Notes on using CUDA at U Iowa, Computer Science

Compiling

To compile a CUDA file (extension `.cu`) you need to use the NVIDIA compiler `nvcc`: For example, to compile `file.cu` to create the executable `file`, use

```
nvcc -o file file.cu -lpthread
```

If you want to use a particular library, you should include that in the command line. For example, the Julia set example from *Cuda by Example* requires the OpenGL Utility Toolkit (`glut`) library:

```
nvcc -o file file.cu -lglut -lpthread
```

To use the mathematics library, add `-lm`. Other compile options are similar to the standard C/C++ compilation options under Unix.

Accessing the CS CUDA-capable computers

To do this you need a CS (Computer Science Department) account. Your user name will be your HawkID, but the password you have to given to you.

From Unix (or Unix-like) systems: From within a terminal, execute the following command.

```
ssh -l <username> l-lnx<machine-id>.divms.uiowa.edu
```

The `<machine-id>` is an integer in the range 100 to 105. After entering your password, you will have terminal access to the computer. The operating system is Linux.

Checking device capabilities

To see what capabilities your machine's CUDA device has once you have terminal access, you need to copy, compile and run the `deviceQuery` program:

```
mkdir cuda-examples; cd cuda-examples
cp -a /opt/cuda/sdk/C C
cp -a /opt/cuda/sdk/shared shared
```

```
cd C
rm -rf src/FunctionPointers src/SobelFilter src/Interval
make
bin/linux/release/deviceQuery
```

The line starting “CUDA Capability Major/Minor” in the output has the capability represented as a number with a decimal point, such as “1.3”. For details as to how to interpret this number, see Appendix G of the *CUDA Programming Guide*. The *CUDA Programming Guide* is available via `http://developer.nvidia.com/nvidia-gpu-computing-documentation`.