

Paul E. McKenney  
December 22, 2020

Role: Maintainer of read-copy update in the Linux kernel, member of ISO SC22 WG21 (C++) standards committee, collaborator on Linux Kernel Memory Model, and strongly interested in validation technologies that can credibly handle concurrent software with 20 billion instances running, and that permit use in datacenters running millions of systems. More than 30 years experience with concurrent software (up to 4096-CPU shared-memory systems), including real time response and energy efficiency. Significant experience with legal issues.

#### 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>

Blog: <http://paulmck.livejournal.com>.

#### Web pages:

1. <http://www.rdrop.com/~paulmck>
2. <https://www.kernel.org/pub/linux/kernel/people/paulmck/>

#### Books.

1. “Is Parallel Programming Hard, And If So, What Can You Do About It?”, First Edition published March 10, 2014 (<https://kernel.org/pub/linux/kernel/people/paulmck/perfbook/perfbook-e1.html>). A Chinese translation entitled “深入理解并行编程” (“In-depth understanding of parallel programming”) was published on July 1, 2017 by 谢宝友 (Xie Baoyou) and 鲁阳 (Lu Yang).

#### Book Chapters.

1. “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. “RCU Usage In the Linux Kernel: Eighteen Years Later”, with Joel Fernandes (lead author), Silas Boyd-Wickizer, and Jonathan Walpole. In ACM SIGOPS Operating Systems Review pp. 47-63.
2. “An HTM-based update-side synchronization for RCU on NUMA systems”, with Seongjae Park (lead author), Laurent Dufour, and Heon Y. Yeom. In EuroSys 2020, pp. 37:1-37:15.
3. “A critical RCU safety property is... ease of use!” In SYSTOR 2019, pp. 132-143.
4. “How verified (or tested) is my code? Falsification-driven verification and testing”, with Alex Groce, Iftekhar Ahmed, Carlos Jensen, and Josie Holmes. In Automated Software Engineering June 2018 (<https://rdcu.be/2SZd>).
5. “Multi-core systems modeling for formal verification of parallel algorithms” with Mathieu Desnoyers and Michel Dagenais. August 2013 SIGOPS Operating Systems Review.
6. “Structured deferral: synchronization via procrastination.” July 2013 Communications of the ACM. Also appeared in ACM Queue in May 2013.
7. “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.
8. “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.
  9. “Scalable Concurrent Hash Tables via Relativistic Programming”, August 2010 Operating Systems Review, with Josh Triplett and Jonathan Walpole.
  10. “Introducing Technology Into the Linux Kernel: A Case Study”, July 2008 Operating Systems Review (Linux issue), with Jonathan Walpole.
  11. “Responsive systems: An introduction”, with Robert F. Berry and Francis N. Parr. April-June 2008 IBM Systems Journal.
  12. “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.
  13. “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.
  14. “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>
  15. “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.
  16. “Differential Profiling”, March 1999 Software--Practice and Experience. Revision of 1995 paper.
  17. “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.
  18. “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.
  19. “Stochastic Fairness Queuing”, June 1991 Internetworking Research and Experience. Revision of the 1990 paper.
  20. “Physical- and Link-Layer Modeling of Packet-Radio Network Performance”, January 1991 IEEE Journal on Selected Areas in Communications, with Peter E. Bausbacher.
  21. “An Interleave Principle for Demonstrating Concurrent Programs”, IEEE Software, October 1984, with Ted G. Lewis and Keith R. Spitz.

#### Refereed Conference Papers.

1. “A Critical RCU Safety Property Is.. Ease of Use!!!”, Systor 2019, June 2019.
2. “Frightening Small Children and Disconcerting Grown-ups: Concurrency in the Linux Kernel”, ASPLOS'18, with Jade Alglave (lead author), Luc Maranget, Andrea Parri, and Alan Stern, March 2018.
3. “Verification of Tree-Based Hierarchical Read-Copy Update in the Linux Kernel”, 2018 Design, Automation, and Test in Europe (DATE18), with Lihao Liang (lead author), Daniel Kroening, and Tom Melham, March 2018.
4. “The RCU-Reader Preemption Problem in VMs”, 2017 USENIX Annual Technical Conference (USENIX ATC 17), with Aravinda Prasad (lead author) and K. Gopinath, July 2017.
5. “How Verified is My Code? Falsification-Driven Verification”, 30th IEEE/ACM International Conference on Automated Software Engineering (ASE 2015), with Alex Groce (lead author) and Iftekhar Ahmed, November 2015.
6. “Resizable, Scalable, Concurrent Hash Tables via Relativistic Programming”, USENIX Annual Technical Conference, with Josh Triplett (lead author) and Jonathan Walpole, June 2011.
7. “‘Real Time’ vs. ‘Real Fast’: How to Choose?”, OLS, July 2008.
8. “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.

9. "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.
10. "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.
11. "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.
12. "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.
13. "Making RCU Safe for Deep Sub-Millisecond Response Realtime Applications", with Dipankar Sarma. USENIX (UseLinux track), June 2004.
14. "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.
15. "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.
16. "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.
17. "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.
18. "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.
19. "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.
20. "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.
21. "Differential Profiling", MASCOTS'95. Describes technique for locating scaling problems that are otherwise difficult to locate. Used habitually by some prominent Linux-community members.
22. "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.
23. "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.
24. "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.
25. "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.

26. “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>.
27. “Physical- and Link-Layer Modeling of Packet-Radio Network Performance”, MILCOM'90, with Peter E. Bausbacher.
28. “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.
29. “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. “A Relaxed Guide to memory\_order\_relaxed”, with Hans Boehm. In CPPCON, September 16, 2020.
2. “Dependency ordering in the Linux kernel”, Linux Plumbers Conference LLVM Microconference, with Will Deacon (lead author) and Peter Zijlstra, August 27, 2020.
3. “A Realtime Tour Through BPF”, Linux Plumbers Conference Real Time Microconference, August 24, 2020.
4. “RCU configuration, operation, and upcoming changes for real-time workloads”, Linux Plumbers Conference Real Time Microconference, September 11, 2019.
5. “Making SCHED\_DEADLINE safe for kernel threads”, Linux Plumbers Conference Scheduler Microconference, September 9, 2019.
6. “RCU's First-Ever CVE, and How I Lived to Tell the Tale”, Beaver BarCamp, April 5, 2019.
7. “RCU's First-Ever CVE, and How I Lived to Tell the Tale”, linux.conf.au, January 23, 2019.
8. “The Exciting New Future of Safe Reclamation for High Performance”, with Michael Wong and Maged Michael, CPPCON 2018, September 25, 2018.
9. “How Will Linux Handle Quantum Computing? An entangled superposition of views”, Beaver BarCamp, April 7, 2018.
10. “What Happened to the Linux-Kernel Memory Model?”, with Jade Alglave, Luc Maranget, Andrea Parri, Alan Stern, 2018 linux.conf.au lightning talk, January 26, 2018.
11. “Can RCU and CPU Hotplug Survive the Attack of the Killer Virtual Environments?”, 2018 linux.conf.au, January 25, 2018.
12. “Decoding Those Inscrutable RCU CPU Stall Warnings : 'They are for your own good! Honest!!!'”, 2018 linux.conf.au Kernel Minisummit, January 22, 2018.
13. “Linux Kernel Memory Model” demo, 2017 Linaro Connect San Francisco, September 29, 2017.
14. “C++ RCU Status”, 2017 SG14 meeting at CPPCON, September 27, 2017.
15. “The landscape of parallel programming models: is it still hard or just OK?”, 2017 CPPCON, with Michael Wong and Maged Michael, September 25, 2017.
16. “Linux-Kernel Memory Ordering Workshop”, 2017 Linux Plumbers Conference Microconference, with Jade Alglave, Luc Maranget, Andrea Parri, Alan Stern, and Will Deacon, September 15, 2017.
17. “How Will Linux Handle Quantum Computing? An entangled superposition of views”, 2017 Linux Plumbers Conference, September 13, 2017.
18. “Decoding Those Inscrutable RCU CPU Stall Warnings : 'They are for your own good! Honest!!!'”, 2017 Open Source Summit North America, September 12, 2017.
19. “Applying Mutation Analysis On Kernel Test Suites: An Experience Report”, 12th International Workshop on Mutation Analysis, with Iftekhar Ahmed, Rahul Gopinath, Carlos Jensen, and Alex Groce, March 2017.
20. “Does RCU Really Work? And if so, how do we know?”, Multicore World, February 2017.
21. “Linux-Kernel Memory Ordering: Help Arrives At Last!”, linux.conf.au, with Jade Alglave, Luc Maranget, Andrea Parri, and Alan Stern, January 2017.
22. “Does RCU Really Work? And if so, how do we know?”, linux.conf.au Kernel Minisummit, January 2017.
23. “Linux-Kernel Memory Ordering: Help Arrives At Last!”, Linux Kernel Summit, with Jade Alglave, Luc Maranget, Andrea Parri, and Alan Stern, November 2016.
24. “Tracing and Linux-Kernel RCU”, Tracing Summit, October 2016.

25. "Linux-Kernel Memory Ordering: Help Arrives At Last!", LinuxCon EU, with Jade Alglave, Luc Maranget, Andrea Parri, and Alan Stern, October 2016.
26. "A lock-free concurrency toolkit for deferred reclamation and optimistic speculation", CPPCON, with Michael Wong and Maged Michael, September 2016.
27. "RCU and C++", CPPCON, September 2016.
28. "Beyond the Issaquah Challenge: High-Performance Scalable Complex Updates", CPPCON, September 2016.
29. "High-Performance and Scalable Updates: The Issaquah Challenge", invited presentation to ACM Applicative Conference, June 2016.
30. "Practical Experience With Formal Verification Tools", Beaver BarCamp, April 2016.
31. "Practical Experience With Formal Verification Tools", Verified Trustworthy Software Systems Specialist Meeting, April 2016.
32. "Linux-Kernel Community Validation Practices", The Royal Society Verified Trustworthy Software Systems Meeting, "Verification in Industry" panel discussion, April 2016.
33. "What Happens When 4096 Cores All Do synchronize\_rcu\_expedited()?", linux.conf.au, February 2016.
34. "Mutation Testing and RCU", linux.conf.au Kernel Miniconf, February 2016.
35. "Formal Verification and Linux-Kernel Concurrency", Dagstuhl Seminar 15191 "Compositional Verification Methods for Next-Generation Concurrency", May 2015.
36. "Linearizability: Who Really Needs It?", Dagstuhl Seminar 15191 "Compositional Verification Methods for Next-Generation Concurrency", May 2015.
37. "Some Examples of Kernel-Hacker Informal Correctness Reasoning", Dagstuhl Seminar 15191 "Compositional Verification Methods for Next-Generation Concurrency", May 2015.
38. "Formal Verification and Linux-Kernel Concurrency", Beaver Barcamp, Corvallis, OR, April 2015
39. "High-Performance and Scalable Updates: The Issaquah Challenge", linux.conf.au, January 2015.
40. "Bare-Metal Multicore Performance in a General-Purpose Operating System (Adventures in Ubiquity)", linux.conf.au, January 2015.
41. "Read-Copy Update (RCU) Validation and Verification for Linux", Galois Tech Talk, November 2014.
42. "C++ Memory Model Meets High-Update-Rate Data Structures", CPPCON, September 2014.
43. "Reordering and Verification at the Linux Kernel", Invited presentation to REORDER and EC2 workshops in Vienna Summer of Logic, July 2014.
44. "Bare-Metal Multicore Performance in a General-Purpose Operating System (Now With Added Energy Efficiency)", Linux Collaboration Summit, March 28, 2014.
45. "But What About Updates?", Linux Collaboration Summit, March 28, 2014.
46. "Bare-Metal Multicore Performance in a General-Purpose Operating System (Now With Added Energy Efficiency)", linux.conf.au, January 9, 2014.
47. "Advances in Validation of Concurrent Software", linux.conf.au, January 8, 2014.
48. "State of Linux -rt Patchset", linux.conf.au Kernel Miniconference, January 7, 2014.
49. "Bare-Metal Multicore Performance in a General-Purpose Operating System (Now With Added Energy Efficiency)", Real-Time Linux Workshop, October 30, 2013.
50. "Configuring RCU In Your Linux Kernel: An Expert Embedded System Integrator's Guide", Real-Time Linux Workshop, October 29, 2013.
51. "Introduction to RCU Concepts: Liberal application of procrastination for accommodation of the laws of physics — for more than two decades", Hands-On Tutorial on Scalability With Userspace RCU, LinuxCon Europe, October 21, 2013.
52. "Improving Energy Efficiency On Asymmetric Multiprocessing Systems", Power-Efficient Scheduling Microconference at Linux Plumbers Conference, September 20, 2013, with Dietmar Eggemann and Robin Randhawa.
53. "But What About Updates?", Scaling Microconference at Linux Plumbers Conference, September 18, 2013.
54. "Bare-Metal Multicore Performance in a General-Purpose Operating System", Linux Plumbers Conference, September 18, 2013.
55. "Advances in Validation of Concurrent Software", Linux Plumbers Conference, September 18, 2013.
56. "Beyond Expert-Only Parallel Programming?", LinuxCon North America, September 16, 2013.
57. "Improving Energy Efficiency On Asymmetric Multiprocessing Systems", HOTPAR 2013, June 21, 2013 with Dietmar Eggemann and Robin Randhawa.
58. "Bare-Metal Multicore Performance in a General-Purpose Operating System", Linux Foundation Enterprise End-User Summit, May 15, 2013.

59. "Bare-Metal Multicore Performance in a General-Purpose Operating System ", Beaver Barcamp, Corvallis, OR, April 20, 2013.
60. "Bare-Metal Multicore Performance in a General-Purpose Operating System ", Multicore World, Wellington, New Zealand, February 19, 2013.
61. "Real-Time Response on Multicore Systems: It Is Bigger Than I Thought", linux.conf.au, Canberra, Australia, January 31, 2013.
62. "Making RCU Respect Your Device's Battery Lifetime", linux.conf.au, Canberra, Australia, January 30, 2013.
63. "Validating Core Parallel Software?", linux.conf.au Open Software miniconference, Canberra, Australia, January 29, 2013.
64. "Accommodating the Laws of Physics: RCU", The SIGPLAN Programming Languages Mentoring Workshop (PLMW) 2013, Rome, Italy, January 22, 2013.
65. "What Is RCU?", Siemens Linux Community Event, December 5, 2012.
66. "CPU Hotplug, RCU, and big.LITTLE", Linaro Connect, November 1, 2012.
67. "Beyond expert-only parallel programming?", RACES'12: Relaxing Synchronization for Multicore and Manycore Scalability, October 21, 2012.
68. "On-Chip Cache Coherence and Real-Time Systems and What is New in RCU for Real Time", Real Time Linux Workshop, October 19, 2012.
69. "Getting RCU Further Out Of The Way", Real Time Microconference of Linux Plumbers Conference, August 31, 2012.
70. "Scheduling and big.LITTLE Architecture", Scheduling Microconference of Linux Plumbers Conference, August 29, 2012.
71. "Real-Time Response on Multicore Systems: It is Bigger Than You Think", Scaling Microconference of Linux Plumbers Conference, August 29, 2012.
72. "Cleaning Up Linux's CPU Hotplug For Real Time and Energy Management ", work-in-progress session at ECRS'12, July 11, 2012, Pisa, Italy. With Thomas Gleixner and Vincent Guittot.
73. "Real-Time Response on Multicore Systems: It Is Bigger Than You Think", keynote at the OSPERT'12 workshop, July 10, 2012, Pisa, Italy.
74. "Retrofitting Parallelism Considered Grossly Sub-Optimal", HotPar'12, June 2012.
75. "Whacking Droids: How to Extract Requirements from Flame Wars", April 4, 2012 to Linux Collaboration Summit, San Francisco, CA USA.
76. "Validating Core Parallel Software", April 4, 2012 to Linux Collaboration Summit, San Francisco, CA USA.
77. "Making RCU Safe For Battery-Powered Devices", February 15, 2012 to Embedded Linux Conference, San Francisco, CA USA.
78. "On migrate\_disable() and latencies", October 21, 2011 to the 13<sup>th</sup> Real Time Linux Workshop in Prague, Czech Republic.
79. "Validating Core Parallel Software", October 16, 2011 to the China Linux Kernel Developer Conference in Nanjing <http://blog.csdn.net/lengyuex/article/details/6874502>.
80. "Confessions of a Recovering Proprietary Programmer", August 1, 2011 to Linaro Connect in Cambourne UK.
81. "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](http://www.digitimes.com.tw/seminar/android_1000428.htm)
82. "Whacking Droids: How to Extract Requirements from Flame Wars", Beaver Barcamp 7 (Oregon State University), April 2011.
83. "Verifying Parallel Software: Can Theory Meet Practice?", Verification of Concurrent Data Structures (Verico), January 2011.
84. "Whacking Droids: How to Extract Requirements from Flame Wars", linux.conf.au, January 2011.
85. "Is Parallel Programming Hard, And, If So, Why?", linux.conf.au Multicore miniconference, January 2011.
86. "Verifying Parallel Software: Can Theory Meet Practice?", linux.conf.au Multicore miniconference, January 2011.
87. "Multi-Core Memory Models and Concurrency Theory: A View from the Linux Community", Dagstuhl Multi-Core Memory Models and Concurrency Theory Workshop, January 2011.
88. "Extreme Energy Efficiency", October 16, 2010 to Barcamp at Oregon State University.
89. "When Do Real Time Systems Need Multiple CPUs?", 12<sup>th</sup> Real-Time Linux Workshop, October 2010.
90. "Simplicity Through Optimization", linux.conf.au, January 2010.

91. “Real Time vs. Real Fast: How to Choose?”, the Eleventh Real-Time Linux Workshop, Dresden, Germany, September 2009.
92. “Deterministic Synchronization in Multicore Systems: the Role of RCU”, the Eleventh Real-Time Linux Workshop, Dresden, Germany, September 2009.
93. “Using a Malicious User-Level RCU to Torture RCU-Based Algorithms”, linux.conf.au, January 2009.
94. “Is Parallel Programming Hard, And If So, Why?”, linux.conf.au, January 2009.
95. “Real Time' vs. 'Real Fast': How to Choose?”, MontaVista Vision 2008, October 2008.
96. “Concurrency and Race Conditions”, Linux Plumbers Conference Student Day, September 2008.
97. “Introducing Technology Into Linux”, 2008 Linux Developer Symposium – China, February 2008.
98. “After 25 Years, C/C++ Understands Concurrency”, linux.conf.au, February 2008.
99. “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.
100. “Real Time Linux Technology: a Deeper Dive”, presentation at MontaVista VISION 2007, October 2007
101. “Real Time Linux Technology”, keynote at MontaVista VISION 2007, October 2007
102. “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 12<sup>th</sup> International Symposium on High-Performance Computer Architecture, February 2006.
103. “Steamroller Testing”. Presentation at the 2006 linux.conf.au conference in Dunedin, New Zealand. January 2006.
104. “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.
105. “Linux Kernel Scalability: Using the Right Tool for the Job”. linux.conf.au, April 2005.
106. “Linux Kernel Scalability: Using the Right Tool for the Job”. Ottawa Linux Symposium, July 2004.
107. “Storage-Related Requirements for the 2.7 Linux Kernel”. Invited presentation in Linux Kernel Summit, July 2003.

#### Unrefereed Journal Articles

1. P1726R4: “Pointer lifetime-end zap”, ISO SC22 WG21 (C++ Language), with Maged Michael, Jens Maurer, Peter Sewell, Martin Uecker, Hans Boehm, Hubert Tong, Niall Douglas, Thomas Rodgers, Will Deacon, Michael Wong, David Goldblatt, Kostya Serebryany, and Anthony Williams, July 8, 2020.
2. “Concurrency bugs should fear the big bad data-race detector (part 2)”, Linux Weekly News, with Marco Elver (lead author), Dmitry Vyukov, Andrey Konovalov, Alexander Potapenko, Kostya Serebryany, Alan Stern, Andrea Parri, Akira Yokosawa, Peter Zijlstra, Will Deacon, Daniel Lustig, Boqun Feng, Joel Fernandes, Jade Alglave, and Luc Maranget, April 14, 2020.
3. “Concurrency bugs should fear the big bad data-race detector (part 1)”, Linux Weekly News, with Marco Elver (lead author), Dmitry Vyukov, Andrey Konovalov, Alexander Potapenko, Kostya Serebryany, Alan Stern, Andrea Parri, Akira Yokosawa, Peter Zijlstra, Will Deacon, Daniel Lustig, Boqun Feng, Joel Fernandes, Jade Alglave, and Luc Maranget, April 8, 2020.
4. P0124R7: “Linux-Kernel Memory Model”, SO SC22 WG21 (C++ Language), with Ulrich Weigand, Andrea Parri, Boqun Feng, and Alan Stern, March 1, 2020.
5. P1726R3: “Pointer lifetime-end zap”, ISO SC22 WG21 (C++ Language), with Maged Michael, Jens Mauer, Peter Sewell, Martin Uecker, Hans Boehm, Hubert Tong, Niall Douglas, Will Deacon, Michael Wong, and David Goldblatt, February 21, 2020.
6. P2055R0: “A Relaxed Guide to memory\_order\_relaxed”, ISO SC22 WG21 (C++ Language), with Hans Boehm, January 12, 2020.
7. “Calibrating your fear of big bad optimizing compilers”, Linux Weekly News, October 11, 2019, with Jade Alglave, Will Deacon, Boqun Feng, David Howells, Daniel Lustig, Luc Maranget, Andrea Parri, Nicholas Piggin, Alan Stern, Akira Yokosawa, and Peter Zijlstra.
8. “Who’s afraid of a big bad optimizing compiler?”, Linux Weekly News, July 15, 2019, with Jade Alglave, Will Deacon, Boqun Feng, David Howells, Daniel Lustig, Luc Maranget, Andrea Parri, Nicholas Piggin, Alan Stern, Akira Yokosawa, and Peter Zijlstra.
9. P1726R1: “Pointer lifetime-end zap”, ISO SC22 WG21 (C++ Language), with Maged Michael, Jens Maurer, Peter Sewell, Martin Uecker, Hans Boehm, Hubert Tong, Niall Douglas, Will Deacon, and Michael Wong, August 1, 2019.

10. P1726R0: "Pointer lifetime-end zap", ISO SC22 WG21 (C++ Language), with Maged Michael, Jens Maurer, Peter Sewell, Martin Uecker, Hans Boehm, Hubert Tong, and Niall Douglas, June 17, 2019.
11. P1382R1: "volatile\_load<T> and volatile\_store<T>", ISO SC22 WG21 (C++ Language), with JF Bastien and Jeffrey Yasskin, March 10, 2019.
12. P1121R1: "Hazard Pointers: Proposed Interface and Wording for Concurrency TS 2", ISO SC22 WG21 (C++ Language), with Maged M. Michael (lead author), Michael Wong, Geoffrey Romer, Andrew Hunter, Arthur O'Dwyer, David S. Hollman, JF Bastien, Hans Boehm, David Goldblatt, Frank Birbacher, Mathias Stearn, January 20, 2019.
13. "The RCU API, 2019 edition", Linux Weekly News, January 23, 2019.
14. P1382R0: "volatile\_load<T> and volatile\_store<T>", ISO SC22 WG21 (C++ Language), with JF Bastien, January 11, 2019.
15. P1122R2: "Proposed Wording for Concurrent Data Structures: Read-Copy-Update (RCU)", ISO SC22 WG21 (C++ Language), with Michael Wong, Maged M. Michael, Geoffrey Romer, Andrew Hunter, Arthur O'Dwyer, David S. Hollman, JF Bastien, Hans Boehm, David Goldblatt, Frank Birbacher, and Erik Rigtorp, November 25, 2018.
16. P1121R0: "Hazard Pointers: Proposed Interface and Wording for Concurrency TS 2", ISO SC22 WG21 (C++ Language), with Maged M. Michael (lead author), Michael Wong, Geoffrey Romer, Andrew Hunter, Arthur O'Dwyer, David S. Hollman, JF Bastien, Hans Boehm, David Goldblatt, Frank Birbacher, and Mathias Stearn, October 5, 2018.
17. P0124R6: "Linux-Kernel Memory Model", ISO SC22 WG21 (C++ Language), with Ulrich Weigand, Andrea Parri, and Boqun Feng, September 27, 2018.
18. P1122R1: "Proposed Wording for Concurrent Data Structures: Read-Copy-Update (RCU)", ISO SC22 WG21 (C++ Language), with Michael Wong, Maged M. Michael, Geoffrey Romer, Andrew Hunter, Arthur O'Dwyer, David S. Hollman, JF Bastien, Hans Boehm, David Goldblatt, Frank Birbacher, July 4, 2018.
19. P1122R0: "Proposed Wording for Concurrent Data Structures: Read-Copy-Update (RCU)", I Michael Wong, Maged M. Michael, Geoffrey Romer, Andrew Hunter, Arthur O'Dwyer, David S. Hollman, JF Bastien, Hans Boehm, David Goldblatt, and Frank Birbacher June 7, 2018.
20. P1074R0: "CWG defect Defined Behavior of Invalid Pointers", ISO SC22 WG21 (C++ Language), with Maged Michael (lead author), David Goldblatt, and Michael Wong, May 7, 2018.
21. P0772R1: "Execution Agent Local Storage", ISO SC22 WG21 (C++ Language), with Nat Goodspeed (lead author), Michael Wong, Jared Hoberock, H. Carter Edwards, Tony Tye, Alex Voicu, Gordon Brown, Mark Hoemmen, May 7, 2018.
22. [P1022R0](#): "Material for 2018 JAX Discussions of Hazard Pointer and Read-Copy-Update (RCU)", ISO SC22 WG21 (C++ Language), with Michael Wong, Maged M. Michael, Geoffrey Romer, Andrew Hunter, Arthur O'Dwyer, David S. Hollman, JF Bastien, Hans Boehm, and David Goldblatt, April 20, 2018.
23. P0566R5: "Proposed Wording for Concurrent Data Structures: Hazard Pointer and Read-Copy Update (RCU)", ISO SC22 WG21 (C++ Language), Michael Wong, Maged M. Michael, Geoffrey Romer, Andrew Hunter, Arthur O'Dwyer, David S. Hollman, JF Bastien, Hans Boehm, David Goldblatt, and Frank Birbacher, May 6, 2018..
24. P0124R5: "Linux-Kernel Memory Model", ISO SC22 WG21 (C++ Language), with Ulrich Weigand, Andrea Parri, and Boqun Feng, April 6, 2018.
25. P0750R1: "Consume", ISO SC22 WG21 (C++ Language), with JF Bastien (lead author), February 11, 2018.
26. P0566R4: "Proposed Wording for Concurrent Data Structures: Hazard Pointer and Read-Copy-Update (RCU)", ISO SC22 WG21 (C++ Language), with Michael Wong, Maged M. Michael, Geoffrey Romer, Andrew Hunter, Arthur O'Dwyer, David S. Hollman, JF Bastien, Hans Boehm, and David Goldblatt, February 11, 2018. (I am lead author for the RCU section, Maged Michael is lead author for the Hazard Pointers section, and Michael Wong is overall lead author.)
27. P0868R2: "Selected RCU Litmus Tests", ISO SC22 WG21 (C++ Language), with Alan Stern, Andrew Hunter, Jade Alglave, and Luc Maranget, February 9, 2018.
28. P0868R1: "Selected RCU Litmus Tests", ISO SC22 WG21 (C++ Language), with Alan Stern, Andrew Hunter, Jade Alglave, and Luc Maranget, November 20, 2017.
29. P0868R0: "Selected RCU Litmus Tests", ISO SC22 WG21 (C++ Language), with Alan Stern, and Andrew Hunter, November 2017.



30. P0566R2: "Proposed Wording for Concurrent Data Structures: Hazard Pointer and Read-Copy-Update (RCU)", ISO SC22 WG21 (C++ Language), with Michael Wong (lead author), Maged Michael, Geoffrey Romer, and Andrew Hunter, July 30, 2017.
31. P0233R5: "Hazard Pointers: Safe Reclamation for Optimistic Concurrency", ISO SC22 WG21 (C++ Language), with Maged M. Michael (lead author), Michael Wong, Arthur O'Dwyer, David Hollman, Geoffrey Romer, and Andrew Hunter, July 30, 2017.
32. P0190R4: "Proposal for New memory order consume Definition", ISO SC22 WG21 (C++ Language), with Michael Wong, Hans Boehm, Jens Maurer, Jeffrey Yasskin, and JF Bastien, July 28, 2017.
33. P0124R3: "Linux-Kernel Memory Model", ISO SC22 WG21 (C++ Language), with Ulrich Weigand, Andrea Parri, and Boqun Feng, June 26, 2017
34. P0566R1: "Proposed Wording for Concurrent Data Structures: Hazard Pointer and Read-Copy-Update (RCU)", ISO SC22 WG21 (C++ Language), with Michael Wong (lead author), Maged Michael, Geoffrey Romer, and Andrew Hunter, June 19, 2017.
35. P0233R4: "Hazard Pointers: Safe Reclamation for Optimistic Concurrency", ISO SC22 WG21 (C++ Language), with Maged M. Michael (lead author), Michael Wong, Arthur O'Dwyer, David Hollman, Geoffrey Romer, and Andrew Hunter, June 18, 2017.
36. "A formal kernel memory-ordering model (part 1)", Linux Weekly News, April 14, 2017, with Jade Alglave, Luc Maranget, Andrea Parri, and Alan Stern.
37. "A formal kernel memory-ordering model (part 2)", Linux Weekly News, April 20, 2017, with Jade Alglave, Luc Maranget, Andrea Parri, and Alan Stern.
38. "RCU and the mid-boot dead zone", Linux Weekly News, March 2017.
39. P0190R3: "Proposal for New memory order consume Definition", ISO SC22 WG21 (C++ Language), with Michael Wong, Hans Boehm, Jens Maurer, Jeffrey Yasskin, and JF Bastien, February 5, 2017.
40. P0461R1: "Proposed RCU C++ API", ISO SC22 WG21 (C++ Language), with Maged Michael, Michael Wong, Isabella Muerte, Arthur O'Dwyer, and David Hollman, February 5, 2017.
41. P0462R1: "Marking memory order consume Dependency Chains", ISO SC22 WG21 (C++ Language), with Torvald Riegel, Jeff Preshing, Hans Boehm, Clark Nelson, Olivier Giroux, Lawrence Crowl, JF Bastien, and Michael Wong, February 5, 2017.
42. P0566R0: "Proposed Wording for Concurrent Data Structures: Hazard Pointer and Read-Copy-Update (RCU)", ISO SC22 WG21 (C++ Language), with Michael Wong (lead author) and Maged Michael, February 6, 2017.
43. P0233R3: "Hazard Pointers: Safe Reclamation for Optimistic Concurrency", ISO SC22 WG21 (C++ Language), with Maged M. Michael (lead author), Michael Wong, Arthur O'Dwyer, and David Hollman, February 6, 2017.
44. P0124R2: "Linux-Kernel Memory Model", ISO SC22 WG21 (C++ Language), with Ulrich Weigand, Andrea Parri, and Boqun Feng, June 26, 2016.
45. P0233R2: "Hazard Pointers: Safe Reclamation for Optimistic Concurrency", ISO SC22 WG21 (C++ Language), with Maged M. Michael (lead author), Michael Wong, and Arthur O'Dwyer, February 6, 2017.
46. P0461R0: "Proposed RCU C++ API", ISO SC22 WG21 (C++ Language), with Maged Michael, Michael Wong, Isabella Muerte, and Arthur O'Dwyer, October 16, 2016.
47. P0279R1: "Read-Copy Update (RCU) for C++", ISO SC22 WG21 (C++ Language), August 25, 2016.
48. "Semantics of MMIO mapping attributes across architectures", Linux Weekly News, August 2016.
49. P0422R0: "Out-of-Thin-Air Execution is Vacuous", ISO SC22 WG21 (C++ Language), with Alan Jeffrey, Ali Sezgin, and Tony Tye, July 27, 2016.
50. P0062R1: "When should compilers optimize atomics?", ISO SC22 WG21 (C++ Language), with JF Bastien (lead author), Peter Dimov, Hal Finkel, Michael Wong, and Jeffrey Yasskin, May 27, 2016.
51. P0190R2: "Proposal for New memory order consume Definition", ISO SC22 WG21 (C++ Language), with Michael Wong, Hans Boehm, Jens Maurer, Jeffrey Yasskin, and JF Bastien, May 25, 2016.
52. P0108R1: "Skeleton Proposal for Thread-Local Storage (TLS)", ISO SC22 WG21 (C++ Language), with JF Bastien, April 4, 2016.
53. P0190R1: "Proposal for New memory order consume Definition", ISO SC22 WG21 (C++ Language), with Michael Wong, Hans Boehm, and Jens Maurer, March 18, 2016.
54. P0279R0: "Read-Copy Update (RCU) for C++", ISO SC22 WG21 (C++ Language), February 14, 2016.
55. P0232R0: "A Concurrency Toolkit for Structured Deferral/Optimistic Speculation", ISO SC22 WG21 (C++ Language), with Michael Wong (lead author) and Maged Michael, February 12, 2016.

56. P0190R0: "Proposal for New memory order consume Definition", ISO SC22 WG21 (C++ Language), with Michael Wong, Hans Boehm, and Jens Maurer, February 10, 2016.
57. P0124R1: "Linux-Kernel Memory Model", ISO SC22 WG21 (C++ Language), with Ulrich Weigand, and Andrea Parri, January 27, 2016.
58. P0098R1: "Towards Implementation and Use of memory order consume", ISO SC22 WG21 (C++ Language), with Torvald Riegel, Jeff Preshing, Hans Boehm, Clark Nelson, Olivier Giroux, and Lawrence Crowl, January 4, 2016.
59. "Some more details on Read-Log-Update", Linux Weekly News, December 2015.
60. "Read-mostly research in 2015", Linux Weekly News, December 2015.
61. P0124R0: "Linux-Kernel Memory Model", ISO SC22 WG21 (C++ Language), with Ulrich Weigand, September 25, 2016.
62. P0062R0: "When should compilers optimize atomics?", ISO SC22 WG21 (C++ Language), with Hans Boehm (lead author), JF Bastien, Peter Dimov, Hal Finkel, Michael Wong, and Jeffrey Yasskin, September 25, 2015.
63. P0108R0: "Skeleton Proposal for Thread-Local Storage (TLS)", ISO SC22 WG21 (C++ Language), with JF Bastien, September 24, 2016.
64. P0097R0: "Use Cases for Thread-Local Storage", ISO SC22 WG21 (C++ Language), with JF Bastien, Pablo Halpern, Michael Wong, Thomas Richard William Scogland, Robert Geva, September 24, 2015.
65. P0098R0: "Towards Implementation and Use of memory order consume", ISO SC22 WG21 (C++ Language), with Torvald Riegel, Jeff Preshing, Hans Boehm, Clark Nelson, Olivier Giroux, and Lawrence Crowl, September 24, 2016.
66. "RCU requirements part 3", Linux Weekly News, August 2015.
67. "RCU requirements part 2 — parallelism and software engineering", Linux Weekly News, August 2015.
68. "Requirements for RCU part 1: the fundamentals", Linux Weekly News, July 2015.
69. "Some Examples of Kernel-Hacker Informal Correctness Reasoning", Dagstuhl Seminar 15191 "Compositional Verification Methods for Next-Generation Concurrency", May 2015.
70. "N4483: Read-Copy Update (RCU) for C++", ISO SC22 WG21 (C++ Language), Evolution Working Group, April 14, 2015.
71. N4444: "Linux-Kernel Memory Model", ISO SC22 WG21 (C++ Language), April 10, 2015.
72. N4376: "Use Cases for Thread-Local Storage" ISO SC22 WG21 (C++ Language), February 6, 2015.
73. N4375: "Out-of-Thin-Air Execution is Vacuous", ISO SC22 WG21 (C++ Language), February 6, 2015.
74. N4374: "Linux-Kernel Memory Model", ISO SC22 WG21 (C++ Language), February 6, 2015.
75. "Creating scalable APIs", Linux Weekly News, February 2015.
76. "N4376: Use Cases for Thread-Local Storage", ISO SC22 WG21 (C++ Language), Evolution Working Group, with JF Bastien, Pablo Halpern, and Michael Wong, February 2015.
77. "N4375: Out-of-Thin-Air Execution is Vacuous", ISO SC22 WG21 (C++ Language), Evolution Working Group, with Alan Jeffrey and Ali Sezgin, February 2015.
78. "Recent read-mostly research", Linux Weekly News, November 2014.
79. "N4324: Use Cases for Thread-Local Storage", ISO SC22 WG21 (C++ Language), Evolution Working Group, with JF Bastien, November 2014.
80. "N4323: Out-of-Thin-Air Execution is Vacuous", ISO SC22 WG21 (C++ Language), Evolution Working Group, with Alan Jeffrey and Ali Sezgin, November 2014.
81. "N4321: Towards Implementation and Use of memory\_order\_consume", ISO SC22 WG21 (C++ Language), Evolution Working Group, with Torvald Riegel, Jeff Preshing, Hans Boehm, Clark Nelson, and Oliver Giroux, November 2014.
82. "N4216: Out-of-Thin-Air Execution is Vacuous", ISO SC22 WG21 (C++ Language), Evolution Working Group, with Alan Jeffrey and Ali Sezgin, October 2014.
83. "The RCU API, 2014 Edition", Linux Weekly News, September 2014.
84. "Axiomatic validation of memory barriers and atomic instructions", Linux Weekly News, with Alan Stern, August 2014.
85. "N4215: Towards Implementation and Use of memory\_order\_consume", ISO SC22 WG21 (C++ Language), Evolution Working Group, with Torvald Riegel, Jeff Preshing, Hans Boehm, Clark Nelson, and Oliver Giroux, May 2014.
86. "N4037: Non-Transactional Implementation of Atomic Tree Move", ISO SC22 WG21 (C++ Language), Evolution Working Group, May 2014.

87. "N4036: Towards Implementation and Use of memory\_order\_consume", ISO SC22 WG21 (C++ Language), Evolution Working Group, with Torvald Riegel and Jeff Preshing, May 2014.
88. "User-space RCU", Linux Weekly News, November 2013.
89. "URCU-protected hash tables", Linux Weekly News, November 2013.
90. "The URCU hash table API", Linux Weekly News, November 2013.
91. "URCU-protected queues and stacks", Linux Weekly News, November 2013.
92. "The URCU stack/queue API", Linux Weekly News, November 2013.
93. "User-space RCU: Atomic-operation and utility API", Linux Weekly News, November 2013.
94. "User-space RCU: Memory-barrier menagerie", Linux Weekly News, November 2013.
95. "The user-space RCU API", Linux Weekly News, November 2013.
96. "The RCU-protected list API", Linux Weekly News, November 2013.
97. "The RCU-barrier menagerie", Linux Weekly News, November 2013.
98. "Simplifying RCU", Linux Weekly News, March 2013.
99. "The new visibility of RCU processing", Linux Weekly News, October 2012.
100. "Signed overflow optimization hazards in the kernel", Linux Weekly News, August 15, 2012.
101. "A big.LITTLE scheduler update", Linux Weekly News, June 12, 2012.
102. "The Linaro Connect scheduler minisummit", Linux Weekly News, February 2012.
103. "How to make a positive difference in a FOSS project", Linaro OCTO, January 2013.  
[http://www.linaro.org/assets/common/MakingaPositiveDifferenceinFOSS\\_Jan2012.pdf](http://www.linaro.org/assets/common/MakingaPositiveDifferenceinFOSS_Jan2012.pdf)
104. "Validating Memory Barriers and Atomic Instructions", Linux Weekly News, December 2011.
105. "3.0 and RCU: what went wrong", Linux Weekly News, July 2011.
106. "ARM kernel consolidation", Linux Weekly News, May 2011.
107. "Concurrent code and expensive instructions", Linux Weekly News, January 2011.
108. "The RCU API, 2010 Edition", December 2010 Linux Weekly News.
109. "Memory-Order Rationale", ISO SC22 WG14 (C Language), October 2010, with Blaine Garst.
110. "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.
111. "Android Suspend Blockers", Linaro OCTO, August 2010, updated January 2012.  
<https://wiki.linaro.org/WorkingGroup/KernelConsolidation/Projects/AndroidSuspendBlockers>
112. "Additional Atomics Errata", ISO SC22 WG14 (C Language), May 2010.
113. "Rationale for C-Language Dependency Ordering", ISO SC22 WG14 (C Language), May 2010.
114. "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.
115. "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.
116. "Dependency Ordering for the C Memory Model", ISO SC22 WG14 (C Language), April 2010, with Clark Nelson, Hans-J. Boehm, and Lawrence Crowl.
117. "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.
118. "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.
119. "Lockdep-RCU", Linux Weekly News, February 2010.
120. "RCU: the Bloatwatch Edition", Linux Weekly News, March 2009.
121. "Hierarchical RCU", Linux Weekly News, November 2008.
122. "Example POWER Implementation for C/C++ Memory Model", ISO SC22 WG21 (C++ Language), Evolution Working Group, August 2008, with Raul Silvera. Updated September 2008.
123. "Integrating and Validating dynticks and Preemptible RCU", Linux Weekly News, April 2008.
124. "A New Interface for C++ std::duration Type", ISO SC22 WG21 (C++ Language), Evolution Working Group, April 2008, with Michael Wong.
125. "RCU part 3: the RCU API", Linux Weekly News, January 2008.
126. "What is RCU? Part 2: Usage", Linux Weekly News, January 2008.
127. "What is RCU, Fundamentally?", Linux Weekly News, December 2007.
128. "The design of preemptible read-copy update", Linux Weekly News, October 2007.

129. "Using Promela and Spin to verify parallel algorithms", Linux Weekly News, August 2007. Includes proof of correctness for QRCU.
130. "Converting Memory Fences to N2324 Form", ISO SC22 WG21 (C++ Language), Evolution Working Group, August 2007.
131. "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.
132. "Priority-Boosting RCU Read-Side Critical Sections", Linux Weekly News, February 2007.
133. "Overview of Linux-Kernel Reference Counting", ISO SC22 WG21 (C++ Language), Evolution Working Group, January 2007.
134. "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.
135. "SMP and Embedded Realtime", Linux Journal, January 2007.
136. "RCU and Unloadable Modules", Linux Weekly News, January 2007.
137. "Sleepable RCU", Linux Weekly News, October 2006.
138. "Memory Ordering in Modern Microprocessors, Part II", Linux Journal, September 2005.
139. "A Realtime Preemption Overview", Linux Weekly News, August 10, 2005.
140. "Memory Ordering in Modern Microprocessors, Part I", Linux Journal, August 2005.
141. "Attempted Summary of "RT patch acceptance" Thread", Linux Weekly News, June 7, 2005.
142. "[RFC] RCU and CONFIG\_PREEMPT\_RT progress", Linux Weekly News, May 9, 2005.
143. "Storage Improvements for 2.6 and 2.7", Linux Journal, August 2004.
144. "Scaling dcache with RCU", Linux Journal, January 2004, with Dipankar Sarma and Maneesh Soni. Shows Linux 2.6 RCU dcache modifications.
145. "Using RCU in the Linux 2.6 Kernel", Linux Journal, October 2003. Shows empirical data comparing RCU with other locking mechanisms.
146. "Benchmark Standards/Metrics and Capacity Planning", UniForum, 1994.
147. "Congestion Avoidance", Connexions, October 1990.
148. "Broadcast Storms, Nervous Hosts, and Load Imbalances", ;login: The USENIX Association Newsletter, Sept/Oct 1988.
149. "Charge Number Accounting Without Kernel Modifications", ;login: The USENIX Association Newsletter, July/Aug 1988.
150. "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. "What is RCU?", March 14, 2019 to University of Portland ACM Club.
2. "Formal Methods and the Linux Kernel : An Experience Report From an RCU Perspective", University of Cambridge (Peter Sewell's group).
3. "What is RCU?", November 22, 2018 to Portland State University CS 533 Operating Systems (Bruce Irvin).
4. "Career Trajectories: A Study in Change" October 16, 2018 to Oregon State University EE 111 Introduction to Electrical and Computer Engineering I. (Matthew Shuman)
5. "Linux-Kernel Memory Ordering: Help Arrives At Last!" May 16, 2018 Galois Tech Talk.
6. "Linux-Kernel Memory Ordering: Help Arrives At Last!" February 20, 2018 to Portland State University CS410/CS510 Advanced Topics in Concurrency (Jon Walpole).
7. "Career Trajectories: A Study in Change" November 2, 2017 to Oregon State University EE 111 Introduction to Electrical and Computer Engineering I. (Matthew Shuman)
8. "What is RCU?", October 23, 2017 to Carnegie Mellon University (CMU) CS 410 Operating Systems class. (Dave Eckhardt)
9. "What is RCU?", September 20, 2017 to Indian Institute of Science (IISc) CSA E0 210 Principles of Programming class. (K. Gopinath)
10. "Beyond the Issaquah Challenge: High-Performance Scalable Complex Updates", September 11, 2017, Silicon Valley Linux Users Group. (Kevin Dankwardt)
11. "Does RCU Really Work? And if so, how would we know?", January 31, 2017 to Northern Arizona University NAU 499 class. (Alex Groce)
12. "High-Performance and Scalable Updates: The Issaquah Challenge", June 6, 2016 to TU Dresden's Distributed Operating Systems class. (Carsten Weinhold)

13. "Formal Verification and Linux-Kernel Concurrency ", June 4, 2015 to Oregon State University's CS569 class (graduate-level software engineering, Alex Groce)
14. "Formal Verification and Linux-Kernel Concurrency ", June 2, 2015 to Oregon State University's CS362 class (undergraduate-level software engineering, Alex Groce)
15. "High-Performance and Scalable Updates: The Issaquah Challenge", June 1, 2015 to TU Dresden's Distributed Operating Systems class. (Carsten Weinhold)
16. "High-Performance and Scalable Updates: The Issaquah Challenge", May 22, 2015 to the Systems group in the Computer Science & Engineering Department of Texas A&M University. (Dilma Da Silva)
17. "Advances in Validation of Concurrent Software", April 2, 2015 to Daniel Kroening's group at University of Oxford.
18. "Education: A Foundation For Your Career", Keynote Address to OSU Career Showcase & Student Conference, December 15, 2014.
19. "But What About Updates?", June 4, 2014 to Portland State University.
20. "Education: A Foundation For Your Career", Keynote Address to OSU Career Showcase & Student Conference, June 16, 2014.
21. "What Is RCU?", May 19, 2014 to TU Dresden. (Carsten Weinhold)
22. "What Is RCU?", November 1, 2013 to University of Cambridge. (Peter Sewell)
23. "What Is RCU?", June 3, 2013 to TU Dresden. (Carsten Weinhold)
24. "What Is RCU?", June 3, 2013 to Indian Institute of Science. (K. Gopinath)
25. "What Is RCU?", May 16, 2012 to TU Dresden. (Carsten Weinhold)
26. "Validating Core Parallel Software", October 28, 2011 to TU Dresden. (Hermann Härtig)
27. "Confessions of a Recovering Proprietary Programmer", October 18, 2011 to Nanjing University.
28. "Confessions of a Recovering Proprietary Programmer", October 17, 2011 to Shanghai Jiaotong University.
29. "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>
30. "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)
31. "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.
32. "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)
33. "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)
34. "Is Parallel Programming Hard, And, If So, Why?" December 4, 2008, presentation to Bart Massey's Portland State University Multicore Computing Practicum class.
35. "Is Parallel Programming Hard, And, If So, Why?", November 24, 2008, presentation to the Oregon Institute of Technology Operating Systems class.
36. "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.
37. "RCU in the Linux Kernel", October 24, 2008, presentation to the University of Rochester Computer Science Department.
38. "Towards a User-Level RCU Implementation", October 10, 2008, presentation to Emmett Witchell's group at University of Texas at Austin.
39. "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).
40. "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.
41. "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.

42. "Linux Realtime Response: Five Realtime Myths Exposed", October 9, 2007. Presentation to UCSC computer science students.
43. "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.
44. "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.
45. "Linux Realtime Response: Five Realtime Myths Exposed", IEEE Computer Society, Oregon Chapter, January 2007. Debunks five realtime myths.
46. "Linux Realtime Response: The Next Frontier in Commercial Computing", Oregon Institute of Technology, November 20, 2006. Revision of the October 2005 AoT presentation.
47. "Extending RCU for Realtime and Embedded Workloads", University of Toronto Cider Seminar, July 18, 2006. Preview of OLS paper.
48. "Linux Realtime Response: The CONFIG\_PREEMPT Patch Set", Oregon Institute of Technology, October 6, 2005.
49. "Linux Realtime Response", IBM Academy of Technology Topical Conference on Real-Time and Safety-Critical Systems, October 24-25, 2005.
50. "Abstraction, Reality Checks, and RCU", Oregon State University, October 6, 2005.
51. "Linux Realtime Response: The CONFIG\_PREEMPT Patch Set", Oregon State University, October 6, 2005.
52. "Abstraction, Reality Checks, and RCU", University of Toronto Cider Seminar, July 26, 2005.
53. "Exploiting Deferred Destruction: An Analysis of Read-Copy-Update Techniques in Operating System Kernels", Operating System class at Portland State University, January 13, 2005.
54. "Linux Kernel Scalability: Using the Right Tool for the Job", Operating System class at Oregon Institute of Technology, Capitol Center campus. November 8, 2004.
55. "Read Copy Update", Operating System class at Oregon Institute of Technology, Capitol Center campus. November 24, 2003.
56. "SMP Programming", Technical training class at Veritas Inc, Mountain View, CA. November 4, 1997.

#### White Papers and Technical Reports

1. "Linux-Kernel Memory Ordering: Help Arrives At Last!", Dagstuhl Workshop on Concurrency with Weak Memory Models: Semantics, Languages, Compilation, Verification, Static Analysis, and Synthesis, with Jade Alglave, Luc Maranget, Andrea Parri, and Alan Stern, November 2016.
2. "Some Examples of Kernel-Hacker Informal Correctness Reasoning", paulmck.2015.06.17a, <http://www2.rdrop.com/users/paulmck/techreports/IntroRCU.2015.06.17a.pdf>, June 2015.
3. "RCU Usage In the Linux Kernel: One Decade Later" (revised), paulmck.2013.02.24a, <http://www2.rdrop.com/users/paulmck/techreports/RCUUsage.2013.02.24a.pdf>, with Silas Boyd-Wickizer and Jonathan Walpole.
4. "RCU Usage In the Linux Kernel: One Decade Later", paulmck.2012.09.17a, <http://www2.rdrop.com/users/paulmck/techreports/survey.2012.09.17a.pdf>, with Silas Boyd-Wickizer and Jonathan Walpole.
5. "Resizable, Scalable, Concurrent Hash Tables", Portland State University TR-11-01 Portland State University TR-11-01, with Josh Triplett and Jonathan Walpole.
6. "Scalable Correct Memory Ordering via Relativistic Programming", Portland State University TR-11-02, with Josh Triplett, Phil Howard, and Jonathan Walpole.
7. "Generalized Construction of Scalable Concurrent Data Structures via Relativistic Programming", Portland State University TR-11-04 with Josh Triplett, Phil Howard, and Jonathan Walpole.
8. "Is Parallel Programming Hard, And If So, Why?", Portland State University TR-09-02, February 2009, with Manish Gupta, Maged Michael, Phil Howard, Josh Triplett, and Jonathan Walpole.
9. "Comments on 'The Transactional Memory / Garbage Collection Analogy'", paulmck.2009.02.04b, <http://www2.rdrop.com/users/paulmck/techreports/gc-tm-comments.2009.02.04b.pdf>, with Jonathan Walpole, February 2009.
10. "RCU Semantics: A First Attempt", paulmck.2005.01.30a, <http://www.rdrop.com/users/paulmck/RCU/rcu-semantics.2005.01.30a.pdf>, with Jonathan Walpole, January 2005.

11. "Performance of Locking Primitives at Low Levels of Contention", paulmck.1998.07.27a, <http://www2.rdrop.com/users/paulmck/techreports/paulmck.1998.07.27a.pdf>. For graphs: <http://www2.rdrop.com/users/paulmck/techreports/paulmck.1998.07.27a.doc>.
12. "Implementation and Performance of Read-Copy-Update", paulmck.1998.06.20a.doc, <http://www2.rdrop.com/users/paulmck/techreports/paulmck.1998.06.20a.pdf>.
13. "Using Thread History to Implement Low-Overhead Solutions to Concurrency Problems", TR-SQNT-97-PEM-2.1, <http://www2.rdrop.com/users/paulmck/techreports/TR-SQNT-97-PEM-2.1.pdf>.
14. "Experiment Design for CATE", SRI Technical Note ITAD-8600-TR-93-74, SRI International, March 1993, with B. Denny.
15. "Traffic Generator Software Release Notes", SRI Technical Note ITAD-8600-TN-93-28, SRI International, February 1993, with D. Lee and B. Denny.
16. "Stochastic Fairness Queueing", paulmck.1990.03.25a, <http://www.rdrop.com/users/paulmck/scalability/paper/sfq.2002.06.04.pdf>.
17. "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. "Paul McKenney of IBM talks RCU, Quantum Computing at Linaro Connect San Francisco 2017", September 29, 2017 by Nicolas Charbonnier (Charbox). <https://www.youtube.com/watch?v=ga90waDOSEc>
2. "Kernel Developer Panel", linux.conf.au Kernel Miniconference, January 7, 2014
3. "Interview: IBM's Linux chief tech officer Paul McKenney", Pat Pilcher at [http://m.nzherald.co.nz/business/news/article.cfm?c\\_id=3&objectid=10860641](http://m.nzherald.co.nz/business/news/article.cfm?c_id=3&objectid=10860641), January 22, 2013.
4. "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.
5. "Linux Kernel Developer Panel", LinuxCon Europe, Prague, Czech Republic, October 26, 2011 <https://events.linuxfoundation.org/events/linuxcon-europe/kernel-panel>.
6. "Why Is Parallel Programming So Hard", Rich Seidner at [http://intelligenceinsoftware.com/feature/expert\\_insight/why\\_is\\_parallel\\_programming\\_so\\_hard/](http://intelligenceinsoftware.com/feature/expert_insight/why_is_parallel_programming_so_hard/), August 2011.
7. "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.
8. "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>)

#### Ph.D. Committees

1. Iftekhar Ahmed, "Improving the Quality of Software Using Mutation Testing and Fault Prediction", Oregon State University, May 2018. Supervisor: Carlos Jensen.

2. Michael Sullivan, "Low-level Concurrent Programming Using the Relaxed Memory Calculus," Carnegie Mellon University, October 2017. Supervisor: Karl Crary.
3. Vincent Nimal, "Static Analyses over Weak Memory," Balliol College, University of Oxford, March 2015. Supervisors: Daniel Kroening and Joel Ouaknine.
4. Phil Howard, "Extending Relativistic Programming to Multiple Writes," Portland State University, February 2012. Major professor: Jonathan Walpole.
5. Josh Triplett, "Relativistic Causal Ordering: A Memory Model for Scalable Concurrent Data Structures," Portland State University, March 2012. Major professor: Jonathan Walpole.
6. Björn Brandenburg, "Scheduling and Locking in Multiprocessor Real-Time Operating Systems," University of North Carolina at Chapel Hill, August 2011. Major professor: James H. Anderson.

US Patents: See <http://www2.rdrop.com/users/paulmck/cv/PIH.2020.12.22a.html>

#### Professional Activities.

- Affiliate Professor, School of Electrical and Computer Science, Oregon State University, 2016.
- Inducted into the Oregon State University Academy of Distinguished Engineers, 2015.
- Attended invitation-only Linux Networking Summit (netconf), 2009.
- Program Chair for 2009, 2011-12, 2014-16, and 2018-20 Linux Plumbers Conference. Member of Program Committee for 2013 Linux Plumbers Conference.
- Maintainer for RCU and the RCU torture tests in the Linux kernel (2009-), reporting to Ingo Molnar.
- Lead maintainer for the Linux-kernel memory model (LKMM) (2018-), again reporting to Ingo Molnar.
- Attended invitation-only Linux Real-Time Summit, 2008-2014, and 2016.
- ISO SC22 WG14/WG21 standards committee (C/C++) as concurrency expert (2007-).
- Member, Oregon State University School of Electrical Engineering and Computer Science (EECS) Industrial Advisory Board, 2006?-.
- Attended invitation-only Linux Kernel Summit, 2003-2005, 2010-2016, and 2018.
  - Member of End-User Panel in 2003.
  - Presented on Real Time in 2005.
  - Member of program committee 2010-2016 and 2018.
  - Presented on scalability in 2010.
  - Lightning talk on RCU callback offloading in 2012.
  - Co-lead of 2013 Power-Aware Scheduling Mini-Summit with Morten Rasmussen.
  - Lead for 2013 Scaling Talks and Hacking Time.
- 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.

- October 2019-present: Facebook software engineer on the Kernel Team. Maintainer of RCU and LKMM in the Linux kernel.



- April 2000 – October 2019: IBM Distinguished Engineer and maintainer of RCU in the Linux kernel. Representative projects: “Scout” for OpenPower little-endian project. Open-source aspects of OpenPower and OpenDaylight Consortia. 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 Liason 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 liason 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 1995-2008. Collected donations for neighborhood fireworks display 1995-2012, 2014-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.)