DAAC 2018

Current Status of the OpenHPC Project

Karl W. Schulz, Ph.D.
Institute for Computational Engineering and Sciences
and Dell Medical School

TEXAS
The University of Texas at Austin

The 2nd Industry/University Joint International Workshop on Data Center Automation, Analytics, and Control (DAAC)
SC'18 • November 12, 2018 • Dallas, TX

openHPC
https://openhpc.community

Outline

- What is OpenHPC: project overview
  - mission/vision
  - members, governance
  - key takeaways
- More technical details
  - packaging conventions
  - build/test environment
- Highlights from latest release

What is the project's mission and vision?

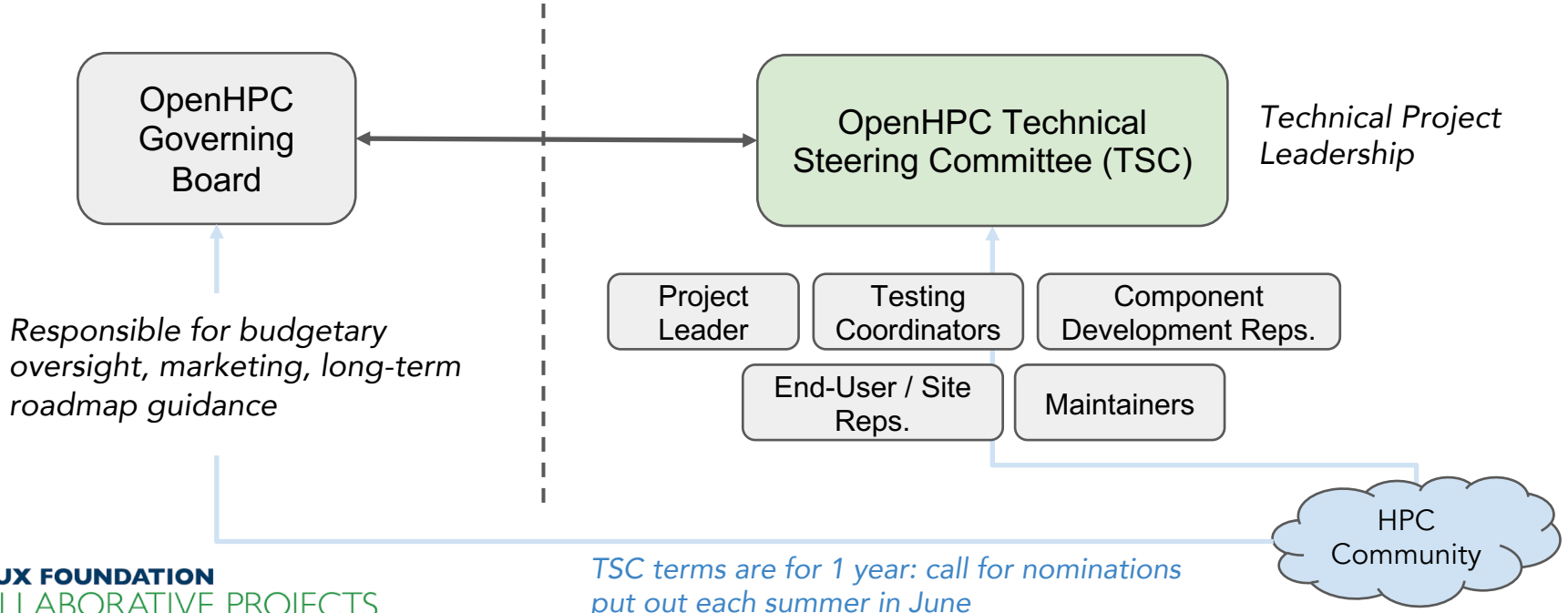
Mission: to provide a reference collection of **open-source HPC software** components and best practices, lowering barriers to deployment, advancement, and use of modern HPC methods and tools.

Vision: OpenHPC components and best practices will enable and accelerate innovation and discoveries by broadening access to state-of-the-art, open-source HPC methods and tools in a consistent environment, supported by a collaborative, worldwide community of HPC users, developers, researchers, administrators, and vendors.

OpenHPC: Project Members

How is the community project governed?

Governance is dual-pronged with a Governing Board + Technical Steering Committee



LINUX FOUNDATION COLLABORATIVE PROJECTS

OpenHPC TSC – Individual Members

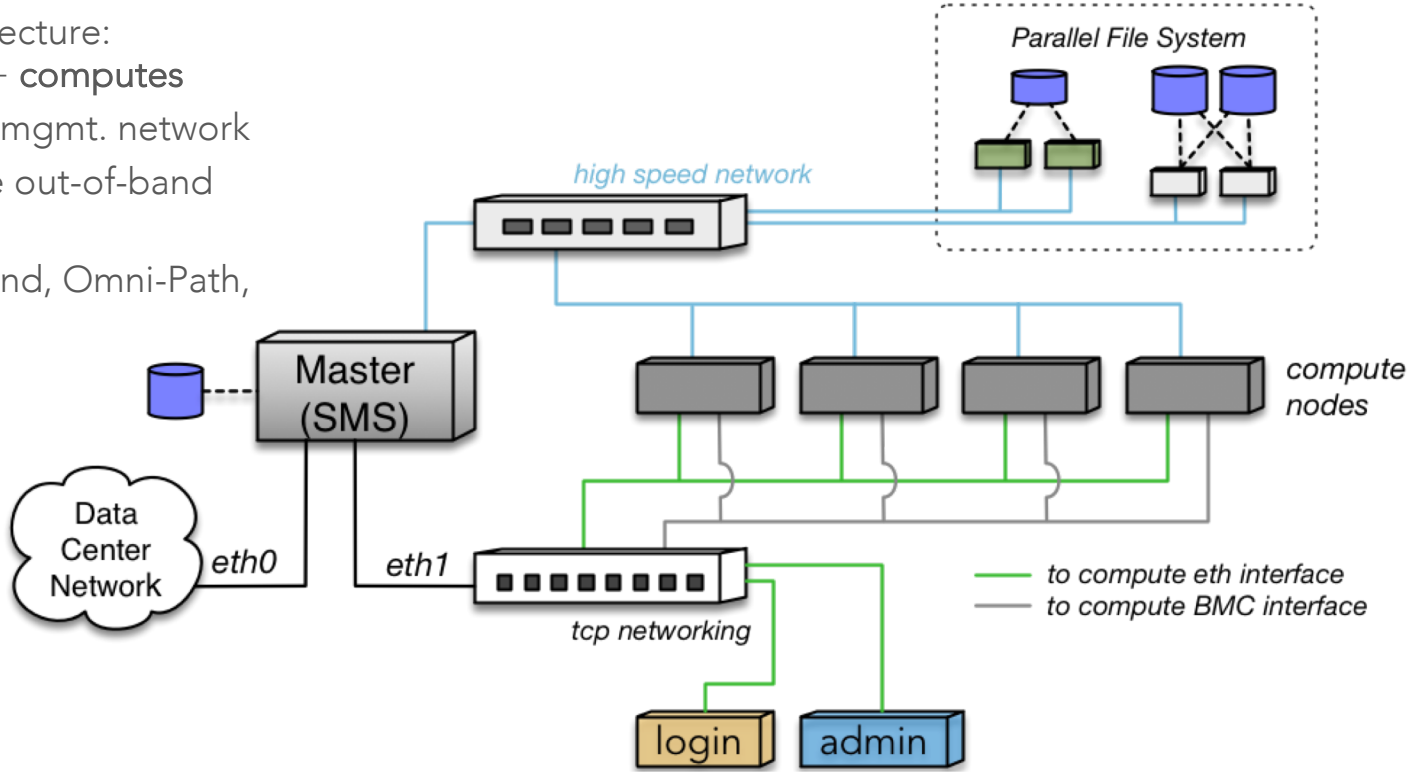
- Reese Baird, Intel (Maintainer)
- David Brayford, LRZ (Maintainer)
- Eric Coulter, Indiana University (End-User/Site Representative)
- Chris Downing, Red Oak Consulting (Maintainer)
- Craig Gardner, SUSE (Maintainer)
- Renato Golin, Linaro (Testing Coordinator)
- Michael Karo, Altair (Maintainer)
- Janet Lebens, Cray (Maintainer)
- Thomas Moschny, ParTec (Maintainer)
- Takayuki Okamoto, Fujitsu (Maintainer)
- Kevin Pedretti, Sandia National Laboratory (Maintainer)
- Paul Peltz, Los Alamos National Laboratory (Maintainer)
- Nam Pho, Harvard Medical School (Maintainer)
- Cyrus Proctor, Texas Advanced Computing Center (Maintainer)
- Adrian Reber, Red Hat (Maintainer)
- Karl W. Schulz, UT Austin (Project Lead, Testing Coordinator)
- Jeff Schutkoske, Cray (Component Development Representative)
- Derek Simmel, Pittsburgh Supercomputing Center (End-User/Site Representative)
- Chris Simmons, UT Dallas (Maintainer)
- Nirmala Sundararajan, Dell (Maintainer)

New for 2018-2019

perhaps someone from this workshop might be interested in volunteering next year?

Target System Architecture

- Basic cluster architecture:
  head node (SMS) + **computes**
- Ethernet fabric for mgmt. network
- Shared or separate out-of-band (BMC) network
- MPI fabric (InfiniBand, Omni-Path, or ethernet-only)



4 Key Takeaways…

OpenHPC: a building block repository

[Key takeaway]

- OpenHPC provides a collection of pre-built ingredients common in HPC environments; fundamentally it is a software repository
- The repository is published for use with Linux distro package managers:
  - yum (CentOS/RHEL)
  - zypper (SLES)

- You can pick relevant bits of interest for your site
  - if you prefer a resource manager that is not included, you can build that locally and still leverage the scientific libraries and development environment
  - similarly, you might prefer to utilize a different provisioning system
  - *public package repositories also makes it easy to include desired elements in customized containers (e.g. Docker, Singularity, Charliecloud)*

Repository Enablement

- Public repositories are available at: http://build.openhpc.community/OpenHPC:/

- There are two primary ways to access the available RPMs
  - enable public OpenHPC repo(s) on head node that can route to internet
  - host a local mirror of the OpenHPC repo(s) within your datacenter (you can download a tarball with entire repo, or rync locally)

- To facilitate the remote network access option, we provide an "ohpc-release" RPM
  - downloadable from front page of GitHub site (https://github.com/openhpc/ohpc)
  - here is an example install for CentOS/x86 and latest v1.3.x release:

```
# export OHPC_GITHUB=https://github.com/openhpc/ohpc/releases/download
# yum -y install $OHPC_GITHUB/v1.3.GA/ohpc-release-1.3-1.el7.x86_64.rpm
```

Repository Enablement (external network access)

- Note that ohpc-release configures two repos:

```
# yum repolist
Loaded plugins: langpacks
repo id              repo name                                          status
OpenHPC              OpenHPC-1.3 - Base                                    821
OpenHPC-updates      OpenHPC-1.3 - Updates                                 686
base                 CentOS-7 - Base                                     9,363
epel                 Extra Packages for Enterprise Linux 7 - x86_64     11,854
```

- With the initial release of a minor branch (e.g. v1.3), the [Updates] repo will be empty
- As subsequent micro releases are available (e.g. v1.3.1, v1.3.2, etc), the [Updates] is populated to match latest release

OpenHPC: A Hierarchical HPC Software Environment

[Key takeaway]

- Designation of compiler/MPI variants is **fundamental** to OpenHPC's build, test, and end-user environment
- OpenHPC packaging is architected to allow multiple variants to be added over time
- example: two gcc variants are installed

```
$ module list
Currently Loaded Modules:
  1) autotools   2) prun/1.2   3) gnu7/7.3.0   4) mvapich2/2.2   5) ohpc

$ module avail

---------------------------------- /opt/ohpc/pub/moduledeps/gnu7-mvapich2 ----------------------------------
   adios/1.13.1    mpiP/3.4.1             phdf5/1.10.2        py3-scipy/1.1.0    superlu_dist/5.3.0
   boost/1.67.0    mumps/5.1.2            pnetcdf/1.9.0       scalapack/2.0.2    tau/2.27.1
   fftw/3.3.7      netcdf-cxx/4.3.0       ptscotch/6.0.4      scalasca/2.3.1     trilinos/12.12.1
   hypre/2.14.0    netcdf-fortran/4.4.4   py2-mpi4py/3.0.0    scorep/4.0
   imb/2018.1      netcdf/4.6.1           py2-scipy/1.1.0     sionlib/1.7.1
   mfem/3.3.2      petsc/3.9.1            py3-mpi4py/3.0.0    slepc/3.9.1                    [ MPI tier ]

------------------------------------- /opt/ohpc/pub/moduledeps/gnu7 -------------------------------------
   R/3.5.0         likwid/4.3.2    mvapich2/2.2    (L)    openmpi3/3.1.0    py2-numpy/1.14.3    superlu/5.2.1
   gsl/2.4         metis/5.1.0     ocr/1.0.1              pdtoolkit/3.25    py3-numpy/1.14.3
   hdf5/1.10.2     mpich/3.2.1     openblas/0.2.20        plasma/2.8.0      scotch/6.0.4        [ compiler tier ]


----------------------------------- /opt/ohpc/pub/modulefiles -----------------------------------
   EasyBuild/3.6.1           clustershell/1.8       gnu7/7.3.0      (L)    ohpc          (L)    prun/1.2    (L)
   autotools          (L)    cmake/3.11.1           hwloc/1.11.10          papi/5.6.0           singularity/2.5.1
   charliecloud/0.2.4        gnu/5.4.0              llvm5/5.0.1            pmix/2.1.1           valgrind/3.13.0
```

Quick (hierarchical) installation example

- Once an OpenHPC repo is enabled, package installation follows standard package manager semantics
- Package interdependencies maintained within RPMs
- Consider example install of PETSc for gnu7/mvapich2 on bare bones head node

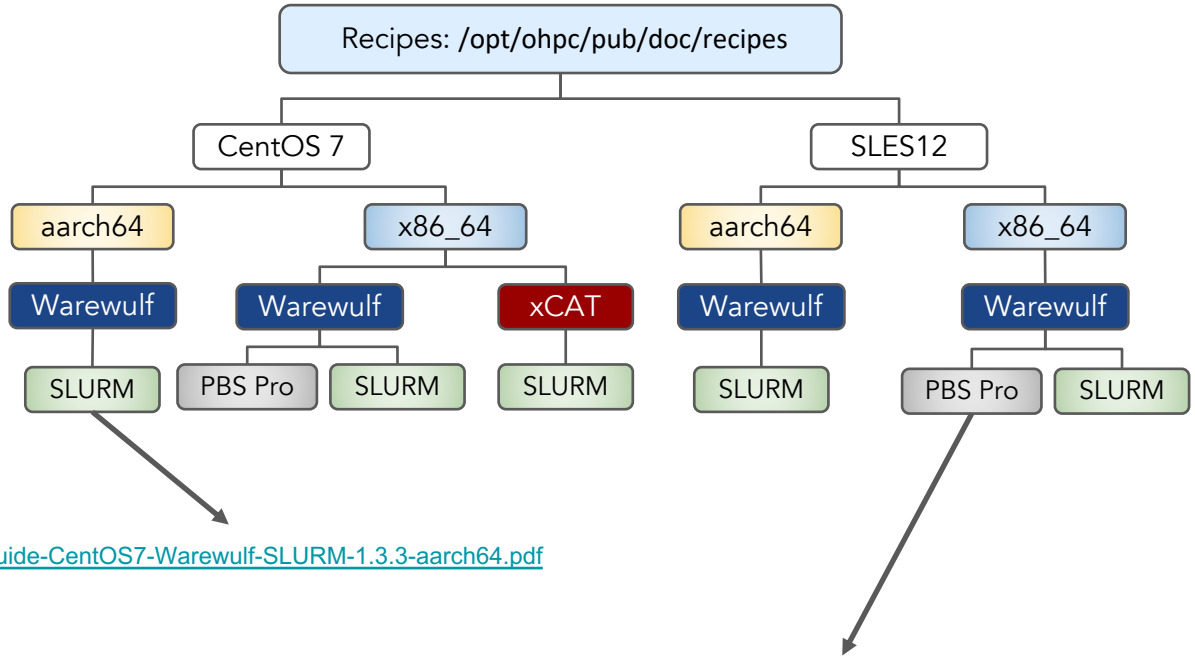
```
# yum install petsc-gnu7-mvapich2-ohpc
…
================================================================================
 Package                        Arch         Version           Repository      Size
================================================================================
Installing:
 petsc-gnu7-mvapich2-ohpc       x86_64       3.7.6-104.1       OpenHPC-updates 13 M
Installing for dependencies:
 gnu7-compilers-ohpc            x86_64       7.1.0-29.3        OpenHPC-updates 51 M
 mvapich2-gnu7-ohpc             x86_64       2.2-26.2          OpenHPC-updates 8.6 M
 openblas-gnu7-ohpc             x86_64       0.2.19-25.1       OpenHPC-updates 3.4 M
 phdf5-gnu7-mvapich2-ohpc       x86_64       1.10.0-83.1       OpenHPC-updates 2.9 M
 prun-ohpc                      noarch       1.1-21.1          OpenHPC         9.5 k
 scalapack-gnu7-mvapich2-ohpc   x86_64       2.0.2-39.1        OpenHPC-updates 3.4 M

Transaction Summary
================================================================================
```

OpenHPC: validated recipes for system install

[Key takeaway]

- In addition to being a package repository, OpenHPC provides validated recipes for bare-metal system installs
- Recipes organized by OS, architecture, and key administrative components
- Latest guides always available on main GitHub site:

  https://github.com/openhpc/ohpc



Install_guide-CentOS7-Warewulf-SLURM-1.3.3-aarch64.pdf

Install_guide-SLE_12-Warewulf-PBSPro-1.3.3-x86_64.pdf

OpenHPC: Multiple Architecture Support

x86_64

```
[train01@sms001 ~]$ module avail

--------------------- /opt/ohpc/pub/moduledeps/gnu7-mpich ---------------------
   adios/1.11.0    mpiP/3.4.1             petsc/3.7.6        scorep/3.0
   boost/1.63.0    mumps/5.1.1            phdf5/1.10.0       sionlib/1.7.1
   fftw/3.3.6      netcdf-cxx/4.3.0       scalapack/2.0.2    superlu_dist/4.2
   hypre/2.11.1    netcdf-fortran/4.4.4   scalasca/2.3.1     tau/2.26.1
   imb/4.1         netcdf/4.4.1.1         scipy/0.19.0       trilinos/12.10.1

----------------------- /opt/ohpc/pub/moduledeps/gnu7 -----------------------
   R/3.3.3        metis/5.1.0         numpy/1.12.1       openmpi/1.10.7
   gsl/2.3        mpich/3.2    (L)    ocr/1.0.1          pdtoolkit/3.23
   hdf5/1.10.0    mvapich2/2.2        openblas/0.2.19    superlu/5.2.1

-------------------------
   EasyBuild/3.2.1
   autotools          (L)
```

aarch64

```
train01@cavium1:~> module avail

--------------------- /opt/ohpc/pub/moduledeps/gnu7-mpich ---------------------
   adios/1.11.0    mpiP/3.4.1             petsc/3.7.6        scorep/3.0
   boost/1.63.0    mumps/5.1.1            phdf5/1.10.0       sionlib/1.7.1
   fftw/3.3.6      netcdf-cxx/4.3.0       scalapack/2.0.2    superlu_dist/4.2
   hypre/2.11.1    netcdf-fortran/4.4.4   scalasca/2.3.1     tau/2.26.1
   imb/4.1         netcdf/4.4.1.1         scipy/0.19.0       trilinos/12.10.1

----------------------- /opt/ohpc/pub/moduledeps/gnu7 -----------------------
   R/3.3.3        metis/5.1.0         ocr/1.0.1          pdtoolkit/3.23
   gsl/2.3        mpich/3.2    (L)    openblas/0.2.19    superlu/5.2.1
   hdf5/1.10.0    numpy/1.12.1        openmpi/1.10.7

------------------------- /opt/ohpc/pub/modulefiles -------------------------
   EasyBuild/3.2.1          hwloc/1.11.6       singularity/2.3
   autotools          (L)   ohpc         (L)   valgrind/3.12.0
```

[Key takeaway]

- aarch64 and x86_64 package repositories are available
- benefit is that OpenHPC provides consistent development environment to the end user across multiple architectures

Additional Technical Details…

More comments on:

- packaging conventions
- infrastructure
- build and test environment

Packaging Conventions

- OpenHPC tries to play nicely with underlying OS'es and other repositories
- We endeavor to avoid namespace conflict in a variety of ways:
  - install paths: end-user oriented components housed under /opt/ohpc/pub (also allow for multiple versions to coexist)
  - package names: RPMs provided via OpenHPC repos include "**-ohpc**" as the suffix for the package name, for example:

    lmod-**ohpc**-7.4.8-11.1.aarch64.rpm

  - dependencies: compiler/MPI variant dependencies are **explicitly** managed

```
%if "%{compiler_family}" == "gnu7"
BuildRequires: gnu7-compilers%{PROJ_DELIM} >= 7.2.0
Requires:      gnu7-compilers%{PROJ_DELIM} >= 7.2.0
%endif
%if "%{mpi_family}" == "mvapich2"
BuildRequires: mvapich2-%{compiler_family}%{PROJ_DELIM}
Requires:      mvapich2-%{compiler_family}%{PROJ_DELIM}
%endif
```

[relevant logic from OHPC_macros]

Packaging conventions: custom ohpc RPM plugin

To avoid namespace collision for resolving dynamic libraries (.so's), we apply an "ohpc" color delimiter to libraries installed in /opt/ohpc/ path

```
# rpm -q --requires gnu7-compilers-ohpc | egrep "libc.so|libgc"
```

default RPM dependency analysis

OHPC 1.3.1 RPM

```
libgcc_s.so.1()(64bit)
libgcc_s.so.1(GCC_3.0)(64bit)
libgcc_s.so.1(GCC_3.3)(64bit)
libgcc_s.so.1(GCC_4.2.0)(64bit)
libgcc_s.so.1(GCC_4.3.0)(64bit)

libc.so.6()(64bit)
libc.so.6(GLIBC_2.10)(64bit)
libc.so.6(GLIBC_2.11)(64bit)
libc.so.6(GLIBC_2.14)(64bit)
libc.so.6(GLIBC_2.16)(64bit)
libc.so.6(GLIBC_2.17)(64bit)
libc.so.6(GLIBC_2.2.5)(64bit)
libc.so.6(GLIBC_2.3)(64bit)
libc.so.6(GLIBC_2.3.2)(64bit)
libc.so.6(GLIBC_2.3.3)(64bit)
libc.so.6(GLIBC_2.6)(64bit)
libc.so.6(GLIBC_2.7)(64bit)
```

.so's contained within ohpc gcc build

.so's required from base OS supplied packages

Build with Updated Approach

```
libgcc_s.so.1()(64bit)(ohpc)
libgcc_s.so.1(GCC_3.0)(64bit)(ohpc)
libgcc_s.so.1(GCC_3.3)(64bit)(ohpc)
libgcc_s.so.1(GCC_4.2.0)(64bit)(ohpc)
libgcc_s.so.1(GCC_4.3.0)(64bit)(ohpc)

libc.so.6()(64bit)
libc.so.6(GLIBC_2.10)(64bit)
libc.so.6(GLIBC_2.11)(64bit)
libc.so.6(GLIBC_2.14)(64bit)
libc.so.6(GLIBC_2.16)(64bit)
libc.so.6(GLIBC_2.17)(64bit)
libc.so.6(GLIBC_2.2.5)(64bit)
libc.so.6(GLIBC_2.3)(64bit)
libc.so.6(GLIBC_2.3.2)(64bit)
libc.so.6(GLIBC_2.3.3)(64bit)
libc.so.6(GLIBC_2.6)(64bit)
libc.so.6(GLIBC_2.7)(64bit)
```

updated RPM dependency analysis using plugin

We introduced this new convention in v1.3.4 release

Packaging Conventions

- Packaging adopts a consistent environment variable schema in modulefiles (we rely on Lmod implementation of modules):
  - `${PACKAGE_DIR} - top level install path of package`
  - `${PACKAGE_BIN} - path to binaries supplied by package`
  - `${PACKAGE_INC} - path to include headers for package`
  - `${PACKAGE_LIB} - path to dynamic libraries for package`

- Recommend using these vars in build scripts, Makefiles, and interactive execution
- Let's consider an end user interaction example with previous PETSc install: assume we are a user wanting to build a PETSC hello world in C

```
$ module load gnu7 mvapich2
$ module load petsc
$ mpicc -I$PETSC_INC petsc_hello.c -L$PETSC_LIB -lpetsc
```

OpenHPC Development Infrastructure

What are we using to get the job done….?

The usual software engineering stuff:

- GitHub (SCM and issue tracking/planning)
- Continuous Integration (CI) Testing (Jenkins)
- Documentation (Latex)
- Bats (unit testing for Bash)

https://github.com/openhpc/ohpc

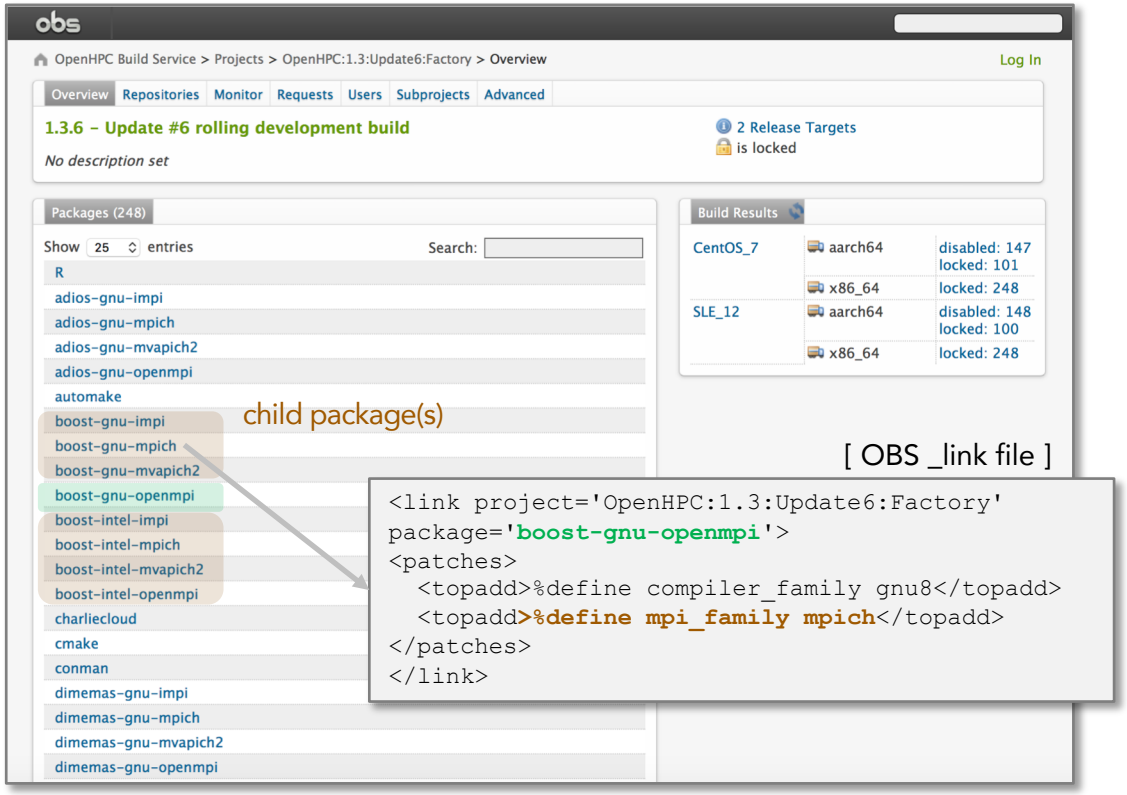
https://build.openhpc.community

Capable build/packaging system:

- Require flexible system to manage builds for multiple distros, multiple compiler/MPI family combinations, and dependencies across packages
- Have engineered a system using Open Build Service (OBS) which is supported by back-end git
  - git houses .spec files, tarballs, patches, documentation recipes, and integration tests
  - OBS performs automated builds and dependency analysis

Reproducible builds using OBS

https://build.openhpc.community

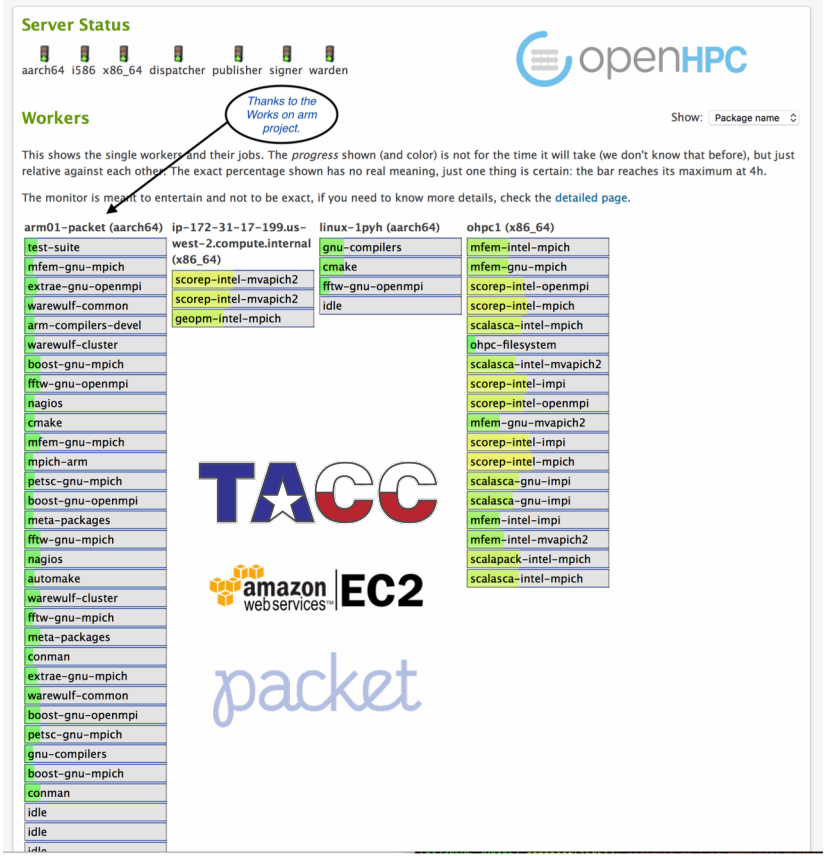


parent package

- We use the **Open Build Service** (OBS) to manage build processes
- OBS can drive builds for multiple OS's and architectures
- Repeatable builds carried out in chroot environment
- **Generates binary and src rpms**
- Client/server architecture supports distributed build slaves and multiple architectures
- OpenHPC adopts a **single** .spec file input for builds
- Dependencies are self-managed and designed to avoid namespace conflict with distro provided variants

Build system - distributed build servers

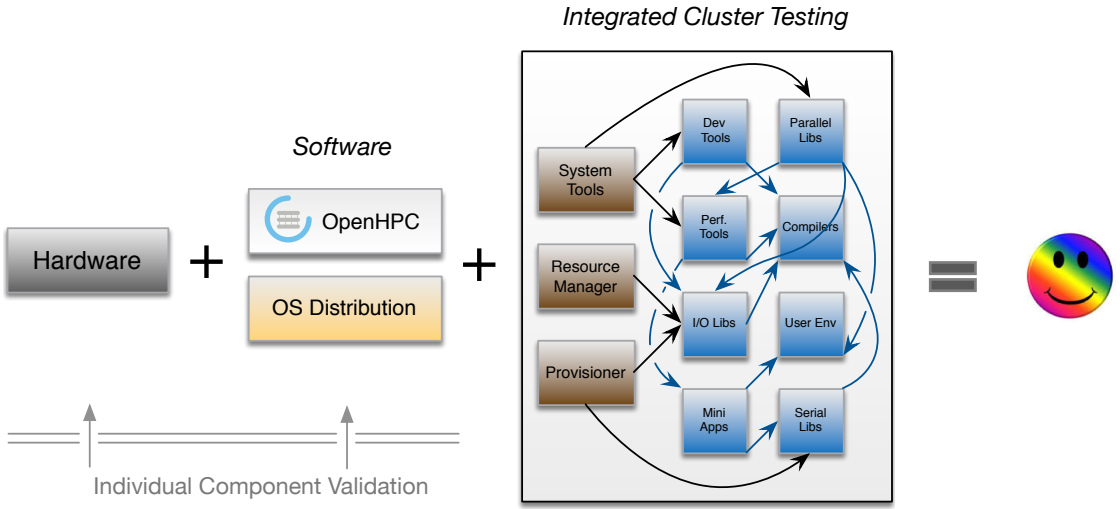
- We leverage ability in OBS to have distributed build servers:
  - x86_64: hosted at TACC (thanks to kind hardware donations) and on EC2
  - aarch64: hosted at Packet (thanks Works on Arm project)



Integration/Test/Validation

Testing is a key element for us and the intent is to build upon existing validation efforts and augment component-level validation with targeted cluster-validation and scaling initiatives including:

- install recipes
- cross-package interaction
- development environment
- mimic use cases common in HPC deployments
- upgrade mechanism



Post Install Integration Tests - Overview

Global testing harness includes a number of embedded subcomponents:

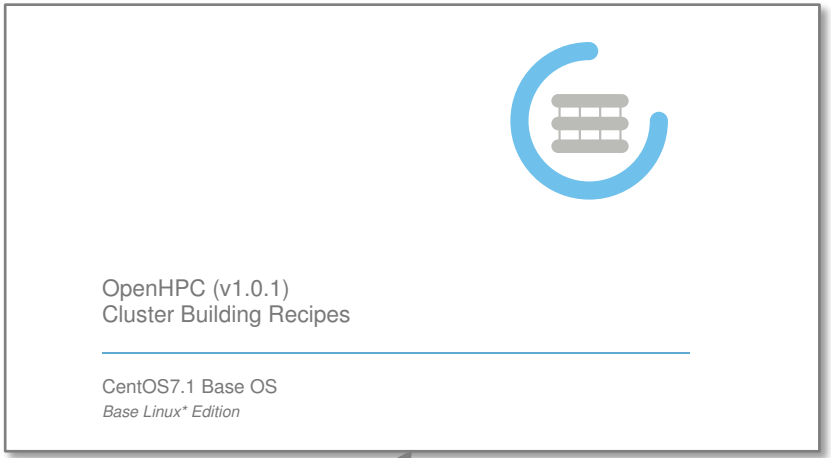
- major components have configuration options to enable/disable
  - end user tests need to touch all of the supported compiler and MPI families
  - we abstract this to repeat the tests with different compiler/MPI environments:
    - gcc/Intel compiler toolchains
    - Intel, OpenMPI, MPICH, MVAPICH2 MPI families

Example ./configure output (non-root)

```
Package version.............. : test-suite-1.0.0

Build user................... : jilluser
Build host................... : master4-centos71.localdomain
Configure date............... : 2015-10-26 09:23
Build architecture........... : x86_64-unknown-linux-gnu
Test suite configuration..... : long

Submodule Configuration:

User Environment:
    RMS test harness......... : enabled
    Munge.................... : enabled
    Apps..................... : enabled
    Compilers................ : enabled
    MPI...................... : enabled
    HSN...................... : enabled
    Modules.................. : enabled
    OOM...................... : enabled
Dev Tools:
    Valgrind................. : enabled
    R base package........... : enabled
    TBB...................... : enabled
    CILK..................... : enabled
Performance Tools:
    mpiP Profiler........ .... : enabled
    Papi..................... : enabled
    PETSc.................... : enabled
    TAU...................... : enabled
```

```
Libraries:
    Adios ................... : enabled
    Boost ................... : enabled
    Boost MPI................ : enabled
    FFTW..................... : enabled
    GSL...................... : enabled
    HDF5..................... : enabled
    HYPRE.................... : enabled
    IMB...................... : enabled
    Metis.................... : enabled
    MUMPS.................... : enabled
    NetCDF................... : enabled
    Numpy.................... : enabled
    OPENBLAS................. : enabled
    PETSc.................... : enabled
    PHDF5.................... : enabled
    ScaLAPACK................ : enabled
    Scipy.................... : enabled
    Superlu.................. : enabled
    Superlu_dist............. : enabled
    Trilinos ................ : enabled
Apps:
    MiniFE................... : enabled
    MiniDFT.................. : enabled
    HPCG..................... : enabled
    PRK...................... : enabled
```

Wider variety of recipes now available



Initially, we started off with a single recipe with the intent to expand.

10 recipes now available with v1.3.6 release

[we test all of these in a CI environment on bare-metal]

We continue to expand recipe option(s) with multiple resource managers, OSes, provisioners, and architectures:

x86_64:

- Install_guide-CentOS7-Warewulf-PBSPro-1.3.5-x86_64.pdf
- Install_guide-CentOS7-Warewulf-SLURM-1.3.5-x86_64.pdf
- Install_guide-CentOS7-xCAT-Stateful-SLURM-1.3.5-x86_64.pdf
- Install_guide-CentOS7-xCAT-Stateless-SLURM-1.3.5-x86_64.pdf
- Install_guide-SLE_12-Warewulf-PBSPro-1.3.5-x86_64.pdf
- Install_guide-SLE_12-Warewulf-SLURM-1.3.5-x86_64.pdf

aarch64:

- Install_guide-CentOS7-Warewulf-PBSPro-1.3.5-aarch64.pdf
- Install_guide-CentOS7-Warewulf-SLURM-1.3.5-aarch64.pdf
- Install_guide-SLE_12-Warewulf-PBSPro-1.3.5-aarch64.pdf
- Install_guide-SLE_12-Warewulf-SLURM-1.3.5-aarch64.pdf

Validating Installation Recipes

- To validate these documentation recipes, we cull instructions directly from the docs (LaTeX) to matching shell scripts -> recipe.sh
- Don't know all options in advance, so use simple input file to define MAC addresses, hostnames, etc
- For convenience, these template scripts are also provided with ohpc docs to aid installation or customization

```
# yum/zypper install docs-ohpc

# ls /opt/ohpc/pub/doc/recipes/*/*/*/*/recipe.sh
/opt/ohpc/pub/doc/recipes/centos7/aarch64/warewulf/slurm/recipe.sh
/opt/ohpc/pub/doc/recipes/centos7/x86_64/warewulf/pbspro/recipe.sh
/opt/ohpc/pub/doc/recipes/centos7/x86_64/warewulf/slurm/recipe.sh
/opt/ohpc/pub/doc/recipes/centos7/x86_64/xcat/slurm/recipe.sh
/opt/ohpc/pub/doc/recipes/sles12/aarch64/warewulf/slurm/recipe.sh
/opt/ohpc/pub/doc/recipes/sles12/x86_64/warewulf/pbspro/recipe.sh
/opt/ohpc/pub/doc/recipes/sles12/x86_64/warewulf/slurm/recipe.sh

# ls /opt/ohpc/pub/doc/recipes/*/input.local
/opt/ohpc/pub/doc/recipes/centos7/input.local
/opt/ohpc/pub/doc/recipes/sles12/input.local
```

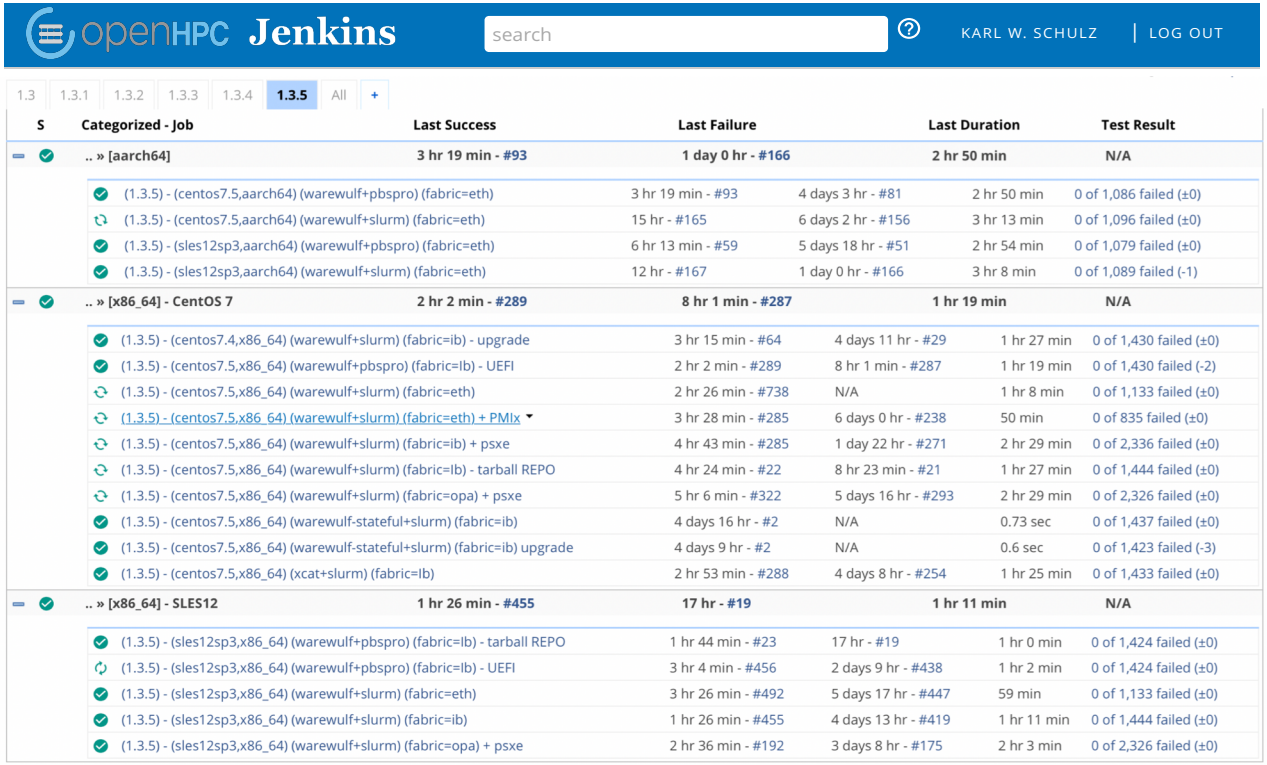
Local System Settings can be supplied to input file:

```
# compute hostnames
c_name[0]=c1
c_name[1]=c2
...
# compute node MAC addresses
c_mac[0]=00:1a:2b:3c:4f:56
c_mac[1]=00:1a:2b:3c:4f:56
...
```

input.local + recipe.sh == installed system (reproducible)

http://test.openhpc.community:8080

Community Jenkins Instance for CI Testing



S	Categorized - Job	Last Success	Last Failure	Last Duration	Test Result
	.. » [aarch64]	3 hr 19 min - #93	1 day 0 hr - #166	2 hr 50 min	N/A
	(1.3.5) - (centos7.5,aarch64) (warewulf+pbspro) (fabric=eth)	3 hr 19 min - #93	4 days 3 hr - #81	2 hr 50 min	0 of 1,086 failed (±0)
	(1.3.5) - (centos7.5,aarch64) (warewulf+slurm) (fabric=eth)	15 hr - #165	6 days 2 hr - #156	3 hr 13 min	0 of 1,096 failed (±0)
	(1.3.5) - (sles12sp3,aarch64) (warewulf+pbspro) (fabric=eth)	6 hr 13 min - #59	5 days 18 hr - #51	2 hr 54 min	0 of 1,079 failed (±0)
	(1.3.5) - (sles12sp3,aarch64) (warewulf+slurm) (fabric=eth)	12 hr - #167	1 day 0 hr - #166	3 hr 8 min	0 of 1,089 failed (-1)
	.. » [x86_64] - CentOS 7	2 hr 2 min - #289	8 hr 1 min - #287	1 hr 19 min	N/A
	(1.3.5) - (centos7.4,x86_64) (warewulf+slurm) (fabric=ib) - upgrade	3 hr 15 min - #64	4 days 11 hr - #29	1 hr 27 min	0 of 1,430 failed (±0)
	(1.3.5) - (centos7.5,x86_64) (warewulf+pbspro) (fabric=lb) - UEFI	2 hr 2 min - #289	8 hr 1 min - #287	1 hr 19 min	0 of 1,430 failed (-2)
	(1.3.5) - (centos7.5,x86_64) (warewulf+slurm) (fabric=eth)	2 hr 26 min - #738	N/A	1 hr 8 min	0 of 1,133 failed (±0)
	(1.3.5) - (centos7.5,x86_64) (warewulf+slurm) (fabric=eth) + PMIx	3 hr 28 min - #285	6 days 0 hr - #238	50 min	0 of 835 failed (±0)
	(1.3.5) - (centos7.5,x86_64) (warewulf+slurm) (fabric=ib) + psxe	4 hr 43 min - #285	1 day 22 hr - #271	2 hr 29 min	0 of 2,336 failed (±0)
	(1.3.5) - (centos7.5,x86_64) (warewulf+slurm) (fabric=lb) - tarball REPO	4 hr 24 min - #22	8 hr 23 min - #21	1 hr 27 min	0 of 1,444 failed (±0)
	(1.3.5) - (centos7.5,x86_64) (warewulf+slurm) (fabric=opa) + psxe	5 hr 6 min - #322	5 days 16 hr - #293	2 hr 29 min	0 of 2,326 failed (±0)
	(1.3.5) - (centos7.5,x86_64) (warewulf-stateful+slurm) (fabric=ib)	4 days 16 hr - #2	N/A	0.73 sec	0 of 1,437 failed (±0)
	(1.3.5) - (centos7.5,x86_64) (warewulf-stateful+slurm) (fabric=ib) upgrade	4 days 9 hr - #2	N/A	0.6 sec	0 of 1,423 failed (-3)
	(1.3.5) - (centos7.5,x86_64) (xcat+slurm) (fabric=lb)	2 hr 53 min - #288	4 days 8 hr - #254	1 hr 25 min	0 of 1,433 failed (±0)
	.. » [x86_64] - SLES12	1 hr 26 min - #455	17 hr - #19	1 hr 11 min	N/A
	(1.3.5) - (sles12sp3,x86_64) (warewulf+pbspro) (fabric=lb) - tarball REPO	1 hr 44 min - #23	17 hr - #19	1 hr 0 min	0 of 1,424 failed (±0)
	(1.3.5) - (sles12sp3,x86_64) (warewulf+pbspro) (fabric=lb) - UEFI	3 hr 4 min - #456	2 days 9 hr - #438	1 hr 2 min	0 of 1,424 failed (±0)
	(1.3.5) - (sles12sp3,x86_64) (warewulf+slurm) (fabric=eth)	3 hr 26 min - #492	5 days 17 hr - #447	59 min	0 of 1,133 failed (±0)
	(1.3.5) - (sles12sp3,x86_64) (warewulf+slurm) (fabric=ib)	1 hr 26 min - #455	4 days 13 hr - #419	1 hr 11 min	0 of 1,444 failed (±0)
	(1.3.5) - (sles12sp3,x86_64) (warewulf+slurm) (fabric=opa) + psxe	2 hr 36 min - #192	3 days 8 hr - #175	2 hr 3 min	0 of 2,326 failed (±0)

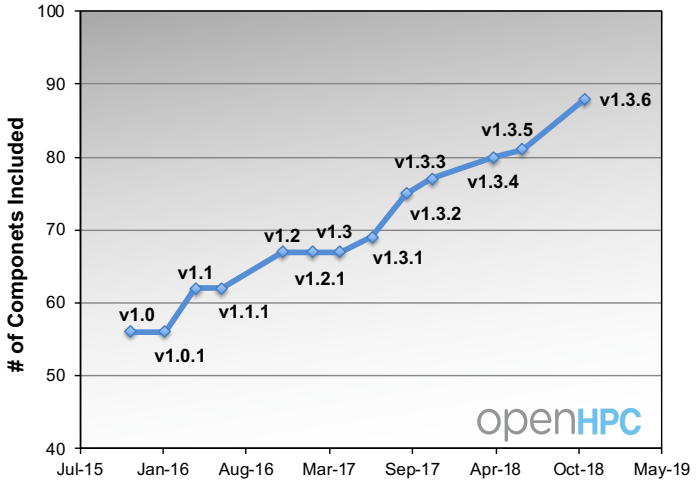
All recipes exercised in CI system (start w/ bare-metal installs + integration test suite)

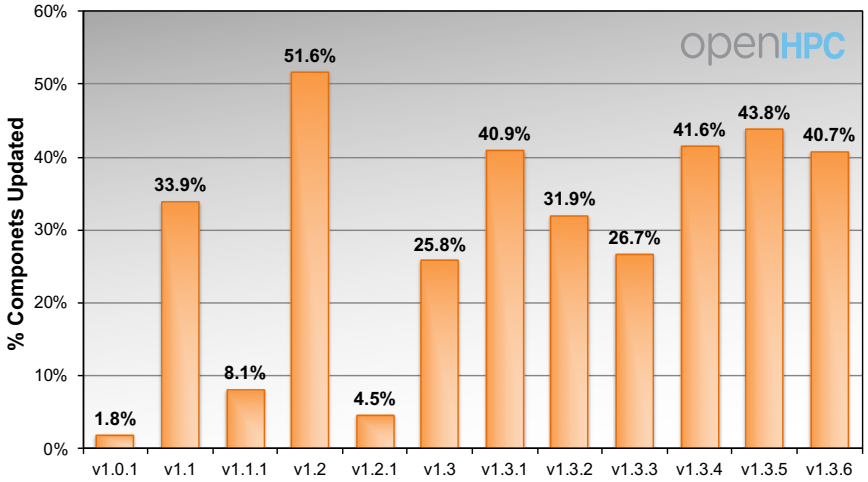
our CI "build" is really a cluster

Latest Release and Stats

Additions and Upstream Version Changes

- Part of the motivation for community effort like OpenHPC is the rapidity of S/W updates that occurs in our space
- We have been doing releases on a roughly quarterly cadence:
  - convention is to go with latest stable release of upstream components
  - additional components added over time





Released November 07, 2018

OpenHPC v1.3.6 - S/W components

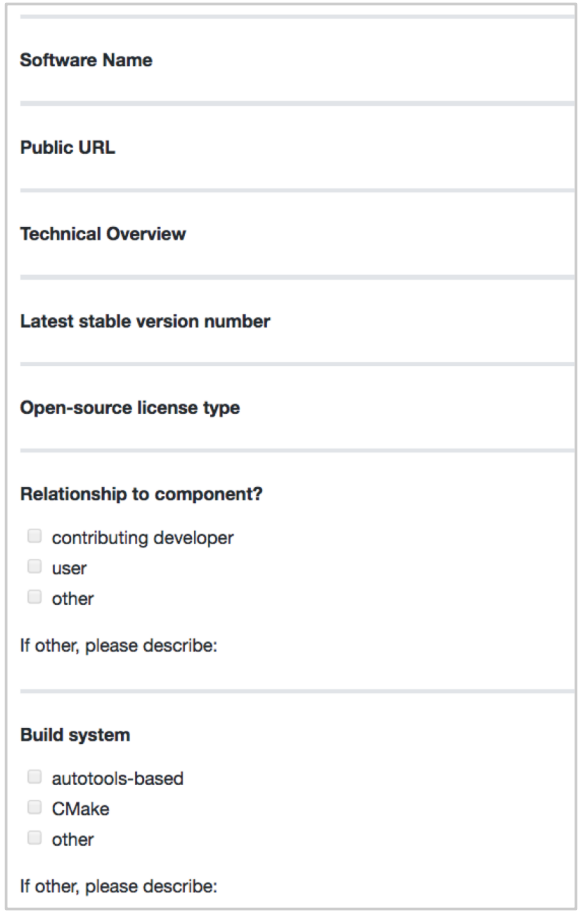
components available 88 | new additions 7 | updates 41%

Functional Areas	Components (new in v1.3.6 release)
Base OS	CentOS 7.5, SLES12 SP3
Architecture	aarch64, x86_64
Administrative Tools	Conman, Ganglia, Lmod, LosF, Nagios, NHC, pdsh, pdsh-mod-slurm, prun, EasyBuild, ClusterShell, mrsh, Genders, Shine, Spack, test-suite
Provisioning	Warewulf, xCAT
Resource Mgmt.	SLURM, Munge, PBS Professional, PMIx
Runtimes	Charliecloud, OpenMP, OCR, Singularity
I/O Services	Lustre client, BeeGFS client
Numerical/Scientific Libraries	Boost, GSL, FFTW, Hypre, Metis, MFEM, Mumps, OpenBLAS, OpenCoarrays, PETSc, PLASMA, Scalapack, Scotch, SLEPc, SuperLU, SuperLU_Dist, Trilinos
I/O Libraries	HDF5 (pHDF5), NetCDF/pNetCDF (including C++ and Fortran interfaces), Adios
Compiler Families	GNU (gcc, g++, gfortran), Clang/LLVM (introduced new gcc8 variant)
MPI Families	MVAPICH2, OpenMPI, MPICH
Development Tools	Autotools, cmake, hwloc, mpi4py, R, SciPy/NumPy, Valgrind
Performance Tools	PAPI, IMB, Likwid, mpiP, pdtoolkit TAU, Scalasca, ScoreP, SIONLib, GeoPM, msr-safe, Dimemas, Extrae, Paraver

- 3rd Party libraries are built for each compiler/MPI family
- Resulting repositories currently comprised of ~700 RPMs

How to Request Additional Software?

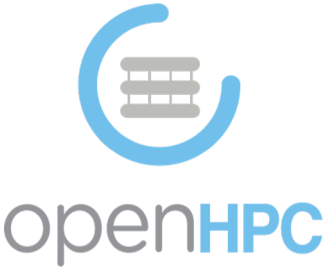
- We have a simple submission site for new requests:
  - https://github.com/openhpc/submissions
- Example components added via this mechanism since the v1.2. release (Nov' 16)
  - BeeGFS client
  - xCAT recipe
  - hwloc
  - Singularity
  - LLVM/clang
  - PLASMA
  - pNetCDF
  - SCOTCH
  - SLEPc
  - PMIx
  - MPI4py
  - Likwid
  - MFEM
  - NHC
  - Charliecloud
  - GeoPM
  - Dimemas/Extrae, Paraver
  - OpenCoarrays



Summary

- Provided an overview of the Linux Foundation project along with some technical updates over the last year:
  - 3 releases thus far this year
  - several new recipe additions
  - more component additions
  - packaging convention changes
- Other related items at SC'18 this year:
  - Birds of a Feather Session on Wed, November 14th (12:15-13:15)
  - happy hour event, Wed November 14th (17:00-19:00)

Thanks for your time….



- OpenHPC Home: http://www.openhpc.community/
- Primary GitHub Site: https://github.com/openhpc/ohpc
- Package Repositories: http://build.openhpc.community/OpenHPC:/
- Component Submission: https://github.com/openhpc/submissions
- System Registry: System Registration Form
- CI Infrastructure: http://test.openhpc.community:8080
- OpenHPC Wiki: https://github.com/openhpc/ohpc/wiki
  - includes links to overview paper and past presentations

- Mailing Lists: http://www.openhpc.community/support/mail-lists/
  - openhpc-announce
  - openhpc-users
  - openhpc-devel