

**Driver/Toolkit/Samples**



# Access to system



- `ssh login.rc.fas.harvard.edu`
- Username “cfest###” where ### is your three digit number.
- Password is “P@Ssword!###” where ### is your three digit number.
- Enter 6 digit code

# Access to system

- Login with previous instructions
- `module load hpc/cuda-5.0.35`
- `cp -r ~jbentz/computefest .`
- Use the modules environment system
  - `module load <ModuleName>` to load a module
  - `module list` to show which modules are loaded
- Use batch system. Every job must be submitted to the queue.
  - `sbatch`—submit a job to the queue
  - `squeue`—show all jobs in the queue
  - `mdel`—delete a job from the queue
- <https://rc.fas.harvard.edu/odyssey-quickstart-guide/>

# Software



- **GPU Driver**
- **CUDA toolkit**
  - Includes all the software necessary for developers to write applications
    - Compiler (nvcc), Libraries, Profiler, Documentation
- **SDK**
  - Not strictly required but a good idea for ensuring your system is running properly.
  - Many examples with code samples illustrating lots of the important programming constructs and techniques.
- [www.nvidia.com/getcuda](http://www.nvidia.com/getcuda)    Above software from NVIDIA is free



# Examine GPU h/w and driver

- `nvidia-smi`
  - -h for help
  - -q for long query of all GPUs
  - PCIe Bus ID
  - Driver Version
  - ECC state
  - Power State/Fans/Temps/Clocks
- `sbatch computefest/runit.nvidia-smi`
  - Open the resulting `*out` file

# nvidia-smi



```
j bentz@rclogin12:~/compute fest
Wed Jan 15 23:05:10 2014
+-----+
| NVIDIA-SMI 5.319.37   Driver Version: 319.37           |
+-----+-----+
| GPU  Name            Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
|   0   Tesla K20m        Off      | 0000:04:00.0     Off  |             0        |
| N/A   34C    P0      48W / 225W | 11MB / 4799MB | 0%        Default   |
+-----+-----+
|   1   Tesla K20m        Off      | 0000:42:00.0     Off  |             0        |
| N/A   33C    P0      44W / 225W | 11MB / 4799MB | 74%        Default   |
+-----+-----+

+-----+-----+
| Compute processes:                                     GPU Memory |
| GPU      PID  Process name                             Usage      |
+-----+-----+
| No running compute processes found                    |
+-----+-----+

~
~
~

"slurm-5107266.out" 20L, 1552C                                20,1                                All
```

# CUDA toolkit



- **Compiler (nvcc)**
- **Libraries**
  - BLAS, FFT, sparse, RNG, NPP, OpenCL
- **Profiler**
  - Visual or command-line profiling available.

# Samples (free download from [nvidia.com](http://nvidia.com))



- `~jbentz/sdk/5.0/bin/linux/release`
- **Sample programs to illustrate CUDA and OpenGL programming constructs and algorithms.**
- **Useful diagnostic tests to query the GPU and its performance**



# sbatch computefest/runit.bandwidth



```
jlbentz@rclogin12:~/computefest
[CUDA Bandwidth Test] - Starting...
Running on...

Device 0: Tesla K20m
Quick Mode

Host to Device Bandwidth, 1 Device(s)
PINNED Memory Transfers
  Transfer Size (Bytes)      Bandwidth (MB/s)
  33554432                  5724.6

Device to Host Bandwidth, 1 Device(s)
PINNED Memory Transfers
  Transfer Size (Bytes)      Bandwidth (MB/s)
  33554432                  6349.7

Device to Device Bandwidth, 1 Device(s)
PINNED Memory Transfers
  Transfer Size (Bytes)      Bandwidth (MB/s)
  33554432                  129314.2

~
~

"slurm-5107264.out" 21L, 471C                21,0-1                All
```

# sbatch computefest/runit.query



```
jlbentz@rclogin12:~/computefest
Device 0: "Tesla K20m"
  CUDA Driver Version / Runtime Version      5.5 / 5.0
  CUDA Capability Major/Minor version number: 3.5
  Total amount of global memory:              4800 MBytes (5032706048 bytes)
  (13) Multiprocessors x (192) CUDA Cores/MP: 2496 CUDA Cores
  GPU Clock rate:                            706 MHz (0.71 GHz)
  Memory Clock rate:                          2600 Mhz
  Memory Bus Width:                           320-bit
  L2 Cache Size:                             1310720 bytes
  Max Texture Dimension Size (x,y,z)          1D=(65536), 2D=(65536,65536), 3
D=(4096,4096,4096)
  Max Layered Texture Size (dim) x layers      1D=(16384) x 2048, 2D=(16384,16
384) x 2048
  Total amount of constant memory:             65536 bytes
  Total amount of shared memory per block:     49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                  32
  Maximum number of threads per multiprocessor: 2048
  Maximum number of threads per block:         1024
  Maximum sizes of each dimension of a block:  1024 x 1024 x 64
  Maximum sizes of each dimension of a grid:   2147483647 x 65535 x 65535
  Maximum memory pitch:                       2147483647 bytes
  Texture alignment:                          512 bytes

27,3 12%
```

# sbatch computefest/runit.matmul



```
jlbentz@rclogin12:~/computefest
[Matrix Multiply Using CUDA] - Starting...
GPU Device 0: "Tesla K20m" with compute capability 3.5

MatrixA(320,320), MatrixB(640,320)
Computing result using CUDA Kernel...
done
Performance= 241.25 GFlop/s, Time= 0.543 msec, Size= 131072000 Ops, WorkgroupSize= 1024 threads/block
Checking computed result for correctness: OK

Note: For peak performance, please refer to the matrixMulCUBLAS example.
[Matrix Multiply CUBLAS] - Starting...
GPU Device 0: "Tesla K20m" with compute capability 3.5

MatrixA(320,640), MatrixB(320,640), MatrixC(320,640)
Computing result using CUBLAS...done.
Performance= 1150.44 GFlop/s, Time= 0.114 msec, Size= 131072000 Ops
Computing result using host CPU...done.
Comparing CUBLAS Matrix Multiply with CPU results: OK
~
~
~
~
"slurm-5107265.out" 18L, 746C 2,1 All
```

# Recap



- **Driver**
  - `nvidia-smi` to query the GPU hardware and state
- **CUDA Toolkit**
  - Development tools for GPU programming
- **SDK/Samples**
  - Sample code as well as diagnostic tests