

GPU Programming Basics: Getting Started

1 Install CUDA

I only have access to an nVidia graphics card on an Apple MacBook Pro and on Amazon's Cloud service, so these instructions are specific to that. For general instructions, visit the nVidia website.

The CUDA SDK is available from website:

<http://developer.nvidia.com/object/gpucomputing.html>

1.1 Apple Mac

Visit the above link and click downloads under 'GPU Computing' in the left column. Click Mac OS and then download and install at least:

- Developer Drivers for MacOS
- CUDA Toolkit
- GPU Computing SDK code samples

These are simple click next lots installs. Then add the following line to your `~/.profile` file:

```
export DYLD_LIBRARY_PATH=/usr/local/cuda/lib:$DYLD_LIBRARY_PATH
```

To check that CUDA is working, run

```
/Developer/CUDA/bin/darwin/release/deviceQuery
```

You're now ready to compile CUDA programs.

1.2 Amazon Cloud GPU Host

Make sure you're signed up for an account at

<http://aws.amazon.com/>

Then:

- Log in to the AWS Management Console
- Change to the Amazon EC2 tab and ensure that the region is set to 'US East' (GPU nodes are only available in this region)
- Click the Launch Instance button and select the quick start AMI titled 'Cluster Instances HVM CentOS 5.5'.
- Choose instance type 'Cluster GPU (cg1.4xlarge, 22GB)'
- Proceed through to launch the instance (you will need to create key pairs if you don't have them already, see Amazon Getting Started Guide)

Once the instance is launched, you can connect using SSH and finish the setup for doing GPU computing as follows.

- To install R you need the EPEL archive, so run:

```
rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-4.noarch.rpm
```

- Then, install R:

```
yum install R
```

- Finally, edit the `~/.bash_profile` file to add:

```
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/local/cuda/lib:$LD_LIBRARY_PATH
```

You're now ready to compile CUDA programs.

2 Compiling CUDA C for use in R

There is a special compiler for CUDA C, so it is no longer possible to run the nice, simple command `R CMD SHLIB myfile.c` and the compile and link steps must be done manually.

I have created two shell scripts with the compiler settings I have found to work on an Apple Mac and the Amazon Cloud: `GPUcompile_mac` and `GPUcompile_linux`. They can simply be invoked by providing the name of the `.cu` file, eg:

```
./GPUcompile_mac VecAdd.cu
```

These will be available online after the talk.