Samiur Khan

samkhn.github.io | samiurkh1n@gmail.com | +1 (347) 599 5929

Work Experience

Meta — Menlo Park, CA
Production Engineer, WhatsApp — January 2024 to September 2024

- Automated load testing of the WhatsApp message bus for all messaging. Made library improvements to comply with internal safe configuration change policies. Written in Python and Erlang. Reduce load testing toil from a whole day to minutes.
- Reduced oncall toil for account management services. Ensure appropriate oncalls were alerted. Added health checkers for customer traffic in internal account management services. Written in Python.
- Onboarded group messaging service to use of SLOs, added alerts for error budget burndown. Written in Python.
- Rewrote customer facing registration service to improve DDoS defensibility, debuggability, and unblock OTP 27 rollout.
- Removed redundant copies in all internal SQL reads. Written in C++. Resulted in 1% CPU power usage drop fleetwide or about $20K/year.

Google — San Francisco, CA
Software Engineer, Cluster Turnup — August 2020 to March 2023

- Worked with three senior engineers on a more fault-tolerant lease-based scheduler for an internal cluster testing service. Increased service availability by implementing leader election at the load balancer. Exposes an API for tinkering with the underlying service scheduler, separating policy and implementation. Added timestamping to underlying request to enable benchmarking and improve SLI accuracy. Written in Golang. Protobuf for data exchange.
- For a distributed system linker, added idempotent operations for the command line utility, improving user perceived snappiness while maintaining logical correctness. Investigated and fixed file system leaks caused when the long-running linker is interrupted and evicted (these leaks happened on another machine so tool leak checker missed it). Reduced memory footprint by 20% and sped up post-interrupt recovery by replacing STL unordered_map with flat (probing, contiguous in memory) hash maps. Written in C++17.
- Wrote new documentation around integration testing for the distributed system linker.
- Participated in weekly on-call rotation, technical interviews, vetted systems during security incidents (CVE-2021-44228), resolved breakages, SLO violations and handled planned production capacity changes.

Ishu — Ann Arbor, MI
Co-Founder and CTO — January 2020 to April 2020

- Built and shipped issue tracking software designed for residential management firms, as a way to improve resident service reliability. Developed with Django (Python 3.7), Bootstrap (CSS), PostgreSQL. Deployed on Google Cloud.
- Tracked finances for the company. Ran technical and sales interviews with co-founder.
- Demo: youtu.be/bxNim90yWjI

Education

University of Michigan, Ann Arbor — Class of 2020
B.S.E., Computer Science and Engineering

- Worked with a team of seven to create a web search engine. Achieved 150 ms latency on queries within CONUS. Wrote Intel architecture cache-friendly, flat hash tables, string_view, optional monad and managed pointer libraries. Worked on the indexing system, implemented as a log structured merge tree without compaction. Improved throughput by factor of four by memory mapping userspace pages to the disk.
- Wrote a distributed key-value store. Odd number of Golang processes used append only logs to publish and consume updates in alignment with the Paxos distributed consensus algorithm.
- Worked with a team of three to write operating system primitives including fork(), exec(), and mmap(). Required implementing a red-black tree based scheduler and an MMU for dealing with page faults.

Skills

- *Programming*: performant data structures, efficient algorithms, profiling, benchmarking on x86 (Google Perftools and Benchmark), networked service design, Linux, Erlang, C++, Python, Rust, CMake, GNU make, grep, sed, Bazel/Buck, bash.
- *Engineering*: Distributed system design, API design, software unit and integration testing, SCM, documentation and technical writing, database design and migration. Familiarity with computer architecture. Business and engineering alignment/impact. User journey analysis.