

Zeyuan Hu

✉ zeyuan.zack.hu@gmail.com 🏠 <https://zhu45.org> 🌐 <https://github.com/xxks-kkk>

EDUCATION

- | | | |
|---|--------------------|-----------------------------|
| University of Texas | Austin, TX | Sept 2017 – May 2019 |
| • M.S. in Computer Science. (GPA: 3.81/4.00) | | |
| University of Wisconsin | Madison, WI | Sept 2010 – Dec 2014 |
| • B.A. Triple Major, Computer Science, Economics with Honors, Mathematics. (GPA: 3.80/4.00) | | |

WORK EXPERIENCE

- | | | |
|--|---------------------------------------|-----------------------------------|
| Cloud Architect Engineer
Omnia Storage Team | State Street Financial Service | June 2019 – September 2019 |
| • Built auto-deployment system of IBM Cloud Object Storage in multi-site clusters using Ansible , Docker | | |
| • Developed a distributed workload generator and performance benchmark toolkit in Go | | |
| Software Engineer
DB2 LUW federation team | IBM | August 2015 – August 2017 |
| • Constructed Hive and Impala wrappers with C++ and Java to support federation database between traditional RDBMS and Hadoop-based data warehouse solution | | |
| • Created automated setup tools with Shell that reduce product configuration time by 75% | | |
| • Enhanced server option optimization tools using C to reduce federation database performance tuning time by 90 % and enable the capability of tuning the product against Hive, Impala, and Spark | | |
| • Resolved over 20 defects, including a severe memory leak issue that impacted a \$1.6 million deal. <i>Awarded IBM Manager's Choice Award 2016</i> | | |
| Software Engineer Internship
HPC infrastructure team | Schlumberger | May 2018 – August 2018 |
| • Implemented a monitoring component of the in-house High-Performance Computing (HPC) engine in C++ to provide the fault tolerance and handle the “straggler” problem | | |
| • Employed SGD algorithm to dynamically learn the best timing for backup executions of the in-progress tasks based on the computation task characteristics | | |
| • Built a C++ code generator that automatically generates the application layer code based on the engine API | | |

SELECTED PROJECTS

- **RustFS** (2018 - 2019). Building a user-space file system that leverages NVMe SSD. [Rust](#), [SPDK](#)
- **Strata with Lease** (2018). Extended Strata file system with Lease mechanism to support concurrent file access across processes. [C](#).
- **HyperPebblesDB** (2018). Constructed a key-value store that is part of LevelDB family with focus on reducing write amplification. [C++](#), [CMake](#), [Autotools](#)
- **Distributed Key-Value Store** (2018). Built a distributed key-value store with [Python](#) that uses *eventually consistency* model with two session guarantees: *Read Your Writes* and *Monotonic Reads*.

LANGUAGES AND TECHNOLOGIES

- **Languages:** C++, C, Python, Go, Rust, Shell, SQL, Java, Lisp, MATLAB
- **Software:** CMake, Autotools, Git, Docker, Ansible, QEMU, Tensorflow, Keras, ClearCase, Hive, Impala, Maven, Hadoop