Crazylychee

Phone: 15118878106 | Email: xu2757197595@outlook.com
Address: Guangzhou China
WebSite: crazylychee.github.io
Objective: Back-end

EDUCATION

Guangdong University of Technology Sep 2022 - Jun 2026

Computer Science and Technology Guangzhou

SUMMARY

Understand the principles of network communication and be familiar with common network protocols such as TCP/UDP, HTTP/HTTPS, etc.
Proficient in Java syntax, collections, multithreading, and concurrent programming fundamentals.
Understand JUC concurrent programming, including the use and principles of locks, AQS, volatile, etc.
Master the basic usage of Spring, SpringMVC, and SpringBoot, and understand the principles of IoC and AOP.
Familiar with the use of MySQL, including the principles of indexes, MVCC, transactions, etc.
Familiar with the use and basic principles of Redis, such as data structures and memory eviction mechanisms.
Familiar with common middleware such as RocketMQ and Nacos, and able to select appropriate middleware based on different scenarios.

PROFESSIONAL EXPERIENCE

Digital Guangdong Network Construction Co., Ltd

Backend Development Intern

Main Responsibilities: During the internship, participated in the development of the "One Network Unified Management" platform for Jieyang City and Yunfu City, focusing on efficient querying of massive data, reliable message delivery, and storage optimization. Enhanced system performance and stability through techniques such as deep pagination optimization, reliable message processing, cold-hot separation storage, and scheduled batch processing of statistical data.

1.Designed a dual-pointer pagination algorithm combined with partitioned bitmap indexing to optimize deep pagination queries for over 50 million event records, reducing the response time for jumping between 10 million record pages from 18.7 seconds to 22 milliseconds.
2.Used RocketMQ to receive mountain body landslide hardware bottom-layer transmission detection events, ensuring reliable message delivery through synchronous disk flushing and cluster synchronization replication mechanisms. Implemented message deduplication strategies using Redis cache tokens to avoid duplicate processing.

3.Based on cold-hot separation strategies, archived hardware detection data to historical tables using ShardingSphere monthly partitioned tables, avoiding the impact of overly large tables on query performance.

4.Utilized XXL-Job for scheduled batch generation of hardware detection case statistics, reducing the real-time query pressure on MySQL and improving system performance. Additionally, gained in-depth understanding of the underlying principles of the XXL-Job framework during its application.