Xuanyi Li

◇ 614-397-8198 ◇ shanelxy@outlook.com ◇ linkedin.com/in/wirybeaver ◇ shanelxy.top ◇ H1B

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, Go, Shell, C++, C, Python, Assembly, SQL
Frameworks: Spring Boot, gRPC, Apache Druid, MySQL, MyBatis, Kafka, MapReduce, QFS
Operations: Terraform, DataDog, Jenkins, Git, Docker, Kubernetes, Gradle, JUnit, Mockito, Linux
AWS: EC2, Lambda, S3, IAM, Route53, RDS, EMR

WORK EXPERIENCE

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

Developed the backend and the big data system to drive Quantcast Audience Platform.

- Major contributor of Quantcast first real-time insights product to enable self-served advertising.
  - Extended Druid's row parser and exploited Druid's Kafka indexing service to ingest 20 thousand streaming records with P95 latency of 30s.
  - Set up DataDog dashboard and alerts to monitor data freshness, Kafka consuming lag, parsing error, unexpected ingestion behavior, etc.
  - Created gRPC API to provide production team with insights data of minute level granularity. Deployed Docker image of Spring Boot application to Quantcast customized Kubernetes.
  - Sped up queries by 6 times using fine tuned Java thread Pool.
  - Wrote dynamic integration test. Deployed the prober on AWS Lambda with Jenkins and Terraform.
- Maintain critical Druid AWS infrastructure, consisting of EC2, EMR, RDS, Route53, S3.
  - Migrated Druid cluster from one AWS provider to another and fixed issues caused by network configuration and hard coded scripts.
  - Tiered Druid cluster to support multi-tenancy which allows engineers to tune query performance in terms of tenants and Druid roles.
  - Segregated the Druid running environment and standardized cluster deployment process with standalone Jenkins pipelines.
- Participate in a team-wide on-call rotation. Communicate with external teams for SLA violations. Find out the root cause of incidents and update runbooks. Positively handoff at the weekly Ops meeting.

Back-end Engineer Intern **Alibaba, China**
Taobao Transactoin Platform 05/10/2018 – 07/27/2018

Built an internal distributed Spring Boot application to automate AI development.

- Simplified the assemble of various AI workflow processes (data pre-processing, offline training, online deployment, etc.) with enhanced Responsibility Chain Pattern.
- Utilized Factory Pattern and Builder Pattern to automate the generation of offline training UI template and the serilization and de-serilization of training parameters in terms of the AI algorithm type.
- Used HttpClient to poll training status and applied MySQL optimistic lock to guarantee consistency.
- Dispatched customized AI models to application servers with Diamond (persisted config management system) and Tair (distributed cache).
- Empowered the coupon team to reduce one week for tweaking their recommendation algorithm.

Deep Learning Research Intern **Singapore University of Technology and Design**
Natural Language Processing Lab 07/03/2016 – 08/31/2016

Ameliorated multi-task learning for text classification and published a paper in IEEE DSAA.

- Improved classical multi-task deep learning model with Keras to detect the sentiment polarity, reviewer identification, subjectivity/objectivity of each text simultaneously.

- Crawled 75,000 rotten tomato reviews with Python. Inserted CNN to catch exclusive features and appended gated control unit to filter noise, improving accuracy by 0.47% and 1.10% respectively.

PROJECTS

Raft based Fault-Tolerant Key/Value Storage

- Implemented the distributed consensus protocol Raft in Go, 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.

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 thread; Used select() to cope with multitasking.
  - 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.

Other Side Projects

- **Lisp 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.
- **Spring Clone:** Wrote an Spring clone from scratch in Java and implemented the feature of IoC (Inverse of Control) and AOP (Aspect Oriented Programming).
- **Seckill Shopping:** Used Redis to reject overselling requests. Combined RocketMQ with local message table to implement the distributed transaction per the choreography-based saga pattern.

PUBLICATION

Xuanyi Li, Weimin Wu, Hongye Su. Convolutional Neural Networks Based Multi-Task Deep Learning for Movie Review Classification. In proceedings of the 4th IEEE DSAA. Tokyo, Japan, 10/2017.