie-Mellon University, April 1993. The same paper as [O'Toole *et al.*, 1993].

[Nettles, 1992] Scott M. Nettles. A Larch specification of copying garbage collection. Research paper CMU-CS-92-219, School of Computer Science, Carnegie Mellon University, December 1992.

[Newell and Shaw, 1957] A. Newell and J. C. Shaw. Programming the logic theory machine. In *Proceedings of the Western Joint Computing Conference*, pages 230–240, 1957.

[Newell and Tonge, 1960] A. Newell and F. M. Tonge. An introduction to information processing language V. *Communications of the ACM*, 3(4):205–211, April 1960.

[Newman *et al.*, 1982a] I. A. Newman, R. P. Stallard, and M. C. Woodward. Alternative approaches to multiprocessor garbage collection. In *Proceedings of the International Conference on Parallel Processing*, pages 205–210. IEEE Press, August 1982.

[Newman *et al.*, 1982b] I. A. Newman, R. P. Stallard, and M. C. Woodward. Performance of parallel garbage collection algorithms. *Computer Studies*, 166, September 1982.

[Newman *et al.*, 1983] I. A. Newman, R. P. Stallard, and M. C. Woodward. Improved multiprocessor garbage collection algorithms. In *Proceedings of the International Conference on Parallel Processing*, pages 367–368. IEEE Press, August 1983.

[Newman *et al.*, 1984] I. A. Newman, R. P. Stallard, and M. C. Woodward. A parallel compaction algorithm for multiprocessor garbage collection. In M. Feilmeier, J. Joubert, and U. Schendel, editors, *Proceedings of Parallel Computing 83 Conference*, pages 450–462, 1984.

[Newman *et al.*, 1987] I. A. Newman, R. P. Stallard, and M. C. Woodward. A hybrid multiple processor garbage collection algorithm. *Computer Journal*, 30(2):119–127, 1987.

[Ng and Glover, 1987] Y. H. Ng and R. J. Glover. Basic memory support for functional languages. In *1987 Proceedings — Fourth International IEEE VLSI Multilevel Interconnection Conference. Santa Clara, June 15–16*, pages 35–40. IEEE Press, 1987.

[Ng, 1996] T.C.T. Ng. Efficient garbage collection for large object-oriented databases. Master's thesis, MIT Laboratory for Computer Science, 1996.

[Nguyen and Rinard, 2007] Huu Hai Nguyen and Martin Rinard. Detecting and eliminating memory leaks using early phase termination. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 15–30.

[Ni *et al.*, 2007] Yang Ni, Vijay Menon, Ali-Reza Adl-Tabatabai, Antony L. Hosking, Richard L. Hudson, J. Eliot B. Moss, Bratin Saha, and Tatiana Shpeisman. Open nesting in software transactional memory. In *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 68–78, 2007.

[Nielsen, 1977] Norman R. Nielsen. Dynamic memory allocation in computer simulation. *Communications of the ACM*, 20(11):864–873, November 1977.

[Nierstras, 1993] O. Nierstras, editor. *Proceedings of 1993 European Conference on Object-Oriented Programming, ECOOP93*, volume 707 of *Lecture Notes in Computer Science*, Kaiserslautern, Germany, July 1993. Springer-Verlag.

[Nierstras, 1995] O. Nierstras, editor. *Proceedings of 1995 European Conference on Object-Oriented Programming, ECOOP95*, Lecture Notes in Computer Science. Springer-Verlag, August 1995.

[Nikhil and Ramachandran, 2000] R.S. Nikhil and U. Ramachandran. Garbage collection of timestamped data in Stampede. In *Proceedings of 19th Annual Symposium on Principles of Distributed Computing (PODC 2000)*, July 2000.

[Nilsen and Gao, 1995] Kelvin Nilsen and H. Gao. The real-time behaviour of dynamic memory management in C++. In *IEEE Real-Time Technologies and Applications Symposium*, pages 142–153, Chicago, May 1995. IEEE Press.

[Nilsen and Schmidt, 1990a] Kelvin D. Nilsen and William J. Schmidt. Hardware support for garbage collection of linked objects and arrays in real-time. In Jul and Juul [Jul and Juul1990].

[Nilsen and Schmidt, 1990b] Kelvin D. Nilsen and William J. Schmidt. A high-level overview of hardware assisted real-time garbage collection. Technical Report TR90-18a, Iowa State University, Department of Computer Science, October 1990.

[Nilsen and Schmidt, 1992a] Kelvin D. Nilsen and William J. Schmidt. Cost-effective object-space management for hardware-assisted real-time garbage collection. *Letters on Programming Language and Systems*, 1(4):338–354, December 1992.

[Nilsen and Schmidt, 1992b] Kelvin D. Nilsen and William J. Schmidt. Hardware-assisted general-purpose garbage collection for hard real-time systems. Technical Report ISU TR92-15, Iowa State University, Department of Computer Science, October 1992.

[Nilsen and Schmidt, 1992c] Kelvin D. Nilsen and William J. Schmidt. Preferred embodiment of a hardware-assisted garbage collection system. Technical Report ISU TR92-17, Iowa State University, Department of Computer Science, November 1992.

[Nilsen and Schmidt, 1993] Kelvin D. Nilsen and William J. Schmidt. Cost-effective object-space management for hardware-assisted real-time garbage collection. *Letters on Programming Languages and Systems*, 1(4):338–354, December 1993.

[Nilsen and Schmidt, 1994] Kelvin D. Nilsen and William J. Schmidt. A high-performance hardware-assisted real time garbage collection system. *Journal of Programming Languages*, 2(1), 1994.

[Nilsen, 1987] Kelvin D. Nilsen. Real-time garbage collection of strings and linked data structures. Technical Report TR 87-5, University of Arizona, Department of Computer Science, January 1987.

[Nilsen, 1988] Kelvin D. Nilsen. Garbage collection of strings and linked data-structures in real-time. *Software Practice and Experience*, 18(7):613–640, 1988.

[Nilsen, 1991] Kelvin D. Nilsen. A high-performance architecture for real-time garbage collection. In Wilson and Hayes [Wilson and Hayes1991a].

[Nilsen, 1992] Kelvin D. Nilsen. Memory cycle accountings for hardware-assisted real-time garbage collection. Technical Report 91-21(c), Iowa State University, Computer Science Department, 1992.

[Nilsen, 1993] Kelvin D. Nilsen. Reliable real-time garbage collection of C++. In Moss et al. [Moss *et al.*1993].

[Nilsen, 1994a] Kelvin D. Nilsen. Cost-effective hardware-assisted real-time garbage collection. In LCT-RTS 1994 [LCT-RTS 19941994].

[Nilsen, 1994b] Kelvin D. Nilsen. Reliable real-time garbage collection of C++. *Computing Systems*, 7(4), 1994.

[Nilsen, 1995a] Kelvin Nilsen. High-level dynamic memory management for object oriented real-time systems. In *Workshop on Object-Oriented Real-Time Systems*, San Antonio, Tx., October 1995.

[Nilsen, 1995b] Kelvin Nilsen. Progress in hardware-assisted real-time garbage collection. In Baker [Baker1995a].

[Nilsen, 1996a] Kelvin Nilsen. Invited note: Java for real-time. *Real-Time Systems Journal*, pages 197–205, September 1996.

[Nilsen, 1996b] Kelvin Nilsen. Issues in the design and implementation of real-time Java. *Java Developer's Journal*, 1(1):44, June 1996.

[Nilsen, 1996c] Kelvin Nilsen. Starting to PERC. *Java Developer's Journal*, 1(2):11, July 1996.

[Nilsen, 1998] Kelvin Nilsen. Adding real-time capabilities to the Java programming language. *Communications of the ACM*, 1998. To appear.

[Nilsen, 2009] Kelvin Nilsen. Differentiating features of the PERC virtual machine. White paper, Aonix, 2009.

[Ning and Xiong, 2006] Zhang Ning and Guangze Xiong. Minimizing GC work by analysis of live objects. *ACM SIGPLAN Notices*, 41(3):20–29, March 2006.

[Nishanov and Schupp, 1998a] Gor Nishanov and Sibylle Schupp. Design and implementation of the fgc garbage collector. Technical Report 98–7, Rensselaer Polytechnic Institute, NY, 1998. Extended version (1 December 1999) available.

[Nishanov and Schupp, 1998b] Gor Nishanov and Sibylle Schupp. Garbage collection in generic libraries. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 86–96.

[Nishida *et al.*, 1988a] Kenji Nishida, Yasunori Kimura, and A. Matsumoto. Evaluation of the effect of incremental garbage collection by MRB on FGHC parallel execution performance. ICOT technical report TR-394, Institute for New Generation Computer Technology, June 1988.

[Nishida *et al.*, 1988b] Kenji Nishida, Yasunori Kimura, A. Matsumoto, and A. Goto. Evaluation of MRB garbage collection on parallel logic programming architectures. In *7th International Conference on Logic Programming, Jerusalem*, pages 83–95. MIT Press, June 1988.

[Nishimura, 2006] Susumu Nishimura. Verifying data-parallel programs with separation logic. In SPACE 2006 [SPACE 20062006], pages 101–104.

[Nitzberg and Lo, 1991] B. Nitzberg and V. Lo. Distributed shared memory: A survey of issues and algorithms. *IEEE Computer*, pages 52–60, 1991.

[Niwa *et al.*, 1986] M. Niwa, M. Yuhara, K. Hayashi, and A. Hattori. Garbage collector with area optimization for FACOM ALPHA. In *COMPCON Spring 86: Thirty-First IEEE Computer Society International Conference. San Francisco, 1986 March 3–6*. IEEE Press, 1986.

[Noble *et al.*, 2000] James Noble, Charles Weir, and Duane Bibby. *Small Memory Software: Patterns for Systems with Limited Memory*. Addison-Wesley, 2000.

[Norcross *et al.*, 2003] Stuart Norcross, Ron Morrison, David S. Munro, and Henry Detmold. Implementing a family of distributed garbage collectors. In *Proceedings of the 2003 Australasian Computer Science Conference, (ACSC2003)*, pages 161–170, Adelaide, January 2003.

[Norcross, 2003] Stuart Norcross. *Deriving Distributed Garbage Collectors for Distributed Termination Algorithms*. PhD thesis, St Andrews University, 2003.

[Nori, 1979] A. K. Nori. A storage reclamation system for an applicative multiprocessor system. Master's thesis, University of Utah, Salt Lake City, Utah, 1979.

[North and Reppy, 1987] S. C. North and John H. Reppy. Concurrent garbage collection on stock hardware. In Kahn [Kahn1987], pages 113–133.

[Novark *et al.*, 2006] Gene Novark, Trevor Strohman, and Emery D. Berger. Custom object layout for garbage-collected languages. Technical report, University of Massachusetts at Amherst, 2006. NEPLS, March, 2006.

[Novark *et al.*, 2009] Gene Novark, Emery Berger, and Benjamin Zorn. Efficiently and precisely locating memory leaks and bloat. In PLDI 2009 [PLDI 20092009].

[Nuth and Halstead, 1989] Peter R. Nuth and Robert H. Halstead. A study of LISP on a multiprocessor (preliminary version). *Lisp Pointers*, 2(3–4):15–32, 1989.

[Nuth, 1987] Peter R. Nuth. Communication patterns in a symbolic multiprocessor. Technical Report MIT/LCS/TR–395, MIT Laboratory for Computer Science, June 1987.

[Oancea *et al.*, 2009] Cosmin E. Oancea, Alan Mycroft, and Stephen M. Watt. A new approach to parallelising tracing algorithms. In Kolodner and Steele [Kolodner and Steele2009], pages 10–19.

[Odaira *et al.*, 2010] Rei Odaira, Kazunori Ogata, Kiyokuni Kawachiya, Tamiya Onodera, and Toshio Nakatani. Efficient runtime tracking of allocation sites in Java. In Fiuczynski et al. [Fiuczynski *et al*.2010], pages 109–120.

[Odijik *et al.*, 1989] Eddy Odijik, M. Rem, and Jean-Claude Sayr, editors. *PARLE'89 Parallel Architectures and Languages Europe*, volume 265/366 of *Lecture Notes in Computer Science*, Eindhoven, The Netherlands, June 1989. Springer-Verlag.

[O'Farrell, 1991] William Gerald O'Farrell. *Garbage Collection Algorithms for the Connection Machine*. PhD thesis, Syracuse University, 1991.

[Ogasawara, 2009] Takeshi Ogasawara. NUMA-aware memory manager with dominant-thread-based copying GC. In OOPSLA 2009 [OOPSLA 20092009], pages 377–390.

[Ogata *et al.*, 2010] Kazunori Ogata, Dai Mikurube, Kiyokuni Kawachiya, Scott Trent, and Tamiya Onodera. A study of java's non-java memory. In OOPSLA 2010 [OOPSLA 20102010], pages 191–204.

[Ogura *et al.*, 1989] Takeshi Ogura, Junzo Yamada, Shin-Ichiro Yamada, and Masa-Aki Tan-No. 20-kbit associative memory LSI for artificial intelligence machines. *IEEE Journal of Solid-State Circuits*, 24(4), August 1989.

[Oiwa, 2009] Yutaka Oiwa. Implementation of the memory-safe full ANSI-C compiler. In PLDI 2009 [PLDI 20092009], pages 259–269.

[Oldehoeft and Allan, 1985] Rod R. Oldehoeft and S. J. Allan. Adaptive exact-fit storage management. *Communications of the ACM*, 28(5):506–511, May 1985.

[Older and Rummell, 1992] William J. Older and John A. Rummell. An incremental garbage collector for WAM-based Prolog. In *Proceedings of ICLP92International Conference on Logic Programming*, pages 369–383, Washington, DC, 1992.

[OMG, 1997] Object Management Group. *Garbage Collection of CORBA Objects*, 1997. Draft RFP.

[Omohundro, 1991] Stephen M. Omohundro. *The Sather Language*. ICSI, Berkeley, 1991.

[Onodera, 1993] Tamiya Onodera. A generational and conservative copying collector for hybrid object-oriented languages. *Software Practice and Experience*, 23(10):1077–1093, October 1993.

[OOPSLA 1986, 1986] *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 21(11), Portland, OR, USA, November 1986.

[OOPSLA 1987, 1987] *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 22(12), Orlando, FL, USA, December 1987.

[OOPSLA 1988, 1988] *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 23(11), San Diego, CA, USA, November 1988.

[OOPSLA 1989, 1989] *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 24(10), New Orleans, LA, USA, October 1989.

[OOPSLA 1990, 1990] *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 25(10), Ottawa, Canada, October 1990.

[OOPSLA 1991, 1991] *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 26(11), Phoenix, AZ, USA, November 1991.

[OOPSLA 1992, 1992] *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 27(10), Vancouver, Canada, October 1992.

[OOPSLA 1993, 1993] *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 28(10), Washington, DC, USA, October 1993.

[OOPSLA 1994, 1994] *Proceedings of the Ninth Annual Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 29(10), Portland, OR, USA, October 1994.

[OOPSLA 1995, 1995] *Proceedings of the Tenth Annual Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 30(10), Austin, TX, USA, October 1995.

[OOPSLA 1996, 1996] *Proceedings of the Eleventh ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 31(10), San Jose, CA, USA, October 1996.

[OOPSLA 1997, 1997] *Proceedings of the Twelfth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 32(10), Atlanta, GA, USA, October 1997.

[OOPSLA 1998, 1998] *Proceedings of the Thirteenth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 33(10), Vancouver, Canada, October 1998.

[OOPSLA 1999, 1999] *Proceedings of the Fourteenth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 34(10), Denver, CO, USA, October 1999.

[OOPSLA 2000, 2000] *Proceedings of the Fifteenth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 35(10), Minneapolis, MN, USA, October 2000.

[OOPSLA 2001, 2001] *Proceedings of the Sixteenth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 36(11), Tampa, FL, USA, November 2001.

[OOPSLA 2002, 2002] *Proceedings of the Seventeenth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 37(11), Seattle, WA, USA, November 2002.

[OOPSLA 2003, 2003] *Proceedings of the Eighteenth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, Anaheim, CA, USA, November 2003.

[OOPSLA 2004, 2004] *Proceedings of the Nineteenth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 39(10), Vancouver, Canada, October 2004.

[OOPSLA 2005, 2005] *Proceedings of the Twentieth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 40(10), San Diego, CA, USA, October 2005.

[OOPSLA 2006, 2006] *Proceedings of the Twenty-First ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 41(10), Portland, OR, USA, October 2006.

[OOPSLA 2007, 2007] *Proceedings of the Twenty-Second ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 42(10), Montréal, Canada, October 2007.

[OOPSLA 2008, 2008] *Proceedings of the Twenty-Third ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 43(10), Nashville, TN, USA, October 2008.

[OOPSLA 2009, 2009] *Proceedings of the Twenty-Fourth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ACM SIGPLAN Notices 44(10), Orlando, FL, USA, October 2009.

[OOPSLA 2010, 2010] *Proceedings of the Twenty-Fourth ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, Reno, NV, USA, October 2010.

[Operowsky, 1989] H. L. Operowsky. *Optimization and Garbage Collection in Ada Programs on Shared Memory Computers*. PhD thesis, New York Academy of Sciences, New York, 1989.

[OptimizeIt,] Borland. *OptimizeIt*. http://www.optimizeit.com.

[Oracle Corporation, 2008] Mission-critical Java. Oracle white paper, Oracle Corporation, October 2008. Describes Oracle JRockit Real Time.

[Organick, 1983] E. I. Organick. *A Programmer's View of the Intel 432 System*. McGraw-Hill, 1983.

[Orlovich and Rugina, 2006] M. Orlovich and R. Rugina. Memory leak analysis by contradiction. In *Proceedings of International Static Analysis Symposium (SAS'06)*, Seoul, South Korea, August 2006.

[Ossia *et al.*, 2002] Yoav Ossia, Ori Ben-Yitzhak, Irit Goft, Elliot K. Kolodner, Victor Leikehman, and Avi Owshanko. A parallel, incremental and concurrent GC for servers. In PLDI 2002 [PLDI 20022002], pages 129–140.

[Ossia *et al.*, 2004] Yoav Ossia, Ori Ben-Yitzhak, and Marc Segal. Mostly concurrent compaction for mark-sweep GC. In Bacon and Diwan [Bacon and Diwan2004], pages 25–36.

[O'Toole and Nettles, 1993a] James W. O'Toole and Scott M. Nettles. Concurrent replicating garbage collection. Technical Report MIT–LCS–TR–570 and CMU–CS–93–138, MIT and CMU, 1993. Also LFP94 and OOPSLA93 Workshop on Memory Management and Garbage Collection.

[O'Toole and Nettles, 1993b] James W. O'Toole and Scott M. Nettles. Real-time replication GC: An implementation report. Technical Report MIT–LCS–TR–568 and CMU–CS–93–136, MIT and CMU, 1993. WWW page says this is unpublished.

[O'Toole *et al.*, 1993] James W. O'Toole, Scott M. Nettles, and David Gifford. Concurrent compacting garbage collection of a persistent heap. In SOSP 1993 [SOSP 19931993], pages 161–174. Also MIT/CMU Technical report MIT–LCS–TR–569. The same paper as [Nettles *et al.*, 1993].

[O'Toole, 1990] James W. O'Toole. Garbage collecting locally, December 1990. Area Exam, Department of Electrical Engineering and Computer Science, MIT.

[O'Toole, 1993] James W. O'Toole. Garbage collecting an object cache. Technical Report MIT/LCS/TM–485, MIT Press, April 1993.

[Ovm,] The Ovm virtual machine. http://www.ovmj.net.

[Owicki and Gries, 1976] Susan Owicki and David Gries. Verifying properties of parallel programs: An axiomatic approach. *Communications of the ACM*, 19(5):279–285, May 1976.

[Owicki and Lamport, 1982] Susan Owicki and Leslie Lamport. Proving liveness properties of concurrent programs. *ACM Transactions on Programming Languages and Systems*, 4(3):455–495, July 1982.

[Owicki, 1981] Susan Owicki. Making the world safe for garbage collection. In POPL 1981 [POPL 19811981], pages 77–86.

[Ozawa *et al.*, 1989] Toshihiro Ozawa, Akira Hosoi, and Akira. Hattori. Generation type garbage collection for parallel logic languages. ICOT technical report TR-512, Institute for New Generation Computer Technology, October 1989.

[Page and Hagins, 1986] Ivor P. Page and Jeff Hagins. Improving the performance of buddy systems. *IEEE Transactions on Computers*, C-35(5):441–447, May 1986.

[Page, 1982] Ivor P. Page. Optimal fit of arbitrary sized segments. *Computer Journal*, 25(1), January 1982.

[Page, 1984] Ivor P. Page. Analysis of a cyclic placement scheme. *Computer Journal*, 27(1):18–25, January 1984.

[Palacharla and Kessler, 1994] S. Palacharla and R. E. Kessler. Evaluating stream buffers as a secondary cache replacement. In ISCA 1994 [ISCA 19941994], pages 24–33.

[Palacz *et al.*, 1994] Krzysztof Palacz, Jan Vitek, Grzegorz Czajkowski, and Laurent Daynès. Incommunicado: efficient communication for isolates. In OOPSLA 1994 [OOPSLA 19941994], pages 262–274.

[Pallas and Ungar, 1988] Joseph Pallas and David Ungar. Multiprocessor Smalltalk: A case study of a multiprocessor-based programming environment. In PLDI 1988 [PLDI 19881988], pages 268–277.

[Pan, 1986] Wilson Pan. Designing an operating system kernal based on concurrent garbage collection. Technical Report 86-04, University of Iowa, Department of Computer Science, 1986.

[Panzer, 1986] Edward J. Panzer. Execution time of marking algorithms during garbage collection in LISP. Master's thesis, California State Polytechnic University, 1986.

[Parents, 1968] R. J. Parents. A simulation oriented memory allocation algorithm. In J. M. Buxton, editor, *Simulation Programming Languages*, pages 199–209. North-Holland, Amsterdam, 1968.

[Pareto, 2001] Lars Pareto. Sized region types. In SPACE 2001 [SPACE 20012001].

[Park and Goldberg, 1991] Young Gil Park and Benjamin Goldberg. Reference escape analysis: Optimizing reference counting based on the lifetime of references. In *Symposium on Partial Evaluation and Semantics-Based Program Manipul ation*, pages 178–189, New Haven, Connecticut, June 1991.

[Park and Goldberg, 1992] Young G. Park and Benjamin Goldberg. Escape analysis on lists. *ACM SIGPLAN Notices*, 27(7):116–127, June 1992.

[Park and Goldberg, 1995] Young G. Park and Benjamin Goldberg. Static analysis for optimising reference counting. *Information Processing Letters*, 55(4):229–234, August 1995.

[Park, 1991] Young G. Park. *Semantic Analyses for Storage Management Optimizations in Functional Language Implementations*. PhD thesis, New York University, 1991.

[PARLE94, 1994] *PARLE'94 Parallel Architectures and Languages Europe*, Lecture Notes in Computer Science. Springer-Verlag, 1994.

[PASTE98, 1998] *ACM SUGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering PARLE'98*. ACM Press, 1998.

[Patil and Fischer, 1997] Harish G. Patil and Charles N. Fischer. Low-cost, concurrent checking of pointer and array accesses in C programs. *Software Practice and Experience*, 27(12):87–110, December 1997.

[Patterson, 1983] David A. Patterson. Smalltalk on a RISC: Architectural investigations. Technical Report CS292R, Computer Science Division, University of California, Berkeley, April 1983.

[Pavlovic *et al.*, 2003] D. Pavlovic, P. Pepper, and D. R. Smith. Colimits for concurrent collectors. In *Verification: Theory and Practice, essays dedicated to Zohar Manna on the occasion of his 64th birthday*, volume 2772 of *Lecture Notes in Computer Science*, pages 568–597. Springer-Verlag, 2003.

[Pavlovic *et al.*, 2010a] Dusko Pavlovic, Peter Pepper, and Douglas R. Smith. Formal derivation of concurrent garbage collectors. In *Tenth international conference on Mathematics of Program Construction (MPC'10)*, Lecture Notes in Computer Science, Québec, Canada, 2010. Springer-Verlag.

[Pavlovic *et al.*, 2010b] Dusko Pavlovic, Peter Pepper, and Douglas R. Smith. Formal derivation of concurrent garbage collectors. Technical report, 2010. The short version of this paper appeared in MPC 2010.

[Pawlam, 1999] Monica Pawlam. Reference objects and garbage collection. Article on Sun's Java Developer Connection site, 1999.

[Payer and Ana, 2007] Hannes Payer and Ana. A compacting real-time memory management system. Master's thesis, University of Salzburg, 2007.

[Paz and Petrank, 2007] Harel Paz and Erez Petrank. Using prefetching to improve reference-counting garbage collectors. In *Proceedings of the 16th International Conference on Compiler Construction (CC'07)*, March 2007.

[Paz *et al.*, 2003] Harel Paz, David F. Bacon, Elliot K. Kolodner, Erez Petrank, and V.T. Rajan. Efficient on-the-fly cycle collection. Technical Report CS–2003–10, Technion University, 2003.

[Paz *et al.*, 2005a] Harel Paz, Erez Petrank, David F. Bacon, V.T. Rajan, and Elliot K. Kolodner. An efficient on-the-fly cycle collection. In CC 2005 [CC 20052005].

[Paz *et al.*, 2005b] Harel Paz, Erez Petrank, and Stephen M. Blackburn. Age-oriented concurrent garbage collection. In CC 2005 [CC 20052005].

[Paz *et al.*, 2007] Harel Paz, Erez Petrank, David F. Bacon, V.T. Rajan, and Elliot K. Kolodner. An efficient on-the-fly cycle collection. *ACM Transactions on Programming Languages and Systems*, 29(4):1–43, 2007. Article 20.

[Paz, 2006] Harel Paz. *Efficient Memory Management for Servers*. PhD thesis, Technion, Israel Institute of Technology, 2006.

[Pearlmutter, 1996] B. Pearlmutter. Garbage collection with pointers to individuals cells. *Communications of the ACM*, 39(12), December 1996.

[Peck, 1971] J. E. L. Peck, editor. *Algol–68 implementation*. North-Holland, Amsterdam, 1971.

[Pedersen and Schoeberl, 2006] Rasmus Pedersen and Martin Schoeberl. Exact roots for a real-time garbage collector. In *Proceedings of the Fourth International Workshop on Java Technologies for Real-time and Embedded Systems*, pages 77–84, 2006.

[Peir *et al.*, 1998] Jih-Kwon Peir, Yongjoon Lee, and Windsor W. Hsu. Capturing dynamic memory reference behavior with adaptive cache topology. In ASPLOS 1998 [ASPLOS 19981998], pages 250–250.

[Peng and Sohi, 1989] Chih-Jui Peng and Gurindar S. Sohi. Cache memory design considerations to support languages with dynamic heap allocation. Technical Report 860, Computer Sciences Department, University of Wisconsin-Madison, July 1989.

[Pepels *et al.*, 1988] E. J. H. Pepels, M. C. J. D. van Eekelen, and M. J. Plasmeijer. A cyclic reference counting algorithm and its proof. Technical Report 88–10, Computing Science Department, University of Nijmegen, 1988.

[Per-øAke Larson and Krishnan, 1998] Paul Per-øAke Larson and Murali Krishnan. Memory allocation for long-running server applications. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 176–185.

[Pereira and Aycock, 2002] D. Pereira and J. Aycock. Dynamic region inference. Technical Report 2002 709 12, University of Calgary, 2002.

[Persson, 1999] Patrik Persson. Live memory analysis for garbage collection in embedded systems. In LCTES 1999 [LCTES 19991999], pages 45–54.

[Persson, 2006a] Mattias Persson. Java technology, IBM style: Garbage collection policies, May 2006. Garbage collection in the IBM SDK 5.0.

[Persson, 2006b] Mattias Persson. Java technology, IBM style: Garbage collection policies, part 1. *IBM developerWorks*, 2006.

[Peterson and Norman, 1977] J. L. Peterson and T. A. Norman. Buddy systems. *Communications of the ACM*, 20(6):421–431, 1977.

[Peterson *et al.*, 2003] Leaf Peterson, Robert Harper, Karl Crary, and Frank Pfenning. A type theory for memory allocation and data abstraction. In POPL 2003 [POPL 20032003].

[Petit-Bianco, 1998] Alexandre Petit-Bianco. Java garbage collection for real-time systems. *Dr. Dobb's Journal*, October 1998.

[Petrank and Kolodner, 2004] Erez Petrank and Elliot K. Kolodner. Parallel copying garbage collection using delayed allocation. *Parallel Processing Letters*, 14(2), June 2004.

[Petrank and Moss, 2006] Erez Petrank and J. Eliot B. Moss, editors. *Proceedings of the Fifth International Symposium on Memory Management*, Ottawa, Canada, June 2006. ACM Press.

[Petrank and Rawitz, 2002a] Erez Petrank and Dror Rawitz. The hardness of cache conscious data placement. In POPL 2002 [POPL 20022002], pages 101–112.

[Petrank and Rawitz, 2002b] Erez Petrank and Dror Rawitz. The hardness of cache conscious data placement. In POPL 2002 [POPL 20022002]. Extended Abstract.

[Petrank and Rawitz, 2003] Erez Petrank and Dror Rawitz. The hardness of cache conscious data placement. Technical report, Technion — Israel Institute of Technology, Haifa, Israel, 2003.

[Petricek and Syme, 2010] Tomas Petricek and Don Syme. Collecting Hollywood's garbage: Avoiding space-leaks in composite events. In jan Vitek and Lea [jan Vitek and Lea2010], pages 53–62.

[Petrov and Vechev, 2002] Peter D. Petrov and Martin T. Vechev. Embedded JVM concurrent garbage collector internals. In *IASTED Networks, Parallel and Distributed Processing, and Applications (NPDPA'02)*, 2002.

[Peyton Jones and Jones, 1998] Simon L. Peyton Jones and Richard Jones, editors. *Proceedings of the First International Symposium on Memory Management*, ACM SIGPLAN Notices 34(3), Vancouver, Canada, October 1998.

[Peyton Jones and Ramsey, 1998] Simon L. Peyton Jones and Norman Ramsey. Machine-independent support for garbage collection, debugging, exception handling and concurrency. Technical Report CS–98–19, University of Virginia, August 1998.

[Peyton Jones *et al.*, 1991] Simon L. Peyton Jones, G. Hutton, and C. K. Hols, editors. *Third Annual Glasgow Workshop on Functional Programming*. Springer-Verlag, 1991.

[Peyton Jones *et al.*, 1992] Simon L. Peyton Jones, G. Hutton, and C. K. Hols, editors. *Fourth Annual Glasgow Workshop on Functional Programming*, Workshops in Computer Science. Springer-Verlag, 1992.

[Peyton Jones, 1992] Simon L. Peyton Jones. Implementing lazy functional languages on stock hardware: The Spineless Tagless G-machine. *Journal of Functional Programming*, 2(2):127–202, April 1992.

[Pfalz, 1977] J. L. Pfalz. *Computer Data Structures*. McGraw-Hill, 1977.

[Phalke and Gopinath, 1995] Vidyadhar Phalke and B. Gopinath. A miss history-based architecture for cache prefetching. In Baker [Baker1995a].

[Phan *et al.*, 2008] Quan Phan, Gerda Janssens, and Zoltan Somogyi. Runtime support for region-based memory management in Mercury. In Jones and Blackburn [Jones and Blackburn2008], pages 61–70.

[Philippsen, 2000] Michael Philippsen. Cooperating distributed garbage collectors for clusters and beyond. *Concurrency and Computation: Practice and Experience*, 12(7):595–610, May 2000. Also published in 8th Int. Workshop on Compilers for Parallel Computers CPC'2000, Aussois, France.

[Phipps, 1999] G. Phipps. Comparing observed bug and productivity rates for Java and C++. *Software Practice and Experience*, 29(4):345–358, April 1999.

[Pieper, 1993] Pieper. Compiler techniques for managing data motion. Technical Report CMU-CS-93-217, Carnegie Mellon University, December 1993.

[Piquer, 1990a] José M. Piquer. Sharing date structures in distributed Lisp. In *Proceedings of High Performance and Parallel Computing in Lisp Workshop*, London, November 1990.

[Piquer, 1990b] José M. Piquer. Un GC parallèle pour un Lisp distribué. *Journées francophones des langages applicatifs*, January 1990. Also Bigre 69, July 1990.

[Piquer, 1991a] José M. Piquer. Indirect reference counting: A distributed garbage collection algorithm. In Aarts et al. [Aarts and others1991].

[Piquer, 1991b] José M. Piquer. *Parallélisme et Distribution en Lisp*. PhD thesis, Ecole Polytechnique, Massy, France, January 1991.

[Piquer, 1992] José M. Piquer. Dynamic revision of choice points during garbage collection in Prolog. In Bekkers and Cohen [Bekkers and Cohen1992].

[Piquer, 1995] José M. Piquer. Indirect mark and sweep: A distributed GC. In Baker [Baker1995a].

[Piquer, 1996] José M. Piquer. Indirect distributed garbage collection: Handling object migration. *ACM Transactions on Programming Languages and Systems*, 18(5):615–647, September 1996.

[Pirinen, 1998] Pekka P. Pirinen. Barrier techniques for incremental tracing. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 20–25.

[Pittomvils *et al.*, 1985] Edwin Pittomvils, Maurice Bruynooghe, and Yves D. Willems. Towards a real time garbage collector for PROLOG. In *1985 Symposium on Logic Programming. Boston, 1985 Jul 15–18*, pages 185–198. IEEE Press, 1985.

[Piumarta *et al.*, 1995] Ian Piumarta, Marc Shapiro, and Paulo Ferreira. Garbage collection in distributed object systems. In *Workshop on Reliability and Scalability in Distributed Object Systems, OOPSLA'95*, Austin, TX, October 1995.

[Piumarta, 1995] Ian Piumarta. Ssp chains – from mobile objects to mobile computing (position paper). In *ECOOP Workshop on Mobility, 1995*, 1995.

[Pixley, 1988] C. Pixley. An incremental garbage collection algorithm for multi-mutator systems. *Distributed Computing*, 3(1):41–50, 1988.

[Pizlo and Vitek, 2006] Filip Pizlo and Jan Vitek. An empirical evaluation of memory management alternatives for real-time Java. In *Proceedings of the 27th IEEE Real-Time Systems Symposium (RTSS)*, 2006.

[Pizlo and Vitek, 2008] Filip Pizlo and Jan Vitek. Memory management for real-time Java: State of the art. In ISORC 2008 [ISORC 20082008], pages 248–254.

[Pizlo *et al.*, 2004] Filip Pizlo, J.Fox, David Holmes, and Jan Vitek. Real-time java scoped memory: Design patterns, semantics. In ISORC 2004 [ISORC 20042004], pages 101–112.

[Pizlo *et al.*, 2007a] Filip Pizlo, Daniel Frampton, Erez Petrank, and Bjarne Steensgard. STOPLESS: A real-time garbage collector for multiprocessors. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 159–172.

[Pizlo *et al.*, 2007b] Filip Pizlo, Antony L. Hosking, and Jan Vitek. Hierarchical real-time garbage collection. In LCTES 2007 [LCTES 20072007], pages 123–133.

[Pizlo *et al.*, 2008a] Filip Pizlo, Erez Petrank, and Bjarne Steensgaard. Path specialization: Reducing phased execution overheads. In Jones and Blackburn [Jones and Blackburn2008], pages 81–90.

[Pizlo *et al.*, 2008b] Filip Pizlo, Erez Petrank, and Bjarne Steensgaard. A study of concurrent real-time garbage collectors. In Gupta and Amarasinghe [Gupta and Amarasinghe2008], pages 33–44.

[Pizlo *et al.*, 2010] Filip Pizlo, Lukasz Ziarek, Petr Maj, Antony L. Hosking, Ethan Blanton, and Jan Vitek. Schism: Fragmentation-tolerant real-time garbage collection. In PLDI 2010 [PLDI 20102010], pages 146–159.

[Plainfoseé and Shapiro, 1992] David Plainfoseé and Marc Shapiro. A distributed GC in an object-support operating system. In Cabrera et al. [Cabrera *et al.*1992].

[Plainfoseé and Shapiro, 1994] David Plainfoseé and Marc Shapiro. A survey of distributed garbage collection techniques. In *Second Closed BROADCAST Workshop*, Bruxelles (Belgique), November 1994. Broadcast Basic Research Action. Superseded by [Plainfossé and Shapiro, 1995].

[Plainfossé and Shapiro, 1991a] David Plainfossé and Marc Shapiro. A distributed garbage collection as an operating system component. In Wilson and Hayes [Wilson and Hayes1991a].

[Plainfossé and Shapiro, 1991b] David Plainfossé and Marc Shapiro. Distributed garbage collection in the system is good. In Cabrera et al. [Cabrera *et al.*1991], pages 94–99.

[Plainfossé and Shapiro, 1992] David Plainfossé and Marc Shapiro. Experience with fault-tolerant garbage collection in a distributed Lisp system. In Bekkers and Cohen [Bekkers and Cohen1992].

[Plainfossé and Shapiro, 1995] David Plainfossé and Marc Shapiro. A survey of distributed garbage collection techniques. In Baker [Baker1995a].

[Plainfossé, 1994] David Plainfossé. *Distributed Garbage Collection and Reference Management in the Soul Object Support System*. PhD thesis, Université Paris-6, Pierre-et-Marie-Curie, Paris (France), June 1994. Available from INRIA as TU-281, ISBN-2-7261-0849-0.

[Plakal and Fischer, 2000] Manoj Plakal and Charles N. Fischer. Concurrent garbage collection using program slices on multithreaded processors. In Chambers and Hosking [Chambers and Hosking2000], pages 94–100.

[Plauger, 1994] P.J. Plauger. Managing the heap. *Journal of C Language Translation*, 6(1), September 1994.

[PLDI 1988, 1988] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 23(7), Atlanta, GA, USA, June 1988.

[PLDI 1989, 1989] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 24(7), Portland, OR, USA, June 1989.

[PLDI 1990, 1990] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 25(6), White Plains, NY, USA, June 1990.

[PLDI 1991, 1991] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 26(6), Toronto, Canada, June 1991.

[PLDI 1992, 1992] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 27(7), San Francisco, CA, USA, June 1992.

[PLDI 1993, 1993] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 28(6), Albuquerque, NM, USA, June 1993.

[PLDI 1994, 1994] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 29(6), Orlando, FL, USA, June 1994.

[PLDI 1995, 1995] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 30(6), La Jolla, CA, USA, June 1995.

[PLDI 1996, 1996] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 31(5), Philadelphia, PA, USA, May 1996.

[PLDI 1997, 1997] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 32(5), Las Vegas, NV, USA, June 1997.

[PLDI 1998, 1998] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 33(5), Montreal, Canada, June 1998.

[PLDI 1999, 1999] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 34(5), Atlanta, GA, USA, May 1999.

[PLDI 2000, 2000] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 35(5), Vancouver, Canada, June 2000.

[PLDI 2001, 2001] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 36(5), Snowbird, UT, USA, June 2001.

[PLDI 2002, 2002] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 37(5), Berlin, Germany, June 2002.

[PLDI 2003, 2003] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 38(5), San Diego, CA, USA, June 2003.

[PLDI 2009, 2009] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices, Dublin, Ireland, June 2009.

[PLDI 2010, 2010] *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices, Toronto, Canada, June 2010.

[PLILP92, 1992] *Fourth International Symposium on Programming Language Implementation and Logic Programming*, volume 631 of *Lecture Notes in Computer Science*, Leuven, Belgium, August 1992. Springer-Verlag.

[Pollack *et al.*, 1982] Fred J. Pollack, George W. Cox, Dan W. Hammerstein, Kevin C. Kahn, Konrad K. Lai, and Justin R. Rattner. Supporting Ada memory management in the iAPX–432. In ASPLOS 1982 [ASPLOS 19821982], pages 117–131.

[Pomerene *et al.*, 1985a] J. Pomerene, T. R. Puzak, R. N. Rechtshaffen, and F. J. Sparacio. Shadow structure to perform D-line prefetching. *IBM Technical Disclosure Bulletin*, 27(5):2987–2988, 1985.

[Pomerene *et al.*, 1985b] J. H. Pomerene, T. R. Puzak, R. N. Rechtshaffen, and F. J. Sparacio. Prefetching confirmation array. *IBM Technical Disclosure Bulletin*, 27(5):2786–2787, 1985.

[Pomerene *et al.*, 1985c] J. H. Pomerene, T. R. Puzak, R. N. Rechtshaffen, and F. J. Sparacio. Prefetching pacing buffer to reduce cache misses. *IBM Technical Disclosure Bulletin*, 27(5):2773–2774, 1985.

[Poon and Peyton Jones, 1985] E. K. Y. Poon and Simon L. Peyton Jones. Cache memories in a functional programming environment. In *Aspenäs Workshop on Implementation of Functional Languages, Göteborg*, 1985. Also UCL Computer Science Internal Note 1680.

[POPL 1981, 1981] *Conference Record of the Eighth Annual ACM Symposium on Principles of Programming Languages*, Williamsburg, VA, USA, January 1981. ACM Press.

[POPL 1982, 1982] *Conference Record of the Ninth Annual ACM Symposium on Principles of Programming Languages*, Albuquerque, NM, USA, January 1982. ACM Press.

[POPL 1983, 1983] *Conference Record of the Tenth Annual ACM Symposium on Principles of Programming Languages*, Austin, TX, USA, January 1983. ACM Press.

[POPL 1984, 1984] *Conference Record of the Eleventh Annual ACM Symposium on Principles of Programming Languages*, Salt Lake City, UT, USA, January 1984. ACM Press.

[POPL 1985, 1985] *Conference Record of the Twelfth Annual ACM Symposium on Principles of Programming Languages*, New Orleans, LA, USA, January 1985. ACM Press.

[POPL 1986, 1986] *Conference Record of the Thirteenth Annual ACM Symposium on Principles of Programming Languages*, St Petersburg Beach, FL, USA, January 1986. ACM Press.

[POPL 1987, 1987] *Conference Record of the Fourteenth Annual ACM Symposium on Principles of Programming Languages*, Munich, Germany, January 1987. ACM Press.

[POPL 1988, 1988] *Conference Record of the Fifteenth Annual ACM Symposium on Principles of Programming Languages*, San Diego, CA, USA, January 1988. ACM Press.

[POPL 1989, 1989] *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages*, Austin, TX, USA, January 1989. ACM Press.

[POPL 1990, 1990] *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages*, San Francisco, CA, USA, January 1990. ACM Press.

[POPL 1991, 1991] *Conference Record of the Eighteenth Annual ACM Symposium on Principles of Programming Languages*, Orlando, FL, USA, January 1991. ACM Press.

[POPL 1992, 1992] *Conference Record of the Nineteenth Annual ACM Symposium on Principles of Programming Languages*, Albuquerque, NM, USA, January 1992. ACM Press.

[POPL 1993, 1993] *Conference Record of the Twentieth Annual ACM Symposium on Principles of Programming Languages*, Charleston, SC, USA, January 1993. ACM Press.

[POPL 1994, 1994] *Conference Record of the Twenty-First Annual ACM Symposium on Principles of Programming Languages*, Portland, OR, USA, January 1994. ACM Press.

[POPL 1995, 1995] *Conference Record of the Twenty-Second Annual ACM Symposium on Principles of Programming Languages*, San Francisco, CA, USA, January 1995. ACM Press.

[POPL 1996, 1996] *Conference Record of the Twenty-Third Annual ACM Symposium on Principles of Programming Languages*, St Petersburg Beach, FL, USA, January 1996. ACM Press.

[POPL 1997, 1997] *Conference Record of the Twenty-Fourth Annual ACM Symposium on Principles of Programming Languages*, Paris, France, January 1997. ACM Press.

[POPL 1998, 1998] *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Programming Languages*, San Diego, CA, USA, January 1998. ACM Press.

[POPL 1999, 1999] *Proceedings of the Twenty-sixth Annual ACM Symposium on Principles of Programming Languages*, San Antonio, TX, USA, January 1999. ACM Press.

[POPL 2000, 2000] *Proceedings of the Twenty-sixth Annual ACM Symposium on Principles of Programming Languages*, Boston, MA, USA, January 2000. ACM Press.

[POPL 2001, 2001] *Conference Record of the Twenty-eighth Annual ACM Symposium on Principles of Programming Languages*, ACM SIGPLAN Notices 36(3), London, England, January 2001.

[POPL 2002, 2002] *Conference Record of the Twenty-ninth Annual ACM Symposium on Principles of Programming Languages*, ACM SIGPLAN Notices 37(1), Portland, OR, USA, January 2002.

[POPL 2003, 2003] *Conference Record of the Thirtieth AnnualACM Symposium on Principles of Programming Languages*, ACM SIGPLAN Notices 38(1), New Orleans, LA, USA, January 2003.

[POPL 2004, 2004] *Proceedings of the Thirty-First Annual ACM Symposium on Principles of Programming Languages*, ACM SIGPLAN Notices 39(1), Venice, Italy, January 2004.

[POPL 2005, 2005] *Proceedings of the Thirty-Second Annual ACM Symposium on Principles of Programming Languages*, ACM SIGPLAN Notices 40(1), Long Beach, CA, USA, January 2005.

[POPL 2006, 2006] *Proceedings of the Thirty-Third Annual ACM Symposium on Principles of Programming Languages*, ACM SIGPLAN Notices 41(1), Charleston, SC, USA, January 2006.

[POPL 2007, 2007] *Proceedings of the Thirty-Fourth Annual ACM Symposium on Principles of Programming Languages*, ACM SIGPLAN Notices 42(1), Nice, France, January 2007.

[POPL 2008, 2008] *Proceedings of the Thirty-Fifth Annual ACM Symposium on Principles of Programming Languages*, ACM SIGPLAN Notices 43(1), San Francisco, CA, USA, January 2008.

[POPL 2009, 2009] *Proceedings of the Thirty-Sixth Annual ACM Symposium on Principles of Programming Languages*, Savannah, GA, USA, January 2009. ACM Press.

[Potanin *et al.*, 2005] Alex Potanin, James Noble, Marcus Frean, and Robert Biddle. Scale-free geometry in OO programs. *Communications of the ACM*, 48(5):99–103, May 2005.

[PPOPP 1990, 1990] *Proceedings of the Second ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ACM SIGPLAN Notices 25(3), Seattle, WA, USA, March 1990.

[PPOPP 1991, 1991] *Proceedings of the Third ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ACM SIGPLAN Notices 26(7), Williamsburg, VA, USA, April 1991.

[PPOPP 1993, 1993] *Proceedings of the Fourth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ACM SIGPLAN Notices 28(7), San Diego, CA, USA, May 1993.

[PPOPP 1995, 1995] *Proceedings of the Fifth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ACM SIGPLAN Notices 30(8), Santa Barbara, CA, USA, July 1995.

[PPOPP 1997, 1997] *Proceedings of the Sixth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ACM SIGPLAN Notices 32(7), Las Vegas, NV, USA, June 1997.

[PPOPP 1999, 1999] *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ACM SIGPLAN Notices 34(8), Atlanta, GA, USA, May 1999.

[PPOPP 2001, 2001] *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ACM SIGPLAN Notices 36(7), Snowbird, UT, USA, June 2001.

[PPPJ 2002, 2002] *Proceedings of the ACM International Symposium on Principles and Practice of Programming in Java*, volume 25 of *ACM International Conference Proceeding Series*, Dublin, Ireland, June 2002.

[PPPJ 2003, 2003] *Proceedings of the ACM International Symposium on Principles and Practice of Programming in Java*, volume 42 of *ACM International Conference Proceeding Series*, Kilkenny City, Ireland, June 2003.

[PPPJ 2004, 2004] *Proceedings of the ACM International Symposium on Principles and Practice of Programming in Java*, volume 91 of *ACM International Conference Proceeding Series*, Las Vegas, NV, USA, June 2004.

[PPPJ 2006, 2006] *Proceedings of the ACM International Symposium on Principles and Practice of Programming in Java*, volume 178 of *ACM International Conference Proceeding Series*, Mannheim, Germany, September 2006.

[PPPJ 2007, 2007] *Proceedings of the ACM International Symposium on Principles and Practice of Programming in Java*, volume 272 of *ACM International Conference Proceeding Series*, Lisbon, Portugal, September 2007.

[PPPJ 2008, 2008] *Proceedings of the ACM International Symposium on Principles and Practice of Programming in Java*, volume 347 of *ACM International Conference Proceeding Series*, Modena, Italy, September 2008.

[PPPJ 2010, 2010] *Proceedings of the ACM International Symposium on Principles and Practice of Programming in Java*, volume 347 of *ACM International Conference Proceeding Series*, Vienna, Austria, September 2010.

[Prakash *et al.*, 1994] S. Prakash, Y.-H. Lee, and T. Johnson. A nonblocking algorithm for shared queues using compare-and-swap. *IEEE Transactions on Computers*, 43(5):548–559, May 1994.

[Prensa Nieto and Esparza, 2000] L. Prensa Nieto and J. Esparza. Verifying single and multi-mutator garbage collectors with Owicki-Gries in Isabelle/HOL. In M. Nielsen and B. Rovan, editors, *Mathematical Foundations of Computer Science (MFCS 2000)*, volume 1893 of *Lecture Notes in Computer Science*, pages 619–628. Springer-Verlag, 2000.

[Printezis and Cutts, 1996] Tony Printezis and Quentin Cutts. Measuring the allocation rate of Napier88. Technical report, Department of Computer Science, University of Glasgow, November 1996.

[Printezis and Detlefs, 2000] Tony Printezis and David Detlefs. A generational mostly-concurrent garbage collector. In Chambers and Hosking [Chambers and Hosking2000], pages 143–154.

[Printezis and Garthwaite, 2002] Tony Printezis and Alex Garthwaite. Visualising the Train garbage collector. In Boehm and Detlefs [Boehm and Detlefs2002], pages 100–105.

[Printezis and Jones, 2002a] Tony Printezis and Richard Jones. GCspy: An adaptable heap visualisation framework. Technical Report 5–02, University of Kent, March 2002. Also University of Glasgow Technical Report.

[Printezis and Jones, 2002b] Tony Printezis and Richard Jones. GCspy: An adaptable heap visualisation framework. In OOPSLA 2002 [OOPSLA 20022002], pages 343–358.

[Printezis *et al.*, 1997] Tony Printezis, Malcolm P. Atkinson, Laurent Daynès, Susan Spence, and Pete Bailey. The design of a new persistent object store for PJama. In *Proceedings of the Second International Workshop on Persistence and Java (PJW2)*, Half Moon Bay, CA, USA, August 1997.

[Printezis, 1996] Tony Printezis. Disk garbage collection strategies for persistent Java. In *Proceedings of the First International Workshop on Persistence and Java*, Drymen, Scotland, September 1996.

[Printezis, 2000] Tony Printezis. *Management of Long-Running High-Performance Persistent Object Stores*. PhD thesis, University of Glasgow, May 2000.

[Printezis, 2001] Tony Printezis. Hot-Swapping between a Mark&Sweep and a Mark&Compact Garbage Collector in a Generational Environment. In JVM 2001 [JVM 20012001].

[Printezis, 2004] Tony Printezis. Garbage collection in the Java HotSpot virtual machine. http://www.devx.com/Java/Article/21977/, 2004.

[Printezis, 2006] Tony Printezis. On measuring garbage collection responsiveness. In *Science of Computer Programming* [Jones2006], pages 164–183.

[Protić *et al.*, 1995] J. Protić, M. Tomašević, and V. Milutinović. A survey of distributed shared memory systems. In *Proceedings of 28th Annual Hawaii International Conference on System Science*, volume I (architecture), pages 74–84, 1995.

[Protić *et al.*, 1997] J. Protić, M. Tomašević, and V. Milutinović. *Distributed Shared Memory: Concepts and Systems*. IEEE Press, August 1997.

[Przybylski *et al.*, 1988] Stephen A. Przybylski, Mark Horowitz, and John Hennessy. Performance tradeoffs in cache design. In *15th Annual International Symposium on Computer Architecture*, pages 290–298, Honolulu, Hawaii, June 1988.

[Przybylski, 1990a] Stephen Przybylski. The performance impact of block sizes and fetch strategies. In ISCA 1990 [ISCA 19901990], pages 160–169.

[Przybylski, 1990b] Steven A. Przybylski. *Cache and Memory Hierarchy Design: A Performance-Directed Approach*. Morgan Kaufman, Palo Alto, CA, 1990.

[Puaut, 1992] Isabelle Puaut. Distributed garbage collection of active objects with no global synchronisation. In Bekkers and Cohen [Bekkers and Cohen1992].

[Puaut, 1993] Isabelle Puaut. *Gestion d'objets actifs dans les systèmes distribués: problématique et mise en oeuvre*. PhD thesis, Université de Rennes I, 1993.

[Puaut, 1994a] Isabelle Puaut. A distributed garbage collector for active objects. In PARLE94 [PARLE941994]. Also INRIA UCIS-DIFUSION RR 2134.

[Puaut, 1994b] Isabelle Puaut. A distributed garbage collector for active objects. In OOPSLA 1994 [OOPSLA 19941994], pages 113–128.

[Puffitsch and Schoeberl, 2008] Wolfgang Puffitsch and Martin Schoeberl. Non-blocking root scanning for real-time garbage collection. In JTRES 2008 [JTRES 20082008], pages 68–76.

[Puffitsch, 2008] Wolfgang Puffitsch. Decoupled root scanning in multi-processor systems. In *CASES '08: Proceedings of the 2008 International Conference on Compilers, Architectures and Synthesis for Embedded Systems*, pages 91–98, Atlanta, GA, USA, 2008. ACM Press.

[Pugh and Chambers, 2004] William Pugh and Craig Chambers, editors. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 39(6), Washington, DC, USA, June 2004.

[Purdom and Stigler, 1970] P. W. Purdom and S. M. Stigler. Statistical properties of the buddy system. *Journal of the ACM*, 17(4):683–697, October 1970.

[Purdom *et al.*, 1971] P. W. Purdom, S. M. Stigler, and Tat-Ong Cheam. Statistical investigation of three storage allocation algorithms. *BIT*, 11:187–195, 1971.

[Purify, 1992] Pure Software, Los Altos, CA. *Purify*, 1992.

[Puschner *et al.*, 2003] P. Puschner, T. Nakajima, and A. Ghafoor, editors. *Proceedings of the 6th International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'03)*. IEEE Press, May 2003.

[Puzak, 1985] Thomas R. Puzak. *Analysis of Cache Replacement Algorithms*. PhD thesis, University of Massachussetts, Department of Electrical and Computer Engineering, February 1985.

[Qian and Hendren, 2002] Feng Qian and Laurie Hendren. An adaptive, region-based allocator for Java. In Boehm and Detlefs [Boehm and Detlefs2002], pages 127–138. Sable Technical Report 2002–1 provides a longer version.

[Qian *et al.*, 2002] Yang Qian, Witawas Srisa-an, T. Skotiniotis, and J. Morris Chang. Java virtual machine timing probes — a study of object life span and GC. In *Proceedings of 21st IEEE International Performance, Computing and Communications Conference (IPCCC)*, April 2002.

[Queinnec and Moreau, 1999] Christian Queinnec and Luc Moreau. Graceful disconnection. In Takayasu Ito and Taiichi Yuasa, editors, *Parallel and Distributed Computing for Symbolic and Irregular Applications, PDCSIA'99*, pages 242–252, Sendai, Japan, July 1999. World Scientific Publishing.

[Queinnec *et al.*, 1989] Christian Queinnec, Barbara Beaudoing, and Jean-Pierre Queille. Mark DURING Sweep rather than Mark THEN Sweep. *Lecture Notes in Computer Science*, 365:224–237, 1989.

[Queinnec, 1988] Christian Queinnec. Dynamic extent objects. *Lisp Pointers*, 2(1), 1988.

[Queinnec, 1994] Christian Queinnec. Sharing mutable objects and controlling groups of tasks in a concurrent and distributed language. In Takayasu Ito and Akinori Yonezawa, editors, *Proceedings of the Workshop on Theory and Practice of Programming (TPPP'94)*, volume 700 of *Lecture Notes in Computer Science*, pages 70–93, Sendai, Japan, November 1994. Springer-Verlag.

[Rafkind *et al.*, 2009] Jon Rafkind, Adam Wick, John Regehr, and Matthew Flatt. Precise garbage collection for C. In Kolodner and Steele [Kolodner and Steele2009], pages 39–48.

[Ram and Patel, 1984] Ashwin Ram and Janak H. Patel. Parallel garbage collection without synchronization overhead. Technical Report CSG-35, University of Illinois, 1984.

[Ram and Patel, 1985] Ashwin Ram and Janak H. Patel. Parallel garbage collection without synchronization overhead. In ISCA 1985 [ISCA 19851985], pages 84–90.

[Ramachandran *et al.*, 2006] Umakishore Ramachandran, Kathleen Knobe, Nissim Harel, and Hasnain A. Mandviwala. Distributed garbage collection algorithms for timestamped data. *IEEE Transactions on Parallel and Distributed Systems*, 17(10):1057–1071, October 2006.

[Ramakrishna, 2002] Y. Srinivas Ramakrishna. Automatic memory management in the Java HotSpot virtual machine. In *JavaOne Conference*, 2002.

[Ramesh and Mehndiratta, 1983] S. Ramesh and S. L. Mehndiratta. The liveness property of on-the-fly garbage collector — a proof. *Information Processing Letters*, 17(4):189–195, November 1983.

[Ramsey *et al.*, 2001] Norman Ramsey, Simon L. Peyton Jones, C. Lindig, T. Nordin, D. Oliva, and P. Nogueira Iglesias. *C-- Reference Manual*, November 2001.

[Rana, 1983] S. P. Rana. A distributed solution to the distributed termination problem. *Information Processing Letters*, 17:43–46, July 1983.

[Randell and Kuehner, 1968] Brian Randell and C. J. Kuehner. Dynamic storage allocation systems. *Communications of the ACM*, 12(7):297–306, May 1968.

[Randell, 1969] Brian Randell. A note on storage fragmentation and program segmentation. *Communications of the ACM*, 12(7):365–372, July 1969.

[Rao, 1978] G. S. Rao. Performance analysis of cache memories. *Journal of the ACM*, 25(3):378–395, July 1978.

[Rashid *et al.*, 1987] Richard Rashid, Avadis Tevanian, Michael Young, David Golub, Robert Baron, David Black, William Bolosky, and Jonathan Chew. Machine-independent virtual memory management for paged uniprocessor and multiprocessor architectures. In ASPLOS 1987 [ASPLOS 19871987], pages 31–39.

[Rathi *et al.*, 1987] B. D. Rathi, J. C. Browne, and G. J. Lipovski. Design of a self-managing secondary memory. In *Proceedings of the Twentieth Hawaii International Conference on System Sciences 1987. (Volume 1 = Architecture, Decision Support Systems and Knowledge-Based Systems.) Kailua-Kona, HI, USA*, pages 293–302, North Hollywood, CA, USA, January 6–9 1987. Western Periodicals Co.

[Rau, 1977] B. R. Rau. *Program Behavior and the Performance of Memory Systems*. PhD thesis, Stanford University, 1977.

[Ravindar and Srikant, 2005] A. Ravindar and Y.N. Srikant. Static analysis for identifying and allocating clusters of immortal objects. In *.NET Technologies 2005*, Plzen, Czech Republic, 2005.

[Reddy, 2004] Uday Reddy. Semantic models of storage. In SPACE 2004 [SPACE 20042004]. Invited talk.

[Reed and Kanodia, 1979] David P. Reed and Rajendra K. Kanodia. Synchronization with eventcounts and sequencers. *Communications of the ACM*, 22(2):115–123, February 1979.

[Reeves, 1979] C. M. Reeves. Free store distribution under random-fit allocation. *Computer Journal*, 22(4):346–351, November 1979.

[Reeves, 1980] C. M. Reeves. Free store distribution under random-fit allocation: Part 2. *Computer Journal*, 23(4):298–306, November 1980.

[Reeves, 1982] C. M. Reeves. A lumped-state model of clustering in dynamic storage allocation. *Computer Journal*, 27(2):135–142, 1982.

[Reeves, 1983] C. M. Reeves. Free store distribution under random-fit allocation, part 3. *Computer Journal*, 26(1):25–35, February 1983.

[Reichenbach *et al.*, 2010] C. Reichenbach, Eddie Aftandilian, Nl Immerman, Sam Guyer, and Yannis Smaragdakis. What can the GC compute efficiently? a language for heap assertions at GC time. In OOPSLA 2010 [OOPSLA 20102010].

[Reid *et al.*, 1999] Alastair Reid, John McCorquodale, Jason Baker, Wilson Hsieh, and Joseph Zachary. The need for predictable garbage collection. In *Proceedings of the ACM SIGPLAN Workshop on Compiler Support for System Software (WCSSS'99)*, May 1999.

[Reingold, 1973] E. M. Reingold. A non-recursive list moving algorithm. *Communications of the ACM*, 16(5):305–307, May 1973.

[Reinhold, 1993] Mark B. Reinhold. *Cache Performance of Garbage-Collected Programming Languages*. PhD thesis, MIT Laboratory for Computer Science, September 1993. Also Technical Memo MIT/LCS/TR–581.

[Reinhold, 1994] Mark B. Reinhold. Cache performance of garbage-collected programs. In PLDI 1994 [PLDI 19941994], pages 206–217.

[Reppy and Gansner, 1986] John H. Reppy and E. R. Gansner. Pegasus: A foundation for programming environments. Technical memorandum, AT&T Bell Laboratories, December 1986. An earlier version appeared in Proceedings of Second ACM/SIGSOFT/SIGPLAN Symposium on Practical Software Development Environments, December 1986, 218–227.

[Reppy, 1993] John H. Reppy. A high-performance garbage collector for Standard ML. Technical memorandum, AT&T Bell Laboratories, Murray Hill, NJ, December 1993.

[Reynolds and O'Hearn, 2001] John C Reynolds and Peter O'Hearn. Reasoning about shared mutable data structure. In SPACE 2001 [SPACE 20012001]. Invited talk.

[Ribeiro *et al.*, 1989] J. C. D. F. Ribeiro, C. D. Stormon, J. V. Oldfield, and M. R. Brule. Content-addressable memories applied to execution of logic programs. *IEE Proceedings, Part E: Computers and Digital Techniques*, 136(5):383–388, 1989.

[Richer and Shapiro, 2001] Nicolas Richer and Marc Shapiro. The memory behaviour of the WWW, or the WWW considered as a persistent store. In Kirby et al. [Kirby *et al.*2001], pages 136–146.

[Richer, 2002] Nicolas Richer. *Stratégies de Gestion Mémoire dans les Mémoires d'Objets Persistantes Automatiques Partitionnées*. PhD thesis, Université Pierre et Marie Curie — Paris VI, May 2002.

[Richter, 2000a] Jeffrey Richter. Garbage collection – part 2: Automatic memory management in the Microsoft .NET framework. *MSDN Magazine*, 15(13):82–92, December 2000.

[Richter, 2000b] Jeffrey Richter. Garbage collection: Automatic memory management in the Microsoft .NET framework. *MSDN Magazine*, 15(11):82–92, November 2000.

[Ridoux, 1987] Olivier Ridoux. Deterministic and stochastic modeling of parallel garbage collection: Towards real-time criteria. In *Computer Architecture News. The 14th Annual International Symposium on Computer Architecture, Pittsburgh, Pennsylvania, June 2–5, 1987*, pages 128–136. ACM Press, 1987.

[Ripley *et al.*, 1978] G. David Ripley, Ralph E. Griswold, and David R. Hanson. Performance of storage management in an implementation of SNOBOL4. *ACM Transactions on Software Engineering*, SE–4(2):130–137, March 1978.

[Ritzau and Fritzson, 2002] Tobias Ritzau and Peter Fritzson. Decreasing memory overhead in hard real-time garbage collection. In A. Sangiovanni-Vincentelli and J. Sifakis, editors, *Second International Workshop on Embedded Software (EMSOFT '02)*, volume 2491 of *Lecture Notes in Computer Science*, Grenoble, October 2002. Springer.

[Ritzau, 1999a] Tobias Ritzau. Real-time reference counting — automatic memory management with short and predictable interruptions. In *Pproceedings of the Svenska Nationella Realtidsföreningen (SNART) Conference*, Linköping, August 1999.

[Ritzau, 1999b] Tobias Ritzau. Real-time reference counting for RT-Java. Master's thesis, Linköping University, March 1999. Licenciate thesis. In Linköping Studies in Science and Technology, No. 748.

[Ritzau, 2000] Tobias Ritzau. Real-time reference counting. In *Java for Embedded Systems workshop*, London, May 2000.

[Ritzau, 2001] Tobias Ritzau. Hard real time reference counting without external fragmentation. In *Proceedings of the JOSES (Java Optimization Strategies for Embedded Systems) workshop at ETAPS 2001*, Genoa, Italy, 2001.

[Ritzau, 2003] Tobias Ritzau. *Memory Efficient Hard Real-Time Garbage Collection*. PhD thesis, Linköping University, May 2003.

[Robertson and Devarakonda, 1990] J. Robertson and M. Devarakonda. Data cache management using frequency-based replacement. In *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*. ACM Press, 1990.

[Robertson, 1979] Edward L. Robertson. Code generation and storage allocation for machines with span-dependent instructions. *ACM Transactions on Programming Languages and Systems*, 1(1):71–83, July 1979.

[Robertz and Henriksson, 2003] Sven Gestegøard Robertz and Roger Henriksson. Time-triggered garbage collection: Robust and adaptive real-time GC scheduling for embedded systems. In LCTES 2003 [LCTES 20032003], pages 93–102.

[Robertz, 2002] Sven Robertz. Applying priorities to memory allocation. In Boehm and Detlefs [Boehm and Detlefs2002], pages 1–11.

[Robertz, 2003] Sven Gestegøard Robertz. Flexible automatic memory management for real-time and embedded systems. Master's thesis, Lund University, 2003. Lic. eng. thesis.

[Robson, 1971] J. M. Robson. An estimate of the store size necessary for dynamic storage allocation. *Journal of the ACM*, 18(3):416–423, July 1971.

[Robson, 1973] J. M. Robson. An improved algorithm for traversing binary trees without auxiliary stack. *Information Processing Letters*, 2(1):12–14, March 1973.

[Robson, 1974] J. M. Robson. Bounds for some functions concerning dynamic storage allocation. *Journal of the ACM*, 21(3):419–499, July 1974.

[Robson, 1977a] J. M. Robson. A bounded storage algorithm for copying cyclic structures. *Communications of the ACM*, 20(6):431–433, June 1977.

[Robson, 1977b] J. M. Robson. Worst case fragmentation of first fit and best fit storage allocation s trategies. *Computer Journal*, 20(3):242–244, August 1977.

[Robson, 1980] J. M. Robson. Storage allocation is NP-hard. *Information Processing Letters*, 11(3):119–125, November 1980.

[Rochfeld, 1971] Arnold Rochfeld. New LISP techniques for a paging environment. *Communications of the ACM*, 14(12):791–795, December 1971.

[Rodrigues and Jones, 1996] Helena C. C. D. Rodrigues and Richard E. Jones. A cyclic distributed garbage collector for Network Objects. In Babaoglu and Marzullo [Babaoglu and Marzullo1996], pages 123–140.

[Rodrigues and Jones, 1998] Helena C. C. D. Rodrigues and Richard E. Jones. Cyclic distributed garbage collection with group merger. In Jul [Jul1998], pages 249–273. Also UKC Technical report 17–97, December 1997.

[Rodrigues, 1998] Helena C.C.D. Rodrigues. *Cyclic Distributed Garbage Collection*. PhD thesis, Computing Laboratory, The University of Kent at Canterbury, 1998.

[Rodriguez-Rivera and Russo, 1997] Gustavo Rodriguez-Rivera and Vince Russo. Non-intrusive cloning garbage collection with stock operating system support. *Software Practice and Experience*, 27(8), August 1997.

[Rodriguez-Rivera *et al.*, 1998] Gustavo Rodriguez-Rivera, Michael Spertus, and Charles Fiterman. A non-fragmenting, non-moving garbage collector. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 79–85.

[Rodriguez-Rivera *et al.*, 2000] Gustavo Rodriguez-Rivera, Mike Spertus, and Charles Fiterman. Conservative garbage collection for general memory allocators. In Chambers and Hosking [Chambers and Hosking2000], pages 71–79.

[Rodriguez-Riviera and Russo, 1997] Gustavo Rodriguez-Riviera and Vince Russo. Cyclic distributed garbage collection without global synchronization in CORBA. In Dickman and Wilson [Dickman and Wilson1997].

[Rodriguez-Riviera, 1995] Gustavo Rodriguez-Riviera. Cyclic distributed garbage collection without global synchronisation, 1995. PhD preliminary examination report.

[Röjemo and Runciman, 1996] Niklas Röjemo and Colin Runciman. Lag, drag, void, and use: heap profiling and space-efficient compilation revisited. In ICFP 1996 [ICFP 19961996], pages 34–41.

[Röjemo, 1992] Niklas Röjemo. A concurrent generational garbage collector for a parallel graph reducer. In Bekkers and Cohen [Bekkers and Cohen1992].

[Röjemo, 1993] Niklas Röjemo. Generational garbage collection is leak-prone. Draft paper, Department of Computer Science, Chalmers University, January 1993.

[Röjemo, 1994] Niklas Röjemo. nhc: A space-efficient haskell compiler. In *Proceedings of the workshop on Implementation of Functional Languages*, School of Information Systems, Univ. of East Anglia, Norwich, September 1994.

[Röjemo, 1995a] Niklas Röjemo. *Garbage Collection, and Memory Efficiency, in Lazy Functional Languages*. PhD thesis, Chalmers University of Technology, Goteborg, Sweden, 1995.

[Röjemo, 1995b] Niklas Röjemo. Generational garbage collection without temporary space leaks for lazy functional languages. In Baker [Baker1995a].

[Röjemo, 1995c] Niklas Röjemo. Highlights from nhc – a space-efficient Haskell compiler. In FPCA 1995 [FPCA 19951995].

[Ronsse and De Bosschere, 1998] M. Ronsse and K. De Bosschere. JiTi: Tracing memory references for data race detection. In E. D'Hollander, F.J. Joubert, and U. Trottenberg, editors, *Parallel Computing: Fundamentals, Applications and New Directions*, volume 12 of *Advances in Parallel Computing*, pages 327–334. North Holland, February 1998.

[Rose and Muller, 1992] John H. Rose and Hans Muller. Integrating the Scheme and C languages. In LFP 1992 [LFP 19921992], pages 247–259.

[Rosenberg and Keedy, 1987] John Rosenberg and J. L. Keedy. Object management and addressing in the MONADS architecture. In Carrick and Cooper [Carrick and Cooper1987].

[Rosenberg and Koch, 1989] John Rosenberg and David Koch, editors. *Proceedings of the Third International Workshop on Persistent Object Systems (January, 1989)*, Workshops in Computing, Newcastle, NSW, Australia, 1989. Springer.

[Rosenberg *et al.*, 1990] J. Rosenberg, F. A. Henskens, A. L. Brown, Ron Morrison, and David Munro. Stability in a persistent store based on a large virtual memory. In *International Workshop on Architectural Support for Security and Pe rsistence of Information*, pages 229–245. Springer Verlag and the British Computer Society, 1990.

[Rosenberg, 1991] John Rosenberg. Architectural support for persistent objects. In Cabrera et al. [Cabrera *et al.*1991], pages 48–60.

[Ross, 1967] D. T. Ross. The AED free storage package. *Communications of the ACM*, 10(8):481–492, August 1967.

[Ross, 1983] R. A. Ross. A garbage collecting associative memory for interactive database systems. In P. Degano and E. Sandewall, editors, *Integrated interactive computing systems. Proceedings of the European Conference ECICS'82 (Stresa, Italy, Sept. 1–3, 1982)*, pages 109–123. Elsevier-North Holland, 1983.

[Roth and Wise, 1998] David J. Roth and David S. Wise. One-bit counts between unique and sticky. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 49–56.

[Roth *et al.*, 1998] Amir Roth, Andreas Moshovos, and Gurindar S. Sohi. Dependence based prefetching for linked data structures. In ASPLOS 1998 [ASPLOS 19981998], pages 115–126.

[Roussel, 1975] P. Roussel. Prolog: Manuel de référence et d'utilisation. Technical report, G.I.A. Université Aix-Marseille, 1975.

[Rovner *et al.*, 1985] Paul Rovner, Roy Levin, and John Wick. On extending Modula-2 for building large, integrated systems. Technical Report 3, DEC Systems Research Center, Palo Alto, CA, Palo Alto, CA, 1985.

[Rovner, 1985] Paul Rovner. On adding garbage collection and runtime types to a strongly-typed, statically-checked, concurrent language. Technical Report CSL–84–7, Xerox PARC, Palo Alto, CA, July 1985.

[Roy *et al.*, 1998] P. Roy, S. Seshadri, A. Silberschatz, S. Sudarshan, and S. Ashwin. Garbage collection in object-oriente databases using transactional cyclic reference counting. *VLDB Journal: Very Large Databases*, 7(3):129–193, 1998.

[RTCSA 1999, 1999] IEEE Press. *Proceedings of the Sixth International Workshop on Real-Time Computing Systems and Applications (RTCSA)*, Hong Kong, 1999. IEEE Computer Society Press.

[RTCSA 2000, 2000] IEEE Press. *Proceedings of the Seventh International Workshop on Real-Time Computing Systems and Applications (RTCSA)*. IEEE Computer Society Press, 2000.

[RTCSA 2002, 2002] IEEE Press. *Proceedings of the NEighthInternational Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE Computer Society Press, March 2002.

[RTCSA 2003, 2003] IEEE Press. *Proceedings of the Ninth International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE Computer Society Press, August 2003.

[RTCSA 2005, 2005] IEEE Press. *Proceedings of the 11th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE Computer Society Press, August 2005.

[RTCSA 2007, 2005] IEEE Press. *Proceedings of the 13th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Daegu, Korea, August 2005. IEEE Computer Society Press.

[Ruby, 1987] J. Ruby. Liveness property of a parallel algorithm. *Information Processing Letters*, 24(4):275–277, 1987.

[Rudalics, 1986] M. Rudalics. Distributed copying garbage collection. In LFP 1986 [LFP 19861986], pages 364–372.

[Rudalics, 1990] M. Rudalics. Correctness of distributed garbage collection algorithms. Technical Report 90–40.0, Johannes Kepler Universität, Linz, 1990.

[Ruf, 2000] Erik Ruf. Effective synchronization removal for Java. In PLDI 2000 [PLDI 20002000], pages 208–218. Escape analysis based on type unification; [Steensgaard, 2000] says it's fast.

[Ruggieri and Murtagh, 1988] Christina Ruggieri and Thomas P. Murtagh. Lifetime analysis of dynamically allocated objects. In POPL 1988 [POPL 19881988], pages 285–293.

[Ruggieri, 1987] Christina Ruggieri. *Dynamic Memory Allocation Techniques Based on the Lifetimes of Objects*. PhD thesis, Purdue University, West Lafayette, Indiana, August 1987.

[Runciman and Röjemo, 1995] Colin Runciman and Niklas Röjemo. Lag, drag and post-mortem heap profiling. In *Implementation of Functional Languages Workshop*, Bøastad, Sweden, September 1995.

[Runciman and Röjemo, 1996a] Colin Runciman and Niklas Röjemo. Heap profiling for space efficiency. In J. Launchbury, E. Meijer, and T. Sheard, editors, *Second International School on Advanced Functional Programming*, volume 1129 of *Lecture Notes in Computer Science*, pages 34–41, Olympia, WA, August 1996. Springer-Verlag.

[Runciman and Röjemo, 1996b] Colin Runciman and Niklas Röjemo. Two-pass heap profiling – a matter of life and death. In W. Kluge, editor, *Selected Papers from the Eighth International Workshop on the Implementation of Functional Languages*, volume 1168 of *Lecture Notes in Computer Science*, pages 222–232, Bonn-Bad-Godesberg, September 1996. Springer-Verlag.

[Runciman and Wakeling, 1992] Colin Runciman and David Wakeling. Heap profiling of lazy functional programs. Technical Report YCS-92-172, University of York, 1992.

[Runciman and Wakeling, 1993a] Colin Runciman and David Wakeling. Heap profiling of a lazy functional compiler. In GWFP 1993 [GWFP 19931993], pages 203–214.

[Runciman and Wakeling, 1993b] Colin Runciman and David Wakeling. Heap profiling of lazy functional programs. *Journal of Functional Programming*, 3(2):217–245, April 1993.

[Runciman, 1995] Colin Runciman. New dimensions in heap profiling. Technical Report YCS-95-256, University of York, 1995.

[Runciman, 1996] Colin Runciman. New dimensions in heap profiling. *Journal of Functional Programming*, 6(4):587–620, 1996.

[Runciman, 2001] Colin Runciman. Heap profiling for theoreticians. In SPACE 2001 [SPACE 20012001]. Invited talk.

[Russell, 1977] D. L. Russell. Internal fragmentation in a class of buddy systems. *SIAM J. Comput.*, 6(4):607–621, December 1977.

[Russinoff, 1994] David M. Russinoff. A mechanically verified incremental garbage collector. *Formal Aspects of Computing*, 6:359–390, 1994.

[Russo, 1991] Vincent F. Russo. Garbage collecting and object-oriented operating system kernel. In Wilson and Hayes [Wilson and Hayes1991a].

[Ryu and Neuman, 1998] Sung-Wook Ryu and B. Clifford Neuman. Garbage collection for distributed persistent objects. In *Workshop on Compositional Software Architectures*, Monterey, CA, January 1998.

[Sachindran and Moss, 2003] Narendran Sachindran and Eliot Moss. MarkCopy: Fast copying GC with less space overhead. In OOPSLA 2003 [OOPSLA 20032003].

[Sachindran *et al.*, 2004] Narendran Sachindran, J. Eliot B. Moss, and Emery D. Berger. MC^2: High-performance garbage collection for memory-constrained environments. In OOPSLA 2004 [OOPSLA 20042004], pages 81–98.

[Sagiv *et al.*, 1999] M. Sagiv, T. Reps, and R. Wilhelm. Parametric shape analysis via 3-valued logic. In POPL 1999 [POPL 19991999]. statically analyse heap paths.

[Sagonas and Wilhelmsson, 2004] Konstantinos Sagonas and Jesper Wilhelmsson. Message analysis-guided allocation and low-pause incremental garbage collection in a concurrent language. In Bacon and Diwan [Bacon and Diwan2004], pages 1–12.

[Sagonas and Wilhelmsson, 2006a] Konstantinos Sagonas and Jesper Wilhelmsson. Efficient memory management for concurrent programs that use message passing. In *Science of Computer Programming* [Jones2006], pages 98–121.

[Sagonas and Wilhelmsson, 2006b] Konstantinos Sagonas and Jesper Wilhelmsson. Mark and split. In Petrank and Moss [Petrank and Moss2006], pages 29–39.

[Sahlin and Carlsson, 1991] Dan Sahlin and Mats Carlsson. Variable shunting for the WAM. Research report SICS/R91–07, Swedish Institute of Computer Science, March 1991.

[Sahlin, 1987] Dan Sahlin. Making garbage collection independent of the amount of garbage. Research Report SICS/R–87/87008, SICS, 1987. Appendix to SICS research report R86009 "Garbage collection for Prolog based on WAM.".

[Salagnac *et al.*, 2005] Guillaume Salagnac, S. Yovine, and D. Garbervetsky. Fast escape analysis for region-based memory management. In *Proceedings of 1st International Workshop on Abstract Interpretation of Object-Oriented Languages (AIOOL05)*, Paris, January 2005.

[Salagnac, 2004] Guillaume Salagnac. Gestion automatique de la mémoire dynamique pour des programmes Java temps-réel embarqués. Report de d.e.a., Université Joseph Fourier, June 2004.

[Salcianu, 2001] Alexandru Salcianu. Pointer analysis and its applications for Java programs. Master's thesis, MIT Press, September 2001.

[Salkild, 1987] Jon D. Salkild. Implementation and analysis of two reference counting algorithms. Master's thesis, University College, London, 1987.

[Sallé, 1984] Patrick Sallé. Syntaxe et sémantique de PLASMA et ALOG. Technical report, LSI-ENSEEIHT, Toulouse, 1984.

[Samples *et al.*, 1986] A. Dain Samples, David M. Ungar, and Paul Hilfinger. SOAR: Smalltalk without bytecodes. In OOPSLA 1986 [OOPSLA 19861986], pages 107–118.

[Samples, 1992] A. Dain Samples. Garbage collection-cooperative C++. In Bekkers and Cohen [Bekkers and Cohen1992].

[Sanchez *et al.*, 2001] Alfonso Sanchez, Luís Veiga, and Paulo Ferreira. Distributed garbage collection for wide area replicated memory. In *Proceedings of the Sixth USENIX Conference on Object-Oriented Technologies and Systems (COOTS'01)*, San Antonio, TX, January 2001.

[Sands, 2001] David Sands. Spikes and ballast: The algebra of space. In SPACE 2001 [SPACE 20012001].

[Sankaran, 1994a] Nandakumar Sankaran. A bibliography on garbage collection. Technical report, Clemson University, February 1994. A small collection!

[Sankaran, 1994b] Nandakumar Sankaran. A bibliography on garbage collection and related topics. *ACM SIGPLAN Notices*, 29(9):140–148, September 1994. A small collection!

[Sansom and Peyton Jones, 1993] Patrick M. Sansom and Simon L. Peyton Jones. Generational garbage collection for Haskell. In Hughes [Hughes1993].

[Sansom and Peyton Jones, 1994] Patrick M. Sansom and Simon L. Peyton Jones. Time and space profiling for non-strict, higher-order, functional languages. Research Report FP–1994–10, University of Glasgow, 1994.

[Sansom, 1991] Patrick M. Sansom. Dual-mode garbage collection. Technical Report CSTR 91–07, Department of Electronics and Computer Science, University of Southampton, June 1991. *Proceedings of Third International Workshop on Implementation of Functional Languages on Parallel Architectures*.

[Sansom, 1992] Patrick M. Sansom. Combining copying and compacting garbage collection. In Peyton Jones et al. [Peyton Jones *et al.*1992].

[Sansom, 1994] Patrick M. Sansom. *Execution Profiling for Non-Strict Functional Languages*. PhD thesis, University of Glasgow, 1994.

[Sarkar and Hall, 2005] Vivek Sarkar and Mary W. Hall, editors. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 40(6), Chicago, IL, USA, June 2005.

[Sarkis and Bielak, 1995] Jean-Pierre Sarkis and Richie Bielak. Implementing stacks. *Eiffel Outlook*, 4(6):6–9, September 1995.

[Sartor *et al.*, 2008a] Jennifer B. Sartor, Martin Hirzel, and Kathryn S. McKinley. No bit left behind: Limits of heap data compression. In Jones and Blackburn [Jones and Blackburn2008], pages 111–120.

[Sartor *et al.*, 2008b] Jennifer B. Sartor, Martin Hirzel, and Kathryn S. McKinley. No bit left behind: The limits of heap data compression (extended version). Technical Report TR–08–17, University of Texas at Austin, 2008.

[Sartor *et al.*, 2010] Jennifer B. Sartor, Stephen M. Blackburn, Daniel Frampton, Martin Hirzel, and Kathryn S. McKinley. Z-rays: Divide arrays and conquer speed and flexibility. In PLDI 2010 [PLDI 20102010], pages 471–482.

[Satishkumar, 1994] S. Satishkumar. Register allocation for accurate garbage collection of C++. Master's thesis, Iowa State University, July 1994. Technical report ISUTR 94–12.

[Saunders, 1974] Robert A. Saunders. The LISP system for the Q–32 computer. In Berkeley and Bobrow [Berkeley and Bobrow1974], pages 220–231.

[Schapiro, 1986] E. Schapiro, editor. *Proceedings of Third International Conference on Logic Programming*, volume 225, London, 1986. Springer-Verlag.

[Schelter and Ballantyne, 1988] W. F. Schelter and M. Ballantyne. Kyoto Common Lisp. *AI Expert*, 3(3):75–77, 1988.

[Schelvis and Bledoeg, 1988] M. Schelvis and E. Bledoeg. The implementation of a distributed Smalltalk. *Lecture Notes in Computer Science*, 322:212–232, 1988.

[Schelvis, 1989] M. Schelvis. Incremental distribution of timestamp packets — a new approach to distributed garbage collection. *ACM SIGPLAN Notices*, 24(10):37–48, 1989.

[Schimpf, 1990] J. Schimpf. Garbage collection for Prolog based on twin cells. In *2nd NACLP Workshop on Logic Programming Architectures and Implementations*. MIT Press, 1990.

[Schmidt and Nilsen, 1992] William J. Schmidt and Kelvin Nilsen. Experimental measurements of a real-time garbage collection architecture. Technical report, Iowa State University, 1992. Technical report ISUTR 92-26.

[Schmidt and Nilsen, 1994] William J. Schmidt and Kelvin D. Nilsen. Performance of a hardware-assisted real-time garbage collector. In ASPLOS 1994 [ASPLOS 19941994], pages 76–85.

[Schmidt, 1992] William J. Schmidt. *Issues in the Design and Implementation of a Real-Time Garbage Collection Architecture*. PhD thesis, Iowa State University, Ames, Iowa, 1992. Technical report ISUTR 92-25.

[Schneider *et al.*, 2001] Daniel Schneider, Bernd Mathiske, Matthias Ernst, and Matthew Seidl. Automatic persistent memory management for the Spotless Java virtual machine on the Palm connected organizer. In JVM 2001 [JVM 20012001].

[Schneider *et al.*, 2006] Scott Schneider, Christos Antonopoulos, and Dimitrios Nikolopoulos. Scalable locality-conscious multithreaded memory allocation. In Petrank and Moss [Petrank and Moss2006], pages 84–94.

[Schoeberl and Puffitsch, 2008] Martin Schoeberl and Wolfgang Puffitsch. Non-blocking object copy for real-time garbage collection. In JTRES 2008 [JTRES 20082008], pages 77–84.

[Schoeberl and Vitek, 2007] Martin Schoeberl and Jan Vitek. Garbage collection for safety critical Java. In JTRES 2007 [JTRES 20072007], pages 85–93.

[Schoeberl, 2006] Martin Schoeberl. Real-time garbage collection for Java. In ISORC 2006 [ISORC 20062006], pages 424–432.

[Schoeberl, 2010] Martin Schoeberl. Scheduling of hard real-time garbage collection. *Real-Time Systems*, 45(3):176–213, 2010.

[Schooler and Stamos, 1987] R. Schooler and James W. Stamos. Proposal for a small Scheme implementation. Technical Memo MIT/LCS/TM–267, MIT Laboratory for Computer Science, October 1987.

[Schorr and Waite, 1967] H. Schorr and W. Waite. An efficient machine independent procedure for garbage collection in various list structures. *Communications of the ACM*, 10(8):501–506, August 1967.

[Schreiner, 1994] Wolfgang Schreiner. Garbage collection on a stack. Technical report, Research Institute for Symbolic Computation (RISC-Linz), Johannes Kepler University, Linz, Austria, January 1994.

[Schulte, 1994] Wolfram Schulte. Deriving reference count garbage collectors. In *Proceedings of the 6th International Symposium on Programming Language Implementation and Logic Programming*, pages 102–116, September 1994.

[Schwartz, 1974] Jacob T. Schwartz. More on copy optimization of SETL programs. SETL Newsletter 131, Courant Inst. of Mathematical Sciences, New York University, June 1974.

[Schwartz, 1976a] Jacob T. Schwartz. A coarser, but simpler and considerably more efficient copy optimization. SETL Newsletter 176, Courant Inst. of Mathematical Sciences, New York University, August 1976.

[Schwartz, 1976b] Jacob T. Schwartz. A coarser, but simpler and considerably more efficient copy optimization technique. SETL Newsletter 176, Courant Inst. of Mathematical Sciences, New York University, August 1976.

[Schwartzbach and Ball, 2006] Michael I. Schwartzbach and Thomas Ball, editors. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, ACM SIGPLAN Notices 41(6), Ottawa, Canada, June 2006.

[Schwarz, 1978] Jerald Schwarz. Verifying the safe use of destructive operations in applicative programs. In *Program Transformations — Proceedings of the 3rd International Symposium on Programming*, pages 395–411, 1978.

[Seidl and Zorn, 1997] Matthew L. Seidl and Benjamin Zorn. Predicting references to dynamically allocated objects. Technical Report CU-CS-826-97, University of Colorado, January 1997.

[Seidl and Zorn, 1999a] Matthew L. Seidl and Benjamin Zorn. Implementing heap-object behavior prediction efficiently and effectively. Technical Report CU-CS-893-99, University of Colorado, dec 1999. Submitted to Software - Practice and Experience.

[Seidl and Zorn, 1999b] Matthew L. Seidl and Benjamin Zorn. Low cost methods for predicting heap object behavior. In *Second Workshop on Feedback Directed Optimization*, pages 83–90, Haifa, Israel, November 1999.

[Seliger, 1990] Robert Seliger. Extending C++ to support remote procedure call, concurrency, exception handling and garbage collection. In *Usenix C++ Conference Proceedings*, pages 241–264. USENIX Association, 1990.

[Seligmann and Grarup, 1995] Jacob Seligmann and Steffen Grarup. Incremental mature garbage collection using the train algorithm. In Nierstras [Nierstras1995], pages 235–252.

[Serrano and Boehm, 2000] Manuel Serrano and Hans-J Boehm. Understanding memory allocation of Scheme programs. In ICFP 2000 [ICFP 20002000].

[Serrano and Feeley, 1996] Manuel Serrano and Marc Feeley. Storage use analysis and its applications. In ICFP 1996 [ICFP 19961996].

[Serrano and Zhuang, 2009] Mauricio J. Serrano and Xiaotong Zhuang. Placement optimization using data context collected during garbage collection. In Kolodner and Steele [Kolodner and Steele2009], pages 69–78.

[Serrano, 1994] Manuel Serrano. *Vers un Compilation Portable* et *Performante des Langages Fonctionnels*. PhD thesis, Université Paris 6, December 1994.

[Seward, 1992] Julian Seward. Generational garbage collection for lazy graph reduction. In Bekkers and Cohen [Bekkers and Cohen1992].

[Sewe *et al.*, 2010] Andreas Sewe, Dingwen Yuan, Jan Sinschek, and Mira Mezini. Headroom-based pretenuring: Dynamically pretenuring objects that live "long enough". In PPPJ 2010 [PPPJ 20102010], pages 29–38.

[Shaham *et al.*, 2000] Ran Shaham, Elliot K. Kolodner, and Mooly Sagiv. On the effectiveness of GC in Java. In Chambers and Hosking [Chambers and Hosking2000], pages 12–17.

[Shaham *et al.*, 2001] Ran Shaham, Elliot K. Kolodner, and Mooly Sagiv. Heap profiling for space-efficient Java. In PLDI 2001 [PLDI 20012001], pages 104–113.

[Shaham *et al.*, 2002] Ran Shaham, Elliot Kolodner, and Mooly Sagiv. Estimating the impact of liveness information on space consumption in Java. In Boehm and Detlefs [Boehm and Detlefs2002], pages 64–75.

[Shaham *et al.*, 2003] R. Shaham, E. Yahav, E. Kolodner, and M. Sagiv. Establishing local temporal heap safety properties with applications to compile-time memory. In *Proceedings of Static Analysis Symposium, SAS'03*, volume 2694 of *Lecture Notes in Computer Science*, pages 483–503, June 2003.

[Shang, 1989] Heping Shang. Consistent global state algorithms and an application in distributed garbage collection. Master's thesis, Concordia University, Canada, 1989.

[Shankar *et al.*, 2008] Ajeet Shankar, Matthew Arnold, and Rastislav Bodik. Jolt: Lightweight dynamic analysis and removal of object churn. In OOPSLA 2008 [OOPSLA 20082008], pages 127–142.

[Shao and Lee, 2003] Zhong Shao and Peter Lee, editors. *Proceedings of the ACM SIGPLAN International Workshop on Types in Language Design and Implementation*, ACM SIGPLAN Notices 38(3), New Orleans, LA, USA, January 2003.

[Shapiro and Ferreira, 1994] Marc Shapiro and Paulo Ferreira. Larchant–RDOSS: a distributed shared persistent memory and its garbage collector. Technical report, Institut National de la Recherche en Informatique et Automatique, November 1994. Superseded by [Shapiro and Ferreira, 1995].

[Shapiro and Ferreira, 1995] Marc Shapiro and Paulo Ferreira. Larchant–RDOSS: a distributed shared persistent memory and its garbage collector. In J.-M. Hélary and M. Raymond, editors, *Workshop on Distributed Algorithms*, number 972 in Lecture Notes in Computer Science, pages 198–214, Le Mont Saint-Michel, September 1995.

[Shapiro *et al.*, 1990] Marc Shapiro, Olivier Gruber, and David Plainfossé. A garbage detection protocol for a realistic distributed object-support system. Rapports de Recherche 1320, INRIA-Rocquencourt, November 1990. Superseded by [Shapiro, 1991].

[Shapiro *et al.*, 1992a] Marc Shapiro, Peter Dickman, and David Plainfossé. Robust, distributed references and acyclic garbage collection. In *Symposium on Principles of Distributed Computing*, pages 135–146, Vancouver, Canada, August 1992. ACM Press. Superseded by [Shapiro *et al.*, 1992b].

[Shapiro *et al.*, 1992b] Marc Shapiro, Peter Dickman, and David Plainfossé. SSP chains: Robust, distributed references supporting acyclic garbage collection. Rapports de Recherche 1799, Institut National de la Recherche en Informatique et Automatique, November 1992. Also available as Broadcast Technical Report 1.

[Shapiro *et al.*, 1992c] Marc Shapiro, Julien Maisonneuve, and Pierre Collet. Implementing references as chains of links. In Cabrera et al. [Cabrera *et al.*1992].

[Shapiro *et al.*, 1994] Marc Shapiro, David Plainfossé, Paulo Ferreira, and Laurent Amsaleg. Some key issues in the design of distributed garbage collection and references. In *Unifying Theory and Practice in Distributed Systems*, Dagstuhl (Germany), September 1994.

[Shapiro *et al.*, 1997] Marc Shapiro, Sytse Kloosterman, and Fabio Riccardi. PerDiS — a persistent distributed store for cooperative applications. In *Proceedings of 3rd Cabernet Plenary Workshop*, Rennes (France), April 1997.

[Shapiro *et al.*, 2000] Marc Shapiro, Fabrice Le Fessant, and Paulo Ferreira. Recent advances in distributed garbage collection. In Krakowiak and Shrivastava [Krakowiak and Shrivastava2000], pages 104–126.

[Shapiro, 1991] Marc Shapiro. A fault-tolerant, scalable, low-overhead distributed garbage collection protocol. In *Proceedings of the Tenth Symposium on Reliable Distributed Systems*, Pisa, September 1991.

[Shapiro, 1993] Marc Shapiro. Flexible bindings for fine-grain and fragmented objects in distributed systems. Rapports de Recherche 2007, Institut National de la Recherche en Informatique et Automatique, 1993.

[Sharma and Soffa, 1991] Ravi Sharma and Mary Lou Soffa. Parallel generational garbage collection. In OOPSLA 1991 [OOPSLA 19911991], pages 16–32.

[Shaw, 1987] Robert A. Shaw. Improving garbage collector performance in virtual memory. Technical Report CSL–TR–87–323, Stanford University, March 1987. Also Hewlett-Packard Laboratories report STL–TM–87–05, Palo Alto, 1987.

[Shaw, 1988] Robert A. Shaw. *Empirical Analysis of a Lisp System*. PhD thesis, Stanford University, 1988. Technical Report CSL–TR–88–351.

[Shen and Martonosi, 2006] John Paul Shen and Margaret Martonosi, editors. *Proceedings of the Twelfth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGPLAN Notices 41(11), San Jose, CA, USA, October 2006.

[Shen and Peterson, 1974] K. K. Shen and J. L. Peterson. A weighted buddy method for dynamic storage allocation. *Communications of the ACM*, 17(10):558–562, October 1974.

[Shen *et al.*, 2004] Xipeng Shen, Yutao Zhong, and Chen Ding. Locality phase prediction. In Mukherjee and McKinley [Mukherjee and McKinley2004], pages 165–176.

[Shin and Malek, 1985] Heonshik Shin and Miroslaw Malek. Parallel garbage collection with associative tag. In *Proceedings of IEEE Conference on Parallel Processing*, pages 369–375, Chicago, 1985. IEEE Press.

[Shivaratri *et al.*, 1992] N.G. Shivaratri, P. Krueger, and M. Singhal. Load distributing for locally distributed systems. *Computer*, 25(12):33–44, December 1992.

[Shivers *et al.*, 1999] O. Shivers, J.W. Clark, and R. McGrath. Atomic heap transactions and fine-grain interrupts. In ICFP 1999 [ICFP 19991999].

[Shore, 1975] J. E. Shore. On the external storage fragmentation produced by first-fit and best-fit allocation strategies. *Communications of the ACM*, 18(8):433–440, August 1975.

[Shore, 1977] J. E. Shore. Anomalous behavior of the fifty-percent rule in dynamic memory allocation. *Communications of the ACM*, 20(11):558–562, November 1977.

[Shuf *et al.*, 2001] Yefim Shuf, Mauricio Serrano, Manish Gupta, and Jaswinder Pal Singh. Characterizing the memory behavior of Java workloads: A structured view and opportunities for optimizations. In *SIGMETRICS'01*, June 2001.

[Shuf *et al.*, 2002a] Yefim Shuf, Manish Gupta, Rajesh Bordawekar, and Jaswinder Pal Singh. Exploiting prolific types for memory management and optimizations. In POPL 2002 [POPL 20022002].

[Shuf *et al.*, 2002b] Yefim Shuf, Manish Gupta, Hubertus Franke, Andrew Appel, and Jaswinder Pal Singh. Creating and preserving locality of Java applications at allocation and garbage collection times. In OOPSLA 2002 [OOPSLA 20022002].

[Shultis, 1985] Jon. Shultis. Imminent garbage collection. Technical Report CU-CS-305-85, University of Colorado, Department of Computer Science, 1985.

[Siebert and Walter, 2001] Fridtjof Siebert and Andy Walter. Deterministic execution of Java's primitive bytecode operations. In JVM 2001 [JVM 20012001].

[Siebert, 1997] Fridtjof Siebert. Implementierung eines eiffel-compilers für sun/sparc. Master's thesis, Universitaet Stuttgart, 1997. Diplomarbeit 1484. In German.

[Siebert, 1998] Fridtjof Siebert. Guaranteeing non-disruptiveness and real-time deadlines in an incremental garbage collector. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 130–137.

[Siebert, 1999a] Fridtjof Siebert. Hard real-time garbage collection in the Jamaica Virtual Machine. In RTCSA 1999 [RTCSA 19991999].

[Siebert, 1999b] Fridtjof Siebert. Real-time garbage collection in multi-threaded systems on a single processor. In *20th IEEE Real-Time Systems Symposium (RTSS'99)*, Phoenix, Arizona, 1999.

[Siebert, 2000] Fridtjof Siebert. Eliminating external fragmentation in a non-moving garbage collector for Java. In *Compilers, Architectures and Synthesis for Embedded Systems (CASES2000)*, San Jose, November 2000.

[Siebert, 2001] Fridtjof Siebert. Constant-time root scanning for deterministic garbage collection. In *Tenth International Conference on Compiler Construction, (CC'01)*, Genoa, April 2001.

[Siebert, 2002] Fridtjof Siebert. *Hard Realtime Garbage Collection in Modern Object Oriented Programming Languages*. aicas Books, 2002.

[Siebert, 2004] Fridtjof Siebert. The impact of realtime garbage collection on realtime Java programming. In ISORC 2004 [ISORC 20042004], pages 33–40.

[Siebert, 2007] Fridtjof Siebert. Realtime garbage collection in the JamaicaVM 3.0. In JTRES 2007 [JTRES 20072007], pages 94–103.

[Siebert, 2008] Fridtjof Siebert. Limits of parallel marking collection. In Jones and Blackburn [Jones and Blackburn2008], pages 21–29.

[Siebert, 2010] Fridtjof Siebert. Concurrent, parallel, real-time garbage-collection. In jan Vitek and Lea [jan Vitek and Lea2010], pages 11–20.

[Siegwart and Hirzel, 2006] David Siegwart and Martin Hirzel. Improving locality with parallel hierarchical copying GC. In Petrank and Moss [Petrank and Moss2006], pages 52–63.

[SIGMOD 1989, 1989] *Proceedings of the ACM SIGMOD International Conference on Management of Data*, ACM SIGMOD Record 18(2), Snowbird, Utah, June 1989.

[SIGPLAN 1979, 1979] *Proceedings of the ACM SIGPLAN Symposium on Compiler Construction*, ACM SIGPLAN Notices 14(7), Denver, CO, USA, August 1979.

[SIGPLAN 1986, 1986] *Proceedings of the ACM SIGPLAN Symposium on Compiler Construction*, ACM SIGPLAN Notices 21(7), Palo Alto, CA, USA, June 1986. ACM Press.

[SIGPLAN 1987, 1987] *Proceedings of the Symposium on Interpreters and Interpretive Techniques*, ACM SIGPLAN Notices 22(7), St Paul, MN, USA, June 1987.

[Siklossy, 1972] L. Siklossy. Fast and readonly algorithms for traversing trees without an auxiliary stack. *Information Processing Letters*, 1(4):149–152, June 1972.

[Singer and Kirkham, 2006a] Jeremy Singer and Chris Kirkham. Visualized adaptive runtime subsystems. In *Proceedings of the 2006 ACM Symposium on Software Visualization*, pages 195–196, Brighton, UK, 2006.

[Singer and Kirkham, 2006b] Jeremy Singer and Chris C. Kirkham. Dynamic analysis of program concepts in Java. In PPPJ 2006 [PPPJ 20062006], pages 31–39.

[Singer *et al.*, 2007a] Jeremy Singer, Gavin Brown, Mikel Lujan, and Ian Watson. Towards intelligent analysis techniques for object pretenuring. In PPPJ 2007 [PPPJ 20072007].

[Singer *et al.*, 2007b] Jeremy Singer, Gavin Brown, Ian Watson, and John Cavazos. Intelligent selection of application-specific garbage collectors. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 91–102.

[Singer *et al.*, 2008] Jeremy Singer, Sebastien Marion, Gavin Brown, Richard Jones, Mikel Lújan, Chris Ryder, and Ian Watson. An information theoretic evaluation of software metrics for object lifetime prediction. In *2nd Workshop on Statistical and Machine Learning Approaches to Architectures and Compilation (SMART'08)*, page 15, Goteborg, Sweden, January 2008.

[Singer *et al.*, 2010] Jeremy Singer, Richard Jones, Gavin Brown, and Mikel Luján. The economics of garbage collection. In jan Vitek and Lea [jan Vitek and Lea2010], pages 103–112.

[Singh *et al.*, 1992] Jaswinder Pal Singh, Harold S. Stone, and Dominique Thiebaut. A model of workloads and its use in miss-rate prediction for fully-associative caches. *IEEE Transactions on Computers*, 41(7):811–825, July 1992.

[Singh, 1988] Ramanand Singh. A realization of multiprocessing garbage collection algorithm for rule-based expert systems. Master's thesis, West Virginia University, 1988.

[Singh, 1990] Tajinder P. Singh. Hardware design of a real-time copying garbage collection system. Master's thesis, Iowa State University, Department of Computer Science, August 1990.

[Singhal *et al.*, 1992] Vivek Singhal, Sheetal V. Kakkad, and Paul R. Wilson. Texas: an efficient, portable persistent store. In Albano and Morrison [Albano and Morrison1992], pages 11–33.

[Sirer *et al.*, 1996] Emin Gün Sirer, Stefan Savage, Przemyslaw Pardyak, Greg P. DeFouw, and Brian N. Bershad. Writing an operating system using Modula-3. In *Workshop on Compiler Support for Systems Software*, 1996.

[Skotiniotis and Chang, 2002] Therapon Skotiniotis and J. Morris Chang. Estimating internal memory fragmentation for Java programs. *Journal of Systems and Software*, 64(3):235–246, December 2002.

[Skubiszewski and Porteix, 1996] M. Skubiszewski and N. Porteix. GC-consistent cuts of databases. Rapport de Recherche 2681, Institut National de la Recherche en Infromatique et Automatique, Rocquencourt, April 1996.

[Skubiszewski and Valduriez, 1997] M. Skubiszewski and P. Valduriez. Concurrent garbage collection in O2. In M. Jarke, M.J. Carey, K.R. Dittrich, F.H. Lochovsky, P. Loucopoulos, and M.A. Jeusfeld, editors, *VLDB'97 Proceedings of 23rd International Conference on Very Large Databases*, pages 356–365, Athens, May 1997. Morgan Kaufman.

[Slater, 1991] Michael Slater. PA workstations set price/performance records. *Microprocessor Report*, 5(6), April 1991.

[Sleator and Tarjan, 1985] Daniel Dominic Sleator and Robert Endre Tarjan. Self-adjusting binary search trees. *Journal of the ACM*, 32(3):562–686, July 1985.

[Slusarek, 1987] Maciej Slusarek. An off-line storage allocation algorithm. *Information Processing Letters*, 24(2):71–75, January 1987.

[Smaragdakis, 2004] Yannis Smaragdakis. General adaptive replacement policies. In Bacon and Diwan [Bacon and Diwan2004], pages 108–119.

[Smetsers *et al.*, 1993] S. Smetsers, E. Barendsen, M. J. C. D van Eekelen, and R. Plasmeijer. Guaranteeing safe destructive updates through a type system with uniqueness information for graphs. Technical Report 93–4, University of Nijmegen, 1993.

[Smith and Morrisett, 1997] Frederick Smith and Greg Morrisett. Mostly copying collection: A viable alternative to conservative mark-sweep. Technical report, Cornell, 1997.

[Smith and Morrisett, 1998] Frederick Smith and Greg Morrisett. Comparing mostly-copying and mark-sweep conservative collection. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 68–78.

[Smith, 1976] Alan J. Smith. A modified working set paging algorithm. *IEEE Transactions on Software Engineering*, C-25(9):907–914, September 1976.

[Smith, 1978] Alan J. Smith. Sequential program prefetching in memory hierarchies. *IEEE Transactions on Computers*, 11(12):7–21, December 1978.

[Smith, 1982] Alan J. Smith. Cache memories. *Computing Surveys*, 14(3):473–530, September 1982.

[Smith, 1985] Alan J. Smith. Cache evaluation and the impact of workload choice. In ISCA 1985 [ISCA 19851985], pages 64–73.

[Smith, 1986] Alan J. Smith. Bibliography and readings on cpu cache memories and related topics. *Computer Architecture News*, 14(1):22–42, January 1986.

[Smith, 1987] Alan J. Smith. Line (block) size choice for cpu cache memories. *IEEE Transactions on Computers*, C-36(9):1063–1075, September 1987.

[Smith, 1989] Jane Smith. A garbage collector for FP9. Technical report, University of New Dundee, 1989.

[Snyder, 1979] A. Snyder. A machine architecture to support an object-oriented language. Technical Memo MIT/LCS/TR–209, MIT Laboratory for Computer Science, March 1979.

[Sobalvarro, 1988] Patrick Sobalvarro. A lifetime-based garbage collector for Lisp systems on general-purpose computers. Bachelor of Science thesis AITR-1417, MIT AI Lab, February 1988.

[Soffa, 2008] Mary Lou Soffa, editor. *Proceedings of the Fourteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM SIGPLAN Notices 43(3), Seattle, WA, USA, March 2008.

[Soman and Krintz, 2006] Sunil Soman and Chandra Krintz. Efficient and general on-stack replacement for aggressive program specialization. In *Proceedings of the International Conference on Software Engineering Research and Practice & Conference on Programming Languages and Compilers, Volume 2*, pages 925–932, Las Vegas, NV, June 2006. CSREA Press.

[Soman *et al.*, 2004a] Sunil Soman, Chandra Krintz, and David Bacon. Dynamic selection of application-specific garbage collectors. Technical Report 2004–09, UCSB, January 2004.

[Soman *et al.*, 2004b] Sunil Soman, Chandra Krintz, and David Bacon. Dynamic selection of application-specific garbage collectors. In Bacon and Diwan [Bacon and Diwan2004], pages 49–60.

[Soman *et al.*, 2006] Sunil Soman, Laurent Daynes, , and Chandra Krintz. Task-aware garbage collection in a multi-tasking virtual machine. In Petrank and Moss [Petrank and Moss2006], pages 64–73.

[Soman *et al.*, 2008] Sunil Soman, Chandra Krintz, and Laurent Daynes. MTM^2: Scalable memory management for multi-tasking managed runtime environments. In ECOOP 2008 [ECOOP 20082008].

[SOSP 1993, 1993] *Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles*, ACM SIGOPS Operating Systems Review 27(5), Asheville, NC, USA, December 1993.

[Sousa, 1993] Pedro Sousa. Garbage collection of persistent objects in a distributed object-oriented platform. In Moss et al. [Moss *et al.*1993].

[SPACE 2001, 2001] *Proceedings of the First Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management*, London, England, January 2001.

[SPACE 2004, 2004] *Proceedings of the Second Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management*, Venice, Italy, January 2004.

[SPACE 2006, 2006] *Proceedings of the Third Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management*, Charleston, SC, USA, January 2006.

[Spalding and Jia, 2006] Frances Spalding and Limin Jia. Asserting memory shape using linear logic. In SPACE 2006 [SPACE 20062006], pages 2–13.

[Sparud, 1993] Jan Sparud. Fixing some space leaks without a garbage collector. In Hughes [Hughes1993].

[Spertus, 1996a] Michael Spertus. Automating memory management. *Object Currents*, January 1996.

[Spertus, 1996b] Michael Spertus. Garbage collection in C++. *Object magazine*, 5(9), March 1996.

[Spertus, 1997] Michael Spertus. C++ and garbage collection. *Dr. Dobb's Journal*, 22(12):36–41, December 1997.

[SPIN,] The SPIN operating system. A collection of papers avilable on the WWWW.

[Spoonhower *et al.*, 2004] Daniel Spoonhower, Guy Blelloch, and Robert Harper. Incremental copying collection with pinning (progress report). In SPACE 2004 [SPACE 20042004].

[Spoonhower *et al.*, 2005] Daniel Spoonhower, Guy Blelloch, and Robert Harper. Using page residency to balance tradeoffs in tracing garbage collection. In Hind and Vitek [Hind and Vitek2005], pages 57–67.

[Spoonhower *et al.*, 2006] Daniel Spoonhower, Joshua Auerbach, David F. Bacon, Perry Cheng, and David Grove. Eventrons: A safe programming construct for high-frequency hard real-time applications. In Schwartzbach and Ball [Schwartzbach and Ball2006], pages 283–294.

[Spoonhower *et al.*, 2008] Daniel Spoonhower, Guy E. Blelloch, Robert Harper, and Phillip B. Gibbons. Space profiling for parallel functional programs. In *Proceedings of the 13th ACM SIGPLAN-International Conference on Functional Programming*, pages 253–264, Victoria, BC, Canada, September 2008.

[Srikant and Shankar, 2008] Y. N. Srikant and Priti Shankar. *The Compiler Design Handbook: Optimizations and Machine Code Generation*, chapter Garbage Collection Techniques. CRC Press, second edition, 2008.

[Srisa-an and Oey, 2005] Witiwas Srisa-an and M. Oey. Remote objects: The next garbage collection challenge. *Journal of Object Technology*, 4(4):155–172, May–June 2005.

[Srisa-an *et al.*, 1999a] Witiwas Srisa-an, C. D. Lo, and J. Morris Chang. A hardware implementation of realloc function. *Integration, the VLSI Journal*, 28:173–184, 1999.

[Srisa-an *et al.*, 1999b] Witiwas Srisa-an, C. D. Lo, and J. Morris Chang. A hardware implementation of realloc function. In *Proceedings of WVLSI'99 IEEE Annual Workshop on VLSI*, pages 697–699, Orlando, FL, April 1999.

[Srisa-an *et al.*, 2000a] Witiwas Srisa-an, J. Morris Chang, and Chia-Tien Dan Lo. Do generational schemes improve the garbage collection efficiency? In *Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software*, pages 58–63, Austin, TX, April 2000. IEEE Press.

[Srisa-an *et al.*, 2000b] Witiwas Srisa-an, Chia-Tien Dan Lo, and J. Morris Chang. Active memory: Garbage-collected memory for embedded systems. In *Second Annual Workshop on Hardware Support for Objects and Microarchitectures for Java*, pages 11–15, Austin, TX, September 2000.

[Srisa-an *et al.*, 2000c] Witiwas Srisa-an, Chia-Tien Dan Lo, and J. Morris Chang. Scalable hardware-algorithm for mark-sweep garbage collection. In *Proceedings of Euromicro Conference on Digital System Design*, pages 274–279, Maastricht, Netherlands, September 2000.

[Srisa-an *et al.*, 2002a] Witiwas Srisa-an, Chia-Tien Dan Lo, and J. Morris Chang. Object resizing and reclamation through the use of hardware bit-maps. *International Journal of Microprocessors and Microsystems*, 25:459–467, 2002.

[Srisa-an *et al.*, 2002b] Witiwas Srisa-an, Chia-Tien Dan Lo, and J. Morris Chang. Performance enhancements to the active memory system. In *Proceedings of IEEE International Conference on Computer Design, (ICCD 2002)*, pages 249–256, Freiburg, Germany, September 2002.

[Srisa-an *et al.*, 2002c] Witiwas Srisa-an, Chia-Tien Dan Lo, and J. Morris Chang. A performance perspective on the active memory system. *International Journal of Microprocessors and Microsystems*, 26(9–10):421–432, December 2002.

[Srisa-an *et al.*, 2003] Witiwas Srisa-an, Chia-Tien Dan Lo, and J. Morris Chang. Active memory processor: A hardware garbage collector for real-time Java embeded devices. *IEEE Transactions on Mobile Computing*, 2(2):89–101, April–June 2003.

[Srisa-an *et al.*, 2005] Witiwas Srisa-an, M. Oey, and S. Elbaum. Garbage collection in the presence of remote objects: A case study. In *Proceedings of the International Symposium on Distributed Objects and Applications*, pages 1065–1082, Agia Napa, Cyprus, October 2005.

[Srisa-an *et al.*, to appear] Witiwas Srisa-an, Chia-Tien Dan Lo, and J. Morris Chang. Hardware support for garbage collection in embedded systems using the active memory module. *International Journal of Microprocessors and Microsystems*, to appear.

[Stallman, 1980] Richard M. Stallman. Phantom stacks: If you look too hard, they aren't there. AI Memo 556, MIT AI Laboratory, July 1980.

[Stamos, 1982] James W. Stamos. A large object-oriented virtual memory: Grouping strategies, measurements, and performance. Technical Report SCG-82-2, Xerox PARC, Palo Alto, CA, May 1982.

[Stamos, 1984] James W. Stamos. Static grouping of small objects to enhance performance of a paged virtual memory. *ACM Transactions on Computer Systems*, 2(3):155–180, May 1984.

[Stamos, 1986] James W. Stamos. Programmer-invoked local garbage collection: A design. Technical Report unpublished draft, MIT, 1986.

[Stanchina and Meyer, 2007] Sylvain Stanchina and Mattias Meyer. Mark-sweep or copying? "a best of both worlds" algorithm and a hardware-supported real-time implementation. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 173–182.

[Standish, 1980] Thomas A. Standish. *Data Structures Techniques*. Addison-Wesley, 1980.

[Stapleton, 1990] Sue M Stapleton. Real-time garbage collection for general purpose languages. Master's thesis, Iowa State University, Department of Computer Science, August 1990.

[Steele, 1975] Guy L. Steele. Multiprocessing compactifying garbage collection. *Communications of the ACM*, 18(9):495–508, September 1975.

[Steele, 1976] Guy L. Steele. Corrigendum: Multiprocessing compactifying garbage collection. *Communications of the ACM*, 19(6):354, June 1976.

[Steele, 1977] Guy L. Steele. Data representation in PDP-10 MACLISP. AI Memo 421, MIT AI Laboratory, 1977.

[Steele, 1978] Guy L. Steele. RABBIT: A compiler for SCHEME. Technical report, MIT Press, May 1978.

[Steele, 1984] Guy L. Steele, editor. *Proceedings of the ACM Conference on Symposium on Lisp and Functional Programming*, Austin, TX, USA, August 1984.

[Steenkiste and Hennessy, 1987] Peter Steenkiste and John Hennessy. Tags and type checking in LISP: Hardware and software approaches. In ASPLOS 1987 [ASPLOS 19871987], pages 50–59.

[Steenkiste, 1987] Peter Steenkiste. *Lisp on a Reduced-Instruction-Set Processor: Characterization and Optimization*. PhD thesis, Stanford University, Also appears as Technical Report CSL-TR-87-324, Stanford University Computer System Laboratory, Palo Alto, CA, March 1987.

[Steenkiste, 1989] Peter Steenkiste. The impact of code density on instruction cache performance. In *Proceedings of Sixteenth Annual International Symposium on Computer Architecture*, pages 252–259, May 1989.

[Steensgaard, 2000] Bjarne Steensgaard. Thread-specific heaps for multi-threaded programs. In Chambers and Hosking [Chambers and Hosking2000], pages 18–24.

[Stefanović and Moss, 1994] Darko Stefanović and J. Eliot B. Moss. Characterisation of object behaviour in Standard ML of New Jersey. In LFP 1994 [LFP 19941994], pages 43–54.

[Stefanović *et al.*, 1998a] Darko Stefanović, J. Eliot B. Moss, and Kathryn S. McKinley. Oldest-first garbage collection. Technical Report 98–81, University of Massachussets, April 1998.

[Stefanović *et al.*, 1998b] Darko Stefanović, J. Eliot B. Moss, and Kathryn S. McKinley. On models for object lifetime. Technical report, University of Massachussets, February 1998.

[Stefanović *et al.*, 1999a] Darko Stefanović, Kathryn S. McKinley, and J. Eliot B. Moss. Age-based garbage collection. In OOPSLA 1999 [OOPSLA 19991999], pages 370–381.

[Stefanović *et al.*, 1999b] Darko Stefanović, J. Eliot B. Moss, and Kathryn S. McKinley. Age-based garbage collection. Technical report, University of Massachussets, April 1999. preliminary version of a paper to appear in OOPSLA'99.

[Stefanović *et al.*, 2000] Darko Stefanović, Kathryn S. McKinley, and J. Eliot B. Moss. On models for object lifetime distributions. In Chambers and Hosking [Chambers and Hosking2000], pages 137–142.

[Stefanović *et al.*, 2003] Darko Stefanović, Matthew Hertz, Stephen Blackburn, Kathryn McKinley, and J. Eliot Moss. Older-first garbage collection in practice: Evaluation in a Java virtual machine. In MSP 2002 [MSP 20022003].

[Stefanović, 1993a] Darko Stefanović. The garbage collection toolkit as an experimentation tool. In Moss et al. [Moss *et al.*1993].

[Stefanović, 1993b] Darko Stefanović. Generational copying garbage collection for Standard ML: a quantitative study. Master's thesis, University of Massachusetts, 1993.

[Stefanović, 1999] Darko Stefanović. *Properties of Age-Based Automatic Memory Reclamation Algorithms*. PhD thesis, University of Massachusetts, 1999.

[Stenning, 1976] V. Stenning. On-the-fly garbage collection. Unpublished notes, cited by [Gries, 1977a], 1976.

[Stephenson, 1983] C. J. Stephenson. New methods of dynamic storage allocation (fast fits). In *Proceedings of the Ninth ACM Symposium on Operating Systems Principles*, ACM SIGOPS Operating Systems Review 17(5), pages 30–32, Bretton Woods, NH, October 1983. ACM Press.

[Stichnoth *et al.*, 1999] James M. Stichnoth, Guei-Yuan Lueh, and Michal Cierniak. Support for garbage collection at every instruction in a Java compiler. In PLDI 1999 [PLDI 19991999], pages 118–127.

[Stone *et al.*, 1992] Harold S. Stone, John Turek, and Joel L. Wolf. Optimal partitioning of cache memory. *IEEE Transactions on Computers*, 41(9):1054–1068, September 1992.

[Stone, 1982] Harold S. Stone. Parallel memory allocation using the FETCH-AND-ADD instruction. Technical report, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, November 1982.

[Stoutamire, 1997] David Stoutamire. *Portable, Modular Expression of Locality*. PhD thesis, University of California at Berkeley, 1997.

[Stoye *et al.*, 1984] Will R. Stoye, T. J. W. Clarke, and Arthur C. Norman. Some practical methods for rapid combinator reduction. In Steele [Steele1984], pages 159–166.

[Straw *et al.*, 1989] A. Straw, F. Mellender, and S. Riegel. Object management in a persistent Smalltalk system. *Software Practice and Experience*, 19(8):719–737, 1989.

[Strobl, 2007] Torsten Strobl. *Modern Concepts Applied to C++ — Object Persistence, Reflection, Events, Garbage Collection and Thread Safety in C++*. Verlag, August 2007.

[Stroustrup, 1991] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley, second edition, December 1991.

[Stroustrup, 1996] Bjarne Stroustrup. Proposal to acknowledge that garbage collection for C++ is possible, May 1996. From the C++ core language mailing list, 27 May 1996.

[Stuckey, 2002] P.J. Stuckey, editor. *Proceedings of Eighteenth International Conference on Logic Programming*, volume 2401 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[Stumm and Zhou, 1990a] M. Stumm and Songnian Zhou. Algorithms implementing distributed shared memory. *IEEE Computing*, 23(5), May 1990.

[Stumm and Zhou, 1990b] Michael Stumm and Songnian Zhou. Fault tolerant distributed shared memory. In *Proceedings of IEEE International Conference on Parallel Distributed Computing*. IEEE Press, December 1990.

[Stygar, 1967] P. Stygar. LISP 2 garbage collector specifications. Technical Report TN-3417/500/00, System Development Corporation, April 1967.

[Subramanian, 1991] Indira Subramanian. Managing discardable pages with an external pager. In *USENIX Mach Symposium*, pages 77–85, Monterey, CA, November 1991. USENIX Association.

[Suganuma *et al.*, 2003] Toshio Suganuma, Toshiaki Yasue, and Toshio Nakatani. A region-based compilation technique for a Java just-in-time compiler. In PLDI 2003 [PLDI 20032003], pages 312–323.

[Sugimoto and others, 1983] S. Sugimoto et al. A multi-microprocessor system for Concurrent Lisp. In *International Conference on Parallel Processing*, June 1983.

[Sullivan and Chillarege, 1991] Mark Sullivan and Ram Chillarege. Software defects and their impact on system availability – a study of field failures in operating systems. In *Digest of the 21st International Symposium on Fault Tolerant Computing*, pages 2–9, June 1991.

[Sultan *et al.*, 2002a] Florin Sultan, Thu D. Nguyen, and Liviu Iftode. Lazy garbage collection of recovery state for fault-tolerant distributed shared memory. *IEEE Transactions on Parallel and Distributed Systems*, 13(10):673–686, October 2002. This paper contains some errors which led to the paper being reprinted in [Sultan *et al.*, 2002b].

[Sultan *et al.*, 2002b] Florin Sultan, Thu D. Nguyen, and Liviu Iftode. Lazy garbage collection of recovery state for fault-tolerant distributed shared memory. *IEEE Transactions on Parallel and Distributed Systems*, 13(10):1085–1098, October 2002.

[Sun and Gehringer, 1997] Jingyu Sun and Edward F. Gehringer. A Smalltalk memory profiler and its performance enhancement. In Dickman and Wilson [Dickman and Wilson1997].

[Sun JNI, 1997, 1997] *Java Native Interface*, 1997. Javasoft's Native Interface for Java.

[Sun Microsystems, 2001] Sun Microsystems. The Java HotSpot Virtual Machine, 2001. Technical White Paper.

[Sun Microsystems, 2006] Sun Microsystems. Memory management in he Java HotSpot Virtual Machine, April 2006. Technical White Paper.

[Sun Microsystems, 2009] Sun Microsystems. Java SE 6 HotSpot virtual machine garbage collection tuning, 2009.

[Sun, 2002a] Sun Microsystems. *Java 2 Platform, Enterprise Edition (J2EE), 1.4 Specification*, 2002.

[Sun, 2002b] Sun Microsystems. *Java 2 Platform, Standard Edition, v 1.4.0: API Specification*, 2002.

[Sundell, 2005] H. Sundell. Wait-free reference counting and memory management. In *Proceedings of the 19th International Parallel and Distributed Processing Symposium (IPDPS 2005)*, Denver, CO, April 2005.

[Suzuki and Terashima, 1995] Mitsugu Suzuki and Motoaki Terashima. Time- and space-efficient garbage collection based on sliding compaction. *Transaction of Information Processing (IPSJ)*, 36(4):925–931, 1995.

[Suzuki *et al.*, 1995] Mitsugu Suzuki, Hiroshi Koide, and Motoaki Terashima. MOA — a fast sliding compaction scheme for a large storage space. In Baker [Baker1995a].

[Swamy *et al.*, 2006] Nikhil Swamy, Michael Hicks, Greg Morrisett, Dan Grossman, and Trevor Jim. Safe manual memory management in Cyclone. In *Science of Computer Programming* [Jones2006], pages 122–144.

[Swanson, 1986] M. Swanson. An improved portable copying garbage collector. OPnote 86–03, University of Utah, February 1986.

[Swinehart *et al.*, 1986] Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, and Robert B. Hagmann. A structural view of the Cedar programming environment. Technical Report CSL–86–1, Xerox Corporation, 1986.

[Szöke, 1977] Péter Szöke. Some remarks on new instances and garbage collection. *ACM SIGPLAN Notices*, 12(6), June 1977. Proceedings of Strathclyde Algol-68 Conference.

[Szymanek and Kuchcinski, 1999] R. Szymanek and K. Kuchcinski. Design space exploration in system level synthesis under memory constraints. In *Euromicro 35*, Milan, September 1999.

[Szymanek and Kuchcinski, 2000] R. Szymanek and K. Kuchcinski. Task assignment and scheduling under memory constraints. In *Euromicro 2000*, 2000.

[Szymanek and Kuchcinski, 2001] R. Szymanek and K. Kuchcinski. A constructive algorithm for memory-aware task assignment and scheduling. In *Ninth International Symposium on Hardware/Software Codesign*, Copenhagen, April 2001.

[Szymanek, 2001] R. Szymanek. Memory aware task assignment and scheduling for multiprocessor embedded systems. Master's thesis, Department of Computer Science, Lund University, June 2001.

[Tadman, 1978] M. Tadman. Fast-fit: A new hierarchical dynamic storage allocation technique. Master's thesis, UC Irvine, Computer Science Department, 1978.

[Takeda, 1990] Tomohiro Takeda. A garbage collecting method for object-oriented concurrent languages. In Jul and Juul [Jul and Juul1990].

[Tanenbaum, 1988] Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall, second edition, 1988.

[Tang *et al.*, 2008] Yan Tang, Qi Gao, and Feng Qin. LeakSurvivor: Towards safely tolerating memory leaks for garbage-collected languages. In *Proceedings of USENIX'08*, pages 307–320, 2008.

[Tarau, 1992] P. Tarau. Ecological memory management in a continuation passing Prolog engine. In Bekkers and Cohen [Bekkers and Cohen1992].

[Tarditi and Diwan, 1993] David Tarditi and Amer Diwan. The full cost of a generational copying garbage collection implementation. In Moss et al. [Moss *et al.*1993].

[Tarditi and Diwan, 1994] David Tarditi and Amer Diwan. Measuring the cost of storage management. Technical Report CMU-CS-94-201, Carnegie Mellon University, 1994. Accepted for publication in Lisp and Symbolic Computation.

[Tarditi and Diwan, 1996] David Tarditi and Amer Diwan. Measuring the cost of storage management. *Lisp and Symbolic Computation*, 9(4), 1996.

[Tarditi, 2000] David Tarditi. Compact garbage collection tables. In Chambers and Hosking [Chambers and Hosking2000], pages 50–58.

[Tarjan, 1992] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal of Computing*, 1(2), 1992.

[Tärnlund, 1984] S.-A. Tärnlund, editor. *Proceedings of Second International Conference on Logic Programming*, Uppsala, Finland, 1984.

[Taura and Yonezawa, 1997a] Kenjiro Taura and Akinori Yonezawa. An effective garbage collection strategy for parallel programming languages on large scale distributed-memory machines. In PPOPP 1997 [PPOPP 19971997], pages 264–275.

[Taura and Yonezawa, 1997b] Kenjiro Taura and Akinori Yonezawa. *Efficient and Reusable Implementation of Fine-Grain Multithreading and Garbage Collection on Distributed-Memory Parallel Computers*. PhD thesis, University of Tokyo, 1997.

[Taylor *et al.*, 1986] George S. Taylor, Paul N. Hilfinger, James R. Larus, David A. Patterson, and Benjamin G. Zorn. Evaluation of the SPUR Lisp architecture. In *Proceedings of the Thirteenth Symposium on Computer Architecture*, June 1986.

[Taylor, 1989] S. Taylor. *Parallel Logic Programming Techniques*. Prentice-Hall, 1989.

[Tel and Mattern, 1991] Gerard Tel and Friedmann Mattern. The derivation of distributed termination detection algorithms from garbage collection schemes — (extended abstract). In Aarts et al. [Aarts and others1991].

[Tel and Mattern, 1993] Gerard Tel and Friedmann Mattern. The derivation of distributed termination detection algorithms from garbage collection schemes. *ACM Transactions on Programming Languages and Systems*, 15(1), January 1993.

[Tel *et al.*, 1987] Gerard Tel, Richard B. Tan, and Jan van Leeuwen. The derivation of on-the-fly garbage collection algorithms from distributed termination detection protocols. *Lecture Notes in Computer Science*, 247:445–455, 1987.

[Tel *et al.*, 1988] Gerard Tel, Richard B. Tan, and Jan van Leeuwen. The derivation of graph marking algorithms from distributed termination detection protocols. *Science Of Computer Programming*, 10(2):107–137, 1988.

[Tel, 1991] Gerard Tel. *Topics in Distributed Algorithms*, volume 1 of *Cambridge international series on parallel computation*. Cambridge University Press, New York, 1991.

[Tel, 1994] Gerard Tel. *Introduction to Distributed Algorithms*. Cambridge University Press, 1994.

[Templ, 1991] J. Templ. Oberon technical notes: Garbage collection on open arrays. ETH Technical Report 156, ETH Eidgenössische Technische Hochschule Zürich, March 1991.

[Terashima and Goto, 1978] Motoaki Terashima and Eiichi Goto. Genetic order and compactifying garbage collectors. *Information Processing Letters*, 7(1):27–32, January 1978.

[Terauchi and Aiken, 2004] T. Terauchi and Alex Aiken. Memory management with use-counted regions. Technical report, University of California, Berkeley, March 2004.

[Terry and Swinehart, 1988] Douglas B. Terry and Daniel C. Swinehart. Managing stored voice in the etherphone system. *ACM Transactions on Computer Systems*, 6(1):3–27, February 1988.

[Thacker and Stewart, 1987] Charles P. Thacker and Lawrence C. Stewart. Firefly: A multiprocessor workstation. In ASPLOS 1987 [ASPLOS 19871987], pages 164–172. Also DEC SRC Research report 23, December 1987.

[Thazhuthaveetil and Pleszkun, 1987] M. J. Thazhuthaveetil and A. R. Pleszkun. On the structural locality of reference in Lisp list access streams. *Information Processing Letters*, 26(2):105–110, 1987.

[Thesen and Pinkerton, 1977] Arne Thesen and Tad Pinkerton. Predicting the availability of contiguous memory. *International Journal of Computer and Information Sciences*, 6(4), December 1977.

[Thiebaut *et al.*, 1992] Dominique Thiebaut, Joel L. Wolf, and Harold S. Stone. Synthetic traces for trace-driven simulation of cache memories. *IEEE Transactions on Computers*, 41(4):388–410, April 1992.

[Thiebaut, 1989] Dominique Thiebaut. The fractal dimension of computer programs and its application to the prediction of the cache miss ratio. *IEEE Transactions on Computers*, pages 1012–1026, July 1989.

[Thomas and Jones, 1994] Stephen P. Thomas and Richard E. Jones. Garbage collection for shared environment closure reducers. Technical Report 31–94, University of Kent and University of Nottingham, December 1994.

[Thomas *et al.*, 1998] S. Thomas, W. Charnell, S. Darnell, B. A. A. Dias, J. G. P. Kramskoy, J. Sextonand, J. Wynn, K. Rautenbach, and W. Plummer. Low-contention grey object sets for concurrent, marking garbage collection. United States Patent Application, 20020042807, 1998.

[Thomas, 1981] R. E. Thomas. A dataflow computer with improved asymptotic performance. Technical Report MIT/LCS/TR–265, MIT Laboratory for Computer Science, 1981.

[Thomas, 1993] Stephen P. Thomas. *The Pragmatics of Closure Reduction*. PhD thesis, The Computing Laboratory, University of Kent at Canterbury, October 1993.

[Thomas, 1995a] Stephen P. Thomas. Garbage collection in shared-environment closure reducers: Space-efficient depth first copying using a tailored approach. *Information Processing Letters*, 56(1):1–7, October 1995.

[Thomas, 1995b] Stephen P. Thomas. Having your cake and eating it: Recursive depth-first copying garbage collection with no extra stack. Personal communication, May 1995.

[Thomborson, 1996] Clark D. Thomborson. When virtual memory isn't enough. Technical Report 136, University of Auckland, November 1996.

[Thompson and Lins, 1988] Simon J. Thompson and Rafael D. Lins. Cyclic reference counting: A correction to Brownbridge's algorithm. Unpublished notes, 1988.

[Thompson, 1987] James G. Thompson. *Efficient Analysis of Caching Systems*. PhD thesis, University of California, Berkeley, October 1987. Also technical report UCB/CSD 87/374.

[Thorelli, 1972] Lars-Erik Thorelli. Marking algorithms. *BIT*, 12(4):555–568, 1972.

[Thorelli, 1976] Lars-Erik Thorelli. A fast compactifying garbage collector. *BIT*, 16(4):426–441, 1976.

[Tick *et al.*, 1992] E. Tick, S. Duvvuru, and R. Sundararajan. A compile time memory-reuse scheme for concurrent logic programs. In Bekkers and Cohen [Bekkers and Cohen1992].

[Tick, 1988] E. Tick. *Memory performance of Prolog Architectures*. Kluwer, 1988.

[Tikir and Hollingsworth, 2005] Mustafa M. Tikir and Jeffery K. Hollingsworth. NUMA-aware Java heaps for server applications. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, page 108.2, Denver, CO, 2005. IEEE Computer Society.

[Ting, 1975] D. W. Ting. Some results of the space requirements of dynamic memory allocation algorithms. Technical Report 75-229, Cornell University Department of Computer Science, February 1975.

[T.J.Bailey, 1975] Norman T.J.Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Griffin, 2nd edition, 1975.

[Tofte and Birkedal, 1998] Mads Tofte and Lars Birkedal. A region inference algorithm. *ACM Transactions on Programming Languages and Systems*, 20(4):734–767, July 1998.

[Tofte and Hallenberg, 2001] Mads Tofte and Niels Hallenberg. Region-based memory management in perspective. In SPACE 2001 [SPACE 20012001]. Invited talk.

[Tofte and Talpin, 1993] Mads Tofte and Jean-Pierre Talpin. A theory of stack allocation in polymorphically typed languages. Technical Report Computer Science 93/15, University of Copenhagen, July 1993.

[Tofte and Talpin, 1994] Mads Tofte and Jean-Pierre Talpin. Implementation of the typed call-by-value λ-calculus using a stack of regions. In POPL 1994 [POPL 19941994], pages 188–201.

[Tofte and Talpin, 1997] Mads Tofte and Jean-Pierre Talpin. Region-based memory management. *Information and Computation*, 132(2):109–176, February 1997. An earlier version of this was presented at [POPL 1994, 1994].

[Tofte *et al.*, 1997] Mads Tofte, Lars Birkedal, Martin Elsman, Niels Hallenberg, Tommy Højfeld Olesen, Peter Sestoft, and Peter Bertelsen. Programming with Regions in the ML Kit. Technical Report DIKU-TR-97/12, Department of Computer Science (DIKU), University of Copenhagen, April 1997.

[Tofte *et al.*, 2001] Mads Tofte, Lars Birkedal, Martin Elsman, Niels Hallenberg, Tommy Højfeld Olesen, and Peter Sestoft. Programming with Regions in the ML Kit, version 4. Technical report, IT University of Copenhagen, October 2001.

[Tofte *et al.*, 2004] Mads Tofte, Lars Birkedal, Martin Elsman, and Niels Hallenberg. A retrospective on region-based memory management. *Higher-Order and Symbolic Computation*, 17(3), September 2004.

[Tofte, 1998] Mads Tofte. A brief introduction to Regions. In Peyton Jones and Jones [Peyton Jones and Jones1998], pages 186–195.

[Tokoro and Pareschi, 1994] M. Tokoro and R. Pareschi, editors. *Proceedings of the Eighth European Conference on Object-Oriented Programming, ECOOP94*, volume 821 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[Tolmach, 1994] Andrew Tolmach. Tag-free garbage collection using explicit type parameters. In PLDI 1994 [PLDI 19941994], pages 1–11.

[Tong and Lau, 2010] Liangliang Tong and Francis C.M. Lau. Exploiting memory usage patterns to improve garbage collections in Java. In PPPJ 2010 [PPPJ 20102010].

[Tong, 1997] Guanshan Tong. *Leveled Garbage Collection For Automatic Memory Management*. PhD thesis, University of Chicago, November 1997.

[Topor, 1979] R. Topor. The correctness of the Schorr–Waite list marking algorithm. *Acta Informatica*, 11(3), 1979.

[Torp-Smith *et al.*, 2008] Noah Torp-Smith, Lars Birkedal, and John C. Reynolds. Local reasoning about a copying garbage collector. *ACM Transactions on Programming Languages and Systems*, 30(4), July 2008.

[Touati and Hama, 1988] Hervé Touati and Toshiyuki Hama. A light-weight Prolog garbage collector. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 922–930, Tokyo, 1988.

[Touati, 1988] Hervé Touati. A prolog garbage collector for Aquarius. Technical Report UCB//CSD-88-443, University of California, Berkeley, August 1988.

[Touraïvane, 1988] Touraïvane. *La Récupération de Mémoire dans les Machines Non-Déterministes*. PhD thesis, Université d'Aix-Marseille, 1988.

[Trancon y Widemann, 2008] Baltasar Trancon y Widemann. A reference counting garbage collection algorithm for cyclical functional programming. In Jones and Blackburn [Jones and Blackburn2008], pages 71–80.

[Tullsen and Eggers, 1993] Dean M. Tullsen and Susan J. Eggers. Limitations of cache prefetching on a bus-based multiprocessor. In ISCA 1993 [ISCA 19931993], pages 278–288.

[Turbak *et al.*, 2008] Franklyn Turbak, David Gifford, and Mark A. Sheldon. *Garbage Collection*, chapter 18. MIT Press, 2008.

[Turner, 1979] David A. Turner. A new implementation technique for applicative languages. *Software Practice and Experience*, 9, 1979.

[Turner, 1981] David A. Turner. Recursion equations as a programming language. In John Darlington, Peter Henderson, and David Turner, editors, *Functional Programming and its Applications*, pages 1–28. Cambridge University Press, January 1981.

[Turner, 1985] David A. Turner. Miranda — a non-strict functional language with polymorphic types. In Jouannaud [Jouannaud1985], pages 1–16.

[Ueda and Morita, 1990] K. Ueda and M. Morita. A new implementation for flat GHC. In Warren and Szeredi [Warren and Szeredi1990], pages 3–17.

[Ugawa *et al.*, 2010] Tomoharu Ugawa, Hideya Iwasaki, and Taiichi Yuasa. Improved replication-based incremental garbage collection for embedded systems. In jan Vitek and Lea [jan Vitek and Lea2010], pages 73–82.

[Ungar and Jackson, 1988] David M. Ungar and Frank Jackson. Tenuring policies for generation-based storage reclamation. *ACM SIGPLAN Notices*, 23(11):1–17, 1988.

[Ungar and Jackson, 1991] David M. Ungar and Frank Jackson. Outwitting GC devils: A hybrid incremental garbage collector. In Wilson and Hayes [Wilson and Hayes1991a].

[Ungar and Jackson, 1992] David M. Ungar and Frank Jackson. An adaptive tenuring policy for generation scavengers. *ACM Transactions on Programming Languages and Systems*, 14(1):1–27, 1992.

[Ungar and Patterson, 1983] David M. Ungar and David A. Patterson. Berkeley Smalltalk: Who knows where the time goes? In Krasner [Krasner1983], pages 189–206.

[Ungar, 1984] David M. Ungar. Generation scavenging: A non-disruptive high performance storage reclamation algorithm. *ACM SIGPLAN Notices*, 19(5):157–167, April 1984. Also published as ACM Software Engineering Notes 9, 3 (May 1984) — Proceedings of the ACM/SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments, 157–167, April 1984.

[Ungar, 1986] David M. Ungar. *The Design and Evaluation of a High Performance Smalltalk System*. ACM distinguished dissertation 1986. MIT Press, 1986.

[Ungureanu and Goldberg, 1997] Christian Ungureanu and Benjamin Goldberg. Formal models of distributed memory management. In ICFP 1997 [ICFP 19971997], pages 280–291.

[Unnikrishnan and Stoller, 2009] Leena Unnikrishnan and Scott D. Stoller. Parametric heap usage analysis for functional programs. In Kolodner and Steele [Kolodner and Steele2009], pages 139–148.

[Unnikrishnan *et al.*, 2000] Leena Unnikrishnan, Scott D. Stoller, and Yanhong A. Liu. Automatic accurate stack space and heap space analysis for high-level languages. Technical Report 538, Indiana University, April 2000.

[Unnikrishnan *et al.*, 2001a] Leena Unnikrishnan, Scott D. Stoller, and Yanhong A. Liu. Automatic accurate live memory analysis for garbage-collected languages. In LCTES 2001 [LCTES 20012001], pages 102–111.

[Unnikrishnan *et al.*, 2001b] Leena Unnikrishnan, Scott D. Stoller, and Yanhong A. Liu. Optimized live heap bound analysis. Technical Report DAR 01-2, SUNY at Stony Brook, October 2001.

[Valois, 1995a] J.D. Valois. *Lock-free data structures*. PhD thesis, Rensselaer Polytechnic Institute, 1995.

[Valois, 1995b] J.D. Valois. Lock-free linked lists using compare-and-swap. In *Proceedings of the 14th ACM Symposium on Principles of Distributed Computing (PODC'95)*, pages 214–222, August 1995.

[van Assche *et al.*, 2006] M. van Assche, J. Goossens, and E. Devillers. Joint garbage collection and hard real-time scheduling. *Journal of Embedded Computing*, 2(3–4), 2006. Also published in RTS'05 International Conference on Real-Time Systems, 2005.

[van de Snepscheut, 1987] Jan van de Snepscheut. Algorithms for on-the-fly garbage collection revisited. *Information Processing Letters*, 24(4):211–216, March 1987.

[van Groningen, 2004] J. van Groningen. Faster garbage collection using prefetching. In C.Grelck and F. Huch, editors, *Proceedings of Sixteenth International Workshop on Implementation and Application of Functional Languages (IFL'04)*, pages 142–152, Lübeck, Germany, 2004.

[Vanderwaart and Crary, 2003] Joseph C. Vanderwaart and Karl Crary. A typed interface for garbage collection. In Shao and Lee [Shao and Lee2003], pages 109–122.

[Vardhan and Agha, 2002] Abhay Vardhan and Gul Agha. Using passive object garbage collection algorithms. In Boehm and Detlefs [Boehm and Detlefs2002], pages 106–113.

[Varhol, 1997] Peter Varhol. Adapting Java for embedded systems. *Computer Design*, page 75, October 1997.

[Vataja and Ukkonen, 1984] P. Vataja and E. Ukkonen. Finding temporary terms in PROLOG programs. In *Fifth Generation Computer Systems 1984, Proceedings of the International Conference. Tokyo, 1984 Nov 6–9*, pages 275–282, Tokyo, 1984. Ohmsha Ltd.

[Vaughan and Dearle, 1992] Francis Vaughan and Alan Dearle. Supporting large persistent stores using conventional hardware. In Albano and Morrison [Albano and Morrison1992].

[Vaughan *et al.*, 1990] Francis Vaughan, T. Schunke, B. Koch, A. Dearle, C. Marlin, and C. Barter. A persistent distributed architecture supported by the mach operating system. In *First USENIX Conference on the Mach Operating System*, pages 123–140. USENIX Association, 1990.

[Vaughan *et al.*, 2000] Francis A. Vaughan, William F. Brodie-Tyrrell, Katrina E. Falkner, and David S. Munro. Bounded parallel garbage collection: Implementation and adaptation. In *Proceedings of 7th Australian Parallel and Real Time PART'2000*, Sydney, 2000.

[Vechev and Bacon, 2004] Martin Vechev and David F. Bacon. Write barrier elision for concurrent garbage collectors. In Bacon and Diwan [Bacon and Diwan2004], pages 13–24.

[Vechev and Petrov, 2003] Martin T. Vechev and Peter D. Petrov. Class unloading with a concurrent garbage collector in an embedded Java VM. In *Embedded Systems and Applications 2003 (ESA'03)*, 2003.

[Vechev *et al.*, 2005] Martin Vechev, David F. Bacon, Perry Cheng, and David Grove. Derivation and evaluation of concurrent collectors. In Black [Black2005].

[Vechev *et al.*, 2006] Martin T. Vechev, Eran Yahav, and David F. Bacon. Correctness-preserving derivation of concurrent garbage collection algorithms. In Schwartzbach and Ball [Schwartzbach and Ball2006], pages 341–353.

[Vechev *et al.*, 2007] Martin T. Vechev, Eran Yahav, David F. Bacon, and Noam Rinetzky. CGCExplorer: A semi-automated search procedure for provably correct concurrent collectors. In Ferrante and McKinley [Ferrante and McKinley2007], pages 456–467.

[Vechev *et al.*, 2010] Martin Vechev, Eran Yahav, and Greta Yorsh. Phalanx: parallel checking of expressive heap assertions. In jan Vitek and Lea [jan Vitek and Lea2010], pages 41–50.

[Vechev, 2007] Martin Vechev. *Derivation and Evaluation of Concurrent Collectors*. PhD thesis, University of Cambridge, 2007.

[Vegdahl and Pleban, 1989] Steven R. Vegdahl and Uwe F. Pleban. The runtime environment of Screme, a Scheme implementation for the 88000. In ASPLOS 1989 [ASPLOS 19891989], pages 172–182.

[Veiga and Ferreira, 2003] Luís Veiga and Paulo Ferreira. Complete distributed garbage collection, an experience with Rotor. *IEE Research Journals – Software*, 150(5), October 2003.

[Veiga and Ferreira, 2004] Luís Veiga and Paulo Ferreira. Asynchronous, complete distributed garbage collection. Technical Report RT/11/2004, INESC–ID, Lisboa, June 2004.

[Veiga and Ferreira, 2005a] Luís Veiga and Paulo Ferreira. Asynchronous complete distributed garbage collection. In *Nineteenth IEEE International Symposium on Parallel and Distributed Processing*, Denver, CO, April 2005.

[Veiga and Ferreira, 2005b] Luís Veiga and Paulo Ferreira. A comprehensive approach for memory management of replicated objects. Technical Report RT/07/2005, INESC-ID, Lisbon, April 2005.

[Veiga *et al.*, 2007] Luís Veiga, P. Pereira, and Paulo Ferreira. Complete distributed garbage collection using DGC-consistent cuts and .NET AOP-support. *IET Software*, 1(6):263–279, 2007.

[Veiga, 2007] Luís Veiga. *OBIWAN: Middleware for Memory Management of Replicated Objects in Distributed and Mobile Computing*. PhD thesis, Univesidade Técnica de Lisboa, Instituto Superior Técnico, March 2007.

[Veillon, 1976] G. Veillon. Transformations de programmes recursifs. *R.A.I.R.O. Informatique*, 10(9):7–20, September 1976.

[Velasco *et al.*, 2004a] Jos Manuel Velasco, , David Atienza, L. Pinuel, and Francky Catthoor. Energy-aware modelling of garbage collectors for new dynamic embedded systems. In *Proceedings of First International Workshop on Power-Aware Real-Time Computing*, Pisa, Italy, 2004.

[Velasco *et al.*, 2004b] Jos Manuel Velasco, David Atienza, Francky Catthoor, , Francisco Tirado, Katzalin Olcoz, and Jose Manuel Mendias. Garbage collector refinement for new dynamic multimedia applications on embedded systems. In *8th Annual Workshop on Interaction between Compilers and Computer Architecture (INTERACT-8 2004)*, pages 25–32, Madrid, Spain, February 2004.

[Velasco *et al.*, 2004c] Jos Manuel Velasco, Antonio Ortiz, Katzalin Olcoz, and Francisco Tirado. Adaptive tuning of reserved space in an Appel collector. In ECOOP 2004 [ECOOP 20042004], pages 543–559.

[Velasco *et al.*, 2004d] Jos Manuel Velasco, Antonio Ortiz, Katzalin Olcoz, and Francisco Tirado. Dynamic management of nursery space organization in generational collection. In *8th Annual Workshop on Interaction between Compilers and Computer Architecture (INTERACT-8 2004)*, pages 33–40, Madrid, Spain, February 2004.

[Velasco *et al.*, 2005a] Jos Manuel Velasco, , David Atienza, Katzalin Olcoz, and Francky Catthoor. Performance evaluation of barrier techniques for distributed tracing garbage collectors. In *Proceedings of the International Conference on Parallel Computing PARCO 2005*, pages 549–556, Malaga, Spain, 2005.

[Velasco *et al.*, 2005b] Jos Manuel Velasco, , David Atienza, Katzalin Olcoz, Francky Catthoor, Francisco Tirado, and Jose Manuel Mendias. Energy characterization of garbage collectors for dynamic applications on embedded systems. In *International Workshop on Integrated Circuit and System Design, Power and Timing Modeling, Optimization and Simulation PATMOS 2005*, pages 69–78, Leuven, Belgium, 2005.

[Veldema and Philippsen, 2007] Ronald Veldema and Michael Philippsen. Supporting huge address spaces in a virtual machine for Java on a cluster. In *The 20th International Workshop on Languages and Compilers for Parallel Computing*, Urbana, IL, October 2007.

[Vengerov, 2009] David Vengerov. Modeling, analysis and throughput optimization of a generational garbage collector. In Kolodner and Steele [Kolodner and Steele2009], pages 1–9.

[Venkatasubramanian *et al.*, 1992a] Nalini Venkatasubramanian, Gul Agha, and Carolyn Talcott. Hierarchical garbage collection in scalable distributed systems. Technical Report UIUCDCS-R-92-1740, Department of Computer Science, University of Illinois at Urbana-Champaign, April 1992.

[Venkatasubramanian *et al.*, 1992b] Nalini Venkatasubramanian, Gul Agha, and Carolyn Talcott. Scalable distributed garbage collection for systems of active objects. In Bekkers and Cohen [Bekkers and Cohen1992], pages 134–147.

[Venkatasubramanian, 1991] Nalini Venkatasubramanian. Hierarchical garbage collection in scalable distributed systems. Master's thesis, University of Illinois, Urbana-Champaign, 1991.

[Venners, 1998a] Bill Venners. *Inside the Java Virtual Machine*. The Java Masters Series. Computing McGraw-Hill, February 1998. Chapter 9: Garbage Collection.

[Venners, 1998b] Bill Venners. Object finalization and cleanup. http://www.javaworld.com/javaworld/jw-06-1998/jw-06-techniques.html, June 1998.

[Venstermans *et al.*, 2005] Kris Venstermans, Lieven Eeckhout, and Koen De Bosschere. Implicit typing for 64-bit object header reduction in Java. In *Proceedings of the fifth ACES Symposium*, pages 78–81, Gent, September 2005. Academia Press.

[Venstermans *et al.*, 2007a] Kris Venstermans, Lieven Eeckhout, and Koen De Bosschere. Java object header elimination for reduced memory consumption in 64-bit virtual machines. *ACM Transactions on Architecture and Code Optimization*, 4(3):30, 9 2007.

[Venstermans *et al.*, 2007b] Kris Venstermans, Lieven Eeckhout, and Koen De Bosschere. Object-relative addressing: Compressed pointers in 64-bit Java virtual machines. In *Proceedings of the 21st European Conference on Object-Oriented Programming (ECOOP '07)*, volume 4609 of *Lecture Notes in Computer Science*, pages 79–100, Berlin Heidelberg, July 2007. Springer-Verlag.

[Vestal, 1987] Stephen C. Vestal. *Garbage Collection: An Exercise in Distributed, Fault-Tolerant Programming*. PhD thesis, University of Washington, Seattle, WA, 1987.

[Vijaykrishnan *et al.*, 2001] N. Vijaykrishnan, M. Kandemir, S. Kim, S. Tomar, A. Sivasubramaniam, and M. J. Irwin. Energy behavior of Java applications from the memory perspective. In JVM 2001 [JVM 20012001].

[Virding, 1995] Robert Virding. A garbage collector for the concurrent real-time language Erlang. In Baker [Baker1995a].

[Vo, 1996] Kiem-Phong Vo. Vmalloc: A general and efficient memory allocator. *Software Practice and Experience*, 26(3):357–374, 1996.

[Voldman *et al.*, 1983] J. Voldman, B. Mandelbrot, L. W. Hoevel, J. Knight, and P. Rosenfeld. Fractal nature of software-cache interaction. *IBM Journal of Research and Development*, 27(2):164–170, March 1983.

[Vuillemin, 1980] Jean Vuillemin. A unifying look at data structures. *Communications of the ACM*, 29(4):229–239, April 1980.

[Wadler, 1976] Philip L. Wadler. Analysis of an algorithm for real-time garbage collection. *Communications of the ACM*, 19(9):491–500, September 1976.

[Wadler, 1984] Philip L. Wadler. Listlessness is better than laziness: Lazy evaluation and garbage collection at compile time. In Steele [Steele1984], pages 45–52.

[Wadler, 1987] Philip L. Wadler. Fixing some space leaks with a garbage collector. *Software Practice and Experience*, 17(9):595–609, September 1987.

[Waite, 1973] W. M. Waite. *Implementing Software for Nonnumeric Applications*. Prentice-Hall, 1973.

[Wakeling and Runciman, 1991] David Wakeling and Colin Runciman. Linearity and laziness. In Hughes [Hughes1991a], pages 215–240.

[Wakeling, 1990] David Wakeling. *Linearity and Laziness*. PhD thesis, University of York, November 1990.

[Walden, 1972] O. C. Walden. A note on Cheney's non-recursive list-compacting algorithm. *Communications of the ACM*, 15(4), April 1972.

[Walker *et al.*, 2000] David Walker, Karl Crary, and Greg Morrisett. Typed memory management in a calculus of capabilities. *ACM Transactions on Programming Languages and Systems*, 24(4):701–771, 2000.

[Walker, 2001] David Walker. On linear types and regions. In SPACE 2001 [SPACE 20012001].

[Walker, 2004] David Walker. Stacks, heaps and regions: One logic to bind them. In SPACE 2004 [SPACE 20042004]. Invited talk.

[Wall and Schwartz, 1991] Larry Wall and Randal L. Schwartz. *Programming Perl*. O'Reilly and Associates, Inc., 1991.

[Wallace and Runciman, 1993] Malcolm Wallace and Colin Runciman. An incremental garbage collector for embedded real-time systems. In *Proceedings of the Chalmers Winter Meeting*, pages 273–288, Tanum Strand, Sweden, 1993. Published as Programming Methodology Group, Chalmers University of Technology, Technical Report 73.

[Wang and Appel, 1999] Daniel C. Wang and Andrew W. Appel. Safe garbage collection = regions + intensional type analysis. Technical report, Princeton, July 1999.

[Wang and Appel, 2000] Daniel C. Wang and Andrew W. Appel. Type-preserving garbage collectors (extended version). Technical report, Department of Computer Science, Princeton University, December 2000.

[Wang and Appel, 2001] Daniel C. Wang and Andrew W. Appel. Type-preserving garbage collectors. In POPL 2001 [POPL 20012001], pages 166–178.

[Wang and Varela, 2006] W. Wang and C.A. Varela. Distributed garbage collection for mobile actor systems: the pseudo root approach. In *Proceedings of the First International Conference on Grid and Pervasive Computing (GPC 2006)*, Taichung, Taiwan, May 2006. Springer-Verlag.

[Wang *et al.*, 1995] Y. Wang, P. Chung, I. Lin, and W.K.Fuchs. Checkpoint space reclamation for uncoordinated checkpointing in message-passing systems. *IEEE Transactions on Parallel and Distributed Systems*, 6(5):546–554, May 1995.

[Wang *et al.*, 2008] Xi Wang, Zhilei Xu, Xuezheng Liu, Zhenyu Guo, Xiaoge Wang, and Zheng Zhang. Conditional correlation analysis for safe region-based memory management. In Gupta and Amarasinghe [Gupta and Amarasinghe2008], pages 45–55.

[Wang, 1989] Thomas Wang. The MM garbage collector for C++. Master's thesis, California State Polytechnic University, October 1989.

[Wang, 1994a] Thomas Wang. Better C: An object-oriented C language with automatic memory manager suitable for interactive applications. *ACM SIGPLAN Notices*, 29(11):104–111, December 1994.

[Wang, 1994b] Thomas Wang. Eliminate memory fragmentation through holes in the heap. *ACM SIGPLAN Notices*, 29(11):112–113, December 1994.

[Warren and Szeredi, 1990] D. H. D. Warren and P. Szeredi, editors. *Proceedings of Seventh International Conference on Logic Programming*, Jerusalem, 1990.

[Warren, 1977] David H. D. Warren. Implementing Prolog — compiling logic programs. D.A.I. Research Report 39, 40, University of Edinburgh, 1977.

[Warren, 1980] David H. D. Warren. An improved Prolog implementation which optimises tail-recursion. In *Workshop on Logic Programming, Debrecen, Hungary*, 1980.

[Warren, 1982] David H. D. Warren. Perpetual processes — an unexploited Prolog technique. *Logic Programming Newsletter*, 3, 1982.

[Warren, 1983] David H. D. Warren. An abstract Prolog instruction set. Technical Note 309, SRI International, 1983.

[Warren, 1984] David A. Warren. Efficient Prolog memory management for flexible control strategies. In *International Symposium on Logic Programming*. IEEE Press, 1984.

[Washabaugh and Kafura, 1990] Douglas Markham Washabaugh and Dennis Kafura. Real-time garbage collection of actors. In *Proceedings of the 11th Real-Time Systems Symposium*, pages 21–30, December 1990.

[Washabaugh and Kafura, 1991] Douglas M. Washabaugh and D. Kafura. Distributed garbage collection of active objects. In *icdcs11*, pages 369–276, May 1991.

[Washabaugh, 1989] Douglas Markham Washabaugh. Real-time garbage collection of actors in a distributed system. Master's thesis, Virginia Polytechnic Institute and State University, 1989.

[Watson and Watson, 1987a] Paul Watson and Ian Watson. An efficient garbage collection scheme for parallel computer architectures. In de Bakker et al. [de Bakker *et al.*1987], pages 432–443.

[Watson and Watson, 1987b] Paul Watson and Ian Watson. Graph reduction in a parallel virtual memory environment. In J. H. Fasel and R. M. Keller, editors, *Graph Reduction: Proceedings of a Workshop at Santa Fe, New Mexico*, volume 279 of *Lecture Notes in Computer Science*, pages 265–274. Springer-Verlag, 1987.

[Watson and Wise, 1976] Watson and David Wise. Tuning Garwick's algorithm for repacking sequential storage. *BIT*, 16(4):442–450, December 1976.

[Watson, 1986] Ian Watson. An analysis of garbage collection for distributed systems. Technical report, Department of Computer Science, University of Manchester, 1986.

[Weemeeuw and Demoen, 1990] P. Weemeeuw and B. Demoen. A la recherche de la mémoire perdue, or: Memory compaction for shared memory multiprocessors — design and specification. In S. Debray and M. Hermenegildo, editors, *2nd North American Conference on Logic Programming*, pages 306–320, 1990.

[Weemeeuw, 1992] P. Weemeeuw. Garbage collection in Aurora: An overview. In Bekkers and Cohen [Bekkers and Cohen1992].

[Wegbreit, 1972a] B. Wegbreit. A generalised compactifying garbage collector. *Computer Journal*, 15(3):204–208, August 1972.

[Wegbreit, 1972b] B. Wegbreit. A space efficient list structure tracing algorithm. *IEEE Transactions on Computers*, pages 1098–1010, September 1972.

[Wegiel and Krintz, 2008a] Michal Wegiel and Chandra Krintz. The mapping collector: Virtual memory support for generational, parallel, and concurrent compaction. In Eggers and Larus [Eggers and Larus2008], pages 91–102.

[Wegiel and Krintz, 2008b] Michal Wegiel and Chandra Krintz. XMem: Type-safe, transparent, shared memory for cross-runtime communication and coordination. In Gupta and Amarasinghe [Gupta and Amarasinghe2008], pages 327–338.

[Wegiel and Krintz, 2009] Michal Wegiel and Chandra Krintz. The single-referent collector: Optimizing compaction for the common case. *ACM Transactions on Architecture and Code Optimization*, 2009.

[Wegiel and Krintz, 2010] Michal Wegiel and Chandra Krintz. Cross-language, type-safe, and transparent object sharing for co-located managed runtimes. In OOPSLA 2010 [OOPSLA 20102010], pages 223–240.

[Weinreb and Moon, 1981] Daniel Weinreb and David Moon. *LISP Machine Manual*. MIT AI Laboratory, fourth edition edition, July 1981.

[Weinstock and Wulf, 1988] Charles B. Weinstock and William A. Wulf. Quickfit: An efficient algorithm for heap storage allocation. *ACM SIGPLAN Notices*, 23(10):141–144, 1988.

[Weinstock, 1976] Charles B. Weinstock. *Dynamic Storage Allocation Techniques*. PhD thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, April 1976.

[Weiser *et al.*, 1989] Mark Weiser, Alan Demers, and Carl Hauser. The Portable Common Runtime approach to interoperability. In *Twelfth ACM Symposium on Operating Systems Principles*. ACM Press, December 1989.

[Weissman, 1967] C. Weissman. *Lisp 1.5 Primer*. Dickenson Publ., Belmont, CA, 1967.

[Weizenbaum, 1962] J. Weizenbaum. Knotted list structures. *Communications of the ACM*, 5(3):161–165, 1962.

[Weizenbaum, 1963] J. Weizenbaum. Symmetric list processor. *Communications of the ACM*, 6(9):524–544, September 1963.

[Weizenbaum, 1964] J. Weizenbaum. More on the reference counter method. *Communications of the ACM*, 7(1):38, 1964.

[Weizenbaum, 1969] J. Weizenbaum. Recovery of reentrant list structures in SLIP. *Communications of the ACM*, 12(7):370–372, July 1969.

[Weng, 1979] K.-S. Weng. An abstract implementation for a generalised dataflow language. Technical Report MIT/LCS/TR228, MIT Laboratory for Computer Science, 1979.

[Wentworth, 1988] E. P. Wentworth. *An Environment for Investigating Functional Languages and Implementations*. PhD thesis, University of Port Elizabeth, South Africa, 1988.

[Wentworth, 1990] E. P. Wentworth. Pitfalls of conservative garbage collection. *Software Practice and Experience*, 20(7):719–727, 1990.

[Whaley and Rinard, 1999] John Whaley and Martin Rinard. Compositional pointer and escape analysis for Java programs. In OOPSLA 1999 [OOPSLA 19991999], pages 187–206. By analysing which objects escape methods/threads, we can allocate on the stack / avoid synchronisations.

[While and Field, 1992] R. Lyndon While and Tony Field. Incremental garbage collection for the Spineless Tagless G-machine. In Evan Ireland and Nigel Perry, editors, *Proceedings of the Massey Functional Programming Workshop 1992*. Department of Computer Science, Massey University, 1992.

[While, 1992] R. Lyndon While. A viable software read-barrier. Departmental Report DoC 92/12, Imperial College, London, 1992.

[White and Dewitt, 1992] Seth J. White and David J. Dewitt. A performance study of alternative object faulting and pointer swizzling strategies. In *18th International Conference on Very Large Data Bases*, Vancouver, British Columbia, Canada, October 1992.

[White and Dewitt, 1994] Seth J. White and David J. Dewitt. Quickstore: A high performance mapped object store. In *SIGMOD94*, pages 395–406, 1994.

[White and Garthwaite, 1998] Derek White and Alex Garthwaite. The GC interface in the EVM. Technical Report SML TR–98–67, Sun Microsystems Laboratories, December 1998.

[White, 1980] Jon L. White. Address/memory management for a gigantic Lisp environment, or, GC Considered Harmful. In *Conference Record of the 1980 Lisp Conference*, pages 119–127, Redwood Estates, CA, August 1980.

[White, 1990] Jon L. White. Three issues in objected-oriented garbage collection. In Jul and Juul [Jul and Juul1990].

[White, 1997] Marc White. Deferred garbage collection. *MacTech magazine*, 13(12), 1997.

[Wholey and Fahlman, 1984] Skef Wholey and Scott E. Fahlman. The design of an instruction set for Common Lisp. In Steele [Steele1984], pages 150–158.

[Wholey *et al.*, 1985] Skef Wholey, Scott E. Fahlman, and Joseph Ginder. Revised internal design of Spice Lisp. Technical report, Carnegie Mellon University, January 1985.

[Wick and Flatt, 2004] Adam Wick and Matthew Flatt. Memory accounting without partitions. In Bacon and Diwan [Bacon and Diwan2004], pages 120–130.

[Wild *et al.*, 1991] J. Wild, Hugh Glaser, and Pieter Hartel. Statistics on storage management in a lazy functional language implementation. In Sendov, editor, *Proceedings of Third Workshop on Parallel and Distributed Processing*, Sofia, April 1991. Elsevier-North Holland. Also appears in Proceedings of 4th International Workshop on Parallel Implementation of Functional Languages, Aachen, September 1992, ed. H. Kuchen and R. Loogen, Aachener Informatik-Berichte 92–16.

[Wileden *et al.*, 1989] J. C. Wileden, A. L. Wolf, C. D. Fisher, and P. L. Tarr. Pgraphite: An experiment in persistent typed object management. *ACM SIGPLAN Notices*, 24(2):130–142, February 1989.

[Wilhelmsson, 2005] Jesper Wilhelmsson. *Efficient Memory Management for Message-Passing Concurrency — part I: Single-threaded execution*. Licentiate thesis, Uppsala University, May 2005.

[Wilkes, 1964a] Maurice V. Wilkes. An experiment with a self-compiling compiler for a simple list-processing language. *Annual Review in Automatic Programming*, 4:1–48, 1964.

[Wilkes, 1964b] Maurice V. Wilkes. Lists and why they are useful. In *Proceedings of the ACM 19th National Conference*. ACM Press, August 1964.

[Willard and Frieder, 1998] B. Willard and O. Frieder. Autonomous garbage collection: Resolving memory leaks in long running network applications. In *Proceedings of the International Conference On Computer Communications and Networks (ICCCN 1998)*, pages 886–896, Lafayette, LO, October 1998. IEEE Press.

[Willard and Frieder, 2000] B. Willard and O. Frieder. Autonomous garbage collection: Resolving memory leaks in long-running server applications. *Computer Communications*, 23(10):887–900, May 2000.

[Williams and Wolczko, 1991] Ifor W. Williams and Mario I. Wolczko. An object-based memory architecture. In Dearle et al. [Dearle *et al.*1991], pages 114–130.

[Williams *et al.*, 1987a] Ifor W. Williams, Mario I. Wolczko, and T. P. Hopkins. Realisation of a dynamic grouped object-oriented virtual memory hierarchy. In Carrick and Cooper [Carrick and Cooper1987], pages 298–308. Persistent Programming Research Report, Universities of Glasgow and St Andrews, number PPRR–44–87.

[Williams *et al.*, 1987b] Ifor W. Williams, Mario I. Wolczko, and Trevor P. Hopkins. Dynamic grouping in an object-oriented virtual memory hierarchy. In Bézivin et al. [Bézivin *et al.*1987], pages 79–88.

[Williams *et al.*, 1990] Ifor Williams, Mario I. Wolczko, and Trevor Hopkins. Realization of a dynamically grouped object-oriented memory hierarchy. Technical report, University of Manchester Department of Computer Science, Manchester, 1990.

[Wilson and Chaudrhy, 1992] Paul R. Wilson and Atif Chaudrhy. Efficient tracing using virtual memory address protection. In preparation, 1992.

[Wilson and Hayes, 1991a] Paul R. Wilson and Barry Hayes, editors. *OOPSLA Workshop on Garbage Collection in Object-Oriented Systems*, October 1991.

[Wilson and Hayes, 1991b] Paul R. Wilson and Barry Hayes. Report on the 1991 workshop on garbage collection in object-oriented systems. In OOPSLA 1991 [OOPSLA 19911991]. Addendum. Also distributed as a special issue of ACM SIGPLAN Notices, and OOPS Messenger 3(4), October 1992.

[Wilson and Johnstone, 1993a] Paul R. Wilson and Mark Johnstone. Truly real-time non-moving implicit-reclamation garbage collection. Rough Draft, March 1993.

[Wilson and Johnstone, 1993b] Paul R. Wilson and Mark S. Johnstone. Truly real-time non-copying garbage collection. In Moss et al. [Moss *et al.*1993].

[Wilson and Kakkad, 1992] Paul R. Wilson and Sheetal V. Kakkad. Pointer swizzling at page fault time: Efficiently and compatibly supporting huge addresses on standard hardware. In Cabrera et al. [Cabrera *et al.*1992], pages 364–377.

[Wilson and Moher, 1989a] Paul R. Wilson and Thomas G. Moher. A card-marking scheme for controlling intergenerational references in generation-based garbage collection on stock hardware. *ACM SIGPLAN Notices*, 24(5):87–92, 1989.

[Wilson and Moher, 1989b] Paul R. Wilson and Thomas G. Moher. Design of the opportunistic garbage collector. In OOPSLA 1989 [OOPSLA 19891989], pages 23–35.

[Wilson *et al.*, 1990] Paul R. Wilson, Michael S. Lam, and Thomas G. Moher. Caching consideration for generational garbage collection: A case study of large and set-associative caches. Technical Report UIC–EECS–90–5, University of Illinois at Chicago EECS Department, Chicago, Illinois, December 1990. Improved version appears in [LFP 1992, 1992].

[Wilson *et al.*, 1991] Paul R. Wilson, Michael S. Lam, and Thomas G. Moher. Effective "static-graph" reorganization to improve locality in garbage-collected systems. In PLDI 1991 [PLDI 19911991], pages 177–191.

[Wilson *et al.*, 1992a] Paul R. Wilson, Michael S. Lam, and Thomas G. Moher. Caching considerations for generational garbage collection. In LFP 1992 [LFP 19921992], pages 32–42.

[Wilson *et al.*, 1992b] Paul R. Wilson, Shubhendu S. Mukherjee, and Sheetal V. Kakkad. Anomalies and adaptation in the analysis and development of prepaging p olicies. *Journal of Systems and Software*, 1992.

[Wilson *et al.*, 1994] Paul R. Wilson, Shubhendu S. Mukherjee, and Sheetal V. Kakkad. Anomalies and adaptation in the analysis and development of prepaging policies. *Journal of Systems and Software*, 1994. Technical Communication.

[Wilson *et al.*, 1995a] Paul R. Wilson, Mark S. Johnstone, Michael Neely, and David Boles. Dynamic storage allocation: A survey and critical review. In Baker [Baker1995a].

[Wilson *et al.*, 1995b] Paul R. Wilson, Mark S. Johnstone, Michael Neely, and David Boles. Memory allocation policies reconsidered. Unpublished manuscript, 1995.

[Wilson, 1988a] Paul R. Wilson. Opportunistic garbage collection. *ACM SIGPLAN Notices*, 23(12):98–102, December 1988.

[Wilson, 1988b] Paul R. Wilson. Two comprehensive virtual copy mechanisms. Master's thesis, University of Illinois at Chicago, Electrical Engineering and Computer Science Department, Chicago, Illinois, 1988.

[Wilson, 1989] Paul R. Wilson. A simple bucket-brigade advancement mechanism for generation-based garbage collection. *ACM SIGPLAN Notices*, 24(5):38–46, May 1989.

[Wilson, 1990a] Paul R. Wilson. Pointer swizzling at page fault time: Efficiently supporting huge address spaces on standard hardware. Technical Report UIC–EECS–90–6, University of Illinois at Chicago, Electrical Engineering and Computer Science Department, Chicago, Illinois, December 1990. Also in Computer Architecture News, 19(4):6–13, June 1991.

[Wilson, 1990b] Paul R. Wilson. Some issues and strategies in heap management and memory hierarchies. In Jul and Juul [Jul and Juul1990]. Also in SIGPLAN Notices 23(1):45–52, January 1991.

[Wilson, 1991] Paul R. Wilson. *Heap Management and Memory Hierarchies*. PhD thesis, University of Illinois at Chicago, December 1991.

[Wilson, 1992a] Paul R. Wilson. Garbage collection and memory hierarchy. In Bekkers and Cohen [Bekkers and Cohen1992].

[Wilson, 1992b] Paul R. Wilson. Operating system support for small objects. In Cabrera et al. [Cabrera *et al.*1992].

[Wilson, 1992c] Paul R. Wilson. Uniprocessor garbage collection techniques. In Bekkers and Cohen [Bekkers and Cohen1992].

[Wilson, 1994] Paul R. Wilson. Uniprocessor garbage collection techniques. Technical report, University of Texas, January 1994. Expanded version of the IWMM92 paper.

[Winsborough, 1989] William Winsborough. Path-dependent reachability analysis for multiple specialization. In E. Lusk and R. Overbeek, editors, *North American Conference on Logic Programming*, pages 113–153. MIT Press, 1989.

[Wise and Friedman, 1977a] David S. Wise and Daniel P. Friedman. The one-bit reference count. In *BIT* [Wise and Friedman1977b], pages 351–9.

[Wise and Friedman, 1977b] David S. Wise and Daniel P. Friedman. The one-bit reference count. *BIT*, 17(3):351–9, 1977.

[Wise and Walgenbach, 1996] David S. Wise and J. Walgenbach. Static and dynamic partitioning of pointers as links and threads. In *Proceedings of the 1996 ACM SIGPLANInternational Conference on Functional Programming(ICFP96)*, ACM SIGPLAN Notices 31(6), pages 42–49. ACM Press, June 1996.

[Wise and Watson, 1976] David S. Wise and Daniel C. Watson. Tuning Garwick's algorithm for repacking sequential storage. *BIT*, 16(4):442–450, December 1976.

[Wise *et al.*, 1994] David S. Wise, Brian Heck, Caleb Hess, Willie Hunt, and Eric Ost. Uniprocessor performance of a reference-counting hardware heap. Technical Report TR-401, Indiana University, Computer Science Department, May 1994.

[Wise *et al.*, 1997] David S. Wise, Caleb Hess, Willie Hunt, and Eric Ost. Research demonstration of a hardware reference-counting heap. *Lisp and Symbolic Computation*, 10(2):151–181, July 1997.

[Wise, 1978] David S. Wise. The double buddy-system. Technical Report 79, Computer Science Department, Indiana University, Bloomington, Indiana, December 1978.

[Wise, 1979] David S. Wise. Morris' garbage compaction algorithm restores reference counts. *ACM Transactions on Programming Languages and Systems*, 1:115–120, July 1979.

[Wise, 1985] David S. Wise. Design for a multiprocessing heap with on-board reference counting. In Jouannaud [Jouannaud1985], pages 289–304.

[Wise, 1993a] David S. Wise. Stop-and-copy and one-bit reference counting. Technical Report 360, Indiana University, Computer Science Department, March 1993.

[Wise, 1993b] David S. Wise. Stop-and-copy and one-bit reference counting. *Information Processing Letters*, 46(5):243–249, July 1993.

[Wiseman, 1966] N. E. Wiseman. A simple list processing package for the PDP–7. In *DECUS Second European Seminar, Aachen*, pages 37–42, October 1966.

[Wiseman, 1985] Simon R. Wiseman. A garbage collector for a large distributed address space. Technical Report 85009, Royal Signals and Radar Establishment, Malvern, UK, 1985.

[Wiseman, 1988] Simon R. Wiseman. *Garbage Collection in Distributed Systems*. PhD thesis, University of Newcastle upon Tyne, 1988.

[Withington, 1991] P. Tucker Withington. How real is "real time" garbage collection? In Wilson and Hayes [Wilson and Hayes1991a].

[Wodon, 1969] P. L. Wodon. Data structure and storage allocation. *BIT*, 9(3):270–282, 1969.

[Wodon, 1971] P. L. Wodon. Methods of garbage collection for Algol–68. In Peck [Peck1971], pages 245–262.

[Wolczko and Williams, 1990] Mario I. Wolczko and Ifor Williams. Garbage collection in high performance system. In Jul and Juul [Jul and Juul1990].

[Wolczko and Williams, 1992] Mario I. Wolczko and Ifor Williams. Multi-level GC in a high-performance persistent object system. In Albano and Morrison [Albano and Morrison1992], pages 396–418.

[Wolczko and Williams, 1993] Mario I. Wolczko and Ifor Williams. An alternative architecture for objects: Lessons from the MUSHROOM project. In Moss et al. [Moss *et al.*1993].

[Wolczko, 1990] Mario I. Wolczko. Garbage collection. In Cliff B. Jones and Roger C. F. Shaw, editors, *Case Studies in Systematic Software Development*, chapter 8, pages 211–233. Prentice-Hall, 1990.

[Wolf, 1989] W. Wolf. Practical comparison of two object-oriented languages. *IEEE Software*, 6(5):61–68, 1989.

[Wollrath *et al.*, 1996] A. Wollrath, R. Riggs, and Jim Waldo. A distributed object model for the Java system. In *Conference on Object-Oriented Technologies*, Toronto, Canada, June 1996. Usenix.

[Wong, 1987] K. F. Wong. Garbage collection coprocessor system. *Electronics Letters*, 23(15):798–800, 1987.

[Woodward, 1981] M. C. Woodward. Multiprocessor garbage collection — a new solution. *Computer Studies*, 115, 1981.

[Wright and Baker-Finch, 1993] D. A. Wright and C. A. Baker-Finch. Usage analysis with natural reduction types. In *Third International Workshop on Static Analysis*, 1993.

[Wright *et al.*, 2005] Greg Wright, Matthew L. Seidl, and Mario Wolczko. An object-aware memory architecture. Technical Report SML–TR–2005–143, Sun Microsystems Laboratories, February 2005.

[Wright *et al.*, 2006] Greg Wright, Matthew L. Seidl, and Mario Wolczko. An object-aware memory architecture. In *Science of Computer Programming* [Jones2006], pages 145–163.

[Wu and Li, 2007] Ming Wu and Xiao-Feng Li. Task-pushing: a scalable parallel GC marking algorithm without synchronization operations. In *IEEE International Parallel and Distribution Processing Symposium (IPDPS) 2007*, Long Beach, CA, March 2007.

[Wu, 1989a] Meizhen Wu. A parallel garbage collection algorithm for virtual memory. Master's thesis, Auburn University, 1989.

[Wu, 1989b] Zhibo Wu. A real time distributed garbage collection method. Research report R 181, University of Exeter. Department of Computer Science, 1989. Year maybe 1992.

[Xian and Xiong, 2005] Yuqiang Xian and Guangze Xiong. Minimizing memory requirement of real-time systems with concurrent garbage collector. *ACM SIGPLAN Notices*, 40(3):40–48, March 2005.

[Xian *et al.*, 2006] Feng Xian, Witawas Srisa-an, and Hong Jiang. Investigating the throughput degradation behavior of Java application servers: A view from inside the virtual machine. In *Proceedings of the 4th International Conference on Principles and Practices of Programming in Java*, pages 40–49, Mannheim, Germany, 2006.

[Xian *et al.*, 2007a] Feng Xian, Witawas Srisa-an, C. Jia, and Hong Jiang. AS-GC: An efficient generational garbage collector for Java application servers. In *European Conference On Object-Oriented Programming (ECOOP)*, Berlin, Germany, July 2007.

[Xian *et al.*, 2007b] Feng Xian, Witawas Srisa-an, and Hong Jiang. Allocation-phase aware thread scheduling policies to improve garbage collection performance. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 79–90.

[Xian *et al.*, 2007c] Feng Xian, Witawas Srisa-an, and Hong Jiang. Evaluating hardware support for reference counting using software configurable processors. In *Proceedings of IEEE 17th International Conference on Application-Specific Systems, Architectures, and Processors*, pages 297–302, Steamboat Springs, CO, September 2007.

[Xian *et al.*, 2007d] Feng Xian, Witawas Srisa-an, and Hong Jiang. MicroPhase: An approach to proactively invoking garbage collection for improved performance. In OOPSLA 2007 [OOPSLA 20072007], pages 77–96.

[Xian *et al.*, 2008] Feng Xian, Witawas Srisa-an, and Hong Jiang. Garbage collection: Java application servers' Achilles heel. *Science of Computer Programming*, 70(2–3), February 2008.

[Yamamoto *et al.*, 1998a] Hirotaka Yamamoto, Kenjiro Taura, and Akinori Yonezawa. Comparing reference counting and global mark-and-sweep on parallel computers. In *Proceedings of Languages, Compilers, and Run-time Systems (LCR98)*, Lecture Notes in Computer Science, May 1998.

[Yamamoto *et al.*, 1998b] Hirotaka Yamamoto, Kenjiro Taura, and Akinori Yonezawa. Performance comparison between reference counting and global GC on distributed-memory parallel computers. In *Proceedings of Joint Symposium on Parallel Processing (JSPP)*, June 1998. In Japanese.

[Yang *et al.*, 2002] Qian Yang, Witawas Srisa-an, Therapon Skotiniotis, and J. Morris Chang. Java virtual machine probes — a study of object life span and GC. In *Proceedings of 21st IEEE International Performance, Computing and Communications Conference (IPCCC)*, Phoenix, AZ, April 2002.

[Yang *et al.*, 2004] Ting Yang, Emery D. Berger, Matthew Hertz, Scott F. Kaplan, and J. Eliot B. Moss. Autonomic heap sizing: Taking real memory into account. In Bacon and Diwan [Bacon and Diwan2004], pages 61–72.

[Yang *et al.*, 2006] Ting Yang, Emery D. Berger, Scott F. Kaplan, and J. Eliot B. Moss. CRAMM: Virtual memory support for garbage-collected applications. In *7th USENIX Symposium on Operating System Design and Implementation*. USENIX Association, 2006.

[Yang, 2001] Hongseok Yang. An example of local reasoning in BI pointer logic: the Schorr-Waite graph marking algorithm. In SPACE 2001 [SPACE 20012001].

[Yasugi and Yonezawa, 1991] Masahiro Yasugi and Akinori Yonezawa. Towards user (application) language-level garbage collection in object-oriented concurrent languages. In Wilson and Hayes [Wilson and Hayes1991a].

[Ye and Keane, 1997] Xinfeng Ye and John Keane. Collecting cyclic garbage in distributed systems. In *International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '97)*, Taipei, Taiwan, December 1997.

[Yeates and de Champlain, 1997a] Stuart A. Yeates and Michel de Champlain. Design of a garbage collector using design patterns. In Christine Mingins, Roger Duke, and Bertrand Meyer, editors, *Proceedings of the Twenty-Fifth Conference of (TOOLS) Pacific.*, pages 77–92, Melbourne, 1997. ISE (Interactive Software Engineering).

[Yeates and de Champlain, 1997b] Stuart A. Yeates and Michel de Champlain. Design patterns in garbage collection. In Robert S. Hanmer and Don Roberts, editors, *Proceedings of the 4th Annual Conference on the Pattern Languages of Programs*, volume 6 "General Techniques", Monticello, IL, 2-5 September 1997. Published as technical report # WUCS-97-34 of Washington University (not peer reviewed).

[Yeates, 1997] Stuart Yeates. Design patterns in garbage collection. Master's thesis, University of Canterbury, Christchurch, New Zealand, June 1997.

[Yelowitz and Duncan, 1977] L. Yelowitz and A. G. Duncan. Abstractions, instantiations and proofs of marking algorithms. *ACM SIGPLAN Notices*, 12(8):13–21, August 1977.

[Yi and Harrison III, 1992] Kwangkeun Yi and Williams Ludwell Harrison III. Interprocedural data flow analysis for compile-time memory management. Technical Report 1244, Center for Supercomputing Research and Development, University of Illinois, August 1992.

[Yip, 1991] G. May Yip. Incremental, generational mostly-copying garbage collection in uncooperative environments. Technical Report 91/8, Digital, Western Research Laboratory, June 1991. Masters Thesis — MIT, Cambridge, MA, 1991.

[Yong and amd J.-B. Yu, 1994] V.-F. Yong and J. Naughton amd J.-B. Yu. Storage reclamation and reorganization in client–server persistent object stores. In *International Conference on Data Engineering*, pages 120–133, Houston, TX, February 1994.

[Yu and Cox, 1996] W. Yu and A. Cox. Conservative garbage collection on distributed shared memory system. In *Sixth International Conference on Distributed Computing Systems ICDCS96*, pages 402–410, May 1996.

[Yu *et al*., 2003] Dachuan Yu, Nadeem A. Hamid, and Zhong Shao. Building certified libraries for PCC: Dynamic storage allocation. In ESOP 2003 [ESOP 20032003], pages 101–127. Also published in Science of Computer Programming, 50(1–3), March 2004.

[Yu *et al*., 2008] Zoe C. H. Yu, Francis C. M. Lau, and Cho-Li Wang. Object co-location and memory reuse for Java programs. *ACM Transactions on Architecture and Code Optimization*, 4(4):1–36, 2008.

[Yuasa and Hagiya, 1985] Taiichi Yuasa and Masumi Hagiya. Kyoto Common Lisp report. Technical report, Teikoku Insatsu Publishing, Kyoto, 1985.

[Yuasa, 1990] Taichi Yuasa. Real-time garbage collection on general-purpose machines. *Journal of Systems and Software*, 11(3):181–198, 1990.

[Yuasa, 1992] Taichi Yuasa. Memory management and garbage collection of an extended Common Lisp system for massively parallel SIMD architecture. In Bekkers and Cohen [Bekkers and Cohen1992].

[Yuasa, 2002] Taichi Yuasa. Return barrier. In *Proceedings of the International Lisp Conference 2002*, 2002.

[Yuhara *et al.*, 1986] M. Yuhara, A. Hattori, M. Niwa, M. Kishimoto, and H. Hayashi. Evaluation of the Facom Alpha Lisp machine. In *13th Annual International Symposium on Computer Architecture — Conference Proceedings. Tokyo, 1986 June 2–5*. IEEE Press, 1986.

[Zabel *et al.*, 2007] Martin Zabel, Thomas B. Preuber, Peter Reichel, and Rainer G. Spallek. Secure, real-time and multi-threaded general-purpose embedded Java microarchitecture. In *DSD '07: Proceedings of the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools*, pages 59–62. IEEE Computer Society, 2007.

[Zave, 1975] Derek A. Zave. A fast compacting garbage collector. *Information Processing Letters*, 3(6):167–169, July 1975.

[Zee and Rinard, 2002a] Karen Zee and Martin Rinard. Write barrier removal by static analysis. *ACM SIGPLAN Notices*, 37(4), April 2002.

[Zee and Rinard, 2002b] Karen Zee and Martin Rinard. Write barrier removal by static analysis. In OOPSLA 2002 [OOPSLA 20022002], pages 191–210.

[Zendra, 2006] Olivier Zendra. Memory and compiler optimizations for low-power and energy. In *Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS'2006)*, page 8, Nantes, France, July 2006.

[Zhang and Hirzel, 2008] Chengliang Zhang and Martin Hirzel. Online phase-adaptive data layout selection. In ECOOP 2008 [ECOOP 20082008], pages 309–334.

[Zhang *et al.*, 1998] Sean Zhang, Barbara G. Ryder, and William A. Landi. Experiments with combined analysis for pointer aliasing. In PASTE98 [PASTE981998].

[Zhang *et al.*, 2006] Chengliang Zhang, Kirk Kelsey, Xipeng Shen, Chen Ding, Matthew Hertz, and Mitsunori Ogihara. Program-level adaptive memory management. In Petrank and Moss [Petrank and Moss2006], pages 174–183.

[Zhao *et al.*, 1987] W. Zhao, K. Ramamritham, and J. A. Stankovic. Scheduling tasks with resource requirements in hard real-time systems. *ACM Transactions on Software Engineering*, 5(13):564–577, May 1987.

[Zhao *et al.*, 2009] Yi Zhao, Jin Shi, Kai Zheng, Haichuan Wang, Haibo Lin, and Ling Shao. Allocation wall: A limiting factor of Java applications on emerging multi-core platforms. In OOPSLA 2009 [OOPSLA 20092009], pages 361–376.

[zhe Han *et al.*, 2006] Long zhe Han, Yeonseung Ryu, Tae sun Chung, Myungho Lee, and Sukwon Hong. An intelligent garbage collection algorithm for flash memory storages. In *Computational Science and its Applications, ICCSA 2006*, volume 3980 of *Lecture Notes in Computer Science*, pages 1018–1027, 2006.

[Zhong and Chang, 2008] Yutao Zhong and Wentao Chang. Sampling-based program locality approximation. In Jones and Blackburn [Jones and Blackburn2008], pages 91–100.

[Zhou *et al.*, 1992] Songnian Zhou, Michael Stumm, Kai Li, and David Wortman. Heterogeneous distributed shared memory. *IEEE Transactions on Parallel and Distributed Systems*, 3(5):540–554, September 1992.

[Zhou, 2000] Neng-Fa Zhou. Garbage collection in B-Prolog. In *First Workshop on Memory Management in Logic Programming Implementations*, July 2000.

[Zigman *et al.*, 2001] John Zigman, Stephen M. Blackburn, and J. Eliot B. Moss. TMOS: a transactional garbage collector. In Kirby et al. [Kirby *et al.*2001], pages 116–135.

[Zigman, 2004] John Zigman. *A General Framework for the Description and Construction of Hierarchical Garbage Collection Algorithms*. PhD thesis, Australian National University, 2004.

[Zilles, 2007] Craig Zilles. Accordion arrays: Selective compression of Unicode arrays in Java. In Morrisett and Sagiv [Morrisett and Sagiv2007], pages 55–66.

[Zorn and Grunwald, 1992a] Benjamin Zorn and Dirk Grunwald. Empirical measurements of six allocation-intensive C programs. Computer Science Technical Report CU-CS-604-92, University of Colorado, July 1992.

[Zorn and Grunwald, 1992b] Benjamin Zorn and Dirk Grunwald. Empirical measurements of six allocation-intensive C programs. *ACM SIGPLAN Notices*, 27(12):71–80, 1992.

[Zorn and Grunwald, 1992c] Benjamin Zorn and Dirk Grunwald. Evaluating models of memory allocation. Computer Science Technical Report CU-CS-603-92, University of Colorado, July 1992.

[Zorn and Grunwald, 1994] Benjamin Zorn and Dirk Grunwald. Evaluating models of memory allocation. *ACM Transactions on Modelling and Computer Simulation*, 4(1), 1994.

[Zorn and Seidl, 1998] Benjamin Zorn and M. Seidl. Segregating heap objects by reference behavior and lifetime. In ASPLOS 1998 [ASPLOS 19981998], pages 12–23.

[Zorn *et al.*, 1987] Benjamin Zorn, Paul Hilfinger, Kinson Ho, and James R. Larus. SPUR Lisp: Design and implementation. Technical Report UCB/CSD 87/373, University of California, Berkeley, October 1987.

[Zorn *et al.*, 1988] Benjamin Zorn, Paul Hilfinger, Kinson Ho, and James R. Larus. A memory allocation profiler for C and Lisp programs. In *Proceedings for the Summer 1988 USENIX Conference*, pages 223–237, June 1988.

[Zorn, 1989] Benjamin G. Zorn. *Comparative Performance Evaluation of Garbage Collection Algorithms*. PhD thesis, University of California at Berkeley, March 1989. Technical Report UCB/CSD 89/544.

[Zorn, 1990a] B. Zorn. Designing systems for evaluation: A case study of garbage collection. In Jul and Juul [Jul and Juul1990].

[Zorn, 1990b] Benjamin Zorn. Barrier methods for garbage collection. Technical Report CU-CS-494-90, University of Colorado, Boulder, November 1990.

[Zorn, 1990c] Benjamin Zorn. Comparing mark-and-sweep and stop-and-copy garbage collection. In LFP 1990 [LFP 19901990].

[Zorn, 1991] Benjamin Zorn. The effect of garbage collection on cache performance. Technical Report CU–CS–528–91, University of Colorado at Boulder, May 1991.

[Zorn, 1992] Benjamin Zorn. The measured cost of garbage collection. Technical Report CU–CS–573–92, University of Colorado at Boulder, Department of Computer Science, Boulder, Colorado, April 1992.

[Zorn, 1993] Benjamin Zorn. The measured cost of conservative garbage collection. *Software Practice and Experience*, 23:733–756, 1993.