Boao (James) Chen

boaochen@brandeis.edu | 607-882-4258

Research Interests

Databases, Data Systems, Machine Learning

Education

Brandeis University, Waltham, MA **Sep 2024 – Present**
Doctor of Philosophy, Computer Science

Cornell University, Ithaca, NY **May 2024**
Bachelor of Arts, Major in Computer Science

Publications

- Ke Li, Ruidong Zhang, Boao Chen, Siyuan Chen, Sicheng Yin, Saif Mahmud, Francois Guimbretiere, Cheng Zhang, *GazeTrak: Exploring Acoustic-based Eye Tracking on a Glass Frame*. MobiCom 2024.
- Ke Li, Ruidong Zhang, Siyuan Chen, Boao Chen, Mose Sakashita, Francois Guimbretiere, Cheng Zhang, *EyeEcho: Continuous Facial Expression Tracking on Glasses with Minimally-obtrusive Acoustic Sensing*. CHI 2024.
- Ke Li, Devansh Agarwal, Ruidong Zhang, Vipin Gunda, Tianjun Mo, Saif Mahmud, Boao Chen, Francois Guimbretiere, Cheng Zhang, *SonicID: User Identification on Smart Glasses with Acoustic Sensing*. IMWUT 2024.

Poster

- Alexander Ott, Boao (James) Chen, Arpita Saha, Subhadeep Sarkar, *Self-Designing LSM Memory Buffer*, 2025 North East Database Day.

Projects

- **Self-Designing LSM Memory Buffer (Ongoing):**
  Designing the optimal memory buffer implementation is critical to achieving optimal performance in any log structured merge (LSM) tree. State-of-the-art LSM engines suboptimally adopt fixed buffer configurations for unknown and dynamic workloads. This traditional design choice often leads to suboptimal performance, especially when the workload shifts over time. We propose building a self-designing LSM memory buffer that can transition between different memory buffer implementations on the fly based on changing workload characteristics. Its machine-learning-based hyperparameter optimization framework learns from historical experimental data and iteratively suggests the optimal configuration of memory buffers when workload changes are detected. Finally, the current memory buffer implementation gradually transitions toward the suggested optimal memory buffer design lazily while maintaining robust performance.

- **LSM Buffer Selection (Ongoing):**
  LSM-trees are known for their ingestion-optimized feature while maintaining competitive read performance. Specifically, one of the key design decisions of LSM-trees is to keep an in-memory buffer to store and append incoming entries. The buffer is sealed, sorted, and flushed to secondary storage when full. We observe that the LSM system's ingestion speed is bounded by this memory buffer design, since the performance of any LSM system is highly dependent on the buffer's configuration, given that all incoming entries must first batch write through the buffer before flushing to secondary storage. In this project, we benchmark multiple buffer implementations under diverse workloads that vary in operation type, frequency, interleaving, and distribution. We examine each design's trade-offs in write latency, point lookup latency, range lookup latency, and memory footprint across various hardware (SSD vs. HDD) and write-ahead logging (WAL) settings. Our goal is to construct a comprehensive handbook that provides guidance in selecting the appropriate in-memory buffer configuration for specific performance targets and workload patterns.

Teaching Experience

- **Spring 2025 –** Teaching Assistant, Database Management Systems (COSI 127B), Brandeis University.
- **Fall 2024 –** Teaching Assistant, Algorithms (COSI 180A), Brandeis University.
- **Spring 2024 –** Teaching Assistant, Applied High-Performance and Parallel Computing (CS 5520), Cornell University.
- **Fall 2023 –** Teaching Assistant, Foundations of Artificial Intelligence (CS 4700), Cornell University.
- **Spring 2023 –** Teaching Assistant, Operating Systems (CS 4410), Cornell University.

Technical Skills

- **Programming Languages:** Python, C/C++, Java, SQL, R, MATLAB
- **Databases:** RocksDB
- **Tools:** PyTorch, TensorFlow, OpenCV, LaTeX, Git