GPU user guide

How to Run a Job with a GPU

Let's run the gravitational N-body simulation found on the CUDA toolkit samples on a GPU. This example is suited for a standard *INCD* user elegible to use the ***hpc*** and ***gpu*** partitions.

> The ***fct*** partition and included resources is meant for users with a *FCT* grant and although the request of GPUs is made on the same way, they have specific instructions to follow found at FCT Calls

> The GPU's are only available at CIRRUS-A infrastruture on Lisbon.

Login on the user interface cirrus.ncg.ingrid.pt

```
$ ssh -l user cirrus.ncg.ingrid.pt
[user@cirrus01 ~]$ _
```

Prepare your working directory

Prepare your environment on a specific directory in order to protect from inter job interferences and create a submission batch script: *** only works for Cuda 10.2

```
[user@cirrus01 ~]$ mkdir myworkdir
[user@cirrus01 ~]$ cd myworkdir
[user@cirrus01 ~]$ cat nbody.sh
#!/bin/bash

#SBATCH --partition=gpu
#SBATCH --gres=gpu
#SBATCH --mem=8192MB
```

```
COMMON=/usr/local/cuda/samples/common
SAMPLE=/usr/local/cuda/samples/5_Simulations/nbody

[ -d ../common ] || cp -r $COMMON ..
[ -d nbody     ] || cp -r $SAMPLE .

module load cuda
cd nbody
make clean
make

if [ -e nbody ]; then
 chmod u+x nbody
 ./nbody -benchmark -numbodies=2560000
fi
```

In this example we copy the n-body CUDA toolkit sample simulation to the working directory, load cuda environment, build the simulation and run it.

Requesting the partition

Standard *INCD* users at *CIRRUS-A* have access to the ***gpu*** partition providing NVIDIA Tesla-T4 GPUs. In order to access these GPUs request the ***gpu*** partition with directive:

```
#SBATCH --partition=gpu
```

The partition ***fct*** provide several types of NVIDIA: T4 and V100S (please check current resources available page). As a general rule and depending on the application, the types of GPUs available on the cluster are similar but the Tesla-V100S perform the same work in half the time when compared with the Tesla-T4. Nevertheless, if you request a Tesla-V100S you may have to wait for resource availability until you have a free Tesla-T4 ready to go.

If you only want a free GPU allocated for your job then the **#SBATCH --grep=gpu*** form would be the best choice.

Requesting the Tesla-T4 GPU

We request the allocation of one GPU NVIDIA Tesla-T4 throught the option:

```
#SBATCH --gres=gpu:t4
```

Standard *INCD* users can access only NVIDIA Tesla-T4 GPUs, so we can simplify the request:

```
#SBATCH --gres=gpu
```

this way we ask for a GPU of any type, the same is valid on partitions with more than one type of GPU if we do not care about the type of allocated GPU to our job.

Requesting memory

Ensure enough memory for your simulation, follow the tips on *Determining Memory Requirements(page_to_be)* page.

On our example 8GB is sufficient to run the simulation:

```
#SBATCH --mem=8192M
```

Submit the simulation

```
[user@cirrus01 ~]$ sbatch nbody.sh
Submitted batch job 1176
```

Monitor your job

You can use the ***squeue*** command line tool

```
[user@cirrus01 ~]$ gqueue
JOBID PARTITION NAME     USER     ST STATIME       NODES CPUS TRES_PER_NODE NODELIST
1176  gpu       nbody.sh  user5    R  RUN0-00:02:33 1     1    gpu:t4       hpc058
```

or use the command ***sacct***, the job is completed when the *State* field mark is *COMPLETED*.

```
[user@cirrus01 ~]$ gacct
   JobID    JobName  Partition    Account  AllocCPUS      ReqGRES    AllocGRES      State ExitCode
------------ ---------- ---------- ---------- ---------- ------------ ------------ ---------- --------
1170      nbody.sh       fct       hpc         2  gpu:v100s:1       gpu:1  COMPLETED      0:0
1171      nbody.sh       fct       hpc         2     gpu:t4:1       gpu:1  COMPLETED      0:0
1175      teste.sh       fct       hpc         1                    COMPLETED      0:0
1176      nbody.sh       gpu       hpc         1       gpu:1       gpu:1  COMPLETED      0:0
```

> if the state is different from *COMPLETED* or *RUNNING* then check your simulation or request help throught the email address **helpdesk@incd.pt** providing the *JOBID*, the submission script, the relevant slurm output files, e.g. *slurm-1176.out* , or other remarks you think it may be helpfull

Check the results at job completion

```
[user@cirrus01 ~]$ ls -l
-rw-r-----+ 1 user hpc  268 Oct 22 13:56 gpu.sh
drwxr-x---+ 3 user hpc 4096 Oct 20 18:09 nbody
-rw-r-----+ 1 user hpc  611 Oct 22 13:41 slurm-1176.out



[user@cirrus01 ~]$ cat slurm-1176.out
...
> Windowed mode
> Simulation data stored in video memory
> Single precision floating point simulation
> 1 Devices used for simulation
GPU Device 0: "Turing" with compute capability 7.5

> Compute 7.5 CUDA device: [Tesla T4]
number of bodies = 2560000
2560000 bodies, total time for 10 iterations: 308586.156 ms
= 212.375 billion interactions per second
= 4247.501 single-precision GFLOP/s at 20 flops per interaction
```

Use QOS to run GPU jobs

- This page it's dedicate to users who want to run GPU's and have a QOS.

GPU JOB submission using QOS

- In this example we will use the atributed QOS=gpu097822021 to be used for GPU and submit a job for the V100 Nvidia.

```
#!/bin/bash
#SBATCH --job-name=prod01
#SBATCH --partition=gpu
#SBATCH --qos=gpu097822021
#SBATCH --gres=gpu:v100s
#SBATCH --output=%x.o%j
#SBATCH --error=%x.o%j

### Prepare the environment
module purge
module load gcc83/openmpi/4.1.1 cuda-11.2

echo hostname
```

Deep Learning Example

The INCD-Lisbon facility provide a few GPU, check the ***Comput Node Specs*** page.

Login on the submit node

Login on the cluster submition node, check the ***How to Access*** page for more information:

```
$ ssh -l <username> cirrus8.a.incd.pt
[username@cirrus01 ~]$ _
```

Alternatives to run the Deep Learning example

We have alternatives to run the *Deep Learning* example, or any other python based script:

1. prepare a user python virtual environment on home directory and launch a batch job;

The next three sections shows how to run the example for each method.

1) Run a Deep Learning job using a prepared CVMFS python virtual environment

Instead of preparing an user python virtual environment we can use the environment already available on the system, named **python/3.10.13**, check it with the command

```
[username@cirrus08 ~]$ module avail
---------------- /cvmfs/sw.el8/modules/hpc/main ------------------
...
intel/oneapi/2023    python/3.8          udocker/alphafold/2.3.2
julia/1.6.7          python/3.10.13 (D)
...
```

> We will find other **python** version, namely version **3.7** and **3.8**, this version do not contain the **tensorflo** module due to **python** version incompatibility.

We will change the submit script **dl.sh** to the following:

```
[username@cirrus08 dl]$ vi dl.sh
#!/bin/bash
#SBATCH -p gpu
#SBATCH --gres=gpu
#SBATCH --mem=64G

module load python/3.10.7
python run.py

[username@cirrus08 dl]$ ls -l
-rwxr-----+ 1 username usergroup   124 Feb 26 16:44 dl.sh
-rw-r-----+ 1 username usergroup  1417 Feb 26 16:46 run.py
```

Submit the Job

```
[username@cirrus08 dl]$ sbatch dl.sh
Submitted batch job 15135448
JOBID    PARTITION NAME       USER       ST TIME       NODES CPUS TRES_PER_NODE  NODELIST
15290034 gpu       dl.sh      jpina      PD 0:00       1     1    gres/gpu
```

Check Job results

On completion check results on standard output and error files:

```
[username@cirrus08 dl]$ ls -l
-rwxr-----+ 1 username usergroup   124 Feb 26 16:44 dl.sh
-rw-r-----+ 1 username usergroup  1417 Feb 26 16:46 run.py
-rw-r-----+ 1 username usergroup 18000 Feb 26 18:51 slurm-15135448.out
```

and procceed as in the previous example.

How to selected a GPU

Select any GPU

- On this example we choose one GPU with at least 8192 MB memory.

```
#!/bin/bash

#SBATCH --partition=gpu
#SBATCH --gres=gpu
#SBATCH --mem=8192MB

COMMON=/usr/local/cuda/samples/common
SAMPLE=/usr/local/cuda/samples/5_Simulations/nbody

[ -d ../common ] || cp -r $COMMON ..
[ -d nbody     ] || cp -r $SAMPLE .

module load cuda
cd nbody
make clean
make

if [ -e nbody ]; then
chmod u+x nbody
./nbody -benchmark -numbodies=2560000
fi
```

Select a specific GPU: V100s

```
#!/bin/bash

#SBATCH --partition=gpu
#SBATCH --gres=gpu:v100s
```

```
COMMON=/usr/local/cuda/samples/common
SAMPLE=/usr/local/cuda/samples/5_Simulations/nbody

[ -d ../common ] || cp -r $COMMON ..
[ -d nbody     ] || cp -r $SAMPLE .

module load cuda
cd nbody
make clean
make

if [ -e nbody ]; then
 chmod u+x nbody
 ./nbody -benchmark -numbodies=2560000
fi
```

GPU list

You can find the full GPU list per cluster here