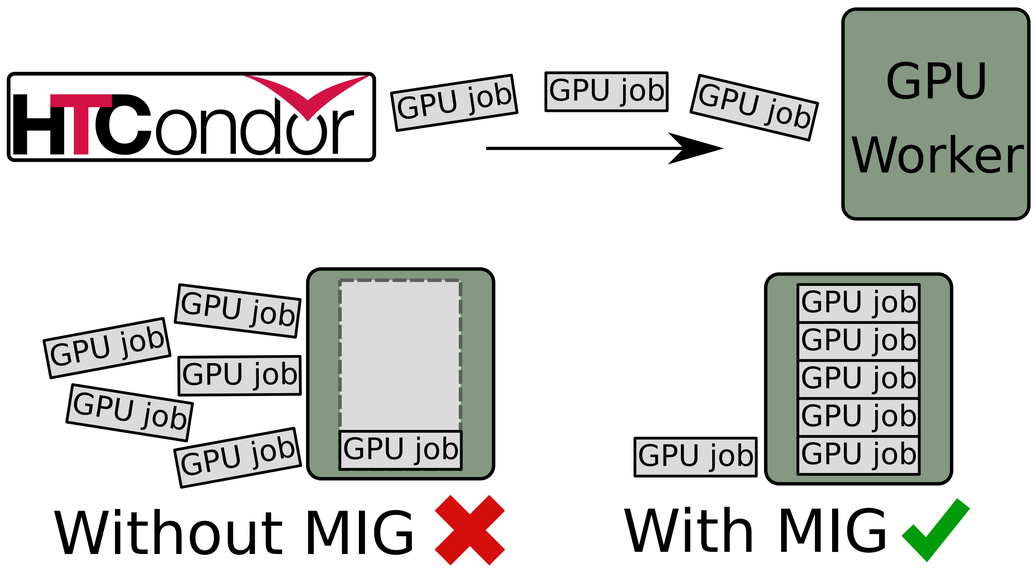
Optimized GPU usage in High Energy Physics applications

Tim Voigtländer*, Günter Quast, Manuel Giffels, Matthias J. Schnepf, Roger Wolf

Overview



Multi Instance GPU (MIG)[1] and similar concepts can increase the throughput of a batch system by splitting a single GPU into multiple smaller ones.

GPU granularity

Combining GPUs and batch systems

- A single GPU posesses a significant amount of processing power.
- One GPU can only be assigned to a single job.
- Jobs differ significantly in GPUs utlization.
  - The remaining portion of the GPU idles during the job.
- This issue is exacerbated by the use of ever more capable GPUs.

Solution a): Merging batch jobs

Circumvent the issue by collecting multiple GPU tasks into one job and then run all scipts concurrently on one GPU.

- Can lead to interference on older GPUs.
- Some of the schedduling has to be done by the user.
- Spread of the scripts can leads to inefficient resource usage.
- Larger job sizes inhibit rules out some possible job slots.

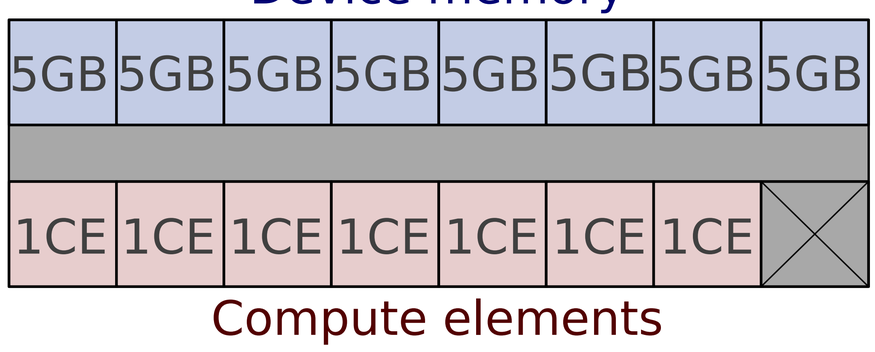
⇒ **This approach solves one issue, but adds management for the user.**

Solution b): Split the GPU up

Reduce the granularity of the GPU by splitting it into smaller pieces. One way to achive this is through the NVIDIA MIG service.

- Only usable with certain NVIDIA GPUs.
- GPU can be split into up to 7 seperate GPU instances.
- The instances can be used as independant GPUs by the batch system.
- Effectively exchanges one large GPU for up to 7 smaller ones.

⇒ **A good solution if the necessary hardware is available.**



Sketch of the MIG building blocks of an A100 GPU. One Compute elements is reserved during MIG mode.

GPUs in the CMS collaboration

The computing challenges in HEP

- LHC's luminosity will dramatically increase with the coming run periods.
- The cost of electricity, and therefore the operating cost of all hardware has risen significantly.

⇒ An increasingly heterogeneous computing infrastructure is necessary to keep up with the demands. This also includes GPUs.

Current GPU applications

Additional to end-user analyses, CMSSW framework support GPUs for:

- Reconstruction
- Simulation
- Generation
- Neural Network (NN) training and interference

GPU resources at TOpAS

What is TOpAS?

- **TOpAS** is a **T**hroughput **Op**timized **A**nalysis **S**ystem.
- It is a shared Tier 3 analysis cluster at GridKa that provides GPUs.

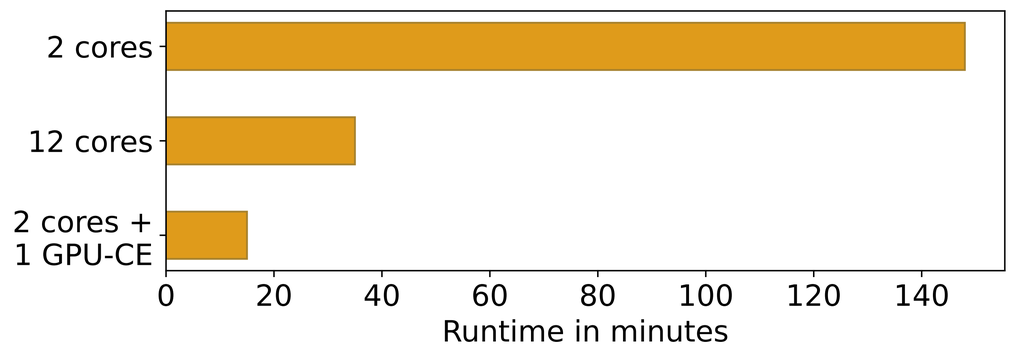
Features

- Access via local groups batch system and via Grid CEs for collaborations.
- All jobs are run in a containerized environment.
- 56 GPUs, in a mix of A100, V100 and V100S are accessible.
- The GPUs can be opportunistically used via an HTCondor[2] batch system.
- Single- and Multi-GPU jobs can be easily requested through ClassAds.
- A 200Gbit/s network connection and an 1PB cache are available.

Benchmark of an A100 GPU in MIG mode

GPU performance benchmark

An exemplary NN training is performed, the CPU used for this is an AMD EPYC 7662 64-Core. The GPU used is an NVIDIA A100 GPU.



Runtimes of the NN training using different hardware.

- Trainings are greatly accelerated by adding more CPU resources.
- They are even further accelerated by added a fraction of a GPU instead.

Throughput benchmark

As an example analysis, two runs of 414 trainings with similar input data were performed on a machine with 8 A100 GPUs.

- Without splitting the GPUs up
  - Total runtime: 9 hours 54 min
- Each GPU split into 7 MIGs
  - Total runtime: 1 hour 17 min

Upcoming Goals

- Test of the MIG approach in production
- Expansion of the available MIG ClassAdds
- Dynamic creation of MIGs

[1] NVIDIA MIG: https://www.nvidia.com/en-us/technologies/multi-instance-gpu/
[2] HTCondor: http://research.cs.wisc.edu/htcondor/

*tim.voigtlaender@kit.edu

ACAT 2022, Bari, poster #185