Zichen Lu

☎ 13061277115 · ✉ mikaovo2000@gmail.com · https://github.com/MikaOvO

Education

Beijing Institute of Technology *Master* in Software Engineering Sep.2022 - Jun.2024

Shandong University *Bachelor* in Software Engineering Sep.2018 - Jun.2022

rank top5% CCF Excellent College Student, National Scholarship

Competitions

International Collegiate Programming Contest (ACM-ICPC) regional **Gold Medal**	Nov.2021
Chinese Collegiate Programming Contest (CCPC) regional **Gold Medal**	Nov.2021
CCF Computer System and Program Design Contest (CCSP) **Gold Medal**	Oct.2020
Aliyun Tianchi CIKM 2022 AnalytiCup Contest **Champion**	Aug.2022 - Oct.2022
2021 Huawei GaussDB Database Contest **2nd place**	Sep.2021 - Nov.2021
2021 Big Data Computing Intelligence Contest **2nd place**	Sep.2021 - Jan.2022
China Minsheng Bank Financial Technology Contest **3rd place**	Aug.2022 - Sep.2022

Intern Experience

Microsoft (China) Company | M365 Software Engineer Intern Jun.2022 - Aug.2022

- Migrate the revoke-check related logic during Yubikey registration from the file system to AzureTable , **greatly improving performance and reducing storage pressure**.
- Referring to the Bot communication logic of the OneDrive team, implement the function of supporting concurrent communication to users who are not registered with TorusBot, which **improves the usage rate of TorusBot**; Refactor some code to make it easier to add new Card styles.

ByteDance Technology Company | Toutiao Recommendation Algorithm Intern Dec.2021 - Apr.2022

- Implement the function that users who follow each other must reach each other when they post articles, and **improve the click-through rate of recommended content**.
- Actively find out that the AUC of the search recommendation related model is low, then re-create the data table (find the relevant data stream and write it into the Clickhouse through Kafka), add the search recall pool, and implement the DeepFM model during rough and final sorting, **increase the DAU in the AB experiment**, which proves the effectiveness.

Project Experience

RPC | C++ Oct.2022 - Dec.2022

- **Background**: The goal is to implement a high-performance remote procedure call under Linux for distributed and microservice communication.
- **Network Programming**: Use **boost::asio** to implement async server; Use **EPOLL, Threadpool, socket** to implement Reactor sync server; Allow users to choose according to their needs。
- **Protocol of Communication**: Use **msgpack protocol** to perform parameter packaging and parsing.
- **Procedure Call**: Use **traits and template trick** in C++ to finish remote procedure call.
- **Client**: Supports blocking until timeout or result/exception return, or asynchronously obtains the corresponding std::future and manages it.
- **Result**: Compared with rest_rpc, **improve the performance of specific scenarios and reduce the difficulty of understanding the code**.

MIT 6.824 Distributed Systems | Go Apr.2022 - Jul.2022

- **Background**: Build a high-performance, fault-tolerant, and highly available distributed server cluster.
- Implement Raft Protocol.
- Implement a KV storage server cluster, which is Load-Balance, can pull configuration, and respond to customer group requests.
- **Result**: Pass all the 6.824 test cases.