



# OpenHPC Community BoF

Karl W. Schulz, Reese Baird, David Brayford, Chris Downing, Kevin Pedretti,  
Paul Peltz, Cyrus Proctor, Derek Simmel, Chris Simmons, Nirmala Sundararajan

OpenHPC Technical Steering Committee (TSC) Members

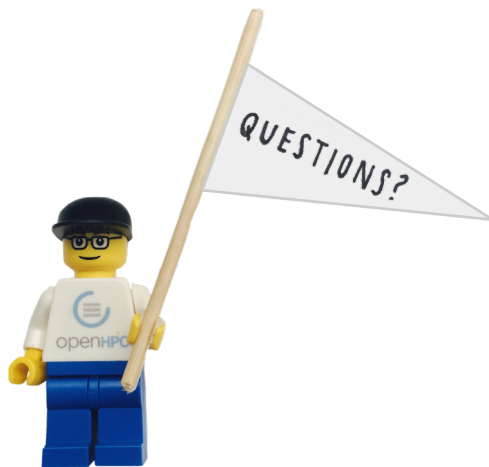
November 14, 2018 • Dallas, TX

# Outline

## Part I: Presentation (~30 min)

- What's New / Changes since last year
- Software in latest release
- Future work

## Part II: Open Forum (~30 min)



# Current Project Members



Indiana University



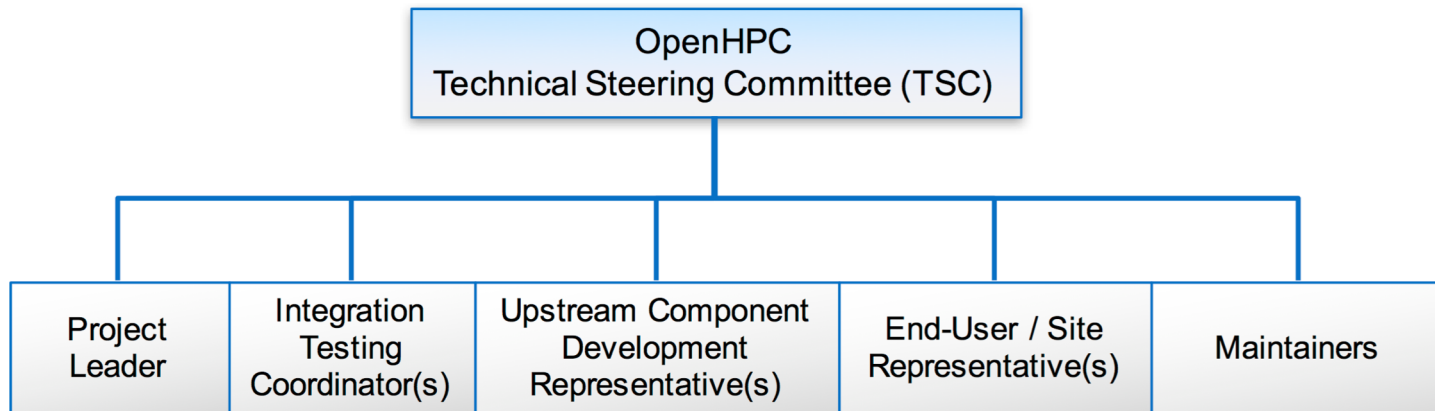
University of Cambridge



Mixture of academics, labs, and industry

# OpenHPC Technical Steering Committee (TSC)

## Role Overview



Note: we completed election of TSC members for the 2018-2019 term in August:

- terms are for 1 year

# OpenHPC TSC – Individual Members

## New members for 2018-2019

- Reese Baird, SpaceX (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)

Interested in participating next year?

- *expect call for nominations next summer (June/July)*

# Updates / New Items

*will next highlight some new items/changes since last year's BoF*

# rsync is now supported for OpenHPC repos

- At SC'17 BoF, we had a request asking if we could enable rsync for mirroring of OpenHPC repositories
- We have control over the back end, so rsync now supported for repos hosted at: <http://build.openhpc.community/OpenHPC/>
- Three methods now available for access:
  - direct repo access via yum/zypper (requires external routing)
  - local repo mirroring (download self-contained tarball with binary/src RPMs)
  - local repo mirroring (via rsync)
- Typical rsync mirroring example for binaries in 1.3 release(s):

```
$ rsync -avzH --exclude src --exclude repocache --delay-updates  
rsync://build.openhpc.community/OpenHPC/1.3/ 1.3
```

# Wider variety of recipes now available

[ Nov. 2015 ]



OpenHPC (v1.0)  
Cluster Building Recipes

CentOS7.1 Base OS  
Base Linux\* Edition

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

*10 recipes now available with v1.3.6 release*

excellent place to start if you are new to the  
project or want to kick the tires

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

x86\_64:

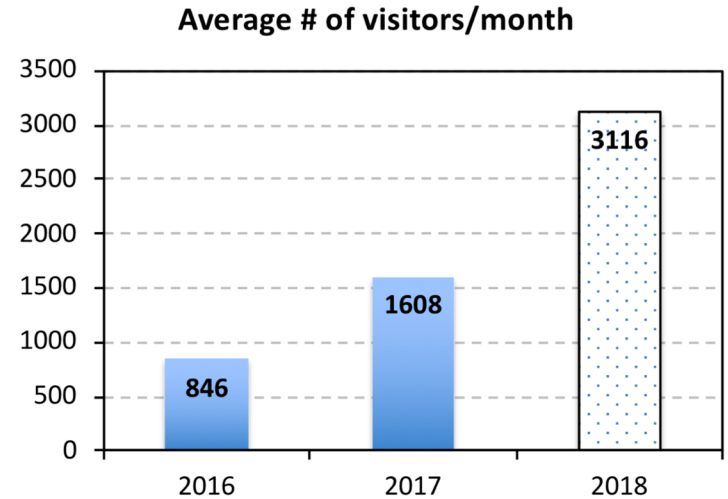
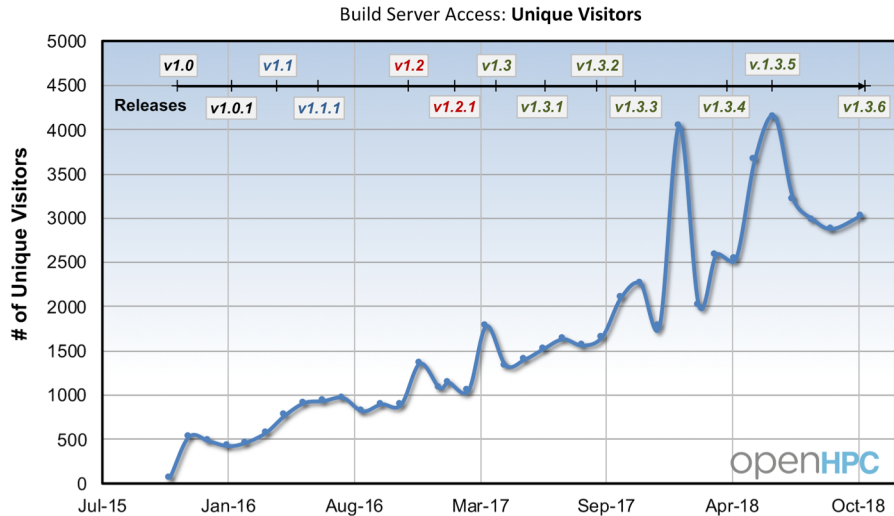
- [Install\\_guide-CentOS7-Warewulf-PBSPPro-1.3.5-x86\\_64.pdf](#)
- [Install\\_guide-CentOS7-Warewulf-SLURM-1.3.5-x86\\_64.pdf](#)
- [Install\\_guide-CentOS7-xCAT-Stateful-SLURM-1.3.5-x86\\_64.pdf](#)
- [Install\\_guide-CentOS7-xCAT-Stateless-SLURM-1.3.5-x86\\_64.pdf](#)
- [Install\\_guide-SLE\\_12-Warewulf-PBSPPro-1.3.5-x86\\_64.pdf](#)
- [Install\\_guide-SLE\\_12-Warewulf-SLURM-1.3.5-x86\\_64.pdf](#)

aarch64:

- [Install\\_guide-CentOS7-Warewulf-PBSPPro-1.3.5-aarch64.pdf](#)
- [Install\\_guide-CentOS7-Warewulf-SLURM-1.3.5-aarch64.pdf](#)
- [Install\\_guide-SLE\\_12-Warewulf-PBSPPro-1.3.5-aarch64.pdf](#)
- [Install\\_guide-SLE\\_12-Warewulf-SLURM-1.3.5-aarch64.pdf](#)



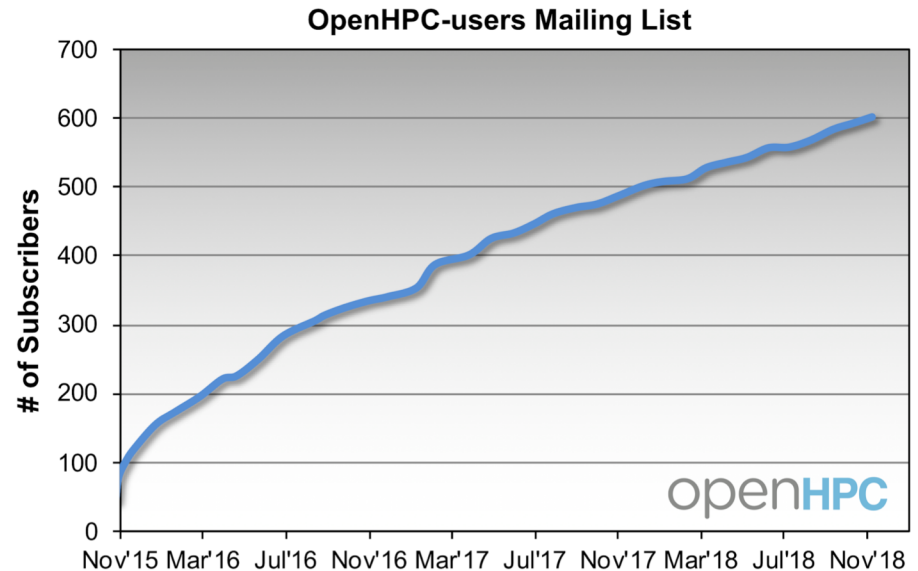
# Project Adoption Growth



- Continued access/download growth since initial release at SC'15
- Example highlights number of unique visitors/month to the OpenHPC build server/repo(s)

# Mailing Lists - <http://www.openhpc.community/support/mail-lists/>

- Three lists currently:
  - [openhpc-announce](#)
  - [openhpc-users](#)
  - [openhpc-devel](#)
- Great place to interact with developers and others using elements from the project
- *Really appreciate the great interaction we've seen on the list*



~90 posts/month in 2018

# OpenHPC Packaging Changes

# Quick Comments on 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:

```
1mod-ohpc-7.4.8-11.1.aarch64.rpm
```

- dependencies: compiler/MPI variant dependencies **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 ]

# RPM dependency conflict(s)

- Although unintended, we realized that previous OpenHPC releases had some namespace collision with base OS packages with respect to **dynamic .so** libraries (coming out of automatic RPM dependency analysis):
- Two examples:

```
# repoquery --whatprovides "libopenblas.so.0() (64bit)"  
openblas-0:0.2.20-3.e17.x86_64  
openblas-gnu7-ohpc-0:0.2.20-1.ohpc.x86_64
```

```
# repoquery --whatprovides "libltdl.so.7() (64bit)"  
libtool-ltdl-0:2.4.2-22.e17_3.x86_64  
libtool-ohpc-0:2.4.6-1.x86_64
```

- Note: any ohpc packages which might require the ohpc variant of openblas or libtool included additional dependency information that call out the need for ohpc version
- No issues if using all ohpc variants. But, problem *could* arise via order of operations if user installs the ohpc variant of openblas, but then installs other packages requiring libopenblas.so from EPEL (since we explicitly avoid installing into distro paths)


# RPM dependencies – custom ohpc plugin

- To deal with this issue, we have adopted a **new** approach for dynamic library analysis that relies on RPM's plugin infrastructure where we can define our own dependency generator hooks on a per path (and per file type basis)
- Triggered via the addition of attribute files in `/usr/lib/rpm/fileattrs`
- We now include in our `ohpc-buildroot` package which is needed to build ohpc packages`
- Let's look at the magic in `ohpc.attr`:

```
__ohpc_provides    /usr/lib/rpm/ohpc-find-provides
__ohpc_requires    /usr/lib/rpm/ohpc-find-requires %{buildroot} /opt/ohpc

__ohpc_path        ^/opt/ohpc
__elf_exclude_path ^/opt/ohpc

__ohpc_magic       ^ELF (32|64)-bit.*$
__ohpc_flags       magic_and_path
```



```
# rpm -ql ohpc-buildroot
/opt/ohpc/admin/ohpc
/opt/ohpc/admin/ohpc/OHPC_setup_compiler
/opt/ohpc/admin/ohpc/OHPC_setup_mpi
/usr/lib/rpm
/usr/lib/rpm/fileattrs
/usr/lib/rpm/fileattrs/ohpc.attr
/usr/lib/rpm/ohpc-find-provides
/usr/lib/rpm/ohpc-find-requires
```

- General approach here is to trigger these scripts for any build that has ELF binaries found in `/opt/ohpc`
  - Provides logic: any `.so` found in `/opt/ohpc` has an `"(ohpc)"` color string appended
  - Requires logic: any `.so` found in the build (or coming in as `BuildRequires`) that lives in `/opt/ohpc` has matching `"(ohpc)"` color string appended. Otherwise, dependencies belonging to packages outside of OpenHPC maintain default string.

# RPM dependencies: custom ohpc plugin

Let's look at some of the resulting Provides: differences for gcc using this approach. Essentially, every `.so` detected in `/opt/ohpc` has delimiter appended

```
# rpm -q --provides gnu7-compilers-ohpc | egrep libgomp
```

## OHPC 1.3.1 RPM

```
libgomp.so.1() (64bit)
libgomp.so.1(GOACC_2.0) (64bit)
libgomp.so.1(GOACC_2.0.1) (64bit)
libgomp.so.1(GOMP_1.0) (64bit)
libgomp.so.1(GOMP_2.0) (64bit)
libgomp.so.1(GOMP_3.0) (64bit)
libgomp.so.1(GOMP_4.0) (64bit)
libgomp.so.1(GOMP_4.0.1) (64bit)
libgomp.so.1(GOMP_4.5) (64bit)
libgomp.so.1(GOMP_PLUGIN_1.0) (64bit)
libgomp.so.1(GOMP_PLUGIN_1.1) (64bit)
libgomp.so.1(OACC_2.0) (64bit)
libgomp.so.1(OMP_1.0) (64bit)
libgomp.so.1(OMP_2.0) (64bit)
libgomp.so.1(OMP_3.0) (64bit)
libgomp.so.1(OMP_3.1) (64bit)
libgomp.so.1(OMP_4.0) (64bit)
libgomp.so.1(OMP_4.5) (64bit)
```

## Build with Updated Approach

```
libgomp.so.1() (64bit) (ohpc)
libgomp.so.1(GOACC_2.0) (64bit) (ohpc)
libgomp.so.1(GOACC_2.0.1) (64bit) (ohpc)
libgomp.so.1(GOMP_1.0) (64bit) (ohpc)
libgomp.so.1(GOMP_2.0) (64bit) (ohpc)
libgomp.so.1(GOMP_3.0) (64bit) (ohpc)
libgomp.so.1(GOMP_4.0) (64bit) (ohpc)
libgomp.so.1(GOMP_4.0.1) (64bit) (ohpc)
libgomp.so.1(GOMP_4.5) (64bit) (ohpc)
libgomp.so.1(GOMP_PLUGIN_1.0) (64bit) (ohpc)
libgomp.so.1(GOMP_PLUGIN_1.1) (64bit) (ohpc)
libgomp.so.1(OACC_2.0) (64bit) (ohpc)
libgomp.so.1(OMP_1.0) (64bit) (ohpc)
libgomp.so.1(OMP_2.0) (64bit) (ohpc)
libgomp.so.1(OMP_3.0) (64bit) (ohpc)
libgomp.so.1(OMP_3.1) (64bit) (ohpc)
libgomp.so.1(OMP_4.0) (64bit) (ohpc)
libgomp.so.1(OMP_4.5) (64bit) (ohpc)
```

# RPM dependencies: custom ohpc plugin

Let's look at some of the resulting Requires: differences for gcc using this approach.

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

## 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)
```

We introduced this  
new convention in  
v1.3.4 release



# Another packaging change in latest release

- Another small change to the package naming schema is introduced in the latest release (v1.3.6)
- This change is much simpler and simply embeds the OpenHPC release version where a package build originates from in the RPM release string
  - motivation: make it easier to look at package name and know which ohpc version it came from

Old Convention  
(v1.3.5 and older)

name version release arch  
docs-ohpc-1.3.5-45.1.x86\_64.rpm

**New Convention**  
(v1.3.6 and newer)

name version release arch  
docs-ohpc-1.3.6-17.1.**ohpc.1.3.6**.x86\_64.rpm

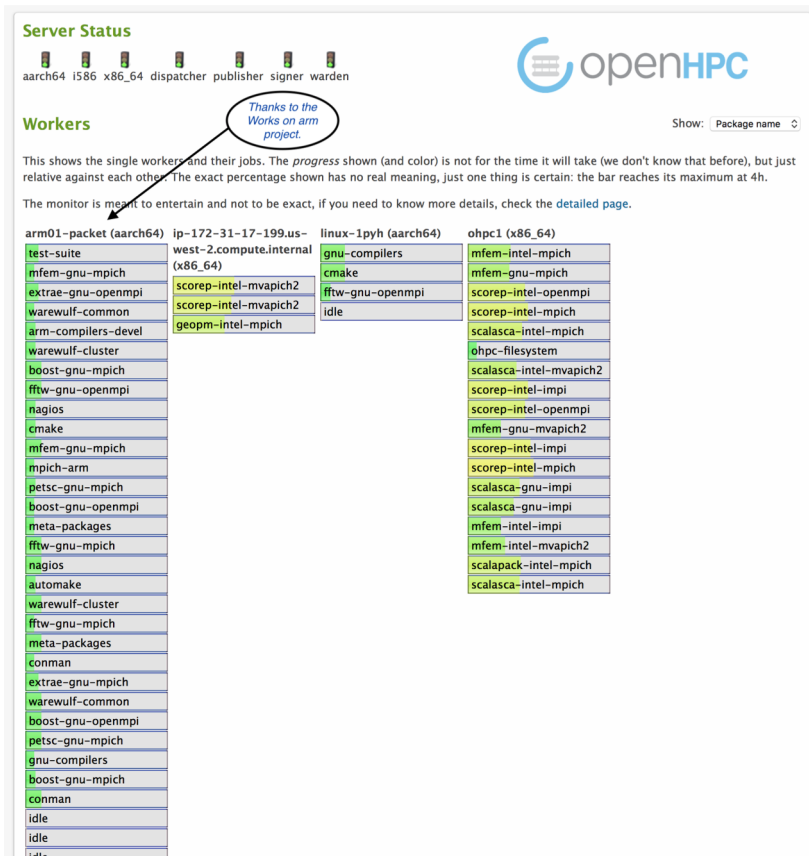
# Build system addition for aarch64

- There is a Works on ARM project that provides CI/CD infrastructure to community projects hosted at Packet

- <https://www.worksonarm.com>
- <https://www.packet.net>

- OpenHPC community requested CI host to support automated builds

- we were kindly granted a ThunderX2 node to support our build infrastructure
- have setup OBS on this host and tied it to our central OBS server
- significantly expands aarch64 build resources for the project
- Example build time reductions: **trilinos-mvapich2** build
  - previous build time: 18.79 hours
  - latest build time on TX2: 4.19 hours



# Requesting Additional Software

- Recall that we have a simple submission site for new requests:

- <https://github.com/openhpc/submissions>

- Items added via this mechanism since 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/Extrac,  
Paraver
- OpenCoarrays

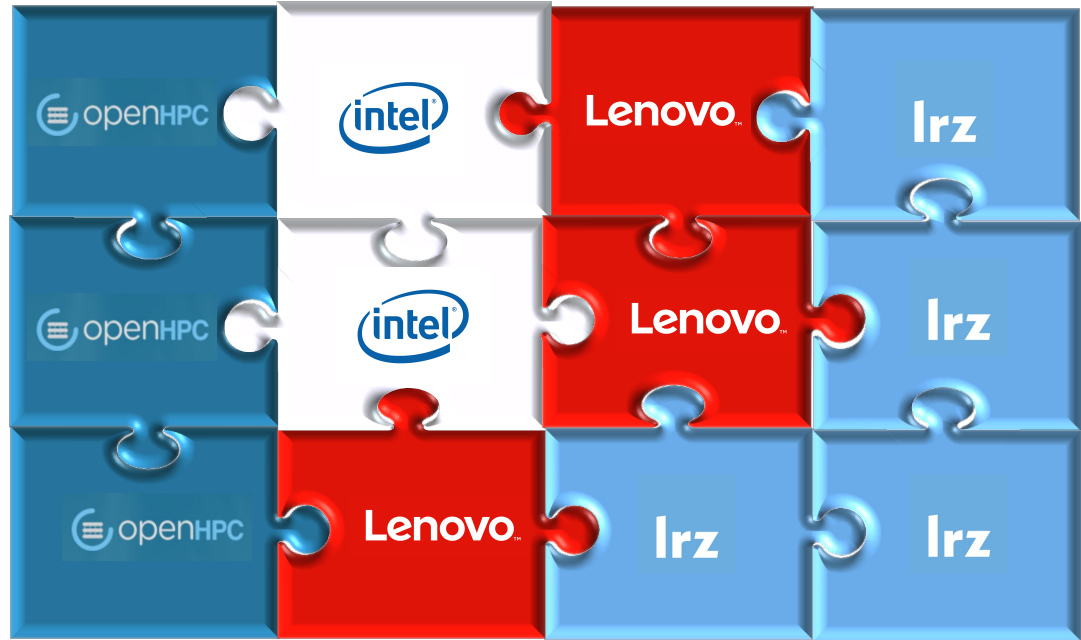
Next Submission Deadline: December 7th, 2018  
(continue thereafter on a rolling quarterly basis)

<b>Software Name</b>
<b>Public URL</b>
<b>Technical Overview</b>
<b>Latest stable version number</b>
<b>Open-source license type</b>
<b>Relationship to component?</b> <input type="checkbox"/> contributing developer <input type="checkbox"/> user <input type="checkbox"/> other
If other, please describe:
<b>Build system</b> <input type="checkbox"/> autotools-based <input type="checkbox"/> CMake <input type="checkbox"/> other
If other, please describe:

# New Large System Leveraging the Project

## SuperMUC-NG @ LRZ

- 26.9 Pflop/s
- 719 Tbyte main memory and
- 70 Pbyte disk storage
- 6,480 Lenovo™ ThinkSystem™ nodes with Intel® Xeon® processors



Lenovo Intelligent Computing Orchestration (LiCO). Intel® Parallel Studio, Intel® MPI.  
LRZ software deployed using Spack

# OpenHPC v1.3.6 Release

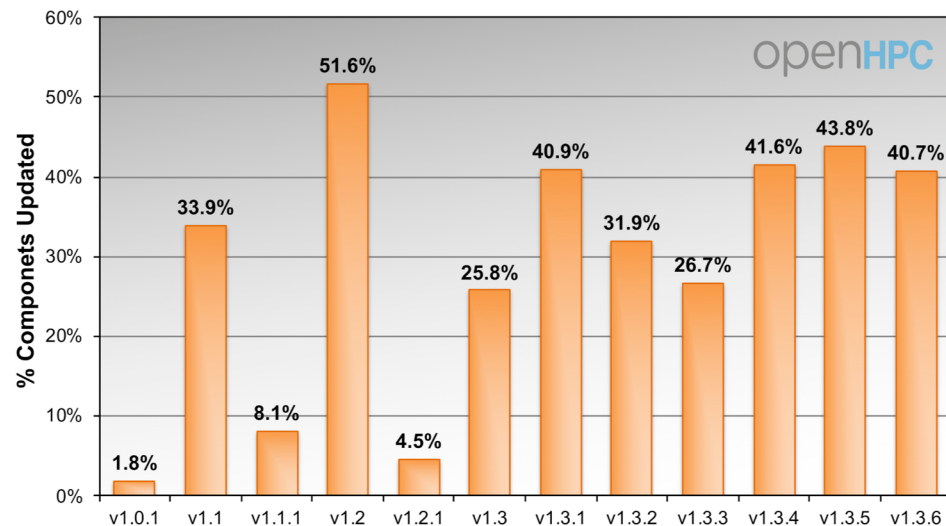
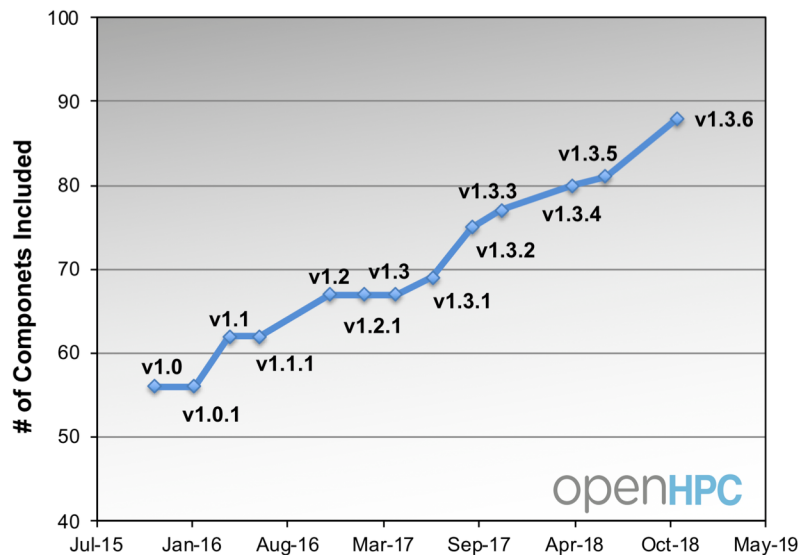
*November 2018*

# OpenHPC v1.3.6 - Current S/W components

Functional Areas	Components	new since last year
Base OS	CentOS 7.5, SLES12 SP3	
Architecture	aarch64, x86_64	
Administrative Tools	Conman, Ganglia, Lmod, LosF, Nagios, <a href="#">NHC</a> , pdsh, pdsh-mod-slurm, prun, EasyBuild, ClusterShell, mrsh, Genders, Shine, Spack, test-suite	
Provisioning	Warewulf, xCAT	
Resource Mgmt.	SLURM, Munge, PBS Professional, PMIx	
Runtimes	<a href="#">Charliecloud</a> , OpenMP, OCR, Singularity	
I/O Services	Lustre client, BeeGFS client	
Numerical/Scientific Libraries	Boost, GSL, FFTW, Hypre, Metis, <a href="#">MFEM</a> , Mumps, OpenBLAS, <a href="#">OpenCoarrays</a> , 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	<a href="#">gnu8 compiler variant introduced in v1.3.6</a>
MPI Families	MVAPICH2, OpenMPI, MPICH	
Development Tools	Autotools, cmake, hwloc, mpi4py, R, SciPy/NumPy, Valgrind	
Performance Tools	<a href="#">Dimemas</a> , <a href="#">Extrac</a> , <a href="#">GeoPM</a> , IMB, <a href="#">Likwid</a> , mpiP, <a href="#">msr-safe</a> , PAPI, <a href="#">Paraver</a> , pdtoolkit, Scalasca, ScoreP, SIONLib, TAU	

- Additional dependencies not provided by BaseOS or community repos are also included
- builds against Intel Parallel Studio toolchain provided for many development packages

# Component growth/update history



# of pre-packaged RPMs available (v1.3.6)

Base OS	aarch64	x86_64	noarch
CentOS 7	407	723	45
SLES 12	414	727	45

# Future Work

- Compatibility builds with ARM HPC compiler (expect in Q1 2019)
- Continued s/w updates and releases, OS updates....
- Ansible-based recipe

<https://github.com/openhpc/ohpc/wiki/User-Resources>

## Other installation recipes leveraging OpenHPC

- [XSEDE Compatible Basic Cluster](#)
  - Quickstart guide for an OpenHPC-based development cluster using ansible and Virtual Box developed by NSF's Extreme Science and Engineering Discovery Environment
- [Vanilla recipe using Ansible \(LANL\)](#)
- [Ansible playbook for OpenHPC \(Linaro\)](#)
- Exploring an archive repository
- *other items based on your feedback...*



# Open Discussion

- Before opening it up for discussion, one last item to mention
- OpenHPC is hosting a **Community Appreciation event (happy hour)** this evening
  - 5-7pm at Bourbon & Banter
  - within walking distance from convention center
  - opportunity to mingle with others from the community



The poster features the OpenHPC logo at the top left, followed by the event title. Below the title is the date and time, then the venue name and address. A QR code is provided for ticket information. The bottom of the poster has a stylized city skyline graphic.

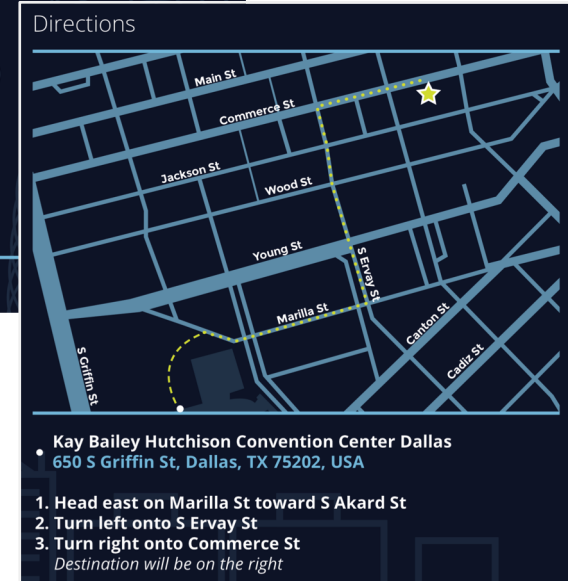
**openHPC**  
openhpc.community

## Community Appreciation Happy Hour

Wed, November 14, 2018  
5:00 PM - 7:00 PM

**Bourbon & Banter**  
1914 Commerce Street  
Dallas, TX 75201

GET YOUR  
TICKETS HERE!



The map shows a grid of streets in Dallas. A yellow star marks the destination at the intersection of Commerce St and S Ervay St. A dashed yellow line indicates the route starting from the convention center at S Griffin St, heading east on Marilla St, turning left onto S Ervay St, and then turning right onto Commerce St.

### Directions

- **Kay Bailey Hutchison Convention Center Dallas**  
650 S Griffin St, Dallas, TX 75202, USA

1. Head east on Marilla St toward S Akard St
2. Turn left onto S Ervay St
3. Turn right onto Commerce St

*Destination will be on the right*

# Open Discussion

# Pointers to Online Resources

- OpenHPC Home: <http://www.openhpc.community/>
- Primary GitHub Site: <https://github.com/openhpc/ohpc>
- Package Repositories: <http://build.openhpc.community/OpenHPC:/>
- OBS Frontend: <https://build.openhpc.community>
- 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