Mengxing Liu

(+86) 18201229929 · liu-mx15@mails.tsinghua.edu.cn · https://liumx10.github.io

Education

Tsinghua University, Computer Science and Technology, *Ph.D.* 2015.9 - Present

- Research interests: non-volatile memory, concurrency control, transactional memory, linearizability theory.
- Honors: National Scholarship, Sohu Scholarship, Guanghua Scholarship, People.com.cn Scholarship.

Tsinghau University, Computer Science and Technology, *Bachelor* 2010.9 - 2015.7

- GPA: 90/100
- Honors: First Prize Scholarship for Freshmen, National Endeavor Fellowship, Peer-to-peer Lending Fellowship, Excellent Theses Award, Excellent Graduate in Department of Computer Science and Technology.

Internship

Hashfuture, Backend team leader 2018.1-Present

- Developed backend for several applications as the team leader with 7 colleagues. Total registered users are more than 1.5 million.
- Developed the backend of the first beta version on my own.
- With the expansion of the team and the bussiness, mainly focused on communication with other teams, the code architecture and database schema, the management of developers, and investigation on new tool chains. New tool chains includes: online log collection, scheduled tasks, container, and shared accounts among applications.
- Designed a solution to transfer money cross applications by using the *Try-Confirm-Cancel* technique, ensuring the data consistency in spite of network errors.

Google Summer of Code, Intern 2017.5-2017.8

- Investigated the possibility of improving the performance of detecting conflicts in PostgreSQL (PG).
- Digged into the design of serializable snapshot isolation in PG and the theory behind it by reading the original paper and related source code.
- Replaced the linked list with skip list and hash table.
- Reported that the proposal did not work after lots of experiments, because an SQL execution takes a little time on tracking read-write conflicts, and other data structures could incur new performance penalty.

Microsoft Research Asia, Research Intern 2016.9-2017.3

- Investigated how to use lock to implement transactions on non-volatile memory (NVM) . We maintained the recoverability of transactions by tracing the dependency relationships among them, and introduced a faster algorithm for detecting in the case of using two-phase lock (2PL).
- Built an in-memory database for evaluation based on RAIN, which is an distributed NVM-oriented storage library similar to RAID.

Tencent, Intern 2013.7-2013.8

- Implemented an enterprise office assistance based on Wechat public platform, which supported inner contact, calendar and signing in.
- Chose the framework for our team and did the main development tasks.

Research Projects

DudeTM: Durable transactional memory on NVM

- DudeTM is a transactional memory library on NVM, providing an easy-to-use interface for programmers to build ACID transactions. The key idea is decoupling a transaction into three totally decoupled components to overcome performance overhead of traditional redo logging and undo logging mechanisms. It obtains 2x to 4x speedup than the state-of-art technique. This work is published in ASPLOS'17.

DudeTX: Durable transactions interactive with locks

- DudeTM is based on DudeTM, but does not provide isolation for developers to make it more flexible. Programmers have to use locks for concurrency control. To gurantee the recoverability, we provided a lock library

to trace the order among lock acquiring/releasing. Transaction commits are based on the lock order. This work is published on Transaction on Storage.

RNTree: High performance persistent B+tree

- B+tree is one of the most important data structures in database and key/value stores. We use HTM carefully to solve two problem of B+tree on NVM: trade-off between sorted leaf node and write amplification problem; low scalability in skewed workload. Its throughput is 4x higher than the state-of-art works. This work is submitted to HPDC'19.

NIL: A CAP-like theorem for concurrent data structures on NVM

- We found and proved the NIL theorem: for concurrent data structures on NVM, it is impossible to get non-blocking, independency and linearizability simultaneously. The paper is under manufacture.

DFS-Rsync: Remote synchronization for DFS

- We offered a new interface for distributed file system: remote synchronization (RSync), to synchronize files between file systems. Matched blocks can be read from the local system so that the network transmission is omitted. We implemented it on HDFS and RSync is about 10x faster than copying directly. We applied a patent on this work.

SSEvent: Cooperative task management without stack ripping in distributed transactions

- There is a trade-off between multi-thread programming and event driven programming when developing distribtued transactions. Event driven programming avoids the overhead of context switch, but incurs stack ripping. We used corountine library to wrap events, making event driven programming similar to multi-thread programming. This work is my bachelor thesis and got the excellent theses award (6/123).

PUBLICATIONS

1. DudeTM: Building Durable Transactions with Decoupling for Persistent Memory
   Mengxing Liu, Mingxing Zhang, Kang Chen, Xuehai Qian, Yongwei Wu, Weimin Zheng, and Jinglei Ren.
   *Proceedings of the 22nd ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'17). (**CCF A**)*
2. DudeTX: Durable Decoupled Transaction
   Mengxing Liu, Mingxing Zhang, Kang Chen, Xuehai Qian, Yongwei Wu, Weimin Zheng, and Jinglei Ren.
   *ACM Transactions on Storage(TOS) 2018 (**CCF A**)*
3. Large Scale Communication in Cloud Needs Hybrid RDMA Schema
   Teng Ma, Mingxing Zhang, Zhuo Song, **Mengxing Liu**, Kang Chen, and Yongwei Wu
   Presented in the Poster Section of OSDI'18.
4. Building Scalable NVM-based B+tree with HTM
   Mengxing Liu, Jiankai Xin, Kang Chen, Yongwei Wu
   Submitted to HPDC'19

SKILLS

- Experienced about c++ and python. Java/Go/Shell/JS/CSS are used in some projects.
- Experienced about transactional theory, understand the transaction implementations of MySQL and PG, and have read the related source codes of PG.
- Experienced about parallel programming and linearizability theory.
- Fundamental knowledge of the CPU architecture. I have implemented a CPU with MIPS32 ISA, which can support our teaching operating system.
- Fundamental knowledge of the distributed system.