CUDA Quick Start Guide

Table of Contents

Chapter 1. Introduction

This guide covers the basic instructions needed to install CUDA and verify that a CUDA application can run on each supported platform.

These instructions are intended to be used on a clean installation of a supported platform. For questions which are not answered in this document, please refer to the Windows Installation Guide and Linux Installation Guide.

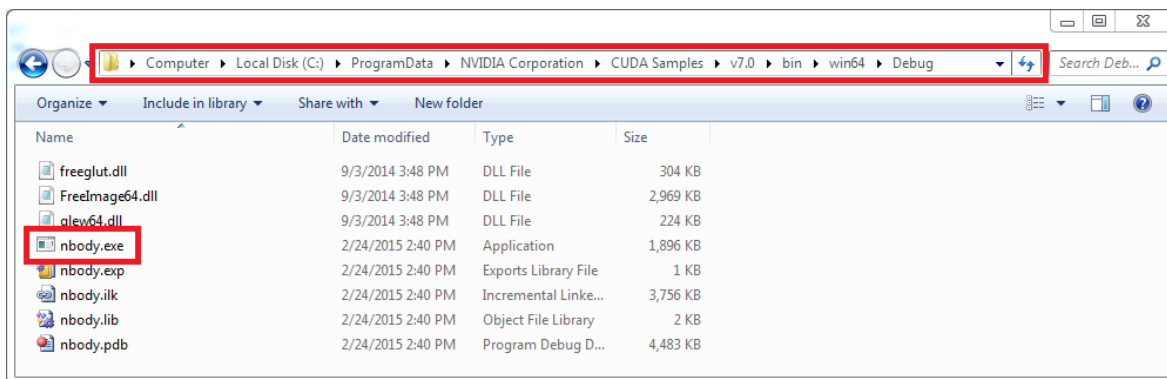The CUDA installation packages can be found on the CUDA Downloads Page.

Chapter 2. Windows

When installing CUDA on Windows, you can choose between the Network Installer and the Local Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. For more details, refer to the Windows Installation Guide.

2.1. Network Installer

Perform the following steps to install CUDA and verify the installation.

1. Launch the downloaded installer package.
2. Read and accept the EULA.
3. Select "next" to download and install all components.
4. Once the download completes, the installation will begin automatically.
5. Once the installation completes, click "next" to acknowledge the Nsight Visual Studio Edition installation summary.
6. Click "close" to close the installer.
7. Navigate to the CUDA Samples' nbody directory.
8. Open the nbody Visual Studio solution file for the version of Visual Studio you have installed.

9. Open the "Build" menu within Visual Studio and click "Build Solution".

10. Navigate to the CUDA Samples' build directory and run the nbody sample.

2.2. Local Installer

Perform the following steps to install CUDA and verify the installation.

1. Launch the downloaded installer package.
2. Read and accept the EULA.
3. Select "next" to install all components.
4. Once the installation completes, click "next" to acknowledge the Nsight Visual Studio Edition installation summary.
5. Click "close" to close the installer.
6. Navigate to the CUDA Samples' nbody directory.
7. Open the nbody Visual Studio solution file for the version of Visual Studio you have installed.
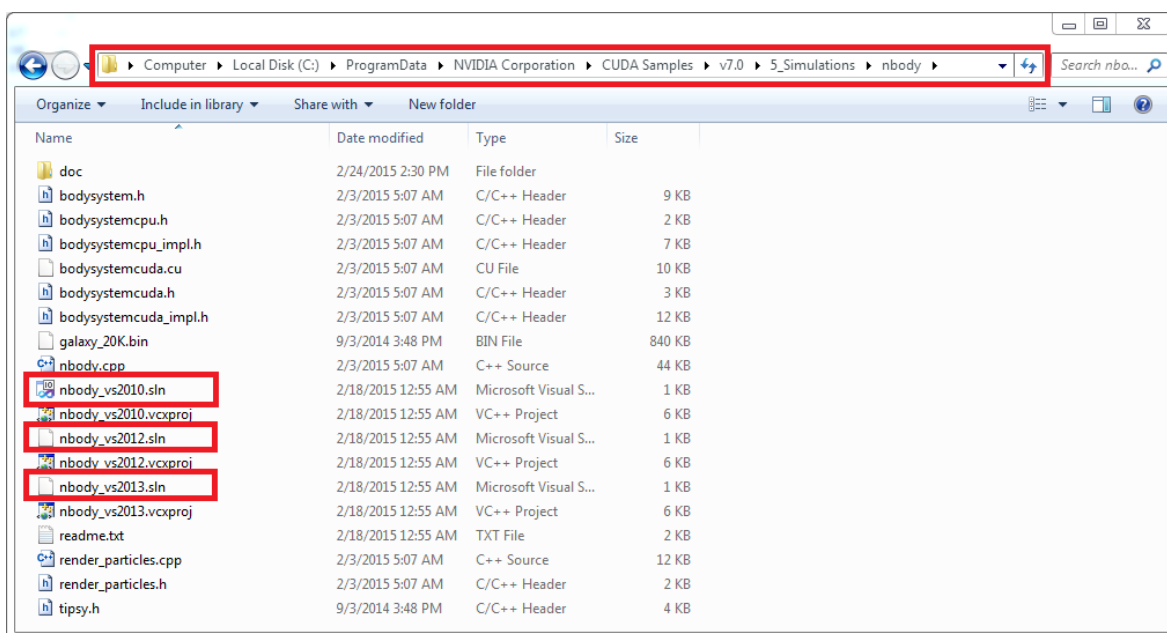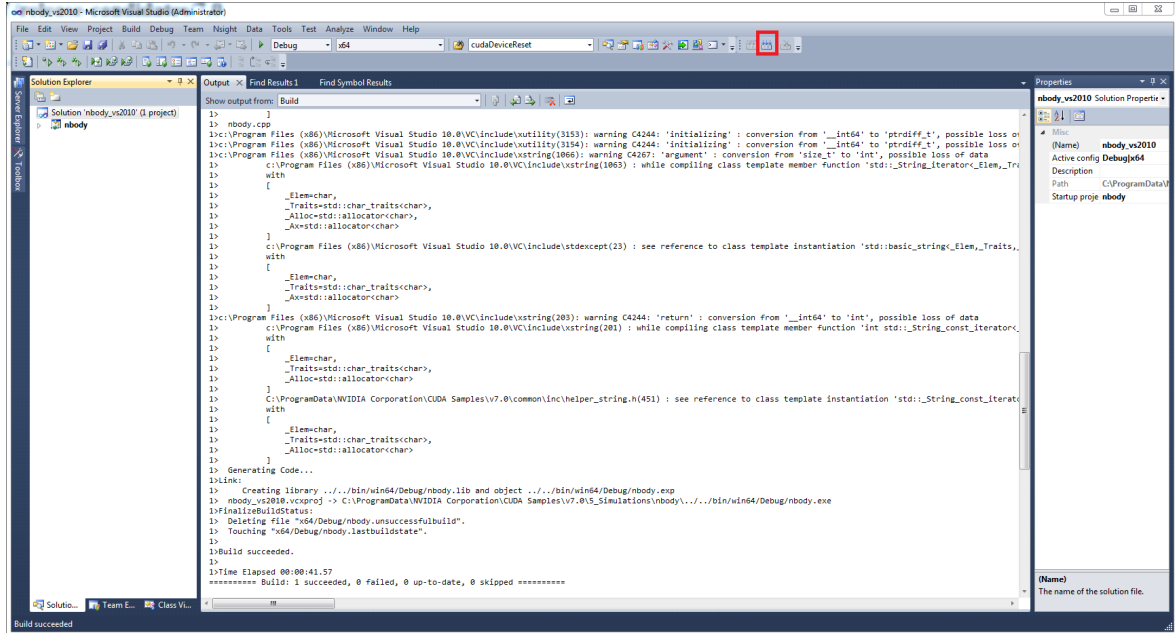
8. Open the "Build" menu within Visual Studio and click "Build Solution".

9. Navigate to the CUDA Samples' build directory and run the nbody sample.
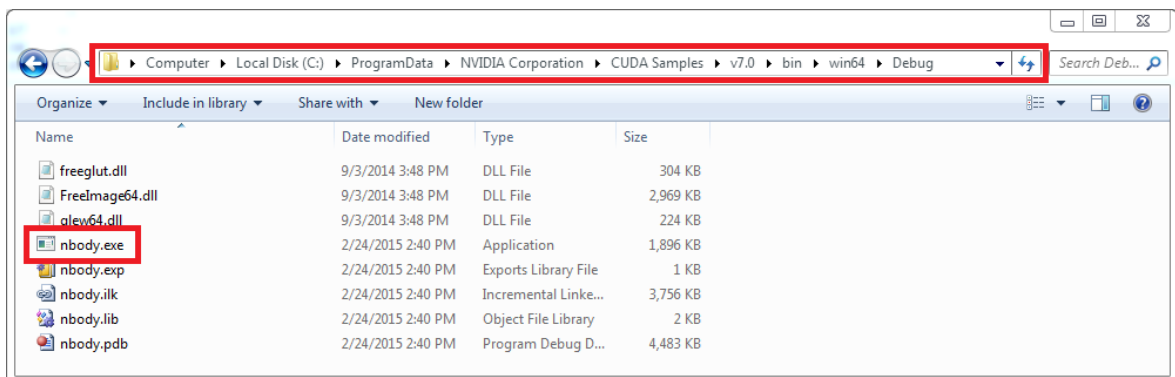
Chapter 3.   Linux

CUDA on Linux can be installed using an RPM, Debian, or Runfile package, depending on the platform being installed on.

3.1.   Linux x86_64

For development on the x86_64 architecture. In some cases, x86_64 systems may act as host platforms targeting other architectures. See the Linux Installation Guide for more details.

3.1.1.   Redhat / CentOS

When installing CUDA on Redhat or CentOS, you can choose between the Runfile Installer and the RPM Installer. The Runfile Installer is only available as a Local Installer. The RPM Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. In the case of the RPM installers, the instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

3.1.1.1.   RPM Installer

Perform the following steps to install CUDA and verify the installation.

1. Install EPEL to satisfy the DKMS dependency by following the instructions at EPEL's website.

2. **Enable optional repos**:

   On **RHEL 7 Linux** only, execute the following steps to enable optional repositories.

   ▶ **On x86_64 workstation:**
   ```
   $ subscription-manager repos --enable=rhel-7-workstation-optional-rpms
   ```
   ▶ **On POWER9 system:**
   ```
   $ subscription-manager repos --enable=rhel-7-for-power-9-optional-rpms
   ```
   ▶ **On x86_64 server:**
   ```
   $ subscription-manager repos --enable=rhel-7-server-optional-rpms
   ```

3. Install the repository meta-data, clean the yum cache, and install CUDA:
   ```
   $ sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
   $ sudo yum clean expire-cache
   $ sudo yum install cuda
   ```

4. Reboot the system to load the NVIDIA drivers.

5. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

6. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

3.1.1.2. Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Disable the Nouveau drivers:

   a). Create a file at `/etc/modprobe.d/blacklist-nouveau.conf` with the following contents:

   ```
   blacklist nouveau
   options nouveau modeset=0
   ```

   b). Regenerate the kernel initramfs:

   ```
   $ sudo dracut --force
   ```

2. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.

3. Run the installer silently to install with the default selections (implies acceptance of the EULA):

```
sudo sh cuda_<version>_linux.run --silent
```

4. Create an xorg.conf file to use the NVIDIA GPU for display:

```
$ sudo nvidia-xconfig
```

5. Reboot the system to load the graphical interface.

6. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

7. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

3.1.2. Fedora

When installing CUDA on Fedora, you can choose between the Runfile Installer and the RPM Installer. The Runfile Installer is only available as a Local Installer. The RPM Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. In the case of the RPM installers, the instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

3.1.2.1. RPM Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the RPMFusion free repository to satisfy the Akmods dependency:
   ```
   $ su -c 'dnf install --nogpgcheck http://download1.rpmfusion.org/free/fedora/
   rpmfusion-free-release-$(rpm -E %fedora).noarch.rpm'
   ```
2. Install the repository meta-data, clean the dnf cache, and install CUDA:
   ```
   $ sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
   $ sudo dnf clean expire-cache
   $ sudo dnf install cuda
   ```
3. Reboot the system to load the NVIDIA drivers.
4. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:
   ```
   $ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
   $ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                            ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
   ```
5. Install a writable copy of the samples then build and run the nbody sample:
   ```
   $ cuda-install-samples-11.2.sh ~
   $ cd ~/NVIDIA_CUDA-11.2_Samples/5_Simulations/nbody
   $ make
   $ ./nbody
   ```

3.1.2.2. Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Disable the Nouveau drivers:

   a). Create a file at `/usr/lib/modprobe.d/blacklist-nouveau.conf` with the following contents:
   ```
   blacklist nouveau
   options nouveau modeset=0
   ```
   b). Regenerate the kernel initramfs:
   ```
   $ sudo dracut --force
   ```
   c). Run the below command:
   ```
   $ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```
   d). Reboot the system.

2. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.
3. Run the installer silently to install with the default selections (implies acceptance of the EULA):
   ```
   sudo sh cuda_<version>_linux.run --silent
   ```
4. Create an xorg.conf file to use the NVIDIA GPU for display:
   ```
   $ sudo nvidia-xconfig
   ```
5. Reboot the system to load the graphical interface.
6. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:
   ```
   $ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
   $ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
   ```

```
                        ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

7. Install a writable copy of the samples then build and run the nbody sample:
```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

3.1.3.   SUSE Linux Enterprise Server

When installing CUDA on SUSE Linux Enterprise Server, you can choose between the Runfile Installer and the RPM Installer. The Runfile Installer is only available as a Local Installer. The RPM Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. In the case of the RPM installers, the instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

3.1.3.1.   RPM Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, refresh the Zypper cache, and install CUDA:
```
$ sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
$ sudo SUSEConnect --product PackageHub/15/x86_64
$ sudo zypper refresh
$ sudo zypper install cuda
```

2. Add the user to the video group:
```
$ sudo usermod -a -G video <username>
```

3. Reboot the system to load the NVIDIA drivers.

4. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:
```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                        ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

5. Install a writable copy of the samples then build and run the vectorAdd sample:
```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

3.1.3.2.   Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.

2. Run the installer silently to install with the default selections (implies acceptance of the EULA):
```
sudo sh cuda_<version>_linux.run --silent
```

3. Create an xorg.conf file to use the NVIDIA GPU for display:
```
$ sudo nvidia-xconfig
```

4. Reboot the system to load the graphical interface.

5. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

6. Install a writable copy of the samples then build and run the vectorAdd sample:

```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

3.1.4.   OpenSUSE

When installing CUDA on OpenSUSE, you can choose between the Runfile Installer and the RPM Installer. The Runfile Installer is only available as a Local Installer. The RPM Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. In the case of the RPM installers, the instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

3.1.4.1.   RPM Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, refresh the Zypper cache, and install CUDA:

```
$ sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
$ sudo zypper refresh
$ sudo zypper install cuda
```

2. Add the user to the video group:

```
$ sudo usermod -a -G video <username>
```

3. Reboot the system to load the NVIDIA drivers.

4. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

5. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

3.1.4.2.   Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Disable the Nouveau drivers:

   a). Create a file at `/etc/modprobe.d/blacklist-nouveau.conf` with the following contents:

   ```
   blacklist nouveau
   options nouveau modeset=0
   ```

   b). Regenerate the kernel initrd:

```
$ sudo /sbin/mkinitrd
```

2. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.

3. Run the installer silently to install with the default selections (implies acceptance of the EULA):

```
sudo sh cuda_<version>_linux.run --silent
```

4. Create an xorg.conf file to use the NVIDIA GPU for display:

```
$ sudo nvidia-xconfig
```

5. Reboot the system to load the graphical interface.

6. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                    ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

7. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

3.1.5.   WSL

These instructions must be used if you are installing in a WSL environment. Do not use the Ubuntu instructions in this case.

1. **Install repository meta-data**

```
$ sudo dpkg -i cuda-repo-<distro>_<version>_<architecture>.deb
```

2. **Install the CUDA public GPG key**

When installing using the local repo:

```
$ sudo apt-key add /var/cuda-repo-<version>/7fa2af80.pub
```

When installing using the network repo:

```
$ sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/
cuda/repos/<distro>/<architecture>/7fa2af80.pub
```

Pin file to prioritize CUDA repository:

```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/
<distro>/<architecture>/cuda-<distro>.pin
$ sudo mv cuda-<distro>.pin /etc/apt/preferences.d/cuda-
repository-pin-600
```

3. **Update the Apt repository cache and install CUDA**

```
$ sudo apt-get update
$ sudo apt-get install cuda
```

3.1.6.   Ubuntu

When installing CUDA on Ubuntu, you can choose between the Runfile Installer and the Debian Installer. The Runfile Installer is only available as a Local Installer. The Debian Installer is

available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. In the case of the Debian installers, the instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

3.1.6.1. Debian Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, install GPG key, update the apt-get cache, and install CUDA:

```
$ sudo dpkg --install cuda-repo-<distro>-<version>.<architecture>.deb
$ sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/
cuda/repos/<distro>/<architecture>/7fa2af80.pub
$ sudo apt-get update
$ sudo apt-get install cuda
```

2. Reboot the system to load the NVIDIA drivers.

3. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

4. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

3.1.6.2. Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Disable the Nouveau drivers:

   a). Create a file at /etc/modprobe.d/blacklist-nouveau.conf with the following contents:

   ```
   blacklist nouveau
   options nouveau modeset=0
   ```

   b). Regenerate the kernel initramfs:

   ```
   $ sudo update-initramfs -u
   ```

2. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.

3. Run the installer silently to install with the default selections (implies acceptance of the EULA):

```
sudo sh cuda_<version>_linux.run --silent
```

4. Create an xorg.conf file to use the NVIDIA GPU for display:

```
$ sudo nvidia-xconfig
```

5. Reboot the system to load the graphical interface.

6. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
```

```
                    ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

7. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

3.1.7.   Debian

When installing CUDA on Debian 10, you can choose between the Runfile Installer and the Debian Installer. The Runfile Installer is only available as a Local Installer. The Debian Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. For more details, refer to the Linux Installation Guide.

3.1.7.1.   Debian Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, install GPG key, update the apt-get cache, and install CUDA:

```
$ sudo dpkg -i cuda-repo-<distro>_<version>_<architecture>.deb
$ sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/
cuda/repos/debian10/x86_64/7fa2af80.pub
$ sudo add-apt-repository contrib
$ sudo apt-get update
$ sudo apt-get -y install cuda
```

2. Reboot the system to load the NVIDIA drivers.

3. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

4. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

3.1.7.2.   Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Disable the Nouveau drivers:

   a). Create a file at /etc/modprobe.d/blacklist-nouveau.conf with the following contents:

   ```
   blacklist nouveau
   options nouveau modeset=0
   ```

   b). Regenerate the kernel initramfs:

   ```
   $ sudo update-initramfs -u
   ```

2. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.

3. Run the installer silently to install with the default selections (implies acceptance of the EULA):

```
$ sudo sh cuda_<version>_linux.run --silent
```

4. Create an xorg.conf file to use the NVIDIA GPU for display:

```
$ sudo nvidia-xconfig
```

5. Reboot the system to load the graphical interface.

6. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                        ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

7. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

3.2. Linux POWER8

For development on the POWER8 architecture.

3.2.1. Ubuntu

When installing CUDA on Ubuntu on POWER8, you must use the Debian Installer. The Debian Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. The instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

3.2.1.1. Debian Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, update the apt-get cache, and install CUDA:

```
$ sudo dpkg --install cuda-repo-<distro>-<version>.<architecture>.deb
$ sudo apt-get update
$ sudo apt-get install cuda
```

2. Reboot the system to load the NVIDIA drivers.

3. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                        ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

4. Install a writable copy of the samples then build and run the vectorAdd sample:

```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

3.2.2. Redhat / CentOS

When installing CUDA on Redhat on POWER8, you must use the RPM Installer. The RPM Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. The instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

3.2.2.1. RPM Installer

Perform the following steps to install CUDA and verify the installation.

1. Install EPEL to satisfy the DKMS dependency by following the instructions at EPEL's website.

2. Install the repository meta-data, clean the yum cache, and install CUDA:
```
$ sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
$ sudo yum clean expire-cache
$ sudo yum install cuda
```

3. Reboot the system to load the NVIDIA drivers.

4. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:
```
$ export PATH=/usr/local/cuda-11.2/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-11.2/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

5. Install a writable copy of the samples then build and run the vectorAdd sample:
```
$ cuda-install-samples-11.2.sh ~
$ cd ~/NVIDIA_CUDA-11.2_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Trademarks

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright