Installing the Cray Programming Environment for Cray CS Systems

S-2800-1603-2

Table of Contents

Introduction 3

Part 1: For the management node 3

Installed File Locations 3

RPMs 3

Installation Prerequisites 4

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

Install RPM Files 4

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 6
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. Cray Debugging Support Tools (CCDB & lgdb) 9
14. Create /etc/profile.d/cray_pe.sh 10
15. Create a cudatoolkit modulefile and .pc file with craypkg-gen utility 10
16. Install the license key 12

Check the modules after installation 12

Part 2: For the compute/login nodes 13

Apply the following changes to compute nodes 13

1. Install Libelf.so.0 13
2. Create /etc/profile.d/cray_pe.sh 13
3. Install CUDA Toolkit 13

Appendix [A] Build mvapich2 RPM with CCE 14

Appendix [B] /etc/profile.d/cray_pe.csh 17

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-8.4.5-0.20160217193131.f13c2a11d6598.x86_64.rpm
2. cray-ccdb-2.0.0-0_201603281626.186df6a2ca437_el6.x86_64.rpm
3. cray-cs-prgenv-1.0.0-109.el6.x86_64.rpm
4. cray-dwarf-15.6.0-0.201601281940.ec5196ec42ab6.el6.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-lgdb-3.0.1-0_201603252118.429b156cffdcb_el6.x86_64.rpm
13. cray-libsci-acc-cray-83-16.03.1-1.201602172103.e7da56a9cb596.el6.x86_64.rpm
14. cray-libsci-cray-83-16.03.1-1.201602221310.1c4b9af9da687.el6.x86_64.rpm
15. cray-papi-5.4.3.1-0.201601281856.700605e4e6afd.el6.x86_64.rpm
16. craype-module-config-cs-1.0-0.x86_64.rpm
17. craype-2.5.3-2.201602191703.6be97994bcce1.x86_64.rpm
18. craypkg-gen-1.3.3-0.201512172126.0bddf4aa6ef2f.x86_64.rpm
19. craypkg-perftools-utils-1.0.0-1.x86_64.rpm
20. cray-set-gcc-libs-1.0.1-06.201507152117.dc1812b155d2e.x86_64.rpm
21. fftw-3.3.4.7-1.201602231958.f4e5db0d39f42.el6.x86_64.rpm
22. fftw_impi-3.3.4.7-1.201602231958.f4e5db0d39f42.el6.x86_64.rpm
23. libelf0-0.8.13-30.3.x86_64.rpm
24. mvapich2_slurm-2.2b.0.0-0.src.rpm
25. mvapich2_slurm-2.2b.0.0-0.x86_64.rpm
26. mvapich2_slurm-gnu49-2.2b.0.0-0.x86_64.rpm
27. mvapich2_slurm-cray84-2.2b.0.0-0.x86_64.rpm
28. perftools-6.3.2-0.201602221832.98b3c632ef5fe.el6.x86_64.rpm
29. S-2800-1603-2.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-16.03]# rpm -ihv  libelf0-0.8.13-30.3.x86_64.rpm
Preparing...          ########################################### [100%]
   1:libelf0          ########################################### [100%]
```

Libelf.so.0 installation can be confirmed with this command.

```
[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-16.03]# 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-16.03]# rpm -ihv --prefix=/opt/cray craype-2.5.3-
2.201602191703.6be97994bcce1.x86_64.rpm
Preparing...           ########################################### [100%]
   1:craype            ########################################### [100%]
craype-2.5.3 is now default.
```

3. GCC 4.8.1 and GCC 4.9.1

Install following 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-16.03]# 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-16.03]# 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-16.03]# 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-16.03]# 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/libcilkrts.so.5 -> /opt/gcc/4.9.1/snos/lib64/libcilkrts.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 8.4.5 with "`--prefix=/opt/cray`".

```
[root@mgmt1 pe-on-ccs-16.03]# rpm -ihv --prefix=/opt/cray cce-8.4.5-
0.20160217193131.f13c2a11d6598.x86_64.rpm
Preparing...                ########################################### [100%]
   1:cce                    ########################################### [100%]
cce-8.4.5 is now default.
cce-8.4.5-0.20160217193131.f13c2a11d6598 has been installed default.
```

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

```
[root@mgmt1 pe-on-ccs-16.03]# 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

Install MVAPICH2 RPMs, which were built with CCE and have a SLURM 14.11.x and CUDA 7.5.x dependency. Use --nodeps when you have the dependency error.

```
[root@mgmt1 pe-on-ccs-16.03]# rpm -ihv mvapich2_slurm-2.2b.0.0-0.x86_64.rpm
Preparing...                ########################################### [100%]
   1:mvapich2_slurm         ########################################### [100%]
[root@mgmt1 pe-on-ccs-16.03]# rpm -ihv mvapich2_slurm-cray84-2.2b.0.0-0.x86_64.rpm --nodeps
Preparing...                ########################################### [100%]
   1:mvapich2_slurm-cray84  ########################################### [100%]
mvapich2_cce-2.2b.0.0 is now default.
[root@mgmt1 pe-on-ccs-16.03]# rpm -ihv mvapich2_slurm-gnu49-2.2b.0.0-0.x86_64.rpm --nodeps
Preparing...                ########################################### [100%]
   1:mvapich2_slurm-gnu49   ########################################### [100%]
mvapich2_cce-2.2b.0.0 is now default.
```

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

7. Cray-impi

```
[root@mgmt1 pe-on-ccs-16.03]# 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-16.03]# rpm -ihv --prefix=/opt/cray cray-libsci-cray-83-16.03.1-
1.201602221310.1c4b9af9da687.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:cray-libsci-cray-83    ########################################### [100%]
cray-libsci-16.03.1 is now default.
```

9. Cray LibSci for Accelerators (LibSci_ACC)

Use '- - nodeps' to allow installing cray-libsci-acc-cray-83-16.03.1-1.201602172103.e7da56a9cb596.el6.x86_64.rpm without dependencies.

```
libcublas.so.7.5()(64bit) is needed by cray-libsci-acc-cray-83-16.03.1-1
libcuda.so.1()(64bit) is needed by cray-libsci-acc-cray-83-16.03.1-1
libcudart.so.7.5()(64bit) is needed by cray-libsci-acc-cray-83-16.03.1-1
```

```
[root@mgmt1 pe-on-ccs-16.03]# rpm -ihv --prefix=/opt/cray cray-libsci-acc-cray-83-16.03.1-
1.201602172103.e7da56a9cb596.el6.x86_64.rpm --nodeps
Preparing...                ########################################### [100%]
   1:cray-libsci-acc-cray-83########################################### [100%]
cray-libsci_acc-16.03.1 is now default.
```

10. FFTW and FFTW_impi

```
[root@mgmt1 pe-on-ccs-16.03]# rpm -ihv fftw-3.3.4.7-1.201602231958.f4e5db0d39f42.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:fftw                   ########################################### [100%]
fftw-3.3.4.7 is now default.

[root@mgmt1 pe-on-ccs-16.03]# rpm -ihv fftw_impi-3.3.4.7-
1.201602231958.f4e5db0d39f42.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:fftw_impi              ########################################### [100%]
fftw_impi-3.3.4.7 is now default.
```

11. CUDA Toolkit and cudatoolkit modulefile

11.1 Install CUDA Toolkit

Download and install CUDA Toolkit

1. Download “cuda_7.5.18_linux.run” from https://developer.nvidia.com/cuda-downloads
2. ./cuda_7.5.18_linux.run
3. Accept the license agreement (enter 'q' and type 'accept')
4. Set the installation directory to ${prefix}/cudatoolkit/7.5.18
5. Answer questions about CUDA 7.5 Samples.

CUDA Toolkit does not include libcuda.so, which comes from the Nvidia driver package. Use the file named "NVIDIA-Linux-x86_64-352.68.run".

1. Run the driver file with the "-x" parameter to extract the files, such as "./NVIDIA-Linux-x86_64-352.68.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_64-352.68.run").
3. Copy the necessary lib*.so. 352.68 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 1.3.3 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-16.03]# rpm -ihv --prefix=/opt/cray craypkg-gen-1.3.3-
0.201512172126.0bddf4aa6ef2f.x86_64.rpm
Preparing...                ########################################### [100%]
   1:craypkg-gen            ########################################### [100%]
[root@mgmt1 pe-on-ccs-16.03]# /opt/cray/admin-pe/set_default_files/set_default_craypkg-gen_1.3.3
craypkg-gen-1.3.3 is now default.
```

12. Cray Performance Measurement & Analysis Tools (Perftools & PAPI)

Install 4 RPMs for Perftools & PAPI

1. perftools-6.3.2-0.201602221832.98b3c632ef5fe.el6.x86_64.rpm
2. cray-papi-5.4.3.1-0.201601281856.700605e4e6afd.el6.x86_64.rpm
3. craypkg-perftools-utils-1.0.0-1.x86_64.rpm
4. cray-dwarf-15.6.0-0.201601281940.ec5196ec42ab6.el6.x86_64.rpm

```
[root@mgmt1 pe-on-ccs-16.03]# rpm -ihv --prefix=/opt/cray cray-papi-5.4.3.1-
0.201601281856.700605e4e6afd.el6.x86_64.rpm cray-dwarf-15.6.0-
0.201601281940.ec5196ec42ab6.el6.x86_64.rpm perftools-6.3.2-
0.201602221832.98b3c632ef5fe.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 6.3.2, has FLEXnet license support enabled.  Please obtain and install the necessary
license key(s) prior to using the software.
************************************************************
perftools-base-6.3.2 is now default.
perftools-6.3.2 is now default.
perftools-lite-6.3.2 is now default.
   4:cray-papi              ########################################### [100%]
[root@mgmt1 pe-on-ccs-16.03]# /opt/cray/admin-pe/set_default_files/set_default_papi_5.4.3.1
papi-5.4.3.1 is now default.
```

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

13. Cray Debugging Support Tools (CCDB & lgdb)

Install CCDB 2.0.0 and lgdb 3.0.1.

```
[root@mgmt1 pe-on-ccs-16.03]# rpm -ivh --prefix=/opt/ cray-lgdb-3.0.1-
0_201603252118.429b156cffdcb_el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:cray-lgdb              ########################################### [100%]
cray-lgdb-3.0.1 is now default.

[root@mgmt1 pe-on-ccs-16.03]# rpm -ivh --prefix=/opt/cray cray-ccdb-2.0.0-
0_201603281626.186df6a2ca437_el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:cray-ccdb              ########################################### [100%]
cray-ccdb-2.0.0 is now default.
```

14. 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.

15. 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/1.3.3

# 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/7.5.18
   ```

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

   ```
   # vim ${prefix}/cudatoolkit/7.5.18/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:
   ```

3. Run 'craypkg-gen –m ${prefix}/cudatoolkit/7.5.18 -o /opt' to create the modulefile for cudatoolkit in "/opt/modulefiles".

   ```
   # craypkg-gen -m ${prefix}/cudatoolkit/7.5.18 -o /opt
   ```

4. Modify /opt/modulefiles/cudatoolkit/7.5.18 to avoid some reoccurring link errors. Comment out the original lines in blue and add new lines like below. Make sure to remove nvblas from PE_PKGCONFIG_LIBS and cudatoolkit_PKGCONFIG_LIBS.

   ```
   ## append-path PE_PKGCONFIG_LIBS
   {culibos:cusolver_static:cufftw_static:cufft_static:curand_static:cudart_st
   atic:npps_static:cudadevrt:cublas_static:nppi_static:nppc_static:cusparse_s
   tatic:cublas_device:OpenCL:c:cublas:nvrtc:cudart:cufft:nvrtc-
   builtins:nppc:nppi:nvblas:curand:nvToolsExt:npps:cusparse:cufftw:cuinj64} #
   Depricated CrayPE variable

   append-path PE_PKGCONFIG_LIBS
   {culibos:cudadevrt:cuinj64:nppc:npps:nvToolsExt:nppi:cublas:curand:nvrtc:nv
   rtc-builtins:cudart} # Depricated CrayPE variable

   ## append-path cudatoolkit_PKGCONFIG_LIBS
   {culibos:cusolver_static:cufftw_static:cufft_static:curand_static:cudart_st
   atic:npps_static:cudadevrt:cublas_static:nppi_static:nppc_static:cusparse_s
   tatic:cublas_device:OpenCL:c:cublas:nvrtc:cudart:cufft:nvrtc-
   builtins:nppc:nppi:nvblas:curand:nvToolsExt:npps:cusparse:cufftw:cuinj64}

   append-path cudatoolkit_PKGCONFIG_LIBS
   {culibos:cudadevrt:cuinj64:nppc:npps:nvToolsExt:nppi:cublas:curand:nvrtc:nv
   rtc-builtins:cudart}
   ```

```
## append-path PE_PKGCONFIG_LIBS {cuda-7.5:cudart-7.5:cublas-7.5:cufft-
7.5:cufftw-7.5:curand-7.5:cusolver-7.5:cusparse-7.5:npps-7.5:nppi-7.5:nppc-
7.5:nvrtc-7.5:nvToolsExt-7.5:cuinj64-7.5} # Depricated CrayPE variable

append-path PE_PKGCONFIG_LIBS {cuda-7.5:cudart-7.5:cublas-7.5:curand-
7.5:npps-7.5:nppi-7.5:nppc-7.5:nvrtc-7.5:nvToolsExt-7.5:cuinj64-7.5} #
Depricated CrayPE variable

## append-path cudatoolkit_PKGCONFIG_LIBS {cuda-7.5:cudart-7.5:cublas-
7.5:cufft-7.5:cufftw-7.5:curand-7.5:cusolver-7.5:cusparse-7.5:npps-
7.5:nppi-7.5:nppc-7.5:nvrtc-7.5:nvToolsExt-7.5:cuinj64-7.5}

append-path cudatoolkit_PKGCONFIG_LIBS {cuda-7.5:cudart-7.5:cublas-
7.5:curand-7.5:npps-7.5:nppi-7.5:nppc-7.5:nvrtc-7.5:nvToolsExt-7.5:cuinj64-
7.5}
```

16. Install the license key

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

CPMAT and CCE 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-16.03]# 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-ccdb/2.0.0(default)          craypkg-gen/1.3.3(default)       perftools/6.3.2(default)
cray-impi/1.0.0                   fftw/3.3.4.7(default)            perftools-base/6.3.2(default)
cray-lgdb/3.0.1(default)          fftw_impi/3.3.4.7(default)       perftools-lite/6.3.2(default)
cray-libsci/16.03.1(default)      mvapich2_cce/2.2b.0.0(default)   PrgEnv-cray/1.0.0(default)
cray-libsci_acc/16.03.1(default) mvapich2_gnu/2.2b.0.0
craype/2.5.3(default)             papi/5.4.3.1(default)

---------------------------------------------- /opt/modulefiles -----------------------------------------------
cce/8.4.5(default)                flexnet-publisher/11.13.1.1    gcc/4.9.1
cudatoolkit/7.5.18(default)       gcc/4.8.1(default)
```

Part 2: For the compute/login nodes

Apply the following 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-2.2b.0.0-0.src.rpm. Recommended to build RPM in the non-root account. Make a backup of your rpmbuild/ directory before the next steps.

How to build:

```
rpm -i mvapich2_slurm-2.2b.0.0-0.src.rpm
```

normal build

```
rpmbuild -ba rpmbuild/SPECS/mvapich-2.2.spec
```

no SLURM

```
rpmbuild -ba --define "_no_slurm 1" rpmbuild/SPECS/mvapich-2.2.spec
```

no CUDA Toolkit

```
rpmbuild -ba --define "_no_ctk 1" rpmbuild/SPECS/mvapich-2.2.spec
```

no SLURM, no CUDA Toolkit

```
rpmbuild -ba --define "_no_slurm 1" --define "_no_ctk 1" rpmbuild/SPECS/mvapich-2.2.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-2.2.spec
```

To build without rpmbuild:

```
cd SOURCES
BUILD_DIR="<desired installation directory> \
SOURCE_VERSION="<Version of MVAPICH2 (e.g. 2.2)>" \
SLURM_PATH="<path to lib64/libslurm.so>" \
GCC_VER="4.9.1" \
CUDA_VER="<CUDA module version to load (e.g. 7.5.18)>" \
./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 2.2b_build_mvapich.config file to change the configure options.

These modules are required to build mvapich2 with CCE. If you don't have the 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/Cudatoolkit)

Q&A:

1. If you have this error during the build you should review the Step 4, page 11. Make sure to remove ***nvblas*** from PE_PKGCONFIG_LIBS and cudatoolkit_PKGCONFIG_LIBS.

```
configure: error: Unable to configure with Fortran support because configure
could not determine the size of a Fortran INTEGER. Consider setting
CROSS_F77_SIZEOF_INTEGER to the length in bytes of a Fortran INTEGER
```

2. If you have the warnings/errors for "Checking user environment" you should install GNU Autotools in your local directory.

```
###################################
## Checking user environment
###################################

Verifying the location of autogen.sh... done
Checking if autotools are in the same location... yes, all in
/home/test_user/local
Checking for autoconf version... >= 2.67
Checking for automake version... >= 1.12.3
Checking for libtool version... >= 2.4
Checking for UNIX find... done
Checking if xargs rm -rf works... yes
```

If you did not get the results like above, setup GNU Autotools in the local directory:

```
mkdir $HOME/local
mkdir build_tools
cd build_tools/

wget https://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.gz
wget https://ftp.gnu.org/gnu/automake/automake-1.15.tar.gz
wget https://ftp.gnu.org/gnu/libtool/libtool-2.4.6.tar.gz
wget https://ftp.gnu.org/gnu/m4/m4-1.4.17.tar.gz

tar xvf autoconf-2.69.tar.gz
cd autoconf-2.69
./configure --prefix $HOME/local
make install
export PATH=$HOME/local/bin:$PATH
```

```
cd ..
tar xvf automake-1.15.tar.gz
cd automake-1.15
./configure --prefix $HOME/local
make install

cd ..
tar xvf libtool-2.4.6.tar.gz
cd libtool-2.4.6
./configure --prefix $HOME/local
make install

cd ..
tar xvf m4-1.4.17.tar.gz
cd m4-1.4.17
./configure --prefix $HOME/local
make install

ls ~/local/bin/
aclocal  aclocal-1.15  autoconf  autoheader  autom4te  automake  automake-1.15
autoreconf  autoscan  autoupdate  ifnames  libtool  libtoolize  m4

cd
rpm -i mvapich2_slurm-2.2b.0.0-0.src.rpm

PATH=$HOME/local/bin:$PATH rpmbuild -ba --define "_no_slurm 1" rpmbuild/SPECS/mvapich-
2.2.spec 2>&1 | tee build.out

ls RPMS/x86_64/
mvapich2_slurm-2.2b.0.0.custom_noslurm-0.x86_64.rpm
mvapich2_slurm-cray84-2.2b.0.0.custom_noslurm-0.x86_64.rpm
mvapich2_slurm-gnu49-2.2b.0.0.custom_noslurm-0.x86_64.rpm
```

3. If you are using a QLogic InfiniBand adapter you should add "--with-device=ch3:psm" option in rpmbuild/SOURCES/2.2b_build_mvapich.config file.

```
CRAY_CONFIG="--enable-f77                                                    \
             --enable-fc                                                     \
             --enable-cxx                                                    \
             --enable-shared                                                 \
             --prefix=${BUILD_DIR}/${SOURCE_VERSION}/${compiler}/${PE_LEVEL} \
              ${BUILD_SLURM_OPTION}                                          \
             --cache-file=/dev/null                                          \
             ${BUILD_CTK_OPTION} --with-device=ch3:psm"

GNU_CONFIG="--enable-f77                                                     \
            --enable-fc                                                      \
            --enable-cxx                                                     \
            --enable-shared                                                  \
            --prefix=${BUILD_DIR}/${SOURCE_VERSION}/${compiler}/${GCC_MAJ_MIN} \
             ${BUILD_SLURM_OPTION}                                           \
            --cache-file=/dev/null                                           \
             ${BUILD_CTK_OPTION} --with-device=ch3:psm"
```

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
```