CUDA LABS – GETTING STARTED GUIDE

Hassan Shojania, January 2009
Revised Andreas Moshovos, September 2010, Feb. 2012

1. SSH to one of the machines from `ug51.eecg.toronto.edu` to `ug75.eecg.toronto.edu` range.

2. Define the necessary environment variables: `"source /cad1/CUDA/cuda.csh"`

   Add the above source command to your `.cshrc` file so it automatically takes effect at every login. The CUDA compilers and runtime need these variables defined to work properly.

   After this stage, you should see `CUDA_HOME` variable already defined when you run `"setenv"` command.

3. Install the SDK: `"sh /cad1/CUDA/InstallSDK.sh".` Accept the default settings. The script executes and finally gives `"* Installation Complete"` message. You should see `"NVIDIA_GPU_Computing_SDK"` directory created in your home directory. It takes around 250 MB of your disk space.

   There should be several subdirectories under “`NVIDIA_GPU_Computing_SDK”.` The CUDA examples are under `“NVIDIA_GPU_Computing_SDK/C”.`

4. Let’s first build a number of shared libraries.

   Change directory by executing `"cd NVIDIA_GPU_Computing_SDK/shared”`

   Compile: `“make”`

   This creates the “release” version of the libraries. Some macros default to nothing in this version. You can also build a version for debugging: “make dbg=1”. If you want to use the emulator you can build the emulated version: “make emu=1” or “make dbg=1 emu=1”. The same options apply to all makefiles found under the CUDA SDK.

   Better compile the debug version as well: “make dbg=1”

5. Change directory to `"NVIDIA_GPU_Computing_SDK/C/common"`.

   Compile the libraries: “make” . Ignore the warnings.

   Compile the debug version as well: “make dbg=1”. Ignore the warnings.

   Read what the library provides: “`less cutil_readme.txt`”.

6. Now let’s compile some of the examples which can be found under the “`NVIDIA_GPU_Computing_SDK/C/src`” directory each on its own subdirectory.

   Change directory to the bandwidth test example: “`cd ../src/bandwidthTest`”.

   Now you are under `NVIDIA_GPU_Computing_SDK/C/src/bandwidthTest”.`

   Compile the example: “make”

The executable is installed in `NVIDIA_GPU_Computing_SDK/C/bin/linux/release`

Compile the debug version as well: “make dbg=1”

Run the release version. You should see something like this:

```
[bandwidthTest] starting...
bandwidthTest Starting...

Running on...

 Device 0: GeForce GTX 480
 Quick Mode

 Host to Device Bandwidth, 1 Device(s), Paged memory
   Transfer Size (Bytes)        Bandwidth(MB/s)
   33554432                     2228.2

 Device to Host Bandwidth, 1 Device(s), Paged memory
   Transfer Size (Bytes)        Bandwidth(MB/s)
   33554432                     1564.1

 Device to Device Bandwidth, 1 Device(s)
   Transfer Size (Bytes)        Bandwidth(MB/s)
   33554432                     119485.0

[bandwidthTest] test results...
PASSED

Press ENTER to exit...
-----------------------------------------------------------
```

7. For creating your own new project, follow these steps:

   There is a “template” project that you can copy and modify:

   (a) Copy the template project:
   ```
   cd  ~/NVIDIA_GPU_Computing_SDK/C/src/template
   mkdir ../myproject
   cp -r * ../myproject
   ```

   (b) Edit the filenames of the project to suit your needs
   ```
   mv template.cu myproject.cu
   mv template_kernel.cu myproject_kernel.cu
   mv template_gold.cpp myproject_gold.cpp
   ```

   *** Alternatively, copy the files from the `deviceQuery` project. This example just uses a `.cu` file in which you can write C code as well.

   (c) Edit the Makefile and source files. Just search and replace all occurrences of "`template`" with "`myproject`". You’ll need to change the Makefile and file “myproject.cu” that includes the test kernel #include <template_kernel.cu>.

   (d) Build the project
   ```
   make
   ```

You can build a debug version with "`make dbg=1`", an emulation version with "`make emu=1`", and a debug emulation with "`make dbg=1 emu=1`". Similarly, you can build versions without debugging support using just “`make`”.

(e) Run the program

```
../../bin/linux32/release/myproject
```

(It should print "`Test PASSED`")

(f) Now modify the code to perform the computation you require.