Introduction to CUDA programming

Class 1:
NVIDIA CUDA C GETTING STARTED GUIDE FOR LINUX

2011-11-16
侯凯希
INTRODUCTION

• 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 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.
SYSTEM REQUIREMENTS

• To use CUDA on your system, you will need the following installed:
  ➢ CUDA-enabled GPU
  ➢ Device driver
  ➢ A supported version of Linux with a gcc compiler and toolchain
  ➢ CUDA software (available at no cost from http://www.nvidia.com/cuda)
INSTALLING CUDA DEVELOPMENT TOOLS

• The installation of CUDA development tools on a system running the appropriate version of Linux consists of four simple steps:
  ➢ Verify the system has a CUDA-enabled GPU and a supported version of Linux.
  ➢ Download the NVIDIA driver and the CUDA software.
  ➢ Install the NVIDIA driver.
  ➢ Install the CUDA software.
VERIFY YOU HAVE A CUDA-ENABLED SYSTEM

• Many NVIDIA products today contain CUDA-enabled GPUs. These include:
  ➢ NVIDIA GeForce ® 8, 9, 200, and 400 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
VERIFY YOU HAVE A CUDA-ENABLED SYSTEM

• To verify which video adapter your system uses, find the model number by going to your distribution's equivalent of System Properties, or, from the command line, enter:

   `lspci | grep -i nvidia`

• If you do not see any settings, update the PCI hardware database that Linux maintains by entering update-pciids (generally found in /sbin) at the command line or update /usr/share/hwdata/pci.ids and rerun the previous lspci command.

VERIFY YOU HAVE A CUDA-ENABLED SYSTEM

Verify You Have a Supported Version of Linux

• To determine which distribution and release number you're running, type the following at the command line:

  `uname -a`

• The `i386` line indicates you are running on a 32-bit system. On 64-bit systems running in 64-bit mode, this line will generally read: `x86_64`. The second line gives the version number of the operating system.
VERIFY YOU HAVE A CUDA-ENABLED SYSTEM

Verify That gcc Is Installed

To verify the version of gcc installed on your system, type the following on the command line:

```bash
gcc --version
```

If an error message displays, you need to install the "development tools" from your Linux distribution or obtain a version of gcc and its accompanying toolchain from the Web.
DOWNLOAD THE NVIDIA DRIVER AND CUDA SOFTWARE

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

➢ The GPU Computing SDK
The GPU Computing SDK includes sample projects that provide source code and other resources for constructing CUDA programs.
INSTALL THE NVIDIA DRIVER

- With the NVIDIA driver and software downloaded, you need to install the driver. Use the following procedure to install the driver:
- 1. Exit the GUI if you are in a GUI environment by pressing Ctrl-Alt-Backspace. Do NOT use init 0 command when you are in remote control.
- 2. Run the driver installation package from the command line as a superuser.

    `sh xxxxx.run`

- 3. Verify that the correct version of the driver is installed. This can be done through your System Properties (or equivalent) or by executing the command `cat /proc/driver/nvidia/version`.
INSTALL THE CUDA SOFTWARE

• This section describes the installation and configuration of the CUDA Toolkit and the GPU Computing SDK, which you previously downloaded.

• 1. Install the CUDA Toolkit by running the downloaded .run file as a superuser. The CUDA Toolkit installation defaults to /usr/local/cuda.
INSTALL THE CUDA SOFTWARE

• 2. Define the environment variables. The PATH variable needs to include /usr/local/cuda/bin. LD_LIBRARY_PATH needs to contain either /usr/local/cuda/lib or /usr/local/cuda/lib64 for 32- or 64-bit operating systems, respectively. The typical way to place these values in your environment is with the following commands:

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

• for 32-bit operating systems, with lib64 replacing lib for 64-bit operating systems as mentioned above. To make such settings permanent, place them in ~/.bash_profile.
INSTALL THE CUDA SOFTWARE

• Install the SDK (located in the second .run file) as a regular user in the default location, $(HOME)/NVIDIA_GPU_Computing_SDK.

• Installing as a regular user avoids access issues.