NVIDIA HPC SDK INSTALLATION GUIDE

TABLE OF CONTENTS

Chapter 1.
INSTALLATIONS ON LINUX

This section describes how to install the HPC SDK in a generic manner on Linux x86_64, OpenPOWER, or Arm Server systems with NVIDIA GPUs. It covers both local and network installations.

For a complete description of supported processors, Linux distributions, and CUDA versions please see the HPC SDK Release Notes.

1.1. Prepare to Install on Linux

Linux installations require some version of the GNU Compiler Collection (including gcc, g++, and gfortran compilers) to be installed and in your $PATH prior to installing HPC SDK software. For HPC compilers to produce 64-bit executables, a 64-bit gcc compiler must be present. For C++ compiling and linking, the same must be true for g++. To determine if such a compiler is installed on your system, do the following:

1. Create a `hello.c` program.
   ```
   #include <stdio.h>
   int main() {
     printf("hello, world!\n");
     return 0;
   }
   ```
2. Compile with the `-m64` option to create a 64-bit executable.
   ```
   $ gcc -m64 -o hello_64_c hello.c
   ```

   Run the `file` command on the produced executable. The output should look similar to the following:
   ```
   $ file ./hello_64_c
   hello_64_c: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), for
   GNU/Linux 2.6.9, dynamically linked (uses shared libs), for GNU/Linux
   2.6.9, not stripped
   ```
3. For support with C++ compilation, g++ version 4.4 is required at a minimum. A more recent version will suffice. Create a `hello.cpp` program and invoke g++ with the `-m64` argument. Make sure you are able to compile, link, and run the simple hello.cpp program first before proceeding.

   ```
   #include <iostream>
   int main() {
   ```

```
    std::cout << "hello, world!\n";
    return 0;
}
```

```
$ g++ -m64 -o hello_64_cpp hello.cpp
```

The `file` command on the `hello_64_cpp` binary should produce similar results as the C example.

> Any changes to your gcc compilers requires you to reinstall the HPC SDK.

For cluster installations, access to all the nodes is required. In addition, you should be able to connect between nodes using `rsh` or `ssh`, including to/from the same node you are on. The hostnames for each node should be the same as those in the cluster machine list for the system (`machines.LINUX` file).

In a typical local installation, the default installation base directory is `/opt/nvidia/hpc_sdk`.

If you choose to perform a network installation, you should specify:

▸ A shared file system for the installation base directory. All systems using the compilers should use a common pathname.
▸ A second directory name that is local to each of the systems where the HPC compilers and tools are used. This local directory contains the libraries to use when compiling and running on that machine. Use the same pathname on every system, and point to a private (i.e. non-shared) directory location.

This directory selection approach allows a network installation to support a network of machines running different versions of Linux. If all the platforms are identical, the shared installation location can perform a standard installation that all can use.

To Prepare for the Installation:

▸ After downloading the HPC SDK installation package, bring up a shell command window on your system.

The installation instructions assume you are using csh, sh, ksh, bash, or some compatible shell. If you are using a shell that is not compatible with one of these shells, appropriate modifications are necessary when setting environment variables.

▸ Verify you have enough free disk space for the HPC SDK installation.

  ▸ The uncompressed installation packages requires 9.5 GB of total free disk space for the HPC SDK slim packages, and 20 GB for the HPC SDK multi pacakges.

1.2. Installation Steps for Linux

Follow these instructions to install the software:

1. Unpack the HPC SDK software.

In the instructions that follow, replace `<tarfile>` with the name of the file that you downloaded.

Use the following command sequence to unpack the tar file before installation.

```
% tar xpfz <tarfile>.tar.gz
```

The tar file will extract an **install** script and an **install_components** folder to a directory with the same name as the tar file.

2. Run the installation script(s).

Install the compilers by running **[sudo] ./install** from the **<tarfile>** directory.

> 💬 **Important** The installation script must run to completion to properly install the software.

To successfully run this script to completion, be prepared to do the following:

▸ Determine whether to perform a local installation or a network installation.
▸ Define where to place the installation directory. The default is `/opt/nvidia/hpc_sdk`.

> 💬 Linux users have the option of automating the installation of the HPC compiler suite without interacting with the usual prompts. This may be useful in a large institutional setting, for example, where automated installation of HPC compilers over many systems can be efficiently done with a script.

To enable the silent installation feature, set the appropriate environment variables prior to running the installation script. These variables are as follows:

NVHPC_SILENT	(required) Set this variable to "true" to enable silent installation.
NVHPC_INSTALL_DIR	(required) Set this variable to a string containing the desired installation location, e.g. **/opt/nvidia/hpc_sdk**.
NVHPC_INSTALL_TYPE	(required) Set this variable to select the type of install. The accepted values are "single" for a single system install or "network" for a network install.
NVHPC_INSTALL_LOCAL_DIR	(required for network install) Set this variable to a string containing the path to a local file system when choosing a network install.
NVHPC_DEFAULT_CUDA	(optional) Set this variable to the desired CUDA version in the form of XX.Y, e.g. 10.1 or 11.0.
NVHPC_STDPAR_CUDACC	(optional) Set this variable to force C++ stdpar GPU-compilation to target a specific compute capability by default, e.g. 60, 70, 75, etc.

The HPC SDK installation scripts install all of the binaries, tools, and libraries for the HPC SDK in the appropriate subdirectories within the specified installation directory.

3. Review documentation.

NVIDIA HPC Compiler documentation is available online in both HTML and PDF formats.

4. Complete network installation tasks.

> 💬 Skip this step if you are not installing a network installation.

For a network installation, you must run the local installation script on each system on the network where the compilers and tools will be available for use.

If your installation base directory is `/opt/nvidia/hpc_sdk` and `/usr/nvidia/shared/23.5` is the common local directory, then run the following commands on each system on the network.

```
/opt/nvidia/hpc_sdk/$NVARCH/23.5/compilers/bin/makelocalrc  -x /opt/nvidia/
hpc_sdk/$NVARCH/23.5 \
     -net /usr/nvidia/shared/23.5
```

These commands create a system-dependent file `localrc.machinename` in the `/opt/nvidia/hpc_sdk/$NVARCH/23.5/compilers/bin` directory. The commands also create the following three directories containing libraries and shared objects specific to the operating system and system libraries on that machine:

```
/usr/nvidia/shared/23.5/lib
/usr/nvidia/shared/23.5/liblf
/usr/nvidia/shared/23.5/lib64
```

> 💬 The **makelocalrc** command does allow the flexibility of having local directories with different names on different machines. However, using the same directory on different machines allows users to easily move executables between systems that use NVIDIA-supplied shared libraries.

Installation of the HPC SDK for Linux is now complete. For assistance with difficulties related to the installation, please reach out on the NVIDIA Developer Forums.

The following sections contain information detailing the directory structure of the HPC SDK installation, and instructions for end-users to initialize environment and path settings to use the compilers and tools.

1.3. End-user Environment Settings

After the software installation is complete, each user's shell environment must be initialized to use the HPC SDK.

> 💬 Each user must issue the following sequence of commands to initialize the shell environment before using the HPC SDK.

The HPC SDK keeps version numbers under an architecture type directory, e.g. Linux_x86_64/23.5. The name of the architecture is in the form of `` `uname -s`_`uname -m` ``. For OpenPOWER and Arm Server platforms the expected architecture name is "Linux_ppc64le" and "Linux_aarch64" respectively. The guide below sets the value of the

necessary uname commands to "NVARCH", but you can explicitly specify the name of the architecture if desired.

To make the HPC SDK available:

In csh, use these commands:

```
% setenv NVARCH `uname -s`_`uname -m`
% setenv NVCOMPILERS /opt/nvidia/hpc_sdk
% setenv MANPATH "$MANPATH":$NVCOMPILERS/$NVARCH/23.5/compilers/man
% set path = ($NVCOMPILERS/$NVARCH/23.5/compilers/bin $path)
```

In bash, sh, or ksh, use these commands:

```
$ NVARCH=`uname -s`_`uname -m`; export NVARCH
$ NVCOMPILERS=/opt/nvidia/hpc_sdk; export NVCOMPILERS
$ MANPATH=$MANPATH:$NVCOMPILERS/$NVARCH/23.5/compilers/man; export MANPATH
$ PATH=$NVCOMPILERS/$NVARCH/23.5/compilers/bin:$PATH; export PATH
```

Once the 64-bit compilers are available, you can make the OpenMPI commands and man pages accessible using these commands.

```
% set path = ($NVCOMPILERS/$NVARCH/23.5/comm_libs/mpi/bin $path)
% setenv MANPATH "$MANPATH":$NVCOMPILERS/$NVARCH/23.5/comm_libs/mpi/man
```

And the equivalent in bash, sh, and ksh:

```
$ export PATH=$NVCOMPILERS/$NVARCH/23.5/comm_libs/mpi/bin:$PATH
$ export MANPATH=$MANPATH:$NVCOMPILERS/$NVARCH/23.5/comm_libs/mpi/man
```

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, CUDA, CUDA-X, GPUDirect, HPC SDK, NGC, NVIDIA Volta, NVIDIA DGX, NVIDIA Nsight, NVLink, NVSwitch, and Tesla are trademarks and/ 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