

- [ME1] E. W. Dijkstra. Solution of a problem in concurrent programming control. *Communications of the ACM*, 8(9):569, 1965.
- [ME2] P. J. Courtois, F. Heymans, and D. L. Parnas. Concurrent control with “readers” and “writers”. *Communications of the ACM*, 14(10):667–668, 1971.
- [ME3] L. Lamport. A new solution of Dijkstra’s concurrent programming problem. *Communications of the ACM*, 17(8):453–455, 1974.
- [ME4] L. Lamport. Concurrent reading and writing. *Communications of the ACM*, 20(11):806–811, 1977.
- [ME5] C. A. R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, 1978.
- [ME6] H. P. Katseff. A new solution to the critical section problem. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, pages 86–88, San Diego, CA, 1978.
- [ME7] J. E. Burns. Mutual exclusion with linear waiting using binary shared variables. *SIGACT News*, 10(2):42–47, 1978.
- [ME8] K. R. Apt, N. Francez, and W. P. de Roever. A proof system for communicating sequential processes. *ACM Transactions on Programming Languages and Systems*, 2(3):359–385, 1980.
- [ME9] G. L. Peterson. Myths about the mutual exclusion problem. *Information Processing Letters*, 12(3):115–116, 1981.
- [ME10] J. E. Burns, P. Jackson, N. A. Lynch, M. J. Fischer, and G. L. Peterson. Data requirements for implementation of N-process mutual exclusion using a single shared variable. *Journal of the ACM*, 29(1):183–205, 1982.
- [ME11] M. Rabin. N-process mutual exclusion with bounded waiting by $4 \log_2 n$ -valued shared variable. *Journal of Computer and System Sciences*, 25(1):66–75, 1982.
- [ME12] G. L. Peterson. Concurrent reading while writing. *ACM Transactions on Programming Languages and Systems*, 5(1):46–55, 1983.
- [ME13] G. L. Peterson. A new solution to Lamport’s concurrent programming problem using small shared variables. *ACM Transactions on Programming Languages and Systems*, 5(1):56–65, 1983.
- [ME14] L. Lamport. The mutual exclusion problem: Part I – a theory of interprocess communication. *Journal of the ACM*, 33(2):313–326, 1986.
- [ME15] L. Lamport. The mutual exclusion problem: Part II – statement and solutions. *Journal of the ACM*, 33(2):327–348, 1986.
- [ME16] L. Lamport. A fast mutual exclusion algorithm. *ACM Transactions on Computer Systems*, 5(1):1–11, 1987.
- [ME17] L. Lamport. The mutual exclusion problem has been solved. *Communications of the ACM*, 34(1):110–111, 1991.
- [ME18] E. A. Lycklama and V. Hadzilacos. A first-come-first-served mutual-exclusion algorithm with small communication variables. *ACM Transactions on Programming Languages and Systems*, 13(4):558–576, 1991.

- [ME19] N. Lynch and N. Shavit. Timing based mutual exclusion. In *Proceedings of the 13th IEEE Real-time Systems Symposium*, pages 2–11, December 1992.
- [ME20] M. Merritt and G. Taubenfeld. Speeding lamport’s fast mutual exclusion algorithm. *Information Processing Letters*, 45(3):137–142, 1993.
- [ME21] J. E. Burns and N. A. Lynch. Bounds on shared memory for mutual exclusion. *Information and Computation*, 107(2):171–184, 1993.
- [ME22] M. Choy and A. K. Singh. Adaptive solutions to the mutual exclusion problem. *Distributed Computing*, 8(1):1–17, 1994.
- [ME23] R. Cypher. The communication requirements of mutual exclusion. In *Proceedings of the Seventh Annual ACM symposium on Parallel Algorithms and Architectures*, pages 147–156, Santa Barbara, CA, 1995.
- [ME24] X. Zhang, Y. Yan, and R. Castañeda. Evaluating and designing software mutual exclusion algorithms on shared-memory multiprocessors. *IEEE Parallel Distributed Technology: Systems & Technology*, 4(1):25–42, 1996.
- [ME25] S. S. Fu and N.-F. Tzeng. A circular list-based mutual exclusion scheme for large shared-memory multiprocessors. *IEEE Transactions on Parallel and Distributed Systems*, 8(6):628–639, 1997.
- [ME26] T.-L. Huang and C.-H. Shann. A comment on “A circular list-based mutual exclusion scheme for large shared-memory multiprocessors”. *IEEE Transactions on Parallel and Distributed Systems*, 9(4):414–416, 1998.
- [ME27] E. Kushilevitz, Y. Mansour, M. O. Rabin, and D. Zuckerman. Lower bounds for randomized mutual exclusion. *SIAM Journal of Computing*, 27(6):1550–1563, 1998.
- [ME28] E. Gafni and M. Mitzenmacher. Analysis of timing-based mutual exclusion with random times. In *Proceedings of the 18th Annual ACM Symposium on Principles of Distributed Computing*, pages 13–21, Atlanta, GA, 1999.
- [ME29] P. Keane and M. Moir. A simple local-spin group mutual exclusion algorithm. In *Proceedings of the 18th Annual ACM Symposium on Principles of Distributed Computing*, pages 23–32, Atlanta, GA, 1999.
- [ME30] Y.-J. Joung. Asynchronous group mutual exclusion. *Distributed Computing*, 13(4):189–206, 2000.
- [ME31] J. H. Anderson and Y.-J. Kim. Adaptive mutual exclusion with local spinning. In *Proceedings of the 14th International Conference on Distributed Computing*, pages 29–43, 2000.
- [ME32] J. H. Anderson and Y.-J. Kim. A new fast-path mechanism for mutual exclusion. *Distributed Computing*, 14(1):17–29, 2001.
- [ME33] J. H. Anderson and Y.-J. Kim. An improved lower bound for the time complexity of mutual exclusion. In *Proceedings of the 20th Annual ACM Symposium on Principles of Distributed Computing*, pages 90–99, Newport, RI, 2001.
- [ME34] Y.-J. Kim and J. H. Anderson. A time complexity bound for adaptive mutual exclusion. In *Proceedings of the 15th International Conference on Distributed Computing*, pages 1–15, 2001.
- [ME35] J. H. Anderson and Y.-J. Kim. Nonatomic mutual exclusion with local spinning. In *Proceedings of the 21st Annual ACM Symposium on Principles of Distributed Computing*, pages 3–12, Monterey, CA, 2002.

- [ME36] J. H. Anderson and Y.-J. Kim. Local-spin mutual exclusion using fetch-and- ϕ primitives. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, page 538, 2003.
- [ME37] J. H. Anderson, Y.-J. Kim, and T. Herman. Shared-memory mutual exclusion: Major research trends since 1986. *Distributed Computing*, 16(2-3):75–110, 2003.
- [ME38] M. Isard and A. Birrell. Automatic mutual exclusion. In *Proceedings of the 11th USENIX workshop on Hot topics in operating systems*, pages 1–6, San Diego, CA, 2007.

- [L1] J. Gray. Locking. In *Proceedings of the Woods Hole Conference on Concurrent Systems and Parallel Computation*, pages 169–176, 1970.
- [L2] J. Gray. Locking in decentralized computer systems. *IBM Journal of Research and Development*, RJ 1346:1–59, 1974.
- [L3] T. E. Anderson. The performance of spin lock alternatives for shared-memory alternatives. *IEEE Transactions on Parallel and Distributed Systems*, 1(1):6–16, 1990.
- [L4] T. Johnson and K. Harathi. A prioritized multiprocessor spin lock. *IEEE Transactions on Parallel and Distributed Systems*, 8(9):926–933, 1997.
- [L5] P. C. Diniz and M. C. Rinard. Lock coarsening: Eliminating lock overhead in automatically parallelized object-based programs. *Journal of Parallel and Distributed Computing*, 49(2):218–244, 1998.
- [L6] D. F. Bacon, R. Konuru, C. Murthy, and M. Serrano. Thin locks: Featherweight synchronization for Java. In *Proceedings of the SIGPLAN '98 Conference on Programming Language Design and Implementation*, pages 258–268, Montreal, Quebec, Canada, 1998.
- [L7] D. L. Detlefs, P. A. Martin, M. Moir, and Guy L. Steele. Lock-free reference counting. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 190–199, Newport, Rhode Island, 2001.
- [L8] M. L. Scott and W. N. Scherer III. Scalable queue-based spin locks with timeout. pages 44–52, Snowbird, UT, 2001.
- [L9] M. L. Scott. Non-blocking timeout in scalable queue-based spin locks. In *Proceedings of the 21st Annual ACM Symposium on Principles of Distributed Computing*, pages 31–40, Monterey, California, 2002.
- [L10] H. Franke, R. Russell, and M. Kirkwood. Fuss, futexes and furwocks: Fast userlevel locking in Linux. In *Proceedings of the Ottawa Linux Symposium*, http://www.linux.org.uk/~ajh/ols2002_proceedings.pdf.gz, pages 479–495, Ottawa, Canada, 2002.
- [L11] J. Regehr and A. Reid. Lock inference for systems software. In *Proceedings of the Second AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software*, Boston, MA, 2003.
- [L12] P. Ha-Hoai and P. Tsigas. Fast, reactive and lock-free multi-word compare-and-swap algorithms. Technical Report 2003-06, 2003.
- [L13] Z. Radovic and E. Hagersten. Hierarchical backoff locks for nonuniform communication architectures. In *Proceedings of the 9th International Symposium on High-Performance Computer Architecture*, pages 241–252, 2003.
- [L14] T. Ogasawara, H. Komatsu, and T. Nakatani. To-lock: Removing lock overhead using the owners' temporal locality. In *Proceedings of the 13th International Conference on Parallel Architectures and Compilation Techniques*, pages 255–266, 2004.
- [L15] V. Kahlon, F. Ivancic, and A. Gupta. Reasoning about threads communicating via locks. In *Proceedings of Conference on Computer Aided Verification*, pages 505–518, 2005.
- [L16] T. Harris and K. Fraser. Revocable locks for non-blocking programming. In *Proceedings of the 10th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 72–82, Chicago, IL, 2005.

- [L17] J. Rose, N. Swamy, and M. Hicks. Dynamic inference of polymorphic lock types. *Science of Computer Programming*, 58(3):366–383, 2005.
- [L18] U. Drepper. Futexes are Tricky. <http://people.redhat.com/drepper/futex.pdf>, December 2005.
- [L19] V. Luchangco, D. Nussbaum, and N. Shavit. A hierarchical CLH queue lock. In *Proceedings of Euro-Par*, pages 801–810, 2006.
- [L20] M. Hicks, J. S. Foster, and P. Prattikakis. Lock inference for atomic sections. In *Proceedings of the First ACM SIGPLAN Workshop on Languages, Compilers, and Hardware Support for Transactional Computing*, June 2006.
- [L21] M. Emmi, J. S. Fischer, R. Jhala, and R. Majumdar. Lock allocation. *Proceedings of the Thirty-fourth Annual ACM Symposium on the Principles of Programming Languages*, 42(1):291–296, 2007.

- [S1] H. T. Kunc. Synchronous and asynchronous parallel algorithms for multiprocessors. In *Algorithms and Complexity*, J. F. Traub, editor, pages 153–200. 1967.
- [S2] E. W. Dijkstra. Cooperating sequential processes. In *Programming Languages*, F. Genuys, editor, pages 43–112. Academic Press, 1968.
- [S3] V. E. Kotov and A. S. Narinyan. On transformation of sequential programs into asynchronous parallel programs. In *Proceedings of 1968 IFIP Congress*, pages 351–357, 1968.
- [S4] E. W. Dijkstra. Hierarchical ordering of sequential processes. *Acta Informatica*, 1:115–138, 1971.
- [S5] A. N. Habermann. Synchronization of communicating processes. *Communications of the ACM*, 15:171–176, March 1972.
- [S6] P. B. Hansen. Concurrent programming concepts. *ACM Computing Surveys*, 5:223–245, December 1973.
- [S7] R. J. Lipton. On synchronization primitive systems. Research Report 22, Yale Computer Science, 1973.
- [S8] R. J. Lipton. Limitations of synchronization primitives with conditional branching and global variables. In *Proceedings of the Sixth annual ACM Symposium on Theory of Computing*, pages 230–241, Seattle, WA, 1974.
- [S9] D. Dolev. Local characterization of models of synchronization primitives. In *Proceedings of the Waterloo Conference on Theoretical Computer Science*, pages 53–60, Boston, MA, 1977.
- [S10] N. N. Mirenkov. Process synchronization. *Cybernetics and Systems Analysis*, 15(1):66–72, 1979.
- [S11] P. B. Henderson and Y. Zalcstein. Synchronization problems solvable by generalized PV systems. *Journal of the ACM*, 27(1):60–71, 1980.
- [S12] J. T. Schwartz. Ultracomputers. *ACM Transactions on Programming Languages and Systems*, 2(4):484–521, 1980.
- [S13] A. Bernstein. Output guards and nondeterminism in “communicating sequential processes”. *ACM Transactions on Programming Languages and Systems*, 2:234–238, April 1980.
- [S14] P. E. Lauer. Synchronization of concurrent processes without globality assumptions. *SIGPLAN Notices*, 16:66–80, September 1981.
- [S15] A. Gottlieb and C. P. Kruskal. Coordinating parallel processors: a partial unification. *SIGARCH Computer Architecture News*, 9:16–24, October 1981.
- [S16] A. Gottlieb, R. Grishman, C. P. Kruskal, K. P. McAuliffe, L. Rudolph, and M. Snir. The NYU Ultracomputer—designing a MIMD, shared-memory parallel machine (extended abstract). In *Proceedings of the 9th Annual Symposium on Computer Architecture*, pages 27–42, Austin, TX, 1982.
- [S17] R. N. Taylor. Complexity of analyzing the synchronization structure of concurrent programs. *Acta Informatica*, 19:57–84, 1983.
- [S18] A. Gottlieb, R. Grishman, C. P. Kruskal, K. P. McAuliffe, L. Rudolph, and M. Snir. The NYU Ultrascomputer – designing an mimd parallel computer. *IEEE Transactions on Computers*, C-32:75–89, 1984.

- [S19] C. Zhu and P. Yew. A synchronization scheme and its applications for large scale multiprocessors. In *Proceedings of the Conference on Distributed Computing Systems*, pages 486–491, San Francisco, CA, May 1984.
- [S20] C. Whitby-Stevens. The Transputer. In *Proceedings of the 12th Annual International Symposium on Computer Architecture*, pages 292–300, Boston, MA, 1985.
- [S21] S. Midkiff and D. Padua. Compiler generated synchronization for DO loops. In *Proceedings of the 1986 International Conference on Parallel Processing*, pages 544–551, St. Charles, IL, August 1986.
- [S22] E. D. Brooks III. The butterfly barrier. *International Journal of Parallel Programming*, 15(4):295–307, 1986.
- [S23] P. Bitar and A. M. Despain. Multiprocessor cache synchronization: Issues, innovations, evolution. In *ISCA '86*, pages 424–433, 1986.
- [S24] C. P. Kruskal, L. Rudolph, and M. Snir. Efficient synchronization of multiprocessors with shared memory. In *Proceedings of the fifth annual ACM symposium on Principles of distributed computing*, pages 218–228, Calgary, Alberta, Canada, 1986.
- [S25] E. H. Jensen, G. W. Hagensen, and J. M. Broughton. A new approach to exclusive data access in shared memory multiprocessors. Technical Report UCRL-97663, Lawrence Livermore National Laboratory, 1987.
- [S26] S. Midkiff and D. Padua. Compiler algorithms for synchronization. *IEEE Transactions on Computers*, C-36(12):1485–1495, December 1987.
- [S27] D. Hensgen, R. Finkel, and U. Manber. Two algorithms for barrier synchronization. *International Journal of Parallel Programming*, 17(1):1–17, 1988.
- [S28] M. Wolfe. Multiprocessor synchronization for concurrent loops. *IEEE Software*, 5(1):34–42, 1988.
- [S29] M. P. Herlihy. Impossibility and universality results for wait-free synchronization. In *Proceedings of the seventh annual ACM Symposium on Principles of distributed computing*, pages 276–290, Toronto, Ontario, Canada, 1988.
- [S30] H. Dietz, T. Schwederski, M. O’Keefe, and A. Zaafrani. Static synchronization beyond VLIW. In *Proceedings of the 1989 ACM/IEEE conference on Supercomputing*, pages 416–425, Reno, Nevada, 1989.
- [S31] R. Gupta. The fuzzy barrier: A mechanism for high speed synchronization of processors. In *Proceedings of the Third International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-III)*, pages 54–63, Boston, MA, 1989.
- [S32] J. R. Goodman, M. K. Vernon, and P. J. Woest. Efficient synchronization primitives for large-scale cache-coherent multiprocessors. In *Proceedings of the Third International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-III)*, pages 64–75, Boston, MA, 1989.
- [S33] C. J. Beckmann and C. D. Polychronopoulos. Fast barrier synchronization hardware. In *Proceedings of the 1990 ACM/IEEE Conference on Supercomputing*, pages 180–189, 1990.
- [S34] A. Dinning. A survey of synchronization methods for parallel computers. *Computer*, 22:66–77, July 1989.
- [S35] R. Gupta and C. R. Hill. A scalable implementation of barrier synchronization using an adaptive combining tree. *International Journal of Parallel Programming*, 18:161–180, June 1990.

- [S36] D.-K. Chen, H.-M. Su, and P.-C. Yew. The impact of synchronization and granularity on parallel systems. In *Proceedings of the 17th International Symposium on Computer Architecture*, pages 239–248, Seattle, WA, 1990.
- [S37] J. Mellor-Crummey and M. L. Scott. Synchronization without contention. In *Proceedings of the Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IV)*, pages 269–278, Santa Clara, CA, 1991.
- [S38] M. Herlihy. Wait-free synchronization. *ACM Transactions on Programming Languages and Systems*, 13:124–149, January 1991.
- [S39] Z. Li. Compiler algorithms for event variable synchronization. In *Proceedings of the 5th international conference on Supercomputing*, pages 85–95, Cologne, West Germany, 1991.
- [S40] D. Kranz, B. H. Lim, and A. Agarwal. Low-cost support for fine-grain synchronization in multi-processors. Technical report, Cambridge, MA, 1992.
- [S41] J.-H. Yang and J. H. Anderson. Fast, scalable synchronization with minimal hardware support. In *Proceedings of the Twelfth ACM Symposium on Principles of Distributed Computing*, pages 171–182, 1993.
- [S42] L. Kontothanassis and R. Wisniewski. Using scheduler information to achieve optimal barrier synchronization. In *Proceedings of the Fourth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, San Diego, CA, May 1993.
- [S43] R. W. Wisniewski, L. Kontothanassis, and M. L. Scott. Scalable spin locks for multiprogrammed systems. Technical report, University of Rochester, 1993.
- [S44] J. M. Stone, H. S. Stone, P. Heidelberger, and J. Turek. Multiple reservations and the oklahoma update. *IEEE Parallel Distributed Technology*, 1:58–71, November 1993.
- [S45] S. Adve, A. L. Cox, H. Dwarkadas, and W. Zwaenepoel. Replacing locks by higher-level primitives. Technical report, Department of Computer Science, 1994.
- [S46] H. Attiya, N. Lynch, and N. Shavit. Are wait-free algorithms fast? *Journal of the ACM*, 41:725–763, July 1994.
- [S47] D.-K. Chen and P.-C. Yew. Redundant synchronization elimination for DOACROSS loops. In *Proceedings of the Eighth International Parallel Processing Symposium*, pages 477–481, Cancùn, Mexico, 1994.
- [S48] M. M. Michael and M. L. Scott. Implementation of atomic primitives on distributed shared memory multiprocessors. In *Proceedings of First Symposium on High Performance Computer Architecture*, pages 222–231, 1995.
- [S49] A. Krishnamurthy and K. Yelick. Optimizing parallel programs with explicit synchronization. In *Proceedings of the SIGPLAN '95 Conference on Programming Language Design and Implementation*, pages 196–204, La Jolla, CA, 1995.
- [S50] R. W. Wisniewski, L. I. Kontothanassis, and M. L. Scott. High performance synchronization algorithms for multiprogrammed multiprocessors. In *Proceedings of the Fifth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 199–206, Santa Barbara, CA, 1995.
- [S51] V. Ramakrishnan, I. D. Scherson, and R. Subramanian. Efficient techniques for fast nested barrier synchronization. In *Proceedings of the Seventh Annual ACM symposium on Parallel Algorithms and Architectures*, pages 157–164, Santa Barbara, CA, 1995.

- [S52] S. L. Scott. Synchronization and communication in the T3E multiprocessor. In *Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-VII)*, pages 26–36, Cambridge, Massachusetts, United States, 1996.
- [S53] P. Diniz and M. Rinard. Synchronization transformations for parallel computing. In *Proceedings of the Twenty-fourth Annual ACM Symposium on the Principles of Programming Languages*, pages 187–200, Paris, France, 1997.
- [S54] L. I. Kontothanassis, R. W. Wisniewski, and M. L. Scott. Scheduler-conscious synchronization. *ACM Transactions on Computer Systems*, 15(1):3–40, 1997.
- [S55] P. C. Diniz and M. C. Rinard. Dynamic feedback: an effective technique for adaptive computing. In *Proceedings of the SIGPLAN '97 Conference on Programming Language Design and Implementation*, pages 71–84, Las Vegas, NV, 1997.
- [S56] H. Han, C.-W. Tseng, and P. J. Keleher. Eliminating barrier synchronization for compiler-parallelized codes on software DSMs. *International Journal of Parallel Programming*, 26(5):591–612, 1998.
- [S57] A. Kägi, D. Burger, and J. R. Goodman. SOFTQOLB: An ultra-efficient synchronization primitive for clusters of commodity workstations. Technical Report 1327, University of Wisconsin-Madison, 1998.
- [S58] M. M. Michael and M. L. Scott. Nonblocking algorithms and preemption-safe locking on multiprogrammed shared memory multiprocessors. *Journal of Parallel and Distributed Computing*, 51:1–26, May 1998.
- [S59] D. S. Nikolopoulos and T. S. Papatheodorou. A quantitative architectural evaluation of synchronization algorithms and disciplines on ccNUMA systems: the case of the SGI Origin2000. In *Proceedings of the 13th international conference on Supercomputing*, pages 319–328, Rhodes, Greece, 1999.
- [S60] P. C. Diniz and M. C. Rinard. Synchronization transformations for parallel computing. *Concurrency – Practice and Experience*, 11(13):773–802, 1999.
- [S61] P. C. Diniz and M. C. Rinard. Eliminating synchronization overhead in automatically parallelized programs using dynamic feedback. *ACM Transactions on Computer Systems*, 17(2):89–132, 1999.
- [S62] M. Rinard and P. Diniz. Eliminating synchronization bottlenecks in object-based programs using adaptive replication. In *Proceedings of the 13th International Conference on Supercomputing*, pages 83–92, Rhodes, Greece, 1999.
- [S63] M. Rinard. Effective fine-grain synchronization for automatically parallelized programs using optimistic synchronization primitives. *ACM Transactions on Computer Systems*, 17(4):337–371, 1999.
- [S64] S. Kumar, D. Jiang, R. Chandra, and J. P. Singh. Evaluating synchronization on shared address space multiprocessors: Methodology and performance. In *Proceedings of the 1999 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 23–34, Atlanta, GA, 1999.
- [S65] B. Saglam and V. Mooney. System-on-a-chip processor synchronization support in hardware. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 633–641, Munich, Germany, 2001.

- [S66] K. Zee and M. Rinard. Write barrier removal by static analysis. In *Proceedings of the 17th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, pages 191–210, Seattle, WA, 2002.
- [S67] M. Herlihy, V. Luchangco, and M. Moir. Obstruction-free synchronization: Double-ended queues as an example. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, pages 522–, 2003.
- [S68] R. Thakur, W. D. Gropp, and B. R. Toonen. Minimizing synchronization overhead in the implementation of MPI one-sided communication. In *Parallel Virtual Machine / Message Passing Interface*, pages 57–67, 2004.
- [S69] J. Li, J. F. Martinez, and M. C. Huang. The thrifty barrier: Energy-aware synchronization in shared-memory multiprocessors. In *Proceedings of the 10th International Symposium on High Performance Computer Architecture*, pages 14–23, 2004.
- [S70] C. Flanagan and S. N. Freund. Automatic synchronization correction. In *Proceedings of Workshop on Synchronization and Concurrency in Object-Oriented Languages*, October 2005.
- [S71] Z. Fang, L. Zhang, J. B. Carter, L. Cheng, and M. Parker. Fast synchronization on shared-memory multiprocessors: An architectural approach. *Journal of Parallel and Distributed Computing*, 65:1158–1170, October 2005.
- [S72] L. Wang and S. D. Stoller. Static analysis of atomicity for programs with lock-free synchronization. Technical Report DAR-04-17, Department of Computer Science, SUNY at Stony Brook, January 2005.
- [S73] L. Wang and S. D. Stoller. Static analysis of atomicity for programs with non-blocking synchronization. In *Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming*, pages 61–71, Chicago, IL, 2005.
- [S74] V. K. Nandivada and D. Detlefs. Compile-time concurrent marking write barrier removal. In *Proceedings of the International Symposium on Code Generation and Optimization*, pages 37–48, 2005.
- [S75] A. Kejariwal, X. Tian, H. Saito, W. Li, M. Girkar, U. Banerjee, A. Nicolau, and C. D. Polychronopoulos. Lightweight lock-free synchronization methods for multithreading. In *Proceedings of the 20th ACM International Conference on Supercomputing*, pages 361–371, Cairns, Australia, 2006.
- [S76] H. Attiya, R. Guerraoui, D. Hendler, and P. Kouznetsov. Synchronizing without locks is inherently expensive. In *Proceedings of PODC*, pages 300–307, 2006.
- [S77] K. Fraser and T. Harris. Concurrent programming without locks. *ACM Transactions on Programming Languages and Systems*, 25, May 2007.
- [S78] Y. Zhang and E. Duesterwald. Barrier matching for programs with textually unaligned barriers. In *Proceedings of the 12th ACM SIGPLAN symposium on Principles and practice of parallel programming*, pages 194–204, San Jose, CA, 2007.
- [S79] W. Zhu, V. C Sreedhar, Z. Hu, and G. R. Gao. Synchronization state buffer: supporting efficient fine-grain synchronization on many-core architectures. In *Proceedings of the 34th International Symposium on Computer Architecture*, pages 35–45, San Diego, CA, 2007.
- [S80] A. Nicolau, G. Li, and A. Kejariwal. Techniques for efficient placement of synchronization primitives. In *Proceedings of the 14th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 199–208, Raleigh, NC, USA, February 2009.

- [S81] A. Nicolau, G. Li, A. V. Veidenbaum, and A. Kejariwal. Synchronization optimizations for efficient execution on multi-cores. In *Proceedings of the 23rd ACM International Conference on Supercomputing*, pages 169–180, New York, NY, 2009.
- [S82] H. Attiya and E. Hillel. Highly concurrent multi-word synchronization. *Theoretical Computer Science*, 412:1243–1262, March 2011.

- [CO1] R. Bayer and M. Schkolmck. Concurrency of operations on b-trees. *Acta Informatica*, 9:1–21, 1977.
- [CO2] C. S. Elhs. Concurrent search and inserts in 2-3 trees. *Acta Informatica*, 14(1):63–86, 1980.
- [CO3] H. T. Kung and Philip L. Lehman. Concurrent manipulation of binary search trees. *ACM Transactions on Database Systems*, 5(3):354–382, 1980.
- [CO4] P. L. Lehman and S. B. Yao. Efficient locking for concurrent operations on b-trees. *ACM Transactions on Database Systems*, 6(4):650–670, 1981.
- [CO5] U. Manber and R. E. Ladner. Concurrency control in a dynamic search structure. In *Proceedings of the First ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 268–282, 1982.
- [CO6] Y. S. Kwong and D. Wood. Method for concurrency m B-trees. *IEEE Transactions on Software Engineering*, SE-8(3):211–223, 1982.
- [CO7] Y. Mond and Y. Raz. Concurrency control in B+ trees using preparatory operations. In *Proceedings of the 11th International Conference on Very Large Data Bases*, pages 331–334, August 1985.
- [CO8] Y. Sagiv. Concurrent operations on B-trees with overtaking. In *Proceedings of the Fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 28–37, Portland, OR, 1985.
- [CO9] V. Lanin and D. Shasha. A symmetric concurrent B-tree algorithm. In *Proceedings of 1986 ACM Fall Joint Computer Conference*, pages 380–389, Dallas, TX, 1986.
- [CO10] M. Hsu and W.-P. Yang. Concurrent operations in extendible hashing. In *Proceedings of the 12th International Conference on Very Large Data Bases*, pages 241–247, 1986.
- [CO11] M. Herlihy and J. M. Wing. Axioms for concurrent objects. In *Proceedings of the Fourteenth Annual ACM Symposium on the Principles of Programming Languages*, pages 13–26, 1987.
- [CO12] V. Lanin and D. Shasha. Concurrent set manipulation without locking. In *Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 211–220, Austin, TX, 1988.
- [CO13] J. Mellor-Crummey. Concurrent queues: Practical fetch-and- ϕ algorithms. Technical Report 229, Computer Science Department, University of Rochester, 1987.
- [CO14] D. W. Jones. Concurrent operations on priority queues. *Communications of the ACM*, 32(1):132–137, 1989.
- [CO15] J. Aspnes and M. Herlihy. Wait-free data structures in the asynchronous PRAM model. In *Proceedings of the Second Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 340–349, 1990.
- [CO16] M. Herlihy. A methodology for implementing highly concurrent data structures. In *Proceedings of the Second ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 197–206, Seattle, WA, 1990.
- [CO17] M. P. Herlihy and J. M. Wing. Linearizability: a correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems*, 12(3):463–492, 1990.

- [CO18] M. Herlihy. Randomized wait-free concurrent objects (extended abstract). In *Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing*, pages 11–21, Montreal, Canada, 1991.
- [CO19] J. Mellor-Crummey and M. L. Scott. Scalable reader-writer synchronization for shared-memory multiprocessors. In *Proceedings of the Third ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 106–113, Williamsburg, VA, 1991.
- [CO20] J. Mellor-Crummey and M. L. Scott. Algorithms for scalable synchronization on shared-memory multiprocessors. *ACM Transactions on Computer Systems*, 9(1):21–65, 1991.
- [CO21] M. Herlihy. A methodology for implementing highly concurrent objects. *ACM Transactions on Programming Languages and Systems*, 15(5):745–770, 1993.
- [CO22] J. H. Anderson and M. Moir. Universal constructions for multi-object operations. In *Proceedings of the 14th Annual ACM Symposium on Principles of Distributed Computing*, pages 184–193, Ottawa, Ontario, Canada, 1995.
- [CO23] G. C. Hunt, M. M. Michael, S. Parthasarathy, and M. L. Scott. An efficient algorithm for concurrent priority queue heaps. *Information Processing Letters*, 60(3):151–157, 1996.
- [CO24] M. M. Michael and M. L. Scott. Simple, fast, and practical non-blocking and blocking concurrent queue algorithms. In *Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing*, pages 267–275, Philadelphia, PA, 1996.
- [CO25] J. H. Anderson and M. Moir. Universal constructions for large objects. *IEEE Transactions on Parallel and Distributed Systems*, 10(12):1317–1332, 1999.
- [CO26] N. Shavit and A. Zemach. Scalable concurrent priority queue algorithms. In *Proceedings of the 18th Annual ACM Symposium on Principles of Distributed Computing*, pages 113–122, Atlanta, GA, 1999.
- [CO27] T. L. Harris. A pragmatic implementation of non-blocking linked-lists. In *Proceedings of the 15th International Conference on Distributed Computing*, pages 300–314, 2001.
- [CO28] M. Raynal. Sequential consistency as lazy linearizability. In *Proceedings of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 151–152, Winnipeg, Manitoba, Canada, 2002.
- [CO29] M. Fomitchев and E. Ruppert. Lock-free linked lists and skip lists. In *Proceedings of the 23rd Annual ACM Symposium on Principles of Distributed Computing*, pages 50–59, St. John’s, Newfoundland, Canada, 2004.
- [CO30] W. N. Scherer III and M. L. Scott. Nonblocking concurrent data structures with condition synchronization. In *In Proc. of the Intl. Symp. on Distributed Computing*, pages 174–187, 2004.
- [CO31] M. Herlihy, V. Luchangco, P. Martin, and M. Moir. Nonblocking memory management support for dynamic-sized data structures. *ACM Transactions on Computer Systems*, 23:2005, 2005.
- [CO32] W. N. Scherer III, D. Lea, and M. L. Scott. Scalable synchronous queues. In *Proceedings of the 11th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 147–156, 2006.