

# Zhaoguo Wang

60 5th Ave, Room 418  
New York, NY, United States  
✉ zhaoguo@nyu.edu

---

## Academic Positions

- 9/2017 - now **Assistant Professor / Faculty Fellow**,  
New York University, Department of Computer Science.
- 9/2014 - 8/2017 **Postdoctoral Researcher**,  
New York University, Department of Computer Science, Supervised by **Jinyang Li**.

---

## Education

- 9/2011 - 6/2014 **Ph.D., Computer Systems**  
*School of Computer Science, Fudan University*, Advisor: **Binyu Zang, Haibo Chen, Jinyang Li**,  
Research area: *Multicore In-Memory Database*.
- 9/2008 - 6/2011 **M.S., Computer Software and Theory**  
*Software School, Fudan University*, Advisor: **Binyu Zang**,  
Research area: *Scalable Full System Emulation*.
- 9/2004 - 6/2008 **B.S., Software Engineering**  
*Software School, Nanjing University*.

---

## Academic Services

- 2017 Communications of the ACM, Reviewer
- 2017 ACM Transactions On Storage (TOS), Reviewer
- 2017 Usenix Annual Technical Conference (Usenix ATC), External Reviewer
- 2017 European Conference on Computer Systems (EuroSys), External Reviewer
- 2011 ACM Asia-Pacific Workshop on Systems (APSys), Local Arrangement Assistant

---

## Teaching

- 2017 Fall Computer System Organization at NYU (CSCI-UA.0201), Lecturer
- 2010 Fall Operating Systems at Fudan, Teaching Assistant
- 2010 Spring Introduction to Computer Systems at Fudan, Teaching Assistant

---

## Publications

### Conference Paper

- 2017 **Zhaoguo Wang**, Lamont Nelson, Changgeng Zhao, Jinyang Li, Shuai Mu. “Paxi: Taking a Raft from Paxos to Anti-Paxos.” *Under submission*
- 2017 Xiaozhou Zhou, **Zhaoguo Wang**, Rong Chen, Haibo Chen, Jinyang Li. “Extract more intra-transaction parallelism with work stealing for OLTP workloads.” In *Proceedings of the 8th ACM Asia-Pacific Workshop on Systems (APSys’17)*. **Best Paper Award**.
- 2017 Xin Wang, Weihua Zhang, **Zhaoguo Wang**, Ziyun Wei, Haibo Chen, Wenyun Zhao. “Eunomia: Scaling Concurrent Search Trees under Contention Using HTM.” In *Proceedings of the 21th ACM symposium on Principles and practice of parallel programming (PPoPP’17)*

- 2016 **Zhaoguo Wang**, Shuai Mu, Yang Cui, Han Yi, Haibo Chen, Jinyang Li. “Scaling Multi-core Databases via Constrained Parallel Execution.” In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. (SIGMOD’16)
- 2015 Chien-Chin Huang, Qi Chen, **Zhaoguo Wang**, Russell Power Jorge Ortiz Jinyang Li and Zhen Xiao. “Spartan: A Distributed Array Framework with Smart Tiling.” In *Proceedings of USENIX Annual Technical Conference*. (Usenix ATC’15)
- 2015 **Zhaoguo Wang**, Han Yi, Ran Liu, Mingkai Dong and Haibo Chen. “Persistent Transactional Memory.” In *Proceedings of IEEE Computer Architecture Letters 14.1 (2015)*: 58-61. (CAL’15)
- 2014 **Zhaoguo Wang**, Hao Qian, Jinyang Li and Haibo Chen. “Using Restricted Transactional Memory to Build a Scalable In-Memory Database.” In *Proceedings of the European Conference on Computer Systems*. ACM, 2014. (EuroSys’14)
- 2013 **Zhaoguo Wang**, Hao Qian, Haibo Chen, and Jinyang Li. “Opportunities and pitfalls of multicore scaling using hardware transaction memory.” In *Proceedings of the 4th Asia-Pacific Workshop on Systems*, p. 3. ACM, 2013. (APSys’13)
- 2011 **Zhaoguo Wang**, Ran Liu, Yufei Chen, Xi Wu, Haibo Chen, Weihua Zhang, and Binyu Zang. “COREMU: a scalable and portable parallel full-system emulator.” In *Proceedings of the 16th ACM symposium on Principles and practice of parallel programming*, pp. 213-222. ACM, 2011. (PPoPP’11)

#### Journal Article

- 2017 Haibo Chen, Heng Zhang, Mingkai Dong, **Zhaoguo Wang**, Yubin Xia and Binyu Zang. “Efficient and Available In-memory KV-Store with Hybrid Erasure Coding and Replication.” To appear in *ACM Transactions on Storage (TOS)* 2017
- 2017 Haibo Chen, Rong Chen, Xingda Wei, Jiabin Shi, Yanzhe Chen, **Zhaoguo Wang** and Binyu Zang. “Fast In-memory Transaction Processing using RDMA and HTM.” *ACM Transactions on Computer Systems (TOCS)*, Volume 35, Issue 1, Article No. 3, July 2017
- 2017 Weihua Zhang, Xin Wang, Shiyu Ji, Ziyun Wei, **Zhaoguo Wang**, Haibo Chen. “Scaling Concurrent Index Structures under Contention Using HTM.” *IEEE Transactions on Parallel and Distributed Systems (TPDS)* July 2017, DOI: 10.1109/TPDS.2017.2729551, available at <http://ieeexplore.ieee.org/document/7987039>
- 2017 Rong Chen, Youyang Yao, Peng Wang, Kaiyuan Zhang, **Zhaoguo Wang**, Haibing Guan, Binyu Zang and Haibo Chen. “Replication-based Fault-tolerance for Large-scale Graph Processing.” *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, May 2017, DOI: 10.1109/TPDS.2017.2703904, available at <http://ieeexplore.ieee.org/document/7927721>

---

#### Talks

##### **Scaling Multicore Databases via Constrained Parallel Execution**

SIGMOD 2016, June 2016, San Francisco, USA  
New York University, June 2016, New York City

##### **Using Restricted Transactional Memory to Build a Scalable In-Memory Database**

EuroSys 2014, April 2014, Amsterdam, Netherlands  
Microsoft Research Asia, February 2014, Beijing, China

##### **Opportunities and pitfalls of multicore scaling using hardware transaction memory**

APSys 2013, July 2013, Singapore  
ChinaSys 2013, May 2013, Beijing, China

##### **COREMU: a scalable and portable parallel full-system emulator**

Microsoft Research Asia, June 2011, Beijing, China  
PPoPP 2011, February 2011, San Antonio, USA

---

## Selected Projects

2/2017 - now **Optimize Raft In Wide-Area,**  
In Progress.

In recent years, Raft has overtaken Paxos as the consensus algorithm of choice for implementing state machine replication in the industry. Students and developers new to consensus tend to find Raft more understandable than Paxos, partly because Raft is specified at pseudo-code level precision while Paxos handwaves many engineering details. However, most existing literatures of Raft are targeting on the usages in local datacenter. This project proposes multiple optimizations in the wide-area setup. They reduce the request latency under low client load, while improve the throughput with high client load.

10/2014 - 8/2017 **Scalable Database Systems for High Contention Workload,**  
SIGMOD'16, PPOPP'17, TPDS'17, APSys'17.

Multicore in-memory database often relies on traditional concurrency control schemes such as two-phase locking (2PL) or optimistic concurrency control (OCC). When the workload exhibits a non-trivial amount of contention, both 2PL and OCC sacrifice much parallel execution opportunity. I use years to build scalable database systems targeting on high contention workload in both transaction layer and storage layer. In the transaction layer, I propose two concurrency control protocols to unleash the parallelism: IC3 (SIGMOD 16) and StealDB (APSys 17). IC3 combines the static analysis of the transaction workload with runtime techniques that track and enforce dependencies among concurrent transactions, while StealDB introduces a scalable work stealing algorithm for 2PL to leverage intra-transaction parallelism. In the storage layer, I propose a new design of concurrent search tree based on HTM (PPOPP'17, TPDS'17). It contains several principles to reduce HTM aborts, including splitting HTM regions with version-based concurrency control to reduce HTM working sets, and so on. All these lead to 3X-10X speedup under high contention, comparing with existing systems.

4/2013 - 6/2016 **Using Hardware Transaction Memory to Scale In-Memory Storage Systems,**  
APSys'13, EuroSys'14, CAL'15.

Recently, an exciting new hardware feature called hardware transaction memory (HTM) becomes available on Intel processors. HTM provides a set of transaction primitives that promise to ease the development of high performance concurrent software. Does HTM live up to its performance promise? I use three years to answer this question from building system with HTM to designing new HTM hardware. First, I built a concurrent skiplist using HTM to understand HTM performance caveats (APSys'13). This work shows that, by carefully limiting the size of a HTM transaction, we can drastically simplify the implementation of concurrent data structures with high performance. Then, I moved beyond data structures to scale a scalable multicore in-memory database (EuroSys'14). Several design strategies are used to shrink the HTM transaction size. It achieves 40% performance improvement than existing scalable system based on fine-grained locking. My last project on HTM is proposing a new design of HTM in the hardware layer, Persistent transaction memory (PTM). It adds durability to transaction memory (TM) by incorporating with the emerging non-volatile memory (CAL'15).

3/2009 - 11/2010 **COREMU: a Scalable and Portable Parallel Full-system Emulator,**  
PPOPP'11, [sourceforge.net/p/coremu/](http://sourceforge.net/p/coremu/).

COREMU is a scalable and portable parallel emulation framework that decouples the complexity of building a mature sequential emulator from providing a parallelized version. The key observation is that CPU cores and devices in current (and likely future) multiprocessors are loosely-coupled and communicate through welldefined interfaces. Based on this observation, COREMU emulates multiple cores by creating multiple instances of existing sequential emulators, and uses a thin library layer to handle the inter-core and device communication and synchronization, to maintain a consistent view of system resources.

---

## Students Mentoring

2015 - now Lamont Nelson, NYU Ph.D. student

I mentored Lamont when he worked with us on improving the performance of distributed transaction. Lamont was awarded the generous NSF Graduate Research Fellowship.

- 2015 - now Quanlu Zhang, Peking University Ph.D. student  
I mentored Quanlu when he worked with us on building a fast recovery in-memory database. Quanlu is going to join Microsoft Research Asia this year.
- 2015 - 2016 Xin Wang, Fudan Master student  
I mentored Xin when he worked with us on building a HTM friendly B+Tree (PPoPP'17). Xin is going to join IBM research this year.
- 2014 - 2015 Yang Cui, NYU Master student  
I mentored Yang when he worked with us on designing scalable concurrency control protocol (SIGMOD'16). Yang is working at Google now.
- 2014 - 2015 Chien-Chin Huang, NYU Ph.D. student  
I mentored Chien-Chin when he worked with us on building a distributed machine learning framework (Usenix ATC'15).
- 2011 - 2014 Han Yi, Fudan Master student  
I mentored Han when he worked with us on building high performance databases (SIGMOD'16, CAL'15). Han is working at Meituan.com now.
- 2011 - 2014 Hao Qian, SJTU Master student  
I mentored Hao when she worked with us on using HTM to improve the performance of in-memory database (Eurosys'14). Hao is working at Microsoft now.

---

## Awards

- 2014 Outstanding graduate student of Fudan University  
2011 Shanghai outstanding graduate student