Provision and use of GPU resources for distributed workloads via the Grid

Daniel Traynor[1,*] and *Terry* Froy[1,**]

[1]Queen Mary University of London, Mile End Road, E1 4NS, UK

Abstract. The Queen Mary University of London WLCG Tier-2 Grid site has been providing GPU resources on the Grid since 2016. GPUs are an important modern tool to assist in data analysis. They have historically been used to accelerate computationally expensive but parallelisable workloads using frameworks such as OpenCL and CUDA. However, more recently their power in accelerating machine learning, using libraries such as TensorFlow and Coffee, has come to the fore and the demand for GPU resources has increased. Significant effort is being spent in high energy physics to investigate and use machine learning to enhance the analysis of data. GPUs may also provide part of the solution to the compute challenge of the High Luminosity LHC. The motivation for providing GPU resources via the Grid is presented. The installation and configuration of the SLURM batch system together with Compute Elements (CREAM and ARC) for use with GPUs is shown. Real world use cases are presented and the success and issues discovered are discussed.

1 Introduction

There are several motivations driving GPU deployment and use:

- GPUs are a commodity, programmable parallel architecture, ubiquitous as CPUs but offer significantly more parallel “streams”.
- GPUs are significantly faster than CPU for appropriate problems and GPU optimised workflows often scale better when adding additional GPUs .
- GPU Performance (FLOPS) per watt is better than CPUs.
- GPUs Performance (FLOPS) per $ is better than CPUs.

However, real world costs and usability mean that many of these points are not as clear cut as some would make out.

When considering the purchase and use of GPUs it’s important to consider the cost and benefit of the GPU over using just CPUs. The typical cost of a high end GPU is about the same as the cost of the server that hosts the GPU. The power usage of a GPU is also about the same as the server. If a GPU provides a speed up of factor of 2 then it becomes value for money to buy the GPU but this requires the GPU to be used 100% of the time. If the GPU is only used 10% of the time then you need a speed up of a factor of 10 to make the purchase

[*] e-mail: d.traynor@qmul.ac.uk
[**] e-mail: t.froy@qmul.ac.uk

Table 1. Capabilities and performance of selected GPUs

	NVIDIA K40	NVIDIA V100	NVIDIA RTX280 SUPER	AMD MI60
RAM	12GB ECC	32GB ECC	8GB	32GB ECC
Memory Bandwidth	288GB/s	900GB/s	496GB/s	1024GB/s
32bit TFLOPs	5	14	11.15	14.7
64bit TFLOPs	1.68	7	0.349	7.4
16bit TFLOPs	N/A	28	22.3	29.5
8bit TFLOPs	N/A	112	89.2	59

of a GPU value for money. It is also important to note that many benchmark comparisons quote the speed improvement of a GPU over a single core of a CPU [1]. With a typical server now containing 32 or more cores, the speed up improvement of using a GPU can look greatly exaggerated. Of course if the speed to get a result is important, e.g. in a trigger system of an experiment, then the additional cost of GPUs may be worth while.

There are several different types of GPUs, some typical example found in table 1. The table reveals several issues:

- Different GPUs have very different performance,
- Retail GPUS, e.g. NVIDIA RTX2080 SUPER, are very bad at 64 bit computation. This is required in many HPC applications,
- Older GPUs are very bad at 8 or 16 bit precession calculations. Many machine learning analyses use 8 or 16 bit precision to speed up calculations. On older GPUs there will be a significant performance hit compared to newer GPUs and in some cases the analysis may not run.

2 Deployment

Compared to a few years ago deployment and use of GPUs has been significantly simplified. NVIDIA provides a well documented Linux software and hardware driver repository that is easy to install and maintain [2]. At QMUL we use the SLURM [3] batch system where jobs get exclusive use of a GPU. To add GPU resources in SLURM you just include the generic resource (Gres) to the node description in the SLURM configuration file, e.g.

```
...
NodeName=cn290 CPUs=32 Gres=gpu:teslaK80:4 RealMemory=128640 ...
...
```

Previously we enabled GPUs via a CREAMCE with the requirement that a user had to request a GPU in the jdl (GPU=1), this proved to be an issue for some users. CREAMCEs are no longer supported and are being decommissioned. They have been replaced by arcCEs [4]. This has simplified the use of GPUs by adding in arcce.conf to the subsection for the GPU queue

```
[queue:centos7_gpu]
...
slurm_requirements= -gres=gpu:1 -n4
...
```

Now all you have to do is submit a job to the centos7_gpu queue and you will get one GPU+4cores+12GB RAM.

3 Case studies

Several different experiments have used GPUs at QMUL with different levels of success.

3.1 Ice Cube

In the IceCube detector passing neutrinos are detected by their occasional interaction in the experiment which produces charged particles that in turn emit Cherenkov radiation. This light is then detected by photomultiplier tubes. GPUs have been used by the IceCube experiment for several years to simulate the propagation of these photons from Cherenkov radiation and this has been shown to be significantly faster than using CPUs alone [5]. Our initial deployment of GPUs at QMUL was driven by the desire to support the IceCube experiment and GPUs were extensively used by local researches and official production.

Occasional drop off in usage was noticed and could be related to changes in central configuration or Grid site misconfigurations after software and driver updates. However, it was upto the site to push to get the site back online for central production while local users were much more proactive. It should be noted the resources QMUL was able to provide were small compared with central IceCube production capability.

3.2 LHC and ATLAS

There are many activities at the LHC using and exploring the use of GPUs ([6], and several different tracks in this conference). QMUL has supported ATLAS use of our GPU resources for some time. One notable case is the "Hyper Parameter Scan with the Deep Learning Heavy Flavour Tagger" by the ATLAS Collaboration [7]. In this case two grid sites, Manchester and QMUL, contributed to the Grid GPU resources used by the analysis. It should be noted that even though deployment of the analysis is via containers there is still a need for drivers and local libs (e.g. CUDA Deep Neural Network library, cudnn) to be installed on the GPU worker node.

3.3 CERN@School

The CERN@school program is a project for secondary school pupils in the U.K. to use CERN designed Timepix detectors [8] in a variety of different experiments [9]. One such experiment was the LUCID detector on board the TechDemoSat-1 launched in late 2014. The detector was used to study the radiation environment in Low Earth Orbit [10]. As a schools based program local compute resources were limited and the data was analysed using computing resources provided by GridPP collaboration. A Metric Based Network approach to classifying the particles was used and the QMUL GPU (and storage) resources were required to significantly speed up the workflow compared to using the CPU alone.

3.4 MoDEAL

The MoEDAL experiment is looking for highly ionising particles, such as magnetic monopoles, in Nuclear track detectors. The basic process is to look for holes in the polymer sheets caused by the highly ionising particle. It was recognised that analysing the images could be done in an a automated approach using machine learning techniques. Using GPUs at QMUL MoEDAL have been developing new methods using Machine learning (Tensorflow) to identify magnetic monopoles signatures in Nuclear track detectors [11].

3.5 e-NMR

The e-NMR [12] project provides a software and compute platform for use in structural biology (bio-NMR). Some of the software used in this area of science can make use of GPU acceleration and the e-NMR project has attempted to make use of GPU enabled grid resources such as QMUL. e-NMR uses docker containers in udocker [13]. Udocker is a basic user tool to execute simple docker containers in user space without requiring root privileges. A major issue with running GPU enabled workloads at QMUL proved to be regular upgrades to the kernel and/or GPU drivers. This was an issue for some application containers, such as DisVis and PowerFit, which must be re-built with the corresponding GPU driver in order to work. This barrier to using GPU resources resulted in no significant use of GPUs at QMUL.

4 Conclusions

The deployment of GPUs at a site and making them available on the grid requires little additional effort than that for normal CPU resources. However, it is harder to get them used on a regular basis to justify the expense of the hardware. Not all GPUs have the same performance, functionality and software support. To make best use of resources at a site users will need knowledge of what resources are available at that site. Hardware development and related software support is in flux and this may cause problems especially if a site is supporting multiple hardware generations and vendors. In addition different software solutions require different balances of hardware features and it is not yet clear what the workload/workflow will be for HEP experiments. This in turn will impact hardware choices sites will have to make. Without accounting for GPUs usage in APEL or similar frameworks there is little motivation for sites to support GPUs unless funded by dedicated funds.

On one hand the present usage of GPUs means in HEP workloads means there is little justification for widespread procurement and deployment of GPU resources on the Grid while on the other hand there is extensive software development work, e.g. for the HL-LHC, that would appear to need these GPU resources in the future. This would appear to be a classic chicken and egg problem that will need to be confronted at some point in the future.

References

[1] Simon Blyth, Meeting the challenge of JUNO simulation with Opticks GPU Optical Photon Acceleration, Plenary talk CHEP2019.

[2] NVIDIA CUDA: https://developer.nvidia.com/cuda-toolkit

[3] A.B. Yoo, M.A. Jette, M. Grondona, SLURM: Simple Linux Utility for Resource Management, JSSPP **2862** (2003).

[4] M. Ellert et al., Advanced Resource Connector middleware for lightweight computational Grids, Future Generation Computer Systems **23** 219 (2007) .

[5] Dmitry Chirkin (For the IceCube Collaboration), Photon tracking with GPUs in IceCube, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **Volume 725**, 141 (2013).

[6] WLCG pre-GDB on GPU utilisation: https://indico.cern.ch/event/689511/

[7] Alessandra Forti et al, Hardware Accelerated ATLAS Workloads on the WLCG, ATL-SOFT-SLIDE-2019-068: https://cds.cern.ch/record/2665661

[8] M.G. Bisogni, M. Campbell, M. Conti, P. Delogu, M.E. Fantacci, E.H.M. Heijne et al., Performance of a 4096-pixel photon counting chip, SPIE **3445** 298 (1998). M. Campbell, E. Heijne, G. Meddeler, E. Pernigotti and W. Snoeys, A readout chip for a 64x64 pixel matrix with 15-bit single photon counting, IEEE Trans. Nucl. Sci. **45** 751 (1998).

[9] T. Whyntie and B. Parker, Investigating the inverse square law with the Timepix hybrid silicon pixel detector: a CERN@school demonstration experiment, Phys. Educ. **48** 344 (2013). T. Whyntie, J. Cook, A. Coupe, R.L. Fickling, B. Parker and N. Shearer, CERN@school: Bringing CERN into the classroom, Nucl. Part. Phys. P. **273–275**, 1265 (2016).

[10] P. Hatfield et al.,The LUCID-Timepix spacecraft payload and the CERN@school educational programme, JINST **13** C10004 (2018).

[11] Jonathan Hays, Machine Learning Monopoles and MoEDAL: https://indico.cern.ch/event/559774/contributions/2669803/attachments/1509702/2354134/MachineLearningMonopolesAndMoedal.pdf

[12] A.M.J.J. Bonvin, A. Rosato and T.A. Wassenaar, The eNMR platform for structural biology, J Struct Funct Genomics. **11(1)** 1 (2010).

[13] J. Gomes et el., Enabling rootless Linux Containers in multi-user environments: The udocker tool, Computer Physics Communications **232**, 84 (2018).