

References

- [AAB⁺00] B. Alpern, C. R. Attanasio, J. J. Barton, M. G. Burke, P. Cheng, J.-D. Choi, A. Cocchi, S. J. Fink, D. Grove, M. Hind, S. Flynn-Hummel, D. Lieber, V. Litvinov, M. F. Mergen, T. Ngo, J. R. Russell, V. Sarkar, M. J. Serrano, J. C. Shepherd, S. E. Smith, V. C. Sreedhar, H. Srinivasan, and J. Whaley. The Jalapeno virtual machine. *IBM Systems Journal*, 39(1):211–238, 2000.
- [AALT95] P. Amarasinghe, J. M. Anderson, M. S. Lam, and C. W. Tseng. An overview of the SUIF compiler for scalable parallel machines. In *Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*, San Francisco, CA, October 1995.
- [ACC⁺90] R. Alverson, D. Callahan, D. Cummings, B. Koblenz, A. Porterfield, and B. Smith. The Tera computer system. In *Proceedings of the ACM International Conference on Supercomputing*, pp. 1–6, Amsterdam, The Netherlands, 1990.
- [ACC⁺99] B. Alpern, M. Charney, J.-D. Choi, A. Cocchi, and D. Lieber. Dynamic linking on a shared-memory multiprocessor. In *Proceedings of the 1999 International Conference on Parallel Architectures and Compilation Techniques (PACT'99)*, pp. 177–182, June 1999.
- [ACL⁺98] A. Adl-Tabatabai, M. Cierniak, G.-Y. Lueh, V. M. Parikh, and J. M. Stichnoth. Fast, effective code generation in a just-in-time Java compiler. In *Proceedings of the ACM SIGPLAN '98 Conference on Programming Language Design and Implementation*, pp. 280–290, June 1998.
- [ACL⁺99] B. Alpern, A. Cocchi, D. Lieber, M. Mergen, and V. Sarkar. Jalapeno - a compiler-supported Java virtual machine for servers. In *Workshop on Compiler Support for Software System (WCSSS 99)*, May 1999.

- [AD98] H. Akkary and M. Driscoll. A dynamic multithreaded processor. In *Proceedings of the 31st Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 226–236, Dallas, TX, December 1998.
- [AEL88] A. W. Appel, J. R. Ellis, and K. Li. Real-time concurrent collection on stock multiprocessors. In *Proceedings of the ACM SIGPLAN '98 Conference on Programming Language Design and Implementation*, pp. 11–20, June 1988.
- [AFG⁺00] M. Arnold, S. Fink, D. Grove, M. Hind, and P. Sweeney. Adaptive optimization in the Jalapeno JVM. In *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming and Systems, Languages, and Applications (OOPSLA) 2000*, pp. 47–65, October 2000.
- [AGH00] K. Arnold, J. Gosling, and D. Holmes. *The Java Programming Language*. Addison-Wesley, third edition, 2000.
- [aJ-100] aJile Systems, Inc. *aJ-100 Reference Manual, Version 2.0*, November 2000. <http://www.ajile.com/downloads/aJ-100ReferenceManual.pdf>.
- [AK87] J. R. Allen and K. Kennedy. Automatic translation of Fortran programs to vector form. *ACM Transactions on Programming Languages and Systems*, 9(4):491–542, October 1987.
- [AP88] D. Antonioli and M. Pilz. Analysis of the Java class file format. Technical Report 98.4, Dept. of Computer Science, University of Zurich, Switzerland, April 1988.
- [Arm98] E. Armstrong. Hotspot: A new breed of virtual machine. *Java World*, March 1998. <http://www.javaworld.com/javaworld/jw-03-1998/jw-03-hotspot.html>.
- [BA97] D. C. Burger and T. M. Austin. The SimpleScalar tool set, version 2.0. Technical Report CS-TR-97-1342, Dept. of Computer Science, Univ. of Wisconsin-Madison, 1997.
- [Bak78] H. G. Baker. List processing in real time on a serial computer. *Communications of the ACM*, 21(4):280–294, April 1978.
- [BAL⁺01] D. F. Bacon, C. R. Attanasio, H. B. Lee, V. T. Rajan, and S. Smith. Java without the coffee breaks: A nonintrusive multiprocessor garbage collector. In *Proceedings of the ACM SIGPLAN 2001 Conference on Programming Language Design and Implementation*, pp. 92–103, June 2001.

- [BBBG99] A. Barisone, F. Belliotti, R. Berta, and A. Gloria. UltraSPARC instruction level characterization of Java virtual machine workload. In *2nd Annual Workshop on Workload Characterization (WWC) for Computer System Design*, pp. 1–24. Kluwer Academic Publishers, 1999.
- [BBD⁺00] G. Bollella, B. Brosgol, P. Dibble, S. Furr, J. Gosling, D. Hardin, and M. Turnbull. *The Real-Time Specification for Java*. Addison-Wesley, June 2000. Further information is available at <http://www.rti.org>.
- [BCF⁺99] M. G. Burke, J.-D. Choi, S. Fink, D. Grove, M. Hind, V. Sarkar, M. J. Serrano, V. C. Sreedhar, H. Srinivasan, and J. Whaley. The Jalapeno dynamic optimizing compiler for Java. In *Proceedings of the ACM 1999 Java Grande Conference*, pp. 129–141, June 1999.
- [BDF⁺00] S. J. Baylor, M. Devarakonda, S. Fink, E. Gluzberg, M. Kalantar, P. Muttineni, E. Barsness, R. Arora, R. Dimpsey, and S. J. Munroe. Java server benchmarks. *IBM Systems Journal*, 39(1):57–81, 2000.
- [BDS91] H. J. Boehm, A. Demers, and S. Shenker. Mostly parallel garbage collection. In *Proceedings of the ACM SIGPLAN '91 Conference on Programming Language Design and Implementation*, pp. 257–264, June 1991.
- [Ber80] V. Berstis. Security and protection of data in the IBM System/38. In *Proceedings of the 7th Annual International Symposium on Computer Architecture*, pp. 245–252, 1980.
- [BG97a] A. Bik and D. Gannon. Automatically exploiting implicit parallelism in Java. *Concurrency: Practice and Experience*, 9(6):579–619, 1997.
- [BG97b] D. Burger and J. R. Goodman. Billion-transistor architectures. *IEEE Computer*, 30(9):46–49, September 1997.
- [BG98] A. Bik and D. Gannon. Javab – A prototype bytecode parallelization tool. In *ACM Workshop on Java for High-Performance Network Computing*, 1998.
- [BGB98] L. A. Barroso, K. Gharachorloo, and E. Bugnion. Memory system characterization of commercial workloads. In *Proceedings of the 26th Annual International Symposium on Computer Architecture*, pp. 3 – 14, 1998.

- [BGM⁺00] L. A. Barroso, K. Gharachorloo, R. McNamara, A. Nowatzky, S. Qadeer, B. Sano, S. Smith, R. Stets, and B. Verghese. Piranha: A scalable architecture based on single-chip multiprocessing. In *Proceedings of the 27th Annual International Symposium on Computer Architecture*, pp. 282–293, June 2000.
- [BK98] K. R. Bowers and D. Kaeli. Characterising the SPEC JVM98 benchmarks on the Java virtual machine. Technical report, Computer Architecture Research Group, Dept. of Electrical and Computer Engineering, Northeastern University, 1998.
- [BKR98] N. Benton, A. Kennedy, and G. Russell. Compiling Standard ML to Java bytecodes. In *3rd ACM SIGPLAN Conference on Functional Programming*, pp. 129–140, September 1998.
- [BSW⁺99] M. Bull, L. Smith, M. Westhead, D. Henty, and R. Davey. A methodology for benchmarking Java Grande applications. In *Proceedings of the ACM 1999 Java Grande Conference*, pp. 81–88, June 1999.
- [BSW⁺00] M. Bull, L. Smith, M. Westhead, D. Henty, and R. Davey. Benchmarking Java Grande applications. In *Proceedings of the Second International Conference on the Practical Applications of Java*, pp. 63–73, Manchester, U.K., April 2000.
- [Cas96] B. Case. Implementing the Java virtual machine. *Microprocessor Report*, 10(4):12–17, March 25 1996.
- [CB94] Z. Cvetanovic and D. Bhandarkar. Characterization of Alpha AXP performance using TP and SPEC workloads. In *Proceedings of the 21st Annual International Symposium on Computer Architecture*, pp. 60–70, Apr. 1994.
- [CFM⁺97] T. Cramer, R. Friedman, T. Miller, D. Seberger, R. Wilson, and M. Wolczko. Compiling Java, just in time. *IEEE Micro*, 17(3):36–43, May-June 1997.
- [CG00] T. Cohen and J. Gil. Self-calibration of metrics of Java methods. In *Technology of Object-Oriented Languages and Systems*, pp. 94–106, November 2000.
- [CHL99] T. M. Chilimbi, M. D. Hill, and J. R. Larus. Cache-conscious structure layout. In *Proceedings of the ACM SIGPLAN '99 Conference on Programming Language Design and Implementation*, pp. 1 – 12, May 1999.

- [CK93] R. F. Cmelik and D. Keppel. Shade: A fast instruction-set simulator for execution profiling. Technical Report SMLI TR-93-12, Sun Microsystems Laboratories, 1993.
- [CL98] T. M. Chilimbi and J. R. Larus. Using generational garbage collection to implement cache-conscious data placement. In *Proceedings of the 1998 International Symposium on Memory Management (ISMM)*, pp. 37–48, Oct. 1998.
- [CO98] M. Chen and K. Olukotun. Exploiting method-level parallelism in single-threaded Java programs. In *Proceedings of the 1998 International Conference on Parallel Architectures and Compilation Techniques (PACT'98)*, pp. 176–184, Paris, France, October 1998.
- [CS00] Y. Chou and J. P. Shen. Instruction path coprocessors. In *Proceedings of the 27th Annual International Symposium on Computer Architecture*, pp. 270–281, June 2000.
- [CSG98] D. Culler, J. P. Singh, and A. Gupta. *Parallel Computer Architecture: A Hardware/Software Approach*. Morgan Kaufmann Publishers, Inc., 1998.
- [CSK⁺99] R. S. Chappel, J. Stark, S. P. Kim, S. K. Reinhardt, and Y. N. Patt. Simultaneous subordinate microthreading (SSMT). In *Proceedings of the 26th Annual International Symposium on Computer Architecture*, pp. 186–195, May 1999.
- [CZ99] D. Colnet and O. Zendra. Optimizations of Eiffel programs: SmallEiffel, the GNU Eiffel compiler. In *Technology of Object-Oriented Languages and Systems*, pp. 341–350, June 1999.
- [Cza00] G. Czajkowski. Application isolation in the Java Virtual Machine. In *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming and Systems, Languages, and Applications (OOPSLA) 2000*, pp. 354–366, Oct. 2000.
- [DBC⁺00] D. Dillenberger, R. Bordawekar, C. Clark, D. Durand, D. Emmes, O. Gohda, S. Howard, M. Oliver, F. Samuel, and R. S. John. Building a Java virtual machine for server applications: The JVM on OS/390. *IBM Systems Journal*, 39(1):194–210, 2000.
- [DC90] E. Debaere and J. Campenhout. *Interpretation and Instruction Path Coprocessing*. MIT Press, 1990.

- [DC00] J. Dolby and A. Chien. An automatic object inlining optimization and its evaluation. In *Proceedings of the ACM SIGPLAN 2000 Conference on Programming Language Design and Implementation*, pp. 345 – 357, June 2000.
- [DDZ93] D. Detlefs, A. Dosser, and B. Zorn. Memory allocation costs in large C and C++ programs. Technical Report CU-CS-665-93, Univ. of Colorado at Boulder, CS Department, 1993.
- [DH99] S. Dieckmann and U. Holzle. A study of the allocation behaviour of the SPECjvm98 Java benchmarks. In *Proceedings of the 13th European Conference on Object Oriented Programming*, pp. 92–115, Lisbon, Portugal, June 1999.
- [DHPW01] C. Daly, J. Horgan, J. Power, and J. Waldron. Platform independent dynamic Java virtual machine analysis: the Java Grande Forum Benchmark Suite. In *Joint ACM Java Grande–ISCOPE 2001 Conference*, pp. 106–115, Stanford, CA, June 2001.
- [DHW⁺97] J. Dean, J. E. Hicks, C. A. Waldspurger, W. E. Weihl, and G. Chrysos. ProfileMe: Hardware support for instruction-level profiling on out-of-order processors. In *Proceedings of the 30th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 292–302, December 1997.
- [Dig92] Digital Equipment Corp. *Alpha Architecture Handbook*, 1992.
- [DL93] D. Doligez and X. Leroy. A concurrent, generational garbage collector for a multithreaded implementation of ML. In *Conference Record of the 20th Annual ACM Symposium on Principles of Programming Languages*, pp. 113–123, January 1993.
- [DLM⁺78] E. Dijkstra, L. Lamport, A. Martins, C. Scholten, and E. Steffens. On-the-fly garbage collection: An exercise in cooperation. *Communications of the ACM*, 21(11):966–975, November 1978.
- [DLM⁺00] K. Driesen, P. Lam, J. Miecznikowski, F. Qian, and D. Rayside. On the predictability of invoke targets in Java byte code. In *Second Annual Workshop on Hardware Support for Objects and Microarchitectures for Java, held in conjunction with the International Conference on Computer Design (ICCD’00)*, Austin, Texas, 17 September 2000.
- [DWH⁺90] A. Demers, M. Weiser, B. Hayes, H. Boehm, D. Bobrow, and S. Shenker. Combining generational and conservative garbage collection: Framework and implementations. In *Conference*

- Record of the 17th Annual ACM Symposium on Principles of Programming Languages*, pp. 261–269, January 1990.
- [EA97] K. Ebcioglu and E. Altman. DAISY: Dynamic compilation for 100% architectural compatibility. In *Proceedings of the 24th Annual International Symposium on Computer Architecture*, pp. 26–37, 1997.
- [Eis00] A. Eisma. Packaging Java applications for embedded systems. <http://www.dagstuhl.de/~ang/00451-extension/eisma-slides.pdf>, 2000.
- [EKE97] M. W. El-Kharashi and F. Elguibaly. Java microprocessors: Computer architecture implications. In *Proceedings of the 1997 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'97)*, volume 1, pp. 277–280, Victoria, BC, Canada, August 20–22, 1997.
- [EKL99] M. W. El-Kharashi, F. Elguibaly, and K. F. Li. Quantitative analysis for Java microprocessor architectural requirements: Instruction set design. In *First Annual Workshop on Hardware Support for Objects and Microarchitectures for Java, held in conjunction with the International Conference on Computer Design (ICCD'99)*, pp. 50–54, Austin, Texas, October 10, 1999.
- [EKL00] M. W. El-Kharashi, F. Elguibaly, and K. F. Li. An operand extraction-based stack folding algorithm for Java processors. In *Second Annual Workshop on Hardware Support for Objects and Microarchitectures for Java, held in conjunction with the International Conference on Computer Design (ICCD'00)*, pp. 22–26, Austin, Texas, September 17, 2000.
- [EMWW02] A. El-Mahdy, I. Watson, and G. Wright. *Java Machine and Integrated Circuit Architecture (JAMAICA): Choosing the Instruction Set Architecture*, chapter 10, this volume. Kluwer Academic Publishers, 2002.
- [FKR⁺99] R. Fitzgerald, T. Knoblock, E. Ruf, B. Steensgard, and D. Tarditi. Marmot: An optimizing compiler for Java. Technical Report 33, Microsoft Research, June 1999.
- [FLR98] M. Frigo, C. E. Leiserson, and K. H. Randall. The implementation of the Cilk-5 multithreaded language. In *Proceedings of the ACM SIGPLAN '98 Conference on Programming Language Design and Implementation*, pp. 212–223, June 1998.

- [Fly95] M. J. Flynn. *Computer Architecture: Pipelined and Parallel Processor Design*. Jones and Bartlett, Boston, 1995.
- [FS92] M. Franklin and G. S. Sohi. The expandable split window paradigm for exploiting fine-grain parallelism. In *Proceedings of the 19th Annual International Symposium on Computer Architecture*, pp. 58–67, Gold Coast, Australia, May 1992.
- [GAS⁺00] M. Gschwind, E. R. Altman, S. Sathaye, P. Ledak, and D. Appenzeller. Dynamic and transparent binary translation. *IEEE Computer*, 33(3):54–59, March 2000.
- [GJS96] J. Gosling, B. Joy, and G. Steele. *The Java Language Specification*. Addison-Wesley, 1996.
- [GJSB00] J. Gosling, B. Joy, G. Steele, and G. Bracha. *The Java Language Specification*. Addison-Wesley, second edition, June 2000.
- [Glo01] C. J. Glossner. *The Delft-Java Engine*. PhD thesis, Delft University Press, Delft, The Netherlands, 2001.
- [Gos95] J. Gosling. Java intermediate bytecodes. In *Proceedings of the ACM SIGPLAN Workshop on Intermediate Representations (IR95)*, pp. 111–118, January 1995.
- [GV97] C. J. Glossner and S. Vassiliadis. The Delft-Java Engine: An introduction. In *Proceedings of the Third International Euro-Par Conference (Euro-Par'97 Parallel Processing)*, pp. 766–770, Passau, Germany, Aug. 26 - 29 1997.
- [GV98] J. Glossner and S. Vassiliadis. Delft-Java link translation buffer. In *Proceedings of the 24th EUROMICRO conference*, volume 1, pp. 221–228, Vasteras, Sweden, August 25-27 1998.
- [GV99] J. Glossner and S. Vassiliadis. Delft-Java dynamic translation. In *Proceedings of the 25th EUROMICRO conference (EUROMICRO '99)*, volume 1, Milan, Italy, September 8-10 1999.
- [GVSS98] S. Gopal, T. Vijaykumar, J. E. Smith, and G. S. Sohi. Speculative versioning cache. In *Proceedings of the Fourth International Symposium on High-Performance Computer Architecture (HPCA-4)*, Las Vegas, NV, 1998.
- [Har01] D. Hardin. Real-time objects on the bare metal: An efficient hardware realization of the Java virtual machine. In *Proceedings of the Fourth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2001)*, pp. 53–59, May 2001.

- [HCJ⁺97] C.-H. Hsieh, M. T. Conte, T. L. Jonson, J. C. Gyllenhall, and W. W. Hwu. A study of the cache and branch performance issues with running Java on current hardware platforms. In *Proc. of IEEE COMPCON*, pp. 211–216, 1997.
- [Hew02] HP ChaiVM white paper. <http://www.hp.com/emso/products/chaivm/chaivm.whitepaper.html>, 2002.
- [HHS⁺00] L. Hammond, B. A. Hubbert, M. Siu, M. K. Prabhu, M. Chen, and K. Olukotun. The Stanford Hydra CMP. *IEEE Micro*, 20(2):71–84, March/April 2000.
- [Hil98] M. D. Hill. Multiprocessors should support simple memory-consistency models. *IEEE Computer*, 31(8):28–34, August 1998.
- [HS00a] T. Heil and J. E. Smith. Concurrent garbage collection using hardware-assisted profiling. In *Proceedings of the 2000 International Symposium on Memory Management (ISMM)*, pp. 281–290, October 2000.
- [HS00b] T. Heil and J. E. Smith. Relational profiling: Enabling thread-level parallelism in virtual machines. In *Proceedings of the 33rd Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 281–290, December 2000.
- [HW98] L. Huelsbergen and P. Winterbottom. Very concurrent mark-&-sweep garbage collection without fine-grain synchronization. In *Proceedings of the 1998 International Symposium on Memory Management (ISMM)*, pp. 166–175, 1998.
- [HWO98] L. Hammond, M. Willey, and K. Olukotun. Data speculation support for a chip multiprocessor. In *Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-8)*, pp. 58–69, San Jose, CA, October 1998.
- [IBM98] IBM Corporation. *PowerPC 604e RISC Microprocessor User's Manual*, Mar. 1998. G522-0330-00.
- [IBM02] IBM Corporation. Virtual machines are not all created equal: The J9 difference, 2002. <http://www.embedded.oti.com/learn/vaesvm.phtml>.
- [IKY⁺99] K. Ishizaki, M. Kawahito, T. Yasue, M. Takeuchi, T. Ogasawara, T. Suganuma, T. Onodera, H. Komatsu, and T. Nakatani. Design,

implementation and evaluation of optimisations in a just-in-time compiler. In *Proceedings of the ACM 1999 Java Grande Conference*, pp. 119–128, June 1999.

- [Ins00] Instantiations, Inc. Jove, super optimizing deployment environment for Java. http://www.instantiations.com/_vti_bin/shtml.dll/JOVE/jovereportdownload.htm, 2000.
- [J-C01] The J-Consortium JEFF working group, 2001. <http://www.j-consortium.org/jeffwg/index.shtml>.
- [JA99] M. Jordan and M. Atkinson. Orthogonal Persistence for the Java Platform: Draft specification, October 1999. <http://www.sun.com/research/forest/index.html>.
- [Jav] Java Grande Forum. Multithreaded benchmarks. <http://www.epcc.ed.ac.uk/javagrande/threads.html>.
- [JSMG00] P. Joisha, M. Serrano, S. Midkiff, and M. Gupta. Enabling efficient code sharing in Java. In *Proceedings of the 15th International Conference on Supercomputing*, pp. 440–453, Sorrento, Italy, June 2000.
- [JSR00a] JSR-30 Expert Group. Java 2 Platform, Micro edition, Connected, limited device configuration, version 1.0, May 2000. <http://www.jcp.org/aboutJava/communityprocess/final/jsr030/index.html>.
- [JSR00b] JSR-37 Expert Group. Java 2 Platform, Micro edition, Mobile information device profile, version 1.0, September 2000. <http://www.jcp.org/aboutJava/communityprocess/final/jsr037/index.html>.
- [JSR01] JSR-36 Expert Group. Java 2 Platform, Micro edition, Connected device configuration, version 1.0, March 2001. <http://www.jcp.org/aboutJava/communityprocess/final/jsr036/index.html>.
- [JW01] D. Jackson and A. Waingold. Lightweight extraction of object models from bytecode. *IEEE Transactions on Software Engineering*, 27(2):194–202, Feb 2001.
- [Kat85] M. G. H. Katevenis. *Reduced Instruction Set Computer Architectures for VLSI*. MIT Press, 1985.

- [KCSL00] I. Kazi, H. Chan, B. Stanley, and D. Lilja. Techniques for obtaining high performance in Java programs. *ACM Computing Surveys*, 32(3):213–240, September 2000.
- [KG97] A. Krall and R. Grafl. Cacao – A 64-bit JavaVM just-in-time compiler. *Concurrency: Practice and Experience*, 9(11):1017–1030, 1997.
- [KH00] J.-S. Kim and Y. Hsu. Memory system behavior of Java programs: Methodology and analysis. In *Proc. of SIGMETRICS 2000*, pp. 264 – 274, June 2000.
- [KHM89] D. A. Kranz, R. H. Halstead, Jr., and E. Mohr. Mul-T: A high-performance parallel Lisp. In *Proceedings of the ACM SIGPLAN '89 Conference on Programming Language Design and Implementation*, pp. 81–90. ACM, June 1989.
- [Kla00] A. Klaiber. The technology behind Crusoe processors. Technical report, Transmeta Corporation, January 2000.
- [Koo89] P. Koopman. *Stack Computers: The New Wave*. Ellis Horwood, 1989.
- [KP98] C. E. Kozyrakis and D. A. Patterson. A new direction for computer architecture research. *IEEE Computer*, 31(11):24–32, November 1998.
- [KPH⁺98] K. Keeton, D. Patterson, Y. He, R. Raphael, and W. Baker. Performance characterization of a Quad Pentium Pro SMP using OLTP workloads. In *Proceedings of the 25th Annual International Symposium on Computer Architecture*, pp. 15–26, 1998.
- [Kra98] A. Krall. Efficient JavaVM just-in-time compilation. In *Proceedings of the 1998 International Conference on Parallel Architectures and Compilation Techniques (PACT'98)*, pp. 12–18, Paris, France, October 1998.
- [KSS96] S. Kleiman, D. Shah, and B. Smaalders. *Programming with Threads*. SunSoft Press, 1996.
- [LB96] B. Lewis and D. J. Berg. *Threads Primer: A Guide to Multithreaded Programming*. SunSoft Press, 1996.
- [LCB⁺98] D. Lee, P. Crowley, J.-L. Baer, T. Anderson, and B. Bershad. Execution characteristics of desktop applications on Windows NT. In *Proceedings of the 25th Annual International Symposium on Computer Architecture*, pp. 27–38, 1998.

- [Lee97] H. B. Lee. BIT: A tool for instrumenting Java bytecodes. In *USENIX Symposium on Internet Technologies and Systems*, pp. 73–82, Monterey, California, U.S.A., December 1997.
- [Lev01a] M. Levy. Java To Go: Part 1. accelerators process byte codes for portable and embedded applications. *Microprocessor Report*, Feb. 12 2001.
- [Lev01b] M. Levy. Java To Go: Part 2. InSilicon takes Java acceleration to JvXtremes. *Microprocessor Report*, March 5 2001.
- [Lev01c] M. Levy. Java To Go: Part 3. Chicory Systems' Java Accelerator pours a HotShot. *Microprocessor Report*, March 26 2001.
- [Lev01d] M. Levy. Java To Go: Part 4. heterogeneous multiprocessing for Java applications. *Microprocessor Report*, June 4 2001.
- [LJV⁺00] T. Li, L. K. John, N. Vijaykrishnan, A. Sivasubramaniam, J. Sabarinathan, and A. Murthy. Using complete system simulation to characterize SPECjvm98 benchmarks. In *Proceedings of the ACM International Conference on Supercomputing*, pp. 22–33, Santa Fe, NM, May 2000.
- [LM96] C.-K. Luk and T. Mowry. Compiler-based prefetching for recursive data structures. In *Proceedings of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-7)*, pp. 222–233, Oct. 1996.
- [LY99] T. Lindholm and F. Yellin. *The Java Virtual Machine Specification*. Addison-Wesley, second edition, 1999.
- [MDO94] A. Maynard, C. Donnelly, and B. Olszewski. Contrasting characteristics and cache performance of technical and multi-user commercial workloads. In *Proceedings of the 6th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-6)*, pp. 145–156, Oct. 1994.
- [Mic99] MS SDK for Java 4.0. <http://www.microsoft.com/java/>, 1999.
- [ML97] T. Mowry and C.-K. Luk. Predicting data cache misses in non-numeric applications through correlation profiling. In *Proceedings of the 30th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 314–320, Dec. 1997.
- [MLG92] T. Mowry, M. Lam, and A. Gupta. Design and evaluation of a compiler algorithm for prefetching. In *Proceedings of the 5th*

- International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-5)*, pp. 62–73, Oct. 1992.
- [MO98] H. McGhan and M. O'Connor. PicoJava: A direct execution engine for Java bytecode. *IEEE Computer*, 31(10):22–30, October 1998.
- [Moo84] D. A. Moon. Garbage collection in a large LISP system. In *Proceedings of the 1984 ACM Symposium on LISP and Functional Programming*, pp. 234–246, August 1984.
- [Mow94] T. Mowry. *Tolerating latency through software-controlled data prefetching*. PhD thesis, Stanford University, Mar. 1994.
- [MPE96] MPEG Software Simulation Group. MPEG-2 encoder/decoder version 1.2. <http://www.mpeg.org/MSSG>, July 1996.
- [MR79] E. Morel and C. Renviose. Global optimization by supression of partial redundancies. *Communications of the ACM*, 22(2):96–103, 1979.
- [Nat00] NaturalBridge, Inc. BulletTrain description. <http://www.naturalbridge.com/technology.html>, 2000.
- [Nef99] J. Neffenger. The Volano report: Which Java platform is fastest, most scalable? *Java World*, March 1999. <http://www.javaworld.com/javaworld/jw-03-1999/jw-03-volanomark.html>.
- [Nic89] A. Nicolau. Run-time disambiguation: Coping with statistically unpredictable dependencies. *IEEE Transactions on Computers*, 38(5):663–678, May 1989.
- [ON94] J. O'Toole and S. Nettles. Concurrent replicating garbage collection. In *Proceedings of the 1994 ACM Conference on LISP and Functional Programming*, pp. 34–42, 1994.
- [ONH⁺96] K. Olukotun, B. Nayfeh, L. Hammond, K. Wilson, and K.-Y. Chang. The case for a single-chip multiprocessor. In *Proceedings of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-7)*, pp. 2–11, October 1996.
- [OSM⁺00] H. Ogawa, K. Shumira, S. Matsuoka, F. Maruyama, Y. Sohda, and F. Kimura. OpenJIT : An open-ended, reflective JIT compiler framework for Java. In *Proceedings of the 14th Euro-*

- pean Conference on Object-Oriented Programming*, pp. 362–387, Cannes, France, June 2000.
- [Pat01] Patriot Scientific Corporation. *PSC1000 Microprocessor*, 2001.
- [Pau01] L. D. Paulson. Handheld-to-handheld fighting over Java. *IEEE Computer*, pp. 21, July 2001.
- [Pen99] Pendragon. *CaffeineMark 3.0*. Pendragon Software Corporation, <http://www.pendragon-software.com/pendragon/cm3/>, 13 May 1999.
- [PF00] M. Plakal and C. N. Fischer. Concurrent garbage collection using program slices on multithreaded processors. In *Proceedings of the 2000 International Symposium on Memory Management (ISMM)*, pp. 94–100, October 2000.
- [PH96] D. A. Patterson and J. L. Hennessy. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, second edition, 1996.
- [PV94] J. Philips and S. Vassiliadis. High-performance 3-1 interlock collapsing ALU's. *IEEE Transactions on Computers*, 43(3):257–268, March 1994.
- [Ram99] Rambus, Inc. *Direct RDRAM 128/144 Mbit (256Kx16/18x32s) Preliminary Information*, May 1999.
- [RBJ01] R. Radhakrishnan, R. Bhargava, and L. K. John. Improving Java performance using hardware translation. In *Proceedings of the ACM International Conference on Supercomputing*, pp. 427–439, Sorrento, Italy, June 18-21, 2001.
- [RH98] P. Richards and D. Hicks. Virtual integration. *AS/400*, pp. 50–56, March 1998.
- [RLV⁺96] T. Romer, D. Lee, G. Voelker, A. Wolman, W. Wong, J. Baer, B. Bershad, and H. Levy. The structure and performance of interpreters. In *Proceedings of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-7)*, pp. 150–159, Oct. 1996.
- [RMH99] D. Rayside, E. Mamas, and E. Hons. Compact Java binaries for embedded systems. In *9th NRC/IBM Center for Advanced Studies Conference*, pp. 1–14, Toronto, Canada, November 1999.

- [RRJ99] R. Radhakrishnan, J. Rubio, and L. K. John. Characterization of Java applications at the bytecode level and at UltraSPARC machine code level. In *The International Conference on Computer Design ICCD'99*, pp. 281–284, Austin, Texas, October 11–13, 1999.
- [RRJV99] R. Radhakrishnan, J. Rubio, L. John, and N. Vijaykrishnan. Execution characteristics of just-in-time compilers. Technical Report TR-990713, Department of Electrical and Computer Engineering, University of Texas at Austin, 1999.
- [RTJ00] R. Radhakrishnan, D. Talla, and L. K. John. Allowing for ILP in an embedded Java processor. In *Proceedings of the 27th Annual International Symposium on Computer Architecture (ISCA-2000)*, pp. 294–305, Vancouver, BC, Canada, June 10–14, 2000.
- [Rub99] J. Rubio. Characterization of Java application at the bytecode level. Master's thesis, Department of Electrical and Computer Engineering, The University of Texas at Austin, May 1999.
- [RVJ⁺01] R. Radhakrishnan, N. Vijaykrishnan, L. John, A. Sivasubramaniam, J. Rubio, and J. Sabarinathan. Java runtime systems: Characterization and architectural implications. *IEEE Transactions on Computers*, 50(2):131–146, February 2001.
- [RVJS00] R. Radhakrishnan, N. Vijaykrishnan, L. K. John, and A. Sivasubramaniam. Architectural issues in Java runtime systems. In *Proceedings of the Sixth International Symposium on High-Performance Computer Architecture (HPCA-6)*, pp. 387–398, Jan. 2000.
- [SBMG00] M. J. Serrano, R. Bordawekar, S. P. Midkiff, and M. Gupta. Quicksilver: a quasi-static compiler for Java. In *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA) 2000*, pp. 66–82, Minneapolis, MN, Oct. 2000.
- [SBV95] G. S. Sohi, S. E. Breach, and T. N. Vijaykumar. Multiscalar processors. In *Proceedings of the 22th Annual International Symposium on Computer Architecture*, pp. 414–425, Santa Margherita Ligure, Italy, May 1995.
- [SD98] Y. Song and M. Dubois. Assisted execution. Technical Report CENG 98-25, University of Southern California, Dept. of EE-Systems, October 1998.

- [Sha98] T. Shanley. *Pentium Pro and Pentium II System Architecture*. Addison-Wesley, 1998.
- [SHSB99] J. E. Smith, T. Heil, S. Sastry, and T. Bezenek. Achieving high performance via co-designed virtual machines. In *International Workshop on Innovative Architecture*, pp. 77–85, Maui, HI, USA, October 1999. IEEE Computer Society.
- [SM98] J. G. Steffan and T. C. Mowry. The potential for using thread-level data speculation to facilitate automatic parallelization. In *Proceedings of the Fourth International Symposium on High-Performance Computer Architecture (HPCA-4)*, pp. 2–13, Las Vegas, NV, 1998.
- [Smi84] J. E. Smith. Decoupled access/execute computer architecture. In *ACM Transactions on Computer Systems*, pp. 289–308, November 1984.
- [SN94] W. J. Schmidt and K. D. Nilsen. Performance of a hardware-assisted real-time garbage collector. In *Proceedings of the 6th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-6)*, pp. 76–85, 1994.
- [SOT⁺00] T. Suganuma, T. Ogasawara, M. Takeuchi, T. Yasue, M. Kawahito, K. Ishizaki, H. Komatsu, and T. Nakatani. Overview of the IBM Java Just-In-Time Compiler. *IBM Systems Journal*, 39(1):175–193, 2000.
- [SRM⁺98] W. Schmidt, R. Roediger, C. Mestad, B. Mendelson, I. Shavit-Lottem, and V. Bortnikov-Sitnitsky. Profile-directed restructuring of operating system code. *IBM Systems Journal*, 37(2):270, 1998.
- [SS95] T. Scholz and M. Schafers. An improved dynamic register array concept for high-performance RISC processors. In *Proceedings of the 28th Hawaii International Conference on System Sciences*, pp. 181–190, 1995.
- [SSGS00] Y. Shuf, M. J. Serrano, M. Gupta, and J. P. Singh. Characterizing memory behavior of Java workloads: A structured view and opportunities for optimizations. Technical report, IBM T.J. Watson Research Center, Yorktown Heights, NY, 2000.
- [SSGS01] Y. Shuf, M. J. Serrano, M. Gupta, and J. P. Singh. Characterizing the memory behavior of Java workloads: a structured view and

- opportunities for optimizations. In *Joint International Conference on Measurements and Modeling of Computer Systems*, pp. 194–205, Cambridge, MA, June 2001.
- [Sta01] Standard Performance Evaluation Corporation (SPEC). SPEC JVM98 benchmarks. <http://www.spec.org/osg/jvm98>, February 2001.
- [Sun] Sun Microsystems, Inc. Java 2 Platform, Micro edition. <http://java.sun.com/j2me/>.
- [Sun99a] Sun Microsystems, Inc. The Java HotSpot performance engine architecture. <http://java.sun.com/products/hotspot/whitepaper.html>, apr 1999.
- [Sun99b] Sun Microsystems, Inc. MAJC architecture tutorial, May 1999. <http://www.sun.com/microelectronics/MAJC/documentation/docs/majctutorial.pdf>.
- [Sun99c] Sun Microsystems, Inc. *picoJava-II Microarchitecture Guide*, March 1999. <http://www.sun.com/microelectronics/communitysource/picojava/techinfo.html>.
- [Sun99d] Sun Microsystems, Inc. *picoJava-II Programmer's Reference Manual*, March 1999. Part no. 805-2800-06.
- [Sun01a] Sun Microsystems, Inc. The Java HotSpot virtual machine. <http://java.sun.com/products/hotspot/>, 7 November 2001.
- [Sun01b] Sun Microsystems, Inc. PICOJAVA technology: Frequently asked questions. <http://www.sun.com/microelectronics/communitysource/picojava/techfaq.html#perf>, 2001.
- [SV97] J. E. Smith and S. Vajapeyam. Trace processors: moving to fourth-generation microarchitectures. *IEEE Computer*, 30:68–74, September 1997.
- [Tak01] D. Takahashi. Java chips make a comeback. *Red Herring*, July 12 2001.
- [TEE⁺96] D. M. Tullsen, S. J. Eggers, J. S. Emer, H. M. Levy, J. L. Lo, and R. L. Stamm. Exploiting choice: Instruction fetch and issue on an implementable simultaneous multithreading processor. In *Proceedings of the 23rd Annual International Symposium on Computer Architecture*, pp. 191–202, May 1996.

- [TEL95] D. M. Tullsen, S. J. Eggers, and H. M. Levy. Simultaneous multithreading: Maximizing on-chip parallelism. In *Proceedings of the 22nd Annual International Symposium on Computer Architecture*, pp. 392–403, June 1995.
- [Tho64] J. E. Thornton. Parallel operation in the Control Data 6600. In *American Federation of Information Processing Societies Conference Proceedings*, volume 26 Part II, FJCC, pp. 33–41, 1964.
- [Tom67] R. M. Tomasulo. An efficient algorithm for exploring multiple arithmetic units. *IBM Journal of Research and Development*, 11(1):25–33, January 1967.
- [Tow00] Tower Technology. TowerJ3—A new generation native Java compiler and runtime environment. <http://www.towerj.com/products/whitepapergnj.shtml>, 2000.
- [TY96] J. Tsai and P. Yew. The superthreaded architecture: Thread-level data speculation to facilitate automatic parallelization. In *Proceedings of the 1996 International Conference on Parallel Architectures and Compilation Techniques (PACT'96)*, pp. 35–46, 1996.
- [VBE94] S. Vassiliadis, B. Blaner, and R. J. Eickemeyer. SCISM: A scalable compound instruction set machine. *IBM Journal of Research and Development*, 38(1):59–78, January 1994.
- [Vij98] N. Vijaykrishnan. *Issues in the Design of a Java Processor Architecture*. PhD thesis, College of Engineering, University of South Florida, July 1998.
- [VPB93] S. Vassiliadis, J. Phillips, and B. Blanar. Interlock collapsing ALU's. *IEEE Transactions on Computers*, 42(7):825–839, July 1993.
- [VRG98] N. Vijaykrishnan, N. Ranganathan, and R. Gadekarla. Object-oriented architectural support for a Java processor. In *Proceedings of the 12th European Conference on Object-Oriented Programming*, pp. 330–354, July 1998.
- [VRHS⁺99] R. Vallee-Rai, L. Hendren, V. Sundaresan, P. Lam, E. Gagnon, and P. Co. Soot - a Java optimization framework. In *9th NRC/IBM Center for Advanced Studies Conference*, pp. 125–135, Toronto, Canada, November 1999.
- [VS98] T. N. Vijaykumar and G. S. Sohi. Task selection for a multi-scalar processor. In *Proceedings of the 31st Annual IEEE/ACM*

International Symposium on Microarchitecture, Dallas, TX, December 1998.

- [Wal99] J. Waldron. Dynamic bytecode usage by object oriented Java programs. In *Technology of Object-Oriented Languages and Systems*, Nancy, France, June 1999.
- [Wel84] A. Welch. A technique for high performance data compression. *IEEE Computer*, 17(6):8–19, June 1984.
- [WEMW02] G. Wright, A. El-Mahdy, and I. Watson. *Dynamic Java Threads on the JAMAICA Single-Chip Multiprocessor*, chapter 11, this volume. Kluwer Academic Publishers, 2002.
- [WH00] D. Whitlock and A. L. Hosking. A framework for persistence-enabled optimization of Java object stores. In *Proceedings of the Ninth International Workshop on Persistent Object Systems*, pp. 4–17, Sept. 2000.
- [Wil92] P. R. Wilson. Uniprocessor garbage collection techniques. In *Proceedings of the 1992 SIGPLAN Intl. Workshop on Memory Management*, pp. 1–42, September 1992.
- [Wil00] T. Wilkinson. *KAFFE, A Virtual Machine to run Java Code*. <http://www.kaffe.org>, July 2000.
- [WL91] M. E. Wolf and M. S. Lam. A loop transformation theory and an algorithm to maximize parallelism. *IEEE Transactions on Parallel and Distributed Systems*, 2(4):452–471, October 1991.
- [Wol01a] M. Wolczko. *Benchmarking Java with Richards and DeltaBlue*. Sun Microsystems Laboratories, http://research.sun.com/people/mario/java_benchmarking/, 2001.
- [Wol01b] M. Wolczko. *The Tracing JVM*. Sun Microsystems Laboratories, <http://www.experimentalstuff.com/Technologies/TracingJVM>, 19 April 2001.
- [Wri01] G. M. Wright. *A single-chip multiprocessor architecture with hardware thread support*. PhD thesis, Department of Computer Science, University of Manchester, UK, January 2001.
- [WW92] M. Wolczko and I. Williams. Multi-level garbage collection in a high-performance persistent object system. In *Proceedings of the 5th International Workshop on Persistent Object Systems*, pp. 395–418, September 1992.

- [YN95] W. Yamamoto and M. Nemirovsky. Increasing superscalar performance through multistreaming. In *Proceedings of the 1995 International Conference on Parallel Architectures and Compilation Techniques (PACT'95)*, pp. 49–58, June 1995.
- [YST⁺94] W. Yamamoto, M. Serrano, A. Talcott, R. Wood, and M. Nemirovsky. Performance estimation of multistreamed, superscalar processors. In *Proceedings of the 27th Hawaii International Conference on System Sciences*, pp. 105–204, January 1994.
- [Zor90] B. Zorn. Comparing mark-and-sweep and stop-and-copy garbage collection. In *Proceedings of the 1990 ACM Conference on LISP and Functional Programming*, pp. 87–98, June 1990.
- [Zor92] B. Zorn. The measured cost of conservative garbage collection. Technical Report CU-CS-573-92, Univ. of Colorado at Boulder, CS Department, February 1992.
- [ZS00] X. Zhang and M. I. Seltzer. HBench: Java: an application-specific benchmarking framework for Java virtual machines. In *Proceedings of the ACM Java Grande Conference*, pp. 62–70, San Francisco, CA, June 2000.

Index

- aJ-100 Java processor, 45
- aJ-80, 53
- aJfile, 43, 45
- BOA, 144
- Barriers, 146, 177
- Benchmarks
 - CaffeineMark, 4
 - DeltaBlue, 4
 - IBM's Java Server Benchmarks, 4
 - Java Business Benchmarks, 4
 - Java Grande Forum Benchmark Suite, 4, 45
 - Multi-threaded Benchmarks, 4
 - Richards, 4
 - SPEC JVM98, 3
 - See also* SPEC JVM98
 - VolanoMark, 4
 - pBOB, 23
- BulletTrain, 139
- Cacao JIT compiler, 199
- Co-designed VMs, 144
- Complete folding group, 59
- Concurrent GC, 146, 149
- Connected
 - Limited Device Configuration (CLDC), 42
 - See also* MIDP
- Constant pool, 12
- DAISY, 144
- DELFT-JAVA processor, 105
- Dynamic Java Translation unit, 107
- Dynamic binary translation, 188
- Dynamic compilers, 124, 139
- Dynamic linking, 106
- Dynamic translation, 57
- Embedded device, 124
- Embedded processors, 80
- Fill unit, 80, 82
- Folding, 56, 59, 80, 82, 108, 189
- Hardware translation, 81
- Hot-spots, 25
- Hybrid compilation, 125
- Instruction path coprocessor, 93
- Instruction-level parallelism, 208
- Interpreter, 81
- invokevirtual, 16, 45
- J-Consortium's *JEFF* format, 125
- J2ME Connected Device Configuration (CDC), 43
- JAMAICA, 188, 208
- JEMBuilder, 45, 47, 49
- JEMCore, 44
- JOVE, 139
- JVM
 - Jikes, 4
 - Sun JDK 1.1.6, 95
 - Sun JDK 1.2, 95
 - Sun Hotspot compiler, 25
 - Tracing JVM, 4
 - Jalapeño, 4, 21
 - Kaffe, 4
 - Strata, 156
 - chaiVM, 139
- Jalapeño Table Of Contents, 127
- Jalapeño, 127
- Java 2 Platform
 - Micro Edition (J2ME), 42
- Java Execution Thread (JET), 165
- Java Virtual Machine, 2, 9, 45, 56, 124, 189
 - See also* JVM
- Javar restructuring compiler, 208
- Jtrans, 195
- Just In Time (JIT) compiler, 81
- Link Translation Buffer, 106, 111
- Load balancing, 209
- Local variable array, 12
- MAJC, 144, 162
- MUSHROOM, 148
- Mark-and-sweep garbage collector, 46
- Marmot, 139
- Method call depth, 192
- Method inlining, 14
- Micro instruction sequencer, 91
- Mobile Information Device Profile (MIDP), 43
 - See also* CLDC
- Multi-threading, 162, 188, 209
- Multi-windows, 210
- Multiscalar, 164
- picoJava-II, 81, 85
- Princeton LCC compiler, 214
- Profile control table, 152

- ProfileMe, 153
- Quasi-static compilation, 125
- Quicksilver quasi-static compiler, 21, 124
- Real-Time Specification for Java, 43
- Register allocation, 199, 214
- Register renaming, 112
- Register window, 194, 210
- Relational Profiling Architecture (RPA), 145
- Reservation station, 57, 112
- SPEC JVM98, 2–3, 22, 34, 68, 85, 126, 157, 162, 189
- Simultaneous multithreading, 144
- Single-chip multiprocessor, 162, 188, 207–208
- Space-time computing, 162
- Speculative thread, 164
- Stack architecture, 56
- Stack cache, 82
- Stack disambiguation, 89
- Static compilers, 139
- Static stack, 195
- Thread creation, 209
- Thread interface unit, 210
- Thread object, 213
- Tomasulo's algorithm, 67
- TowerJ, 139
- Tracing JVM, 85
- Translate coprocessor, 94
- Translated code buffer, 99
- VLIW, 162
- VLSI, 188, 208
- Virtual method, 16