

Stable Cray Support in EasyBuild 2.7

Petar Forai

Major Areas of Interest Specific to Cray Support

- Version pinning in Cray toolchain to achieve reproducible builds
- New toolchain naming scheme that is not specific to XC and XK machines
- Removing the need to externally setup the environment before EasyBuild runs & no more module purge during sanity checking step with more robust module files
- Removal of FFTW in toolchain
- Providing more powerful external module metadata support to deal with “strange” Cray shipped module files
- Getting rid of smaller issues related to Python and Cray MPI and EB interactions
- Toolchain options now default to always build dynamic/shared executables and libraries

Change of Toolchain Name

- Now named named `Cray{Gnu|Intel|CCE}-<PE-Release-Date>` instead of old `CrayGNU-5.2.40` based on `PrgEnv-gnu/5.2.40`
- Name is now based on PE release document instead of PrgEnv module file

```
easyblock = 'CrayToolchain'

name = 'CrayGNU'
version = '2015.11'

homepage = 'http://docs.cray.com/books/S-9407-1511'
description = """Toolchain using Cray compiler wrapper, using PrgEnv-gnu module (PE release: November 2015).\n"""

toolchain = {'name': 'dummy', 'version': 'dummy'}

[...]
```

How are TC component versions picked?

- Go to <http://docs.cray.com/books/S-9408-1603/#versions> for XC and <http://docs.cray.com/books/S-9407-1603/#versions> for XE/XK
- Seems to be fine with XC vs XE/XK Systems
- TC name would be CrayGNU-<2nd Column>
- PE release contained updated products picked by consensus of the community
- Example: PE 15.11 contains GCC-4.9 and GCC-5.1

Publication Number	Release Date	Supported Software Releases
S-9407-1602	Feb 2016	Release Announcements 1.0
S-9407-1601	Jan 2016	Release Announcements 1.0
S-9407-1512	Dec 2015	Release Announcements 1.0
S-9407-1511	Nov 2015	Release Announcements 1.0
S-9407-1509	Sep 2015	Release Announcements 1.0
S-9407-1508	Aug 2015	Release Announcements 1.0
S-9407-1507	Jul 2015	Release Announcements 1.0
S-9407-1506	Jun 2015	Release Announcements 1.0
S-9407-1505	May 2015	Release Announcements 1.0
S-9407-1504	Apr 2015	Release Announcements 1.0
S-9407-1503	Mar 2015	Release Announcements 1.0

Version Pinning

- PrgEnv module file loads module default compiler
- Loads module default libsci
- Loads machine specific lower level system libs and Cray addons
- MPI not included
- Those issues are addressed now through version pinning toolchain definitions
- Provides most important pinning now (ABI for C++/Fortran, BLAS numerics and MPI)

```
[...]  
dependencies = [  
  ('PrgEnv-gnu', EXTERNAL_MODULE),  
  ('gcc/4.9.3', EXTERNAL_MODULE),  
  ('cray-libsci/13.2.0', EXTERNAL_MODULE),  
  ('cray-mpich/7.2.6', EXTERNAL_MODULE),  
]  
  
moduleclass = 'toolchain'
```

```
/opt/cray/modulefiles/PrgEnv-gnu/5.2.82:  
  
conflict      PrgEnv  
conflict      PrgEnv-x1  
conflict      PrgEnv-x2  
conflict      PrgEnv-cray  
conflict      PrgEnv-intel  
conflict      PrgEnv-pgi  
conflict      PrgEnv-pathscale  
setenv        PE_ENV GNU  
setenv        XTOS_VERSION 5.2.82  
setenv        CRAYOS_VERSION 5.2.82  
prepend-path  PE_PRODUCT_LIST GNU:GCC  
module        load gcc  
module        load craype/2.4.0  
module        load totalview  
module        load cray-libsci  
module        load udreg/  
2.3.2-1.0502.10518.2.17.ari  
module        load ugni/  
6.0-1.0502.10863.8.29.ari  
module        load pmi  
module        load dmapp/  
7.0.1-1.0502.11080.8.76.ari  
module        load gni-headers/  
4.0-1.0502.10859.7.8.ari  
module        load xpmem/  
0.1-2.0502.64982.5.3.ari  
module        load dvs/  
2.5_0.9.0-1.0502.2188.1.116.ari  
module        load alps  
module        load rca/  
1.0.0-2.0502.60530.1.62.ari  
module        load atp  
setenv        CRAY_PRGENVGNU loaded
```

New Cray TC module files

- Due to the complex logic in Cray shipped module files the environment is fragile and sensitive to module purge & friends
- EasyBuild needed external setup of the environment (unload of unwanted **PrgEnv-*<flavour>***) - ie shell script to setup
- As of 2.7 load of the module file provides the same environment as EB operates in
- Could only build in one shell session for same **PrgEnv-*<flavour>***
- Builds of multiple pieces of software in the same shell session would break the environment as sanity checking would trigger module purge and restore the environment through module loads of the dependencies
- Sanity check step now restores initial starting environment without running purge before
- Removed FFTW from TC definition as it caused some issues with Python, use as external dependency on a case by case basis instead of always link in FFTW through the wrapper

```
conflict CrayGNU

module unload PrgEnv-cray
module unload PrgEnv-intel
module unload PrgEnv-pgi

if { ![ is-loaded PrgEnv-gnu/5.2.82 ] } {
    module load PrgEnv-gnu/5.2.82
}

if { [ is-loaded gcc ] } {
    module swap gcc gcc/4.9.3
} else {
    module load gcc/4.9.3
}

if { [ is-loaded cray-libsci ] } {
    module swap cray-libsci cray-libsci/13.2.0
} else {
    module load cray-libsci/13.2.0
}

if { [ is-loaded cray-mpich ] } {
    module swap cray-mpich cray-mpich/7.2.6
} else {
    module load cray-mpich/7.2.6
}

setenv    EBVERSIONCRAYGNU    "2015.11"
setenv    EBDEVELCRAYGNU     "$root/
easybuild/CrayGNU-2015.11-easybuild-devel"

# Built with EasyBuild version 2.7.0
```

Better Prefix in Meta Data Configuration for Cray Shipped Modules

- EasyBuild Release now ships a default configuration file that matches the list of required dependencies for all configs currently available for Cray based toolchains in `etc/cray_external_modules_metadata.cfg`
- Mechanism provided to deal with module files that have no environment variable the points to the root of the installation directory
- FFTW is an example of such a module

```
[fftw/3.3.4.5]
name = FFTW
version = 3.3.4.5
prefix = FFTW_INC/..
```

```
setenv FFTW_DIR /opt/cray/fftw/3.3.4.4/sandybridge/lib
setenv FFTW_INC /opt/cray/fftw/3.3.4.4/sandybridge/include
```

Default to building dynamic Executables and shared libraries

- Before 2.7 not setting toolchain opts resulted in using default wrapper settings (ie static builds)

```
[...]  
toolchainopts = {'pic': True, 'dynamic': True,}  
[...]
```

- Not required any more and will use dynamic by default, but 'pic' option should still be used where appropriate (see other toolchain configs)

Smaller Fixes

- Fixed issue caused by VSC-base (an EasyBuild dependency) that would trigger a call into the Cray MPI library without calling `mpi_init()` from MPI4Py
- EasyBuild could not run when Python built with EasyBuild was loaded as the logging framework would trip the MPI4Py import that failed

Further Outlook

- Address sanity checking and running tests in some way
- Ie executing and/or benchmarking build artefacts (ie numpy for `np.dot()` needs to run $<1s$) after the build for sanity checking
- Provide PGI based toolchain
- Check integration of `craype-pkggen` if build doesnt create `pkg-config .pc` files by itself