

Grid GPUs update

A. Forti

WFMS

21 February 2019



Test images

- Basic tests
 - Most basic has no output and only checks tensor flow can see the gpus, and data and code are correctly loaded. No input.
 - Second test runs a more complex ML code. Input in rucio.
- Useful to make sure nodes are correctly setup.
 - Used also on Titan with some modifications

gpu-basic-test

Basic test using an tensorflow image with python3. It checks that a GPUs is available and runs a minimal code on it.

Can be used when setting up nodes or Panda Resources.

To use it with singularity on the GPU WN

```
singularity -s exec --nv \  
docker://gitlab-registry.cern.ch/hepimages/public/gpu-basic-test \  
python /test-gpu.py
```

gpu-atlasml-test

More complicated test. Requires also an input

Can be used when setting up nodes or Panda Resources.

To use it with singularity on the GPU WN

```
singularity exec --nv --pwd /data -B <trainingfile_location>:/data \  
docker://gitlab-registry.cern.ch/hepimages/public/gpu-atlasml-test \  
python /btagging/DL1_c_vs_b_slim.py trainingfile.h5 10 gpu 50000
```

trainingfile.h5 file can be downloaded from rucio

```
atlasSetup  
lsetup rucio  
rucio get user.aforti:gpu.basic.training.h5
```

<https://gitlab.cern.ch/hepimages/public>



Queues status

- 4 queues
 - Manchester (10 GPUs): working since November
 - ARC-CE/HTCondor, 1 GPU per VM, no MPI
 - QMUL(6 GPUs): working since January
 - CREAM/SLURM, 2 GPUs per node, limited to 1 per job with cgroups, no MPI
 - OSCER: being debugged
 - HTCondor/SLURM, 80 GPU multiple GPUs per node, MPI enabled, oversubscribed
 - BNL: being setup
 - HTCondor/SLURM, MPI, no network access
- Other sites that might have GPUs on the grid in the future
 - MWT2, RAL-LCG2, UKI-SOUTHGRID-RALPP, Waterloo



APF/Harvester

- BNL needs to use local edge harvester
- Currently using APF dev for the first 3 queues
 - We need to move to harvester also for the other 3
 - More permanent system and would have same system for all the queues
 - What do we need to do to setup a harvester dev with pilot2 for this?



Current Brokering

- Setting up separate queues to submit with prun
- Targeted queues with `--site` until only Manchester worked
 - Now we can select multiple sites using `--cmtConfig nvidia-gpu`
 - `--cmtConfig` should be renamed when we have pcontainer
- Queue were set in brokeroff to avoid standard jobs to set them online
 - We removed all the standard releases tags from the queues
 - We only left our dummy gpu ones
 - We added a catchall = gpu flag to broker only jobs that have the architecture set
 - This should become something more complex in the future
`gpucount = #`

```
prun --exec "python /btagging/DL1_c_vs_b_slim.py trainingfile.h5 10 cpu 50000" --containerImage docker://gitlab-registry.cern.ch/hepimages/public/gpu-atlasml-test --noBuild --forceStaged --outDS user.aforti.gpu.training.h5.test.1.20190221080432 --inDS user.aforti:gpu.basic.training.h5 --cmtConfig nvidia-gpu
```



Brokering (2)

- Current solution is for proof of concept and can last until we have a better defined brokering plan.
 - Doesn't solve users requests and doesn't solve how to pass parameters to JDL.
 - SLURM for example requires GPUNumber=#
 - Currently set manually
- Work with prun hopefully prelude to a more common approach.
 - Started a google doc to put together what we know so far
 - GPU brokering google doc

