Basic Info

CUDA machine: ssh dawang@ghc44.ghc.andrew.cmu.edu (32-46)

Setting up the CUDA enviroment:

- The developer driver provides a set of interfaces for the operating system to talk to the GPU subsystem.
- The CUDA toolkit provides a compiler, a debugger, a performance profiler, and a set of optimized CUDA libraries.
- The CUDA SDK provides an infrastructure and examples to help users quickly get start on using the CUDA infrastructure.

Samples -> NVIDIA_CUDA-6.5_Samples

Run CUDA Visual Profiler

```
ssh -X dawang@ghc32.ghc.andrew.cmu.edu
computeprof &
```

Optimize matrix_mul.cu & cuda_kmeans.cu

Matmul

现在的代码只实现了2次幂矩阵大小的乘积，我们要做的是

1. 优化到 150GFLOPS
2. 支持各种矩阵大小

运行 matrix mul

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/cppunit/lib
./matrix_mul -i ../matrix_mul_02.dat
```

提交

```
git commit -a -m “description"
git push origin master
```

初始代码的问题在于，一个 block 有线程限制，最多只有1024个，那么对于 33*33的矩阵，线程数目就会超过限制，而我们这里只开了一个 grid，所以会有问题。

Kmeans

目前的程序会在test 3和4中 fail，弄清楚为什么(hint: compute delta kernel function)，要做的是

1. 更新代码跑通所有测试
2. 1.5x speedup

GHC server GPU status

```
           GPU1              GPU2
ghc25   Quadro NVS 295
ghc26   Quadro FX 580   GeForce GTX 480
ghc27   Quadro NVS 295  GeForce GTX 480
ghc28   Quadro NVS 295  GeForce GTX 670
ghc29   N/A N/A
ghc30   Quadro NVS 295  GeForce GTX 650
ghc31   Quadro NVS 295  GeForce GTX 480
ghc32   Quadro NVS 295  GeForce GTX 670
ghc33   Quadro NVS 295  GeForce GTX 670
ghc34   Quadro NVS 295  GeForce GTX 670
ghc35   Quadro NVS 295  GeForce GTX 670
ghc36   Quadro NVS 295  GeForce GTX 670
ghc37   Quadro NVS 295  GeForce GTX 670
ghc38   Quadro NVS 295  GeForce GTX 680
ghc39   Quadro NVS 295  GeForce GTX 670
ghc40   Quadro NVS 295  GeForce GTX 670
ghc41   GeForce GTX 780
ghc42   Quadro NVS 295  GeForce GTX 680
ghc43   Quadro NVS 295  GeForce GTX 670
ghc44        N/A             N/A
ghc45   Quadro NVS 295  GeForce GTX 480
ghc46   Quadro NVS 295  GeForce GTX 480

GeForce GTX 670 will be used for grading.
But, you can use any GPUs (GeForece series) for development.
```

Q&A

> Q1. Is CUDA core the smallest unit in GPU?

Is CUDA core the smallest unit in GPU? -> Yes

> Q2. There will be at most one thread running on a CUDA core, right?

There will be at most one thread running on a CUDA core, right? -> Yes, at most one thread at a time. (context switching is available, so we can assign more threads than number of cuda cores)

> Q3. For a streaming multiprocessor, can it have more than one block running on it?

For a streaming multiprocessor, can it have more than one block running on it? -> Multiple thread

blocks can be assigned to a streaming multiprocessor. Streaming multiprocessor runs a thread block at a time and switching to other thread blocks (context switching)