References

1. F. Abolhassan, J. Keller, and W.J. Paul. On the Cost–Effectiveness of PRAMs. In *Proceedings of the 3rd IEEE Symposium on Parallel and Distributed Processing*, pages 2–9, 1991.
2. A. Adl-Tabatabai, C. Kozyrakis, and B. Saha. Unlocking concurrency. *ACM Queue*, 4(10): 24–33, Dec 2006.
3. S.V. Adve and K. Gharachorloo. Shared memory consistency models: A tutorial. *IEEE Computer*, 29: 66–76, 1995.
4. A. Aggarwal, A.K. Chandra, and M. Snir. On Communication Latency in PRAM Computations. In *Proceedings of 1989 ACM Symposium on Parallel Algorithms and Architectures* (*SPAA'89*), pages 11–21, 1989.
5. A. Aggarwal, A.K. Chandra, and M. Snir. Communication complexity of PRAMs. *Theoretical Computer Science*, 71: 3–28, 1990.
6. A. Aho, M. Lam, R. Sethi, and J. Ullman. *Compilers: Principles, Techniques & Tools*. Pearson-Addison Wesley, Boston, 2007.
7. A.V. Aho, M.S. Lam, R. Sethi, and J.D. Ullman. *Compilers: Principles, Techniques, and Tools*. 2nd Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, 2006.
8. S.G. Akl. *Parallel Computation – Models and Methods*. Prentice Hall, Upper Saddle River, 1997.
9. K. Al-Tawil and C.A. Moritz. LogGP Performance Evaluation of MPI. *International Symposium on High-Performance Distributed Computing,* page 366, 1998.
10. A. Alexandrov, M. Ionescu, K.E. Schauser, and C. Scheiman. LogGP: Incorporating Long Messages into the LogP Model – One Step Closer Towards a Realistic Model for Parallel Computation. In *Proceedings of the 7th ACM Symposium on Parallel Algorithms and Architectures* (*SPAA'95*), pages 95–105, Santa Barbara, July 1995.
11. E. Allen, D. Chase, J. Hallett, V. Luchangco, J.-W. Maessen, S. Ryu, G.L. Steele, Jr., and S. Tobin-Hochstadt. *The Fortress Language Specification, version 1.0 beta*, March 2007.
12. R. Allen and K. Kennedy. *Optimizing Compilers for Modern Architectures*. Morgan Kaufmann, San Francisco, 2002.
13. S. Allmann, T. Rauber, and G. Rünger. Cyclic Reduction on Distributed Shared Memory Machines. In *Proceedings of the 9th Euromicro Workshop on Parallel and Distributed Processing*, pages 290–297, Mantova, Italien, 2001. IEEE Computer Society Press.
14. G.S. Almasi and A. Gottlieb. *Highly Parallel Computing*. Benjamin Cummings, New York, 1994.
15. G. Amdahl. Validity of the Single Processor Approach to Achieving Large-Scale Computer Capabilities. In *AFIPS Conference Proceedings*, volume 30, pages 483–485, 1967.
16. K. Asanovic, R. Bodik, B.C. Catanzaro, J.J. Gebis, P. Husbands, K. Keutzer, D.A. Patterson, W.L. Plishker, J. Shalf, S.W. Williams, and K.A. Yelick. The Landscape of Parallel Computing Research: A View from Berkeley. Technical Report UCB/EECS-2006–183, EECS Department, University of California, Berkeley, December 2006.

DOI 10.1007/978-3-642-04818-0,

17. M. Azimi, N. Cherukuri, D.N. Jayasimha, A. Kumar, P. Kundu, S. Park, I. Schoinas, and A. Vaidya. Integration challenges and tradeoffs for tera-scale architectures. *Intel Technology Journal*, 11(03), 2007.
18. C.J. Beckmann and C. Polychronopoulos. Microarchitecture Support for Dynamic Scheduling of Acyclic Task Graphs. Technical Report CSRD 1207, University of Illinois, 1992.
19. D.P. Bertsekas and J.N. Tsitsiklis. *Parallel and Distributed Computation*. Athena Scientific, Nashua, 1997.
20. R. Bird. *Introduction to Functional Programming Using Haskell*. Prentice Hall, Englewood Cliffs, 1998.
21. P. Bishop and N. Warren. *JavaSpaces in Practice*. Addison Wesley, Reading, 2002.
22. F. Bodin, P. Beckmann, D.B. Gannon, S. Narayana, and S. Yang. Distributed C++: Basic Ideas for an Object Parallel Language. In *Proceedings of the Supercomputing'91 Conference*, pages 273–282, 1991.
23. D. Braess. *Finite Elements*. 3rd edition, Cambridge University Press, Cambridge, 2007.
24. P. Brucker. *Scheduling Algorithms*. 4th edition, Springer-Verlag, Berlin, 2004.
25. D.R. Butenhof. *Programming with POSIX Threads*. Addison-Wesley Longman Publishing Co., Inc., Boston, 1997.
26. N. Carriero and D. Gelernter. Linda in Context. *Communications of the ACM*, 32(4): 444–458, 1989.
27. N. Carriero and D. Gelernter. *How to Write Parallel Programs*. MIT Press, Cambridge, 1990.
28. P. Charles, C. Grothoff, V.A. Saraswat, C. Donawa, A. Kielstra, K. Ebcioglu, C. von Praun, and V. Sarkar. X10: An Object-Oriented Approach to Non-uniform Cluster Computing. In R. Johnson and R.P. Gabriel, editors, *Proceedings of the 20th Annual ACMSIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications* (*OOPSLA*), pages 519–538, ACM, October 2005.
29. A. Chin. Complexity Models for All-Purpose Parallel Computation. In *Lectures on Parallel Computation*, chapter 14. Cambridge University Press, Cambridge, 1993.
30. M.E. Conway. A Multiprocessor System Design. In *Proceedings of the AFIPS 1963 Fall Joint Computer Conference*, volume 24, pages 139–146, Spartan Books, New York, 963.
31. T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, 2001.
32. D.E. Culler, A.C. Arpaci-Dusseau, S.C. Goldstein, A. Krishnamurthy, S. Lumetta, T. van Eicken, and K.A. Yelick. Parallel Programming in Split-C. In *Proceedings of Supercomputing*, pages 262–273, 1993.
33. D.E. Culler, A.C. Dusseau, R.P. Martin, and K.E. Schauser. Fast Parallel Sorting Under LogP: From Theory to Practice. In *Portability and Performance for Parallel Processing*, pages 71–98. Wiley, Southampton, 1994.
34. D.E. Culler, R. Karp, A. Sahay, K.E. Schauser, E. Santos, R. Subramonian, and T. von Eicken. LogP: Towards a Realistic Model of Parallel Computation. *Proceedings of the 4th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (*PPoPP'93*), pages 1–12, 1993.
35. D.E. Culler, J.P. Singh, and A. Gupta. *Parallel Computer Architecture: A Hardware Software Approach*. Morgan Kaufmann, San Francisco, 1999.
36. H.J. Curnov and B.A. Wichmann. A Synthetic Benchmark. *The Computer Journal*, 19(1): 43–49, 1976.
37. D. Callahan, B.L. Chamberlain, and H.P. Zima. The Cascade High Productivity Language. In *IPDPS*, pages 52–60. IEEE Computer Society, 2004.
38. W.J. Dally and C.L. Seitz. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. *IEEE Transactions on Computers*, 36(5): 547–553, 1987.
39. DEC. The Whetstone Performance. Technical Report, Digital Equipment Corporation, 1986.
40. E.W. Dijkstra. Cooperating Sequential Processes. In F. Genuys, editor, *Programming Languages*, pages 43–112. Academic Press Inc., 1968.

41. J. Dongarra. Performance of various Computers using Standard Linear Equations Software in Fortran Environment. Technical Report CS-89–85, Computer Science Department, University of Tennessee, Knoxville, 1990.
42. J. Dongarra and W. Gentzsch, editors. *Computer Benchmarks*. Elsevier, North Holland, 1993.
43. J.J. Dongarra, I.S. Duff, D.C. Sorenson, and H.A. van der Vorst. *Solving Linear Systems on Vector and Shared Memory Computers*. SIAM, Philadelphia, 1993.
44. J. Duato, S. Yalamanchili, and L. Ni. *Interconnection Networks – An Engineering Approach*. Morgan Kaufmann, San Francisco, 2003.
45. M. Dubois, C. Scheurich, and F. Briggs. Memory Access Buffering in Multiprocessors. In *Proceedings of the 13th International Symposium on Computer Architecture* (*ISCA'86*), pages 434–442, ACM, 1986.
46. J. Dümmler, T. Rauber, and G. Rünger. Mixed Programming Models using Parallel Tasks. In J. Dongarra, C.-H. Hsu, K.-C. Li, L.T. Yang, and H. Zima, editors, *Handbook of Research on Scalable Computing Technologies*. Information Science Reference, July 2009.
47. T. El-Ghazawi, W. Carlson, T. Sterling, and K. Yelick. *UPC: Distributed Shared Memory Programming*. Wiley, New York, 2005.
48. J.R. Ellis. *Bulldog: A Compiler for VLIW Architectures*. MIT Press, Cambridge, MA, USA, 1986.
49. T. Ellis, I. Phillips, and T. Lahey. *Fortran90 Programming*. Addison-Wesley, Wokingham, 1994.
50. J.T. Feo. An analysis of the computational and parallel complexity of the Livermore loops. *Parallel Computing*, 7: 163–185, 1988.
51. D. Flanagan. *Java in a Nutshell*. O'Reilly, Sebastopol, 2005.
52. M.J. Flynn. Some computer organizations and their effectiveness. *IEEE Transactions on Computers*, 21(9): 948–960, 1972.
53. S. Fortune and J. Wyllie. Parallelism in Random Access Machines. In *Proceedings of the 10th ACM Symposium on Theory of Computing*, pages 114–118, 1978.
54. High Performance Fortran Forum. High performance Fortran language specification. *Scientific Programming*, 2(1): 1–165, 1993.
55. Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard, Version 1.3*. www.mpi-forum.org, 2008.
56. Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard, Version 2.1*. www.mpi-forum.org, 2008.
57. I. Foster. *Designing and Building Parallel Programs*. Addison-Wesley, Reading, 1995.
58. I. Foster. Compositional parallel programming languages. *ACM Transactions on Programming Languages and Systems*, 18(4): 454–476, 1996.
59. I. Foster. Globus Toolkit Version 4: Software for Service-Oriented Systems. In *Proceedings of the IFIP International Conference on Network and Parallel Computing*, pages 2–13, Springer LNCS 3779, 2006.
60. T.L. Freeman and C. Phillips. *Parallel Numerical Algorithms*. Prentice Hall, Upper Saddle River, 1992.
61. A. Frommer. *Lösung linearer Gleichungssysteme auf Parallelrechnern*. Vieweg, Braunschweig, 1990.
62. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.
63. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM Parallel Virtual Machine: A User's Guide and Tutorial for Networked Parallel Computing*. MIT Press, Cambridge, 1996. Web page: www.netlib.org/pvm3/book/pvm book.html.
64. A. George, J. Liu, and E. Ng. User's Guide for SPARSPAK: Waterloo Sparse Linear Equations Package. Technical Report CS-78–30, Department of Computer Science, University of Waterloo, 1980.
65. A. George and J.W.-H. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. PrenticeHall, Englewood Cliffs, 1981.

66. P.B. Gibbons. A More Practical PRAM Model. In *Proceedings of the 1989 ACM Symposium on Parallel Algorithms and Architectures* (*SPAA'89*), pages 158–168, 1989.
67. M.B. Girkar and C. Polychronopoulos. Automatic extraction of functional parallelism from ordinary programs. *IEEE Transactions on Parallel and Distributed Systems*, 3(2): 166–178, 1992.
68. C.J. Glass and L.M. Li. The Turn Model for Adaptive Routing. In *Proceedings of the 19th International Symposium on Computer Architecture* (*ISCA'92*), pages 278–287, ACM, 1992.
69. S. Goedecker and A. Hoisie. *Performance Optimization of Numerically Intensive Codes*. SIAM, Philadelphia, 2001.
70. B. Goetz. *Java Concurrency in Practice*. Addison Wesley, Reading, 2006.
71. G. Golub and Ch. Van Loan. *Matrix Computations*. 3rd edition, The Johns Hopkins University Press, Baltimore, 1996.
72. G. Golub and J. Ortega. *Scientific Computing*. Academic Press, Boston, 1993.
73. A. Gottlieb, R. Grishman, C. Kruskal, K. McAuliffe, L. Rudolph, and M. Snir. The NYU ultracomputer – designing an MIMD shared memory parallel computer. *IEEE Transactions on Computers*, 32(2): 175–189, February 1983.
74. M.W. Goudreau, J.M. Hill, K. Lang, W.F. McColl, S.D. Rao, D.C. Stefanescu, T. Suel, and T. Tsantilas. A proposal for a BSP Worldwide standard. Technical Report, BSP Worldwide, www.bsp-worldwide.org, 1996.
75. A. Grame, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Programming*. Addison Wesley, Reading, 2003.
76. T. Grün, T. Rauber, and J. Röhrig. Support for efficient programming on the SB-PRAM. *International Journal of Parallel Programming*, 26(3): 209–240, 1998.
77. A. Gupta, G. Karypis, and V. Kumar. Highly scalable parallel algorithms for sparse matrix factorization. *IEEE Transactions on Parallel and Distributed Systems*, 8(5): 502–520, 1997.
78. J.L. Gustafson. Reevaluating Amdahl's law. *Communications of the ACM*, 31(5): 532–533, 1988.
79. W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. Springer, New York, 1994.
80. K. Hammond and G. Michaelson, editors. *Research Directions in Parallel Functional Programming*. Springer-Verlag, Springer, 1999.
81. J. Handy. *The Cache Memory Book*. 2nd edition, Academic Press, San Diego, 1998.
82. P.J. Hatcher and M.J. Quinn. *Data-Parallel Programming*. MIT Press, Cambridge, 1991.
83. J. Held, J. Bautista, and S. Koehl. From a Few Cores to Many – A Tera-Scale Computing Research Overview. Intel White Paper, Intel, 2006.
84. J.L. Hennessy and D.A. Patterson. *Computer Architecture – A Quantitative Approach*. 4th edition, Morgan Kaufmann, Boston, 2007.
85. M. Herlihy and J.E.B. Moss. Transactional Memory: Architectural Support for Lock-Free Data Structures. In *Proceedings of the 20th Annual International Symposium on Computer Architecture* (*ISCA'93*), pages 289–300, 1993.
86. M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49: 409–436, 1952.
87. T. Heywood and S. Ranka. A practical hierarchical model of parallel computation. *Journal of Parallel and Distributed Computing*, 16: 212–249, 1992.
88. J.M.D. Hill, B. McColl, D.C. Stefanescu, M.W. Goudreau, K. Lang, S.B. Rao, T. Suel, T. Tsantilas, and R. Bisseling. BSPlib The BSB Programming Library. Technical Report TR-29–97, Oxford University, May 1997.
89. M. Hill, W. McColl, and D. Skillicorn. Questions and answers about BSP. *Scientific Programming*, 6(3): 249–274, 1997.
90. C.A.R. Hoare. Monitors: An operating systems structuring concept. *Communications of the ACM*, 17(10): 549–557, 1974.
91. R. Hockney. A fast direct solution of Poisson's equation using Fourier analysis. *Journal of the ACM*, 12: 95–113, 1965.

92. R.W. Hockney. *The Science of Computer Benchmarking*. SIAM, Philadelphia, 1996.
93. R. Hoffmann and T. Rauber. Fine-Grained Task Scheduling using Adaptive Data Structures. In *Proceedings of the of Euro-Par*, volume 5168 of Lecture Notes in Computer Science, pages 253–262, Springer, 2008.
94. P. Hudak and J. Fasel. A gentle introduction to Haskell. *ACM SIGPLAN Notices*, 27(5): May 1992.
95. K. Hwang. *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. McGraw-Hill, New York, 1993.
96. F. Ino, N. Fujimoto, and K. Hagihara. LogGPS: A Parallel Computational Model for Synchronization Analysis. In *PPoPP '01: Proceedings of the Eighth ACM SIGPLAN Symposium on Principles and Practices of Parallel Programming*, pages 133–142, ACM, New York, 2001.
97. J.D. Jackson. *Classical Electrodynamics*. 3rd edition, Wiley, New York and Chichester, 1998.
98. J. Jájá. *An Introduction to Parallel Algorithms*. Addison-Wesley, New York, 1992.
99. M. Johnson. *Superscalar Microprocessor Design*. Prentice Hall, Englewood Cliffs, 1991.
100. S. Johnsson and C. Ho. Optimum broadcasting and personalized communication in hypercubes. *IEEE Transactions on Computers*, 38(9): 1249–1268, 1989.
101. J. Keller, C.W. Keßler, and J.L. Träff. *Practical PRAM Programming*. Wiley, New York, 2001.
102. J. Keller, T. Rauber, and B. Rederlechner. Conservative Circuit Simulation on Shared–Memory Multiprocessors. In *Proceedings of the 10th Workshop on Parallel and Distributed Simulation* (*PADS'96*), pages 126–134, ACM, 1996.
103. K. Kennedy, C. Koelbel, and H. Zima. The Rise and Fall of High Performance Fortran: An Historical Object Lesson. In *HOPL III: Proceedings of the Third ACM SIGPLAN Conference on History of Programming Languages*, pages 7–1–7–22, ACM, New York, 2007.
104. T. Kielmann, H.E. Bal, and K. Verstoep. Fast Measurement of LogP Parameters for Message Passing Platforms. In *IPDPS '00: Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing*, pages 1176–1183, Springer, London, 2000.
105. S. Kleiman, D. Shah, and B. Smaalders. *Programming with Threads*. Prentice Hall, Englewood Cliffs, 1996.
106. G. Koch. Discovering Multi-core: Extending the Benefits of Moore's Law. Intel White Paper, Technology@Intel Magazine, 2005.
107. P.M. Kogge. An Exploitation of the Technology Space for Multi-Core Memory/Logic Chips for Highly Scalable Parallel Systems. In *Proceedings of the Innovative Architecture for Future Generation High-Performance Processors and Systems*, IEEE, 2005.
108. M. Korch and T. Rauber. A comparison of task pools for dynamic load balancing of irregular algorithms. *Concurrency and Computation: Practice and Experience*, 16: 1–47, January 2004.
109. D. Kuck. Platform 2015 Software-Enabling Innovation in Parallelism for the Next Decade. Intel White Paper, Technology@Intel Magazine, 2005.
110. J. Kurose and K. Ross. *Computer Networking, 3. Auflage*. Addison Wesley, Wokingham, 2005.
111. L. Lamport. How to make a multiprocessor computer that correctly executes multiprocess programs. *IEEE Transactions on Computers*, 28(9): 690–691, September 1979.
112. J.R. Laurs and R. Rajwar. *Transactional Memory*. Morgan & Claypool Publishers, San Rafael, 2007.
113. D. Lea. *Concurrent Programming in Java: Design Principles and Patterns*. Addison Wesley, Reading, 1999.
114. E.A. Lee. The problem with threads. *IEEE Computer*, 39(5): 33–42, 2006.
115. F.T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann, San Mateo, 1992.
116. D.E. Lenoski and W.Weber. *Scalable Shared-Memory Multiprocessing*. Morgan Kaufmann, San Francisco, 1995.

117. B. Lewis and D.J. Berg. *Multithreaded Programming with Pthreads*. Prentice Hall, New Jersey, 1998.
118. J.W.H. Liu. The role of elimination trees in sparse factorization. *The SIAM Journal on Matrix Analysis and Applications*, 11: 134–172, 1990.
119. D.T. Marr, F. Binus, D.L. Hill, G. Hinton, D.A. Konfaty, J.A. Miller, and M. Upton. Hyper-threading technology architecture and microarchitecture. *Intel Technology Journal*, 6(1): 4–15, 2002.
120. T. Mattson, B. Sandor, and B. Massingill. *Pattern for Parallel Programming*. Pearson – Addison Wesley, Reading, 2005.
121. F. McMahon. The Livermore Fortran Kernels: A Computer Test of the Numerical Performance Range. Technical Report UCRL-53745, Lawrence Livermore National Laboratory, Livermore, 1986.
122. M. Metcalf and J. Reid. *Fortran 90/95 Explained*. Oxford University Press, Oxford, 2002.
123. R. Miller and L. Boxer. *Algorithms Sequential and Parallel*. Prentice Hall, Upper Saddle River, 2000.
124. E.G. Ng and B.W. Peyton. A Supernodal Cholesky Factorization Algorithm for Shared-Memory Multiprocessors. Technical Report, Oak Ridge National Laboratory, 1991.
125. L.M. Ni and P.K. McKinley. A survey of wormhole routing techniques in direct networks. *IEEE Computer*, 26: 62–76, February 1993.
126. B. Nichols, D. Buttlar, and J. Proulx Farrell. *Pthreads Programming*. O'Reilly & Associates, Sebastopol, 1997.
127. J. Nieplocha, J. Ju, M.K. Krishnan, B. Palmer, and V. Tipparaju. The Global Arrays User's Manual. Technical Report PNNL-13130, Pacific Northwest National Laboratory, 2002.
128. Nvidia. NVIDIA GeForce 8800 GPU Architecture Overview. Technical Report TB-02787–001 v01, Nvidia, 2006.
129. S. Oaks and H. Wong. *Java Threads. 3. Auflage*. O'Reilly, Sebastopol, 2004.
130. *OpenMP C and C++ Application Program Interface, Version 1.0*. www.openmp.org, October 1998.
131. *OpenMP Application Program Interface, Version 2.5*. www.openmp.org, May 2005.
132. *OpenMP Application Program Interface, Version 3.0*. www.openmp.org, May 2008.
133. J.M. Ortega. *Introduction to Parallel and Vector Solutions of Linear Systems*. Plenum Publishing Corp., New York, 1988.
134. J.M. Ortega and R.G. Voigt. *Solution of Partial Differential Equations on Vector and Parallel Computers*. SIAM, Philadelphia, 1985.
135. P.S. Pacheco. *Parallel Programming with MPI*. Morgan Kaufmann, San Francisco, 1997.
136. C.H. Papadimitriou and M. Yannakakis. Towards an Architecture-Independent Analysis of Parallel Algorithms. In *Proceedings of the 20th ACM Symposium on Theory of Computing*, pages 510–513, 1988.
137. D.A. Patterson and J.L. Hennessy. *Computer Organization & Design – The Hardware/Software Interface*. 4th edition, Morgan Kaufmann, San Francisco, 2008.
138. S. Pelegatti. *Structured Development of Parallel Programs*. Taylor and Francis, London, 1998.
139. L. Peterson and B. Davie. *Computer Networks – A Systems Approach*, *3. Auflage*. Morgan Kaufmann, Los Altos, 2003.
140. G.F. Pfister. *In Search of Clusters*. 2nd edition, Prentice Hall, Upper Saddle River, 1998.
141. A. Podehl, T. Rauber, and G. Rünger. A shared-memory implementation of the hierarchical radiosity method. *Theoretical Computer Science*, 196(1–2): 215–240, 1998.
142. C.D. Polychronopoulos. *Parallel Programming and Compilers*. Kluwer Academic Publishers, Norwell, 1988.
143. S. Prasad. *Multithreading Programming Techniques*. McGraw-Hill, New York, 1997.
144. R. Rajwar and J. Goodman. Transactional execution: Towards reliable, high-performance multithreading. *IEEE Micro*, 23(6): 117–125, 2003.

145. S. Ramaswamy, S. Sapatnekar, and P. Banerjee. A framework for exploiting task and data parallelism on distributed-memory multicomputers. *IEEE Transactions on Parallel and Distributed Systems*, 8(11): 1098–1116, 1997.
146. T. Rauber and G. Rünger. A transformation approach to derive efficient parallel implementations. *IEEE Transactions on Software Engineering*, 26(4): 315–339, 2000.
147. T. Rauber and G. Rünger. Deriving array distributions by optimization techniques. *Journal of Supercomputing*, 15: 271–293, 2000.
148. T. Rauber and G. Rünger. Tlib – A library to support programming with hierarchical multiprocessor tasks. *Journal of Parallel and Distributed Computing*, 65(3): 347–360, 2005.
149. T. Rauber, G. Rünger, and C. Scholtes. Execution behavior analysis and performance prediction for a shared-memory implementation of an irregular particle simulation method. *Simulation: Practice and Theory*, 6: 665–687, 1998.
150. J.K. Reid. On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations. In *Large Sparse Sets of Linear Equations*, pages 231–254. Academic Press, New York, 1971.
151. M. Rosing, R.B. Schnabel, and R.P. Waever. The DINO Parallel Programming Language. Technical Report CU-CS-501–90, Computer Science Dept., University of Colorado at Boulder, Boulder, 1990.
152. E. Rothberg and A. Gupta. An evaluation of left-looking, right-looking and multifrontal approaches to sparse Cholesky factorization on hierarchical-memory machines. *International Journal of High Speed Computing*, 5(4): 537–593, 1993.
153. G. Rünger. Parallel Programming Models for Irregular Algorithms. In *Parallel Algorithms and Cluster Computing*, pages 3–23. Springer Lecture Notes in Computational Science and Engineering, 2006.
154. Y. Saad. *Iterative Methods for Sparse Linear Systems*. International Thomson Publ., London, 1996.
155. Y. Saad. Krylov subspace methods on supercomputers. *SIAM Journal on Scientific and Statistical Computing*, 10: 1200–1332, 1998.
156. J. Savage. *Models of Computation: Exploring the Power of Computing*. Addison-Wesley Longman Publishing Co., Inc., Boston, 1997.
157. C. Scheurich and M. Dubois. Correct Memory Operation of Cache-Based Multiprocessors. In *Proceedings of the 14th International Symposium on Computer Architecture* (*ISCA'87*), pages 234–243, ACM, 1987.
158. D. Sima, T. Fountain, and P. Kacsuk. *Advanced Computer Architectures*. Addison-Wesley, Harlow, 1997.
159. J.P. Singh. *Parallel Hierarchical N-Body Methods and Their Implication for Multiprocessors*. PhD Thesis, Stanford University, 1993.
160. D. Skillicorn and D. Talia. Models and languages for parallel computation. *ACM Computing Surveys*, 30(2): 123–169, 1998.
161. B. Smith. Architecture and applications on the HEP multiprocessor computer systems. *SPIE* (*Real Time Signal Processing IV*), 298: 241–248, 1981.
162. M. Snir, S. Otto, S. Huss-Ledermann, D. Walker, and J. Dongarra. *MPI: The Complete Reference*. MIT Press, Cambridge, 1996. Web page: www.netlib.org/utk/papers/mpi book/mpi book.html.
163. M. Snir, S. Otto, S. Huss-Ledermann, D. Walker, and J. Dongarra. *MPI: The Complete Reference, Vol. 1: The MPI Core*. MIT Press, Cambridge, 1998. Web page: mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=4579.
164. W. Stallings. *Computer Organization and Architecture*. 7th edition, Prentice Hall, Upper Saddle River, 2009.
165. R.C. Steinke and G.J. Nutt. A unified theory of shared memory consistency. *Journal of the ACM*, 51(5): 800–849, 2004.
166. J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer, New York, 2002.

167. H.S. Stone. Parallel processing with the perfect shuffle. *IEEE Transactions on Computers*, 20(2): 153–161, 1971.
168. H.S. Stone. An efficient parallel algorithm for the solution of a tridiagonal linear system of equations. *Journal of the ACM*, 20: 27–38, 1973.
169. H. Sutter and J. Larus. Software and the concurrency revolution. *ACM Queue*, 3(7): 54–62, 2005.
170. S. Thompson. *Haskell – The Craft of Functional Programming*. Addison-Wesley, Reading, 1999.
171. L.G. Valiant. A bridging model for parallel computation. *Communications of the ACM*, 33(8): 103–111, 1990.
172. L.G. Valiant. A Bridging Model for Multi-core Computing. In *Proceedings of the ESA*, volume 5193, pages 13–28, Springer LNCS, 2008.
173. E.F. van de Velde. *Concurrent Scientific Computing*. Springer, New York, 1994.
174. R.P. Weicker. Dhrystone: A synthetic system programming benchmark. *Communications of the ACM*, 29(10): 1013–1030, 1984.
175. M. Wolfe. *High Performance Compilers for Parallel Computing*. Addison-Wesley, Redwood City, 1996.
176. Xelerated. Xelerator X11 Network Processor. Technical report, Xelerated, www.xelerated.com, accessed 2009.
177. M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 2: 77–79, 1991.
178. S.N. Zheltov and S.V. Bratanov. Measuring HT-Enabled Multi-Core: Advantages of a Thread-Oriented Approach. *Technology & Intel Magazine*, December 2005.
179. A.Y.H. Zomaya, editor. *Parallel & Distributed Computing Handbook*. Computer Engineering Series. McGraw-Hill, New York, 1996.

Index