The Garbage Collection Bibliography
Richard Jones
Computing Laboratory
University of Kent at Canterbury
April 4, 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/gc bib.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 (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].


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

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


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.


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].

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 2002], pages 185–188.


[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 Blackburn2008], 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 Blackburn 2008], pages 1–10.


38


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

52


[LFP 1990, 1990] Proceedings of the ACM Conference on Symposium on Lisp and Functional Pro-
gramming, Nice, France, June 1990.


[LFP 1994, 1994] Proceedings of the ACM Conference on Symposium on Lisp and Functional Pro-
gramming, Orlando, FL, USA, June 1994.

[Lhotáš and Hendren, 2002] Ondrej Lhotáš and Laurie Hendren. Run-time evaluation of opportuni-
ties for object inlining in Java. In Joint ACM Java Grande - ISCOPE Conference, pages 175–184,

Yale University, February 1985.


University, 1986.


GCs.

based on the lifetimes of objects. AI Memo 569a, MIT, April 1981.

TM–184, Laboratory for Computer Science, MIT, Cambridge, MA, July 1980 and AI Lab Memo
569, 1981.

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.

embedded Java chips. In RTCSA 2000 [RTCSA 20002000].

algorithm. In Proceedings of the International Conference on Grid and Pervasive Computing,


lambda calculus. In Proceedings, Seventh Annual IEEE Symposium on Logic in Computer Sci-

defensible semantics for dynamic Prolog code. In Lassez [Lassez1987].


in block-structured coroutines. ACM Transactions on Programming Languages and Systems,


75


[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.


[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.


[Plainfossé and Shapiro, 1992] David Plainfossé and Marc Shapiro. A distributed GC in an objectsupport operating system. In Cabrera et al. [Cabrera et al.1992].


[Puaut, 1994a] Isabelle Puaut. A distributed garbage collector for active objects. In PARLE’94 [PARLE94], Also INRIA UCRIS-DIFUSION RR 2134.


[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.


104


[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.


[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.


IEEE Press.


[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 others 1991].


[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.


