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

Daily Work:

- Keep improving the reliability, availability and maintainability of distributed Apache Druid, which is coded in Terraform, hosted on AWS and comprised of EC2, Zookeeper, RDS, Route53, S3.
- Ingest massive batch 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, JFrog, 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. Communicate with external teams for SLO violations. Track down the root cause of incidents and update run books. Positively handoff at the weekly Ops meeting.

Achievement Highlights:

- Major contributor of Quantcast first real-time insights product to enable self-served advertising. This product consumes 1 billion streaming events per day with P95 latency of 30s. Fixed the bug of Druid's built-in IntermediateRowParser to allow skipping records without any desired labels.
- Owned the back-end of Pacing Visualization where daily budget goal is plotted alongside actual spend. This feature empowers customers to diagnose when they are having under or over delivery.
  - Incrementally synchronized from up-stream's Snowflake to PostgreSQL by exploiting JDBC drivers.
  - Used Mybatis Dynamic SQL and Type-Handler to deal with variant SQL queries and SQL Array type.
  - Accelerated time series aggregation by roughly 3.5 times through fine tuned Java Thread-Pool.
- Key stakeholder of the company-wide project to roll out advertising into global. Made API compatible with old underlying data and query behavior. Worked with PM to decouple multi-team deployment.
- Reduced the query concurrency by half by inserting new hierarchy aggregation dimensions via internal MapReduce. Sped up the re-harvesting by 16 times faster with Broadcast Join.
- Designed and implemented color deployment policy for Druid cluster to replace the awkward rolling update policy. It dramatically shrinks the cluster-wide deployment time from 1 workday to 1 hour meanwhile guarantees zero query down time.
- Tiered Druid cluster to support multi-tenancy to prevent queries, issued by different tenants, from intervening each other. This feature brings benefits to tune query performance in terms of query pattern.
- Wrote extendable probes, hosted on AWS Lambda, to emit data freshness into a centralized Datadog monitor. It compacts services having different SLO, segment granularity and data source type (batch or streaming) so as to reduce redundant code and maintenance workload, both current and anticipated.
- Others: Fixed sev2 bugs caused by jute.maxbuffer and mmap. Migrated Terraform back end with multi workspaces from S3 to Terraform Cloud. Integrated Fluentd to emit cluster logs to AWS ElasticSearch.

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

Built an internal distributed Spring Boot application to automate the ML (machine learning) lifestyle.

- Utilized Factory Pattern and Builder Pattern to automate the generation of offline training UI template and the serilization / de-serilization of training parameters in terms of the ML algorithm type.
- Used HttpClient to poll training status and applied MySQL optimistic lock to guarantee consistency.
- Dispatched customized ML models to application servers with internal persisted config management system and 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 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.

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.