Vivek **Khandelwal**

(+91) 7879835123 | vivekkhandelwal1424@gmail.com | live:vivekkhandelwal1424

Work Experience

Nod.ai
India

AI Compiler Engineer — *Nov 2021 - Present*

- I am working on adding support and optimizations for deep learning models in the torch-mlir. I also actively contribute to the open-source torch-mlir framework. `https://github.com/llvm/torch-mlir/commits?author=vivekkhandelwal1`.

Qualcomm
Bangalore, India

Machine Learning Compiler Engineer — *July 2021 - Oct 2021*

- I worked on optimizing Machine Learning models on Qualcomm processors for inferences using Glow compiler infrastructure.

Education

Indian Institute of Science
Bangalore, India

M.Tech in Computer Science and Engineering — *Aug. 2019 - June 2021*

- CGPA : 8.5/10
- Courses: Computer Architecture, Compiler Design, Theory and Practice of Computer Systems Security, Deep Learning for Natural Language Processing, Machine Learning, Practical Data Science, Linear Algebra and Probability, Design and Analysis of Algorithms, Cryptography.

Madhav Institute of Technology & Science
Gwalior(M.P.), India

B.E. in Computer Science and Engineering — *Aug. 2015 - July 2019*

- CGPA : 7.77/10

Publication

MLIR-Based Code Generation for GPU Tensor Cores

Published in the 31st ACM SIGPLAN International Conference on Compiler Construction (CC'22)

- The work presents an MLIR based code-generation pipeline for Tensor Cores on NVIDIA GPUS, which achieves performance comparable to that of hand-tuned libraries like cuBLAS, cuBLASLt, and cuTensor. Link: `https://dl.acm.org/doi/10.1145/3497776.3517770`

Skills

Programming	C++, C, Python. Previously worked with HTML, CSS, MySQL
Technologies	MLIR, Torch-MLIR, LLVM, Clang, Pytorch, Linux, Git, Wordpress
Libraries	CUDA, Numpy, Keras, Pandas, Scikit-learn, Matplotlib, OpenAI Gym & Baselines

Projects

Automatic Code Generation for GPU Tensor Cores using MLIR
June 2020 - June 2021

M.Tech. Project | Under the guidance of Prof. Uday Kumar Reddy B.

- We used the MLIR infrastructure to build a transformation and lowering pipeline to automatically generate near-peak performance code for matrix-matrix multiplication (matmul) and matmul fused with simple point-wise operators targeting tensor cores on NVIDIA GPUs.
- We achieved performance that is 0.80× to 1.60× of cuBLAS on NVIDIA's Ampere microarchitecture.

Loop Interchange Pass in MLIR
May 2020 - June 2020

- Implemented a loop interchange pass on Affine Dialect in MLIR driven by a cost model that takes in account spatial and temporal locality - both self and group and parallelism(for multi cores).
- The pass finds out the optimal loop permutation which optimizes the code for both spatial & temporal locality. It also tries to bring the parallel loops to outermost position to reduce the frequency of synchronization and maintains the balance between locality and parallelism.

Tool to Support Nested Functions in C
Mar. 2020 - Apr. 2020

- Built a tool to perform source to source transformation in Clang(LLVM's C/C++ compiler front end) to support nested functions(closures). The tool was built using Clang LibTooling and ASTMatchers.
- This tool enables a programmer to write a C program with nested functions emulated using labeled blocks by taking it as an input and outputting a transformed valid C program.