The OSU College of Engineering DGX System for Advanced GPU Computing

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

Each DGX server:
- Has 16 Nvidia Tesla V100 GPUs
- Has 28TB of disk, all SSD
- Has two 24-core Intel Xeon 8168 Platinum 2.7GHz CPUs
- Has 1.5TB of DDR4-2666 System Memory
- Runs the CentOS 7 Linux operating system

Overall compute power:
- Each V100 NVidia Tesla card has 5,120 CUDA Cores and 640 Tensor Cores
- This gives each 16-V100 DGX server a total of 81,920 CUDA cores and 10,240 Tensor cores
- This gives the entire 6-DGX package a total of 491,520 CUDA Cores and 61,440 Tensor Cores
Performance Comparison with one of our previous Systems

**How to SSH to the DGX Systems**

BTW, you can also use the *rabbit* machine:

```
ssh rabbit.engr.oregonstate.edu
```

It is a good place to write your code and get it to compile. It is *not* a good place to do the final run of your code.

How to SSH to the DGX Systems

**Performance Comparison with one of our previous Systems**

**DGX2 vs. Rabbit for Monte Carlo Calculations**

- **DGX-2’s Tesla**
- **Rabbit’s Titan Black**

# of Monte Carlo Trials vs. Mega Monte Carlo Trials/Second

**How to SSH to the DGX Systems**

```
flip3 151% ssh submit-c.hpc.engr.oregonstate.edu
```

**submit-c 142% module load slurm**

Type this right away to set your path correctly
How to Check on the DGX Systems

**Submit a CUDA job to the DGX Systems using Slurm**

Create a bash shell file that looks like this:

```bash
#!/bin/bash
#SBATCH -J MatrixMult
#SBATCH -A cs475-575
#SBATCH -p class
#SBATCH --gres=gpu:1
#SBATCH -o matrixmul.out
#SBATCH -e matrixmul.err
#SBATCH --mail-type=BEGIN,END,F fail
#SBATCH --mail-user=joe.parallel@oregonstate.edu

/usr/local/apps/cuda/cuda-10.1/bin/nvcc -o matrixMul matrixMul.cu
./matrixMul
```

Submit the job described in your shell file:

```
submit-c 143% sbatch submit.bash
Submitted batch job 474
```

Check the output:

```
submit-c 144% cat matrixmul.err
```

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.
Auto-Notifications via Email

#SBATCH --mail-user=joeparallel@oregonstate.edu

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

Our IT people are getting real tired of fielding the bounced emails when people spell their own email address wrong.

<table>
<thead>
<tr>
<th>From</th>
<th>Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slurm workload manager</td>
<td>Slurm Job_id=3980 Name=MatrixMul Ended, Run time 00:00:12, COMPLETED, ExitCode 0</td>
</tr>
<tr>
<td>Slurm workload manager</td>
<td>Slurm Job_id=3980 Name=MatrixMul Began, Queued time 00:00:01</td>
</tr>
</tbody>
</table>
Submitting a Loop

submitloop.bash:

```bash
#!/bin/bash
#SBATCH -J MatrixMul
#SBATCH -A cs475-575
#SBATCH -p class
#SBATCH --gres=gpu:1
#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH --mail-user=joeparallel@oregonstate.edu

for t in 1 2 4 8 16 32
do
    /usr/local/apps/cuda/cuda-10.1/bin/nvcc -DNUMT=$t -o matrixMul matrixMul.cu
    ./matrixMul
done
```

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

submit-c 154%
tail -f matrixmul.err
Displays the latest output added to matrixmul.err. Keeps doing it forever. Control-c to get out of it.

Results for Multiplying two 1024x1024 Matrices:
Varying the CUDA Block Size

![GigaFlops during Matrix Multiplication](chart.png)

(Each CUDA block was actually $NUMT \times NUMT$ threads)
Use slurm's `scancel` if your Job Needs to Be Killed

```
submit-c 163% sbatch submitloop.bash
Submitted batch job 476
```

```
submit-c 164% scancel 476
```

---

Submitting an OpenCL job to the DGX System using Slurm

```
submit.bash:

#!/bin/bash
#SBATCH --job-name MatrixMult
#SBATCH -A cs475-575
#SBATCH -p class
#SBATCH --gres=gpu:1
#SBATCH -o printinfo.out
#SBATCH -e printinfo.err
#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH --mail-user=joeparallel@oregonstate.edu

g++ -o printinfo printinfo.cpp /usr/local/apps/cuda-10.1/lib64/libOpenCL.so.1.1
./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 reference, rabbit's graphics card has 15 Compute Units