Ke Li

(780)-885-0852 | kegrad2023@gmail.com | Toronto, ON, Canada
damianli.com | Linkedin

Education

University of Toronto Toronto, ON
Master of Engineering in Computer Engineering & Identity, Privacy and Security (IPS) *Jan. 2024 – Dec 2025*

- Coursework: Computer Security, Cloud Computing, Deep Learning & Neural Network, Parallel Programming, Performant System with Rust
- GPA: 4.0/4.0

University of Alberta Edmonton, AB
Bachelor of Science in Computer Science *Sep. 2019 – Aug 2023*

- Coursework: Operating System, Computer Networks, Computer Architecture, Web Development, Mobile App Development, Database Management, Machine Learning, Agile Methodology
- Awards: *Dean's Honor Roll (22-23)*

Experience

Student Developer Jan. 2023 – Apr. 2023
University of Alberta ALT Lab *Edmonton, AB*

- Developed a website that generates interactive word graphs based on Cree words and their domains using React, D3.js and Docker.
- Reviewed 50+ PR and contributed 4K+ lines of code to the codebase via Git.
- Utilized Jest for unit testing and Cypress for E2E testing.

Back-End Engineer Intern May 2021 – Aug 2021
Nandou Six Star System integration Co., LTD *Wuhan, China*

- Wrote backend code that handled external HTTP requests from third party endpoints.
- Gained experience in Linux, Gunicorn, Nginx, SQLite and Django REST framework.

Projects

VeloxDB Github Aug. 2024 – Present

- Built a high-performance **Key-Value Storage** Library from scratch, support highly customizable database operations API and storage of basic C++ data types as keys and values.
- Implemented **Memtable** module using Red-Black tree for in-memory key-value storage and **PageManager** module for managing data in persistent media storage.
- Adding Expandable **Buffer Pool** module with LRU, CLOCK and RANDOM Eviction Policies.
- Struct SST file with static B+ Tree.
- Created **SSTFileManager** module to handle SST files. Integrated **Protocol Buffers** for serialization and deserialization of SST files and index metadata.
- Wrote unit tests and benchmark for all modules using **Google Test** to ensure correctness and reliability.
- **Utilized**: C++, CMake, Google Test, Protocol Buffers

Distributed Linux Performance Analysis and Monitoring System Jun. 2024 – Jul. 2024

- Developed a Docker-based setup to build project environments with dependencies, facilitating easy deployment across multiple servers.
- Implemented the **Monitor** module using the Factory Design Pattern to create an abstract monitoring interface, including CPU status, system load, software interrupts, memory, and network monitoring.
- Built a Distributed System using **gRPC**; Deploying server on target machines and client library used by monitor and display modules, ensuring low coupling and high modularity.
- Utilized **Protocol Buffers** for serialization to define comprehensive data structures for the project.
- **Utilized**: C++, CMake, Docker, gRPC, Protocol Buffers, Qt

Technical Skills

Languages: C++, Rust, Python, C, JavaScript, Java
Frameworks: React, Django, Cypress, Jest, OpenMP, Material-UI, Ant-Design
Developer Tools: CMake, gRPC, protobuf, Qt, Git, Docker, Kubernetes, AWS
Libraries: Scikit-learn, Pandas, NumPy, Matplotlib, Scapy, D3.js, ReactFlow, Mininet