Initial Assessment of the AMD MI50 GPGPUs for Scientific and Machine Learning Applications

Keith Obenschain, Douglas Schwer, and Alisha Sharma
Laboratories for Computational Physics and Fluid Dynamics, Naval Research Laboratory
Contact: emerging_architectures@nrl.navy.mil

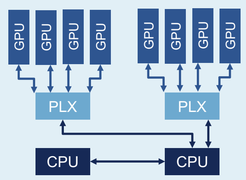
Competition in the High-Performance Computing GPGPU market has emerged with GPGPUs from Advanced Micro Devices (AMD) and Intel targeting future Exascale class systems. The new AMD Radeon Instinct MI50 hints at the capabilities of AMD's future GPUs. This study takes a first look at the MI50 performance on characteristic scientific and ML applications.

Highlights

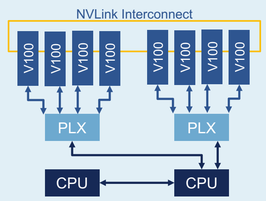
- Able to port ("hipify") most CUDA code to HIP with the `hipify-pearl` tool
- Manual porting required for optimal performance in several critical sections
- Competitive performance for both scientific and machine learning codes
- Debugging and profiling tools will be key for future work
- **Initial results are promising, the overall environment needs to mature**

System Configurations

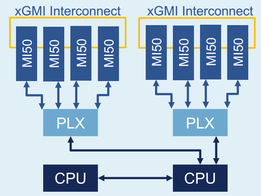
NVIDIA PCI Express System: Dual-Socket Intel Xeon 6130, GPU-GPU communication via PCI Gen 3 x16 (GTX only), Centos 7.6, nodes have 8 NVIDIA GTX 1080Ti (11GB) or RTX 2080Ti (11GB).



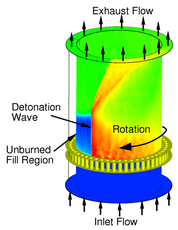
NVIDIA NVLink System: Dual-Socket Intel Xeon 6142, GPU-GPU communication via NVLink, Ubuntu 18.04, nodes have 8 V100 GPUs (32GB).



AMD MI50 System: Dual-Socket Intel Xeon 6230, Supermicro SYS-4029GP-TRT2, GPU-GPU communication via xGMI, Centos 7.6, node has two hives of 4 AMD MI50 GPUs (32GB)



Computational Fluid Dynamics: Modeling a Rotating Detonation Engine



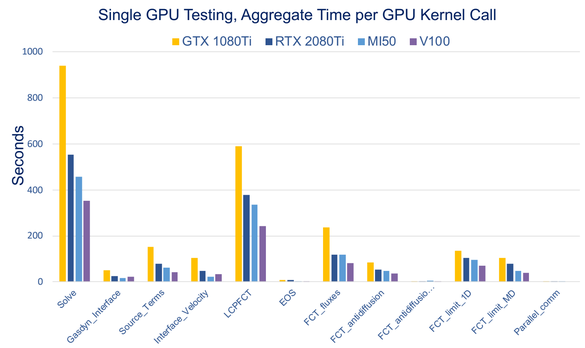
- Thermodynamically the detonation cycle is much more efficient than the Brayton cycle due to pressure rise through the combustion process
- Difficult to realize this advantage due to unsteady nature of the detonation process and extreme pressures, heating rates, and wave-speeds
- ***Rotating detonation engines are the most promising method for achieving the efficiency gains of the detonation cycle***

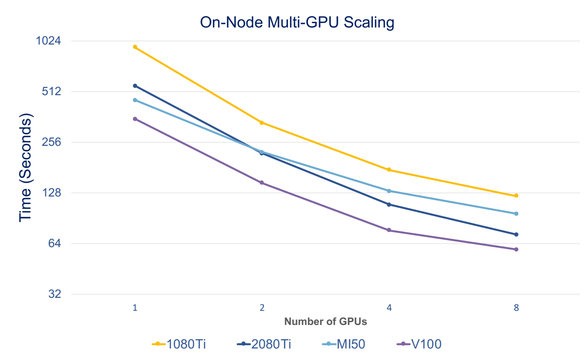
CUDA GPU Implementation based on 30+ years of research in high-fidelity modeling

- Production code, used in conjunction with experimental efforts
- Multidimensional FCT algorithm developed at the Naval Research Laboratory
- Integrated with detonation models (IPM) and detailed chemistry
- Earlier versions used extensively for basic detonation studies and PDE research
- 20,000 lines of CUDA (9.2) code
- Does not use any of the numerical CUDA libraries
- 2080Ti is competitive despite slower memory bandwidth and rated double-precision capability

AMD ROCm HIP Implementation

- Based on CUDA version using hipify-perl to convert CUDA to AMD GPU compatible code
- Most kernels worked perfectly with hipify and hipcc compiler
- A few manageable issues that worked well with CUDA but not with HIP
  - `static __global__` kernels did not work appropriately
  - One kernel had to be rewritten due to large stack allocation causing hang
  - Had to remove the `switch/case` constructs and replace with `if/if else`
- In general less stable than NVIDIA/CUDA, but still works





Machine Learning: Training ResNet50

Is ROCm-enabled hardware and software viable for machine learning workloads?
To answer this question, we evaluated distributed TensorFlow 1.15 training for image classification across multiple nodes and devices on both NVIDIA and AMD hardware.

Benchmark

Average ResNet-50 v1 training throughput on ImageNet

Training Options

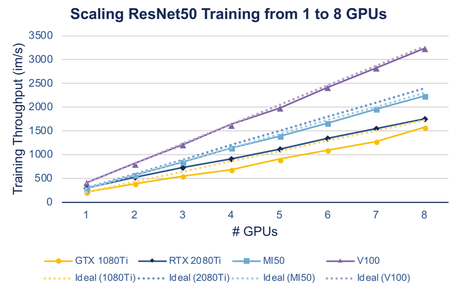
```
$ singularity exec [--nv|--rocm] <container> \
    python tf_cnn_benchmarks.py --num_gpus N \
      --variable_update replicated --all_reduce_spec nccl \
      --model resnet50 --data_name imagenet --optimizer momentum \
      --nodistortions --gradient_repacking 1 --ml_perf
```

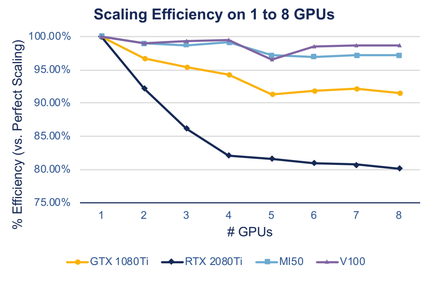
CUDA vs. ROCm

We used the following ROCm-specific libraries for the MI50s:

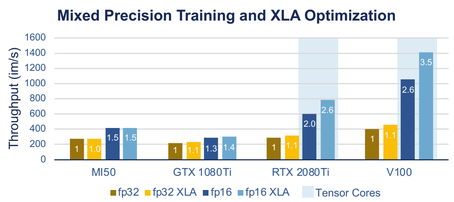
- ROCm 3.3
- RCCL, the ROCm Collective Communications Library
- TensorFlow 1.15.2 (includes RCCL support)

More details and code samples can be found at
https://emerging-architectures.github.io/amd_mi50_benchmarks

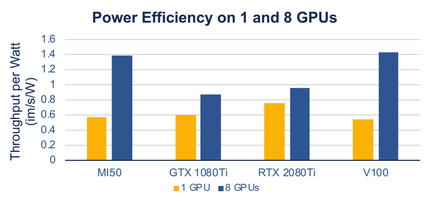




Scaling on 1–8 GPUs. The two charts above show scaling ResNet50 training of ImageNet on NVIDIA and AMD hardware. The left diagram shows the raw throughput for each condition, and the right diagram shows the scaling efficiency. The V100 clearly has the highest training throughput, with the MI50 performing similarly to the 2080Ti. System topology (including GPU-GPU links, if present) are described in the diagrams to the left.



Mixed precision training on 1 GPU. Tensor Cores in newer NVIDIA hardware lead to a significant speedup in FP16 training. XLA optimization magnifies this effect.



Training power efficiency. The RTX is most efficient for training on a single GPU. While the MI50 is slower than the V100, they have a similar power efficiency.

This work is sponsored by the Office of Naval Research through the Emerging and Alternative Architectures project under the NRL 6.1 Materials Science Task Area.