| 9th WSS Workshops |

CUDA PROGRAMMING

CUDA Programming In Python

Amir Mirzaei

Reminder

- Nvidia GPUs are very well suited for massively parallel applications without complex control paths
- CUDA is a framework for writing applications running on GPUs.
- CUDA programming language is an extension of C++ programming language.

CUDA for Python: Motivation

- Writing GPU applications typically require experience and time.
- Not all users know how to write programs in C++.
- Python is commonly used by scientists and research in many different fields. (Just like MatLab few years ago)
- Many tasks are much simpler in Python.
- GPUs give a considerable performance improvement and it’s impossible to ignore them in more popular languages such as Python.

CUDA for Python: How?

- Writing functions in C++ (CUDA) and calling them from Python using **Python C bindings**.
- Using runtime compiler (nvrtc) and cuda-python to build GPU kernels in python programs.
- Using Numba to build GPU kernels in python programs.
- Using pre-developed libraries (such as cupy, cudf, tensorflow & torch)

GPUs for ML

- ML models are often just a very large number of algebraic operations.
- Most of these operations can be executed concurrently.
- Most ML models don’t have complex control paths.
- Modern GPUs include special arithmetic units for tensor operations, which are highly common in ML models/

Therefore they are the type of application that can benefit from GPU programming.

TensorRT

- TensorRT is a library developed by NVIDIA for faster inference on NVIDIA graphics processing units (GPUs).
- It can give around 4 to 5 times faster inference on many real-time services and embedded applications.

TensorRT: How it works?

1. Weight and Activation Precision Calibration
2. Layers and Tensor Fusion
3. Kernel auto-tuning
4. Dynamic Tensor Memory
5. Multiple Stream Execution

| 9th WSS General Meeting |

THE END