GPU Resources at OSC

GPU Computing Group

April 13, 2012

GPU Resources Available at OSC

- Glenn, IBM Cluster 1350
  - 72 CUDA-enabled graphics devices (Quadro FX 5800 GPUs)
- Oakley, HP Intel Xeon Cluster
  - 128 NVIDIA Tesla M2070 GPUs
  - Operational as of $\sim$ March 20, 2012
- See `http://www.osc.edu/supercomputing/hardware/` for details.

Glenn Cluster

- 36 GPU-capable nodes on Glenn, connected to 18 Quadro Plex S4's for a total of 72 CUDA-enabled graphics devices
- Each node has access to two Quadro FX 5800-level graphics cards
- Each Quadro Plex S4 contains 4 Quadro FX 5800 GPUs
- 240 cores per GPU
- 4GB Memory per card

Oakley Cluster

- From the OSC webpage:

  OSCs new $4.1 million HP-built, Intel Xeon processor-based supercomputer, dubbed the Oakley Cluster, will feature more cores (8,328) on half as many nodes (694) as the centers current flagship system, the IBM Opteron 1350 Glenn Cluster. The Oakley Cluster can achieve 88 teraflops, tech-speak for performing 88 trillion calculations per second, or, with acceleration from NVIDIA Tesla graphic processing units (GPUs), a total peak performance of 154 teraflops.

- 64 nodes configured with two NVIDIA Tesla M2070 GPUs

Using OSC GPU Computing – First Steps

- After logging into either Glenn or Oakley, issue the following from the command line:

  ```
  module load cuda
  ```

- This will be needed any time you login if you want to compile or run a GPU program from the command line.

- The current CUDA version is 4.1.28.

Using OSC GPU Computing – First Steps

- Changes to your Makefile:

```
CUDA_HOME=${CUDA_INSTALL_PATH}

CUDA_INC=-I${CUDA_HOME}/include

CUDA_LIB=-L$CUDA_HOME/lib64 -lcudart

CUDA_CC=nvcc
```

- Example:

```
INCLUDE_FLAGS += -I /usr/local/cuda-3.1/cuda/include

LDLIBS += -L /usr/local/cuda-3.1/cuda/lib64 -lcuda -lcudart
```

Using OSC GPU Computing – First Steps

- To compile at the command line:

  ```
  $(CUDA_CC) $(CUDA_FLAGS) $(CUDA_INC) -o [cuda.cu.obj] [cuda.cu]
  ```

  or just

  ```
  nvcc -arch sm_13 -c -o myprogram.o muyprogram.cu
  ```

  (same as in the stat department system)

Running programs in batch mode

- Example of a typical batch submission file:

```
#PBS -l walltime=01:00:00
#PBS -l nodes=1:ppn=8:gpu=2
#PBS -N compute_test
#PBS -j oe
#PBS -S /bin/csh
module load cuda
cp /nfs/13/osu4077/NNI_GPU/tvdist $TMPDIR
cd $TMPDIR
./tvdist > myout
cp myout /nfs/13/osu4077/NNI_GPU/
```

Submitting and checking programs

- Commands are the same as for CPU programs:
  - `qsub jobscript`
  - `qstat`
  - `qstat -u osu4077`
  - `qdel [pid]`
  - `showstart [pid]`

Some additional notes

- If you only want to use one GPU, do not use the `cudaSetDevice` command. The system will use one of the available devices.

- If you want to use both GPU devices at once (in a single program), then you should use `cudaSetDevice` to indicate what functions each device will perform.

- GPU usage is charged differently than CPU usage (and I don't really understand either) – so think carefully about RU available on your account.

- Be careful to free memory allocated on the GPU. It is possible to bring down a node.