Zhihao Ma

(+1) 213-705-4123 | zhihaoma@usc.edu | | |

Education

University of Southern California — Los Angeles, US
Master of Science in Computer Science, GPA: 3.92/4.0 — *Aug. 2022 – May. 2024 (Expected)*

Sun Yat-Sen University — Guangzhou, China
Master of Engineering in Computer Technology, GPA: 3.9/4.0 — *Sep. 2019 – July. 2022*

Sun Yat-Sen University — Guangzhou, China
Bachelor of Engineering in Traffic Engineering, GPA: 3.9/4.0 — *Sep. 2015 – July. 2019*

Skills

- **Languages**: C/C++, Java, Python, Scala, JavaScript, HTML, CSS, SparkSQL
- **Frameworks**: AWS, Spark, Kubernetes, Docker, Springboot, Spring, React

Career Experience

Amazon — Irvine, US
Software Development Engineer Intern | Java, Scala, SparkSQL — *May.2023 – Aug.2023*

- Worked on **re-ranking** contents to improve the personalized recommendation performance on Amazon Deals website
- Performed **data analysis**, designed deals related features and built datasets of customers' preference on deals
- Implemented a **Stream Model** that consumes contents from different strategies and trained it with deals features
- Created a workflow using **AWS** Distributed Job Scheduler (DJS) to automatically train models on a daily basis
- Integrated the model to the production pipeline. The offline performance was improved 8% in nDCG

Baidu — Shenzhen, China
Software Development Engineer Intern | Python, Flask, Kubernetes, MySQL — *April.2021 – Aug.2021*

- Designed a baseline for a multi-agent competition. Implemented a hierarchical framework where strategies are chosen by algorithm (PPO+GAE) and then executed by pre-defined rules. Achieved **99%** win rate against random agents
- Implemented a **scoring system** that supported downloading solutions, pairing participants and updating their rank
- Developed a **multi-process** program for running simulations simultaneously using inner XPARL framework
- Utilized **Kubernetes** to manage distributed application across multiple machines that could serve over 1000 teams

Huawei — Shenzhen, China
Research Intern | Python — *May.2020 – Oct.2020*

- Introduced First-Order Logic and Multi-Hop Reasoning into Hierarchical Reinforcement Learning. The proposed model kept competitive performance against baselines and decreased 4.5 times inference time and 64% memory

Projects

Trojan: A High Performance Key-Value Store (Go)

- Built on the **bitcask** model and featured by **low latency** and **high throughput** for its **write-once** and **append-only** nature. Keys and indexes are stored in the memory while values are in the disk.
- Implemented various memory index structures (e.g. **BTree, ART, B+ Tree**) and transaction manager
- Applied lock mechanism to support **concurrent storage and query** and it's easy and fast to backup and recover.
- Supported Redis Serialization Protocol and 5 Redis data structures (e.g. String, Hash, Set, List, SortedSet)

BusTub Relational DBMS (C++)

- Implemented a **Buffer Pool Manager** with **LRU-K** Eviction policy and hashmap to fetch in and flesh out pages
- Built a robust **concurrent B+ Tree** based indexer and iterator for faster retrieval and tranversal of data
- Supported **query execution and optimization** by constructing core executors including sequential scan, insert, delete, projection, filter, nested loop join, index join, hash join, aggregation, sort, limit and top-N
- Constructed a lock manager to grant/block resource lock for concurrent transaction control

Tiny WebServer (C++)

- Adopt **non-blocking** socket and **edge-trigger** event handling mode to support high concurrency workload
- Apply **thread pool management** to asynchronously execute requests and thus avoids head-of-line blocking
- Encapsulate and support connections with MySQL database for persistent storage operations
- Support procedure call with **CGI** and **HTTP** Protocol and **asynchronous** consumer-producer **logging**

MAXP Innovation Competition (Graph Learning Task based on DGL)

- Predicted the category of paper(nodes) given a **large-scale** paper diagram(over 1.5m nodes and 20m edges)
- Added edges to top 20 similar nodes for each isolated node, augmented node features with node2vec **embeddings** after performing **PCA** on original features and built a heterogeneous graph with cite, cited, self-loop, similar relationships
- Implemented a multi-head self attention **GAT** and masked the label during mini-batch **k-fold** training
- Blended models trained on original and **heterogeneous graph** and the whole solution ranked 3rd/1477.