the Garbage Collection Bibliography

Richard Jones
R.E.Jones@kent.ac.uk
School of Computing
University of Kent
September 13, 2023

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 ©1999-2023, 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.


[Bacon et al., 2012a] David F. Bacon, Perry Cheng, and Sunil Shukla. And then there were none: a stall-free real-time garbage collector for reconfigurable hardware. In *PLDI 2012* [PLDI 2012], pages 23–34.


[Ben-David et al., 2019] Naama Ben-David, Guy E. Blelloch, Yihan Sun, and Yuhanao Wei. Multiversion concurrency with bounded delay and precise garbage collection. In SPAA 2019 [SPAA 20192019].

[Ben-David et al., 2021] Naama Ben-David, Guy E. Blelloch, Panagiota Fatourou, Eric Ruppert, Yihan Sun, and Yuanhao Wei. Space and time bounded multiversion garbage collection. In 35th International Symposium on Distributed Computing (DISC), Freiburg, Germany, October 2021.


[Calder and Zorn, 2005] Brad Calder and Benjamin G. Zorn, editors. Workshop on Memory System Performance, Chicago, IL, pages 60–73.


[Cameron et al., 2015] Callum Cameron, Jeremy Singer, and David Vengerov. The judgment of Forseti: Economic utility for dynamic heap sizing of multiple runtimes. In Bond and Hosking [Bond and Hosking2015], pages 143–156.


[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].


28


[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, 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.


[Haible, 2005] Bruno Haible. Weak datastructures. This talk at the European Common Lisp meeting o2 24 April 2005 explains the benefits and drawbacks of weak references. It generalizes the data types of weak pointer, weak list and weak hash-table. It explains how to implement these data types correctly and efficiently., April 2005.


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


[Higuera and Issarny, 2005] M. Teresa Higuera and Valerie Issarny. Improving the memory management

[Higuera et al., 2002] Maria Teresa Higuera, Valerie Issarny, Michel Banatre, Gilbert Cabillic, Jean-
Philippe Lesot, and Frederic Parain. Memory management for real-time Java: an efficient solution

[Higuera-Toledano and Issarny, 2001] Maria 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 effi-

[Higuera-Toledano, 2002] Maria Teresa Higuera-Toledano. Solutions à la Gestion Mémoire pour

[Higuera-Toledano, 2006a] M. Teresa Higuera-Toledano. Analyzing the memory management se-
nemic and requirements of the Real-Time specification of Java JSR-0000001. In ISORC 2006
[ISORC 20062006], pages 419–423.

techniques for embedded real-time Java systems. In 12th international Conference on Embedded and

[Higuera-Toledano, 2006c] Maria Teresa Higuera-Toledano. The indeterministic behaviour of scoped
memory in Real-Time Java. In 4th ACS/IEEE International Conference on Computer Systems and

[Higuera-Toledano, 2007a] Maria Teresa Higuera-Toledano. Allowing cycles references among scoped
memory areas in the Real-Time Specification for Java. In ISORC 2007 [ISORC 20072007], pages
110–114.

[Higuera-Toledano, 2007b] Maria Teresa Higuera-Toledano. Name-based write barriers in real-time
Java. In IEEE international Conference on Computer and Information Technology (CIT-07), pages

[Higuera-Toledano, 2008] Maria 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-Toledano, 2011] Maria Teresa Higuera-Toledano. Using transactional memory to synchro-
nize an adaptive garbage collector in real-time Java. In 14th IEEE International Symposium on
Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, pages 152–161,
Newport Beach, CA, March 2011.

[Higuera-Toledano, 2014] Maria Teresa Higuera-Toledano. Building the Java heap with bricks in an em-
bdedded real-time environment. In Proceedings of the 2014 IEEE/ACM 18th International
Symposium on Distributed Simulation and Real Time Applications (DS-RT’14), pages 57–66. IEEE Computer

[Higuera, 2003] Maria Teresa Higuera. Memory management design to the concurrent execution of
RTSJ applications. In JTURES 2003 [JTURES 20032003], pages 479–489.


University of California, Berkeley, November 1987. Also UCB/CSD Technical report 87/381.


64


[Hölzlze and Ungar, 1995] Urs Hölzlze and David Ungar. Do object-oriented languages need special hardware support? In Nierstrasz [Nierstrasz1995].


[Hoseinzadeh and Swanson, 2021] Antony L. Hosking and Ali-Reza Adl-Tabatabai, editors. Workshop on Memory System Performance and Correctness, San Jose, CA, October 2021. ASPLOS 2021 was a virtual event.


[Hosking and Finocchi, 2019] Antony Hosking and Irene Finocchi, editors. 16th ACM SIGPLAN International Conference on Managed Programming Languages and Runtimes, Athens, Greece, October 2019. ACM Press.


[Hussein et al., 2015a] Ahmed Hussein, Antony L. Hosking, Mathias Payer, and Christopher A. Vick. Dont race the memory bus: Taming the GC leadfoot. In Bond and Hosking [Bond and Hosking2015].


[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.


[Kal., Kaleida Labs. ScriptX Architectural Overview.


82


[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 Juil and Juul [Juil and Juul1990].


[Marti et al., 2006b] 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.


Erez Petrank and Doug Lea, editors. 7th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, Newport Beach, CA, March 2011. ACM Press.


[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.


[Pirkelbauer et al., 2017] Peter Pirkelbauer, Amalee Wilson, Hadia Ahmed, and Reed Milewicz. Memory management for concurrent data structures on hardware transactional memory. In TRANSACT 2017 [TRANSACT 20172017].


[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.


[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].


[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 94–133.


123


Reichenbach et al., 2010] C. Reichenbach, Eddie Aftandilian, NI Immerman, Sam Gufer, and Yannis Smaradakis. 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.


[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.


[Shimchenko et al., 2022] Marina Shimchenko, Mihail Popov, and Tobias Wrigstad. Analyzing and predicting energy consumption of garbage collectors in OpenJDK. In Wrigstad and Gonzalez Boix [Wrigstad and Gonzalez Boix2022].


[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.


[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.


151


