Getting Started

NVIDIA CUDA C
Installation and Verification on Mac OS X

November 2009

Table of Contents

Chapter 1. Introduction

CUDA—Supercomputing on Desktop Systems

NVIDIA® CUDA™ is a general purpose parallel computing architecture introduced by NVIDIA. It includes the CUDA Instruction Set Architecture (ISA) and the parallel compute engine in the GPU. To program to the CUDA architecture, developers can, today, use C, one of the most widely used high-level programming languages, which can then be run at great performance on a CUDA enabled processor.

The CUDA architecture and its associated software were developed with several design goals in mind:

- Provide a small set of extensions to standard programming languages, like C, that enable a straightforward implementation of parallel algorithms. With CUDA and C for CUDA, programmers can focus on the task of parallelization of the algorithms rather than spending time on their implementation.
- Support heterogeneous computation where applications use both the CPU and GPU. Serial portions of applications are run on the CPU, and parallel portions are offloaded to the GPU. As such, CUDA can be incrementally applied to existing applications. The CPU and GPU are treated as separate devices that have their own memory spaces. This configuration also allows simultaneous computation on both the CPU and GPU without contention for memory resources.

CUDA-enabled GPUs have hundreds of cores that can collectively run thousands of computing threads. Each core has shared resources, including registers and memory. The on-chip shared memory allows parallel tasks running on these cores to share data without sending it over the system memory bus.

This guide will show you how to install and check the correct operation the CUDA Development Tools.

System Requirements

To use CUDA Development Tools on your system, you will need the following installed:

- CUDA-enabled GPU
- Mac OS X v. 10.5.6 or later
- The gcc compiler and toolchain installed via Xcode
- CUDA driver and software (available at no cost from http://www.nvidia.com/cuda)

About This Document

This document is intended for readers familiar with the Mac OS X environment and the compilation of C programs from the command line. You do not need previous experience with CUDA or experience with parallel computation.

Chapter 2. Installing the CUDA Development Tools

The installation of the CUDA development tools on a system running Mac OS X consists of three simple steps:

- Verify the system has a CUDA-enabled GPU, a supported version of Mac OS X, and that gcc has been installed via Xcode.
- Download the CUDA driver and software.
- Install the CUDA driver and software.

Test your installation by compiling and running one of the sample programs in the CUDA software to validate that the hardware and software are running correctly and communicating with each other.

Verify CUDA-enabled GPU

Many NVIDIA products today contain CUDA-enabled GPUs. These include:

- NVIDIA GeForce® 8, 9, and 200 series GPUs
- NVIDIA Tesla™ computing solutions
- Many of the NVIDIA Quadro® products

An up-to-date list of CUDA-enabled GPUs can be found on the NVIDIA CUDA Web site at http://www.nvidia.com/object/cuda_learn_products.html. The Release Notes for the CUDA Toolkit also contain a list of supported products.

To verify which video adapter your Mac OS X system uses, under the **Apple** menu select **About This Mac**, click the **More Info ...** button, and then select **Graphics/Displays** under the **Hardware** list.

Verify the Correct Version of Mac OS X

The CUDA Development Tools require an Intel-based Mac running Mac OS X v. 10.5.6 or later. To check which version you have, go to the Apple menu on the desktop and select **About This Mac**. You should see a dialog box similar to Figure 1.

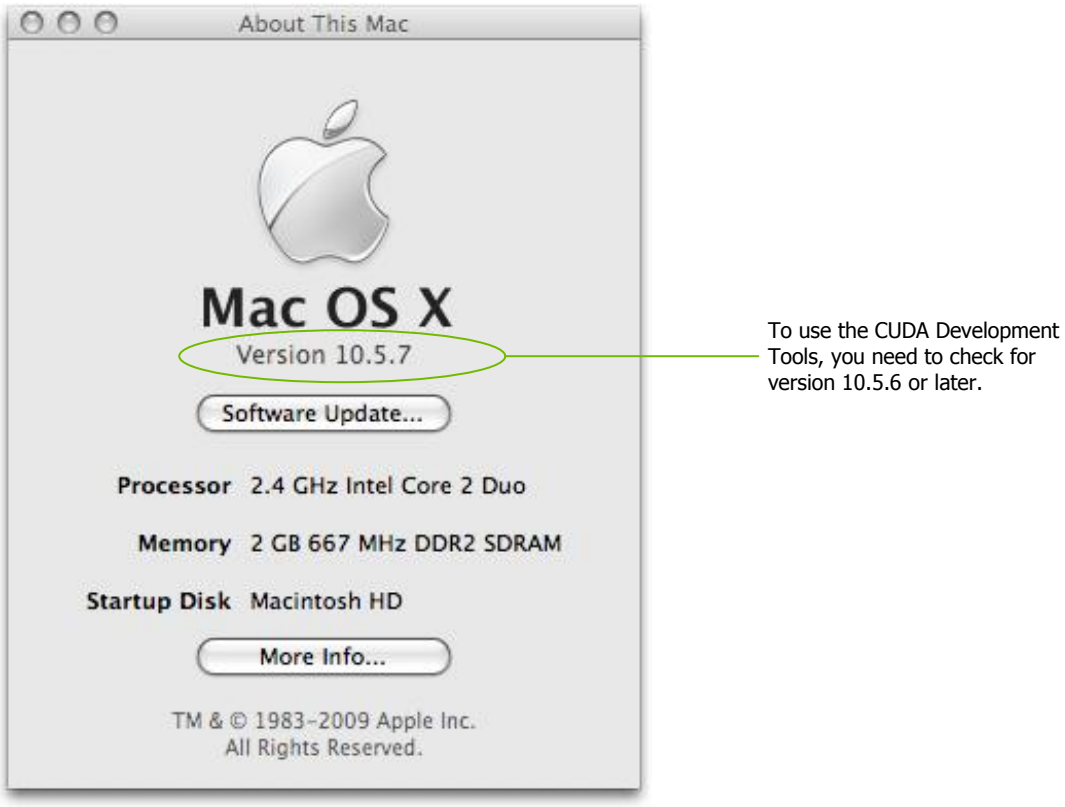

Figure 1. About This Mac Dialog Box

Note: New versions of CUDA Development Tools can require later versions of Mac OS X, so always verify that you are running the right release for the version of CUDA you are using.

Verify That gcc Is Installed

The gcc compiler and toolchain are installed via installation of Xcode. The Xcode development environment is found on the Xcode Developer Tools DVD that ships with new Mac systems and with Leopard, if you buy the operating-system upgrade. When installing Xcode, the package that contains gcc and the necessary tools is called Developer Tools Essentials. One can check that gcc is installed via the `/usr/bin/gcc --help` command issue from a Terminal window.

Download the CUDA Driver and Software

The CUDA driver and software is available from the main CUDA download site at http://www.nvidia.com/object/cuda_get.html.

On this page, choose **Mac OS** from the **Operating System** menu, and download the CUDA Driver, CUDA Toolkit, and the CUDA SDK packages for the latest version of the Development Tools.

Note that there are two CUDA driver packages: one for NVIDIA GeForce GPUs and one for NVIDIA Quadro GPUs.

The CUDA Toolkit contains the tools needed to compile and build a CUDA application in conjunction with the **nvcc** compilation driver. It includes tools, libraries, header files, and other resources.

The CUDA SDK includes sample projects that provide source code and other resources for constructing CUDA programs.

Installing the CUDA Driver and Software

Follow these few steps for a successful installation. For information not listed here, see the documentation in **/usr/local/cuda/doc**.

Download the CUDA software from http://www.nvidia.com/object/cuda_get.html and save the .pkg files as directed in the previous section.

Uninstall previous versions of the CUDA Toolkit and CUDA SDK if they have previously been installed. Do this by deleting the files from **/usr/local/cuda**, their default installation location. Adjust accordingly if you placed the files elsewhere. (If you wish to keep the files so you can compile for different versions of CUDA, then rename the existing directories and modify your makefile accordingly.)

Install the CUDA Driver by installing the appropriate CUDA driver package based on the GPU in the system (either NVIDIA GeForce GPU or NVIDIA Quadro GPU).

Install the CUDA Toolkit by following the installation notes in the NVIDIA CUDA Toolkit release notes. The CUDA Toolkit installation defaults to **/usr/local/cuda**. Several environment variables need to be defined in the installation:

PATH needs to add **/usr/local/cuda/bin**. In addition, **DYLD_LIBRARY_PATH** needs to contain **/usr/local/cuda/lib**. The typical way to place these values in your environment is with the following commands:

```
export PATH=/usr/local/cuda/bin:$PATH
export DYLD_LIBRARY_PATH=/usr/local/cuda/lib: ↵
       $DYLD_LIBRARY_PATH
```

To make these settings permanent, place them in **~/.bash_profile**.

Install the CUDA SDK by following the installation notes in the NVIDIA CUDA SDK release notes. The installation process will place the files in **/Developer/GPU Computing**. Note that this differs from the default location of the previous version of the SDK, which was placed in **/Developer/CUDA**.

Verify the Installation

Before proceeding, it's important to verify that the CUDA programs can find and communicate correctly with the CUDA-enabled hardware. To do this, you will need to compile and run some of the included sample programs.

Compiling the Examples

The version of the CUDA Toolkit can be checked by running **`nvcc -V`** in a terminal window. **`nvcc`** is the command to run the compiler driver that compiles CUDA programs. It calls the gcc compiler for C code and the NVIDIA PTX compiler for the CUDA code.

NVIDIA includes sample programs in source form in the CUDA SDK. You should compile them all by changing to **`/Developer/GPU Computing/C`** and typing **`make`**. The resulting binaries will be installed in the directory **`/Developer/GPU Computing/C/bin/darwin/release`**. If the compilation fails because it doesn't find gcc, the compiler should be installed via an Xcode installation.

Running the Binaries

The sample projects use libraries pointed to by **`DYLIB_LIBRARY_PATH`**, as described earlier, so make sure it points to the right directory.

In addition, the executables need to find **`libcutil.a`** in **`/Developer/GPU Computing/C/lib`** and the corresponding include file in **`/Developer/GPU Computing/C/common/inc`**. They also need to access graphics libraries in **`/Developer/GPU Computing/C/common/lib/darwin`**. If you have installed the CUDA software as explained earlier, these locations are defaults and the programs should run without difficulty.

Once these required files are in place, go to **`/Developer/GPU Computing/C/bin/darwin/release`** and run **`deviceQuery`**. If CUDA is installed and configured correctly, the output for **`deviceQuery`** should look similar to Figure 2.

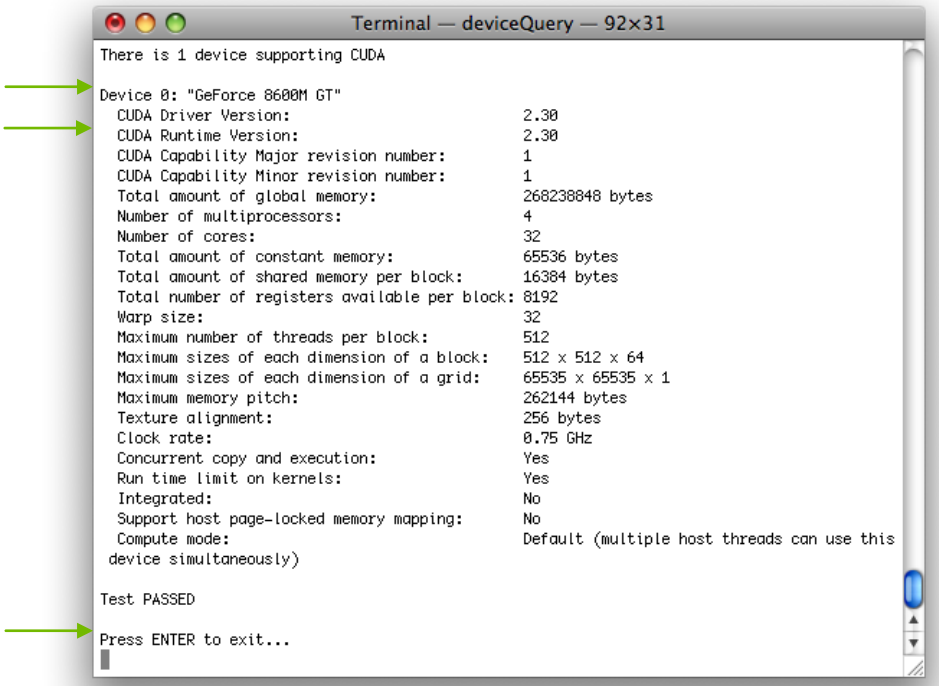

Figure 2. Valid Results from Sample CUDA **`deviceQuery`** Program

Note that the parameters for your CUDA device will vary. The key lines are the first and second ones that confirm a device was found and what model it is. Also, the next-to-last line, as indicated, should show that the test passed.

Running the **`bandwidthTest`** program ensures that the system and the CUDA device are able to communicate correctly. Its output is shown in Figure 3.

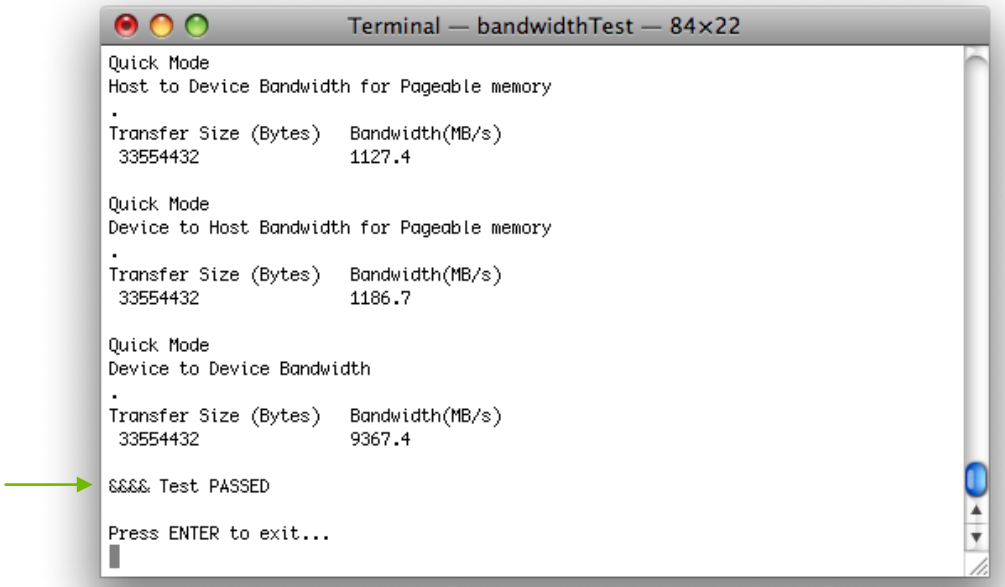

Figure 3. Valid Results from Sample CUDA **`bandwidthTest`** Program

Note that the measurements for your CUDA device description will vary from system to system. The important point is that you obtain measurements, and that the second-to-last line (highlighted) confirms that all necessary tests passed.

Should the tests not pass, make sure you have an NVIDIA GPU on your system that supports CUDA and make sure it is properly installed.

To see a graphical representation of what CUDA can do, run the sample **`particles`** executable.

Chapter 3. Additional Considerations

What's Next?

Now that you have CUDA-enabled hardware and the software installed, you can examine and enjoy the numerous included programs. To begin using CUDA to accelerate the performance of your own applications, consult the *CUDA Programming Guide*, located in **`/usr/local/cuda/doc`**.

For tech support on programming questions, consult and participate in the bulletin board and mailing list at http://forums.nvidia.com/index.php?showforum=71.

Notice

Trademarks

Copyright

NVIDIA Corporation | 2701 San Tomas Expressway | Santa Clara, CA 95050 | **www.nvidia.com**