GPGPU support in CREAM-CE update

Lisa.Zangrando (at pd.infn.it)
Istituto Nazionale di Fisica Nucleare (INFN)
Italy

EGI-Engage is co-funded by the Horizon 2020 Framework Programme of the European Union under grant number 654142

Introduction

- Work as part of JRA2.4 task (INFN,CIRMMP, IISAS)
- Progresses tracked in https://wiki.egi.eu/wiki/GPGPU-CREAM
- CREAM developers team of INFN is involved
- Requirements collected at http://bit.ly/Lisbon-GPU-Session
- Work plan (the task ends in May 2016)
  - Indentifying the relevant GPGPU-related parameters supported by the different LRMS, and abstract them to significant JDL attributes
  - Implementing the needed changes in CREAM-core and BLAH components
  - Writing the info-providers according to GLUE 2.1
  - Testing and certification of the prototype
  - Releasing a CREAM-CE update with full GPGPU support (at least for Torque)

Work done so far/1

- Started from previous analysis and work of EGI Virtual Team (2012) and GPGPU Working Group (2013-2014)
- CIRMMP/MoBrain use-case and test-bed
  - AMBER and GROMACS applications with CUDA 5.5
  - 3 nodes (2x Intel Xeon E5-2620v2) with 2 NVIDIA Tesla K20m GPUs per node at CIRMMP
  - Torque 4.2.10 (source compiled with NVML libs) + Maui 3.3.1 (patched)
  - EMI3 CREAM-CE
- Tested local AMBER job submission with the different Torque/NVIDIA GPGPU support options, e.g.:

```
$ qsub -l nodes=1:gpus=2:default
$ qsub -l nodes=1:gpus=2:exclusive_process
```

Work done so far/2

- GPUMode can have the following values for Torque:
  - default: shared mode available for multiple processes
  - exclusive thread: Only one COMPUTE thread is allowed to run on the GPU
  - prohibited: no COMPUTE contexts are allowed to run on the GPU
  - exclusive_process: only one COMPUTE process is allowed to run on the GPU

New JDL attributes "GPUNumber" and "GPUMode“ defined and implemented in CREAM and BLAH parser

Work done so far/3

- First prototype deployed at CIRMMP and remote submission tested:

```
$ glite-ce-job-submit -o jobid.txt -d -a -r
cegpu.cerm.unifi.it:8443/cream-pbs-batch test.jdl
$ cat test.jdl
[
   executable = "test_gpu.sh";
   inputSandbox = { "test_gpu.sh" };
   stdoutput = "out.out";
   outputsandboxbasedesturi = "gsiftp://localhost";
   stderror = "err.err";
   outputsandbox = { "out.out","err.err","min.out","heat.out" };
   GPUNumber=2;
   GPUMode="exclusive_process";
]
```

Work done so far/4

- GLUE2.1 draft analysed as a base for writing GPU-aware Torque infoprovider
  - **ExecutionEnvironment**
    - is usually defined statically by YAIM during the deployment/configuration of the service
    - can be obtained from *pbsnodes* command:

```
GLUE2ExecutionEnvironmentPhysicalGPUs
GLUE2ExecutionEnvironmentLogicalGPUs
GLUE2ExecutionEnvironmentGPUVendor
GLUE2ExecutionEnvironmentGPUModel
GLUE2ExecutionEnvironmentGPUVersion
GLUE2ExecutionEnvironmentGPUClockSpeed
```

```
Gpus = 2

gpu_status =
gpu[1]=gpu_id=0000:42:00.0;gpu_pci_device_id=271061214;gpu_pci_location_id=0000:42:00.0;gpu_product_name=Tesla K20m;gpu_display=Enabled;gpu_memory_total=5119 MB;gpu_memory_used=11 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_utilization=0%;gpu_ecc_mode=Disabled;gpu_temperature=18 C,gpu[0]=gpu_id=0000:04:00.0;gpu_pci_device_id=271061214;gpu_pci_location_id=0000:04:00.0;gpu_product_name=Tesla K20m;gpu_display=Enabled;gpu_memory_total=5119 MB;gpu_memory_used=13 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_utilization=0%;gpu_ecc_mode=Disabled;gpu_temperature=16 C,driver_ver=319.37,timestamp=Thu Sep 17 10:18:07 2015
```

Work done so far/5

– **ComputeManager**

- In the current draft GPU-related attributes declared only for cloud:

  `GLUE2CloudComputeManagerTotalGPU`

  `GLUE2CloudComputeManagerGPUVirtualizationType`

- GPUs must be considered also in the context of a classical Computing Service, e.g. the attributes below are missing:

  `GLUE2ComputeManagerTotalGPU`

  `GLUE2ComputeManagerGPUVirtualizationType`

Discussion thread at https://issues.infn.it/jira/browse/CREAM-177

Next steps

- Verifying if parameters identified for Torque have their analogous in other LRMSes or additional parameters have to be abstracted to JDL attributes
- But which LRMSes to support?
- RPs contacted so far and potentially interested:
  - STFC/Emerald (LSF8 based GPU cluster)
  - INFN-CNAF (LSF9 based GPU cluster)
  - IFCA (SGE based GPU cluster)
- We are looking for NGI/RP providing SLURM and Condor based GPU clusters

Next steps

- Interested RPs can join the testbed providing us:
  - one or more testing application examples that we can execute on their GPU cluster and the commands/options used for their submission
  - remote access to a server that can submit the above application sample jobs to their GPU cluster
  - hopefully (not mandatory), admin rights to allow us to deploy and test the CREAM prototype directly on top of their GPU cluster LRMS (as was done at CIRMMP)

Next steps

- Some final considerations:
  - INFN CREAM team produces infoproviders for Torque and SLURM only
    - for LSF rely on CERN and for SGE rely on CESGA
    - no infoproviders for Condor are available
  - Accounting sensors, mainly relying on LRMS logs, were in the past developed by the APEL team (only for SLURM there was direct involvement of the INFN team)
    - APEL team must be involved for having GPGPU accounting
  - Cloud integration activities carried out by IISAS partner, more info in https://wiki.egi.eu/wiki/GPGPU-FedCloud

Thank you for your attention.

Questions?

EGI-Engage is co-funded by the Horizon 2020 Framework Programme
of the European Union under grant number 654142