

Maintaining Large Software Stacks in a Cray Ecosystem with Gentoo Portage

Colin MacLean



Large Software Stacks

- Most modules are, at most, a handful of packages
- Currently available tools ill-suited for large comprehensive sets of packages on Cray
 - Most notably Python
- Need tools to build software distributions
 - Resolve complicated dependency trees
 - Periodic releases of collections of packages, not many individual modules
 - Avoid complex runtime dependencies
 - Avoid environment bloat

Use of Gentoo Prefix on Blue Waters

- Gentoo Prefix used to manage the Python stack on Blue Waters
 - Over 400 packages installed
 - Includes non-Python dependencies. Some system libraries too old.
 - Built against the optimized cray-mpich and libsci libraries
 - Both MPI and non-MPI versions with minimal redundancy
 - Most ebuids work without modification
 - 18 modified (mostly minor)
 - 6 new

Potential Package Managers

- Pip
 - Original scope was limited to a Python stack
 - Can't build non-Python dependencies
 - Can't tweak build environment on a per-package basis
- SWTools
 - HPC software build and install tool
 - Creating build script too complex for large dependency trees

Potential Package Managers

- EasyBuild
 - HPC software management tool
 - Basic level of dependency calculation
 - Acceptable for a handful of packages
 - Inadequate for hundreds of packages
 - Hard coded versions and configuration options in build files
 - Has package repository
- Anaconda
 - Python distribution
 - Primarily binary based, but can build from source
 - Hard to control build environment
 - Anaconda binaries using MPI need rebuild to use cray-mpich

Potential Package Managers

- **Prefixed Gentoo Portage**

- Gentoo Portage Alt project allows Gentoo GNU userland install on top of Linux/Unix hosts
- Dependency calculations based on range of compatible versions. Suitable for rolling periodic distributions.
- Portage USE flags
 - Allow for fine-grained control of configuration options
 - Used in dependency calculations
- SLOTS
 - Feature allows for simultaneous install of multiple versions of some packages

- SUBSLOTS
 - Feature used to trigger required rebuilds on package updates
- Ebuild files
 - Bash scripts with access to powerful eclass APIs
 - Eclasses can provide default build procedures that may also be heavily modified if necessary
- Repositories with tens of thousands of ebuilds available
- Overlay repositories allow for customized ebuilds
- Consistent default build environment. Per-package environment customization.

Package Manager Selection

- Gentoo Prefix chosen
 - Make Environment Modules aware
 - Easier than significantly improving package management features
 - Need to install more than just Python packages

Initial state of Gentoo Prefix

- Bootstrapping script self-sufficient
 - Bootstrapped its own gcc/binutils
 - Host programs not available in prefix environment
 - Host MPI only major library visible to prefix (through sys-cluster/native-mpi package)
- After bootstrap, environment still constrained to prefix
- Unaware of Environment Modules framework

Compiler Environment for Gentoo Prefix

- Unpatch Gentoo Prefix uses GCC compiled with --sysroot
 - Need to use host compilers
 - --sysroot won't work
- Add prefix directories via -I and -L options to compiler flags
 - Breaks some packages
- Set CPATH and LIBRARY_PATH
 - Searched after -I and -L like system paths
- Provide compiler a modified “specs” file

Step 1: Modify Bootstrapping Script

- Modify environment to allow Prefix to see host compilers and libraries
- Change header and library search method
 - Generate CPATH/LIBRARY_PATH for host compiler.
 - Prefix had expected gcc built with --sysroot option.
- Generate rpaths for LDFLAGS
- Remove checks for host paths in environment
- Set up basic default Environment Modules

Step 2: Modify Prefix Environment

- Gentoo Prefix loads its environment from `$EPREFIX/etc/profile`
 - Fix paths to make host system visible to Prefix environment
 - Load default set of environment modules if Eselect Environment Modules not available
- Gentoo Portage executes in non-interactive non-login Bash shell
 - Source `$EPREFIX/etc/profile`
- Add Prefix paths to `C_PATH` and `LIBRARY_PATH` in `$EPREFIX/etc/env.d`
- Create Portage profile for Blue Waters
 - Default compiler flags, USE flags, forced “cray” USE flag, `package.provided` of host provided dependencies

Enhancement: Environment Modules Awareness

- Environment Modules module for Eselect
 - Sets default set of Environment Modules for the Prefix
 - Generates environment file \$EPREFIX/etc/env.d/01modules
 - Stores settings in \$EPREFIX/etc/envmod.conf
 - Regenerate environment with `eselect envmod update`
 - Ensures reproducible build environment

```
[bwpv-0.2.0] cmaclean@h2ologin3 ~ $ eselect envmod-PrgEnv list
[1]  PrgEnv-cray/4.1.40
[2]  PrgEnv-cray/4.2.15
[3]  PrgEnv-cray/4.2.24
[4]  PrgEnv-cray/4.2.34
[5]  PrgEnv-cray/5.2.14
[6]  PrgEnv-cray/5.2.40
[7]  PrgEnv-cray/5.2.82 (default)
[8]  PrgEnv-gnu/4.1.40
[9]  PrgEnv-gnu/4.2.15
[10] PrgEnv-gnu/4.2.24
[11] PrgEnv-gnu/4.2.34
[12] PrgEnv-gnu/5.2.14
[13] PrgEnv-gnu/5.2.40
[14] PrgEnv-gnu/5.2.82 (default) *
[15] PrgEnv-intel/4.1.40
[16] PrgEnv-intel/4.2.15
[17] PrgEnv-intel/4.2.24
[18] PrgEnv-intel/4.2.34
[19] PrgEnv-intel/5.2.14
[20] PrgEnv-intel/5.2.40
[21] PrgEnv-intel/5.2.82 (default)
[22] PrgEnv-pgi/4.1.40
[23] PrgEnv-pgi/4.2.15
[24] PrgEnv-pgi/4.2.24
[25] PrgEnv-pgi/4.2.34
[26] PrgEnv-pgi/5.2.14
[27] PrgEnv-pgi/5.2.40
[28] PrgEnv-pgi/5.2.82 (default)
```

Enhancement: Environment Modules Awareness

- Environment Modules in Portage

- Sometimes, an odd package may require a different build environment
- Patch Portage to load, unload, and swap modules based on environment variable
 - Use package.env
- Prevent build of packages known to break with certain module configurations
- Add module requirements for packages

- ENVMOD

- Load: ENVMOD="module"
- Unload: ENVMOD="-module"
- Swap: ENVMOD="%module1:module2"
- Space separated list

- ENVMOD_RESTRICT (ebuild)

- Space separated list of forbidden modules

- ENVMOD_REQUIRE (ebuild)

- Space separated list of required modules

Enhancement: Prefix Chaining

- Based on old patch to Portage
- Creates a child prefix
- Minimal setup
 - \$EPREFIX/etc/profile
 - \$EPREFIX/etc/portage/*
- Chain setup script
 - Generates etc/profile and etc/portage/* based off parent prefix
 - Optionally generates and installs environment module into parent prefix
- Portage uses parent prefixes for dependency resolution
 - Specify DEPEND, RDEPEND, HDEPEND, PDEPEND
 - Allows for a chained prefix with independent runtime dependencies
- Interaction with parent prefix entirely read-only

Enhancement: Prefix Chaining: Different Configurations

- Chained Prefixes can change any Portage build setting
 - USE flags
 - Compiler
 - Even CHOST, if desired
- Can have different default Environment Modules
- Can be used to support multiple BLAS choices
- On Blue Waters
 - BWPY: Built for login nodes. No MPI.
 - BWPY-MPI: h5py rebuilt with USE="mpi". Mpi4py added.
 - BWPY-Tensorflow: Required beta Google-Protobuf. Installed in chained prefix to keep stable BWPY.

Enhancement: Prefix Chaining: Multiuser

```
These are the packages that would be merged, in order:

Calculating dependencies... done!
[ebuild N ] sci-astronomy/erfa-1.1.0:0/1::gentoo_prefix USE="-static-libs" 0 KiB
[ebuild N ] dev-python/astropy-helpers-0.4.1::gentoo_prefix PYTHON_TARGETS="python2_7 python3_3 -python3_4" 0 KiB
[ebuild N ] sys-devel/flex-9999::BWGentooPrefix USE="nls -static {-test}" 0 KiB
[ebuild N ] dev-lang/fortran-4.4-r3::gentoo_prefix USE="-examples" 0 KiB
[ebuild N ] sci-libs/cfitsio-3.360:0/1::gentoo_prefix USE="fortran threads tools -doc -examples -static-libs" 0 KiB
[ebuild N ] sci-astronomy/wcslib-4.23:0/4::gentoo_prefix USE="fortran -doc -fits -pgplot -static-libs" 0 KiB
[ebuild N ] dev-python/astropy-0.4.4::gentoo_prefix USE="-doc {-test}" PYTHON_TARGETS="python2_7 python3_3 -python3_4" 0 KiB

Packages resolved from readonly installations:

[readonly DEP ] dev-python/numpy-1.9.2 from /u/staff/cmaclean/py (dev-python/numpy[python_targets_python2_7(-),python_targets_python3_3(-),-python_single_target_python2_7(-),-python_single_target_python3_3(-),-python_single_target_python3_4(-)] by dev-python/astropy-0.4.4)
[readonly DEP ] sys-libs/zlib-1.2.8-r1 from /u/staff/cmaclean/py (sys-libs/zlib:0= by dev-python/astropy-0.4.4)
[readonly DEP ] dev-python/cython-0.22 from /u/staff/cmaclean/py (dev-python/cython[python_targets_python2_7(-),python_targets_python3_3(-),-python_single_target_python2_7(-),-python_single_target_python3_3(-),-python_single_target_python3_4(-)] by dev-python/astropy-0.4.4)
[readonly DEP ] dev-python/setuptools-17.1.1 from /u/staff/cmaclean/py (dev-python/setuptools[python_targets_python2_7(-),python_targets_python3_3(-),-python_single_target_python2_7(-),-python_single_target_python3_3(-),-python_single_target_python3_4(-)] by dev-python/astropy-0.4.4)
[readonly DEP ] dev-lang/python-2.7.10 from /u/staff/cmaclean/py (>=dev-lang/python-2.7.5-r2:2.7 by dev-python/astropy-0.4.4)
[readonly DEP ] dev-lang/python-3.3.3 from /u/staff/cmaclean/py (>=dev-lang/python-3.3.2-r2:3.3 by dev-python/astropy-0.4.4)
[readonly DEP ] dev-lang/python-exec-2.0.2 from /u/staff/cmaclean/py (>=dev-lang/python-exec-2:=[python_targets_python2_7(-),python_targets_python3_3(-),-python_single_target_python2_7(-),-python_single_target_python3_3(-),-python_single_target_python3_4(-)] by dev-python/astropy-0.4.4)
[readonly DEP ] sys-devel/m4-9999 from /u/staff/cmaclean/py (sys-devel/m4 by sys-devel/flex-9999)
[readonly DEP ] app-arch/xz-utils-5.2.1-r1 from /u/staff/cmaclean/py (app-arch/xz-utils by sys-devel/flex-9999)
[readonly DEP ] sys-libs/zlib-1.2.8-r1 from /u/staff/cmaclean/py (sys-libs/zlib by sci-libs/cfitsio-3.360)
[readonly RDEP ] virtual/fortran-0 from /u/staff/cmaclean/py (=virtual/fortran-0 by sci-libs/cfitsio-3.360)
[readonly RDEP ] virtual/fortran-0 from /u/staff/cmaclean/py (virtual/fortran by sci-libs/cfitsio-3.360)
[readonly DEP ] virtual/pkgconfig-0-r1 from /u/staff/cmaclean/py (=virtual/pkgconfig-0-r1 by sci-astronomy/wcslib-4.23)
[readonly DEP ] virtual/pkgconfig-0-r1 from /u/staff/cmaclean/py (virtual/pkgconfig by sci-astronomy/wcslib-4.23)

Total: 7 packages (7 new), Size of downloads: 0 KiB

Would you like to merge these packages? [Yes/No] █
```

- Multiuser: Create chained prefixes in home directories
- Users can build specific versions and configurations of software using ebuilds (ex: PETSc)

Patching ebuilds

- Add “cray” to ebuild’s IUSE variable
- Put Cray platform-specific modifications in “if use cray” block
- Don’t break ebuild for non-Cray platforms

```
36 pkg_setup() {  
37     if use mpi; then  
38         if use cray; then  
39             export CRAY_ADD_RPATH=yes  
40             export CRAYPE_LINK_TYPE=dynamic  
41             export CC=cc  
42         else  
43             export CC=mpicc  
44         fi  
45     fi  
46 }
```

Patched h5py ebuild

