

Installing the Cray Programming Environment for Cray CS Systems

S-2800-1508

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	5
4. Cray Compiling Environment (CCE).....	5
5. Cray Cluster Solution Programming Environment (CRAY-CS-PrgEnv).....	6
6. MVAPICH2 2.0.1	6
7. Cray Scientific and Math Libraries (LibSci).....	6
8. Cray LibSci for Accelerators (LibSci_ACC)	6
9. CUDA Toolkit and cudatoolkit modulefile	7
10. Cray Performance Measurement & Analysis Tools (Perftools & PAPI).....	7
11. Create /etc/profile.d/cray_pe.sh.....	8
12. Create a cudatoolkit modulefile and .pc file with craypkg-gen utility	9
13. Install the license key	10
Check the modules after installation	10
Part 2: For the compute/login nodes	11
Apply below changes to compute nodes	11
1. Install Libelf.so.0	11
2. Create /etc/profile.d/cray_pe.sh.....	11
3. Install CUDA Toolkit	11
Appendix [A] Build mvapich2 RPM with CCE	12
Appendix [B] /etc/profile.d/cray_pe.csh	13

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

The contents of the tar file are:

1. cce-8.3.14-102.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-gcc-4.8.1-64.x86_64.rpm
5. cray-gcc-gmp-4.3.2-2.x86_64.rpm
6. cray-gcc-mpc-0.8.1-2.x86_64.rpm
7. cray-gcc-mpfr-2.4.2-2.x86_64.rpm
8. cray-libsci-acc-cray-83-3.2.0-2.201507202252.dda83f469f97b.el6.x86_64.rpm
9. cray-libsci-cray-83-13.1.0-1.201507201541.04f25b440650d.el6.x86_64.rpm
10. cray-papi-5.4.1.1-0.x86_64.rpm
11. craype-2.3.0-1.201503241702.b32ad20992800.x86_64.rpm
12. craypkg-gen-1.1.2-0.x86_64.rpm
13. craypkg-perftools-utils-1.0.0-1.x86_64.rpm
14. cray-set-gcc-libs-1.0.0-02.x86_64.rpm
15. libelf0-0.8.13-30.3.x86_64.rpm
16. perftools-6.2.5-0.x86_64.rpm
17. perftools-clients-6.2.5-0.x86_64.rpm
18. mvapich_slurm-cray83-2.0.1.3-0.x86_64.rpm
19. mvapich_slurm-gnu48-2.0.1.3-0.x86_64.rpm
20. mvapich2-2.0.1-cray-83-1507.spec
21. mvapich2-2.0.1-gnu-48-1507.spec
22. cray-flexnet
23. S-2800-1508.pdf

Installation Prerequisites

Before beginning installation, need to follow the below steps.

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 log in the one of compute nodes and create the links in there. /global is NFS shared directory.

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

Libelf.so.0 installation can confirm with below 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)

```
[root@mgmt1 ~]# rpm -ihv craype-2.3.0-1.201503241702.b32ad20992800.x86_64.rpm
Preparing...      ##### [100%]
1:craype         ##### [100%]
craype-2.3.0 is now default.
```

3. GCC 4.8.1

Install below 5 RPMs packages before install 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-1.0.0-02.x86_64.rpm

```
[root@mgmt1 ~]# 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.
```

4. Cray Compiling Environment (CCE)

Use '- - nodeps' to allow installing cce-8.3.14-102.x86_64.rpm without below dependency.

```
libpmi.so.0()(64bit) is needed by cce-8.3.14-102.x86_64
librca.so.0()(64bit) is needed by cce-8.3.14-102.x86_64
```

```
[root@mgmt1 ~]# rpm -ihv cce-8.3.14-102.x86_64.rpm --prefix=/opt/cray --nodeps
Preparing...      ##### [100%]
1:cce            ##### [100%]
cce-8.3.14 is now default.
cce-8.3.14-102 has been installed default.
```

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

```
[root@mgmt1 ~]# 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 build with CCE and have SLURM and CUDA 6.5.x dependency. Use --nodeps when you have the dependency error.

```
[root@mgmt1 ~]# rpm -ihv mvapich_slurm-cray83-2.0.1.3-0.x86_64.rpm --nodeps
Preparing...      ##### [100%]
 1:mvapich_slurm-cray83 ##### [100%]
mvapich2_cce-2.0.1.3 is now default.
[root@mgmt1 ~]# rpm -ihv mvapich_slurm-gnu48-2.0.1.3-0.x86_64.rpm --nodeps
Preparing...      ##### [100%]
 1:mvapich_slurm-gnu48 ##### [100%]
mvapich2_cce-2.0.1.3 is now default.
```

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

7. 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 ~]# rpm -ihv cray-libsci-cray-83-13.1.0-1.201507201541.04f25b440650d.el6.x86_64.rpm
Preparing...      ##### [100%]
 1:cray-libsci-cray-83 ##### [100%]
cray-libsci-13.1.0 is now default.
```

8. Cray LibSci for Accelerators (LibSci_ACC)

Use '- - nodeps' to allow installing cray-libsci-acc-cray-83-3.2.0-2.201507202252.dda83f469f97b.el6.x86_64.rpm without dependencies.

```
libcublas.so.6.5()(64bit) is needed by cray-libsci-acc-cray-83-3.2.0-2
libcuda.so.1()(64bit) is needed by cray-libsci-acc-cray-83-3.2.0-2
libcudart.so.6.5()(64bit) is needed by cray-libsci-acc-cray-83-3.2.0-2
```

```
[root@mgmt1 ~]# rpm -ihv cray-libsci-acc-cray-83-3.2.0-2.201507202252.dda83f469f97b.el6.x86_64.rpm
--nodeps
Preparing...      ##### [100%]
 1:cray-libsci-acc-cray-83##### [100%]
cray-libsci_acc-3.2.0 is now default.
```

9. CUDA Toolkit and cudatoolkit modulefile

9.1 Install CUDA Toolkit

Download and install CUDA Toolkit

1. Download "cuda_6.5.14_linux_64.run" from http://developer.download.nvidia.com/compute/cuda/6_5/rel/installers/cuda_6.5.14_linux_64.run
2. `./cuda_6.5.14_linux_64.run`
3. Accept the license agreement (enter 'q' and type 'accept')
4. Set the installation directory to `${prefix}/cudatoolkit/${version}`
5. Answer questions about
 - Install a symbolic link
 - CUDA 6.5 Samples.

CUDA Toolkit does not include `libcuda.so`, which comes from the Nvidia driver package. Used the file named "NVIDIA-Linux-x86_64-340.87.run".

1. Run the driver file with the "-x" parameter to extract the files, such as `./NVIDIA-Linux-x86_64-340.87.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_64-340.87.run").
3. Copy the necessary `lib*.so.340.87` 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`.

9.2 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 12.

```
[root@mgmt1 ~]# 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.
```

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

Install 5 RPMs for `Perftools & PAPI`

1. `perftools-6.2.5-0.x86_64.rpm`
2. `perftools-clients-6.2.5-0.x86_64.rpm`
3. `cray-papi-5.4.1.1-0 x86_64.rpm`

4. craypkg-perftools-utils-1.0.0-1.x86_64.rpm
5. cray-dwarf-14.5.0-0.x86_64.rpm

```
[root@mgmt1 ~]# rpm -ihv cray-papi-5.4.1.1-0.x86_64.rpm cray-dwarf-14.5.0-0.x86_64.rpm perftools-6.2.5-0.x86_64.rpm perftools-clients-6.2.5-0.x86_64.rpm craypkg-perftools-utils-1.0.0-1.x86_64.rpm
Preparing... ##### [100%]
 1:craypkg-perftools-utils##### [ 20%]
 2:perftools-clients ##### [ 40%]
perftools-6.2.5 is now default.
perftools-lite-6.2.5 is now default.
 3:cray-dwarf ##### [ 60%]
 4:perftools ##### [ 80%]
*****
This software, Perftools 6.2.5, has FLEXnet license support enabled. Please obtain and install the necessary license key(s) prior to using the software.
*****
perftools-6.2.5 is now default.
perftools-lite-6.2.5 is now default.
 5:cray-papi ##### [100%]
papi-5.4.1.1 is now default.
```

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

11. 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.

12. 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/6.5.14`
2. Customize the .pc files: Move "-lcuda" from "libs.private:" to "libs:" in "cupti.pc" (after the "-lcupti" flag)

```
# vim ${prefix}/cudatoolkit/6.5.14/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/6.5.14 -o /opt'` to create the modulefile for cudatoolkit in `"/opt/modulefiles"`

```
# craypkg-gen -m ${prefix}/cudatoolkit/6.5.14 -o /opt
```

4. Modify line 24 of `/opt/modulefiles/cudatoolkit/6.5.14` to avoid some reoccurring link errors.

```
24 append-path PE_PKGCONFIG_LIBS
   culibos:cudadevrt:cublas_device:cufftw:cublas:npps:nppi:cuinj64:
   curand:cufft:cudart:cusparse:nvToolsExt:nppc
```

13. Install the license key

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

CPMAT and CCE CentOS Flexnet license manager 11.12.1:

```
cray-flexnet-11.12.1_readme
cray-flexnet-daemon-11.12.1-1.0000.9450.6.1.ari.el6.x86_64.rpm
cray-flexnet-installation-instructions-11.12.1.txt
cray-flexnet-manager-11.12.1-1.0000.9450.6.1.ari.el6.x86_64.rpm
cray-flexnet-publisher-switch-11.12.1-1.0000.9450.6.1.ari.el6.x86_64.rpm
cray-flexnet-utils-11.12.1-1.0000.9450.6.1.ari.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 ~]$ module avail

----- /opt/cray/craype/default/modulefiles -----
craype-abudhabi      craype-hugepages256M  craype-interlagos-cu  craype-sandybridge
craype-abudhabi-cu   craype-hugepages2M    craype-istanbul       craype-shanghai
craype-accel-host    craype-hugepages32M  craype-ivybridge      craype-target-compute_node
craype-accel-nvidia20  craype-hugepages4M   craype-mc12          craype-target-local_host
craype-accel-nvidia35  craype-hugepages512M  craype-mc8           craype-target-native
craype-barcelona     craype-hugepages64M  craype-network-aries  craype-xeon
craype-haswell       craype-hugepages8M   craype-network-gemini
craype-hugepages128M  craype-intel-knc     craype-network-infiniband
craype-hugepages16M  craype-interlagos    craype-network-none

----- /opt/cray/modulefiles -----
cray-libsci/13.1.0(default)  mvapich2_cce/2.0.1.3(default)  perftools-lite/6.2.5(default)
cray-libsci_acc/3.2.0(default)  mvapich2_gnu/2.0.1.3          PrgEnv-cray/1.0.0(default)
craype/2.3.0(default)         papi/5.4.1.1(default)
craypkg-gen/1.1.2(default)     perftools/6.2.5(default)
----- /opt/modulefiles -----
cce/8.3.14(default)          cudatoolkit/6.5.14          gcc/4.8.1(default)
```

Part 2: For the compute/login nodes

Apply below changes to compute nodes

1. Install Libelf.so.0

Libelf.so.0 need 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 included spec files. Recommended to build RPM in the non-root account.

Required files:

1. mvapich2-2.0.1-cray-83-1507.spec
2. mvapich2-2.0.1-gnu-48-1507.spec
3. mvapich2-2.0.1.tar.gz

Need below modules for building mvapich2 rpm with CCE. So if you don't have cudatoolkit go to [Step 9 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
```