the Garbage Collection Bibliography

Richard Jones
Computing Laboratory
University of Kent at Canterbury
August 31, 2017

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 ©1999-2017, Richard Jones


[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 (VLDB’95), Zurich, Switzerland, September 1995.


[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.


[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].


25


[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.


[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 Blackburn 2008], pages 101–110.


[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.


51


[Hicks, 1993] James Hicks. Experiences with compiler-directed storage reclamation. In Hughes [Hughes1993].


[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.


ScriptX Architectural Overview.


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.


[Kurita et al., 1990] Satoshi Kurita, 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].


Kai Li. Real-time concurrent collection in user mode. In Jul and Juul [Jul and Juul1990].


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


[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.


[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.


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


100


[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.


[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.


108


[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.


[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 Ball2006], 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].


[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.


[Ungureanu and Goldberg, 1997] Christian Ungureanu and Benjamin Goldberg. Formal models of

[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. Au-
tomatic 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

1995.

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

[van de Snepscheut, 1987] Jan van de Snepscheut. Algorithms for on-the-fly garbage collection re-

and F. Huch, editors, Proceedings of Sixteenth International Workshop on Implementation and


October 1997.

[Vataja and Ukkonen, 1984] P. Vataja and E. Ukkonen. Finding temporary terms in PROLOG pro-

[Vaughan and Dearle, 1992] Francis Vaughan and Alan Dearle. Supporting large persistent stores
using conventional hardware. In Albano and Morrison [Albano and Morrison1992].

A persistent distributed architecture supported by the mach operating system. In First USENIX

[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 Pro-

[Vechev and Bacon, 2004] Martin Vechev and David F. Bacon. Write barrier elision for concurrent

[Vechev and Petrov, 2003] Martin T. Vechev and Peter D. Petrov. Class unloading with a concur-
rent 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.


