Xuanyi Li

◇ 614-397-8198 ◇ shanelxy@outlook.com ◇ linkedin.com/in/wirybeaver ◇ github.com/wirybeaver ◇ shanelxy.top

EDUCATION

The Ohio State University **Columbus, United States**
Master of Science in Computer Science; GPA (3.97/4.0) 08/2017 – 05/2019

Zhejiang University **Hangzhou, China**
Bachelor of Engineering in Automation; GPA (3.97/4.0) 09/2013 – 07/2017

TECHNICAL SKILLS

Programming Languages: Java, C++, C, Bash, Go, SQL, Assembly, Python
Frameworks: Spring Boot, gRPC, Apache Druid, PostgreSQL, MyBatis, MapReduce, Fluentd, QFS, Kafka
Operations: Terraform, Jenkins, DataDog, Linux, Gradle, JUnit, Mockito, Git, Docker, Kubernetes
AWS: EC2, RDS, Lambda, S3, Route53, Route53 Health Check, IAM, EMR

WORK EXPERIENCE

Software Engineer **Quantcast, Seattle**
Audience Platform 07/15/2019 – Present

- Code, Operate and Optimize distributed Apache Druid cluster via Terraform, which is hosted on AWS and comprises EC2, EMR, RDS, Route53, Route53 Health Check, S3.
- Ingest massive batch (terabyte) and Kafka streaming data into Druid through Quantcast MapReduce and customized Druid indexing extension respectively.
- Implement robust gRPC APIs and build Sprint Boot applications that deliver impactful insights to advertisers and publishers.
- Leverage Gradle, Jenkins, AWS Lambda, Docker, Quantcast Kubernetes and DataDog to foster an agile CI/CD environment and ensure the compliance of service health metircs for each owned projects.
- Participate in a team-wide on-call rotation. Fixed sev2 bugs caused by jute.maxbuffer and mmap.
- Welcome to get more achievement details in the link shanelxy.top/assets/ResumeXuanyi.pdf

Selected System Programming Projects

Raft based Fault-Tolerant Key/Value Storage in Go

- Implemented the distributed consensus protocol Raft with Go channel, including leader election, heartbeats, log replication and persistence determination.
- Optimized log backtracking by add a conflictIndex in RPC reply to bring stale follower up to date quickly.
- Created a key/value service on top of Raft to cope with concurrent and duplicated client requests.
- Implemented snapshotting to avoid log grows without bound.

Disk Oriented Storage Manager for the SQLite DBMS in C++

- Developed thread-safe buffer pool manager, encompassing extendible hash table and LRU policy to move physical pages back and forth from main memory to disk.
- Built B+Tree index to support insertion, deletion, point search and iterator.
- Implemented latch crabbing protocol to allow multiple threads access and modify the B+Tree index.

Multiplayer Tictactoe with Stream, Multicast and Failover

- Developed multi-threading servers in C
  - Monitored game resources and handle various types of requests from multiple clients (New Game, Reconnect, Move, End) in the TCP threads.
  - Optimized with I/O multiplexing function select() to increase the connection capacity.
  - Replied to multicasting requests for failover in the UDP thread.
- Built the client to multicast for a new server or connect with the backup server while noting that the original server crashes. Seamlessly resume the game from last move.

List Interpreter

- Implemented Lisp interpreter in Java to parse the expression into binary tree, evaluate arguments, bind them to associated formal parameters and recursively evaluate function body.