Xi Liang

xiliang.me | xiliang678@gmail.com | Bay Area, CA

Education

University of Chicago 2016.09 – 2022.03
Ph.D. in Computer Science Advisor: Prof. Sanjay Krishnan

Design and build efficient data systems with a specialty in approximate query processing.
Ph.D. Thesis: Synopses for Efficient and Reliable Approximate Query Processing

Selected Papers

JanusAQP: Efficient Partition Tree Maintenance for Dynamic Approximate Query Processing
Xi Liang, Stavros Sintos, Sanjay Krishnan. **ICDE 2023**.

Combining Aggregation and Sampling (Nearly) Optimally for Approximate Query Processing
Xi Liang, Stavros Sintos, Zechao Shang, Sanjay Krishnan. **SIGMOD 2021**.

Fast and Reliable Missing Data Contingency Analysis with Predicate-Constraints
Xi Liang, Zechao Shang, Aaron J. Elmore, Sanjay Krishnan, Michael Franklin. **SIGMOD 2020**.

Opportunistic View Materialization with Deep Reinforcement Learning
Xi Liang, Aaron J. Elmore and Sanjay Krishnan. arXiv:1903.01363, 2019.

Experiences

Software Engineer | Databricks | Bay Area, CA 2022.03 – Present

- Query optimization, materialized view.

Research Assistant | University of Chicago | Chicago, IL 2016.09 – 2022.03

- I designed and implemented the following research prototypes using Java and Python from scratch.
- **JanusAQP**: Built on Apache Kafka and based on a novel dynamic partition tree that extends our previous work PASS, JanusAQP is a dynamic AQP system that can process insertions and deletions of data efficiently while provide query result with better accuclow latency.
- **PASS**: A system that combines offline-aggregates with stratified sampling for Approximate Query Processing. PASS features a partitioning algorithm that can minimize the worst-case error and a unique design that enables sample skipping to achieve a higher accuracy without sacrificing the latency.
- **PC**: A missing data analysis framework based on our novel synopsis called predicate-constraints. PC not only make it possible to encode beliefs and assumptions of missing data in a logical and formal way but also can be used to derive tight hard bounds for aggregate queries to achieve optimal reliability.
- **DQM**: A system that manages materialized views with deep reinforcement learning. Integrated with Apache SparkSQL, DQM demonstrates that robust view management to reduce the latency over evolving workloads can be achieved using reinforcement learning.
- **CCJS**: A cross-cluster job system that I built in the first year of my Ph.D. As the infrastructure of my research project, CCJS is designed to scale across different clusters that might be using different job schedulers (Slurm, Cobalt, etc). With CCJS, I was able to manage and run millions of jobs using free and heterogeneous computing resources to collect data for my research.

Software Engineer | Aurender Inc. | South Korea 2009.11 – 2013.08

- As one of the three software engineers the company started with, I worked on the award-winning music server "Aurender" until the company was acquired.
- As the only software engineer in charge of the server-side. I designed and implemented core features of the system from scratch. Including music library management; customized hardware support and control; remote-upgrading for shipped products; GUIs of the music player; RESTful APIs to interact with the iOS app, etc. (C++/Python/QT/SQLite/LinuxDeviceDriver)

Awards

2017, 2020: University Unrestricted Fellowship, University of Chicago.
2014: Outstanding Student Award, National Chiao Tung University.