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 ©1999-2024, 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.


12


[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.


[Braginsky et al., 2013] Anastasia Braginsky, Alex Kogan, and Erez Petrank. Drop the anchor: Lightweight memory management for non-blocking data structures. In SPAA 2013 [SPAA 20132013], pages 33–42.


23


[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].


[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.


34


[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.


The case for garbage collection in C++. In Jul and Juul [Jul and Juul 1990]. Also University of California, Santa Cruz technical report UCSC-CRL-90-37.


Lisp II garbage collector. AI Memo 19, MIT AI Laboratory, 1960.


[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 02 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 [Hughes 1993].


73


[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.


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 Jul and Juul [Jul 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.


[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.


[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].


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 94–99.


[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.


143


Vetter et al., 2011: Jeffrey Vetter, Madanlal Musuvathi, and Xipeng Shen, editors. Workshop on Memory System Performance and Correctness, San Jose, CA, June 2011.


Wadler, 1984: Philip L. Wadler. Listlessness is better than laziness: Lazy evaluation and garbage collection at compile time. In Steele [Steele1984], pages 45–52.


154


