Kasra Jamshidi

Vancouver BC, Canada · contact@kjamsh.com · https://kjamsh.com

Skills

C++, multi-threading, profiling, performance engineering, testing.
Distributed systems, fault tolerance, graph algorithms, big data processing.
Lua, Python, Docker, Cypher, PostgreSQL, JS, Elixir, Java, Scala.

Education

Simon Fraser University, BC, Canada

2019-Present PhD Computer Science
2014-2019 BSc. Computer Science

Experience

APR 2019 - PRESENT

Lead C++ Developer – *Parallel & Distributed Computing Lab*

- Lead all greenfield development and architectural decisions. **Supervise up to 4 developers** on various projects.
- Built a distributed, fault tolerant stream processing system for an RDMA-enabled cluster using C++23. Serves big data analytics queries on mutating graph datasets, sustaining an average output throughput of **200M (3.5GB) records per second**.
  - Deployed using Docker on a 32-node cluster
  - Custom lockfree arena allocator to reduce context switches in critical path
  - Custom Paxos implementation to take advantage of RDMA and provide Byzantine fault tolerance
  - Asynchronous RDMA network layer implementation
- Designed and implemented Peregrine, a programmable parallel graph mining system that is **700x faster** than the previous state-of-the-art with **8x fewer CPUs**, while using **100x less memory**. https://github.com/pdclab/peregrine
  - Performance scales nearly ideally with physical CPU cores (e.g., 48 cores lead to 41x speedup)
  - Handles datasets approaching memory limits using commodity machines (e.g., 32GB)
  - Custom lockfree aggregator
- Developed automatic query optimizer that improves graph mining execution speed by **10-34x** (saving **24 hours** or more on some queries) with end-to-end overhead in the milliseconds.

JAN 2017 - MAR 2018

Founding Developer – *Polly Language Exchange/Lingvu*

- Developed web chat app using OpenResty that pairs users seeking to learn each other's native languages, leveraging Redis queues to fairly match users. Implemented a microservice for finding nearby conversation partners using Phoenix web framework for Elixir and PostgreSQL geospatial.
  - Backend: OpenResty (NGINX), Lua, Redis, Phoenix/Elixir, PostgreSQL.
  - Frontend: WebRTC with vanilla JS, Angular 2.
  - Deployment: Vagrant and DigitalOcean.

JUN 2016 - DEC 2016

Software Intern – *Nexedi France*

- Developed a React web-app to compare open-source enterprise solutions. Implemented offline-capable indexing and fuzzy search using Levenshtein distance.
- Wrote documentation and tutorials implementing sample Python data science analyses using scikit-learn and other common libraries on the Wendelin Exanalytics system.