Containers & GPUs

A. Forti

GridPP technical meeting

2 August 2019

ATLAS work

- We have two lines of work
  - Make more GPU accessible to users on the grid
    - Parallelized code is becoming more spread
    - Users write in CPUs because they don't have accessible GPUs
    - First PoC, talk at the S&C week in December,
  - Using GPUs at HPC sites
    - HPC sites in the future will have large fractions on their computing power on GPUs
    - Access not meant to be for single small users more for power users

GPUs on the grid

- Deployment of GPUs on the grid depends
  - Exposing GPU
  - Singularity installed
  - Pilot2
  - Panda Queue pointing to GPU resources being enabled for containers
    - Some panda queues may require an extra parameter
- Simplest way
  - Treat the GPUs like CPUs
  - 1 GPU per job
    - Each GPUs has a set of node resources
    - Users have no choice at submission time

Grid GPU resources

Queue	Resource type	CE/BS	N GPUs	GPU Models	GPU memory	Host Memory	Status	Comments
ANALY_BNL_GPU_TEST	HPC	HTCondor-CE/slurm	12	P100	16GB	21.3GB		will require special treatment from harvester; also shared with Jupyter users who have priority
ANALY_OU_OSCER_GPU_TEST	HPC/grid	HTCondor-CE/slurm	80	K40	12GB	12GB		production queue has oversubscription problems and atlas has low priority
ANALY_QMUL_GPU_TEST	grid	CREAM-CE/slurm	2,2	K40,K80	12GB	12GB	Working	GPUNumber=x for now is hardcoded in the dev APF JDL,number of GPUs per job limited by cgroups, K80=2K40, so total of 6 gpu slots avalable.
ANALY_MANC_GPU_TEST	grid	ARC-CE/HTCondor	6, 4	VT100, K40	12GB	12 GB	Working	single queue, no submission parameters, 1 GPU per job
ANALY_MWT2_GPU	grid	HTCondor-CE/HTCondor	8	2080Ti	24GB	24 GB	Working	single queue, no submission parameters, 1 GPU per job
ANALY_INFN-T1_GPU	grid	HTCondor-CE/HTCondor	2	K40	12GB	12GB	testing	single queue, no submission parameters, 1 GPU per job

- Different GPU models but all nvidia
- Different CE/batch systems:
  - HTC-CE, ARC, CREAM-CE/HTC, slurm
- MWT2 and INFN-T1 on harvester/pilot2
- Manchester, QMUL still on APF-dev/pilot2
  - QMUL needs an extra param in the JDL
    - How can we move QMUL to harvester?

Sites setups

- Manchester ARC-CE/HTCondor
  - Added a HasGPUs custom ClassAd to connect the queue with the GPU nodes
- MWT2 HTCondor-CE/HTCondor
  - Also added a HasGPUs classad (no comunication there)
- QMUL CREAM-CE/slurm
  - See Dan's talk
  - Need an extra parameter GPUnumber in the JDL

Brokering

- Simple brokering changes
  - Remove all the tags from the queue to avoid standard jobs being brokered there
  - Add a generic nvidia-gpu architecture that can be selected by the user
  - Add a gpu flag to catchall to require that the job explicitly sets an architecture

computingsite (3)	ANALY_MANC_GPU_TEST (43)	ANALY_MWT2_GPU (1)	ANALY_QMUL_GPU_TEST (41)

MWT2 still had tags, GPU jobs weren't brokered because the queue was full of other jobs

Brokering (2)

- Brokering rules above doesn't satisfy more complicated requirements
  - pcontainer --resource nvidia-gpu=3 --resource cpu=1.5 --resource mem=300
    - We might not need this level of complexity at grid sites
- For some sites we still need to add extra parameters to the JDL whether the user requests it or not
  - Is AGIS the right place? What if we actually implement the command above?
- Started a document for GPU brokering a while ago
  - Solutions need to be integrated with more general brokering rules particularly if using architecture

Test images

- Sites setup running basic test images that test
  - GPU exist
  - CUDA libraries installed
  - Singularity configured
    - nv
    - underlay
  - Data and functions are loaded on the GPU
- Used at all sites
  - Also BNL and Titan gave it a go
  - HC tests in the future

Public images

gpu-basic-test

Basic test using an tensorflow image with python3. It checks that a GPUs is avaialble 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
```

- Will add also first HP scan to the list if possible

Command line

```
prun --containerImage \
docker://lukasheinrich/dltests:48cef2-gpu \
--exec "python /btagging/train.py --configs %IN \
--validationfile /data/DL1_files_hp_test \
--trainingfile /data/DL1_files_hp_training \
--variables /data/DL1_files_hp_variables \
--validation_config /data/DL1_files_hp_validconf \
--outputfile out.json" \
--inDS user.mguth:user.mguth.dl1.hp.optimisation.configs40 \
--secondaryDS IN2#4#user.mguth.dl1.hp.optimisation.files \
--outDS user.${RUCIO_ACCOUNT}.hp.test.$(date +%Y%m%d%H%M%S) \
--forceStaged \
--noBuild \
--forceStagedSecondary \
--reusableSecondary=IN2 \
--outputs out.json \
--nFilesPerJob 2 \
--disableAutoRetry \
--cmtConfig nvidia-gpu \
--tmpDir /tmp \
--respectSplitRule
```

- Complications mostly due to handling of input data when splitting jobs

Things to implement

- Runcontainer still doesn't do directIO
  - Less important for GPUs
    - Input are not root files (yet)
- In parallel we need to integrate the user proxy
  - ADCINFR-114
  - Need some answers from Fernando and Tadashi
    - Active thread
- pcontainer needs more options
  - Command line above fails

```
aforti@vm26>source run-hpscan
usage: pcontainer [options]
  HowTo is available at https://twiki.cern.ch/twiki/bin/view/PanDA/PandaContainer
pcontainer: error: unrecognized arguments: --inDS user.mguth:user.mguth.dl1.hp.optimisation.configs40 --secondaryDS I
N2#4#user.mguth.dl1.hp.optimisation.files --forceStaged --forceStagedSecondary --reusableSecondary=IN2 --nFilesPerJob
 2 --disableAutoRetry --tmpDir /tmp
```

Use cases

- So far we have run a couple of HP scans
  - B-tagging HP scan we run in December is becoming a sort of testing suite for brokering
  - Second HP scan took too long to run and half of the jobs failed
    - Highlighted queue length and efficiency of the jobs
    - Also input file replication
- Other possible payloads
  - ML workload on Jet/EtMiss ML paper
  - GAN workload (also to run on GPU)
  - Attila's reconstruction+GPU tests
  - others....
- System not stable enough to ask users but we do need some dedicated people willing to help

SKA and Dirac

- SKA makes heavy use of ML too
  - And uses singularity containers to run payloads
- We tried to run some of their workloads on the GPUs but Dirac needs development work on scheduling different type of pilots.
  - Sending all jobs to all queues was solved with tags
  - GPU queues get enough pilots and then Dirac doesn't run other workloads because it thinks there are enough pilots but not the right jobs for those queues
- Icecube has used the GPUs for years with glide-in method
  - Similar to panda I think with specific associated queues

GPU efficiency

- Currently GPU models we run are not really efficient
  - Depending on GPU model 5%-25% occupancy
- Two reasons
  - Users don't have access to GPUs cannot improve their code
  - Models are too small, need bigger models with more parameters
- Still this is a known problem from Vakho's talk about the experience on the trigger

- Hard to saturate GPU: more than 55 clients to saturate a single NVIDIA GTX1080 (Pascal)
  - Adding more clients leads to a bottleneck with network transfer and the APE server capabilities

Heterogeneous Athena 3/5/2019

- How can we solve this? Is there anyone looking at this problem?

Containers & software optimization

- All this was done using containers
- Containers solve the main problem of software distribution
  - Users that couldn't run before because the infrastructure didn't support the softare can do so now
- There is a fair question about optimizing software according to the architecture
  - But how much do we want to push it? If this requires deep knowledge of the site architecture the site should provide a base image that the users can build on
  - Do we have the exprts at each HPC site to do it?

Benchmarking

- WLCG/HSF will start a specific effort on benchmarking, costs and optimization
- WLCG benchmarking WGs will also look into it
  - Asked already for the tests workloads
    - ATLAS tests quick but not large enough IMO
- ATLAS ML group is starting to look at monitoring by dumping info directly from the jobs in a JSON
- ….

Is it worth it?

- Neutrino experiments use GPUs all the time
  - That's why some sites already have them
  - Icecube big user with glide-ins
- ATLAS initial effort being progressively increased
  - Now also have a GPU dedicated software team as well as a growing ML effort
- SKA not organised yet, but do have workloads to try as well
  - May help pinpointing work to do with dirac for special queues
- If you have GPUs worth putting them online
  - in a simple setup, i.e. without complicated brokering