Minjia Zhang

Microsoft Building 99
14820 NE 36th St
Redmond, WA 98052, USA

minjiaz@microsoft.com
614-940-0520
website: zhangminjia.me

Education

Doctor of Philosophy (Ph.D.) **Ohio State University** ***Graduated*** **August 2016**

- **Major:** Computer Science and Engineering GPA: 3.66
- **Research Focus:** Concurrency and Parallelism, Efficient Runtime systems, Distributed Systems
- **Advisor:** Dr. Michael D. Bond

Master of Science *(M.S.)* HUST **Graduated August 2010**

- **Major:** Computer Science

Bachelor of Science (B.S.) HUST ***Graduated June 2008***

- **Major**: Computer Science and Technology **GPA:** 3.87

Professional Experience

Microsoft Research, Redmond, WA **10/2016 – present**

Senior Research Software Development Engineer
Manager: Yuxiong He
Working on building efficient and scalable artificial intelligence infrastructure.
Worked on optimizing the serving latency of numerical answer generation for web Q&A scenarios and achieved 10X speedup comparing with the existing severing approach.

Microsoft Research, Redmond, WA **05/2016 – 08/2016**

Research Intern
Mentors: Kathryn McKinley, Yuxiong He, Sameh Elnikety
Worked on supporting global snapshot transactions in SQL Data Warehouse.

- Worked on the design of supporting global snapshot transactions using logical counter and centralized time authority.
- [C#/C++] Implemented and integrated global snapshot transactions in SQL DW.

Programming Language and Software System Lab, Ohio State University **05/2011 – 05/2016**

Research Assistant
Advisor: Michael D. Bond
Hardware is becoming more parallel. Writing concurrent programs to utilize parallel hardware is hard and error-prone. Exploring and building efficient runtime systems and runtime analysis to make complex, concurrent programs more reliable and scalable. To achieve the best performance and scalability, exploited optimizations opportunities in managed runtime: JIT compiler, garbage collector, adaptive system, and utilized both static analysis and dynamic analysis.

- Transactional memory helps improve the programmability of writing concurrent programs. Designed and built software transactional memory systems with strong isolation, low overhead, and strong progress guarantees. Demonstrated that the system was significantly faster than existing three state-of-the-art STMs [PPoPP 2015].
- Introduced a novel relaxed dependence tracking mechanism for capturing and enforcing cross-thread dependences. Built RT, and two client analysis based on RT: a dependence recorder and an STM system. RT reduced 49% execution time on average and achieved 6X maximally speedup [CC 2016].
- Worked on data ownership locking, non-blocking synchronization, static analysis to eliminate redundant instrumentation in dependence tracking with biased reader-writer lock (Octet) [OOPSAL 2013].
- Worked on the speculation approach to support the SBRS memory model. [ASPLOS 2015].
- Worked on the online & offline profiling in runtime system in hybrid tracking [PPoPP 2016].
- Strong memory models that throw consistency exceptions trade one problem (undefined semantics) for another

(poor availability). Showed how to improve availability while still providing well-defined semantics for programs with data races, including how to relax semantics in a principled way for better availability. [ISMM 2017].

Microsoft Research, Redmond, WA **05/2015 – 11/2015**

Research Intern

Mentors: Kathryn McKinley, Sameh Elnikety, Yuxiong He, Srikumar Rangarajan

Microsoft is expanding its cloud service. SQL Azure is a scalable, globally distributed cloud database service from Microsoft. Built, optimized and evaluated a mechanism that helped improve the availability, performance and semantics of SQL Azure's Elastic Query service.

- [C#/C++] Designed and implemented global-snapshot and session-consistency for availability and read-scaling.
- Supported requested snapshots by modifying the transaction and multi-versioning subsystems in SQL Server.
- Supported session-consistency by modifying the HaDr and log-shipping subsystems.
- Integrated requested snapshots, loosely synchronized clocks, the session-consistency support, a custom load-balancer, and the EQ frontend to provide global-snapshot and session-consistent reads in the production code.
- Worked on implementing Clock-SI on top of global transactions.
- Set up a simulated, distributed virtual cluster and conducted performance evaluation that demonstrated the benefit of both read-scaling and global snapshot.
- System demo was selected to appear at the SQL Server MVP Tell & Show event.

Network Based Computing Lab, Ohio State University **09/2010 – 03/2011**

Research Assistant

Mentor: Dhabaleswar K. Panda

Memcached is an important component for large-scale data processing. However, memcached is written with sockets and do not deliver best performance on datacenters with high performance networks. Performed case studies on the new design of memcached with high performance interconnects and demonstrated its associated benefits.

- [C] Examined the challenges in redesigning the network and I/O part of memcached with InfiniBand and RDMA. Changed Memcached to utilize one-sided RDMA reads and writes over InfiniBand. Improved the throughput significantly and reduced memcched's latency to the scale of μs. [ICPP 2011]

Service Computing Technology and System Lab & Cluster and Grid Computing Lab, HUST **09/2008 – 08/2010**

Research Assistant

Advisor: Hai Jin Mentors: Song Wu, Xuanhua Shi

VNIX is a three-layer platform for management of virtualization resource in a distributed and cloud computing environment. Added new functionalities and optimizations to help administrators to manage a large number of distributed VMs. [ICPADS 2010]

- [JavaScript] Implemented the front end of virtual network management in VNIX.
- [C] Reduced the average data transferring size in live migration by 44.1% by compressing dirtied memory with LZO compression algorithm.

Center for Biomedical Imaging and Bioinformatics, HUST **11/2006 – 09/2007**

Undergraduate Researcher

Mentors: Enmin Song, Renchao Jin

The human genome project constantly requires searching common segmentations of tens of thousands of genes. It is therefore important to find a fast algorithm to do common genome segmentation search.

- [C] Proposed and implemented several substring algorithms (including Ukkonen's suffix tree algorithm) to solve this problem, and proved that the genome classification problem could be solved in linear time complexity.

Refereed Conference Publications

- **Minjia Zhang**, Swarnendu Biswas, Michael Bond, "Avoiding Consistency Exceptions Under Strong Memory Consistency Models", In 2017 ACM SIGPLAN International Symposium on Memory Management (ISMM 2017), June 2017, Barcelona, Spain
- Swarnendu Biswas, Man Cao, **Minjia Zhang**, Michael Bond and Ben Wood, "Lightweight Data Race Detection

for Production Runs", In 26th International Conference on Compiler Construction (CC), February 2017.

- **Minjia Zhang**, Swarnendu Biswas, Michael D. Bond, "Relaxed Dependence Tracking for Parallel Runtime Support", In 25th International Conference on Compiler Construction (CC), March 2016.
- Man Cao, **Minjia Zhang**, Aritra Sengupta, and Michael D. Bond, "Drinking from Both Glasses: Combining Pessimistic and Optimistic Tracking of Cross-Thread Dependences", In ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), March 2016.
- Swarnendu Biswas, **Minjia Zhang**, Michael D. Bond, and Brandon Lucia, "Valor: Efficient, Software-Only Region Conflict Exceptions" (**Distinguished Artifact Award, Distinguished Paper Award**), In ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), October 2015.
- **Minjia Zhang**, Jipeng Huang, Man Cao, and Michael D. Bond, "Low-Overhead Software Transactional Memory with Progress Guarantees and Strong Semantics", In ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), February 2015.
- Aritra Sengupta, Swarnendu Biswas, **Minjia Zhang**, Michael D. Bond, and Milind Kulkarni, "Hybrid Static-Dynamic Analysis for Statically Bounded Region Serializability", In ACM Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), March 2015.
- Man Cao, **Minjia Zhang**, and Michael D. Bond, "Drinking from Both Glasses: Adaptively Combining Pessimistic and Optimistic Synchronization for Efficient Parallel Runtime Support", In the 5th Workshop on Determinism and Correctness in Parallel Programming (WODET), March 2014.
- Michael D. Bond, Milind Kulkarni, Man Cao, **Minjia Zhang**, Meisam Fathi Salmi, Swarnendu Biswas, Aritra Sengupta, and Jipeng Huang "Octet: Capturing and Controlling Cross-Thread Dependences Efficiently ", In ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), October 2013.
- Jithin Jose, Hari Subramoni, Miao Luo, **Minjia Zhang**, Jian Huang, Md. Wasi-ur-Rahman, Nusrat S. Islam, Xiangyong Ouyang, Hao Wang, Sayantan Sur, and D. K. Panda, "Memcached Design on High Performance RDMA Capable Interconnects", In International Conference on Parallel Processing (ICPP), September 2011.
- **Minjia Zhang**, Hai Jin, Song Wu, Xuanhua Shi "VirtCFT: A Transparent VM-level Fault-Tolerant System for Virtual Clusters", In International Conference on Parallel and Distributed System (ICPADS), Dec 2010.

Journal Publications

- Man Cao, **Minjia Zhang**, Aritra Sengupta, Swarnendu Biswas, and Michael D. Bond, "Hybridizing and Relaxing Dependence Tracking for Efficient Parallel Runtime Support", In ACM Transactions on Parallel Computing (TOPC), April 2017.

Other Publications

Technical report:

- **Minjia Zhang**, Swarnendu Biswass, Michael D. Bond, "All That Glitters is Not Gold: Improving Availability and Practicality of Exception-Based Memory Models, Technical Report #OSU-CISRC-4/16-TR01, Computer Science & Engineering, Ohio State University, Apr 2016.
- **Minjia Zhang**, Swarnendu Biswass, Michael D. Bond, Optimizing Parallel Runtime Support with Asynchronous Coordination, Technical Report #OSU-CISRC-11/15-TR23, Computer Science & Engineering, Ohio State University, Nov 2015.
- Swarnendu Biswas, **Minjia Zhang**, Michael D. Bond, and Brandon Lucia, "Efficient, Software-Only Data Race Exceptions", Technical Report #OSU-CISRC-3/15-TR04, Computer Science & Engineering, Ohio State University, Mar 2015.
- Swarnendu Biswas, **Minjia Zhang**, and Michael D. Bond, "Lightweight Data Race Detection for Production Runs", Technical Report #OSU-CISRC-1/15-TR01. Computer Science & Engineering, Ohio State University, Jan 2015.
- **Minjia Zhang**, Jipeng. Huang, Man Cao, and Mike D. Bond. LarkTM: Efficient, strongly atomic software transactional memory. Technical Report OSU-CISRC-11/12-TR17, Computer Science & Engineering, Ohio State University, Nov 2012.

Thesis

- **Minjia Zhang**, "Efficient and Scalable Runtime Support for Parallelism", PhD's Dissertation, Ohio State University, USA, December, 2016
- **Minjia Zhang,** "A Study of Fault-Tolerance for Virtual Clusters with Coordinated Checkpointing", Master's Thesis, Huazhong University of Science and Technology, China, June, 2010
- **Minjia Zhang,** Designing and Development of an Integrated Web Services Framework for Digital Business Card, Bachelor's Thesis, Huazhong University of Science and Technology, China, June, 2008

Short paper:

- **Minjia Zhang**, Swarnendu Biswas, Michael D. Bond, "On the Problem of Consistency Exceptions in the Context of Strong Memory Models", PPoPP' 17 Poster paper, February, 2017, Austin, Texas, USA
- **Minjia Zhang**, "SIRe: An Efficient Snapshot Isolation-based Memory Model for Detecting and Tolerating Region Conflicts", SPLASH '15 Companion, October, 2015, Pittsburgh, PA, USA

Presentations

- Presented work on detecting and tolerating region conflicts to support region snapshot isolation at ACM Student Research Competition, OOPSLA 2015, Pittsburg, PA, USA
- Presented work on low-overhead and scalable software transactional memory with strong progress guarantees at the 20st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), 2015, San Francisco, CA, USA
- Presented work on building efficient, strongly atomic, and scalable software transactional memory at ACM Student Research Competition, PLDI 2013, Seattle, WA, USA

Honors and Awards

- Proposed project was accepted by the very first Microsoft AI School as one of 10 projects out of 535 total project proposals (<2 % acceptance rate) 2017
- Bronze medal in SPLASH 2015 Student Research Competition 2015
- OOPSLA'15 Distinguished Paper Award 2015
- OOPSLA'15 Distinguished Artifact Award 2015
- NSF Travel Grants to attend PPoPP'15, PACT'15, SPLASH'15
- Granted the Artifact Evaluation badge by the PPoPP 2015 Committee 2015
- Silver medal in PLDI 2013 Student Research Competition 2013
- Awarded "University Fellowship", graduate student fellowship, OSU 2010, 2011,2013
- Awarded Chinese Government Scholarship (top 2%), HUST 2008
- Cumulatively ranked sixth in the Department of Computer Science and Technology among 500+ students. 2008
- Awarded Teyiu (Honor) Student at HUST (<2%) 2007
- Awarded merit certificate and scholarship in 2005, 2006, 2007 (7 semesters) for academic performance 2005-2007
- Exempted from National College Entrance Examination (NCEE) in China for excellent academic performance (< 0.3%) 2004
- Awarded 1st Prize in Chinese National Math Olympiad Competition 2004

Service

- Reviewer, the 14th IEEE International Conference on Automatic Computing 2017
- Reviewer, Journal of Concurrency and Computation: Practice and Experience 2017
- Committee Member, PLDI 2017 Artifact Evaluation
- Subreviewer, 7th Workshop on the Theory of Transactional Memory (WTTM 2016)
- Committee Member (final rational lead), SPLASH 2015 Artifact Evaluation
- Committee Member , PLDI 2015 Artifact Evaluation
- ACM student member

Technical Skills

- **Programming:** Java, C/C++, C#, Python, Matlab, SQL, 5 years of experience in developing low-overhead runtime support systems for concurrent programs in JVM.
- **Deep Learning Framework: Tensorflow, CNTK**
- **Distributed Enterprise Computing**: EJB, Mobile computing (Android), XHTML, CSS, JSP, JavaScript, and AJAX.
- **Data Processing:** SQL Server, DB2, MySQL, HDFS, HBase, and Memcached.
- **Benchmarks:** DaCapo, Java Grande, STAMP, TPC-C, TPC-H, YCSB
- **Tools and Libraries:** OpenMP, Intel MKL, Multi-Shard Query library, Virtualization technology (Xen), Microsoft technology stack (C#, .NET, TPL, TFS, VS, XTS, MSBuild, Windbg, BuddyTest, Azure).