

Paul E. McKenney
December 29, 2011

Education.

- BS Computer Science and BS Mechanical Engineering, Oregon State University, 1981.
- MS Computer Science, Oregon State University, 1988 (with 14 credit-hours of coursework from Stanford University).
- Ph.D Computer Science and Engineering, OGI School of Science & Engineering at Oregon Health & Science University, 2004. Dissertation entitled “Exploiting Deferred Destruction: An Analysis of Read-Copy-Update Techniques in Operating System Kernels” available at: <http://www.rdrop.com/users/paulmck/RCU/RCUdissertation.2004.07.14e1.pdf>

Books.

1. “Is Parallel Programming Hard, And If So, What Can You Do About It?”, netbook announced January 2, 2011 (<http://kernel.org/pub/linux/kernel/people/paulmck/perfbook/perfbook.html>).

Book Chapters.

2. “A Pattern Language for Parallelizing Existing Programs on Shared-Memory Multiprocessors”, PLoP’95. Published as a chapter of Volume II of Pattern Languages of Program Design (Addison-Wesley, June 1996). The material in this chapter has been incorporated into a number of training programs from diverse companies, both in the area of pattern languages and in the area of parallel programming. It is also cited by two volumes of Doug Schmidt’s Pattern-Oriented Software Architecture series.

Journal Articles.

1. “User-Level Implementations of Read-Copy Update”, with Mathieu Desnoyers, Alan Stern, Michel R. Dagenais, and Jonathan Walpole. February 2012 IEEE Transactions on Parallel and Distributed Systems. Describes what RCU (including semi-formal semantics), how to implement it in userspace, and how it performs.
2. “Why The Grass May Not Be Greener On The Other Side: A Comparison of Locking vs. Transactional Memory”, August 2010 Operating Systems Review, with Maged M. Michael, Josh Triplett, and Jonathan Walpole.
3. “Scalable Concurrent Hash Tables via Relativistic Programming”, August 2010 Operating Systems Review, with Josh Triplett and Jonathan Walpole.
4. “Introducing Technology Into the Linux Kernel: A Case Study”, July 2008 Operating Systems Review (Linux issue), with Jonathan Walpole.
5. “Responsive systems: An introduction”, with Robert F. Berry and Francis N. Parr. April-June 2008 IBM Systems Journal.
6. “The read-copy-update mechanism for supporting real-time applications on shared-memory multiprocessor systems with Linux”, with Dinakar Guniguntala, Josh Triplett, and Jonathan Walpole. April-June 2008 IBM Systems Journal.
7. “Performance of memory reclamation for lockless synchronization”, with Thomas E. Hart, Angela Demke Brown, and Jonathan Walpole. December 2007 Journal of Parallel and Distributed Computing. Journal version of the 2006 IPDPS paper, but with significant revisions.
8. “Enabling Autonomic System Software with Hot-Swapping”, January 2003 IBM Systems Journal (web), with Jonathan Appavoo, Kevin Hui, Craig A. N. Soules, Robert W. Wisniewski, Dilma Da Silva, Orran Krieger, Marc Auslander, David Edelsohn, Ben Gamsa, Gregory R. Ganger, Paul McKenney, Michal Ostrowski, Bryan Rosenburg, Michael Stumm, and Jimi Xenidis. Describes how autonomic system software can be constructed using a hot-swapping approach based on read-copy update. <http://www.research.ibm.com/journal/sj/421/appavoo.pdf>
9. “Experience With an Efficient Parallel Kernel Memory Allocator”, March 2001 Software—Practice and Experience, with Jack Slingwine and Phil Krueger. Revision of 1993 paper with measurements from use and a bit of NUMA analysis.
10. “Differential Profiling”, March 1999 Software—Practice and Experience. Revision of 1995 paper.

11. "Selecting Locking Primitives for Parallel Programs", October 1996 Communications of the Association for Computing Machinery. This is cited by "The Art of Multiprocessor Programming" by Herlihy and Shavit.
12. "Efficient Demultiplexing of Incoming TCP Packets", Spring 1992 Computing Systems, with Ken Dove. Revision of the 1992 paper. It is cited by the textbook "Network Algorithmics" by George Varghese.
13. "Stochastic Fairness Queuing", June 1991 Internetworking Research and Experience. Revision of the 1990 paper.
14. "Physical- and Link-Layer Modeling of Packet-Radio Network Performance", January 1991 IEEE Journal on Selected Areas in Communications, with Peter E. Bausbacher.
15. "An Interleave Principle for Demonstrating Concurrent Programs", IEEE Software, October 1984, with Ted G. Lewis and Keith R. Spitz.

Refereed Conference Papers.

1. "Resizable, Scalable, Concurrent Hash Tables via Relativistic Programming", USENIX Annual Technical Conference, with Josh Triplett and Jonathan Walpole, June 2011.
2. "'Real Time' vs. 'Real Fast': How to Choose?", OLS, July 2008.
3. "Integrating IBM's Real-Time Java Implementation with the -rt Linux Patch set", LinuxConf Europe, September 2007, with John Kacur, Josh Triplett, Sripathi Kodi, Dinakar Guniguntala, and Tim Chavez.
4. "Extending RCU for Realtime and Embedded Workloads", with Dipankar Sarma, Ingo Molnar, and Suparna Bhattacharya. OLS, July 2006. Describes ongoing work on adapting RCU for realtime workloads.
5. "Making Lockless Synchronization Fast: Performance Implications of Memory Reclamation", with Thomas E. Hart and Angela Demke Brown. IPDPS 2006 best-paper award. Compares performance of RCU, EBR, HPBR, etc.
6. "Towards Hard Realtime Response from the Linux Kernel on SMP Hardware", with Dipankar Sarma. linux.conf.au, April 2005. Overviews realtime approaches with Linux, also discusses the role of RCU.
7. "Issues with Selected Scalability Features of the 2.6 Kernel", OLS, July 2004, with Dipankar Sarma. Describes scalability, denial-of-service, and realtime issues with the Linux kernel at that time.
8. "Making RCU Safe for Deep Sub-Millisecond Response Realtime Applications", with Dipankar Sarma. USENIX (UseLinux track), June 2004.
9. "RCU vs. Locking Performance on Different CPUs", 2004 linux.conf.au. Shows that some simple rules of thumb apply across a wide range of CPUs.
10. "Storage-Related Requirements for the 2.7 Linux Kernel". Invited presentation in Linux Kernel Summit, July 2003.
11. "Using Read-Copy Update Techniques for System V IPC in the Linux 2.5 Kernel", 2003 FREENIX, with Andrea Arcangeli, Mingming Cao, and Dipankar Sarma. Shows order-of-magnitude improvement in performance obtained by applying RCU to Linux's System V IPC. Also has performance results for RCU infrastructure, which determined which implementation made it into the kernel.
12. "Fairlocks—a High-Performance Fair Locking Scheme", 2002 Parallel and Distributed Computing and Systems, with Swaminathan Sivasubramanian, Jack F. Vogel, and John Stultz. Presents a simple NUMA-aware locking scheme that avoids lock starvation.
13. "Read-Copy Update", 2002 Ottawa Linux Symposium, with Dipankar Sarma, Andrea Arcangeli, Andi Kleen, Orran Krieger, and Rusty Russell. Compares and contrasts a number of Linux read-copy-update implementations developed since the 2001 paper, one of which was accepted into the Linux 2.5 kernel in October 2002.
14. "Read-Copy Update", July 2001 Ottawa Linux Symposium, with Jonathan Appavoo, Andi Kleen, Orran Krieger, Rusty Russell, Dipankar Sarma, and Maneesh Soni. Described Linux implementations of the read-copy update techniques described in the PDCS'98 paper.
15. "Practical Performance Estimation on Shared-Memory Multiprocessors", November 1999 PDCS. Method of quickly estimating performance with reasonable accuracy. Variants of this can be used during design, before either code, compilers, or hardware is available.
16. "Read-Copy Update: Using Execution History to Solve Concurrency Problems", October 1998 PDCS, with Jack Slingwine. Novel set of synchronization primitives that can result in near-zero synchronization overhead in restricted, but commonly occurring, situations. In more specialized cases, this set of primitives can result in *negative* overheads. Read-copy update was used in production in Sequent's DYNIX/ptx operating system in 1993, and was accepted into the Linux 2.5 kernel in October of 2002.

17. "Differential Profiling", MASCOTS'95. Describes technique for locating scaling problems that are otherwise difficult to locate. Used habitually by some prominent Linux-community members.
18. "Efficient Kernel Memory Allocation on Shared-Memory Multiprocessors", Winter'93 USENIX, with Jack Slingwine. This paper demonstrated a memory allocator that could run faster on shared-memory parallel machines than previous allocators could on single-CPU machines, and, at the same time, scale linearly, giving multiple order-of-magnitude speedups on large SMP machines. This paper has been studied in a number of advanced college courses, and is referenced by in "UNIX Internals: The New Frontiers" by Uresh Vahalia.
19. "Efficient Demultiplexing of Incoming TCP Packets", Spring 1992 SIGCOMM'92, with Ken Dove. This paper influenced the design of IPv6, and is referenced by volume 2 of "TCP/IP Illustrated" by Wright and Stevens, by "Gigabit Networking" by Craig Partridge, and by "Network Algorithmics" by George Varghese.
20. "Efficient Buffer Allocation on Shared-Memory Multiprocessors", IEEE HPCS'92, with Gary Graunke. The algorithm described in this paper formed the basis for the memory-allocation algorithm described in the 1993 USENIX paper.
21. "Stochastic Fairness Queuing", INFOCOM'90. This paper has been referenced by many computer-communications-network-traffic-control papers, and is still referenced by new work in this area. It is also referenced by the textbooks "Gigabit Networking" by Craig Partridge and "Network Algorithmics" by George Varghese. An implementation of stochastic fairness queuing was added to Linux in 1997 by Alexey Kuznetsov.
22. "Packet Recovery in High-Speed Networks Using Coding and Buffer Management", INFOCOM'90, with Nachum Shacham. This was the first paper that contained a realistic analysis of the costs and benefits of forward error correction (FEC) for recovering from congestion-induced packet losses. It is frequently referenced by ongoing work in this area, and is referenced by "Gigabit Networking" by Craig Partridge. As of 2007, it was listed in the top 10,000 most-frequently cited articles according to <http://citeseer.ist.psu.edu/allarticles.html>.
23. "Physical- and Link-Layer Modeling of Packet-Radio Network Performance", MILCOM'90, with Peter E. Bausbacher.
24. "Packet Radio Network Research, Development, and Application", MILCOM'89, with David A. Beyer, Michael S. Frankel, John M. Hight, Diane S. Lee, Mark G. Lewis, Jacques Naar, Richard G. Ogier, Nachum Shacham, and William T. Zaumen.
25. "High-Speed Event Counting and Classification Using a Dictionary Hash Technique", ICPP'89. This paper is referenced for its introduction of perturbable hash functions.

Workshop Presentations (and non-paper conference presentations)

1. "On migrate_disable() and latencies", October 21, 2011 to the 13th Real Time Linux Workshop in Prague, Czech Republic.
2. "Validating Core Parallel Software", October 16, 2011 to the China Linux Kernel Developer Conference in Nanjing <http://blog.csdn.net/lengyuex/article/details/6874502>.
3. "Confessions of a Recovering Proprietary Programmer", August 1, 2011 to Linaro Connect in Cambourne UK.
4. "Is Parallel Programming Hard, And If So, What Can You Do About It?" April 28, 2011 to the Android System Developers Forum, Taipei, Taiwan. http://www.digitimes.com.tw/seminar/android_1000428.htm
5. "Whacking Droids: How to Extract Requirements from Flame Wars", Beaver Barcamp 7 (Oregon State University), April 2011.
6. "Verifying Parallel Software: Can Theory Meet Practice?", Verification of Concurrent Data Structures (Verico), January 2011.
7. "Whacking Droids: How to Extract Requirements from Flame Wars", linux.conf.au, January 2011.
8. "Is Parallel Programming Hard, And, If So, Why?", linux.conf.au Multicore miniconference, January 2011.
9. "Verifying Parallel Software: Can Theory Meet Practice?", linux.conf.au Multicore miniconference, January 2011.
10. "Multi-Core Memory Models and Concurrency Theory: A View from the Linux Community", Dagstuhl Multi-Core Memory Models and Concurrency Theory Workshop, January 2011.
11. "Extreme Energy Efficiency", October 16, 2010 to Barcamp at Oregon State University.
12. "When Do Real Time Systems Need Multiple CPUs?", 12th Real-Time Linux Workshop, October 2010.
13. "Simplicity Through Optimization", linux.conf.au, January 2010.

14. “Real Time vs. Real Fast: How to Choose?”, the Eleventh Real-Time Linux Workshop, Dresden, Germany, September 2009.
15. “Deterministic Synchronization in Multicore Systems: the Role of RCU”, the Eleventh Real-Time Linux Workshop, Dresden, Germany, September 2009.
16. “Using a Malicious User-Level RCU to Torture RCU-Based Algorithms”, linux.conf.au, January 2009.
17. “Is Parallel Programming Hard, And If So, Why?”, linux.conf.au, January 2009.
18. “‘Real Time’ vs. ‘Real Fast’: How to Choose?”, MontaVista Vision 2008, October 2008.
19. “Concurrency and Race Conditions”, Linux Plumbers Conference Student Day, September 2008.
20. “Introducing Technology Into Linux”, 2008 Linux Developer Symposium – China, February 2008.
21. “After 25 Years, C/C++ Understands Concurrency”, linux.conf.au, February 2008.
22. “Why The Grass May Not Be Greener On The Other Side: A Comparison of Locking vs. Transactional Memory”, PLOS 2007 with Maged Michael and Jon Walpole.
23. “Real Time Linux Technology: a Deeper Dive”, presentation at MontaVista VISION 2007, October 2007
24. “Real Time Linux Technology”, keynote at MontaVista VISION 2007, October 2007
25. “Linux Realtime Response: Challenges in Making Linux Ready for Real Time Computing”. Presentation at the First Workshop on Real Time, Interactive and Digital Media Supercomputing (RIDMS-1), part of the 12th International Symposium on High-Performance Computer Architecture, February 2006.
26. “Steamroller Testing”. Presentation at the 2006 linux.conf.au conference in Dunedin, New Zealand. January 2006.
27. “Realtime and Linux”, with Ingo Molnar. Invited presentation in Linux Kernel Summit, July 2005. Compares and contrasts a number of realtime approaches put forward for Linux.
28. “Linux Kernel Scalability: Using the Right Tool for the Job”. linux.conf.au, April 2005.
29. “Linux Kernel Scalability: Using the Right Tool for the Job”. Ottawa Linux Symposium, July 2004.

Unrefereed Journal Articles

1. “Validating Memory Barriers and Atomic Instructions”, Linux Weekly News, December 2011.
2. “3.0 and RCU: what went wrong”, Linux Weekly News, July 2011.
3. “ARM kernel consolidation”, Linux Weekly News, May 2011.
4. “Concurrent code and expensive instructions”, Linux Weekly News, January 2011.
5. “The RCU API, 2010 Edition”, December 2010 Linux Weekly News.
6. “Memory-Order Rationale”, ISO SC22 WG14 (C Language), October 2010, with Blaine Garst.
7. “Omnibus Memory Model and Atomics Paper”, ISO SC22 WG21 (C++ Language), with Mark Batty, Clark Nelson, Hans Boehm, Anthony Williams, Scott Owens, Susmit Sarkar, Peter Sewell, Tjark Weber, Michael Wong, and Lawrence Crowl, August 2010, updated November 2010.
8. “Additional Atomics Errata”, ISO SC22 WG14 (C Language), May 2010.
9. “Rationale for C-Language Dependency Ordering”, ISO SC22 WG14 (C Language), May 2010.
10. “Updates to C++ Memory Model Based on Formalization”, ISO SC22 WG14 (C Language), April 2010, with Mark Batty, Clark Nelson, N.M. McLaren, Hans Boehm, Anthony Williams, Peter Dimov, and Lawrence Crowl, updated May 2010.
11. “Explicit Initializers for Atomics”, ISO SC22 WG14 (C Language), April 2010, with Mark Batty, Clark Nelson, N.M. McLaren, Hans Boehm, Anthony Williams, Peter Dimov, and Lawrence Crowl. Updated May 2010.
12. “Dependency Ordering for the C Memory Model”, ISO SC22 WG14 (C Language), April 2010, with Clark Nelson, Hans-J. Boehm, and Lawrence Crowl.
13. “Explicit Initializers for Atomics”, ISO SC22 WG21 (C++ Language), March 2010, with Mark Batty, Clark Nelson, N.M. McLaren, Hans Boehm, Anthony Williams, Peter Dimov, and Lawrence Crowl.
14. “Updates to C++ Memory Model Based on Formalization”, ISO SC22 WG21 (C++ Language), February 2010, with Mark Batty, Clark Nelson, N.M. McLaren, Hans Boehm, Anthony Williams, Peter Dimov, and Lawrence Crowl, updated March 2010.
15. “Lockdep-RCU”, Linux Weekly News, February 2010.
16. “RCU: the Bloatwatch Edition”, Linux Weekly News, March 2009.
17. “Hierarchical RCU”, Linux Weekly News, November 2008.
18. “Example POWER Implementation for C/C++ Memory Model”, ISO SC22 WG21 (C++ Language), Evolution Working Group, August 2008, with Raul Silvera. Updated September 2008.
19. “Integrating and Validating dynticks and Preemptable RCU”, Linux Weekly News, April 2008.

20. "A New Interface for C++ std::duration Type", ISO SC22 WG21 (C++ Language), Evolution Working Group, April 2008, with Michael Wong.
21. "RCU part 3: the RCU API", Linux Weekly News, January 2008.
22. "What is RCU? Part 2: Usage", Linux Weekly News, January 2008.
23. "What is RCU, Fundamentally?", Linux Weekly News, December 2007.
24. "The design of preemptible read-copy update", Linux Weekly News, October 2007.
25. "Using Promela and Spin to verify parallel algorithms", Linux Weekly News, August 2007. Includes proof of correctness for QRCU.
26. "Converting Memory Fences to N2324 Form", ISO SC22 WG21 (C++ Language), Evolution Working Group, August 2007.
27. "C++ Data-Dependency Ordering", ISO SC22 WG21 (C++ Language), Evolution Working Group, April 2007. Updated August 2007, February 2008, May 2008, June 2008, and September 2008. Portions voted into Working Draft on June 2008 and the remainder voted in on September 2008.
28. "Priority-Boosting RCU Read-Side Critical Sections", Linux Weekly News, February 2007.
29. "Overview of Linux-Kernel Reference Counting", ISO SC22 WG21 (C++ Language), Evolution Working Group, January 2007.
30. "A simple and efficient memory model for weakly ordered architectures", Programming Language C++, Evolution Working Group, January 2007 with Raul Silvera, Michael Wong, and Bob Blainey. Makes case for weakly ordered primitives in programming languages. Updated May 2007.
31. "SMP and Embedded Realtime", Linux Journal, January 2007.
32. "RCU and Unloadable Modules", Linux Weekly News, January 2007.
33. "Sleepable RCU", Linux Weekly News, October 2006.
34. "Memory Ordering in Modern Microprocessors, Part II", Linux Journal, September 2005.
35. "A Realtime Preemption Overview", Linux Weekly News, August 10, 2005.
36. "Memory Ordering in Modern Microprocessors, Part I", Linux Journal, August 2005.
37. "Attempted Summary of "RT patch acceptance" Thread", Linux Weekly News, June 7, 2005.
38. "[RFC] RCU and CONFIG_PREEMPT_RT progress", Linux Weekly News, May 9, 2005.
39. "Storage Improvements for 2.6 and 2.7", Linux Journal, August 2004.
40. "Scaling dcache with RCU", Linux Journal, January 2004, with Dipankar Sarma and Maneesh Soni. Shows Linux 2.6 RCU dcache modifications.
41. "Using RCU in the Linux 2.6 Kernel", Linux Journal, October 2003. Shows empirical data comparing RCU with other locking mechanisms.
42. "Benchmark Standards/Metrics and Capacity Planning", UniForum, 1994.
43. "Congestion Avoidance", Connexions, October 1990.
44. "Broadcast Storms, Nervous Hosts, and Load Imbalances", ;login: The USENIX Association Newsletter, Sept/Oct 1988.
45. "Charge Number Accounting Without Kernel Modifications", ;login: The USENIX Association Newsletter, July/Aug 1988.
46. "User Defined Files", Fall 1981 Operating Systems Review, with Kirk A. Bailey, Lee Boynton, Gary J. Oliver, and David Regan.

Courses and Guest Lectures

1. "Validating Core Parallel Software", October 28, 2011 to TU Dresden.
2. "Confessions of a Recovering Proprietary Programmer", October 18, 2011 to Nanjing University.
3. "Confessions of a Recovering Proprietary Programmer", October 17, 2011 to Shanghai Jiaotong University.
4. "Synchronization and Scalability in the Macho Multicore Era" July 27, 2010 to the RETIS group at Scuola Superiore Sant'Anna, Pisa, Italy. <http://retis.sssup.it/?q=node/89>
5. "Is Parallel Programming Hard, And If So, Why?" November 12, 2009 to Jay Bockelman's Oregon Institute of Technology Operating Systems Class. (IsParallelProgrammingHard.OIT.2009.11.12a.odp)
6. "Introduction to Performance, Scalability, and Real-Time Issues on Modern Multicore Hardware: Is Parallel Programming Hard, And If So, Why?" July 13-17, 2009 to the ACACES 2009 Fifth International Summer School on Advanced Computer Architecture and Compilation for Embedded Systems.
7. "Is Parallel Programming Hard, And If So, Why?" May 27, 2009 presentation to Jay Bockelman's Oregon Institute of Technology Computer Organization class. (IsParallelProgrammingHard.OIT.2009.05.27b.odp)

8. "Confessions of a Recovering Proprietary Programmer." May 21, 2009 presentation to Tim Budd's Oregon State University open-source-software class. (Confessions.2009.05.18a.odp)
9. "Is Parallel Programming Hard, And, If So, Why?" December 4, 2008, presentation to Bart Massey's Portland State University Multicore Computing Practicum class.
10. "Is Parallel Programming Hard, And, If So, Why?", November 24, 2008, presentation to the Oregon Institute of Technology Operating Systems class.
11. "Why The Grass May Not Be Greener On The Other Side: A Comparison of Locking vs. Transactional Memory", October 31, 2008, presentation to the University of North Carolina Chapel Hill Computer Science Department.
12. "RCU in the Linux Kernel", October 24, 2008, presentation to the University of Rochester Computer Science Department.
13. "Towards a User-Level RCU Implementation", October 10, 2008, presentation to Emmett Wittchell's group at University of Texas at Austin.
14. "Why The Grass May Not Be Greener On The Other Side: A Comparison of Locking vs. Transactional Memory", March 10, 2008, presentation to PSU computer science students (CSE510).
15. "Why The Grass May Not Be Greener On The Other Side: A Comparison of Locking vs. Transactional Memory", January 23, 2008, presentation to the Department of Computer Science and Automation, India Institute of Science, Bangalore, India.
16. "Why The Grass May Not Be Greener On The Other Side: A Comparison of Locking vs. Transactional Memory: Five Realtime Myths Exposed", October 29, 2007. Presentation to OIT computer science students.
17. "Linux Realtime Response: Five Realtime Myths Exposed", October 9, 2007. Presentation to UCSC computer science students.
18. "A Day in My Life and My Road to IT", Computing and Technology Trends, May 12, 2007, University of Portland. Lecture to students considering IT on how to deal with globalization and other dangerous opportunities.
19. "A Day in the Life / My Road to IT", Future Potential in Computing, February 10, 2007, Willamette University. Lecture to students considering IT on how to deal with globalization and other dangerous opportunities.
20. "Linux Realtime Response: Five Realtime Myths Exposed", IEEE Computer Society, Oregon Chapter, January 2007. Debunks five realtime myths.
21. "Linux Realtime Response: The Next Frontier in Commercial Computing", Oregon Institute of Technology, November 20, 2006. Revision of the October 2005 AoT presentation.
22. "Extending RCU for Realtime and Embedded Workloads", University of Toronto Cider Seminar, July 18, 2006. Preview of OLS paper.
23. "Linux Realtime Response: The CONFIG_PREEMPT Patch Set", Oregon Institute of Technology, October 6, 2005.
24. "Linux Realtime Response", IBM Academy of Technology Topical Conference on Real-Time and Safety-Critical Systems, October 24-25, 2005.
25. "Abstraction, Reality Checks, and RCU", Oregon State University, October 6, 2005.
26. "Linux Realtime Response: The CONFIG_PREEMPT Patch Set", Oregon State University, October 6, 2005.
27. "Abstraction, Reality Checks, and RCU", University of Toronto Cider Seminar, July 26, 2005.
28. "Exploiting Deferred Destruction: An Analysis of Read-Copy-Update Techniques in Operating System Kernels", Operating System class at Portland State University, January 13, 2005.
29. "Linux Kernel Scalability: Using the Right Tool for the Job", Operating System class at Oregon Institute of Technology, Capitol Center campus. November 8, 2004.
30. "Read Copy Update", Operating System class at Oregon Institute of Technology, Capitol Center campus. November 24, 2003.
31. "SMP Programming", Technical training class at Veritas Inc, Mountain View, CA. November 4, 1997.

White Papers and Technical Reports

1. "Is Parallel Programming Hard, And If So, Why?", Portland State University TR-09-02, February 2009.
2. "Experiment Design for CATE", SRI Technical Note ITAD-8600-TR-93-74, SRI International, March 1993, with B. Denny.
3. "RCU Semantics: A First Attempt", <http://www.rdrop.com/users/paulmck/RCU/rcu-semantics.2005.01.30a.pdf>, with Jonathan Walpole, January 2005.

4. "Traffic Generator Software Release Notes", SRI Technical Note ITAD-8600-TN-93-28, SRI International, February 1993, with D. Lee and B. Denny.
5. "Position Estimation", Oregon State University technical report, 1988. OSU Master's Project. Described method of calibrating acoustic-navigation transponder arrays. (Prior to GPS, this was the preferred method of accurately determining the position of surface ships used both to conduct SONAR searches for salvage purposes and to map gravitational and geomagnetic data for mineral-exploration purposes. Still the method of choice for submersibles below the depths penetrated by GPS signals.)

Interviews and Panels

1. "Interview with Paul McKenney, author of 'Is Parallel Programming Hard, And, If So, What Can You Do About It?'" Sean McManus at <http://softtalkblog.com/2011/11/18/interview-with-paul-mckenney-author-of-is-parallel-programming-hard-and-if-so-what-can-you-do-about-it/>, November 2011.
2. "Linux Kernel Developer Panel", LinuxCon Europe, Prague, Czech Republic, October 26, 2011 <https://events.linuxfoundation.org/events/linuxcon-europe/kernel-panel>.
3. "Why Is Parallel Programming So Hard", Rich Seidner at <http://intelligencesoftware.com/feature/expert-insight/why-is-parallel-programming-so-hard/>, August 2011.
4. "Teach Parallel" presentation at Supercomputing 2009 hosted by Paul Steinberg (Intel) and Tom Murphy (Contra Costa College). <http://isntv.blip.tv/file/2977764/>, November 2009.
5. "Shrinking slices: Looking at realtime for Linux, PowerPC, and Cell". McLaren Harris in DeveloperWorks, August 15, 2005. <http://www.psu.com/forums/showthread.php/4405-An-interview-with-Paul-E-McKenney-RTOS-powerPC-and-Cell>

Letters to the Editor

1. "To Learn Software Engineering, Study Application Logic", Communications of the ACM, 52(5) May 2009.
2. "Hope for Educational Sea Change", Communications of the ACM, 44(6) June 2001.
3. "Mastering the Basics the Highest Priority", Communications of the ACM, 43(3) March 2000.
4. "Renaissance Reflections", Scientific American, 280(5) May 1999.

Reviews

1. IPSI BgD Transactions on Internet Research, January 2006. (<http://internetjournals.net/journals/tar/2006/TARVol2Num1.pdf>)
2. IPSI BgD Transactions on Advanced Research, January 2006 (<http://internetjournals.net/journals/tar/2006/TARVol2Num1.pdf>)

Patents

1. #8,055,918, "Optimizing Preemptible Read-Copy Update for Low-Power Usage by Avoiding Unnecessary Wakeups", November 2011.
2. #8,020,160, "User-Level Read-Copy Update That Does Not Require Disabling Preemption or Signal Handling", September 2011.
3. #7,971,243, "Security Module", June 2011.
4. #7,935,778, "Efficient Support of Consistent Cyclic Search with Read-Copy Update and Parallel Updates", May 2011.
5. #7,953,708, "Optimizing Grace Period Detection for Preemptible Read-Copy Update on Uniprocessor Systems", May 2011.
6. #7,934,062, "Read/Write Lock With Reduced Reader Lock Sampling Overhead in Absence of Writer Lock Acquisition", April 2011.
7. #7,904,438, "Updating First Data Value Before Second Data Value", March 2011.
8. #7,904,436, "Realtime-Safe Read Copy Update With Lock-Free Readers", March 2011.
9. #7,873,612, "Atomically Moving List Elements Between Lists Using Read-Copy Update", January 2011.
10. #7,844,802, "Instructions for Ordering Execution in Pipelined Processes", November 2010.
11. #7,836,034, "Method and Apparatus for Updating First Data Value Before Second Data Value", November 2010.

12. #7,831,796, "Apparatus, System, and Method for Dynamically Allocating Main Memory Among a Plurality of Applications", November 2010.
13. #7,818,306, "Read-Copy-Update (RCU) Operations with Reduced Memory Barrier Usage", October 2010. The CONFIG_PREEMPT_RCU version of Linux-kernel real-time RCU.
14. #7,814,082, "Efficient Support of Consistent Cyclic Search with Read-Copy-Update", October 2010. Generation numbers to allow RCU readers to see consistent snapshots of the RCU-protected data structure.
15. #7,809,896, "Efficient Sharing of Memory Between Applications Running Under Different Operating Systems on a Shared Hardware System", October 2010.
16. #7,748,003, "Hard Real-Time Response", June 2010. Migrate tasks among CPUs to isolate real-time tasks from non-real-time behavior.
17. #7,747,805, "Adaptive Reader-Writer Lock", June 2010. Automatically switch between exclusive locking, reader-writer locking, and RCU based on measured behavior.
18. #7,734,881, "Adapting RCU for Real-Time Operating System Usage", June 2010. Immediately complete grace period on uniprocessors when it is safe to do so.
19. #7,734,879, "Efficiently Boosting Priority of Read-Copy Update Readers in a Real-Time Data Processing System", June 2010. Use modular arithmetic and per-CPU pairs of counters to implement preemptible RCU.
20. #7,734,833, "Method for Scheduling Operations Called by a Task on a Real-Time or Non-Real-Time Processor", June 2010. Hard real-time execution by reserving non-real-time CPUs.
21. #7,721,291, "Apparatus, System, and Method for Automatically Minimizing Real-Time Task Latency and Maximizing Non-Real Time Task Throughput", May 2010. Conditionally execute preemption points depending on real-time needs.
22. #7,653,791, "Realtime-Safe Read Copy Update With Per-Processor Read/Write Locks", January 2010.
23. #7,587,615, "Utilizing Hardware Transactional Approach to Execute Code After Initially Utilizing Software Locking by Employing Pseudo-Transactions", September 2009. Software variant of transactional lock elision.
24. #7,500,036, "Quad Aware Locking Primitive", March 2009. Use a bit vector to organize round-robin lock handoff. Also issued as #I238944.
25. #7,487,320, "Apparatus and System for Dynamically Allocating Main Memory Among a Plurality of Applications", February 2009. The Adaptive Replacement Cache (ARC) enhanced to allow the size of the cache to grow and shrink.
26. #7,478,204, "Efficient Sharing of Memory Between Applications Running Under Different Operating Systems on a Shared Hardware System", January 2009. Safe mechanisms to allow data sharing across OSes, even where one is real-time and the other non-real-time.
27. #7,472,233, "Memory Allocator for a Multiprocessor Computer System", December 2008. NUMA-aware parallel memory allocation.
28. #7,472,228, "Read-Copy Update Method", December 2008. Deferring the start of grace periods to increase the number of updates that the overhead of a given grace period may be amortized over. Also issued as #ZL200510108687.1.
29. #7,454,581, "RCU Grace-Period Detection Without Atomic Instructions that Gracefully Handles Large Numbers of CPUs", November 2008. Modified token-passing approach.
30. #7,447,868, "Using Vector Processors to Accelerate Cache Lookups", November 2008.
31. #ZL200510108687.1, "Read-Copy Update System and Method", November 2008. Controlled batching to amortize grace-period-detection overhead over larger numbers of update requests. Also issued as US Patent #7,472,228, December 2008.
32. #7,430,627 "Adaptive Reader-Writer Lock", September 2008.
33. #7,426,511 "Efficient Support of Consistent Cyclic Search With Read-Copy Update", September 2008.
34. #7,398,376 "Instructions for Ordering Execution in Pipelined Processes", July 2008.
35. #7,395,383 "Realtime-Safe Read-Copy Update With Per-Processor Read/Write Locks", July 2008.
36. #7,395,263 "Realtime-Safe Read-Copy Update With Lock-Free Readers", July 2008.
37. #7,392,376 "Security Module", June 2008, with Orran Y. Krieger and Boas Betzler.
38. #7,353,346 "Read-Copy-Update (RCU) Operations With Reduced Memory Barrier Usage", April 2008, with Suparna Bhattacharya.
39. #7,349,926 "Atomic Renaming and Moving of Data Files While Permitting Lock-Free Look-Ups", March 2008, with Dipankar Sarma and Maneesh Soni.
40. #ZL200510115128.3 (China) "Cluster-Wide Read-Copy Update System and Method", January 2008 with Julian Satran.

41. #7,287,135 “Adapting RCU for Real-Time Operating System Usage”, October 2007, with Dipankar Sarma.
42. #7,191,272 “Adaptive Reader-Writer Lock”, March 2007.
43. #3880389 (Japan) “Nestable Reader-Writer Locking for Multiprocessor Systems”, November 2006.
44. #7,030,734 “Verification of Proper Coupling of Power Line Communication Units to Power Circuits”, April 2006, with Craig Butler, Akimasa Fleshler, and Stuart Friedberg.
45. #6,996,812 “Software Implementation of Synchronous Memory Barriers”, February 2006. Work around CPU architectures that do not provide a synchronous memory-barrier instruction.
46. #I238944 (Taiwan) “Quad-Aware Locking Primitive (krlock)”, September 2005, with R. L. Rockhold and R. Rajamony. Can be thought of as a tree-shaped queued lock. Also issued as US Patent #7,500,036, March 2009.
47. #6,886,162 “High Speed Methods for Maintaining a Summary of Thread Activity for Multiprocessor Computer Systems”, April 2005. Tweak to RCU to avoid N^2 overhead, where N is the number of CPUs.
48. #6,842,809 “Apparatus, Method and Computer Program Product for Converting Simple Locks in a Multiprocessor System”, January 2005, with Luke Browning, Tom Mathews, and James Moody. Mechanism to automatically switch between simple spinlock and NUMA-aware lock on NUMA systems. (Referenced by 1 patent.)
49. #6,823,511 “Reader-Writer Lock for Multiprocessor Systems”, November 2004, with Brent Kingsbury. Light-weight reader-writer semaphore, optimized for consecutive write requests and for SMP/NUMA systems. (Referenced by 1 patent.)
50. #6,785,888 “Memory Allocator for a Multiprocessor Computer System”, August 2004, with Phillip E. Krueger, Stuart A. Friedberg, and Brent A. Kingsbury. SMP/NUMA-friendly counter with range limits.
51. #6,779,090 “Spinlock for Shared Memory”, August 2004, with William L. Irwin III, Swaminathan Sivasubramanian, and John Stultz. Simplified NUMA-semi-aware spinlock. (Referenced by 1 patent.)
52. #6,678,772 “Adaptive Reader-Writer Lock”, January 2004. Dynamically switches between different locking primitives as needed to optimize performance. (Referenced by 1 patent.)
53. #6,668,310 “High Speed Counters”, December 2003. Combine “target” and “value” fields from distributed reference counter in order to manipulate them with a single atomic instruction.
54. #6,629,152 “Message Passing Using Shared Memory of a Computer”, September 2003, with Brent Kingsbury and Jean-Marie Sulmont. Shared-memory message-passing using atomic operations, optimized for NUMA machines. (Referenced 1 patent.)
55. #6,615,316 “Using Hardware Counters to Estimate Cache Warmth for Process/Thread Schedulers”, September 2003, with Phil Krueger. Markov models to estimate cache warmth given the performance-counter data available on commodity microprocessors.
56. #6,505,286 “User Specifiable Allocation of Memory for Processes in a Multiprocessor Computer having a Non-Uniform Memory Architecture”, January 2003, with Brent Kingsbury, Corene Casper, and Phil Krueger.
57. #6,480,918 “Lingering Locks with Fairness Control for Multi-Node Computer Systems”, November 2002, with Kevin Closson and Raghu Malige. (Referenced by 26 patents.)
58. #6,473,819 “Scalable Interruptible Queue Locks for Shared-Memory Multiprocessors”, October 2002, with Benedict Joseph Jackson, Ramakrishnan Rajamony, and Ronald Lynn Rockhold. (Referenced by 3 patents.)
59. #6,418,517 “Optimized Function Execution for a Multiprocessor Computer System”, July 2002, with Chandrasekhar Pulamarasetti. (Referenced by 4 patents.)
60. #6,247,041 “Multiprocessor Computer System with User Specifiable Process Placement”, June 2001, with Brent Kingsbury et al. (Referenced by 1 patent.)
61. #6,219,690 “Apparatus and method for achieving reduced overhead mutual exclusion and maintaining coherency in a multiprocessor system utilizing execution history and thread monitoring”, April 2001, with Jack Slingwine. “Change in mode” aspect of RCU.
62. #6,205,528 “User specifiable allocation of memory for processes in a multiprocessor computer having a non-uniform memory architecture”, March 2001, with Brent Kingsbury, Corene Casper, and Phil Krueger. (Referenced by 14 patents.)
63. #5,727,209 “Apparatus and Method for Achieving Reduced Overhead Mutual Exclusion and Maintaining Coherency in a Multiprocessor System Utilizing Execution History and Thread Monitoring”, March 1998, with Jack Slingwine. “Read-copy-update” aspect of RCU.

64. #5,608,893 “Method for Maintaining Data Coherency Using Thread Activity Summaries in a Multicomputer System”, March 1997, with Jack Slingwine. “Synchronize data in different computers” aspect of RCU. (Referenced by 2 patents.)
65. #5,442,758 “System and Method for Achieving Reduced Overhead Mutual Exclusion and Maintaining Coherency in a Multiprocessor System Utilizing Execution History and Thread Monitoring”, August 1995, with Jack Slingwine. “Summary of execution history and thread monitoring infrastructure” aspect of RCU. (Referenced by 77 patents.)

More than 20 additional patent applications pending.

Professional Activities.

- ISO SC22/WG14 standards committee (C) as concurrency expert (2009-2011).
- Attended invitation-only Linux Networking Summit (netconf), 2009.
- Program Chair for 2009 and 2011 Linux Plumbers Conference.
- Maintainer for RCU and the RCU torture tests in the Linux kernel (2009-), reporting to Ingo Molnar.
- Attended invitation-only Linux Real-Time Summit, 2008, 2009, and 2011.
- ISO SC22/WG21 standards committee (C++) as concurrency expert (2007-2011).
- Member, Oregon State University Industrial Advisory Board (EECS Department).
- Attended invitation-only Linux Kernel Summit, 2003, 2004, 2005, 2010, and 2011. Member of program committee 2010 and 2011.
- Elected to IBM Academy of Technology, 2002
- Received Toastmasters Advanced Toastmaster (ATM) certification, 1997.
- Finance Chair of ETACOM conference, 1995-6.
- Chair of Oregon Chapter of IEEE Computer Society, 6/92-6/95. Helped to revitalize this chapter after an extended period of inactivity.
- “Careers in Engineering and Science” presentation at ASE Midsummer Conference for Saturday Academy students, 1995-1998.
- Secretary of Oregon Section of IEEE Computer Society, 1995.
- Treasurer of Oregon Section of IEEE Computer Society, 1994.
- Received Toastmasters Competent Toastmaster (CTM), 1994.
- Charter member of Sequent Toastmasters, 1993.
- Member End-to-End Task Force in Internet Activities Board 1988-1990.
- Member of ANSI X3J16 (C++ standards committee) in 1990
- President of Pyramid User’s Group, 1987.
- Member of May 1983 Computer Science delegation to People’s Republic of China. Sponsored by the People-to-People Citizen Ambassador Program. Delegation led by Dr. Gio Weiderhold of Stanford University.
- Member of ACM and IEEE. Past member of IETF and SAE (Society of Automotive Engineers).

Work Experience.

- April 2000 – present: IBM Distinguished Engineer and maintainer of RCU in the Linux kernel. Representative projects: Linux Technology Center Embedded Architect. IBM’s Linaro Technical Steering Committee representative. Improve RCU’s energy efficiency. Realtime RCU implementation for Linux kernel. Chair, Linux Technology Center Technical Board (2004). Linux Technology Center System Architect covering xSeries Linux and open-source issues. Storage Software Architect (Linux, performance, intellectual-property, and testing issues). Lead team constructing prototype embedded hypervisor for PPC CPUs. Reviewed software aspects of Sony Consumer Entertainment’s Broadband-Engine architecture (later part of Sony PS3). LTC Liaison to IBM Research’s K42 project (three joint papers, one joint project). Technical lead/architect for effort to implement and exploit read-copy update in Linux. Formed industry consortium for Linux on NUMA machines. LTC System architect in Linux Technology Center. Member of eBusiness Server Development Council and Advanced eBusiness Council. Software architect for high-end IA64 systems. NUMA-aware locking in AIX.
- Sept. 1990 – March 2000: Sequent Computer Systems, Inc. Representative projects: Light-weight reader-writer semaphore. Increase per-CPU hardclock rate from 100Hz to 1000Hz. Modify memory allocation algorithms to accommodate PCI boards with fixed-upper-32-bit DMA address limitations. Chief Technologist for World-Wide Engineering. Sequent OEM-interface owner for Bravo. Ongoing engineering training in software debugging and performance programming. Head up ptx4.4 crash-

dump team. Architectural decision-maker for PTX4.3 technical conflicts. Implementation of algorithms to take advantage of non-uniform memory access (NUMA) performance potential (quad-aware locks (user & kernel), quad-aware reference counters, remote execution, memory allocation, STREAMS-buffer allocation, online-offline processing, global TLB shoot, read-copy lock mechanism, quads-on-Symmetry simulator). Design of algorithms to take advantage of NUMA performance potential (above plus kernel debugger, cache simulator, (very soft) realtime enhancements, queued locks, timeouts, and callback load balancing). PTX4.3 planning and prototyping. Spearhead use of WWW intranet for design documentation. PTX4.1 virtual-memory system performance analysis, design of performance improvements. PTX4.1 and PTX4.0 testing and debuggability improvements. Technical liaison to Intel. Software-quality analysis of PTX2.1. High-performance mutual-exclusion primitives for PTX4.0. Performance analysis of Windows NT. TCP4.0 performance improvements and bug fixes. Performance estimation for TPC benchmarks on ESP hardware. Member of Sequent Patent Committee. Improve testing of ptx/BACKUP as part of redlight (customer critical situation) effort. Prototyped modified locking in STREAMS subsystem. Prototyped 3x performance improvement in poll() system call. Fully parallel general-purpose in-kernel memory allocator for PTX2.0 (orders-of-magnitude performance increase). Global TLB-flush mechanism for PTX2.0. Performance analysis of FDDI. Develop tools to enable differential profiling of SMP software (allows quickly focusing on hot spots in large software systems). Performance analysis support for TPC benchmark efforts. Election algorithms for ptx/CLUSTERS 1.0. Fully parallel STREAMS buffer allocator for ptx1.4 (orders-of-magnitude performance increase). DMA-mapping performance improvements in ptx1.3. Demonstrated 5,000-connection capability in TCP. Analysis of configuration restrictions on initial VME offering (this analysis led to changes that greatly relaxed these restrictions).

- Jan. 1986 - Sept. 1990: SRI International. Representative projects: Development of routing and congestion-avoidance algorithms for Research Internet Gateway project. Analysis of routing and congestion avoidance algorithms in packet-radio project. Prototyping of a cell-based packet switch. System administrator for Unix-based timesharing machine.
- June 1981 - Dec. 1985: Self-employed contract programmer. Representative projects: Calibration software for acoustic navigation system. Part of team producing software release for card-key security system. Part of team designing and implementing card-key dining-hall system. Designed and implemented stockbroker client-tracking system. Designed and implemented energy-management system, which won an award from a California utility for its energy savings.
- June 1977 - Sept. 1980: Oregon State University Computer Center. Maintained, operated, and rewrote student housing assignment and billing system.
- March 1977 - Sept. 1977: Teaching Assistant for PDP-12 assembly-language introduction class (CS215 at Oregon State University).
- Sept. 1975 – Dec. 1975: Constructed computer-dating program as charity fund-raiser for local chapter of National Honor Society.

Community Activities.

- Webmaster for Precision Education to Accelerate Kids in School (PEAKS), October 1999-May 2000.
- Member of Board of Directors of Families for Early Autism Treatment (FEAT), Oregon, 1995-1999. Vice President of FEAT of Oregon, May 1996 through Feb 1998. Webmaster for FEAT of Oregon, July 1997 through Jun 1999. President of FEAT of Oregon, Feb 1998-Jan 1999.
- Chair of conference committee for the 1999 Innovative and Effective Interventions for Autism Conference, sponsored by FEAT of Oregon. More than 500 attendees from across the U.S., Canada, Europe, and the Pacific Rim.
- Western U.S. and Canada Coordinator for 1997 Princeton Child Development Institute (PCDI) teleconference.
- Member of conference committee for the 1996 Pacific Northwest Early Intervention Conference (treatment for autism, PDD, and related neurological disorders), sponsored by FEAT of Oregon. (Primary focuses: publicity, continuing education, audio-visual, efficiency expert, attendance projections. 725 attendees from across U.S., Canada, Mexico, and the Pacific Rim.)
- Organized Fourth-of-July neighborhood fun run 1995-present. Collected donations for neighborhood fireworks display 1995-present.

Other Activities.

- Ran in the annual Hood-to-Coast Relay from 1991-2000, in 2002-2003, and in 2010.
- Ran in the annual Rainier-to-Pacific Relay from 2001-2002.
- Ran in the annual Civil-War Relay from 1993-2002.
- Member of SRI International's Corporate Cup track team, which placed first in Division II in the National Corporate Cup Track Meet in 1986 through 1989, and placed second in 1990. (Ran the 10K and a few relays with distances of 800 meters or greater.)