

# Installing the Cray Programming Environment for Cray CS Systems

---

S-2800-1512

## Table of Contents

<b>Introduction .....</b>	<b>3</b>
<b>Part 1: For the management node .....</b>	<b>3</b>
<b>Installed File Locations .....</b>	<b>3</b>
<b>RPMs.....</b>	<b>3</b>
<b>Installation Prerequisites.....</b>	<b>4</b>
1. Check module is installed .....	4
2. Make the directories in /global .....	4
3. Create symbolic links in /opt in the management node .....	4
4. Create symbolic links in /opt in the compute nodes.....	4
5. Set environment variables .....	4
<b>Install RPM Files .....</b>	<b>4</b>
1. Libelf.so.0 .....	5
2. Cray Programming Environment (CRAYPE) .....	5
3. GCC 4.8.1 and GCC 4.9.1 .....	5
4. Cray Compiling Environment (CCE) .....	6
5. Cray Cluster Solution Programming Environment (CRAY-CS-PrgEnv) .....	6
6. MVAPICH2 2.0.1 .....	7
7. Cray-impi .....	7
8. Cray Scientific and Math Libraries (LibSci) .....	7
9. Cray LibSci for Accelerators (LibSci_ACC).....	7
10. FFTW and FFTW impi .....	8
11. CUDA Toolkit and cudatoolkit modulefile .....	8
12. Cray Performance Measurement & Analysis Tools (Perftools & PAPI) .....	9
13. Create /etc/profile.d/cray_pe.sh .....	9
14. Create a cudatoolkit modulefile and .pc file with craypkg-gen utility.....	10
15. Install the license key .....	11
<b>Check the modules after installation .....</b>	<b>12</b>
<b>Part 2: For the compute/login nodes .....</b>	<b>13</b>
<b>Apply below changes to compute nodes.....</b>	<b>13</b>
1. Install Libelf.so.0.....	13
2. Create /etc/profile.d/cray_pe.sh .....	13
3. Install CUDA Toolkit.....	13
<b>Appendix [A] Build mvapich2 RPM with CCE .....</b>	<b>14</b>
<b>Appendix [B] /etc/profile.d/cray_pe.csh.....</b>	<b>16</b>

## Introduction

The information in this guide is intended for system administrators receiving their first release of this product or upgrading from a previous release. This guide assumes the administrator has a good understanding of Cray and Linux system administration. The installation process must be run as root. If you attempt to run the process as a user without root privileges, the installation aborts. This guide shows installations in the management node as Part1 and installations in compute nodes as Part2.

## Part 1: For the management node

### Installed File Locations

This guide assumes /global is the shared directory. The installation process will install Cray-developed packages under /global/opt/cray/name/version and third-party products (typically) in /global/opt/name. Module files are installed in either /global/opt/modulefiles or /global/opt/cray/modulefiles. The dynamic libraries for Cray products are installed in /global/opt/cray/lib64. This path is added to the library search path for ldconfig to update the cache for shared libraries.

### RPMs

1. cce-8.4.2-0.20151111211600.81ce0cb9c1fe9.x86\_64.rpm
2. cray-cs-prgenv-1.0.0-109.el6.x86\_64.rpm
3. cray-dwarf-14.5.0-0.x86\_64.rpm
4. cray-fftw impi-3.3.4.6-3.201511161819.f49fd603a747b.x86\_64.rpm
5. cray-flexnet
6. cray-gcc-4.8.1-64.x86\_64.rpm
7. cray-gcc-4.9.1-13.sles11.x86\_64.rpm
8. cray-gcc-gmp-4.3.2-2.x86\_64.rpm
9. cray-gcc-mpc-0.8.1-2.x86\_64.rpm
10. cray-gcc-mpfr-2.4.2-2.x86\_64.rpm
11. cray-impi-1.0.0-0.x86\_64.rpm
12. cray-libsci-acc-cray-83-3.3.0-1.201509141624.c2ce335f3498b.el6.x86\_64.rpm
13. cray-libsci-cray-83-13.3.0-4.201511170532.1fdf5e0a72ced.el6.x86\_64.rpm
14. cray-papi-5.4.1.3-0.201510201520.eee9e763b5968.el6.x86\_64.rpm
15. craype-2.5.0-4.201511232258.dd1575fab6151.x86\_64.rpm
16. craype-module-config-cs-1.0-0.x86\_64.rpm
17. craypkg-gen-1.1.2-0.x86\_64.rpm
18. craypkg-perftools-utils-1.0.0-1.x86\_64.rpm
19. cray-set-gcc-libs-1.0.1-06.201507152117.dc1812b155d2e.x86\_64.rpm
20. fftw-3.3.4.6-3.201511232001.f49fd603a747b.el6.x86\_64.rpm
21. libelf0-0.8.13-30.3.x86\_64.rpm
22. mvapich2\_slurm-2.0.1.4-0.src.rpm
23. mvapich2\_slurm-2.0.1.4-0.x86\_64.rpm
24. mvapich2\_slurm-cray84-2.0.1.4-0.x86\_64.rpm
25. mvapich2\_slurm-gnu49-2.0.1.4-0.x86\_64.rpm
26. perftools-6.3.1-3.201511171644.3aa972aa0d37d.el6.x86\_64.rpm
27. S-2800-1512.pdf

## Installation Prerequisites

Complete the following steps prior to installation.

### 1. Check module is installed

```
# module --version  
VERSION=3.2.10
```

### 2. Make the directories in /global

Make directories in /global

```
[root@mgmt1 ~]# mkdir /global/opt  
[root@mgmt1 ~]# mkdir /global/opt/cray  
[root@mgmt1 ~]# mkdir /global/opt/gcc  
[root@mgmt1 ~]# mkdir /global/opt/modulefiles
```

This is the directory structure after new creation.

```
[root@mgmt1 ~]# tree /global/opt/  
/global/opt/  
├── cray  
├── gcc  
└── modulefiles
```

### 3. Create symbolic links in /opt in the management node

Create symbolic links (also soft link) in the management node.

```
[root@mgmt1 ~]# ln -s /global/opt/cray /opt/cray  
[root@mgmt1 ~]# ln -s /global/opt/gcc /opt/gcc  
[root@mgmt1 ~]# ln -s /global/opt/modulefiles /opt/modulefiles
```

### 4. Create symbolic links in /opt in the compute nodes

Create symbolic links in the compute nodes. The simple way to create symbolic links in all nodes is through a compute node. Log in to one of the compute nodes and create the links from there. The mount point /global is NFS a shared directory by default.

```
[root@mgmt1 ~]# ssh prod-0001  
[root@prod-0001 ~]# ln -s /global/opt/cray /opt/cray  
[root@prod-0001 ~]# ln -s /global/opt/gcc /opt/gcc  
[root@prod-0001 ~]# ln -s /global/opt/modulefiles /opt/modulefiles
```

### 5. Set environment variables

The environment variables need to be set before beginning RPM installation.

```
export CRAY_INSTALL_DEFAULT=1  
export CRAY_CPU_TARGET=haswell
```

Haswell CPUs:

(Required) This defines the CPU type on your system, which in turn determines which CPU-specific optimized libraries are installed. Use of the correct CPU-specific libraries is extremely important to obtaining best performance from your Cray system.

## Install RPM Files

RPM packages need to be installed in the following order.

## 1. Libelf.so.0

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv libelf0-0.8.13-30.3.x86_64.rpm
Preparing...      ##### [100%]
1:libelf0        ##### [100%]
```

Libelf.so.0 installation can be confirmed with the command below.

```
[root@mgmt1 ~]# /sbin/ldconfig -p | grep libelf
libelf.so.1 (libc6,x86-64) => /usr/lib64/libelf.so.1
libelf.so.0 (libc6,x86-64) => /usr/lib64/libelf.so.0
libelf.so (libc6,x86-64) => /usr/lib64/libelf.so
```

## 2. Cray Programming Environment (CRAYPE)

Install craype-module-config-cs first, then craype.

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv --prefix=/opt/cray craype-module-config-cs-1.0-0.x86_64.rpm
Preparing...      ##### [100%]
1:craype-module-config-cs##### [100%]
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv --prefix=/opt/cray craype-2.5.0-4.201511232258.dd1575fab6151.x86_64.rpm
Preparing...      ##### [100%]
1:craype         ##### [100%]
craype-2.5.0 is now default.
```

## 3. GCC 4.8.1 and GCC 4.9.1

Install below 4 RPMs packages before installing the cray-gcc-4.8.1-64.x86\_64.rpm because of some interdependencies.

1. cray-gcc-gmp-4.3.2-2.x86\_64.rpm
2. cray-gcc-mpc-0.8.1-2.x86\_64.rpm
3. cray-gcc-mpfr-2.4.2-2.x86\_64.rpm
4. cray-set-gcc-libs-1.0.1-06.201507152117.dc1812b155d2e.x86\_64.rpm

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv --prefix=/opt/cray cray-set-gcc-libs-1.0.1-06.201507152117.dc1812b155d2e.x86_64.rpm
Preparing...      ##### [100%]
1:cray-set-gcc-libs ##### [100%]
set-gcc-libs 1.0.1 is now default.
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv cray-gcc-gmp-4.3.2-2.x86_64.rpm cray-gcc-mpc-0.8.1-2.x86_64.rpm
cray-gcc-mpfr-2.4.2-2.x86_64.rpm
Preparing...      ##### [100%]
1:cray-gcc-gmp   ##### [ 33%]
2:cray-gcc-mpfr  ##### [ 67%]
3:cray-gcc-mpc   ##### [100%]
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv cray-gcc-4.8.1-64.x86_64.rpm
Preparing...      ##### [100%]
1:cray-gcc       ##### [100%]
set-gcc-libs: Creating /opt/cray/gcc-libs/libasan.so.0 -> /opt/gcc/4.8.1/snos/lib64/libasan.so.0
set-gcc-libs: Creating /opt/cray/gcc-libs/libatomic.so.1 -> /opt/gcc/4.8.1/snos/lib64/libatomic.so.1
set-gcc-libs: Creating /opt/cray/gcc-libs/libgcc_s.so.1 -> /opt/gcc/4.8.1/snos/lib64/libgcc_s.so.1
set-gcc-libs: Creating /opt/cray/gcc-libs/libgfortran.so.3 -> /opt/gcc/4.8.1/snos/lib64/libgfortran.so.3
set-gcc-libs: Creating /opt/cray/gcc-libs/libgomp.so.1 -> /opt/gcc/4.8.1/snos/lib64/libgomp.so.1
set-gcc-libs: Creating /opt/cray/gcc-libs/libitm.so.1 -> /opt/gcc/4.8.1/snos/lib64/libitm.so.1
set-gcc-libs: Creating /opt/cray/gcc-libs/libmudflap.so.0 -> /opt/gcc/4.8.1/snos/lib64/libmudflap.so.0
set-gcc-libs: Creating /opt/cray/gcc-libs/libmudflapth.so.0 -> /opt/gcc/4.8.1/snos/lib64/libmudflapth.so.0
set-gcc-libs: Creating /opt/cray/gcc-libs/libquadmath.so.0 -> /opt/gcc/4.8.1/snos/lib64/libquadmath.so.0
set-gcc-libs: Creating /opt/cray/gcc-libs/libssp.so.0 -> /opt/gcc/4.8.1/snos/lib64/libssp.so.0
set-gcc-libs: Creating /opt/cray/gcc-libs/libstdc++.so.6 -> /opt/gcc/4.8.1/snos/lib64/libstdc++.so.6
set-gcc-libs: Creating /opt/cray/gcc-libs/libtsan.so.0 -> /opt/gcc/4.8.1/snos/lib64/libtsan.so.0
set-gcc-libs: Created /opt/cray/gcc-libs links to /opt/gcc/4.8.1
gcc-4.8.1 is now default.
cray-gcc-4.8.1-64 has been installed default.
```

## Install GCC 4.9.1 with cray-gcc-4.9.1-13.sles11.x86\_64.rpm in RedHat/CentOS.

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv cray-gcc-4.9.1-13.sles11.x86_64.rpm
Preparing...      ##### [100%]
 1:cray-gcc      ##### [100%]
set-gcc-libs: Creating /opt/cray/gcc-libs/libasan.so.1 -> /opt/gcc/4.9.1/snos/lib64/libasan.so.1
set-gcc-libs: Replacing /opt/cray/gcc-libs/libatomic.so.1 -> /opt/gcc/4.8.1/snos/lib64/libatomic.so.1
set-gcc-libs: with -> /opt/gcc/4.9.1/snos/lib64/libatomic.so.1
set-gcc-libs: Creating /opt/cray/gcc-libs/libcilkrtts.so.5 -> /opt/gcc/4.9.1/snos/lib64/libcilkrtts.so.5
set-gcc-libs: Replacing /opt/cray/gcc-libs/libgcc_s.so.1 -> /opt/gcc/4.8.1/snos/lib64/libgcc_s.so.1
set-gcc-libs: with -> /opt/gcc/4.9.1/snos/lib64/libgcc_s.so.1
set-gcc-libs: Replacing /opt/cray/gcc-libs/libgfortran.so.3 -> /opt/gcc/4.8.1/snos/lib64/libgfortran.so.3
set-gcc-libs: with -> /opt/gcc/4.9.1/snos/lib64/libgfortran.so.3
set-gcc-libs: Replacing /opt/cray/gcc-libs/libgomp.so.1 -> /opt/gcc/4.8.1/snos/lib64/libgomp.so.1
set-gcc-libs: with -> /opt/gcc/4.9.1/snos/lib64/libgomp.so.1
set-gcc-libs: Replacing /opt/cray/gcc-libs/libitm.so.1 -> /opt/gcc/4.8.1/snos/lib64/libitm.so.1
set-gcc-libs: with -> /opt/gcc/4.9.1/snos/lib64/libitm.so.1
set-gcc-libs: Creating /opt/cray/gcc-libs/liblsan.so.0 -> /opt/gcc/4.9.1/snos/lib64/liblsan.so.0
set-gcc-libs: Replacing /opt/cray/gcc-libs/libquadmath.so.0 -> /opt/gcc/4.8.1/snos/lib64/libquadmath.so.0
set-gcc-libs: with -> /opt/gcc/4.9.1/snos/lib64/libquadmath.so.0
set-gcc-libs: Replacing /opt/cray/gcc-libs/libssp.so.0 -> /opt/gcc/4.8.1/snos/lib64/libssp.so.0
set-gcc-libs: with -> /opt/gcc/4.9.1/snos/lib64/libssp.so.0
set-gcc-libs: Replacing /opt/cray/gcc-libs/libstdc++.so.6 -> /opt/gcc/4.8.1/snos/lib64/libstdc++.so.6
set-gcc-libs: with -> /opt/gcc/4.9.1/snos/lib64/libstdc++.so.6
set-gcc-libs: Replacing /opt/cray/gcc-libs/libtsan.so.0 -> /opt/gcc/4.8.1/snos/lib64/libtsan.so.0
set-gcc-libs: with -> /opt/gcc/4.9.1/snos/lib64/libtsan.so.0
set-gcc-libs: Creating /opt/cray/gcc-libs/libubsan.so.0 -> /opt/gcc/4.9.1/snos/lib64/libubsan.so.0
set-gcc-libs: Creating /opt/cray/gcc-libs/libvtv.so.0 -> /opt/gcc/4.9.1/snos/lib64/libvtv.so.0
set-gcc-libs: Created /opt/cray/gcc-libs links to /opt/gcc/4.9.1
cray-gcc-4.9.1-13.sles11 has been installed non-default.
```

## 4. Cray Compiling Environment (CCE)

Install cce-8.4.2-0.20151111211600.81ce0cb9c1fe9.x86\_64.rpm with "--prefix=/opt/cray".

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv --prefix=/opt/cray cce-8.4.2-
0.20151111211600.81ce0cb9c1fe9.x86_64.rpm
Preparing...      ##### [100%]
 1:cce           ##### [100%]
cce-8.4.2 is now default.
cce-8.4.2-0.20151111211600.81ce0cb9c1fe9 has been installed default.
```

## 5. Cray Cluster Solution Programming Environment (CRAY-CS-PrgEnv)

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv cray-cs-prgenv-1.0.0-109.el6.x86_64.rpm
Preparing...      ##### [100%]
 1:cray-cs-prgenv ##### [100%]
cray-cs-prgenv-1.0.0-109.el6 has been installed default.
```

## 6. MVAPICH2 2.0.1

Install MVAPICH2 RPMs, which were built with CCE and have a SLURM 14.11.x and CUDA 7.0.x dependency. Use `--nodeps` when you have the dependency error.

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv mvapich2_slurm-2.0.1.4-0.x86_64.rpm
Preparing...      ##### [100%]
 1:mvapich2_slurm  ##### [100%]
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv mvapich2_slurm-cray84-2.0.1.4-0.x86_64.rpm --nodeps
Preparing...      ##### [100%]
 1:mvapich2_slurm-cray84 ##### [100%]
mvapich2_cce-2.0.1.4 is now default.
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv mvapich2_slurm-gnu49-2.0.1.4-0.x86_64.rpm --nodeps
Preparing...      ##### [100%]
 1:mvapich2_slurm-gnu49 ##### [100%]
mvapich2_cce-2.0.1.4 is now default.
```

See Appendix [A] if need detailed information to build with CCE.

## 7. Cray-impi

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv cray-impi-1.0.0-0.x86_64.rpm
Preparing...      ##### [100%]
 1:cray-impi      ##### [100%]
```

## 8. Cray Scientific and Math Libraries (LibSci)

Make sure `CRAY_CPU_TARGET` is set and `mvapich2` is installed before installing LibSci and LibSci\_ACC.

```
export CRAY_CPU_TARGET=haswell
```

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv --prefix=/opt/cray cray-libsci-cray-83-13.3.0-4.201511170532.1fdf5e0a72ced.el6.x86_64.rpm
Preparing...      ##### [100%]
 1:cray-libsci-cray-83 ##### [100%]
cray-libsci-13.3.0 is now default.
```

## 9. Cray LibSci for Accelerators (LibSci\_ACC)

Use `'- - nodeps'` to allow installing `cray-libsci-acc-cray-83-3.3.0-1.201509141624.c2ce335f3498b.el6.x86_64.rpm` without dependencies.

```
libcublas.so.7.0()(64bit) is needed by cray-libsci-acc-cray-83-3.3.0-1
libcuda.so.1()(64bit) is needed by cray-libsci-acc-cray-83-3.3.0-1
libcudart.so.7.0()(64bit) is needed by cray-libsci-acc-cray-83-3.3.0-1
```

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv --prefix=/opt/cray cray-libsci-acc-cray-83-3.3.0-1.201509141624.c2ce335f3498b.el6.x86_64.rpm --nodeps
Preparing...      ##### [100%]
 1:cray-libsci-acc-cray-83##### [100%]
cray-libsci_acc-3.3.0 is now default.
```

## 10. FFTW and FFTW\_impi

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv fftw-3.3.4.6-3.201511232001.f49fd603a747b.el6.x86_64.rpm
Preparing...      ##### [100%]
 1:fftw           ##### [100%]
fftw-3.3.4.6 is now default.

[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv cray-fftw_impi-3.3.4.6-
3.201511161819.f49fd603a747b.x86_64.rpm
Preparing...      ##### [100%]
 1:cray-fftw_impi ##### [100%]
fftw_impi-3.3.4.6 is now default.
fftw_impi-3.3.4.6 is now default.
```

## 11. CUDA Toolkit and cudatoolkit modulefile

### 11.1 Install CUDA Toolkit

Download and install CUDA Toolkit

1. Download "cuda\_7.0.28\_linux.run" from [http://developer.download.nvidia.com/compute/cuda/7\\_0/Prod/local\\_installers/cuda\\_7.0.28\\_linux.run](http://developer.download.nvidia.com/compute/cuda/7_0/Prod/local_installers/cuda_7.0.28_linux.run)
2. ./cuda\_7.0.28\_linux.run
3. Accept the license agreement (enter 'q' and type 'accept')
4. Set the installation directory to \${prefix}/cudatoolkit/\${version}
5. Answer questions about CUDA 7.0 Samples.

CUDA Toolkit does not include libcuda.so, which comes from the Nvidia driver package. Use the file named "NVIDIA-Linux-x86\_64-346.82.run".

1. Run the driver file with the "-x" parameter to extract the files, such as "./NVIDIA-Linux-x86\_64-346.82.run -x". (You may need to run "chmod +x" on the file first).
2. "cd" into the newly created directory (in this case, it is named "NVIDIA-Linux-x86\_64-346.82.run").
3. Copy the necessary lib\*.so. 346.82 files to an appropriate lib64 directory. It looks like the installer usually puts them under /usr/lib64.
4. Run "ldconfig" to create the symbolic links for the library files.
5. Also had to manually run "ln -s libcuda.so.1 libcuda.so" to create the /usr/lib64/libcuda.so symbolic link, which wasn't automatically added by ldconfig.

### 11.2 Install Craypkg-gen utility

The craypkg-gen 1.1.2 utility provides the system admins a tool to integrate third party software with the Cray software stack. Craypkg-gen assists with integration by creating .pc files for C, C++, and Fortran libraries and pkg-config enabled modulefiles.

Create .pc files and modulefiles for cudatoolkit to supply perftools the needed .pc files for CUDA. Actual files will be created in Step 14.



```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv craypkg-gen-1.1.2-0.x86_64.rpm
Preparing...      ##### [100%]
 1:craypkg-gen    ##### [100%]
craypkg-gen-1.1.2 is now default.
```

## 12. Cray Performance Measurement & Analysis Tools (PerfTools & PAPI)

Install 4 RPMs for PerfTools & PAPI

1. perfTools-6.3.1-3.201511171644.3aa972aa0d37d.el6.x86\_64.rpm
2. cray-papi-5.4.1.3-0.201510201520.eee9e763b5968.el6.x86\_64.rpm
3. craypkg-perfTools-utils-1.0.0-1.x86\_64.rpm
4. cray-dwarf-14.5.0-0.x86\_64.rpm

```
[root@mgmt1 pe-on-ccs-15.12]# rpm -ihv --prefix=/opt/cray cray-papi-5.4.1.3-
0.201510201520.eee9e763b5968.el6.x86_64.rpm cray-dwarf-14.5.0-0.x86_64.rpm perfTools-6.3.1-
3.201511171644.3aa972aa0d37d.el6.x86_64.rpm craypkg-perfTools-utils-1.0.0-1.x86_64.rpm
Preparing...      ##### [100%]
 1:craypkg-perfTools-utils##### [ 25%]
 2:cray-dwarf     ##### [ 50%]
 3:perfTools     ##### [ 75%]
*****
This software, PerfTools 6.3.1, has FLEXnet license support enabled. Please obtain and install the necessary
license key(s) prior to using the software.
*****
perfTools-base-6.3.1 is now default.
perfTools-6.3.1 is now default.
perfTools-lite-6.3.1 is now default.
 4:cray-papi     ##### [100%]

[root@mgmt1 pe-on-ccs-15.12]# /opt/cray/admin-pe/set_default_files/set_default_papi_5.4.1.3
papi-5.4.1.3 is now default.
```

Use '- - nodeps' to install perfTool when you have dependency errors.

## 13. Create /etc/profile.d/cray\_pe.sh

Need to create /etc/profile.d/cray\_pe.sh file in the management node, in order to modify the users' default module environment.

Check module version and update the value in /etc/profile.d/cray\_pe.sh file

```
# module --version
VERSION=3.2.10

# vi /etc/profile.d/cray_pe.sh
```

```

#!/bin/sh

if [ -d /usr/Modules/3.2.10/init ]; then
#-----#
# system-wide profile.modules          #
# Initialize modules for all sh-derivative shells #
#-----#
trap "" 1 2 3

case "$0" in
-bash|bash|*/bash) . /usr/Modules/3.2.10/init/bash ;;
-ksh|ksh|*/ksh) . /usr/Modules/3.2.10/init/ksh ;;
-sh|sh|*/sh) . /usr/Modules/3.2.10/init/sh ;;
*) . /usr/Modules/3.2.10/init/sh ;; # default for scripts
esac

trap - 1 2 3
fi

module use /opt/modulefiles
module use /opt/cray/modulefiles
module use /opt/cray/craype/default/modulefiles

export LD_LIBRARY_PATH=/opt/cray/lib64:/usr/lib64:${LD_LIBRARY_PATH}

```

Apply the change and check.

```

# source /etc/profile.d/cray_pe.sh
# module avail

-- List all available modulefiles ---

```

See Appendix [B] if need to set module environment for csh derivative shells.

#### 14. Create a cudatoolkit modulefile and .pc file with craypkg-gen utility

Create .pc files and modulefiles for cudatoolkit to supply perftools the needed .pc files for CUDA.

```

# module load craypkg-gen/1.1.2

# craypkg-gen --help
Usage: craypkg-gen [options] ${PREFIX}/product-name/product-version
Options:
  -h, --help          show this help message and exit
  -m, --modulefile    Generate a modulefile
  -p, --pcfiles       Generate pkgconfig file templates
  -o OUTPUT, --output-prefix=OUTPUT
                      Non-default path for generated files

```

Create a cudatoolkit modulefile

1. Run craypkg-gen to create .pc files for cuda libraries
 

```
# craypkg-gen -p ${prefix}/cudatoolkit/7.0.28
```

2. Customize the .pc files: Move "-lcuda" from "libs.private:" to "libs:" in "cupti.pc" (after the "-lcupti" flag)

```
# vim ${prefix}/cudatoolkit/7.0.28/extras/CUPTI/lib64/pkgconfig/cupti.pc
```

```
14 Cflags: ${cudatoolkit_includedir}
15 Libs: ${cudatoolkit_libdir} -lcupti -lcuda
16 Libs.private: -ldl -lstdc++ -lpthread -lrt -lm #external-libs-private#
17 #libraries detected, but provided by the compiler: gcc_s c
```

3. Run 'craypkg-gen -m \${prefix}/cudatoolkit/7.0.28 -o /opt' to create the modulefile for cudatoolkit in "/opt/modulefiles"

```
# craypkg-gen -m ${prefix}/cudatoolkit/7.0.28 -o /opt
```

4. Modify line 45 and 52 of /opt/modulefiles/cudatoolkit/7.0.28 to avoid some reoccurring link errors.

```
45 append-path PE_PKGCONFIG_LIBS
   cublas_device:culibos:cudadevrt:nppi:cuinj64:nvrtc:nvToolsExt:cu
   rand::npps:nvrtc-builtins:cudart:nppc:cublas
```

```
52 append-path PE_PKGCONFIG_LIBS cuda-7.0:cudart-7.0:cublas-
   7.0:curand-7.0:npps-7.0:nppi-7.0:nppc-7.0:nvToolsExt-
   7.0:cuinj64-7.0
```

## 15. Install the license key

If you need to install the license server for CCE and Perftools, use files in the tar file.

CPMAT and CCE CentOS Flexnet license manager 11.13.1:

1. cray-flexnet-installation-instructions.txt
2. cray-flexnet-daemon-11.13.1.1-4.el6.x86\_64.rpm
3. cray-flexnet-manager-11.13.1.1-4.el6.x86\_64.rpm
4. cray-flexnet-publisher-switch-11.13.1.1-4.el6.x86\_64.rpm
5. cray-flexnet-utils-11.13.1.1-4.el6.x86\_64.rpm

Get License Files from Cray:

Send your chosen FlexNET HostID(s), server hostname(s), and (optional) port number(s) back to Cray to have a license file generated. Send license key request to license\_keys <license\_keys@cray.com>.

## Check the modules after installation

```
[USER@mgmt1 pe-on-ccs-15.12]# module avail

----- /opt/cray/craype/default/modulefiles -----
craype-accel-nvidia20    craype-haswell          craype-network-infiniband
craype-accel-nvidia35    craype-ivybridge        craype-sandybridge

----- /opt/cray/modulefiles -----
cray-impi/1.0.0          fftw/3.3.4.6(default)   perftools/6.3.1(default)
cray-libsci/13.3.0(default)  fftw impi/3.3.4.6(default)  perftools-base/6.3.1(default)
cray-libsci_acc/3.3.0(default)  mvapich2_cce/2.0.1.4(default)  perftools-lite/6.3.1(default)
craype/2.5.0(default)        mvapich2_gnu/2.0.1.4      PrgEnv-cray/1.0.0(default)
craypkg-gen/1.1.2(default)    papi/5.4.1.3(default)

----- /opt/modulefiles -----
cce/8.4.2(default)  cudatoolkit/7.0.28  gcc/4.8.1(default)  gcc/4.9.1
```

## Part 2: For the compute/login nodes

### Apply below changes to compute nodes

#### 1. Install Libelf.so.0

Libelf.so.0 needs to be installed in compute nodes.

```
[root@prod-0001 ~]# rpm -ihv libelf0-0.8.13-30.3.x86_64.rpm
Preparing...      ##### [100%]
1:libelf0        ##### [100%]
```

#### 2. Create /etc/profile.d/cray\_pe.sh

Create the cray\_pe.sh in /etc/profile.d directory in compute nodes.

```
# vi /etc/profile.d/cray_pe.sh
```

```
#!/bin/sh

if [ -d /usr/Modules/3.2.10/init ]; then
#-----#
# system-wide profile.modules          #
# Initialize modules for all sh-derivative shells #
#-----#
trap "" 1 2 3

case "$0" in
-bash|bash|*/bash) . /usr/Modules/3.2.10/init/bash ;;
-ksh|ksh|*/ksh) . /usr/Modules/3.2.10/init/ksh ;;
-sh|sh|*/sh) . /usr/Modules/3.2.10/init/sh ;;
*) . /usr/Modules/3.2.10/init/sh ;; # default for scripts
esac

trap - 1 2 3
fi

module use /opt/modulefiles
module use /opt/cray/modulefiles
module use /opt/cray/craype/default/modulefiles

export LD_LIBRARY_PATH=/opt/cray/lib64:/usr/lib64:${LD_LIBRARY_PATH}
```

Apply the change and check.

```
# source /etc/profile.d/cray_pe.sh
# module avail
```

See Appendix [B] if need to set module environment for csh derivative shells.

#### 3. Install CUDA Toolkit

Install CUDA toolkit in compute nodes.

## Appendix [A] Build mvapich2 RPM with CCE

Cray LibSci and LibSci\_ACC need mvapich2, which is built with CCE. Build MVAPICH2 RPM with mvapich2\_slurm-2.0.1.4-0.src.rpm. Recommended to build RPM in the non-root account.

How to build :

```
rpm -i mvapich2_slurm-2.0.1.4-0.src.rpm
```

# normal build

```
rpmbuild -ba SPECS/mvapich-2.0.1.spec
```

# no SLURM

```
rpmbuild -ba --define "_no_slurm 1" SPECS/mvapich-2.0.1.spec
```

# no CUDA Toolkit

```
rpmbuild -ba --define "_no_ctk 1" SPECS/mvapich-2.0.1.spec
```

# no SLURM, no CUDA Toolkit

```
rpmbuild -ba --define "_no_slurm 1" --define "_no_ctk 1" SPECS/mvapich-2.0.1.spec
```

Building outside of the system default build location may require --define "\_topdir \$PWD".

To build the package from SRPM:

```
rpm -i <package>.src.rpm
cd ~/rpmbuild
rpmbuild -ba SPECS/mvapich-2.0.1.spec
```

To build without rpmbuild:

```
cd SOURCES
BUILD_DIR=<desired installation directory> \
SOURCE_VERSION=<Version of MVAPICH2 (e.g. 2.0.1)> \
SLURM_PATH=<path to lib64/libslurm.so> \
GCC_VER="4.9.1" \
CUDA_VER=<CUDA module version to load (e.g. 7.0.28)> \
./build_mvapich
```

Additional Environment variables:

```
AT_PATH=<Path to recent Autotools install>
```

The rpath values will be reset if the command "patchelf" is in the user PATH. Modifying the MVAPICH2 configuration requires editing the 2.0.1\_build\_mvapich.config file to change the configure options.

Modules below are needed for building mvapich2 rpm with CCE. If you don't have cudatoolkit go to [Step 11 CUDA Toolkit and cudatoolkit modulefile](#) and install it.

Required Modules:

1. `module load craype`
2. `module load craype-haswell`
3. `module load craype-network-infiniband`
4. `module load PrgEnv-cray`
5. **`module load cudatoolkit`**

The configuration options and paths in the spec files should be matched with your system. (Ex. SLURM/CUDA)

## Appendix [B] /etc/profile.d/cray\_pe.csh

Create /etc/profile.d/cray\_pe.csh to initialize modules for all csh derivative shells. Check module version and match the values in /etc/profile.d/cray\_pe.csh file

```
# module --version
VERSION=3.2.10
```

```
# vi /etc/profile.d/cray_pe.csh
```

```
#!/bin/sh

#-----#
# system-wide profile.modules          #
# Initialize modules for all csh-derivative shells      #
#-----#

if ( -d /usr/Modules/3.2.10/init ) then
  # Trap interrupts
  onintr -

  switch ( $0 )
  case -tcsh:
  case tcsh:
  case */tcsh:
    source /usr/Modules/3.2.10/init/tcsh
    breaksw
  case -csh:
  case csh:
  case */csh:
    source /usr/Modules/3.2.10/init/csh
    breaksw
  default:
    source /usr/Modules/3.2.10/init/csh
    breaksw
endsw

  # Restore interrupts
  onintr

  module use /opt/modulefiles
  module use /opt/cray/modulefiles
  module use /opt/cray/craype/default/modulefiles
endif

set path=($path /usr/local/cuda/bin)
if ($?LD_LIBRARY_PATH) then
  setenv LD_LIBRARY_PATH /opt/cray/lib64:/usr/lib64:${LD_LIBRARY_PATH}
else
  setenv LD_LIBRARY_PATH /opt/cray/lib64:/usr/lib64
endif
```