Triton for HPC

Scalable Tools Workshop 2024

https://bit.ly/46JJlrD

Triton Introduction

- Open-source Python-like language
- Compiles Triton language code into CUDA binaries
- Cross-Platform. Supports NV, AMD and Intel GPUs
- Enables researchers to write high efficient GPU code, without any CUDA experience
  - Most of which are on par with what an expert produces

Pros for using Triton in HPC

- Triton language is more like a “high-performance Python”
  - Compared to C++/Fortran, the syntax is similar to Python
    - Saving Programmer’s effort
    - Faster development, faster iteration
  - The performance of Triton Kernels achieves ~80% of CUDA counterparts

- Triton can be used in some libraries not well supported by present libraries.
  - e.g. matrix operations like solving eigenvalues and jacobies

Cons and Difficulties - 1

- Data format issues.
  - Triton performs well in low precision data formats, but not in high precision ones.
    - E.g. fp16, fp8 and even fp4 is what Triton is good at.
    - However, most HPC applications requires at least the precision of fp64.
    - Triton doesn’t support fp64 for now.
    - Although it is possible to develop fp64 support for Triton, performance tuning is needed.

Cons and Difficulties - 2

- Tech Depts
  - Most of HPC libraries were written in C++/Fortran
    - C++ HPC libraries were written by professional library programmers
      - Computational scientists only need to focus the usage of these libraries. They don’t care about the development cost
    - Fortran language is more similar to mathematical language used in matrix computation
      - Fortran language and BLAS have been debugged for 20+ years.
        - Not necessary for a Triton refactor.
    - Python is mostly used for data loading

Potential Application

- Sparse Tensor Operations
  - AMR(Adaptive Mesh Refinement): 3D space -> 3D cubes
  - Calculating vortex in turbulence
- GPU outperforms CPU in fp64 computation.
  - Thus, GPU programming is inevitable
  - Which is Triton really good at

Attendees

Ben Woodard

Hao Wu

Keren Zhou

William Jalby

Yuning Xia