Jinyang Li

DOB: 02/28/1998 Email: lijinyang@pku.edu.cn

EDUCATION BACKGROUND

Peking University Beijing, China
Major: Computer Science 09/2016-06/2020
Degree: Bachelor of Science **GPA:** 3.619/4.000
Related Courses:
Mathematics: Linear Algebra, Advanced Mathematics, Set Theory and Graph Theory, Probability Theory and Statistics(A), Algebraic Architecture and Combinatorial Mathematics, Game Theory
Computer Science: Practice of Programming in C&C++, Data Structure and Algorithm(A), Introduction to Computer Systems, Algorithm Design and Analysis, Computer Architectures, Operating Systems(A), Introduction to Database Systems, Big Data Management Technology

RESEARCH EXPERIENCES

Project name: Windows Sketches 01/2019 -- 05/2019

- The project aims to estimate the frequency, persistency (is the number of time windows where an item appears) and recency (the last time window where an item appears) of all items at the same time in a data stream. We use counters to record persistency, and also log the last time window an item appears so as to avoid duplicates, which can also be used as recency.
- We work as a team of 3, and I am a team member. The final paper is in preparation. I am the second author.

Project name: Elastic Cuckoo filter 01/2019 -- 06/2019

- The project aims to overcome two shortcomings of the Cuckoo filter: 1) its size can only be a power of 2; 2) the size of a Cuckoo filter cannot be dynamically tuned. The reason of the first shortcoming is the usage of XOR operations, and we find different items can be restricted in different ranges of 2^n, so we use a hash function to do this. As for the second shortcoming, we extend a Cuckoo filter by copying and performing lazy updates: delete redundant copies in a bucket only when it is accessed in an insertion. And we shrink a Cuckoo filter by moving each fingerprint from bucket i to $i/2$.
- We work in a team of 4, and I am the team leader, responsible for part of implementation, part of experiments and most of paper writing. The paper has been submitted to TPDS (first author).

Project name: Data Spread 07/2019 -- now

- The project aims to combine spreadsheets with databases to benefit from both. I am responsible to bridge the gap that spreadsheets support users to manipulate orders of rows, while databases do not have a meaning of order.
- I design a data structure to maintain the mapping from positions of rows (to represent a row in spreadsheets) to rowIDs (to query a row in database), which supports queries, updates(insertions, deletions) and order manipulations (reorder, swap, move). I need to deal with challenge 1 that an update causes cascading updates of positions of other rows, and challenge 2 that order manipulations need collective updates. So I design a self-balancing tree called Stree. Furthermore, there is challenge 3 that the data structure should adapt to unbalanced reads and writes in varying workloads, so I design Dynamic Array, whose idea is to cut arrays when necessary and use Stree to rank arrays. It achieves far better performance when the number of writes in a workload is small. The ordered access in my

research has not been solved before, and experimental results show that these two data structures have shorter average response time and more stable performance than traditional methods (standard arrays, linked list and Counted-Btree).

✧ I am responsible for all tasks about ordered access in this project by myself, including implementation, experiments and paper writing. Now experiments have been finished and I am preparing the paper draft. I plan to submit to VLDB next month.

PUBLICATIONS

- Fast and accurate stream processing by filtering the cold. Tong Yang, Jie Jiang, Yang Zhou, Long He, **Jinyang Li**, Bin Cui, Steve Uhlig, Xiaoming Li. **VLDBJ 2019.**
- HeavyKeeper: An Accurate Algorithm for Finding Top- k Elephant Flows. Tong Yang, Haowei Zhang, **Jinyang Li**, Junzhi Gong, Steve Uhlig, Xiaoming Li. **IEEE/ACM Transactions on Networking 2019.**

EXTRACURRICULAR AND SKILL

Programming language: C, C++, Java, Python, Go, Verilog, SQL

Computer skill: basic data structure and algorithm, UI design