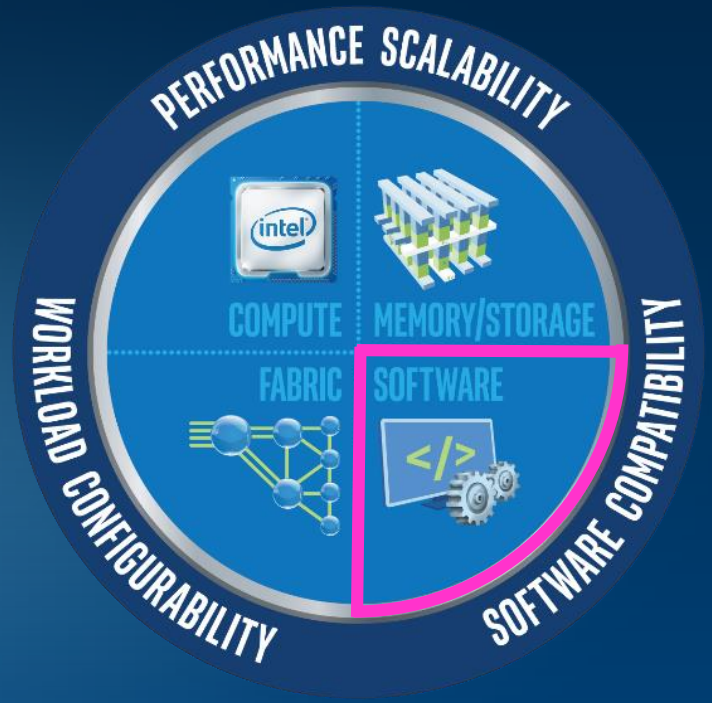
0 to 60 With Intel® HPC Orchestrator

HPC Advisory Council Stanford Conference — February 7 & 8, 2017



Steve Jones
Director, High Performance Computing Center
Stanford University

David N. Lombard
Chief Software Architect Intel® HPC Orchestrator
Senior Principal Engineer, Extreme Scale Computing, Intel

Dellarontay Readus
HPC Software Engineer, and CS Undergraduate
Stanford University

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.
-

Agenda

- Demo Outline
- Intel® HPC Orchestrator and OpenHPC
- Installation Walkthrough
- Intel® Parallel Studio XE
- Modules Support
- Management and Maintenance
- Workload Management
- 3rd-Party Packages
- Demo of Built Cluster
- Moving Forward

Demo Outline

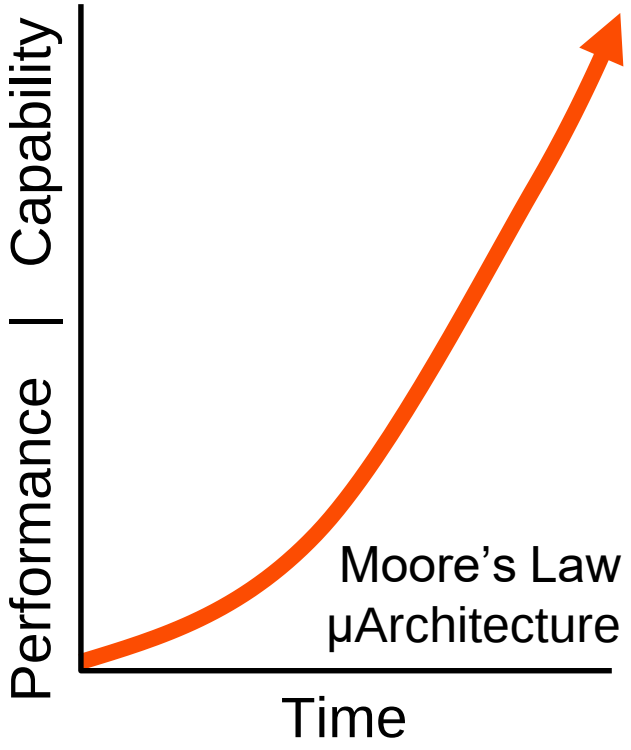
Intel® HPC Orchestrator & OpenHPC

Intel® Scalable System Framework (Intel® SSF)

A systems approach for Innovation

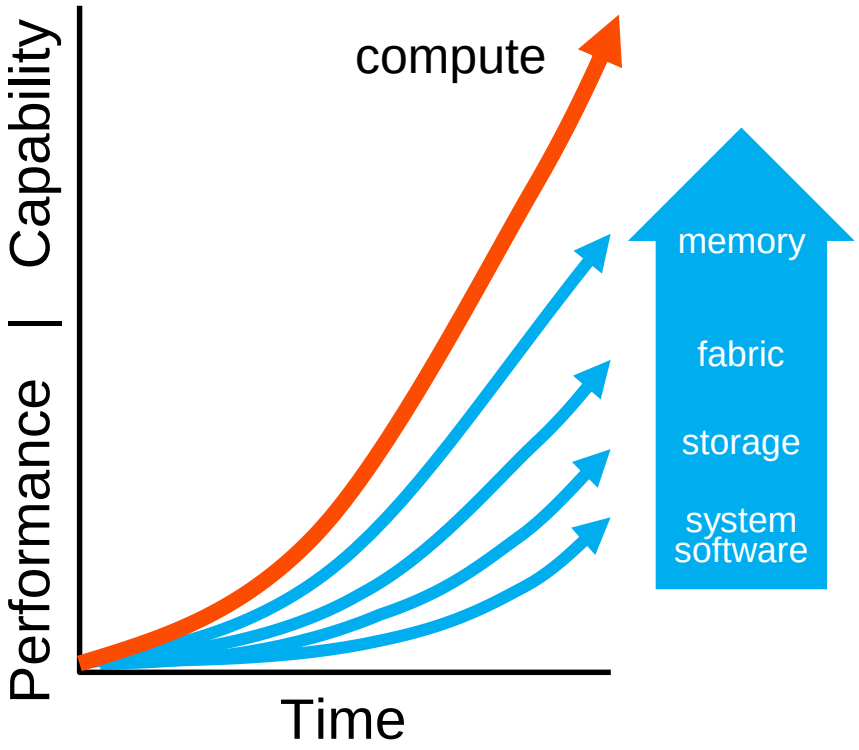
Continue the pace of

Faster Compute



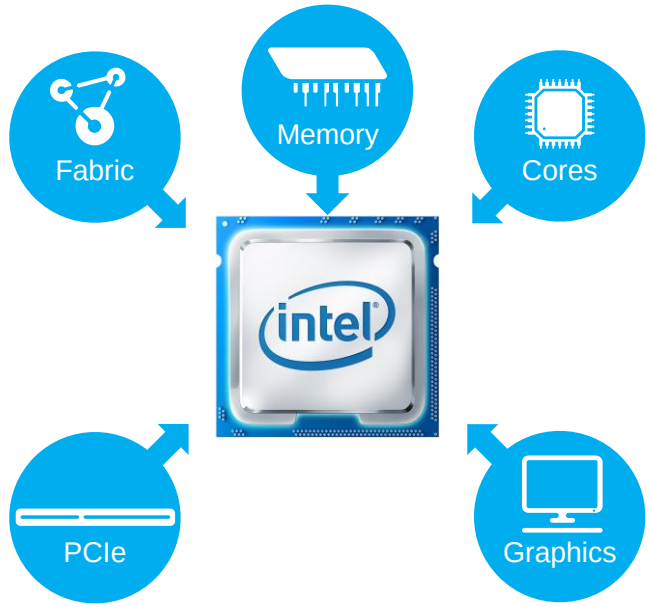
Develop new technologies to

Remove Bottlenecks



Achieve full potential via

Tighter Integration



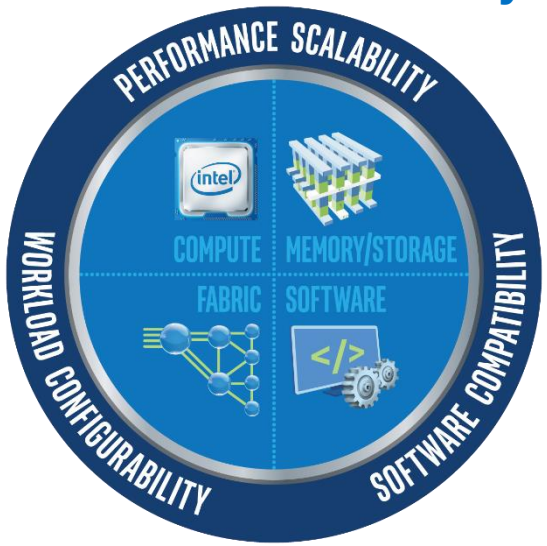
Better power, performance, density, scaling & cost

Execution of Vision

Intel® HPC Orchestrator products are the realization of the software portion of Intel® Scalable System Framework

openHPC

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



Intel® Scalable System Framework (SSF)

A Holistic Design Solution for All HPC

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

Intel HW Validation, SW Alignment & Technical Support

Intel® HPC Orchestrator

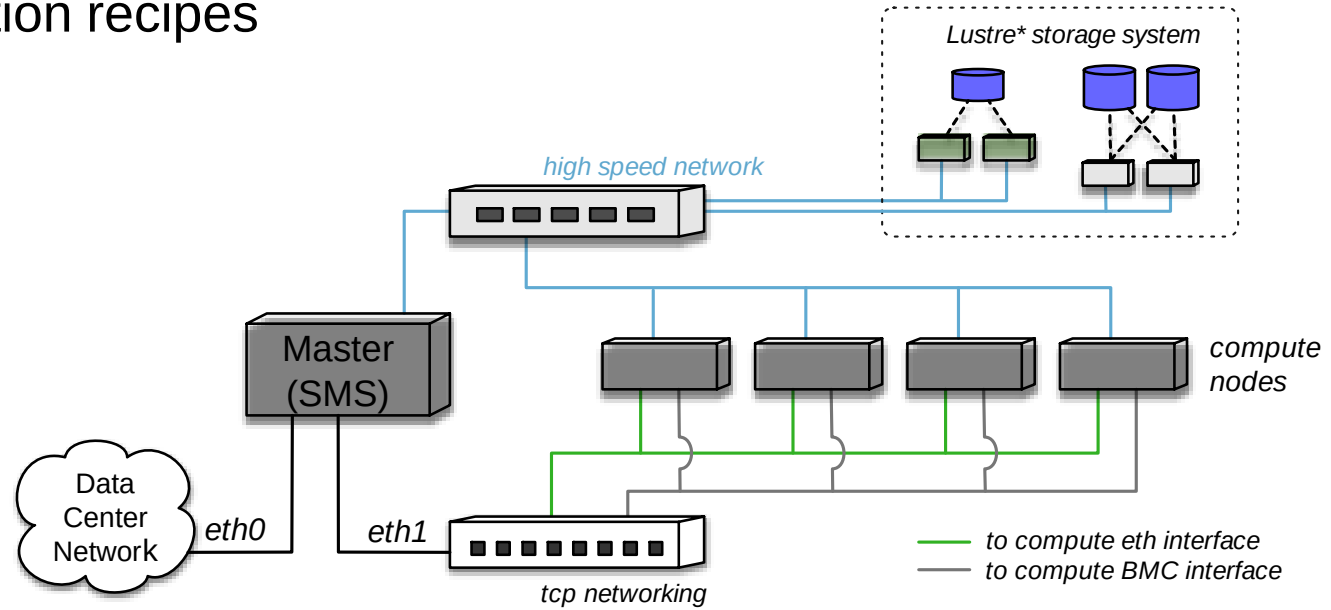
- Intel's distribution of OpenHPC
- Intel HW validation
- Expose best performance for Intel HW
- Advanced testing & premium features
- Product technical support & updates

What is OpenHPC?

OpenHPC is a community effort endeavoring to:

- provide collection(s) of pre-packaged components that can be used to help install and manage flexible HPC systems throughout their lifecycle
- leverage standard Linux delivery model to retain admin familiarity (i.e., package repos)
- allow and promote multiple system configuration recipes that leverage community reference designs and best practices
- implement integration testing to gain validation confidence
- provide additional distribution mechanism for groups releasing open-source software
- provide a stable platform for new R&D initiatives

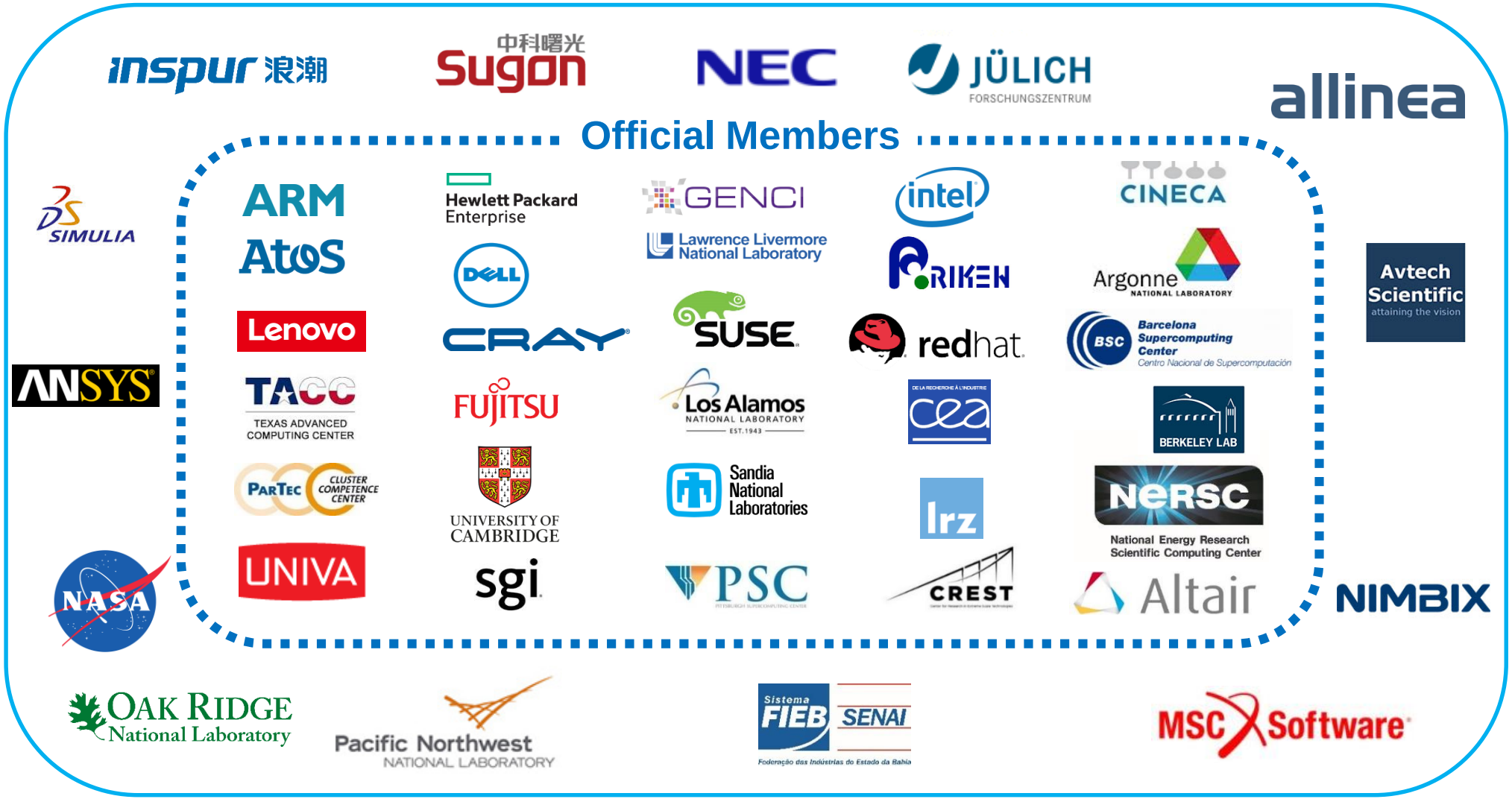
Typical cluster architecture



Courtesy of OpenHPC*

Participation as of January'17

Mixture of Academics, Labs, OEMs, and ISVs/OSVs



Courtesy of OpenHPC*

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

Project member participation interest? Please contact
Kevlin Husser or Jeff ErnstFriedman <jernstfriedman@linuxfoundation.org>

Individuals

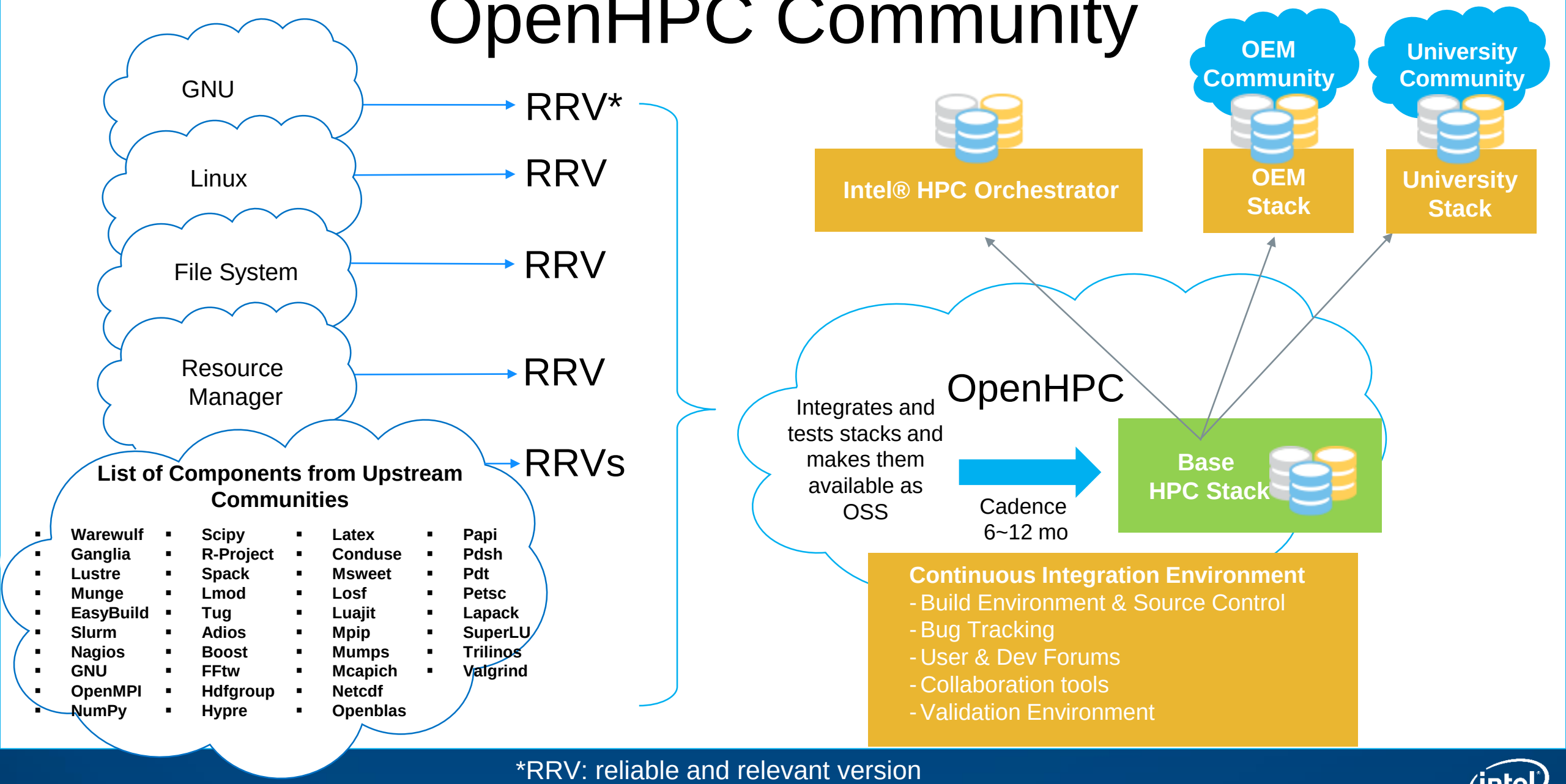
- Reese Baird, Intel (Maintainer)
- Pavan Balaji, Argonne National Laboratory (Maintainer)
- David Brayford, LRZ (Maintainer)
- Todd Gamblin, Lawrence Livermore National Labs (Maintainer)
- Craig Gardner, SUSE (Maintainer)
- Yiannis Georgiou, ATOS (Maintainer)
- Balazs Gerofi, RIKEN (Component Development Representative)
- Jennifer Green, Los Alamos National Laboratory (Maintainer)
- Eric Van Hensbergen, ARM (Maintainer, Testing Coordinator)
- Douglas Jacobsen, NERSC (End-User/Site Representative)
- Chulho Kim, Lenovo (Maintainer)
- Greg Kurtzer, Lawrence Berkeley National Labs (Component Development Representative)
- Thomas Moschny, ParTec (Maintainer)
- Karl W. Schulz, Intel (Project Lead, Testing Coordinator)
- Derek Simmel, Pittsburgh Supercomputing Center (End-User/Site Representative)
- Thomas Sterling, Indiana University (Component Development Representative)
- Craig Stewart, Indiana University (End-User/Site Representative)
- Scott Suchyta, Altair (Maintainer)
- Nirmala Sundararajan, Dell (Maintainer)

Courtesy of OpenHPC* ***https://github.com/openhpc/ohpc/wiki/Governance-Overview***

Intel® HPC Orchestrator Product Family

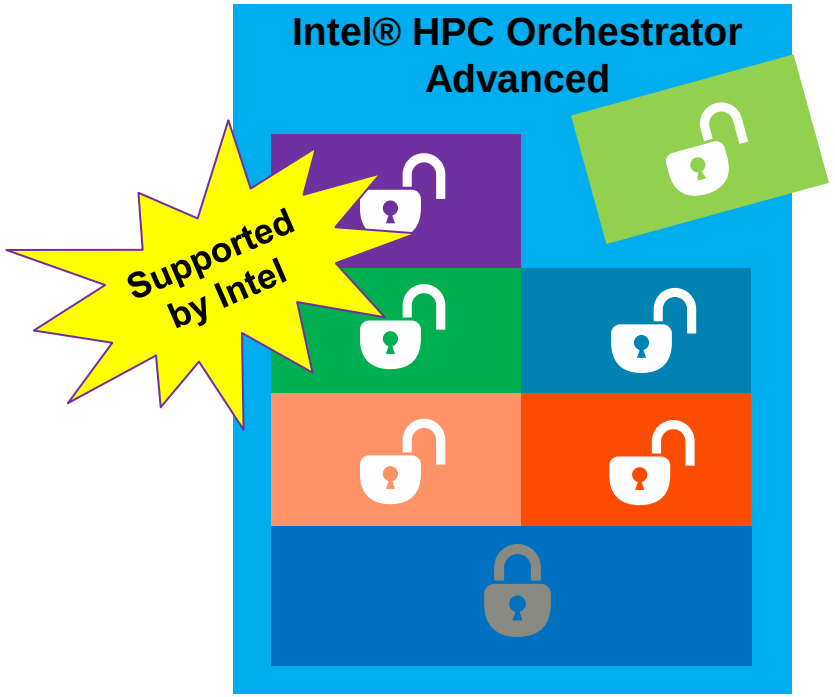
Product		Target Customer
 **CUSTOM** **Highly modular hierarchically configured stack**		**High End HPC users**
 **ADVANCED** **Modular stack with a flat configuration**	***GA today***	**Technical & commercial users with vertical HPC applications**
 **TURNKEY** **Turnkey stack with a flat configuration. A variety of stacks may be available as offerings**		**HPC ISVs, SIs and customers looking for a turnkey solution for On-Prem and CSP access methodologies, appliances**

OpenHPC Community



*RRV: reliable and relevant version

Intel® HPC Orchestrator Value Proposition



- Integrated open source and proprietary software
- Modular build; customizable; with validated updates
- Advanced integration testing, testing at scale (1,000 nodes)
- Optimization for Intel® Scalable System Framework components, such as Intel® Omni-Path Architecture and Intel® Xeon Phi™
- Level 3 technical support provided by Intel

Benefits

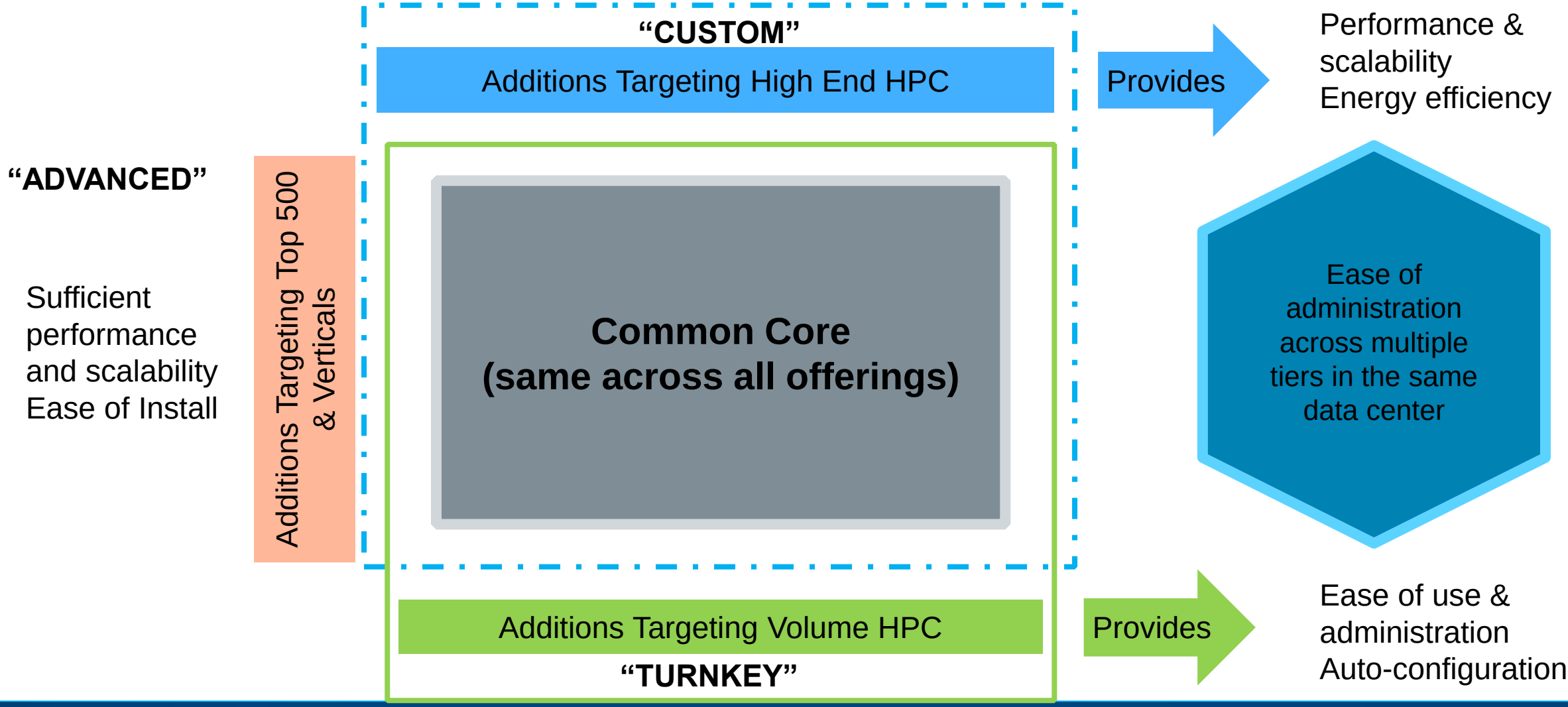
End Users - hardware innovation reflected in SW; spend time on scientific work rather than testing ; faster on path to exascale

Sys Admins - reduce R&D to build and maintain a fully integrated SW stack, easier mgmt. of clusters from multiple vendors

Developers/ISVs – reduce time and resources for constantly retesting apps as new open source components are released and updated

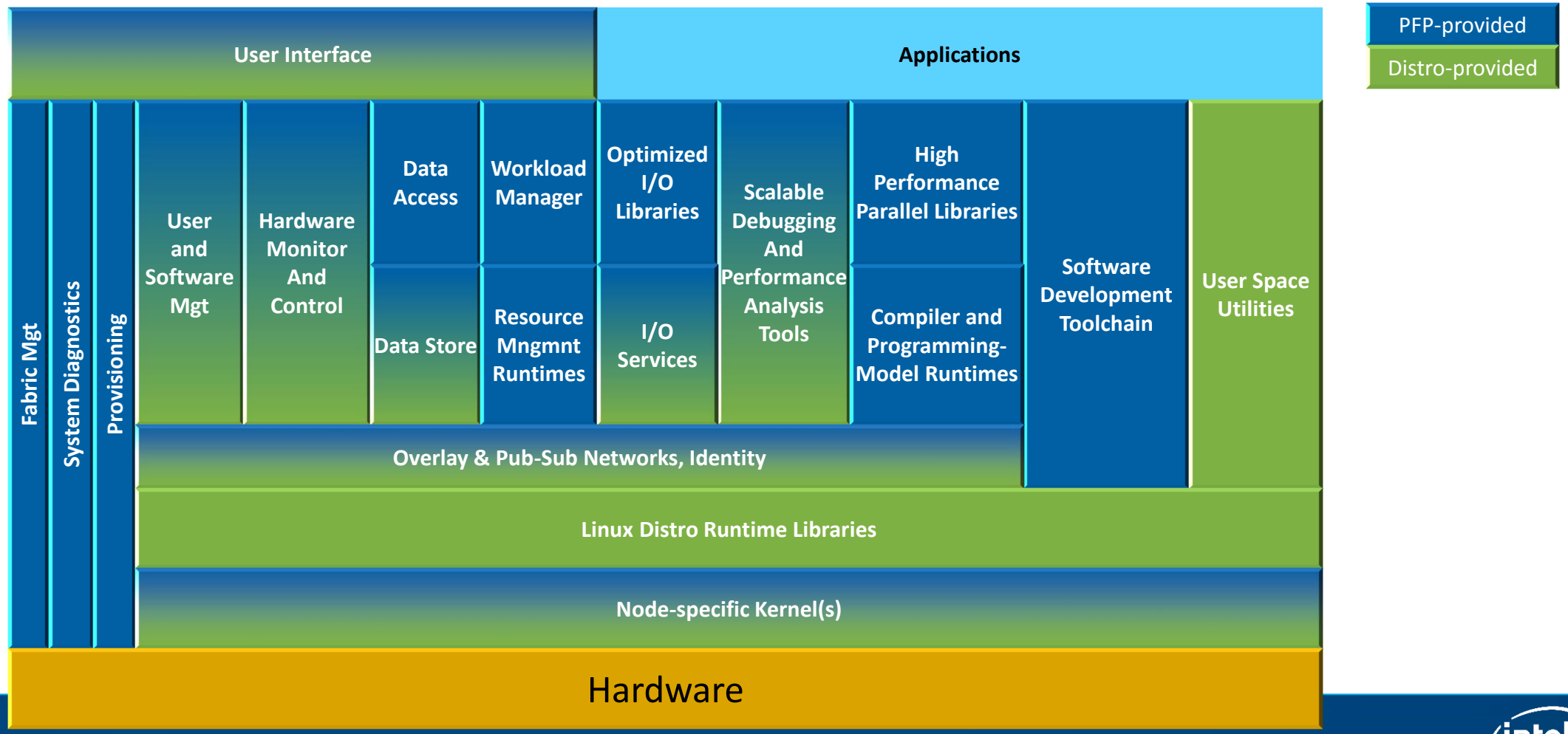
OEMs – reduce R&D to build and maintain a fully integrated SW platform, focus on providing differentiation on top of the system stack

Base Stack and Derivatives



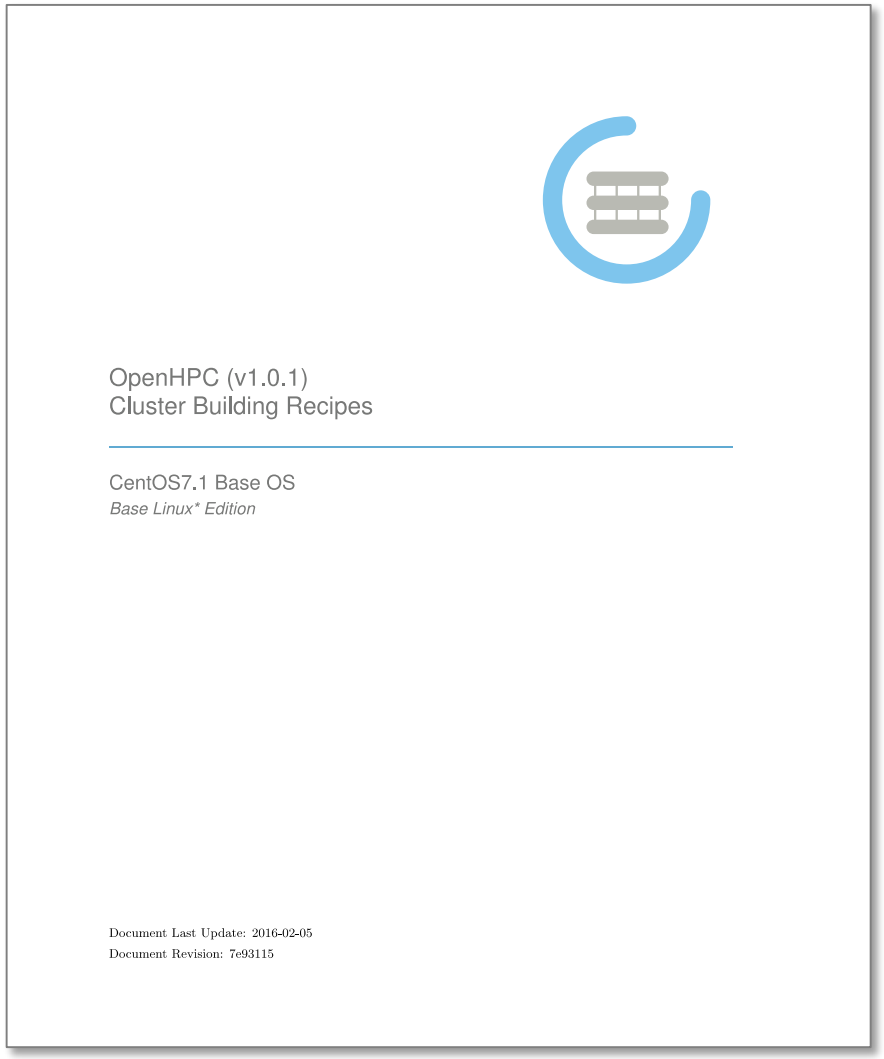
Modular Stack View

- Intra-stack APIs to allow for customization/differentiation (OEMs enabling)
- Defined external APIs for consistency across versions (ISVs)



Installation Walkthrough

Documentation Overview



OpenHPC (v1.0.1)
Cluster Building Recipes

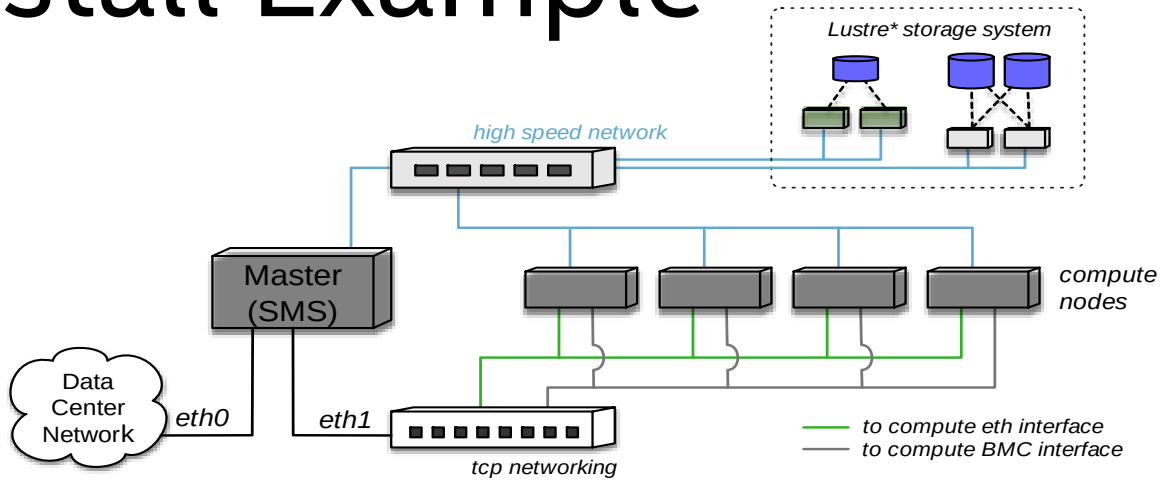
CentOS7.1 Base OS
Base Linux Edition*

Document Last Update: 2016-02-05
Document Revision: 7e93115

Contents

Basic Cluster Install Example

- Starting install guide/recipe targeted for flat hierarchy
- Image-based provisioner (Warewulf)
  - PXE boot
  - Stateless CNs
  - Optionally connect external Lustre* file system
- Hardware-specific information to support (remote) bare-metal provisioning



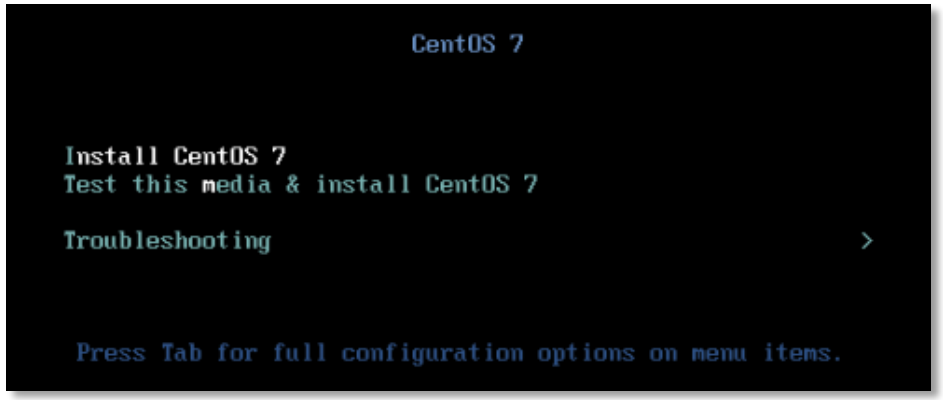
- `${sms_name}` # Hostname for SMS server
- `${sms_ip}` # Internal IP address on SMS server
- `${sms_eth_internal}` # Internal Ethernet interface on SMS
- `${eth_provision}` # Provisioning interface for computes
- `${internal_netmask}` # Subnet netmask for internal network
- `${ntp_server}` # Local ntp 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

Stack Overview: Bare metal install

Step1: Example OpenHPC 1.2 recipe assumes base OS is first installed on chosen master (SMS) host - e.g. install CentOS7.2 on SMS



Step2: Enable OpenHPC repo using pre-packaged ohpc-release (or mirror repo locally)

```
# export OHPC_GITHUB=https://github.com/openhpc/ohpc/releases/download
# rpm -ivh ${OHPC_GITHUB}/v1.2.1.GA/ohpc-release-1.2-1.x86_64.rpm
```

Stack Overview: Bare metal install (cont.)

– Note that ohpc-release enables two repos:

```
# yum repolist
repo id                                  repo name
OpenHPC                                  OpenHPC-1.2 - Base
OpenHPC-updates                          OpenHPC-1.2 - Updates
base                                     CentOS-7 - Base
epel                                     Extra Packages for Enterprise Linux 7 - x86_64
```

– Step3: install desired building blocks to build cluster or add development tools. **Convenience aliases** are provided to group related functionality

```
[sms]# yum -y groupinstall ohpc-base
[sms]# yum -y groupinstall ohpc-warewulf
```

Add provisioning components

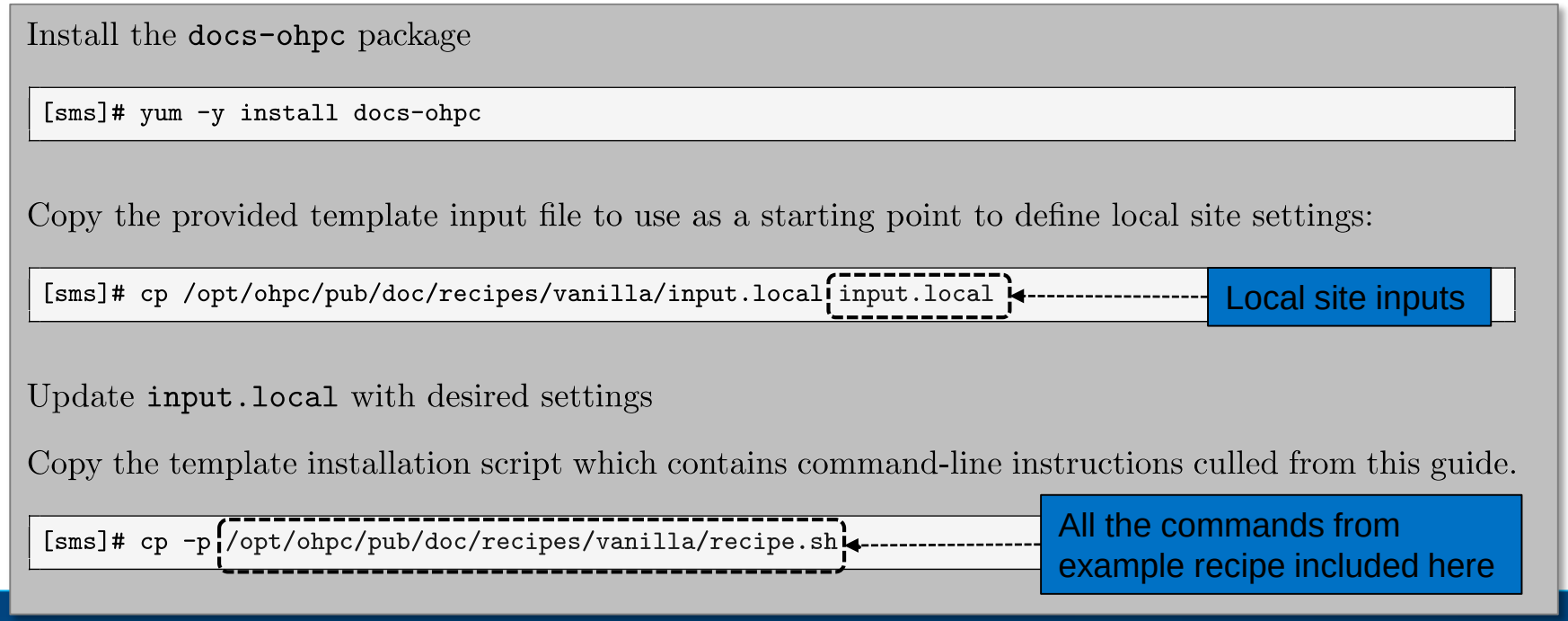
```
[sms]# yum -y install pbspro-server-ohpc
```

Add PBS Professional components

**note that community recipe is purposefully very transparent on config file edits and assumes Linux familiarity*

Stack Overview: Bare metal install (cont.)

- Recipe guides necessarily have a number of things to “cut-and-paste” if you want to reproduce them
- We have a motivating need to automate during the validation process:
  - Cull out relevant commands automatically for use during CI testing
  - A template starting script is available with the documentation RPM which can be used for local installation and customization



Intel® Parallel Studio XE

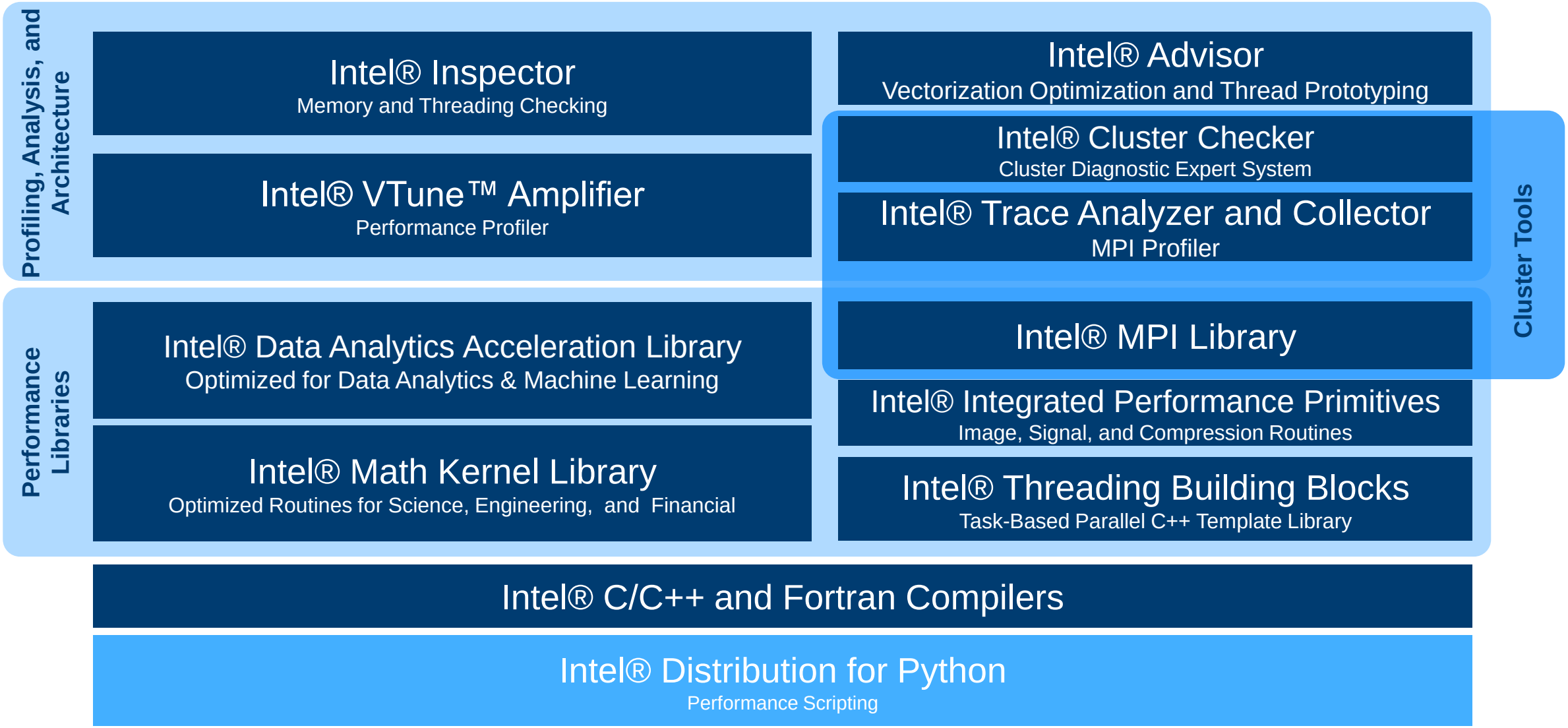
Create Faster Code…Faster

Intel® Parallel Studio XE

- High Performance Scalable Code
  - C++, C, Fortran*, Python* and Java*
  - Standards-driven parallel models: OpenMP*, MPI, and TBB
- New for 2017
  - 2nd generation Intel® Xeon Phi™ and AVX-512
    - Optimized compilers and libraries
    - Vectorization and threading optimization tools
    - High bandwidth memory optimization tools
  - Faster Python application performance
  - Faster deep learning on Intel® architecture

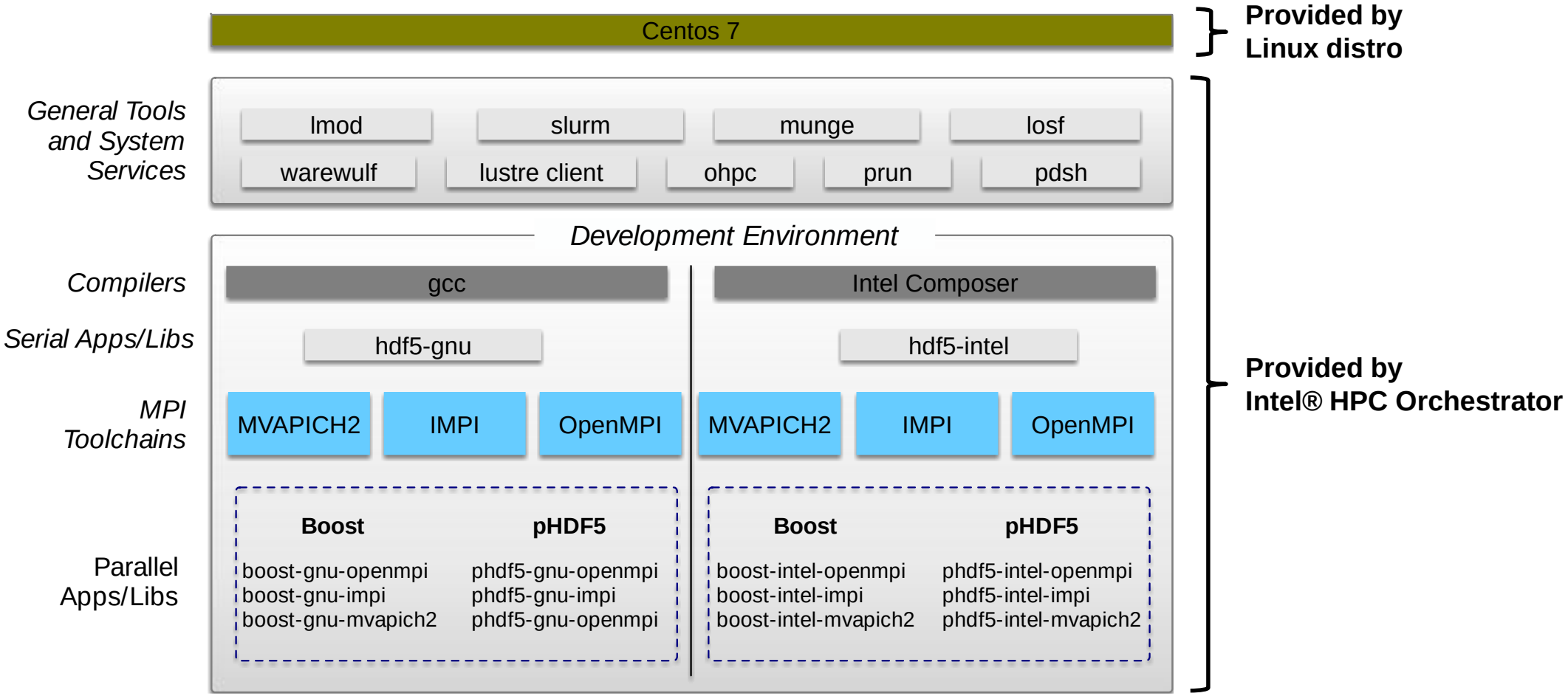
http://intel.ly/perf-tools

Intel® Parallel Studio XE



Modules Support

Hierarchical Overlay



Courtesy of OpenHPC*

Hierarchical Software - User experience

- Goal is to have a component hierarchy reflected in the user environment
- User sees compatible software based on the currently loaded environment
- Manage compiler/MPI dependencies via Lmod “families”

```
$ module list
Currently Loaded Modules:
   1) gnu/5.3.0          2) impi/5.1.3.181   3) ohpc    4) boost/1.60.0

$ module avail
------ /opt/intel/hpc-orchestrator/pub/moduledeps/gnu-impi ------
   boost/1.60.0    phdf5/1.8.16

-------- /opt/intel/hpc-orchestrator/pub/moduledeps/gnu----------
   hdf5/1.8.16    impi/5.1.3.181    mvapich2/2.1    openmpi/1.10.1

--------- /opt/intel/hpc-orchestrator/pub/modulefiles -----------
   autotools    hpc-orchestrator   gnu/5.3.0    intel/16.0.2.181

$ echo $BOOST_LIB
/opt/intel/hpc-orchestrator/pub/libs/gnu/impi/boost/1.60.0/lib

$ module swap gnu intel
Due to MODULEPATH changes the following have been reloaded:
  1) boost/1.60.0  2) impi/5.1.3.181

$ echo $BOOST_LIB
/opt/intel/hpc-orchestrator/pub/libs/intel/impi/boost/1.60.0/lib
```

Management and Maintenance

System Monitoring and Control

Control

- Intel® Cluster Checker – system health
- genders – cluster configuration DB
- mrsh – remote shell using Munge-based authentication
- pdsh – parallel distributed shell (mrsh or ssh)
- powerman – OOB power control
- prun – abstract parallel launch
- ssh – distro-provided

Monitoring

- conman – remote console access
- Ganglia – scalable node monitoring and visualization
- Nagios – servers, switches, applications, and services

Workload Management

Workload Management

- The workload managers (WLM) run user workflows on system
  - –Workflows are built around specifics of WLM
  - –WLM choice is *very important* to user community → key modularity driver!
- Two leading WLMs are supported initially
  - –SLURM (currently supported)
    - –Initially developed at LLNL; now supported by SchedMD
    - –Widely used on Top-500 systems
  - –PBS Professional (in OpenHPC; available in HPC Orchestrator presently)
    - –Initially developed for NASA; now supported by Altair
    - –Widely used in volume HPC systems

3rd-Party Packages

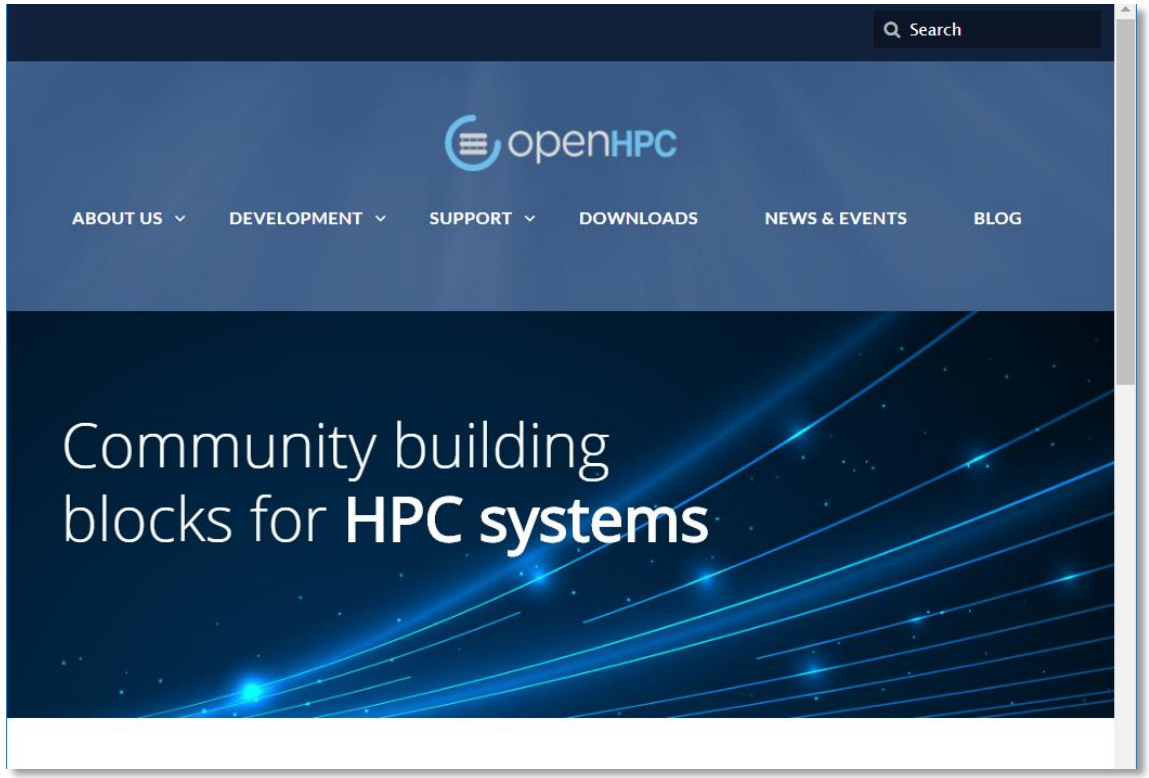
3rd-Party Packages

Functional Areas	Components
Base OS compatibility	RHEL 7.2 BU7 / SLES12 SP1 / CentOS 7.2 (coming soon)
Administrative Tools	Conman, Ganglia, Intel® Cluster Checker, Lmod, Losf, Nagios, pdsh, prun
Provisioning	Warewulf
Resource Mgmt	Slurm, Munge, (PBS Professional coming soon)
I/O Services	Lustre Client
I/O Libraries	HDF5 (pHDF5), NetCDF (including C++ & Fortran libraries), Adios
Compiler Families	GNU (gcc, g++, gfortran), Intel Parallel Studio XE** (icc, icpc, ifort)
MPI Families	OpenMPI, MVAPICH2, Intel MPI**
Development Tools	Autotools (autoconf, automake, libtool), Valgrind, R, SciPy, Numpy
Performance Tools	PAPI, Intel IMB, mpiP, pdtoolkit, TAU, Intel Advisor, Intel Trace Analyzer & Collector**, Intel VTune Amplifier**
Numerical/Scientific Libraries	Boost, GSL, FFTW, Metis, PETSc, Trilinos, Hypre, SuperLU, Mumps, Intel MKL**

Moving Forward

Further Information

openhpc.community



mike.sheppard@intel.com
thomas.a.krueger@intel.com
www.intel.com/hpcorchestrator

Backup

OpenHPC

OpenHPC: 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*

Information/Places to Interact

http://openhpc.community (general info)

https://github.com/openhpc/ohpc (GitHub site)

https://github.com/openhpc/submissions (new submissions)

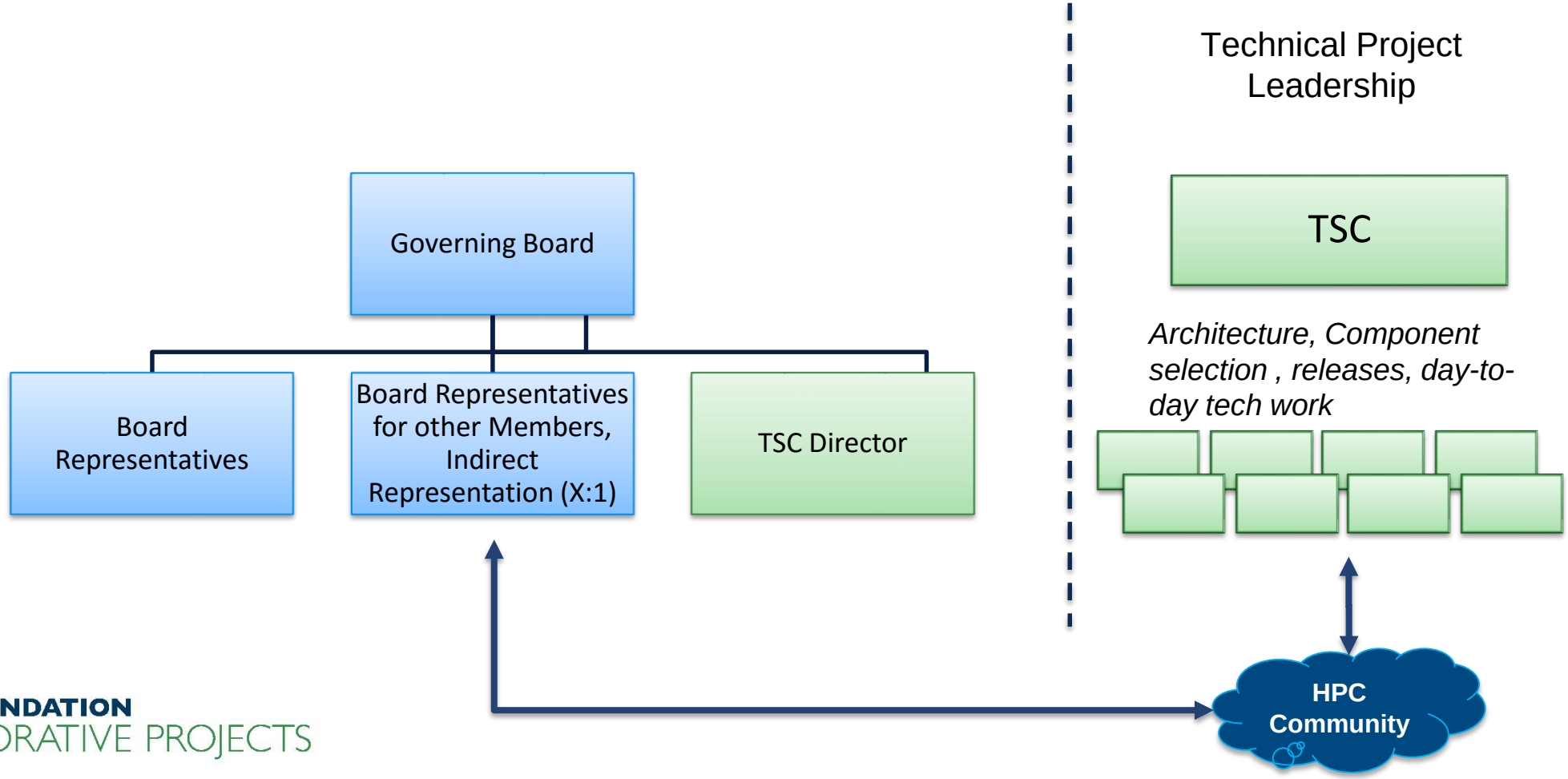
https://build.openhpc.community (build system/repos)

http://www.openhpc.community/support/mail-lists/ (email lists)

- openhpc-announce
- openhpc-users
- openhpc-devel

Community Governance Overview

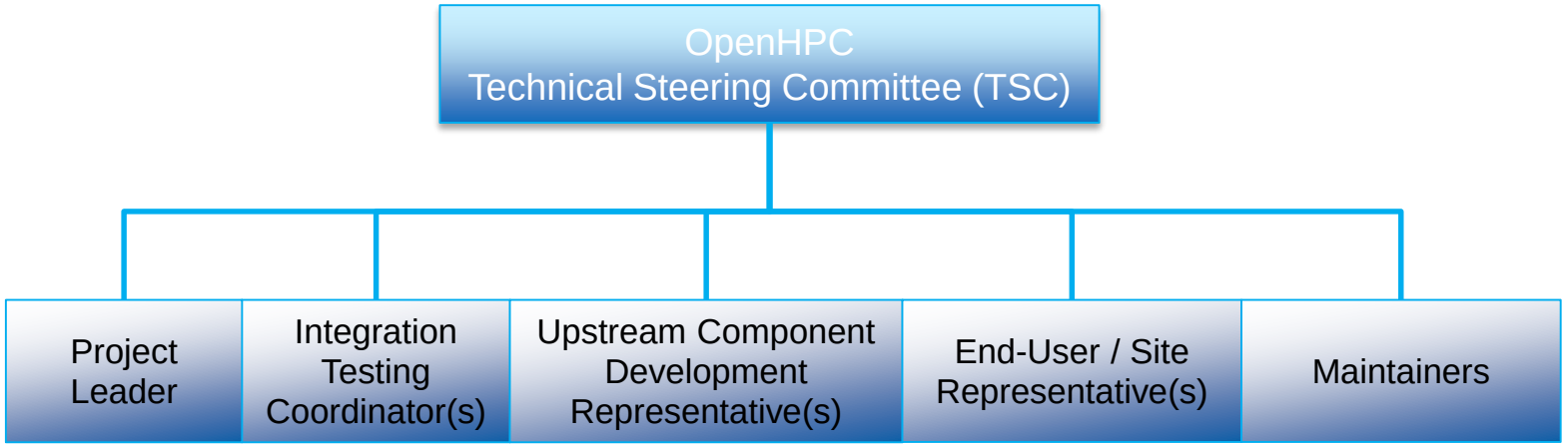
Governing Board + Technical Steering Committee



LINUX FOUNDATION
COLLABORATIVE PROJECTS

Courtesy of OpenHPC*

OpenHPC TSC - Role Overview



LINUX FOUNDATION
COLLABORATIVE PROJECTS

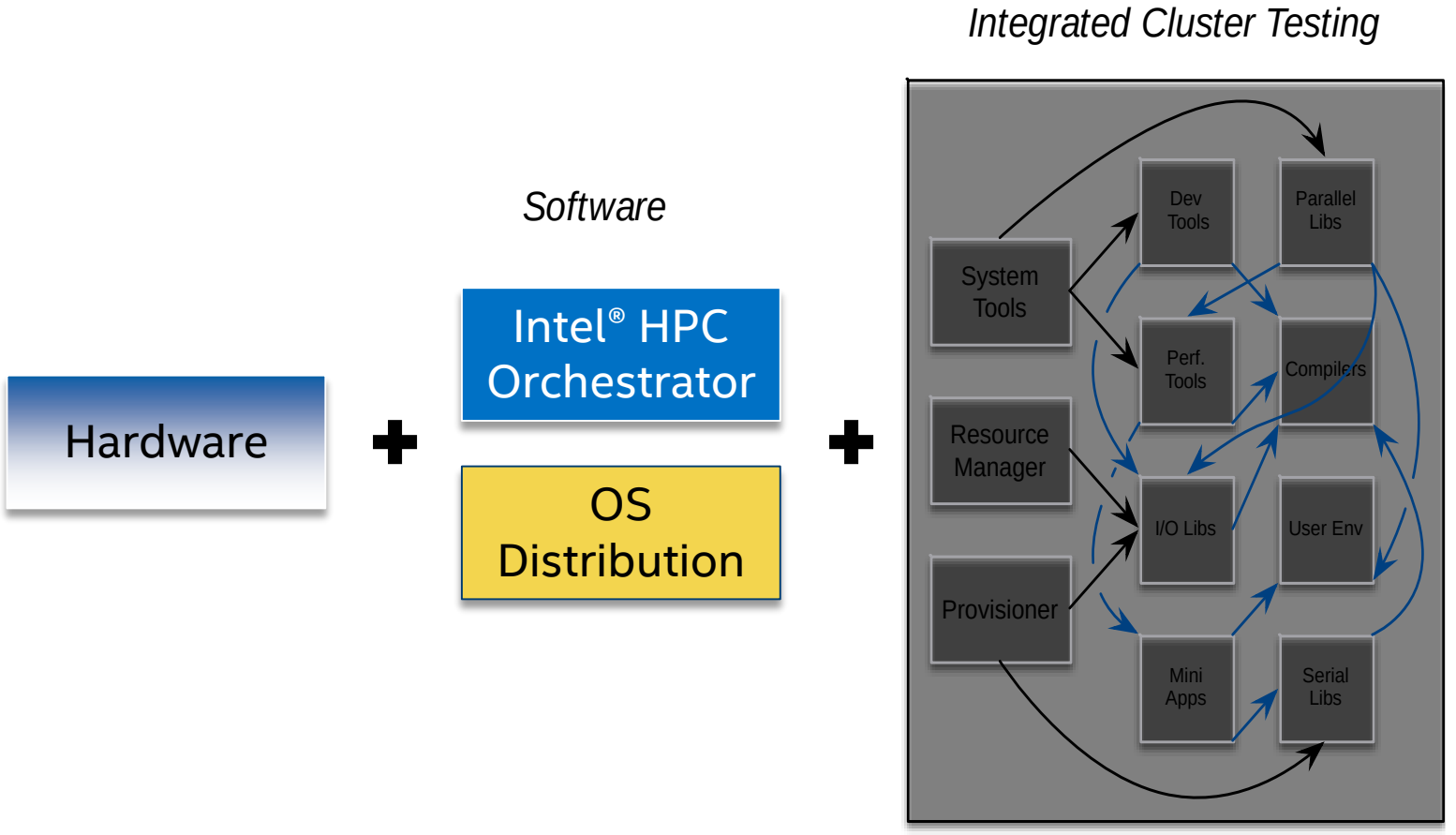
Courtesy of OpenHPC*

intel

Testing and Validation

Integration/Test/Validation

- Cross-package interaction
- Each end-user test need to touch all of the supported compiler & MPI families
- Abstracted to repeat the tests with different compiler/MPI environments:
  - gcc/Intel compiler toolchains
  - Intel MPI, OpenMPI, MVAPICH2



Post Install Integration Tests - Overview

- Major components have configuration options to enable/disable
- Suite of integration tests
  - Installed by default by the install script
  - Root-level tests
  - User-level tests: short & long versions

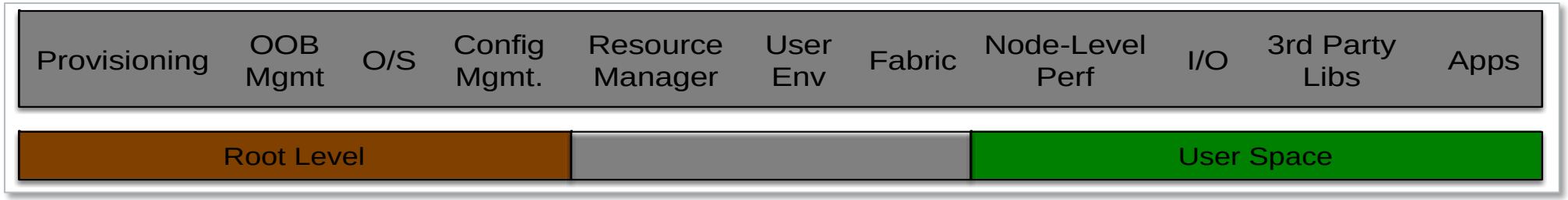
```
>>>------------Launching Integration Testsuite----------------
/home/test/jenkins

HPC Orchestrator Integration Test Configuration:

    Root Level Testing = true
    User Level Testing = true
    Enable Long Tests  = true

Running Root-Level Tests
```

Type of test	Approximate running time
Root-level	single digit number of minutes
User-level short	less than 30 minutes
User-level long	a few hours



Post Install Integration Tests – Root level

```
----------------------------------------- SUMMARY -----------------------------------------

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

Build user................... : root
Build host................... : sms002
Configure date............... : 2016-11-07 04:51
Build architecture........... : x86_64-unknown-linux-gnu
Test suite configuration..... : short

Submodule Configuration:

    Base operating system..... : enabled
    Out of band tools......... : enabled
    Hardware benchmarks....... : disabled
    Cluster checker........... : enabled
    Lustre client............. : disabled
    spack..................... : enabled
-------------------------------------------------------------
```

```
PASS: admin/run
PASS: bos/run
PASS: oob/run
PASS: clck/run
PASS: admin/spack/run
============================================================================
Testsuite summary for test-suite 1.0.0
============================================================================
# TOTAL: 5
# PASS:  5
# SKIP:  0
# XFAIL: 0
# FAIL:  0
# XPASS: 0
# ERROR: 0
============================================================================
```

Post Install Integration Tests – User level

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

Build user................... : orchtest
Build host................... : sms002
Configure date............... : 2016-11-07 04:56
Build architecture........... : x86_64-unknown-linux-gnu
Test suite configuration..... : long

Submodule Configuration:

User Environment:
    Packaging tests.......... :
    RMS test harness......... :
    Munge.................... :
    Compilers................ :
    MPI...................... :
    Modules.................. :
    OOM...................... :
Dev Tools:
    Autotools................ :
    EasyBuild................ :
    Valgrind................. :
    R base package........... :
    CILK..................... :
Performance Tools:
    mpiP Profiler............ :
    Papi..................... :
    TAU...................... :
```

```
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
```

```
============================================
Testsuite summary for test-suite 1.0.0
============================================
# TOTAL: 39
# PASS:  39
# SKIP:  0
# XFAIL: 0
# FAIL:  0
# XPASS: 0
# ERROR: 0
============================================
```

Legal Disclaimer and Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

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.

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.

Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804