Jiaqi Li

Software Engineer Intern | jiaqili4@andrew.cmu.edu | 650-281-8875 | Mountain View, CA | linkedin.com/in/lijiaqi1579

EDUCATION

Carnegie Mellon University Silicon Valley Aug. 2023 – Expected Dec. 2024
M.S. in Software Management Mountain View, CA

- Relevant Courses: Foundation of Software Management, Integrated Thinking of Innovation, Entrepreneurship & Innovation, etc.

Wuhan University **GPA : 90.03/100** Sep. 2020 – June 2023
M.Eng. in Computer Science Wuhan, China

- Relevant Courses: Advanced Database Technology, Parallel Algorithm, Advanced Computer Graphics, Cloud Computing, etc.

Wuhan University **GPA : 87.96/100** Sep. 2016 – June 2020
B.Eng. in Computer Science Wuhan, China

- Relevant Courses: Computer Architecture, Operator System, Database, Data Structure, Object-Oriented Programming, etc.

WORK EXPERIENCE

Alibaba Cloud, Virtual Private Cloud Team (C/C++) June 2022 – Sep. 2022
Software Engineer Intern Hangzhou, China

- Deployed the integrated OpenStack Neutron with VMWare NSX to achieve cross-platform management and control plane.
- Adopted the Token Bucket Algorithm to improve distributed QoS performance on the gateway of VPC.
- Refactored the virtual gateway management program with the idea of object programming, reducing the total code amount by 50% and greatly decreasing the workload of iterative development.

PROJECTS

High-Performance Framework for Machine Learning Computation (C++/Python/Cpython) Sep. 2020 – Oct. 2022

- Used the LLVM Pass methods to optimize the computation graphs based on Relay IR, e.g., operator fusion.
- Designed efficient operator implements with TIR in the CUDA level and backward operators to support model training.
- Adopted a fine-grained graph partition method and pipeline method to achieve model paralleling on heterogeneous devices.
- Developed an efficient framework for deep learning computation based on TVM compiler, which could achieve a speedup ranging from 1.5x to 5x comparing with TensorFlow and PyTorch when conducting model inference.

Automatic Graph-Level Compiler for Model Inference (C++/Python/Cpython/Rust) Sep. 2021 – Dec. 2021

- Employed Python with Static Single Assignment principle to implement Domain Specific Language for graph-level compiler, achieving the functions of graph generation, graph analysis, operator generation, graph optimization, etc.
- Designed the algorithms to automatically optimize computation graphs with C++ language, such as optimal graph rewriting.
- Adopted a max-flow min-cut algorithm and dynamic programming to reduce the complexity of graph rewriting from hours to seconds.

Embedded Multimedia Player (C/C++) Sep. 2019 – Nov. 2019

- Designed the program on Keil platform with C languages to build a multimedia system based on STM32.
- Transplanted UCOSII operating system to RT-Thread Nano to achieve multitasking feature.
- Built picture display module, touch screen induction module, audio/video play module, and audio record module.

WeChat Mini App - School Bus Assistant (Java/JavaScript/SQL) Sep. 2018 – Dec. 2018

- Developed student-side and driver-side applets to provide functions of location display, online payment, appointment, etc.
- Applied Java Servlet and MySQL with Spring Boot on AWS to design the database and the back-end program; Designed UI/UX with WXML and WXSS in the view layer and JavaScript in the logic layer.

RESEARCH EXPERIENCE

GLite: A Fast and Efficient Automatic Graph-Level Optimizer for Large-Scale DNNs Sep. 2021 – Feb. 2022
First Author & Oral Presenter, 59th Design Automation Conference (DAC) San Francisco, CA

- Designed a Domain-specific Language to support users to create computation graph objects, load graphs, profile graphs, optimize graph, etc; Developed Python API to call the Cython-packaged C++ implements.
- Applied matched substitutes on input graphs with backtracking methods to achieve automatic optimal graph generation.
- Proposed a max-flow min-cut graph partition method to reduce complexity and a dynamic programming method to avoid repeated subgraph matching; The compiling time is decreased from hours to seconds.

2017 & 2018 TREC Video Retrieval Evaluation Jan. 2017 – Aug. 2018
Author & Co-Developer, the National Institute of Standards and Technology (NIST) Wuhan, China

- Optimized a pre-trained Places365-VGG model on PyTorch framework for scene recognition based on global features.
- Fine-tuned the pre-trained ILSVRC13-VGG model, combined with Bag-of-word method to detect and classify local objects in the video on the Caffe-SSD framework; Visualized the object detection results by Matlab.
- Implemented a person re-identification methods integrating the local SSD feature with the global feature from classification networks.
- Developed an interactive Windows program for dataset manual tagging by C#.

SKILLS

Programming Languages : C/Cpp, Python, Java, SQL, Shell, CSharp, JavaScript, Rust, HTML/CSS, CUDA, Perl, etc.
Tools & Frameworks : Git, Linux, Matlab, PyTorch, TensorFlow, OpenStack, Docker, Spark, Visual Studio, JetBrains, MySQL, etc.