Tejas Lodaya

https://tejaslodaya.com

tlodaya@purdue.edu

+1-765-409-0686

Research interests : systems (distributed, data-intensive computation), storage and ML

Education

- **Purdue University** — West Lafayette, Indiana
  M.S in Computer Science (advised by Dr. Jianguo Wang) — *2023*

Professional Experience

- **Swiggy** — Bangalore, India
  Technical Lead — *May 2019 - Aug 2021*
  - Built end-to-end platform to deploy deep learning models at scale using tensorflow serving. Achieved latencies of 15ms at peak 10k RPS using sidecar containers and gRPC network calls
  - Maintained and operated ML platform with over 5k feature jobs and ~150 scala models in production
  - Streamlined visibility and monitoring of ML models by feature pipeline quality checks, alerts on data drifts, erroneous models, and infrastructure failures. Reduced TAT for model failures from 1 day to 2 hours
  - Safely migrated 900+ production jobs to databricks as a result of complete spark infrastructure outage (S1 severity) in 7 hours. No degradation in business metrics or model prediction accuracy to any consumers of the runtime
  - Developed real-time and end-of-ride map-matching algorithm to snap driver GPS pings to the underlying road network. Powering accurate driver payouts, missing road detection, ETA predictions, and order assignments
  - Architected map-reduce style spatial querying and data manipulation engine for massive storage of point, line, and polygon data. Provides a real-time, cost-effective, and performant solution for efficient OLAP queries
  - Built a million-scale approximate nearest neighbour search using vector similarity to discourage fraudsters from reusing images to claim refund. Achieved latencies of 250ms at peak 3k RPS using ES, social graph and caching
  - Engineered a low-throughput, high-latency prototyping platform to serve python-based models at scale. Enables data scientists to conduct experiments and validate hypothesis without rewrite in high-performant scala/tensorflow
  - Deployed smart payments model into checkout springboot service with a peak throughput of 2mn requests per day. Generated real-time features from the order flow and wrote JUnit cases for compile-time testing
  - Devised a performance testing framework to measure how fast models can fetch features and produce results when deployed in critical software systems. Helps gauge whether model is production-ready and within latency budgets
  - Oversaw design and solutioning of platform's automated retraining capability, model orchestration framework, centralized feature store, and monitoring and alerting framework
- **Freshworks** — Chennai, India
  Senior Software Engineer — *Oct 2018 - May 2019*
  - Migrated legacy codebase from in-memory redis cluster to disk-based cassandra, reducing burn-rate by $250k per year. Implemented memcached to increase the key-fetch rate further and minimize latency
  - Architected database model for storing normalized term frequency and document frequency across articles, achieving average O(1) read and write speeds
  - Worked on language-agnostic spell-correct microservice achieving average search complexity of O(1), at the cost of pre-calculation time and storage space of n deletions
- **Noodle.ai** — Bangalore, India
  Software Engineer — *Jul 2017 - Sep 2018*
  - Built and orchestrated demand forecasting ecosystem for real-time consumption (using R). Wrote DAGs using airflow as the workflow schedule system to run batch jobs
  - Worked on scaling compute by employing SPMD on N cores using parallel backends like doSnow, doParallel in R
  - Developed an incremental learning framework using the global-local ensemble model, where global serves as a long term model and local serves as a short term model

Open Source Contributions

- **Variational Recurrent Autoencoder**: Unsupervised, feature-based time-series clustering algorithm in pytorch
- **Troop**: A simple library to perform chunkwise processing on data.frame across multiple cores of a single machine using SNOW clusters with a low memory footprint

Skills

Python, Scala, Golang, C++, Redis, PostgreSQL, DynamoDB, Kafka, Spark, Tensorflow, Airflow, Apache Flink