the Garbage Collection Bibliography
Richard Jones
Computing Laboratory
University of Kent at Canterbury
March 23, 2015

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 http://www.cs.ukc.ac.uk/people/staff/rej/gcbib/gcbib.html

Copyright ©1999, 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].


[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].


[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.


[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].


[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.


[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.


Robert Fitzgerald and David Tarditi. The case for profile-directed selection of garbage collectors. In Chambers and Hosking (Chambers and Hosking 2000).


40


49


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


Richard Jones. The garbage collection page. The definitive on-line resource for garbage collection material.


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].


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.


[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 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.


[Piimarta et al., 1995] Ian Piimarta, 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.


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


[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].


[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.


[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].


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


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.


Richard M. Stallman. Phantom stacks: If you look too hard, they aren’t there. AI Memo 556, MIT AI Laboratory, July 1980.


[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].


123


