Yihe Huang, Ph.D.

Santa Clara, CA – United States of America

 +1 (734) 834-3656 • ✉ hzwilliamhuang@gmail.com
 🌐 huangyihe.github.io

I am a researcher and software engineer in *systems software*. My received my Ph.D. from Harvard University, where I was advised by Professor Eddie Kohler [2]. I have broad research interests in systems. My research covers operating systems, high-performance runtime systems, distributed systems, database systems, and software systems for emerging hardware. I currently work at Google's Sunnyvale, CA office as a software engineer on the Spanner database team.

I go by my preferred name "William". Fun fact: my legal first name Yihe （以何） means "why" in Classical Chinese and pronounces like "yee-heh". Pronouns: he/him/his.

Work Experience

Google **Sunnyvale, CA, United States**
Software Engineer *July 2020–Present*

I work on Spanner, Google's planet-scale distributed database for Google's internal and Cloud customers. I work on distributed concurrency control and transaction processing, a core component of the system. My mission is to make Spanner transactions more scalable and easy-to-use for our customers. Duties include investigating and understanding the suboptimal system performance encountered in production due to complex interaction between the workload and different system components, and proposing and executing on architectural improvements to the system to address these issues.

Prior Work Experience

Google (Brain and Cloud) **Sunnyvale, CA, United States**
Software Engineering Intern *May 2019–August 2019*

Designed and implemented improved data transfer mechanisms for Tensorflow using RDMA-like semantics and user-level networking. Vastly improved data processing efficiency.

Microsoft **Cambridge, United Kingdom**
Research Intern *May 2018–August 2018*

Engineering the multi-version concurrency control (MVCC) [4] and garbage collection subsystem of the FaRM distributed database [5]. Designed and implemented a new garbage collection routine so that long-running read-only transactions no longer hold up system-wide MVCC garbage collection. This enables the system to schedule long analytic queries with concurrent update transactions without blocking/rollbacks or memory-resource-induced deadlocks.

Oracle **Burlington, MA, United States**
Research Intern *June 2016–August 2016*

Explored performance characteristics and design trade-offs of software systems designed for emerging byte-addressable, non-volatile memory devices. Designed and implemented caching and parallel write-ahead logging schemes for NoSQL databases to efficiently utilize non-volatile memory for persistence. Work published in USENIX ATC'18 and patent granted in July 2019.

Skills

- Years of experience developing software in modern C++.
- Years of experience in profiling and understanding the performances of C++ programs.
- Expert in database transaction processing, concurrency control, and transactional memory.
- Extensive experience in parallel programming, concurrency, and synchronization.

- Experience working with emerging technology such as nonvolatile random access memory (NVRAM) and RDMA (fast kernel-bypass networking).
- Proficient in other major programming languages including C, Python, Go, JavaScript.
- Experienced user of development tools like Git and various UNIX utilities and toolchains (compilers and build systems).
- Proficient in typesetting in LaTeX.
- Knowledgeable in concepts such as computer architecture, performance engineering, operating system kernels, file systems, networking, databases, and distributed systems.

Education

Harvard University   **Cambridge, MA, United States**
Ph.D. & S.M. in Computer Science   *September 2014–May 2020*

My thesis studies concurrency control performance in main-memory database systems. The goal is to design a transaction processing system that can understand the workload and use workload-specific information to reduce false contention. My work also concerns applying emerging persistent memory technology to key-value stores and transaction processing systems.

University of Michigan   **Ann Arbor, MI, United States**
B.S.E. in Computer Engineering   *September 2012–May 2014*

Graduated with a 3.9/4 GPA and started on research projects on computer systems.

Shanghai Jiao Tong University   **Shanghai, China**
B.S.E. in Electrical and Computer Engineering   *September 2010–August 2014*

Graduated with a 3.8/4 GPA from a program with strong emphasis on both theoretical foundations and large-scale, industry-sponsored capstone design projects.

Notable Projects

Publications

- **Opportunities for Optimism in Contended Main-Memory Multicore Transactions** *Published at the 46th International Conference on Very Large Data Bases (VLDB'20).* **Best Paper Award.** *With co-authors William Qian, Eddie Kohler, Barbara Liskov, and Liuba Shrira, Brandies/Harvard/MIT*

  Identified problems in prior measurements on optimistic concurrency control (OCC)'s performance in high contention workloads, and proposed semantics optimizations to OCC that improves its throughput under high contention by 5x, outperforming more sophisticated CC mechanisms.

  Available at `https://doi.org/10.14778/3377369.3377373`

- **On Main-memory Multi-core Transaction Performance** *Poster and talk at the Student Research Competition at the 27th ACM Symposium on Operating Systems Principles (SOSP'19).*

  An investigation into secondary, non-feature implementation factors that turn out to have nontrivial and significant impact on transaction processing performance in modern main-memory database systems.

- **Closing the Performance Gap Between Volatile and Persistent Key-Value Store Using Cross-Referencing Logs** *Published at the 2018 USENIX Annual Technical Conference (ATC'18). With co-authors Matej Pavlovic, Virendra J. Marathe, Margo Seltzer, Tim Harris, Steve Byan, Harvard University/EPFL/Oracle Corporation*

  In this paper we introduced Bullet, a key-value data structure store designed for byte-addressable persistent memory. Bullet uses a DRAM front end and a novel data structure called cross-referenced logs to manage low-latency reads and parallel writes to persistent data.

  Available at `https://www.usenix.org/conference/atc18/presentation/huang`

- **Type-aware Transactions for Faster Concurrent Code** *Published at the Eleventh European Conference on Computer Systems (EuroSys'16). With co-authors Nathaniel Herman, Jeevana Priya Inala, Lillian Tsai, Eddie Kohler, Barbara Liskov, and Liuba Shrira, Brandeis/Dropbox/Harvard/MIT*

  In this paper we introduced STO, a fast type-based transactional memory system that provides composable and

efficient transactional semantics to data structures.

Available at `https://dx.doi.org/10.1145/2901318.2901348`

- **The Impact of Timestamp Granularity in Optimistic Concurrency Control** *ArXiv article. With co-authors Hao Bai, Eddie Kohler, Barbara Liskov, Liuba Shrira, Brandies/Harvard/MIT*

  Performance study signifying the importance of controlling granularity, a very old and well-known concept, in the context of modern main-memory transaction processing.

  Available at `https://arxiv.org/abs/1811.04967`

- **Persistent Memory Transactions** *ArXiv article and poster at the 2017 USENIX Annual Technical Conference (ATC'17). With co-authors Virendra J. Marathe, Achin Mishra, Amee Trivedi, Faisal Zaghloul, Sanidhya Kashyap, Margo Seltzer, Tim Harris, Steve Byan, Bill Bridge, and Dave Dice, Georgia Tech/Harvard/IBM/Oracle Corporation/UMass Amherst/Yale*

  Thorough evaluation of performance trade-offs in different persistent transaction runtime implementations.

  `https://arxiv.org/abs/1804.00701`
  `https://www.usenix.org/conference/atc17/poster-session`

Patents

- **Data Structure Store in Persistent Memory**, United States Patent US20180260324A1
  Filed with Virendra Marathe, Margo Seltzer, Steve Byan, assignee Oracle International Corporation.

  Patent granted on July 23, 2019. Available to view at `https://patents.google.com/patent/US20180260324A1`.

Past Research Projects

- **SLYKey: A Transparent Peer-to-peer Public Key Directory** *with Lilian Tsai, Serena Wang, and Prof. James Mickens, Harvard University*

  By applying the ideas behind the Bitcoin blockchain, we aimed to build a public key infrastructure that makes end-to-end secure communication actually tractable and transparent to public audits. A prototype was implemented in the Go programming language with a complete protocol interface plus a solution for blockchain synchronization issues.

- **Real Semantics: Capturing Floating-point Imprecision** *with Hannah Blumberg, Daniel King, Paola Mariselli, and Prof. Eddie Kohler, Harvard University*

  We took on a unusual path about program correctness to dynamically capture erroneous floating-point operations that lead to sub-optimal and imprecise, but non-fatal results. A custom LLVM [3] Intermediate Representation interpreter (lli) was built as a result of this project and it helped us discover many interesting behaviors, if not bugs, in many real-world numeric programs like matrix multiplication, numerical integration, and ray-tracing algorithms used in game engines.

- **Persistent Masstree: Designing File Systems for Byte-addressable, Persistent Memory** *with Prof. Margo Seltzer, Harvard University*

  Emerging technologies with drastically different performance characteristics make us rethink about early-day assumptions on which most of today's software systems are built. We took Masstree [6], a highly concurrent and cache-friendly in-memory data structure, and added support for crash consistency if it were to operate in a persistent environment to explore the engineering and performance overheads of such modifications. We hope the findings provide some insights of designing a future generation of file systems.

- **Kernel Support for Multi-threaded Program Deterministic Record and Replay** *with Prof. Jason Flinn, University of Michigan*

  Deterministic record and replay systems are powerful debugging platforms that greatly ease the engineering challenges associated with writing correct parallel programs. A key aspect of these systems is to properly record scheduling information so that interleaved threads can be replayed in the observed order. My contribution was to add support for x86 atomic (lock-prefixed) assembly instructions to the underlying record and replay system.

Other Projects (Non-systems Related)

- **Ethereum Blockchain Compression**

Proposed a new off-line storage format for the Ethereum blockchain. We can reduce the storage space occupied by the blockchain (by as many as 58%) by discovering and leveraging patterns in the ways people use Ethereum both as a cryptocurrency and a programming platform.

- **Study of Generalized Deferred Acceptance (Game Theory)**

  A simulation study of a generalized deferred acceptance mechanism where there are costs associated with proposals/rejections.

- **Formal Proof with the Coq [1] Proof Assistant**

  Modeling and reasoning of a memory-word-based toy language, in an attempt to formally prove the Scalable Commutativity Rule.

Dissertations and Theses

- **On the Design and Implementation of High-performance Transaction Processing in Main-memory Databases**
  PhD Dissertation, Harvard University, May 2020
  Available at `https://huangyihe.github.io/pdf/HUANG-DISSERTATION-2020.pdf`

- **Software Systems for Advanced Memory Technologies**
  PhD Candidate Qualification Exam/Master's Thesis, Harvard University, May 2016
  Available at `https://huangyihe.github.io/pdf/Yihe_QualsDocs.pdf`

- **Gas Turbine IGV Actuation Control System Integration**
  Bachelor's Thesis, Shanghai Jiao Tong University, August 2014
  Mechanical modeling, control system design, and PLC programming
  Available at `https://huangyihe.github.io/pdf/sjtu-capstone-report.pdf`

- **Multi-core Cache-coherent Microprocessor Design**
  Bachelor's Thesis, University of Michigan, April 2013
  Computer architecture design in synthesizable Verilog HDL

Miscellaneous

- **Harvard University Teaching Fellow** CS 61, Fall 2015, Fall 2018 (Head TF)
- **PPoPP'17 Artifact Evaluation Committee** Fall 2016
- **Harvard University Research Assistant** 2015–2020
- **UM College of Engineering Dean's Honor List** Fall 2012–Winter 2014
- **Served on the UM-SJTU Joint Institute Honor Council** Fall 2011–Summer 2012
- **Ministry of Education (China) National Scholarship for College Students** Academic Year 2011

References

[1] The Coq proof assistant. `https://coq.inria.fr/`.

[2] Eddie Kohler. `https://read.seas.harvard.edu/~kohler/`.

[3] The LLVM compiler infrastructure project. `https://llvm.org/`.

[4] P. A. Bernstein and N. Goodman. Multiversion concurrency control – theory and algorithms. *ACM Transactions on Database Systems (TODS)*, 8(4):465–483, 1983.

[5] A. Dragojević, D. Narayanan, E. B. Nightingale, M. Renzelmann, A. Shamis, A. Badam, and M. Castro. No compromises: distributed transactions with consistency, availability, and performance. In *Proceedings of the 25th symposium on operating systems principles*, pages 54–70. ACM, 2015.

[6] Y. Mao, E. Kohler, and R. T. Morris. Cache craftiness for fast multicore key-value storage. In *Proceedings of the 7th ACM european conference on Computer Systems*, pages 183–196. ACM, 2012.