“GPUs on the grid” pre-GDB

Andrew McNab
University of Manchester
LHCb and GridPP

GPUs in WLCG preGDB

- Emphasis on mechanics of using GPUs via the grid
  - So not about applications of GPUs themselves

- Agenda (https://indico.cern.ch/event/689511/)
  - Two talks about sites (QMUL and Manchester)
  - Experiment talks: ALICE, LHCb, CMS on OSG
  - Discussion

- ~25 people, 50:50 local or via Vidyo
- Lots of useful discussion and questions

Dan Traynor: “GPUs at QMUL”

My GPUs

1* NVIDIA K40c
recycled
Dell workstation

Free

2* NVIDIA K80
HPE DL380

~£10K

1* NVIDIA 1080 Ti
founders edition.
Alienware Aura 2017

~£2.5K

3

Dan Traynor: “GPUs at QMUL”

Not all GPUs are the same

	K40	1080 Ti	V100
RAM	12GB EEC	11GB	16GB EEC
32bit(GFLOPS)	~5,000	~11,000	~14,000
64bit (GFLOPS)	1:3	1:32	1:2
16bit(GFLOPS)	NA	1:64	2:1
8bit (GFLOPS)	NA	4:1	8:1

Nvida GPUs only (cuda) best supported and most used, but should also consider AMD/intel GPUs/MICs/FPGAs solutions

5

Dan Traynor: “GPUs at QMUL”

Integrate with SLURM

- /etc/slum/slum.conf

```
...
# Configure support for our GPUs
GresTypes=gpu
...
NodeName=cn456 CPUs=8 Gres=gpu:teslaK40c:1 RealMemory=11845 Sockets=1
CoresPerSocket=4 ThreadsPerCore=2 State=UNKNOWN CoreSpecCount=1
MemSpecLimit=768
```

- /etc/slurm/gres.conf

```
...
NodeName=cn456 Name=gpu Type=teslaK40c File=/dev/nvidia0
```

- /etc/slurm/cgroup.conf

```
...
ConstrainDevices=yes
```

- /etc/slurm/cgroup_allowed_devices_file.conf

```
...
/dev/nvidia*
```

- submit jobs: `sbatch --gres=gpu:1 -n1 test_gpu.sh`

7

Dan Traynor: “GPUs at QMUL”

Integrate with Cream CE

- Development of new CREAM CE version, specifically for CentOS 7.https://wiki.egi.eu/wiki/GPGPU-CREAM
- QMUL is using a patch for our SL6 Cream CEs.
- Introduced new JDL parameters that will be passed to the batch system: GPUNumber, MICNumber and GPUModel. This works with SLURM and should work with SGE. e.g.
  - GPUNumber=1;GPUModel=teslaK80;
- At present our users submit direct to the CE.

9

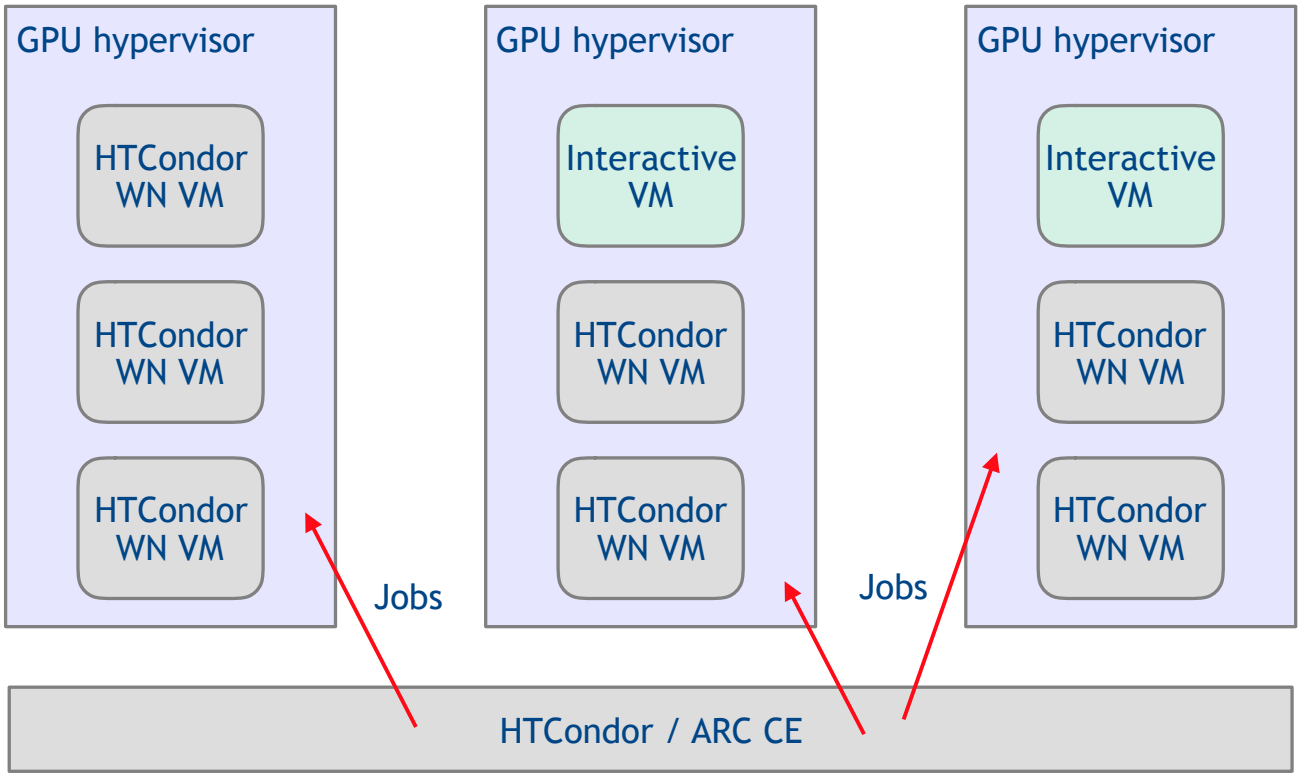
Andrew McNab: “GPUs at Manchester”

GPU machines

- Two K40 machines from 2014
  - Interactive: 2 x K40 , 2 Xeon 8-Core HT, 128 GB RAM
  - Batch: 4 x K40, 2 Xeon 12-Core HT, 128 GB RAM
  - K40 has 12GB GPU memory
  - Both machines 4U rack mounted
- Three V100(PCIe) machines from 2018
  - Each 3 x V100, 2 Xeon 8-Core HT, 128 GB RAM
  - V100 has 16GB GPU memory
  - 4U rack mounted

Andrew McNab: “GPUs at Manchester”

Configuration: 2018 machines



Andrew McNab: “GPUs at Manchester”

Experiences / conclusion

- GPU code, hardware etc solid and straightforward to operate
- Users take to interactive resources very easily
- They can be helped to get things up and working on the batch/grid resources
  - Some papers published using these results
  - But it’s always tempting just to stay with interactive resources once you’ve finished debugging there
- Experiments that are organised (eg Icecube) can keep resources busy with workloads managed centrally
- Our aim with the 2018 VM-based configuration is flexibility, so we can change the balance of interactive vs batch

David Rohr: “GPUs in ALICE in Run 3”

Run 3 processing

- **Which of the ALICE Run 3 computing workload can benefit from GPUs?**

Processing type	Location	GPU usage
Synchronous real-time reconstruction	O2 Farm	yes
Asynchronous raw-data reconstruction on the O2 farm.	O2 Farm	yes
Asynchronous raw-data reconstruction elsewhere	GRID	If possible
Monte-Carlo Reconstruction	GRID	If possible
Monte-Carlo Simulation	GRID	Not foreseen yet, but also not excluded. However this depends to a large extend on the MC libraries.
Analysis	GRID	Not foreseen yet, but we will certainly have a look when the GPU reconstruction is in good shape.

David Rohr: “GPUs in ALICE in Run 3”

Details on the GPU implementation

- **This summary reflects the approach currently used in the HLT.**
- **Since it has proven successful, we plan to follow the same strategy for Run 3.**
  - We employ a single source-code, that can run on CPUs (with OpenMP), with CUDA, and with OpenCL.

  - **CPU and GPU tracker (in CUDA) share common source files.**
  - **Specialist wrappers for CPU and GPU exist, that include these common files.**

common.cpp:

```
__DECL FitTrack(int n) {
….
}
```

cpu_wrapper.cpp:

```
#define __DECL void
#include ``common.cpp``

void FitTracks() {
  for (int i = 0;i < nTr;i++) {
    FitTrack(n);
  }
}
```

cuda_wrapper.cpp:

```
#define __DECL __device void
#include ``common.cpp``

__global void FitTracksGPU() {
  FitTrack(threadIdx.x);
}

void FitTracks() {
  FitTracksGPU<<<nTr>>>();
}
```

→ **Same source code for CPU and GPU version**

- The macros are used for API-specific keywords only.
- The fraction of common source code is above 90%.

13.2.2018 David Rohr, drohr@cern.ch

David Rohr: “GPUs in ALICE in Run 3”

Details on GPU models

- **We have been working with both consumer and professional GPUs.**
  - The driving factor is price / performance (both for the current High Level Trigger and for the upcoming O2 farm).
  - We do not use many special GPU features, and with software optimized for CPUs and GPUs, the GPU speedup for us is usually x4 to x8 v.s. a desktop CPU and x2 to x3 v.s. a many-core server GPU.
  - For instance: In the ALICE HLT, one GPU yields roughly as much tracking performance as a full compute node (dual socket).
    - I.e., by using GPUs, we cut the size of the farm by half.
    - This makes sense if a GPU is significantly cheaper than a server. Not all pro-GPUs are, but we found a suited model for Run 2.
- **We do (currently) not use professional-only features like GPU-Direct or full duplex DMA, and we do not need double.**
  - Technically, we have no advantage from professional-grade GPU.
    - We might make use of the larger memory for O2 (for Run 2 it doesn’t matter), but consumer cards should work as well.
    - The professional support is an aspect.
  - We had one issue with one batch of GTX580 gamer cards, having an issue after in average several months of operation.
    - No problem for games, but in the cluster, this yields several failures per day.
    - In our case, we replaced these cards by different ones that worked, which was still cheaper than the professional ones.
  - It is not clear to what scale one can efficiently use the gamer cards.
  - Cooling of gamer cards (having fans) is an issue for HPC, but at our tracking load, the GPU remain cool even in 2U servers.
  - If one needs the professional features, there is not way but using professional grade cards.
- **Recently, NVIDIA changed the EULA prohibiting consumer cards for data center usage.**
  - Academic usage is still allowed, but for how long? How do we deal with such an uncertainty?

David Rohr: “GPUs in ALICE in Run 3”

GPU usage on the GRID for Run 3

- **How could ALICE use GPUs in the GRID for Run 3?**
  - Which GPUs would ALICE support?
  - What requirements does ALICE have?
  - Do we need / use special GPU queues?
- **Requirements:**
  - The only hard requirement is a large GPU memory for raw data reconstruction.
  - For MC reconstruction, every GPU currently available is sufficient.
- **GPU models:**
  - For good performance:
    - Ideally, a similar model as in the O2 farm, because we collet most experience there.
    - For the development, we anyhow have a look at the new NVIDIA / AMD models, and will support most of them eventually at good performance.
  - Any GPU will usually be better than no GPU
    - If a GPU is available (NVIDIA, AMD, OpenCL 2.2), we will support it, and therefore it would make sense to use it.
    - We might require a validation run for untested models, or exclude uncommon models, to avoid unforeseen issues and limit the number of platforms.
- **GPU queues:**
  - Technically, we do not need a special queue. The software can auto-detect the GPUs and use them transparently, and fall back to the CPU in case no GPU is present or if there is a problem setting up the GPU.
  - However, special queues would enable us to use the available resources more efficiently by placing compute jobs where they run best:
    - Raw reconstruction on GPU nodes with large memory.
    - MC reconstruction on any GPU node.
    - Other jobs (MC generation, analysis) on pure CPU nodes.

Andrew McNab: “GPUs in LHCb DIRAC”

GPUs in LHCb

- GPUs are increasingly being used by LHCb users
  - eg to accelerate highest level fitting at the ntuple level
  - using custom code or packages like GooFit
- A lot of interest in using GPUs in online for Run3
  - “Online” in LHCb isn’t just the HLT, as we aim to have the final, calibrated, reconstructed data coming out of the farm
- So in the distributed computing team we identify a need for integrating GPUs into the “the grid”
  - Always using DIRAC in our case
- What I talk about here goes into DIRAC so is available to other DIRAC installations (ILC, Belle II, GridPP, EGI, …)

Andrew McNab: “GPUs in LHCb DIRAC”

Matching strategy

- For **pilot submission** to CEs we went for GPU queues for now
  - We don’t use CREAM/ARC JDL requirements for instance
- For **payload matching**, we use DIRAC Tags mechanism
- Jobs are tagged with a generic GPU tag in the DIRAC JDL
- Queues are tagged with a RequiredTag option in the DIRAC CS
  - Only payload jobs with the GPU tag will match pilots in these queues
  - So GPU queues are not filled with random jobs!
- We can also add this Tag at the pilot level (see later)

Kenyi Hurtado + Edgar Fajardo: “GPUs on OSG for CMS”

Current status of GPU in the GRID of OSG

- Four sites offering GPU’s: Nebraska, UCSD, Syracuse, Vanderbilt (CMS only).
- All behind normal HTCondor-CE
- OSG accounting (Gracc) has been updated and the probes to account GPU time
- Future versions of condor would include reporting of GPU Wall Time as they do with CPU.

10

Kenyi Hurtado + Edgar Fajardo: “GPUs on OSG for CMS”

Once a pilot lands on a worker node

- All jobs are run inside a wrapper script (behind users back) that checks for the “+SingularityImage” jobAd and runs the job inside that image
- For this to work the wrapper script for GPU always binds to the drivers: “singularity --bind /usr/lib64/nvidia:/host-libs”
- This causes troubles when sites do not have the drivers in the standard location “/usr/lib64/nvidia”
- It will be fixed once all sites move to Singularity 2.4.2 and "-- nv" can be used.

8

Kenyi Hurtado + Edgar Fajardo: “GPUs on OSG for CMS”

Once a pilot lands on a worker node

- Runs condor_gpu_discovery
- Updates the StartdAd CUDACapability, CUDADeviceName depending on what is found in the worker node, and the CudaDeviceNumber(s) assigned to the pilot.
- Number of GPU’s per pilot is determined by the site and configured at the factory level. Most sites prefer one GPU per pilot to increase turnaround.

9

Kenyi Hurtado + Edgar Fajardo: “GPUs on OSG for CMS”

Submitting TensorFlow jobs

- GPU:
  - OSG-supported singularity container with GPU components

- GPU resources can be requested by simply using the "request_gpus" classads in the submit file.
- Some classads like:
  - CUDACapability (e.g: >=3), CUDADeviceName (e.g: Tesla K20m)

  can be used for matching.

htcondor submit file

```
Universe = vanilla

#... (request resources, define logfiles, etc)
Executable = tf_matmul_wrapper_cvmfs.sh
transfer_input_files = tf_matmul.py
request_gpus = 1
Requirements = HAS_SINGULARITY == True
&& CUDACapability >= 3
+SingularityImage =
"/cvmfs/singularity.opensciencegrid.org/opensciencegrid/tensorflow-gpu:latest"
```

23

Themes / issues

- Gamer GPUs vs Enterprise GPUs
  - Nvidia attempting to use CUDA licensing to stop people like us from using much cheaper gamer GPUs
- ALICE portable binaries work with any GPU or none
- How should experiments channel GPU jobs to GPU machines
  - JDL requirements? queues? Discovery by pilots?
- No physics difference between GPU and non-GPU processing?
  - Ongoing validation of new GPU resources to check this
- Everyone was reserving one GPU per job
  - This also makes accounting straightforward, using wallclock time and some GPU benchmark

Next steps

- Keep talking between experiments and sites!
- Updates at future GDBs or another preGDB
- Some projects (eg OSG/CMS use of HTCondorCE) are creating de facto interfaces
  - It would be good for other projects to adopt the same ones
  - eg the dictionary of what names to use in ClassAd requirements
- Co-ordination/communication between application activities also relevant to WLCG advice to sites
  - eg Can everyone use gamer GPUs? Do we all have similar GPU memory requirements?