OSU College of Engineering DGX System for Advanced GPU Computing

Mike Bailey
mjb@cs.oregonstate.edu

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License

The OSU 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 other 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

submit-c 143% 

squeue

PARTITION NAME ST TIME NODES NODELIST(REASON)

GPU up 7-00:00:00 1 compute-gpu

class up 1:00:00 2 compute-dgx2-

dgxs up 7-00:00:00 2 compute-dgxs-

dgx2 up 7-00:00:00 5 compute-dgx2-

gpu up 7-00:00:00 1 compute-gpu

How to Check on the DGX Systems

submit-c 144% 

sinfo

PARTITION AVAIL TIMELIMIT NODES STATE NODELIST

dgxs up 7-00:00:00 2 idle compute-dgxs-

dgx2 up 7-00:00:00 2 idle compute-dgx2-

gpu up 7-00:00:00 1 idle compute-gpu

Submitting a Batch CUDA job to the DGX System

submit-c 143% sbatch submit.bash

Submit the job described in your bash file

Submitting a Batch CUDA job to the DGX System

submit-c 144% cat montecarlo.err

Check the output (I like sending my output to standard error, not standard output)
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.

Auto-Notifications via Email

```bash
#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 Showed up in my Email (which I spelled correctly)

```bash
From: Submit workload manager
Subject: Slurm_job_id-15391 Name=NamHuuMai Ended. Run time 03:33:12, COMPLETED. ExitCode 0
Slurm workload manager
Slurm_job_id-15391 Name=NamHuuMai/batch. Quasar time 03:33:11
```

Submit a Loop

```bash
submitloop.bash:
#!/bin/bash
#SBATCH -J MonteCarlo
#SBATCH -A cs475-575
#SBATCH -p classgpufinal
#SBATCH --constraint=v100
#SBATCH --gres=gpu:1
#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH --mail-user=joeparallel@oregonstate.edu
```

```bash
for t in 2048 8192 131072 2097152
  do
    for b in 8 16 32 64 128 256
      do
        /usr/local/apps/cuda/11.7/bin/nvcc -DNUMTRIALS=$t -DBLOCKSIZE=$b -o montecarlo montecarlo.cu
        ./montecarlo
      done
  done
```

These lines are bash code

Use slurm’s `scancel` if your Job Needs to Be Killed

```bash
# sbatch -c 163% sbatch submitloop.bash
Submitted batch job 475
```

```bash
# sbatch -c 164% sbatch submitloop.bash
Submitted batch job 476
```

```bash
scancel 476
```

Submitting an OpenCL job to the DGX System

```bash
submit.bash:
#!/bin/bash
#SBATCH -J PrintInfo
#SBATCH -A cs475-575
#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
```

```bash
g++ -o printinfo printinfo.cpp /usr/local/apps/cuda/11.7/lib64/libOpenCL.so.1.1 -lm -fopenmp
./printinfo
```

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

```bash
submit-c 163% sbatch submitloop.bash
Submitted batch job 475
```

```bash
submit-c 164% sbatch submitloop.bash
Submitted batch job 476
```

```bash
scancel 476
```

Submitting an OpenCL job to the DGX System
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.