This bibliography may be freely used for non-commercial purposes. It may also be freely distributed provided that this notice is included. I would be most grateful to receive additions, corrections and URLs of electronically available papers. The bibliography is also available in BibTeX and HTML forms from https://www.cs.kent.ac.uk/people/staff/rej/gcbib/gcbib.html

Copyright © 1999-2020, Richard Jones


[Amsoleg 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.


[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].


15


[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].


[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 2002]*.


[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.


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


[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.


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


[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.


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.


47


[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

[Hennessey, 1993] Wade Hennessey. Real-time garbage collection in a multimedia programming lan-
guage. In Moss et al. [Moss et al., 1993].


[Henning, 1998] Michi Henning. Binding, migration and scalability in CORBA. *Communications of the

[Henriksson, 1994] Roger Henriksson. Scheduling real-time garbage collection. In *Proceedings of NW-


[Henriksson, 1997] Roger Henriksson. Predictable automatic memory management for embedded sys-
tems. In Dickman and Wilson [Dickman and Wilson, 1997].


on Embedded Software (EMSOFT)*, volume 2211 of *Lecture Notes in Computer Science*, Tahoe City,


[Herlihy and Moss, 1992] Maurice Herlihy and J. Eliot B. Moss. Lock-free garbage collection for multi-


Morgan Kaufman, April 2008.

[Herlihy and Wing, 1990] Maurice Herlihy and Jeannette M. Wing. Linearizability: A correctness con-
dition 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 prob-
lem: A mechanism for supporting dynamic-sized lock-free data structures. In *Proceedings of the 16th

[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 manage-
ment: 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 2005]*)

[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.


58


[Jinsight, Jinsight]. Visualisation tools for Java.


[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.


[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.


[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].


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].


[Moss et al., 1993] Eliot Moss, Paul R. Wilson, and Benjamin Zorn, editors. OOPSLA Workshop on Garbage Collection in Object-Oriented Systems, October 1993.


83


[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.


[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.


[Plainfossée and Shapiro, 1992] David Plainfossée and Marc Shapiro. A distributed GC in an object-support operating system. In Cabrera et al. [Cabrera et al.1992].


93


[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].


[Richter 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.


garbage collection with group merger. In Jul [Jul998], pages 249–273. Also UKC Technical re-

puting Laboratory, The University of Kent at Canterbury, 1998.

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 Fiter-
man. A non-fragmenting, non-moving garbage collector. In Peyton Jones and Jones
[Peyton Jones and Jones1998], pages 79–85.

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 dis-
tributed garbage collection without global synchronization in CORBA. In Dickman and Wilson
[Dickman and Wilson1997].


profiling and space-efficient compilation revisited. In ICFP 1996 [ICFP 19961996], pages 34–41.

In Bekkers and Cohen [Bekkers and Cohen1992].

of Computer Science, Chalmers University, January 1993.

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 Lan-

[Röjemo, 1995b] Niklas Röjemo. Generational garbage collection without temporary space leaks for

1995 [FPCA 19951995].

for data race detection. In E. D’Hollander, F.J. Joubert, and U. Trottenberg, editors, Parallel Comput-
ing: Fundamentals, Applications and New Directions, volume 12 of Advances in Parallel Computing,

[Rose and Muller, 1992] John H. Rose and Hans Muller. Integrating the Scheme and C languages. In

MONADS architecture. In Carrick and Cooper [Carrick and Cooper1987].

[Rosenberg and Koch, 1989] John Rosenberg and David Koch, editors. Proceedings of the Third Inter-
national Workshop on Persistent Object Systems (January, 1989), Workshops in Computing, Newcastle,
NSW, Australia, 1989. Springer.

Stability in a persistent store based on a large virtual memory. In International Workshop on Ar-
chitectural Support for Security and Persistence of Information, pages 229–245. Springer Verlag and the

[Cabrera et al.1991], pages 48–60.


[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.


104


[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].


[SPIN, ] The SPIN operating system. A collection of papers available on the WWW.


[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 Ball 2006], pages 283–294.


[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].


121


Wilson, 1992b] Paul R. Wilson. Operating system support for small objects. In Cabrera et al. [Cabrera et al.1992].


