Andy Zhang

jz359@cornell.edu • (408) 839-8887 • github.com/jz359 • linkedin.com/in/jz359

EDUCATION

Cornell University August 2016 – May 2020

M.Eng., Computer Science GPA 3.90

B.S., Cum Laude, Computer Science // Minor, Electrical & Computer Engineering GPA 3.69

Relevant Courses (* = Teaching Assistant, † = Graduate Teaching Assistant):

CS 6850 – Structure of Information Networks	CS 6110 – Theory of Programming Languages	CS 4820 – Analysis of Algorithms
CS 6480 – Systems and Formal Methods	CS 5430 – System Security	CS 4787 – Large-Scale ML
CS 6450 – Advanced Computer Networks	CS 5414 – Distributed Computing†	CS 4410 – Operating Systems*
CS 6114 – Network Programming Languages	CS 5412 – Cloud Computing	CS 4320 – Database Systems

SKILLS

Languages: Java, Python, Rust, Go, C/C++, TLA+/PlusCal, JavaScript, SQL, OCaml, Bash

Technologies: Git, React, Redux, Kubernetes, Docker, Firebase, PostgreSQL, TensorFlow, OpenCV, Django

PROFESSIONAL EXPERIENCE

Google Inc., *Software Engineering Intern* Summer 2019

- Designed and implemented an end-to-end data auditing flow for the Cloud Jobs API Backend team.
- Migrated legacy servers, datastores, and messaging systems to a microservice architecture for new API resources.
- Implemented a diff pipeline to efficiently compute, store, track, and view over 400 terabytes of audited data in Spanner.
- Extended the data service layer to support a consistent Object-Relational Mapping (ORM) with Spanner tables.

Google Inc., *Software Engineering Intern* Summer 2018

- Designed and implemented a framework-agnostic experiment-disabling system for the Play Platform Team.
- Deployed microservices to interface with Play servers for mission-critical control over feature deployment.
- Improved development safety by reducing end-to-end experiment-disabling time from 10 minutes to 30 seconds.
- Collaborated with YouTube and infrastructure teams to design for compatibility with YouTube servers.

Cornell Unmanned Air Systems, *Software Engineer* Fall 2016 - Present

- Developed software for real-time Automatic Detection, Localization, and Classification (ADLC) of multiclass targets imaged from high-altitude autonomous aircrafts.
- Added client and server-side geofencing features to eliminate false-positive targets in the ADLC pipeline.

PROJECTS

Janus Store, *a sharded, replicated datastore in C++* Spring 2019

- Implemented the Janus protocol for efficient distributed transaction processing on a sharded, replicated key-value store.
- Designed a benchmarking framework to evaluate against state-of-the-art protocols like TAPIR and OCC+MultiPaxos.

Paxos, *a distributed consensus algorithm in Python* Fall 2018

- Implemented the Multi-Decree Paxos consensus algorithm for fault-tolerant state machine replication.
- Built a networking subsystem with randomized exponential backoff for communication between multithreaded servers.

APAX, *a distributed storage system in OCaml* Winter 2017

- Implemented the Raft consensus algorithm for storing and updating data across a distributed server cluster.
- Wrote an algorithm for log replication and leader election over RPCs to maintain consensus and fault tolerance.