INTEL® HPC DEVELOPER CONFERENCE
FUEL YOUR INSIGHT
Legal Notices and Disclaimers

• Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

• No computer system can be absolutely secure.

• Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/performance.

• Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

• No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

• Intel, the Intel logo and others are trademarks of Intel Corporation in the U.S. and/or other countries.

• *Other names and brands may be claimed as the property of others.

• © 2016 Intel Corporation.
Simplified System Software Stack Development and Maintenance

Karl W. Schulz  
Technical Project Lead  
Datacenter Group, OpenHPC

John Westlund  
Systems SW Engineer  
Datacenter Group

November 2016
Agenda

• The HPC system software ecosystem problems we all deal with

• OpenHPC* community

• Intel® HPC Orchestrator

• How to make use of these system software solutions
Agenda

• The HPC system software ecosystem problems we all deal with

• OpenHPC* community

• Intel® HPC Orchestrator

• How to make use of these system software solutions
State of System Software Efforts in HPC Ecosystem

THE REALITY: We will not be able to get where we want to go without a major change in system software development

Fragmented efforts across the ecosystem – “Everyone building their own solution.”

A desire to get exascale performance & speed up software adoption of hardware innovation

New complex workloads (ML\(^1\), Big Data, etc.) drive more complexity into the software stack

\(^1\)Machine Learning (ML)
Community Effort to Realize Desired Future State

A Shared Repository

Stable HPC Platform Software that:

- Fuels a vibrant and efficient HPC software ecosystem
- Takes advantage of hardware innovation & drives revolutionary technologies
- Eases traditional HPC application development and testing at scale
- Extends to new workloads (ML, analytics, big data)
- Accommodates new environments (i.e., cloud)
Agenda

• Why a community system software stack?

• OpenHPC* community

• Intel® HPC Orchestrator

• How to make use of these system software solutions
A Brief History...

**June 2015**
ISC ‘15
- BoF\(^1\) discussion on the merits/interest in a Community Supported HPC Repository and Management Framework

**Nov 2015**
SC ‘15
- Follow-on BoF\(^1\) for a Comprehensive Open Community HPC Software Stack

**Nov ‘15-May ‘16**
Linux* Foundation
- Working group collaborating to define participation agreement, initial governance structure and solicit volunteers

**July 2016**
Linux Foundation
- announces technical, leadership and member investment milestones with founding members and formal governance structure

\(^1\) Birds of a Feather (BoF)
Community 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.

Courtesy of [openHPC](https://openhpc.org)
OpenHPC* Participation as of Nov 2016

- OpenHPC is a Linux Foundation Project initiated by Intel and gained wide participation right away
- The goal is to collaboratively advance the state of the software ecosystem
- Governing board is composed of Platinum members (Intel, Dell, HPE, SUSE) plus reps from Silver & Academic, Technical committees

29 Members

- Argonne National Laboratory
- Center for Research in Extreme Scale Technologies – Indiana University
- University of Cambridge

WWW.OpenHPC.Community

*Other names and brands may be claimed as the property of others.

Project member participation interest? Please contact Jeff ErnstFriedman: jernstfriedman@linuxfoundation.org
OpenHPC* Technical Steering Committee (TSC)

Role Overview

OpenHPC*
Technical Steering Committee (TSC)

- Project Leader
- Integration Testing Coordinator(s)
- Upstream Component Development Representative(s)
- End-User / Site Representative(s)
- Maintainers

https://github.com/openhpc/ohpc/wiki/Governance-Overview
## Stack Overview

We have assembled a variety of common ingredients required to deploy and manage an HPC Linux* cluster including provisioning tools, resource management, I/O libs, development tools, and a variety of scientific libraries.

<table>
<thead>
<tr>
<th>Operator Interface</th>
<th>Applications (not part of initial stack)</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Diagnostics</td>
<td>ISV Applications</td>
</tr>
<tr>
<td>Data Collection</td>
<td>Fabric Mgmnt</td>
</tr>
<tr>
<td>Resource Mgmnt</td>
<td>Optimized I/O Libraries</td>
</tr>
<tr>
<td>Resource Mgmnt</td>
<td>Scalable Debugging &amp; Perf Analysis Tools</td>
</tr>
<tr>
<td>DB Schema</td>
<td>High Performance Parallel Libraries</td>
</tr>
<tr>
<td>Workload Manager</td>
<td>Compiler &amp; Programming Model Runtimes</td>
</tr>
<tr>
<td>I/O Services</td>
<td>SW Development Toolchain</td>
</tr>
<tr>
<td>SW Development Toolchain</td>
<td>User Space Utilities</td>
</tr>
</tbody>
</table>

- Overlay & Pub-sub Networks, Identity
- Linux* Distro Runtime Libraries
- Node-specific OS Kernel(s)

**Hardware**

*Other names and brands may be claimed as the property of others.*
Stack Overview Continued

- Packaging efforts have **HPC in mind** and include compatible modules (for use with Lmod) with development libraries/tools
- Endeavoring to provide hierarchical development environment that is cognizant of different compiler and MPI families
- Include common conventions for env variables
- Development library install example:
  
  ```
  # yum install petsc-gnu-mvapich2-ohpc
  ```

- End user interaction example with above install: (assume we are a user wanting to build a PETSC hello world in C)
  
  ```bash
  $ module load petsc
  $ mpicc -I$PETSC_INC petsc_hello.c -L$PETSC_LIB -lpetsc
  ```

Courtesy of [openHPC](https://openhpc.com)
Basic Cluster Install Example

- Starting install guide/recipe targeted for flat hierarchy
- Leverages image-based provisioner (Warewulf)
  - PXE\(^1\) boot (stateless)
  - optionally connect external Lustre\(^*\) file system
- Obviously need hardware-specific information to support (remote) bare-metal provisioning

---

Figure 1: Overview of physical cluster architecture.

```
- $\{sms\_name\} # Hostname for SMS server
- $\{sms\_ip\} # Internal IP address on SMS server
- $\{sms\_eth\_internal\} # Internal Ethernet interface on SMS
- $\{eth\_provision\} # Provisioning interface for computes
- $\{internal\_netmask\} # Subnet netmask for internal network
- $\{ntp\_server\} # Local ntp server for time synchronization
- $\{bmc\_username\} # BMC username for use by IPMI
- $\{bmc\_password\} # BMC password for use by IPMI
- $\{c\_ip\{0\}\}, $\{c\_ip\{1\}\},... # Desired compute node addresses
- $\{c\_bmc\{0\}\}, $\{c\_bmc\{1\}\},... # BMC addresses for computes
- $\{c\_mac\{0\}\}, $\{c\_mac\{1\}\},... # MAC addresses for computes
- $\{compute\_regex\} # Regex for matching compute node names (e.g. c*)
```

Optional:
```
- $\{mgs\_fs\_name\} # Lustre MGS mount name
- $\{sms\_ipoib\} # IPoIB address for SMS server
- $\{ipoib\_netmask\} # Subnet netmask for internal IPoIB
- $\{c\_ipoib\{0\}\}, $\{c\_ipoib\{1\}\},... # IPoIB addresses for computes
```

\(^1\)Preboot eXecution Environment (PXE)
Hierarchical Overlay for OpenHPC® Software

General Tools and System Services
- Imod
- slurm
- munge
- losf
- warewulf
- lustre client
- ohpc
- prun
- pdsh

Compilers
- gcc

Serial Apps/Libs
- hdf5-gnu

MPI Toolchains
- MVAPICH2
- IMPI
- OpenMPI

Parallel Apps/Libs
- Boost
  - boost-gnu-openmpi
  - boost-gnu-impi
  - boost-gnu-mvapich2
- pHDF5
  - phdf5-gnu-openmpi
  - phdf5-gnu-impi
  - phdf5-gnu-openmpi

Development Environment

Intel Composer
- hdf5-intel

OpenMPI

Boost
- boost-intel-openmpi
- boost-intel-impi
- boost-intel-mvapich2
- pHDF5
  - phdf5-intel-openmpi
  - phdf5-intel-impi
  - phdf5-intel-mvapich2

Standalone 3rd party components

Distro Repo

OHPC Repo

Courtesy of openHPC
## OpenHPC* 1.1.1 – Current SW Components

<table>
<thead>
<tr>
<th>Functional Areas</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base OS</td>
<td>CentOS 7.2, SLES12 SP1</td>
</tr>
<tr>
<td>Administrative Tools</td>
<td>Conman, Ganglia, Lmod, LosF, Nagios, pdsh, prun, EasyBuild, ClusterShell,</td>
</tr>
<tr>
<td></td>
<td>mrsh, Genders, Shine, Spack</td>
</tr>
<tr>
<td>Provisioning</td>
<td>Warewulf</td>
</tr>
<tr>
<td>Resource Mgmt.</td>
<td>SLURM, Munge</td>
</tr>
<tr>
<td>Runtimes</td>
<td>OpenMP, OCR</td>
</tr>
<tr>
<td>I/O Services</td>
<td>Lustre client (community version)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functional Areas</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical/Scientific Libraries</td>
<td>Boost, GSL, FFTW, Metis, PETSc, Trilinos, Hypre, SuperLU, SuperLU_Dist, Mumps, OpenBLAS, Scalapack</td>
</tr>
<tr>
<td>I/O Libraries</td>
<td>HDF5 (pHDF5), NetCDF (including C++ and Fortran interfaces), Adios</td>
</tr>
<tr>
<td>Compiler Families</td>
<td>GNU (gcc, g++, gfortran)</td>
</tr>
<tr>
<td>MPI Families</td>
<td>MVAPICH2, OpenMPI</td>
</tr>
<tr>
<td>Development Tools</td>
<td>Autotools (autoconf, automake, libtool), Valgrind,R, SciPy/NumPy</td>
</tr>
<tr>
<td>Performance Tools</td>
<td>PAPI, IMB, mpiP, pdtoolkit TAU</td>
</tr>
</tbody>
</table>
OpenHPC* Development Infrastructure
What are we using to get the job done?

The usual software engineering stuff:

- GitHub* (SCM\(^1\) and issue tracking/planning)
- Continuous Integration (CI) Testing (Jenkins)
- Documentation (Latex)

Capable build/packaging system

- At present: we target a common delivery/access mechanism that adopts Linux sysadmin familiarity
- Require Flexible System to manage builds
- A system using Open Build Service (OBS) supported by back-end git

Courtesy of openHPC

\(^1\) Software Configuration Management (SCM)
Build System - OBS

- Manages build process
- Drives builds for multiple repositories
- Generates binary and src rpms
- Publishes corresponding package repositories
- Client/server architecture supports distributed build slaves and multiple architectures

https://build.openhpc.community
Integration/Test/Validation

- Install Recipes
- Cross-package interaction
- Development environment
- Mimic use cases common in HPC deployments
- Upgrade mechanism

![Diagram of Integration/Cluster Testing](image)

**Individual Component Validation**

**Integrated Cluster Testing**

- Dev Tools
- Parallel Libs
- System Tools
- Perf. Tools
- Compilers
- Resource Manager
- I/O Libs
- User Env
- Provisioner
- Mini Apps
- Serial Libs

**Software**

- OpenHPC
- OS Distribution

**Hardware**

*Courtesy of [openHPC](https://openhpc.io)*
Integration/Test/Validation

• Standalone integration test infrastructure
• Families of tests that could be used during:
  • initial install process (can we build a system?)
  • post-install process (does it work?)
  • developing tests that touch all of the major components (can we compile against 3rd party libraries, will they execute under resource manager, etc.)
• Expectation is that each new component included will need corresponding integration test collateral
• These integration tests are included in GitHub* repo
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.................... : jiluser
Build host.................... : master4-centos71.localdomain
Configure date................. : 2015-10-26 09:23
Build architecture............ : x86_64-unknown-linux-gnu
Test suite configuration...... : 1ong
```

### Submodule Configuration:

#### User Environment:
- RMS test harness
- Munge
- Apps
- Compilers
- MPI
- HSN
- Modules
- COM

#### Dev Tools:
- Valgrind
- R base package
- TBB
- CILK

#### Performance Tools:
- mpiP Profiler
- Papi
- PETSc
- TAU

### Libraries:
- Adios
- Boost
- Boost MPI
- FFTW
- GSL
- HDF5
- hypre
- IMB
- Metis
- MUMPS
- NetCDF
- Numpy
- OPENBLAS
- PETSc
- PHDF5
- ScalAPACK
- Scipy
- Superlu
- Superlu_dist
- Trilinos

### Apps:
- MiniFE
- MiniFFT
- HPCG
- PRK

Note: more than 1,000 jobs submitted to RM as part of the current test suite
New software additions?

• A common question posed to the project is how to request new software components? In response, the TSC has endeavored to formalize a simple submission/review process

• Submission site now exists for this purpose:
  
  https://github.com/openhpc/submissions

• Expecting to do reviews every quarter (or more frequent if possible)
  - just completed first iteration of the process now
  - next submission deadline: December 4th, 2016
How to contribute to OpenHPC*

- Use elements of the stack and provide feedback
- Suggest additional components for selection
- Make software of potential interest for inclusion available as open-source
- Participate in user/developer forums, TSC

http://openhpc.community (General info)
https://github.com/openhpc/ohpc (GitHub site)
https://github.com/openhpc/submissions (Submissions)
https://build.openhpc.community (Build system/repos)
http://www.openhpc.community/support/mail-lists/ (email lists)

opathy-announce, openhpc-users, openhpc-devel
Agenda

• Why a community system software stack?

• OpenHPC* community

• Intel® HPC Orchestrator

• How to make use of these system software solutions
OpenHPC* to Intel® HPC Orchestrator to Intel® Scalable System Framework

- Open Source Community under Linux Foundation*
- Ecosystem innovation building a consistent HPC SW Platform
- Platform agnostic
- 29 global members
- Multiple distributions

Intel® HPC Orchestrator

- Intel’s distribution of OpenHPC*; Intel HW optimized
- Expose best performance for Intel HW
- Advanced testing & premium features
- Product technical support & updates

Intel® Scalable System Framework
Holistic Design Solution for All HPC

- Small clusters through supercomputers
- Compute and data-centric computing
- Standards-based programmability
- On-Premise and cloud-based

* Focusing on Software Portion
Cycle supporting Product and Project advancement

- Product
  - Integrated into Intel® HPC Orchestrator
  - Customer Requests
  - New Technology

- Community
  - Evaluated for Maturity and Fit
  - OpenHPC* TSC Approval
  - Community Suggestions
  - Pull Requests
Intel® HPC Orchestrator System Architecture

- **Master Node**
  - Provisioning
  - Resource Management
  - Fabric Manager
  - Monitoring

- **Sub-Master Nodes**

- **Login Nodes**

- **Router Nodes**

- **Compute Nodes**

- **NFS / LAN / Internet**

- **Lustre® Parallel File System**

- **User Access**

- **Privileged Access**

- **Eth**

- **Fabric**

- **Perimeter**

- **Vision**
<table>
<thead>
<tr>
<th>Functional Areas</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base OS Compatibility</td>
<td>RHEL 7.2, SLES12 SP1, CentOS 7.2</td>
</tr>
<tr>
<td>Administrative Tools</td>
<td>Conman, Powerman, Ganglia, Nagios, Lmod, pdsh, ClusterShell, EasyBuild, Spack, mrsh, Genders, Shine</td>
</tr>
<tr>
<td>Provisioning</td>
<td>Warewulf</td>
</tr>
<tr>
<td>Resource Management</td>
<td>Slurm, MUNGE</td>
</tr>
<tr>
<td>I/O Services</td>
<td>Lustre client (Intel® Enterprise Edition for Lustre)</td>
</tr>
<tr>
<td>Numerical/Scientific Libraries</td>
<td>Boost, GSL, FFTW, Metis, PETSc, Trilinos, Hypre, SuperLU, SuperLU_Dist, MUMPS, OpenBLAS, Scalapack</td>
</tr>
<tr>
<td>I/O Libraries</td>
<td>HDF5 (pHDF5), NetCDF (including C++ and Fortran interfaces), ADIOS</td>
</tr>
<tr>
<td>Compiler Families</td>
<td>GNU (gcc, g++, gfortran), Intel® Parallel Studio XE</td>
</tr>
<tr>
<td>MPI Families</td>
<td>MVAPICH2, OpenMPI, Intel® MPI</td>
</tr>
<tr>
<td>Developer Tools</td>
<td>Autotools (autoconf, automake, libtool), Valgrind, R, SciPy/NumPy</td>
</tr>
<tr>
<td>Performance Tools</td>
<td>PAP, IMB, mpiP, pdtoolkit, TAU</td>
</tr>
</tbody>
</table>
Intel® HPC Orchestrator Enhancements

- Advanced integration testing & extensive validation
- Professional support for
  - All Intel components
  - Components where Intel maintains a support contract
- Best Effort Support for all other components
- Enhanced Documentation
  - Components Description Guide
  - Troubleshooting Guide, including Knowledge Base
  - Readme, Release Notes
  - Technical Update
- Validated security patches & updates
Intel® HPC Orchestrator Enhancements

• Early new hardware integration with System Software
• Inclusion of proprietary Intel Software
  - Intel® Parallel Studio XE 2017 (Cluster Edition)\(^1\)
  - Intel® Solutions for Lustre\(^*\) (Client) \(^1\)
• Planned Additional components
  - Support for high availability
  - Visualization tools
• SLES 12 SP1 Base OS redistribution available
• Integrated Test Suite
• Intel® Cluster Checker Supportability Extensions
Intel® Cluster Checker Supportability Extensions

New set of extensions to Intel® Cluster Checker
Baseline: system data collected when it is in a good, dependable state
Collects baseline data for:

- RPMs
  - Head Node
  - Virtual Node File System
- Configuration files (along with whitelist/blacklist capabilities)
- Hardware/Firmware

Compare current state of system with baseline
Intel® HPC Orchestrator: Summary

Benefits

**OEMs** – reduce R&D

**ISVs/Developers** – reduce time and man hours constantly retesting apps

**IT Admins** - reduce R&D to build and maintain a fully integrated SW stack

**End Users** - hardware innovation reflected in SW faster on path to exascale

- Integrated open source and proprietary components
- Modular build; Customizable; Validated updates
- Advanced integration testing, testing at scale
- Level 3 technical support provided by Intel
- Optimization for Intel® Scalable System Framework components
- Available through OEM & Channel Partners in Q4’16
Additional Sources of Information

OpenHPC* community – [www.openhpc.community](http://www.openhpc.community)


THANK YOU!
**Intel® Cluster Checker Supportability Extensions**

**Collecting RPM baseline data**

- Create nodefile
  ```
  # cat nodefile
  ```

- Run rpm-baseline command
  ```
  # rpm-baseline -f <path-to-nodefile>
  ```

- Data captured in
  ```
  /var/tmp/rpms-baseline.txt
  ```

**Example Output**

<table>
<thead>
<tr>
<th>Node name</th>
<th>RPM name</th>
<th>Version</th>
<th>Release</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>sms</td>
<td>libpciaccess</td>
<td>0.13.4</td>
<td>2.el7</td>
<td>x86_64</td>
</tr>
<tr>
<td>c1, c2</td>
<td>libpciaccess</td>
<td>0.13.4</td>
<td>2.el7</td>
<td>x86_64</td>
</tr>
</tbody>
</table>
Intel® Cluster Checker Supportability Extensions

Collecting **Files** baseline data

```
# files-baseline --f <path-to-nodefile>
```

Data captured in /var/tmp/files-baseline.txt

```
[sms]# cat /var/tmp/files-baseline.txt
sms, /etc/sysconfig/httpd, -rw-r--r--, root, root, 65947590cfc1df04aebc4df81983e1f5
.
.
c1, /etc/os-release, -rw-r--r--, root, root, 1359aa3db05a408808522a89913371f3
.
.
c2, /etc/sysconfig/munge, -rw-r--r--, root, root, e0505efde717144b039329a6d32a798f
.
```

Permissions

- File
- Owner
- Group
- MD5 Sum
Intel® Cluster Checker Supportability Extensions

Collecting **Hardware** baseline data

```bash
# hw-baseline -f <path-to-nodefile>
```

Data captured in `/var/tmp/hw-baseline.txt`

```bash
[sms]# cat /var/tmp/hw-baseline.txt
sms, 00:0d.0, Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode]
  .
  .
c1, 00:03.0, Intel Corporation 82540EM Gigabit Ethernet Controller
c1, 00:07.0, Intel Corporation 82371AB/EB/MB PIIX4 ACPI
  .
  .
c2, 00:05.0, Intel Corporation 82801AA AC'97 Audio Controller
  .
  .
```

**Bus**:Device:Function  
**Hardware description**
Intel® Cluster Checker Supportability Extensions

Comparing/Analyzing:

- Collect current system state
  
  ```
  # clck-collect -f <path-to-nodefile> -m uname -m files_head -m files_comp
  ```

- Analyze current system state against baseline
  
  ```
  # clck-analyze -f <path-to-nodefile> -l files
  ```

1 undiagnosed sign:

1. The file `/etc/pam.d/ppp` has been added since the baseline was generated.
   
   [ Id: files-added ]
   
   [ Severity: 25%; Confidence: 90% ]
   
   [ Node: RHEL2 ]

This analysis took 0.388902 seconds.

FAIL: All checks did not pass in health mode.
Community Workflow

List of Components from Upstream Communities
- Warewulf
- Ganglia
- Lustre
- Munge
- EasyBuild
- Slurm
- Nagios
- GNU
- OpenMPI
- NumPy
- R
- Project
- Adios
- Boost
- HDF5
- HYPRE
- Losf
- Numa
- OpenBLAS
- PAPI
- Pdsh
- Petsc
- Lua
- mpfr
- MUMPS
- MVAPICH
- NetCDF
- OpenBlas
- PAPI
- Petsc
- Lapack
- SuperLU
- Trilinos
- Valgrind

RRVs

Intel HPC Orchestrator

OpenHPC

Integrates and tests HPC SW stacks and makes available as OSS

Base HPC Stack

Continuous Integration Environment
- Build Environment & Source Control
- Bug Tracking
- User & Dev Forums
- Collaboration tools
- Validation Environment

RRV* = reliable and relevant version

* RRV = reliable and relevant version