

Attention Space Owners: The Illinois Wiki will be changed to Read-Only mode on January 10, 2024. For more information, go to [Illinois Wiki Retirement FAQ](#) for more information.

[Dashboard](#) / [Home](#)

Architecture, Compilers, Parallel Computing And Systems Phd Qualifying Examination

Created by Unknown User (conadmin), last modified by Torrellas, Josep on Sep 17, 2021

All students taking the qualifying exam in this area should subscribe to the architecture@cs.illinois.edu mailing list to ensure they receive all qual related announcements from the area.

This area has three different tracks: architecture, compilers, and parallel programming. Each track has a core and a specialization reading list.

- [Architecture core and specialization reading list](#)
- [Compilers core and specialization reading list](#)
- [Parallel programming core and specialization reading list](#)

Students in this area may also choose one of the tracks in the [Programming Languages, Formal Systems and Software Engineering](#) area for a non-specialization track.

Preparation for the exam

You are asked to:

- Read the following literature: 1) the core and specialization list for your chosen track, and 2) the core list for two additional tracks (one of these tracks may be from the programming languages, formal methods, and software engineering area).
- Prepare a presentation of one paper (selected by area faculty) from the core or specialization list of your chosen track. You can find the chosen paper in the corresponding track page.

Papers selected for Qual presentations

Architecture track: Norman P. Jouppi, et al. 2017. In-Datacenter Performance Analysis of a Tensor Processing Unit. In Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA '17). 1–12.

Compilers track: [Andreas Gal](#), [Brendan Eich*](#), [Mike Shaver](#), [David Anderson](#), [David Mandelin](#), [Mohammad R. Haghighat](#), [Blake Kaplan](#), [Graydon Hoare](#), [Boris Zbarsky](#), [Jason Orendorff](#), [Jesse Ruderman](#), [Edwin Smith](#), [Rick Reitmaier](#), [Michael Bebenita](#), [Mason Chang](#), [Michael Franz](#). "Trace-based Just-in-Time Type Specialization for Dynamic Languages", In Proceedings of the ACM SIGPLAN 2009 Conference on Programming Language Design and Implementation (PLDI). June 2009.

Parallel Programming track: [Milind Kulkarni](#), [Martin Burtcher](#), [Rajasekhar Inkulu](#), [Keshav Pingali](#) and [Calin Cascaval](#). "How Much Parallelism is There in Irregular Applications?" Principles and Practices of Parallel Programming (PPOP), February, 2009.

Exam process

The exam is oral and lasts 90 minutes. You will start by presenting the paper that was selected. The faculty on your committee will ask you about the whole reading list, not necessarily specifically about the paper you present.

No labels

Attention Space Owners: The Illinois Wiki will be changed to Read-Only mode on January 10, 2024. For more information, go to [Illinois Wiki Retirement FAQ](#) for more information.

[Dashboard](#) / [Home](#)

Architecture Track

Created by Unknown User (conadmin), last modified by Torrellas, Josep on Sep 17, 2021

Architecture Track

PhD qualifying examination reading list

CORE (5):

1. B. Sinharoy, R. Kalla, W. J. Starke, H. Q. Le, R. Cargnoni, J. A. Van Norstrand, B. J. Ronchetti, J. Stuecheli, J. Leenstra, G. L. Guthrie, D. Q. Nguyen, B. Blaner, C. F. Marino, E. Retter, and P. Williams. 2011. IBM POWER7 multicore server processor. *IBM J. Res. Dev.* 55, 3 (May 2011), 191-219.
2. Brightwell, R.; Predretti, K.T.; Underwood, K.D.; Hudson, T.; "SeaStar Interconnect: Balanced Bandwidth for Scalable Performance," *IEEE Micro* vol.26, no.3, pp.41-57, May-June 2006.
3. Stephen W Keckler, William J Dally, Brucek Khailany, Michael Garland, David Glasco. GPUs and the future of parallel computing. *IEEE Micro*, Volume 31, Issue 5, Pages 7-17.
4. Norman P. Jouppi, et al. 2017. In-Datacenter Performance Analysis of a Tensor Processing Unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA '17)*. 1–12.
5. Daniel Lenoski, James Laudon, Kourosh Gharachorloo, Anoop Gupta, John Hennessy. The directory-based cache coherence protocol for the DASH multiprocessor. *International Symposium on Computer Architecture (ISCA)*, 1990.

SPECIALIZATION (15):

1. Ikuo Magaki, Moein Khazraee, Luis Vega Gutierrez, and Michael Bedford Taylor. "ASIC Clouds: Specializing the Datacenter" *ISCA'16*
2. Yu-Hsin Chen, Joel Emer, and Vivienne Sze. "Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks" *ISCA'16*
3. Arthur Perais, André Sez nec, "EOLE: Paving the way for an effective implementation of value prediction," 2014 *ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, Minneapolis, MN, 2014, pp. 481-492
4. J. Gregory Steffan and Todd 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*, pages 2-13, February, 1998.
5. Ling Ren, Xiangyao Yu, Christopher W. Fletcher, Marten van Dijk, Srinivas Devadas. "Design Space Exploration and Optimization of Path Oblivious RAM in Secure Processors" *ISCA'13*
6. Hadi Esmaeilzadeh, Emily Blem, Renée St. Amant, Karthikeyan Sankaralingam, Doug Burger. "Dark Silicon and the End of Multicore Scaling" *ISCA'11*
7. Nathan Binkert, Al Davis, Norman P. Jouppi, Moray McLaren, Naveen Muralimanohar, Robert Schreiber, and Jung Ho Ahn. 2011. "The role of optics in future high radix switch design." In *Proceedings of the 38th annual international symposium on Computer architecture (ISCA '11)*. ACM, New York, NY, USA, 437-448. PDF

8. Mohit Tiwari, Hassan M G Wassel, Bitu Mazloom, Shashidhar Mysore, Frederic T Chong, Timothy Sherwood. "Complete Information Flow Tracking from the Gates Up" ASPLOS'09
9. Kevin Lim, Parthasarathy Ranganathan, Jichuan Chang, Chandrakant Patel, Trevor Mudge, Steven Reinhardt. "Understanding and Designing New Server Architectures for Emerging Warehouse-Computing Environments" ISCA'08.
10. Gabriel H. Loh. "3D-Stacked Memory Architectures for Multi-core Processors." ISCA'08.
11. Luis Ceze, James M. Tuck, Pablo Montesinos, and Josep Torrellas. "BulkSC: Bulk Enforcement of Sequential Consistency." ISCA'07.
12. Shekhar Borkar. "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation" IEEE Micro'05
13. S. Adve and K. Gharachorloo. "Shared Memory Consistency Models: A Tutorial" IEEE Computer, 1996.
14. Moinuddin Qureshi, Viji Srinivasan, and Jude A. Rivers. "Scalable High-Performance Main Memory System Using Phase-Change Memory Technology." International Symposium on Computer Architecture (ISCA) 2009.
15. D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Zeisler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. "Razor: A lowpower pipeline based on circuit-level timing speculation. In International Symposium on Microarchitecture" December 2003.

No labels

Attention Space Owners: The Illinois Wiki will be changed to Read-Only mode on January 10, 2024. For more information, go to [Illinois Wiki Retirement FAQ](#) for more information.

[Dashboard](#) / [Home](#)

Compilers Track

Created by Unknown User (conadmin), last modified by Torrellas, Josep on Sep 17, 2021

Compilers Track

PhD qualifying examination

PhD qualifying examination

Below is the reading list for the compiler track. Students taking the Breadth exam in Compilers are required to take CS 426 (or equivalent). Students taking the Depth exam in Compilers are required to take CS 426 (or equivalent) *and* CS 526.

All ACM conference and journal papers are available free to UIUC students and staff from the ACM Digital Library. To access the ACM Digital Library from an off-campus computer, please use the University's library proxy service:

(<http://dl.acm.org.proxy2.library.illinois.edu/dl.cfm?coll=portal&dl=ACM>)

For some of the topics, we recommend reading one or several book chapters instead of a paper; for others, we advise that in addition to the paper in this list you refer to a book chapter or section, where you might find a more clear explanation.

Below is the list of books that you might need.

ON RESERVE at GRAINGER LIBRARY: The Dragon Book and Keith Cooper's book "Engineering a Compiler" are on reserve in Grainger Library for CS 426 and CS 526.

- [Alfred V. Aho](#), [Monica S. Lam](#), Ravi Sethi, and [Jeffrey D. Ullman](#). "Compilers: Principles, Techniques, and Tools". (The Dragon book!) Published by Pearson, Addison-Wesley. Second Edition, 2007.
- Keith Cooper and Linda Torczon. "Engineering a Compiler". Published by Morgan Kaufman, Second Edition, 2011.

CORE

- 1) Aho, Lam, Sethi and Ullmann – Chapters 8.4, 9
- 2) [Michael E. Wolf and Monica S. Lam](#). "A Data Locality Optimizing Algorithm," ACM SIGPLAN Conference on Programming Languages Design and Implementation, June 1991, pp. 30-44.
- 3) [A. Milanova, A. Rountev, and B. Ryder](#). "Parameterized Object Sensitivity for Points-To Analysis for Java", ACM Transactions on Software Engineering Methodology, 14(1), pp. 1--41, 2005.
- 4) [B.Ramakrishna Rau, Michael S. Schlansker, P. P. Tirumalai](#). "Code Generation Schema for Modulo Scheduled Loops", In Proceedings of the 25th Annual International Symposium on Microarchitecture(MICRO 25), 1992.

SPECIALIZATION

Internal Organization

- 5) [Marc Auslander and Martin Hopkins](#). "An Overview of the PL.8 Compiler," Proceedings of the 1982 SIGPLAN Symposium on Compiler Construction, Boston, MA pp. 22-31, 1982.
- 6) [Chris Lattner and Vikram Adve](#). "LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation", In

[Proceedings of the international Symposium on Code Generation and Optimization \(CGO '04\), 2004.](#)

Data Flow Analysis

7) J. B. Kam and J. D. Ullman. "Global Data Flow Analysis and Iterative Algorithms", Journal of the ACM, 23(1), pp. 158-171, Jan. 1976. [https://docs.google.com/viewer?url=http://rsim.cs.illinois.edu/arch/qual_papers/compilers/kam.pdf&embedded=true] *This paper covers fundamental material in the area. For additional explanation of this material, refer to the Dragon book: Aho, Lam, Sethi and Ullmann – Chapter 9 (See reserve list above)*

8) Lazy Code Motion. Found in the Dragon book , Section 9.5, or Keith Cooper and Linda Torczon. "Engineering a Compiler", Section 10.3.2, Published by Morgan Kaufman, Second Edition, 2011. (*see reserve list above*)

SSA

9) Ron Cytron, Jeanne Ferrante, Barry K. Rosen, Mark N. Wegman, and F. Kenneth Zadeck. "Efficiently Computing Static Single Assignment Form and the Control Dependence Graph", ACM Transactions on Programming Languages and Systems, 13 (4), pp. 451-490, October 1991.

Interprocedural Analysis

10) "Program Analysis via Graph Reachability", Thomas Reps, Invited paper, InProceedings of the 1997 International Symposium on Logic Programming, Oct. 1997.

Pointer analysis

11) Ben Hardekopf, Calvin Lin: "The Ant and the Grasshopper: Fast and Accurate Pointer Analysis for Millions of Lines of Code", InProceedings of the ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation (PLDI), June 2007.

Vectorization

12) Sam Larsen and Saman Amarasinghe. "Exploiting Superword Level Parallelism with Multimedia Instruction Sets", InProceedings of the ACM SIGPLAN 2000 Conference on Programming Language Design and Implementation (PLDI), June 2000.

Program Synthesis

13) Matteo Frigo. "A Fast Fourier Transform Compiler", InProceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation (PLDI), June 1999.

14) Kamen Yotov, Xiaoming Li, Gang Ren, Michael Cibulskis, Gerald DeJong, María Jesús Garzarán, David Padua, Keshav Pingali, Paul Stodghill, and Peng Wu. "A Comparison of Empirical and Model-Driven Optimization", InProceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI), June 2003.

Dynamic Analysis

15) Chi-Keung Luk, Robert Cohn, Robert Muth, Harish Patil, Artur Klauser, Geoff Lowney, Steven Wallace, Vijay Janapa Reddi, and Kim Hazelwood. "Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation", InProceedings of the ACM SIGPLAN 2005 Conference on Programming Language Design and Implementation (PLDI), June 2005.

16) Andreas Gal, Brendan Eich* , Mike Shaver, David Anderson, David Mandelin, Mohammad R. Haghighat, Blake Kaplan, Graydon Hoare, Boris Zbarsky, Jason Orendorff, Jesse Ruderman , Edwin Smith, Rick Reitmaier, Michael Bebenita, Mason Chang, Michael Franz. "Trace-based Just-in-Time Type Specialization for Dynamic Languages", InProceedings of the ACM SIGPLAN 2009 Conference on Programming Language Design and Implementation (PLDI). June 2009.

Native Code Generation

17) Preston Briggs, Keith Cooper, and Linda Torczon. "Improvements to Graph Coloring Register Allocation," ACM Transactions on Programming Languages and Systems, 16(3), pp. 428-455, May 1994.

18) Sorav Bansal, Alex Aiken. "Automatic Generation of Peephole Superoptimizers", In Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 2006.

Correctness

19) Thomas Ball, Rupak Majumdar, Todd Millstein, and Sriram K. Rajamani. "Automatic Predicate Abstraction of C Programs", In Proceedings of the ACM SIGPLAN 2001 Conference on Programming Language Design and Implementation (PLDI). June 2001.

20) Yichen Xie and Alex Aiken. "Saturn: A Scalable Framework for Error Detection Using Boolean Satisfiability", ACM Transactions on Programming Languages and Systems, 29 (3), May 2007.

21) Rastislav Bodik, Rajiv Gupta and Vivek Sarkar. "ABCD: Eliminating Array Bounds Checks on Demand", In Proceedings of the ACM SIGPLAN 2001 Conference on Programming Language Design and Implementation (PLDI), June 2000.

No labels

Attention Space Owners: The Illinois Wiki will be changed to Read-Only mode on January 10, 2024. For more information, go to [Illinois Wiki Retirement FAQ](#) for more information.

[Dashboard](#) / [Home](#)

Parallel Programming Track

Created by Unknown User (conadmin), last modified by Torrellas, Josep on Sep 17, 2021

Parallel Programming Track

PhD qualifying examination

Core

- 1) A.Y. Grama, A. Gupta, and V. Kumar. "Isoefficiency: Measuring the Scalability of Parallel Algorithms and Architectures," *Parallel & Distributed Technology: Systems & Applications*, IEEE, see also *IEEE Concurrency*, Volume 1, Issue 3, Aug. 1993, pp. 12-21.
- 2) M.S. Warren and J.K. Salmon. "A Parallel Hashed Oct-Tree N-body Algorithm," *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*, pp. 12-21.
- 3) W. Daniel Hillis and Guy L. Steele. "Data Parallel Algorithms," *Communications of the ACM*, December 1986, pp. 1170-1183.

Specialization

Algorithms: Key Concepts

- 4) David E. Culler, Richard M. Karp, David Patterson, Abhijit Sahay, Eunice E. Santos, Klaus Erik Schauser, Ramesh Subramonian, Thorsten von Eicken. "LogP: A Practical Model of Parallel Computation," *Communications of the ACM*, Volume 39, Issue 11, November 1996, pp. 78 - 85.
- 5) J.L. Gustafson, G.R. Montry, R.E. Benner. "Development of Parallel Methods for a 1024-processor Hypercube," *SIAM J. Sci. Stat. Comput.*, 9(4), pp. 609-638.
- 6) R.M. Karp and Y. Zhang. "Randomized Parallel Algorithms for Backtrack Search and Branch-and-bound Computation," *Journal of the ACM*, 40(3), July 1993, pp. 765-789.

Specific Algorithms

- 7) A. Gupta, G. Karypis, and V. Kumar. "Highly Scalable Parallel Algorithms for Sparse Matrix Factorization," *Parallel and Distributed Systems*, *IEEE Transactions on* Volume 8, Issue 5, May 1997, pp. 502-520.
- 8) K. Thearling and S. Smith. "An Improved Supercomputer Sorting Benchmark," *Conference on High Performance Networking and Computing*, *Proceedings of the 1992 ACM/IEEE conference on Supercomputing*, pp. 14-19, 1992.
- 9) L.V. Kale and Sanjeev Krishnan. "A Comparison Based Parallel Sorting Algorithm," *International Conference on Parallel Processing*, August 1993, pp. 196-200.
- 10) R. C. Agarwal, S. M. Balle, F. G. Gustavson, M. Joshi, and P. Palkar. "A Three-dimensional Approach to Parallel Matrix Multiplication," *IBM Journal of Research and Development*, Volume 39, Number 5, 1995.

State-space Search and Discrete Event Simulation

- 11) V. Nageshwara Rao and Vipin Kumar. "Superlinear Speedup in Parallel State-Space Search," Lecture Notes In Computer Science, Vol. 338, Proceedings of the Eighth Conference on Foundations of Software Technology and Theoretical Computer Science, pp. 161-174, 1988.
- 12) L.V. Kale, B. Ramkumar, V. Saletore, and A.B. Sinha. "Prioritization in Parallel Symbolic Computing," Lecture Notes in Computer Science, Vol. 748, pp. 12-41, 1993.
- 13) Richard M. Fujimoto. "Parallel Discrete Event Simulation," Communications of the ACM, Vol. 33, Issue 10, pp. 30-53, 1990.

Benchmarks and Performance

- 14) Fabrizio Petrini, Darren J. Kerbyson, Scott Pakin. "The Case of the Missing Supercomputer Performance: Achieving Optimal Performance on the 8,192 Processors of ASCI Q," Conference on High Performance Networking and Computing, Proceedings of the 2003 ACM/IEEE Conference on Supercomputing, 2003.

Runtime Issues

- 15) V. Bala, J. Bruck, R. Cypher, P. Elustondo, A. Ho, Ching-Tien Ho, S. Kipnis, and M. Snir. "CCL: A Portable and Tunable Collective Communication Library for Scalable Parallel Computers," IEEE Transactions on Parallel and Distributed Systems, Volume 6, Issue 2, Feb. 1995, pp. 154 - 164.

Parallel Architectures

- 16) A. Gara et. al. "Overview of the Blue Gene/L System Architecture," IBM Journal of Research and Development, Vol. 49, Number 2/3, March/May 2005, pp. 195-212.
- 17) William J. Dally and Hiromichi Aoki. "Deadlock-free Adaptive Routing in Multicomputer Networks Using Virtual Channels," IEEE Transactions on Parallel and Distributed Systems, 4(4), pp. 466-475, April 1993.
- 18) Charles E. Leiserson. "Fat-trees: Universal Networks for Hardware-efficient Supercomputing," IEEE Transactions on Computers, Vol. 34, No. 10, pp. 92-901, Oct. 1985.

New Papers

- 19) Milind Kulkarni, Martin Burtscher, Rajasekhar Inkulu, Keshav Pingali and Calin Cascaval. "How Much Parallelism is There in Irregular Applications?" Principles and Practices of Parallel Programming (PPoPP), February, 2009.
- 20) N. S. Arora, R. D. Blumofe and C. G. Plaxton. "Thread Scheduling for Multiprogrammed Processors," Theory of Computing Systems 34, 115-144 (2001).

No labels