The OSU College of Engineering DGX System for Advanced GPU Computing

OSU’s College of Engineering has six Nvidia DGX-2 systems

Performance Comparison with one of our other Systems

How to SSH to the DGX Systems

How to Check on the DGX Systems

Submitting a Non-batch Test-CUDA job to the DGX System

Performance Comparison with one of our other Systems

How to SSH to the DGX Systems
Create a bash shell file that looks like this:

```
#!/bin/bash
#SBATCH  -J  MonteCarlo
#SBATCH  -A  cs475-575
#SBATCH  -p  classgputest
#SBATCH  --constraint=v100
#SBATCH  --gres=gpu:1
#SBATCH  -o  montecarlo.out
#SBATCH  -e  montecarlo.err
#SBATCH  --mail-type=BEGIN,END,FAIL
#SBATCH  --mail-user=joeparallel@oregonstate.edu
module load cuda/10.1
nvcc -o montecarlo montecarlo.cu
./montecarlo
```

Submit the job described in your bash file.

This is the partition name that we use for our class when debugging and testing.

Note: A single dash (-) is used for a single character flag. A double dash (--) is used for a word (more than a single character) flag.

These 3 lines are bash code:

1. Submit the job described in your bash file.
2. Check the output. (I like sending my output to standard error, not standard output).
3. Submit-c 143% sbatch submit.bash

---

**Submitting a Loop**

```
#!/bin/bash:
#SBATCH  -J  MonteCarlo
#SBATCH  -A  cs475-575
#SBATCH  -p  classgpufinal
#SBATCH  --constraint=v100
#SBATCH  --gres=gpu:1
#SBATCH  -o  montecarlo.out
#SBATCH  -e  montecarlo.err
#SBATCH  --mail-type=BEGIN,END,FAIL
#SBATCH  --mail-user=joeparallel@oregonstate.edu
module load cuda/10.1
for t in 2048 8192 131072 2097152
   do
      for b in 8 16 32 64 128 256
         do
            nvcc -DNUMTRIALS=$t -DBLOCKSIZE=$b -o montecarlo montecarlo.cu
            ./montecarlo
         done
   done
```

Submit-c 153% sbatch submitloop.bash
Submitted batch job 475

---

**Submitting a Batch Final-CUDA job to the DGX System**

```
#!/bin/bash
#SBATCH  -J  MonteCarlo
#SBATCH  -A  cs475-575
#SBATCH  -p  classgputest
#SBATCH  --constraint=v100
#SBATCH  --gres=gpu:1
#SBATCH  -o  montecarlo.out
#SBATCH  -e  montecarlo.err
#SBATCH  --mail-type=BEGIN,END,FAIL
#SBATCH  --mail-user=joeparallel@oregonstate.edu
module load cuda/10.1
```

Submit the job described in your bash file.

This is the partition name that we use for our class when debugging and testing.

Note: A single dash (-) is used for a single character flag. A double dash (--) is used for a word (more than a single character) flag.

These 3 lines are bash code:

1. Submit the job described in your bash file.
2. Check the output. (I like sending my output to standard error, not standard output).
3. Submit-c 144% sbatch submit.bash

---

**Auto-Notifications via Email**

```
$SBATCH --mail-user=joeparallel@oregonstate.edu
```

You don't have to ask the system to email information to you, but if you do, please be sure you get your own email address right!

Our IT people are getting really tired of fielding the bounced emails when people misspell their own email address.

---

**What is the Difference Between the Partitions classgputest and classgpufinal?**

**classgputest** lets your program get into the system sooner, but it might be running alongside other jobs, so its performance might suffer. But you don't care because you are just debugging and testing, not taking performance numbers for your report.

**classgpufinal** makes your program wait in line until it can get dedicated resources so that you get performance results that are much more representative of what the machine can do, and thus are worthy to be listed in your report.

---

**What Showed up in my Email (which I spelled correctly)**

```
From:  Slurm workload manager
Subject:  Slurm Job 475 Name=MonteCarlo End:  Run time 00:00:12, COMPLETED, ExitCode 0
```

---

**Use slurm's scancel if your Job Needs to Be Killed**

```
submit-c 154% sbatch submitloop.bash
Submitted batch job 475
```

---

**Displaying the latest output added to montecarlo.err**

```
submit-c 15% tail -f montecarlo.err
```

Keep doing it forever.

Control-c to get out of it.
Submitting an OpenCL job to the DGX System

submit.bash:

```
#!/bin/bash
#SBATCH  -J  PrintInfo
#SBATCH  -A  cs475-575
#SBATCH  -p  classgpufinal
#SBATCH  --constraint=v100
#SBATCH  --gres=gpu:1
#SBATCH  -o  printinfo.out
#SBATCH  -e  printinfo.err
#SBATCH  --mail-type=BEGIN,END,FAIL
#SBATCH  --mail-user=joeparallel@oregonstate.edu
module load cuda/10.1

g++ -o printinfo printinfo.cpp /usr/local/apps/cuda/cuda-10.1/lib64/libOpenCL.so.1.1 -lm -fopenmp
./printinfo
```

Here’s what printinfo got on one graphics card on the DGX System

Number of Platforms = 1
Platform #0:
Name    = 'NVIDIA CUDA'
Vendor  = 'NVIDIA Corporation'
Version = OpenCL 1.2 CUDA 11.2.153'
Profile = FULL_PROFILE
Number of Devices = 1
Device #0:
Type = 0x0004 = CL_DEVICE_TYPE_GPU
Device Vendor ID = 0x10de (NVIDIA)
Device Maximum Compute Units = 80
Device Maximum Work Item Dimensions = 3
Device Maximum Work Item Sizes = 1024 x 1024 x 64
Device Maximum Work Group Size = 1024
Device Maximum Clock Frequency = 1530 MHz

Device Extensions:
- cl_khr_global_int32_base_atomics
- cl_khr_global_int32_extended_atomics
- cl_khr_local_int32_base_atomics
- cl_khr_local_int32_extended_atomics
- cl_khr_fp64
- cl_khr_byte_addressable_store
- cl_khr_icd
- cl_khr_gl_sharing
- cl_nv_compiler_options
- cl_nv_device_attribute_query
- cl_nv_pragma_unroll
- cl_nv_copy_opts
- cl_nv_create_buffer

For comparison, rabbit’s graphics card has 15 Compute Units