Mingyong Ma

3869 Miramar St, San Diego, US

☎ 858-539-6919 ✉ m7ma@ucsd.edu My Website github.com/mamioma

Education

University of California, San Diego **Sep. 2022 – Dec 2023**
Master of Science in Computer Science *San Diego, California*

Wuhan University **Sep. 2018 – June 2022**
*Undergraduate Remote Sensing Science and Technology (**GIS** major)* *Wuhan, China*

Experience

Adobe **June 2023 – Sep 2022**

Software Engineer Intern *San Jose, California*

- Building Adobe primary AI platform infrastructure **Firefall**, improve API call from **sync** to **async**. Using **Jmeter** for **load-testing**, able to generate **1600 TPS** (token per second).
- Implementing using **REST API** that able to **CRUD** a task, and save it in **postgres DB** with **almebic** version control.
- Generating a **docker** container for each fine-tuning task, scheduling according to the load on **Azure blob Storage**
- **Reduced the network I/O** from **13GB** to **32MB** per **inference** call. By utilizing **PEFT**, the based model is consistent for every fine-tuning job, thus is stored in **in-memory-buffer** of the **Docker** container, with only the **Lora layer** being stored in **Azure Blob Storage**. Therefore, only the Lora layer (32MB) instead of the entire model (13GB) is downloaded into Docker container.
- Facilitate the fine-tuning of **FlanT5** and **llama2** models, with the capability to independently store the base model and fine-tuning layer utilizing **PEFT**.

Amazon **June 2022 – August 2022**

Software engineer Intern *Shenzhen*

- Developed an **image processing** algorithm that combines **deep learning** techniques with the **Unsharp** algorithm, achieving 20% superior results compared to the camera algorithm used in tablets.
- Utilized **Canny Operator** for **edge enhancement** and **Unsharp** for mid-frequency enhancement. And introduced **ESR-GAN** to restore general real-world images by synthesising pairs with a more practical **degradation** process.
- Achieved automatic **object detection** on portraits utilizing **YOLOv5** and Implemented more refined **Super-Resolution** for every portraits.
- Conducted an evaluation of our proposed algorithm using **Imatest** software in the Amazon lab, observing an increase in **MTF-50**, which showcases an improvement in image sharpness.

Lenovo **Nov 2021 – Feb 2022**

Data Analytic Intern *Beijing*

- Developed Spark SQL Catalyst Expressions (i.e., SQL functions) using Java/Scala to optimize the performance of DataFrame Transformation. Employed **Spark DataFrame** and **MapReduce** for data extraction from IDC and GFK.
- Conducted **time series forecasting** to predict future sales of Lenovo's notebook products and tablets, utilizing Lenovo's historical sales data as well as data from other companies such as IDC and GFK.
- Increased the forecasting accuracy of the model by 4.2% by implementing machine learning algorithms such as **Prophet** and deep learning models like **LSTM** or **GRU**.

Projects

Database acceleration | *C++14, mutex* **March 2023**

- Realized a Database index method utilizing **B+ Tree**, which shows **10 times faster** performance compared with **Hash index** or **file scan** on **range search**.
- Built a **Buffer Pool** on top of I/O layer, and realize **Buffer Replacement Policy** and **LRU clock algorithm**.
- Built a **B+ Tree** on top of Buffer Pool, supporting **CRUD** operation. Besides, it can save more than **50GB** data.

Distributed Cloud File System | *Go, gRPC* **January 2023**

- Created a **fault tolerant cloud file storage** service called SurfStore (client and server communicating using **gRPC**).
- The SurfStore service is composed of the following two services: **BlockStore**: Stores these **blocks**, and when given an identifier, retrieves and returns the appropriate block. **MetaStore**: Manages the metadata of files and mapping of filenames to blocks (hash marshalled by **SHA-256**).
- The clients' file data is stored in local database with **version**. When invoking into client, the **sync** operation will occur, and new files added to base directory will be uploaded to the cloud, files that were sync'd to the cloud from other clients will be downloaded to base directory, and any files which have ”edit conflicts” will be resolved.
- Store and manage the block in different BlockStore using **Consistent Hashing Ring**.
- Ensure the MetaStore is fault tolerant and stays consistent regardless of minority of server failures by **RAFT protocol**.