Qing Yi

yi7@llnl.gov

Education

- **Rice University**, Houston, TX, USA.
  Ph.D. in Computer Science, May 2002;
  M.S. in Computer Science, Dec 1999.
- **Institute of Computing Technology, Chinese Academy of Science**, P.R.China.
  M.S. in Computer Science, June, 1995
- **Shandong University**, P.R.China.
  B.S. in Computer Science, June, 1992

Employment

- **Senior Computer Scientist**, Lawrence Livermore National Laboratory.
  Sep 2023 - present
- **Staff Software Engineer**, Google.
  Jan 2020 - Aug 2023
- **Associate Professor**, Computer Science Department, University of Colorado at Colorado Springs.
  Aug, 2012 – Dec 2021.
- **Assistant Professor**, Computer Science Department, University of Texas at San Antonio.
  Aug 2005 – Aug 2012.
- **Post-doctorate associate**, Lawrence Livermore National Laboratory, Livermore CA.
  Jan 2003 – Aug 2005.
- **Post-doctorate associate** , Computer Science department, Rice University, Houston, TX.
  June 2002 - Dec. 2002.
- **Software engineer**, Beijing Zhenzhong Electronic Inc., P.R.China
  June 1995 - July 1996.

Awards and honors

- **NSF Career Award**, *Multilayer Code Synthesis For Correctness and Performance*, $399,953, Aug,2008 - July, 2013.
- **The R&D 100 award**, *ROSE: Making Compiler Technology Accessible To All Programmers.* As part of the ROSE development team lead by Dr. Daniel Quinlan from Lawrence Livermore National Laboratory. 2009.

Industrial Activities

- **Jan 2020-Aug 2023**, Staff software engineer, Google. Worked on the XLA compiler infrastructure for TPUs. Improved existing compiler optimizations to address customer requests, developed new program analysis and optimization capabilities as needed, to support the various deep neural network (artificial intellgence) software infrastructures including tensorflow, Jax, and Pytorch.
- **June 1995 - July 1996**, Software engineer, Beijing Zhenzhong Electronic Inc., P.R.China. Worked on developing a compiler for the main hand-held device product of the company.

Research interests

Compiler optimization for high performance computing; parallel, distributed, and heterogeneous computing; Software correctness and productivity; programming languages and domain-specific abstractions.

Research Activities

- **2020-2022** Principle investigator of NSF award CCF-1910488 (2019-2022), developing new compiler techniques to automatically enhance the whole-program level parallelization and computation efficiency of large software applications, thereby simultaneously enhancing their productivity, correctness assurance, performance, and hardware portability.

◇ **2014-2019** Principle Investigator of NSF award CCF-1421443 (2014-2019) and some DOE/LLNL subcontracts, developing a new methodology for compiler optimization, where the data structure and algorithm implementations of software applications are independently normalized and categorized into commonly occurring patterns, compiler optimizations are made customizable components that can be flexibly composed, and all optimizations are closely coordinated and collectively specialized to attain a highest level of performance. The pattern-based specialization specifically targets dense/sparse matrix codes, stencil computations, and graph/machine learning algorithms.

◇ **2012-2018** Principle investigator of NSF award CCF-1218179 (2012-2018), developing a library of highly concurrent scalable data abstractions with associated programming interface and optimization support to automate the efficient use of such abstractions in multi-threaded C/C++ applications on multicore architectures.

◇ **2005-2018** Lead Principle Investigator of NSF awards CCF-1421443 (2014-2018), CCF-0747357/1261811 (2008-2017) and CCF-0833203/1261778 (2008-2014), DOE award SC0001770 (2009-2013), and several DOE/LLNL subcontracts. Focus on developing a new optimization model that effectively support

1. developers to interact with advanced optimizing compilers to provide both domain-specific knowledge and high-level optimization strategies;
2. computational specialists to exercise programmable control and effectively specialize both domain-specific and general-purpose compiler optimizations to attain a highest level of performance;
3. the impact of architecture-sensitive optimizations accurately modeled, extensively parameterized, and automatically tuned to attain portable high performance.

The research developed POET, an interpreted program transformation language (POHLL'07, SPE'11) designed to support programmable control and parameterization of compiler optimizations for performance tuning using empirical search (LCSD'07, LCPC'08, SMART'10) and machine learning (ICMLA'16) techniques. The ROSE source-to-source optimizing compiler from DOE/LLNL has been adapted to automatically produce parameterized transformation scripts in POET (CGO'11,LCPC'15) to support the automated parameterization and tuning of compiler optimizations (HIPEAC'11, CF'11, CF'12, CF'15). POET has been used as an optimization scripting language to automatically attain high performance for both HPC library kernels (LCSD'07, TACO'12, SC'13, MICRO'14, LCPC'16) and scientific applications (ICPP'11, ICPP'13, Cluster'16).

◇ **Jan 2003 – Aug 2005** Post-doc at DOE/Lawrence-Livermore National Laboratory, supervised by Dr. Daniel Quinlan, developing the ROSE source-to-source compiler infrastructure to optimize scientific applications through the recognition of user-defined abstractions.

- Develop control-flow, data-flow, loop, and data layout compiler optimizations to user-defined library abstractions in C++ (LCPC'04, LCPC'05, POHLL'06).
- Develop techniques to automate the parallelization of C++ applications using OpenMP directives (LCPC'03).

◇ **Sep 1996 - Dec, 2002** Ph.D student and later post-doc in the Computer Science Department, Rice University, supervised by Professor Ken Kennedy, developing advanced loop optimizations to improve the memory hierarchy performance of modern architectures. Ph.D dissertation *Transforming Complex Loop Nests for Locality* includes:

- A new compiler optimization technique named *Dependence hoisting*, which collectively accomplishes the fusion and interchange of arbitrarily nested loops (Journal of Supercomputing, 2004).
- A framework for automatically transforming arbitrarily nested loops into cache oblivious recursive forms to improve hierarchical memory performance of computational kernels (PLDI'00).

- **Sep 1992 - May 1995** M.S student in Institute of Computing Technology, Chinese Academy of Science. Developed an automatic Chinese-handwriting online recognition system. Dissertation titled *Chinese character encoding and automatic recognition.*

Research Grants

- **NSF small-core award CCF-1910488**, Principal Investigator, $499,510. Whole-application Coordiated Parallelization Through The Optimization Of Abstraction Hierarchies. Oct, 2019 - Sep 2022.
- **NSF small-core award CCF-1421443**, Principal Investigator, $477,631. Specializing Compilers For High Performance Computing Through Coordinated Data and Algorithm Optimizations. Aug, 2014 - July 2020.
- **DOE/LLNL subcontract**: Optimization Analysis In the ROSE Compiler Infrastructure: $86315. Principle investigator. Sep 2018-Dec 2019
- **NSF award SBIR Phase I**: Enhancing the Performance of scientific applications Through intelligent advices (with PI Brandon Nesterenko from Crestone Computing LLC): $225K ($37943 subcontracted to UCCS). Co-Principle investigator. July 2018-June 2019
- **NSF REU Site award IIS-1359275**, Co-Principle investigator (PI is Dr. Jugal Kalita), $359,901. Machine Learning: Theory and Applications. 2014-2018.
- **NSF small-core award CCF-1218179**, Principle Investigator, $250,000 (part of $500,000 allocated for the overall project). Collaborative Research: Programming Interface And Runtime For Self-Tuning Scalable C/C++ Data Structures. June, 2012 - May 2018.
- **DOE/LLNL contract: B619563**, Principle Investigator, $53,389. Specializing Optimization of Stencil Computations in DOE Applications on Many-Core Architectures. Sep 2016 - Aug 2019.
- **NSF I-corp award IIP-1723712**, Principle Investigator, $50,000. I-Corps: Optimized Compiler Applications. Feb 1-July 31, 2017.
- **NSF CAREER award CCF-0747357**, Principal Investigator, $399,953. Multilayer Code Synthesis For Correctness and Performance. Aug, 2008 - July, 2013 (extended through July, 2017).
- **Multiple NSF REU supplemental awards**, Research Experiences for Undergraduates (REU). Principal Investigator.
- **DOE/LLNL subcontract B607283**, Principal Investigator, $53,793. Integrated Programmable Compiler Optimization For Scientific Applications. Feb, 2014 - Feb 2016.
- **NSF HECURA award CCF-0833203**, $462,000, Principal Investigator (with co-PIs R. Clint Whaley and Daniel Quinlan), Programmable Code Optimization and Empirical Tuning For High-End Computing. Sep, 2008 - Aug, 2014.
- **DOE/LLNL subcontract B602385**, $50,798, Principle investigator. Automatically Migrating Stencil Computations For Many-Core Architectures. Dec 2012-Jan, 2014.
- **DOE Office of Science award DE-SC0001770**, $360,000, Principal Investigator (with co-PIs R. Clint Whaley, Daniel Quinlan, and Apan Qasem). A Multi-Language Environment For Programmable Code Optimization and Empirical Tuning. Sep, 2009 - Sep, 2013.
- **NSF CRI award CNS-0855247**, $227,178, Co-Principal Investigator (with PI Kleanthis Psarris; Co-PIs Ali S. Tosun and Dakai Zhu), II-NEW: Enhanced Parallelization For High Performance Computing. 08/2009 - 07/2012.
- **Lawrence Livermore National Laboratory B574748**, $51,710, Principal Investigator, Program Analysis and Optimization For the Empirical Tuning of Scientific Applications. 04/2008 - 10/2009.
- **Lawrence Livermore National Laboratory B555671**, $70193, Principal Investigator, Semantics-driven Optimization of Scientific Applications. 01/2006 - 03/2008.

Teaching Activities

◇ **8/2012-12/2021** Associate professor of Computer Science, University of Colorado at Colorado Springs. Developed and taught the following courses.

cs4100/5100: Compiler Design, undergraduate and graduate level, Spring, 2013, 2014, 2015, 2016, 2017, 201
cs3160: Concepts of Programming Languages, undergraduate level, Fall, 2013, 2014, 2015, 2016.
Spring/Fall 2017, Spring 2018.
cs5080:Advanced Topics of High Performance Computing, graduate level. Fall 2014 and Fall 2017
cs3060: Object-oriented programming in C++, undergraduate level, Fall 2012, 2015.

Dissertations supervised:

- S. Faizur Rahman (Sep 2009-Aug 2014), Ph.D thesis (UCCS). *Collectively Optimizing Scientific Applications For Both Performance And Power.*
- Jichi Guo (May 2009-Dec 2014), Ph.D thesis (UCCS). *Optimizing Large-Scale Scientific Applications Through The Integration of High-Level Semantics And Analytical Performance Modeling.*
- Jiange Zhang (Aug 2014-Dec 2020), Ph.D thesis (UCCS), dissertation topic: *Efficiently Synchronized Data Abstractions.*
- Brandon Nesterenko (Aug 2015-Dec 2020), Ph.D Thesis (UCCS), dissertation topic: *Understanding and Optimizing Large Scientific Applications.*

Supervised undergraduate Students participation in research:
Spencer Powell (2017 -), Howie Hill (2017 -), Nathan Harman (2016-2017), Aaron Wilson (June-Dec, 2016), Adam Barker (2014-2016), Brandon Nesteranko (Mar-Dec 2013)

◇ **7/2014**: guest lecturer of the DragonStar series, taught a 5-day course on advanced compiler optimizations at Chinese University of Science and Technology.
(http://www.cs.uccs.edu/~qyi/classes/Dragonstar/index.htm)

◇ **8/2005-8/2012** Assistant professor, Computer Science Department, University of Texas at San Antonio. Developed and taught the following courses.

cs3723: Programming languages, undergraduate level.
Fall 2005; Fall 2006; Spring&Fall 2007, Fall 2008, Spring&Fall 2009, Spring 2011.
cs4713: Compiler writing, undergraduate level. Spring 2006, Fall 2007.
cs5363: Programming languages and compilers. Graduate level.
Fall 2006, Fall 2008, Fall 2010.
cs6363: Advanced compiler construction. Graduate level. Spring 2010.
cs6463: Advanced topics: Program analysis and its use in software development.
Graduate level. Spring, 2008.

Dissertations supervised:

- Akshatha Bhat (Sep 2010-May 2012) MS thesis (UTSA). *Automatically Tuning Task Scheduling Policies on Multicore Architectures.*
- Anitha Marneni (Sep 2007-Aug 2010) MS thesis (UTSA). *Automatic Generation Of Object-Oriented Class Implementations From Behavioral Specifications.*
- Dang Nguyen (Dec 2008-Dec 2009). Undergraduate Honors thesis (UTSA). *Analyzing The Runtime States Of Object-Oriented Implementations.*

Journal Publications

◇ **Compiler-Driven Approach for Automating Non-Blocking Synchronization In Concurrent Data Abstractions**, Jiange Zhang, Qing Yi, Christina Perterson, Damian Dechev. Concurrency and Computation: Practice and Experience. Feb 2024; 36(5):e7935. doi: 10.1002/cpe.7935

◇ **Enhancing the Effectiveness of Inlining in Automatic Parallelization**, Jichi Guo, Qing Yi, and Kleanthis Psarris. Int J Parallel Prog 50, 65–88 (Feb 2022). doi:10.1007/s10766-021-00722-1

- ◇ **Layout-oblivious compiler optimization for matrix computations**. Huimin Cui, Qing Yi, Jingling Xue, and Xiaobing Feng. ACM Transactions on Architecture and Code Optimization. Vol 9, No 4, pages 35:1-20. Jan, 2013.
- ◇ **POET: A Scripting Language For Applying Parameterized Source-to-source Program Transformations**. Qing Yi. Software Practice & Experience. John Wiley&Sons. Vol 42, issue 6, pages 675-706. May, 2012.
- ◇ **Transforming Complex Loop Nests For Locality**. Qing Yi, Ken Kennedy, and Vikram Adve. The Journal Of Supercomputing, Vol 27, pages 219-264, 2004
- ◇ **Improving Memory Hierarchy Performance Through Combined Loop Interchange and Multi-level Fusion**. Qing Yi and Ken Kennedy. International Journal of High Performance Computing Applications, Vol 18, No.2, pages 237–253. SAGE journals. 2004.
- ◇ **Advanced Optimization Strategies in the Rice dHPF compiler**. John Mellor-Crummey, Vikram Adve, Bradley Broom, Daniel Chavarria-Miranda, Robert Fowler, Guohua Jin, Ken Kennedy and Qing Yi. Concurrency and Computation: Practice and Experience, 14(8-9): pages 741-767. Wiley. 2002.

Refereed Conference Publications

- ◇ **Modeling Optimization of Stencil Computations Via Domain-level Properties**, Brandon Nesterenko, Qing Yi, Pei-Hung Lin, Chunhua Liao, Brandon Runnels. PMAM '22: Proceedings of the Thirteenth International Workshop on Programming Models and Applications for Multicores and ManycoresApril 2022 Pages 35–44. doi:10.1145/3528425.3529103
- ◇ **An Adaptive Overlap-Pipelined Multitasking Superscalar Processor**, Mong T. Sim and Qing Yi, In the 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2020.
- ◇ **An Adaptive Multitasking Superscalar Processor**, Mong T. Sim and Qing Yi, In the 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 2019, pp. 1293-1299, doi: 10.1109/ICCC47050.2019.9064185.
- ◇ **Automating Non-Blocking Synchronization In Concurrent Data Abstractions**, Jiange Zhang, Qing Yi, and Damian Dechev. In The 34th IEEE/ACM International Conference on Automated Software Engineering (ASE 2019). Nov 2019. San Diego, CA, USA.
- ◇ **Transitioning Scientific Applications to using Non-VolatileMemory for Resilience** Brandon Nesterenko, Xiao Liu, Qing Yi, Jishen Zhao, and Jiange ZHang. In The International Symposium on Memory Systems (Memsys'2019). Sep 2019. Washington DC,USA.
- ◇ **Accelerating Parallel Graph Computing with Speculation,** Shuo Ji, Yinliang Zhao, and Qing Yi In The ACM International Conference on Computing Frontiers (CF'2019). May 2019. Alghero, Sardinia, Italy.
- ◇ **Improving Resource Utilization through Demand Aware Process Schedulings,** Brandon Nestenko, Qing Yi, and Jia Rao. In The 2018 International Conference on Parallel Processing (ICPP '18). Aug, 2018. Oregan, USA.
- ◇ **Compiler-Assisted Overlapping of Communication and Computation in MPI Applications.** Jichi Guo, Qing Yi, Jiayuan Meng, Junchao Zhang, and Pavan Balaji. In IEEE Cluster 2016, Sep 12-16, 2016. Taipei, Taiwan.
- ◇ **Characterizing and Optimizing the Performance of Multithreaded Programs Under Interference.** Yong Zhao, Jia Rao, and Qing Yi. In the 25th International Conference on Parallel Architectures and Compilation Techniques (PACT '16), September 11-15, 2016, Haifa, Israel.
- ◇ **Automatic Algorithm Selection in Computational Software Using Machine Learning.** Matthew C. Simpson, Qing Yi, and Jugal Kalita. In The 15th IEEE International Conference on Machine Learning and Applications (IEEE ICMLA'16). Dec 18-20, 2016. Anaheim, California, USA.

◇ **Just-in-time Component-wise Power and Thermal Modeling.** Shah Mohammad Faizur Rahman, Qing Yi and Houman Homayoun. In ACM International Conference on Computing Frontiers (CF'15). May 18-21, 2015. Ischia, Italy.

◇ **Automatic Detection of Information Leakage Vulnerabilities in Browser Extensions.** Rui Zhao, Chuan Yue and Qing Yi. In the 24th International World Wide Web Conference (WWW'15). May 18-22, 2015. Florence, Italy.

◇ **Specializing Compiler Optimizations Through Programmable Composition For Dense Matrix Computations.** Qing Yi, Qian Wang, and Huimin Cui. In The 47th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'14). Dec 13-17, 2014. Cambridge, UK.

◇ **Analytically Modeling Application Execution for Software-Hardware Co-Design.** Jichi Guo, Jiayuan Meng, Qing Yi, Vitali Morozov, and Kalyan Kumaran. In 28th IEEE International Parallel & Distributed Processing Symposium (IPDPS'14). May 19-23, 2014. PHOENIX, Arizona, USA.

◇ **AUGEM:Automatically Generate High Performance Dense Linear Algebra Kernels on x86 CPUs.** Qian Wang, Xianyi Zhang, Yunquan Zhang, and Qing Yi. In the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'13). Denver, CO. Nov, 2013.

◇ **Enhancing Performance Portability of MPI Applications Through Annotation-Based Transformations.** Md. Ziaul Haque, Qing Yi, James Dinan, and Pavan Balaji. In the 42nd International Conference on Parallel Processing (ICPP'13). Lyon, France. Oct, 2013.

◇ **Effective Use of Non-blocking Data Structures in a Deduplication Application.** Steven Feldman, Akshatha Bhat, Pierre Laborde, Qing Yi, and Damian Dechev. In The ACM International Conference on Systems, Programming, Languages and Applications: Software for Humanity (SPLASH'13). Wavefront experimence report. Indianapolis, USA. Oct, 2013.

◇ **Vectorization Past Dependent Branches Through Speculation.** Majedul Haque Sujon, R. Clint Whaley, and Qing Yi. In the 22nd International Conference on Parallel Architectures and Compilation Techniques (PACT'13). Edinburgh, Scotland. Sep, 2013.

◇ **A Highly Parallel Reuse Distance Analysis Algorithm on GPUs**. Huimin Cui, Qing Yi, Jingling Xue, Lei Wang, Yang Yang, and Xiaobing Feng. In 26th IEEE International Parallel & Distributed Processing Symposium (IPDPS'12). May 21-25, 2012. Shanghai, China.

◇ **Studying The Impact Of Application-level Optimizations On The Power Consumption Of Multi-Core Architectures**. S. Faizur Rahman, Jichi Guo, Akshatha Bhat, Carlos Garcia, Majedul H. Sujon, Qing Yi, Chunhua Liao, and Daniel Quinlan. In ACM International Conference on Computing Frontiers (CF'12). May 15-17, 2012. Cagliari, Italy.

◇ **Enhancing the Role of Inlining in Effective Interprocedural Parallelization**. Jichi Guo, Mike Stiles, Qing Yi, and Kleanthis Psarris. In International Conference On Parallel Processing (ICPP'11), Sep, 2011. Taipei, Taiwan.

◇ **Collective specification and verification of behavioral models and object-oriented implementations**. Qing Yi, Jianwei Niu, and Anitha R. Marneni. In International Conference On Software and Data Technologies (ICSOFT'11), July 18-21, Seville, Spain, 2011.

◇ **Understanding Stencil Code Performance On MultiCore Architectures**. S. Faizur Rahman, Qing Yi, and Apan Qasem. In ACM International Conference on Computing Frontiers (CF'11). May 3-5, 2011. Ischia, Italy.

◇ **Automated Programmable Control and Parameterization of Compiler Optimizations**. Qing Yi. In IEEE International Symposium on Code Generation and Optimization (CGO'11). Apr 02-06, 2011. Chamonix, France.

- **Automated empirical tuning of scientific codes for performance and power consumption**. S. F. Rahman, J. Guo, and Q. Yi. In High-Performance and Embedded Architectures and Compilers (HIPEAC'11), Pages 107–116, Heraklion, Greece, Jan 2011.
- **Exposing tunable parameters in multithreaded numerical code**. A. Qasem, J. Guo, F. Rahman, and Q. Yi. In 7th IFIP International Conference on Network and Parallel Computing (NPC'10), Best paper award, Pages 46–60, Zhengzhou, China, Sept. 2010.
- **Automated Transformation for Performance-Critical Kernels**. Qing Yi and R. Clint Whaley. ACM SIGPLAN Symposium on Library-Centric Software Design, Pages 109–119, Montreal, Canada. Oct, 2007.
- **Classification and Utilization of Abstractions for Optimization**. Dan Quinlan, Markus Schordan, Qing Yi, and Andreas Saebjornsen. The First International Symposium on Leveraging Applications of Formal Methods (ISoLA'04), Pages 57–73. Paphos, Cyprus, Oct, 2004.
- **Improving Memory Hierarchy Performance Through Combined Loop Interchange and Multi-level Fusion**. Qing Yi and Ken Kennedy. LACSI Symposium, Pages 237–253. Santa Fe, NM. Oct. 2002.
- **Transforming Loops To Recursion For Multi-Level Memory Hierarchies**. Qing Yi, Vikram Adve, and Ken Kennedy. ACM SIGPLAN conference of Programming Language Design and Implementation (PLDI'00), Pages 169–181, Vancouver, Canada. June. 2000.
- **High Performance Fortran Compilation Techniques for Parallelizing Scientific Codes**. Vikram Adve, Guohua Jin, John Mellor-Crummey, and Qing Yi. Supercomputing (SC'98), Pages 1–23. Orlando, FL, USA. Nov. 1998

Refereed Workshop Publications

- **Automating the Exchangeability of Shared Data Abstractions,** Jiange Zhang, Qian Wang, Qing Yi, and Huimin Cui. In The 31st International Workshop on Languages and Compilers for Parallel Computing (LCPC'18), Salt Lake City, UT, USA
- **Using Memory-style Storage to Support Fault Tolerance in Data Centers.** Xiao Liu, Qing Yi, and Jishen ZHao. In The 2016 USENIX Workshop on Cool Topics in Sustainable Data Centers (CoolDC '16). Mar 19, 2016. Santa Clara, Ca, USA.
- **Automatically Optimizing Stencil Computations on Many-core NUMA Architectures.** Pei-Hung Lin, Qing Yi, Daniel Quinlan, Chunhua Liao and Yongqing Yan. In The 29th International Workship on Languages and Compilers for Parallel Computing (LCPC 2016). Page 116-130. Sep 28-30, 2016. Rochester, NY, USA.
- **Interactive Composition Of Compiler Optimizations.** Brandon Nesterenko, Wenwen Wang, and Qing Yi. In The 28th International Workshop on Languages and Compilers for Parallel Computing (LCPC'15). Sep 9-11, 2015. Raleigh, NC, USA.
- **Extensive Parameterization And Tuning of Architecture-Sensitive Optimizations**. Qing Yi and Jichi Guo. In The Sixth International Workshop on Automatic Performance Tuning (iWapt'11). Proceedings of the Internaltional Conference on Computational Science (ICCS). Pages 2156-2165. June, 2011.
- **Improving Autotuning Efficiency and Portability Through Feedback Diagnostics**. Qing Yi, Santosh Sarangkar, and Apan Qasem. In The Fifth International Workshop on Automatic Performance Tuning (iWapt'10, position paper). Berkely, CA. June, 2010.
- **Automated timer generation for empirical tuning**. J. Magee, Q. Yi, and R. C. Whaley. In The 4th Workshop on Statistical and Machine learning approaches to ARchitecture and compilaTion (SMART'10). Pages 33–47. Pisa,Italy., Jan. 2010.
- **Exploring the Optimization Space of Dense Linear Algebra Kernels**. Qing Yi and Apan Qasem. In The 21th International Workshop on Languages and Compilers for Parallel Computing (LCPC'08), Pages 343–355, Edmonton, Canada. Aug, 2008.

- **POET: Parameterized Optimizations for Empirical Tuning.** Qing Yi, Keith Seymour, Haihang You, Richard Vuduc and Dan Quinlan. In Workshop on Performance Optimization for High-Level Languages and Libraries (POHLL'07). Pages 1-8. Mar 2007.
- **Annotating user-defined abstractions for optimization**. Dan Quinlan, Markus Schordan, Richard Vuduc, and Qing Yi. In Workshop on Performance Optimization for High-Level Languages and Libraries (POHLL'06), Pages 465-472, Rhodes Island, Greece. April 2006.
- **Toward the Automated Generation of Components from Existing Source Code**. Dan Quinlan, Qing Yi, Gary Kumfert, Thomas Epperly, Tamara Dahlgren, Markus Schordan, and Brian White. In The Second Workshop on Productivity and Performance in High-end Computing (PPEC'05), Pages 12-19, San Francisco, Feb, 2005.
- **Applying Data Copy to Improve Memory Performance of General Array Computations**. Qing Yi. In The 18th International Workshop on Languages and Compilers for Parallel Computing (LCPC'05), Pages 91–105, Hawthorne, New York. Oct 2005.
- **Applying Loop Optimizations to Object-oriented Abstractions Through General Classification of Array Semantics**. Qing Yi and Dan Quinlan. In The 17th International Workshop on Languages and Compilers for Parallel Computing (LCPC'04), Pages 253-267, West Lafayette, Indiana, USA. Sep. 2004.
- **Automatic Blocking Of QR and LU Factorizations for Locality**. Qing Yi, Ken Kennedy, Haihang You, Keith Seymour, and Jack Dongarra. In The Second ACM SIGPLAN Workshop on Memory System Performance (MSP'04), Pages 12-22, Washington, DC, USA. June. 2004.
- **Semantic-Driven Parallelization of Loops Operating on User-Defined Containers**. Dan Quinlan, Markus Schordan, Qing Yi and Bronis de Supinski. In The 16th Annual Workshop on Languages and Compilers for Parallel Computing (LCPC'03). Pages 524-538. College Station, TX, USA. Oct. 2003.
- **A C++ infrastructure for Automatic Introduction and Translation of OpenMP Directives**. Dan Quinlan, Markus Schordan and Qing Yi. In Workshop on OpenMP Applications and Tools (WOMPAT'03), Pages 13-25, Toronto, Ontario, Canada. June. 2003

Other Publications

- **Book chapter: Optimizing And Tuning Scientific Codes**. Qing Yi. SCALABLE COMPUTING AND COMMUNICATIONS: THEORY AND PRACTICE. Samee U. Khan, Lizhe Wang, and Albert Y. Zomaya. Wiley-IEEE Computer Society Press. Jan, 2013. ISBN: 978-1-1181-6265-1

Services Professional

- **Program Committee member of conferences:** SC'23, SC'19, CC'19, HiPC'19, PACT'18, ICS'18, SC'17, HPCC'17, IPDPS'2017, CCGrid'2017, SC'2016, CCGrid'2016, CGO'2016, HiPC'2016, ICPADS'2016, ICPP'2015, APPT'2015, CC'2014, IPDPS'2014, IPDPS'2013, PACT'2013, APPT'2013, ICS'2012, ICPP'2012, NPC'2012, CloudCom'2012, ICS'2011, NPC'2011, NPC'2010, PACT'2009, CGO'2009.
- **NSF Panelist**. 2024, 2022, 2017, 2015, 2012, 2011, 2010, 2009. NSF proposal external reviewer, 2015, 2014.
- **Technical program track chairs:** vice-chair of Programming Systems, The International Conference for High Performance Computing, Networking, Storage and Analysis (SC) 2017. Program track-chair in the area of Programming Models and Runtime Systems, CCGrid, 2015.
- **Extended Review Committee member: ISCA'2019**
- **Program Committee member of workshops:** PAMLA'2018, PMAM'2018, PMAM'2017, PMAM'2016, LCPC'2015, PMAM'2015, PMAM'2014, IWAPT'2011, SMART'2011, GPUScA'2011, GPUScA'2010, SMART'2010, POHLL'2008, POHLL'2007, MSPC'2006.

- **External Reviewer for journals:** Transactions on Cloud Computing (2018) Parallel Computing (2006,2008-2018); Journal of Parallel and Distributed Computing (JPDC) (2015,2016); Concurrency and Computation: Practice and Experience (2016,2014); ACM Transactions on Architecture and Code Optimization (2014,2015,2016); Scientific Programming (2015), Transactions on Embedded Computing Systems (2015), International Journal of Parallel, Emergent and Distributed Systems(2013), Journal of Parallel and Distributed Processing(2013), Software: Practice and Experience (2009,2010,2011,2012); Journal of Computational Science (2010); Journal of Signal Processing Systems(2009); IEEE Transactions on Parallel and Distributed Systems (2006).
- **External Reviewer for conferences:** CC'2013, PPOPP'2011, IPDPS'2010, ICS'2006, EuroPar'2004, IPDPS'2004.
- **Poster chair, International Conference on High-Performance And Embedded Architectures and Compilers**. Year 2012 and 2013.
- **Local arrangements chair**, ACM SIGPLAN annual symposium on Principles and Practices of Parallel Programming, San Antonio, TX. 2011.
- **Book Review:** *Crafting a Compiler* by Charles Fischer, Ron Cytron,and Richard LeBlanc. The Second Edition. Addison Wesley. 2009.