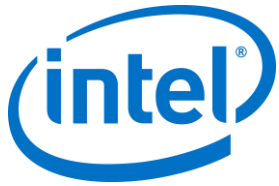

Intel HPC Platform Specification

April 2, 2019
Version 2018.0

Chapter 0. Introduction

0.1. Overview

The Intel HPC Platform Specification defines both software and hardware requirements that form foundations for high performance computing solutions. This document takes a modular approach to defining these requirements and allows application developers to work on a known base solution without knowing the specifics of a system. This approach promotes application interoperability and execution on any solution that complies with a given set of requirements. This specification targets the community working with scalable systems, including designers and builders of clusters, programmers, system administrators, and users.

Each section in this document defines a set of requirements. A reference architecture is a defined set of sections that, when combined, create the basis for a solution. While a reference architecture requires a specific set of sections, additional optional sections may be added to further customize the system. Figure 1 below shows an example reference architecture (the group of sections A-D) with optional sections that allow customization.

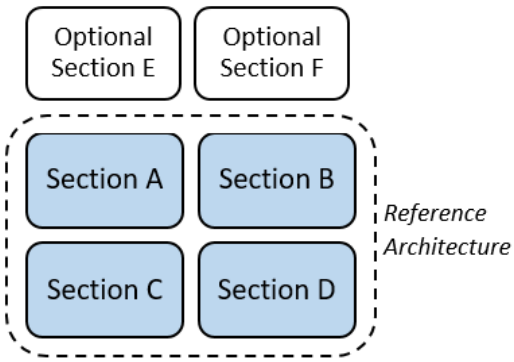

Figure 1: Example of a reference architecture

This platform specification documents industry best practices for Intel-based solutions across a wide domain of application spaces. Currently, this document includes the reference architecture for a classic HPC cluster, but over time, it will also include a family of reference architectures that span small systems through supercomputers. Future reference architectures may support cloud, data analytics, and machine learning in addition to traditional HPC workloads. Chapter 1 contains the core requirements common to all reference architectures, while section 2.1 defines the requirements for a classic HPC cluster. Chapters 3 and 4 define capabilities and components that can enhance a reference architecture.

An implementation may choose the sections to which it was conformant, subject to dependencies between sections. However, an implementation claiming conformance to a section must satisfy all the requirements of the section. In cases where requirements overlap between sections, the implementation must conform to the most restrictive requirement. Each section has a corresponding identifier that indicates conformance with the section requirements (see section 1.1.1 for details). These identifiers allow applications, administrators, and users to discover the capabilities of a given solution implementation.

0.1.1. Conventions

A. Wording

The key words/phrases "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119[1].

Text with a gray background is advisory.

This is advisory text.

B. Units

This document has standardized on using binary units for values representing computational values in bytes and bits. Binary units are defined by IEC 80000-13:2008 and IEEE 1541-2002. These definitions include the Ki (kibi- 210), Mi (mebi- 220), and Gi (gibi- 230) prefixes.

The computing industry uses decimal units for both decimal and binary values, leading to confusion of actual values. At small quantities, the difference between binary and decimal units is minimal, i.e. 1.024 KB = 1 KiB. However, larger values introduce significant discrepancy, i.e. 1 TiB is approximately 10% greater than 1 TB.

As a technical specification, all defined values must be precise; therefore, this document uses binary units when the underlying architecture uses binary format. This includes, but is not limited to, memory, persistent storage, and data transfer. An exception will be made when referencing other specifications, if values in those specifications accurately use different units.

When this document uses SI decimal units (K, M, G, etc.), values will always be powers of 10.

C. Power States

This document shall refer to system power states using ACPI (Advanced Configuration and Power Interface) terminology. Power states are defined under the ACPI Specification 6.2. errata A[2]

The following power states may be referenced in the Intel HPC Platform Specification requirements.

- S0: Working. System is powered on. Also G0.
- S1-S4: System sleeping states. The system is in a lower-power state, but retains context allowing it to resume. In particular, the S1 state retains full context for low-latency resumption of operations.
- S5: System is powered-off, with no saved context, and a full boot sequence is required to restore operation. Power to the system is maintained, allowing for out-of-band operations.
- G3: System is unpowered, except for RTC battery.

[1] RFC 2119: https://www.ietf.org/rfc/rfc2119.txt

[2] ACPI Specification 6.2. errata A: http://www.uefi.org/sites/default/files/resources/ACPI%206_2_A_Sept29.pdf

0.1.2. Terminology

COTS

Commercial off-the-shelf (COTS) components are standard manufactured products, not custom products.

Compute node

A compute node is a node reserved for running user workloads and is the primary computational resource of a system.

Distribution or system software distribution

A system software distribution comprises an operating system kernel, user-space tools and libraries, as well as the documentation. The components of the distribution may originate from different sources, but they are packaged together and released by the distribution supplier; the distribution supplier may be a commercial entity or community based.

External node

An external node provides resources to the system but is not managed as part of the system.

Fully-qualified path name

A fully-qualified path name is the full path of a directory or file, including the root directory in the virtual filesystem hierarchy.

Head node

Some systems combine the logical functions of login, management, and/or service nodes into a single physical system known as the head node.

Interconnected nodes

If one node is capable of communicating to another node (directly or indirectly with switch hops) via the fabric, then they are interconnected.

Job

A job is a user workload running on one or more allocated compute nodes. Depending on the system configuration, a job may be launched as a script submitted to the resource manager / scheduler or may be launched directly from the login node.

Login node

A login node is typically the primary point for user interaction with the system. Users initiate jobs from this node, either by submitting jobs to the resource manager or by directly starting job execution. The login node may be shared with multiple users and serves as an interactive resource to stage input and output files, monitor workloads, and build software.

Managed sensor

A hardware device that provides environmental or event data to Management Control Services.

Managed system

A physical device, such as a switch, server, or appliance that is the target of Management Control Services. Depending on functionality provided, Management Control Services may read information, set configuration, or remotely control a Managed System.

Management channel

A logical communication link from Management Control Services to a Managed System. The physical interface may be dedicated to the management channel or may be shared with other communication protocols. Multiple management channels may exist on the same physical interface.

Management control services

Software or firmware responsible for issuing management commands and for receiving alerts. These may be multiple services or a single service providing all required functions. Nodes that run these services are often referred to as management nodes. In a cluster, management nodes are a type of service node typically used to perform low-level system management and monitoring.

Node

A node is a single system, comprised of processor(s), memory, and network interface(s) all under the view and control of a single Operating System image.

Primary channel or Primary management channel

The default management channel providing full administrative privileges and management capabilities.

Service node

A service node provides non-computational resources to the system in support of a job. Users do not typically access a service node directly, but instead through the service that the node supplies.

User-accessible node

Any node where a user can directly start processes. User access may be temporarily granted in cases where a resource manager is used to schedule user workloads.

User environment management

User environment management provides a way to manage the execution environment of a user.

For example, user environment management may adapt the PATH and LD_LIBRARY_PATH variables to meet the needs of a given application or sets of applications.

Chapter 1. Core

1.1. Core

1.1.1. Configuration Information and Compliance

A. Every node shall provide a text file with the pathname "/etc/intel-hpc-platform-release" containing the `INTEL_HPC_PLATFORM_VERSION` field using POSIX* shell variable assignment syntax. The value of the field shall be a colon separated list of section identifiers indicating the sections to which the implementation conforms.

Advisory

The value of `INTEL_HPC_PLATFORM_VERSION` contained in the /etc/intel-hpc-platform-release file provides the mechanism for a system to list compliance to specific versions of each section. `INTEL_HPC_PLATFORM_VERSION` should be updated accordingly to reflect changes in compliance as a system is modified over time.

B. The `INTEL_HPC_PLATFORM_VERSION` identifier for this section is `core-2018.0`.

1.1.2. Hardware

A. Each compute node shall have at least one Intel® 64 architecture processor.

Advisory

This section contains the minimum hardware requirements. Additional hardware may be necessary for a fully functional node.

1.1.3. Operating System and Kernel

A. The compute node kernel shall be based on Linux* kernel version 3.10.0 or later.

Advisory

The compute node kernel version should be based on functionality in Linux* kernel version 4.15 or later.

Major commercial and popular Linux* distributions backport kernel features and functionality while maintaining version strings corresponding to an earlier kernel base. The intent of this requirement is to ensure a common base that matches these popular distributions without requiring an exception list for distributions that maintain older version strings. It is up to the distribution to provide a kernel built with appropriate flags for security, hardware functionality, or system features. It is up to the system provider to select a distribution that includes the desired kernel features.

B. Login and service node kernels shall be based on Linux* kernel version 3.10.0 or later.

Advisory

The login and service nodes kernel version should be based on functionality in Linux* kernel version 4.15 or later.

The Linux* kernel requirements for login and service nodes is separated from compute nodes to permit solutions that use a mix of OS environments. For example, a solution may use commercial distributions for login and service nodes with a comparable open source distribution for compute nodes. In general, the same advisories for compute nodes apply to the kernel requirements for login and service nodes.

1.1.4. Programming Interfaces

A. All required APIs shall be defined using the LP64[3] programming model on all nodes.

Advisory

Where explicitly required for compatibility with existing practice or current program binaries, other programming model APIs may be provided in addition to LP64.

B. The ABI behavior shall conform to the Intel® 64 architecture on all nodes.

Advisory

The Intel® 64 architecture ABI is also known as 'x86_64'.

Where explicitly required for compatibility with existing practice or current program binaries, other ABIs may be provided in addition to Intel® 64 architecture.

1.1.5. Runtime Environment

A. At login, the environment variable `$HOME` shall be set on all nodes to the fully-qualified pathname of the user's home directory.

B. The environment variable `$TMPDIR` shall be set on all nodes to the fully-qualified pathname of a node-private temporary directory. If the temporary directory is contained within a shared filesystem, then the value of `$TMPDIR` shall be unique to each node.

Advisory

The temporary directory is not required to be local or persistent. On user accessible nodes, the environment variable `$SCRATCH` should be set to the fully-qualified pathname of a persistent directory on a high performance filesystem to host temporary files for user workloads.

The value of `$TMPDIR` may be set dynamically as long as the value is set and meets the requirement for the duration that a non-privileged user or job has access to the node.

C. User environment management shall be used to allow multiple runtime environments to coexist on user accessible nodes.

Advisory

This includes multiple versions of the same component.

It is left to the implementer to choose which runtime environments are default, if any.

[3] LP64 Programming Model: http://www.unix.org/version2/whatsnew/lp64_wp.html

The module naming convention should use the name and version of the component, e.g., `impi/2018.3.222`. Where a version is not applicable, the version may be omitted. A hierarchical organizational scheme should be used to handle dependencies, e.g., on a compiler family.

1.2. Core Intel Runtime Environment

The runtime environment contains many requirements to ensure that functionality and performance opportunities are present. This section is separated from the core layer as the runtime environment may update at a different frequency.

This section enumerates the requirements for baseline interfaces readily available in common Linux distributions and key Intel software runtime components available free of charge. These requirements are the minimum set of elements that must be present to be compliant.

1.2.1. Configuration Information and Compliance

A. The `INTEL_HPC_PLATFORM_VERSION` identifier for this section is core-intel-runtime-`2018.0`.

B. If an implementation claims compliance to this section, then `INTEL_HPC_PLATFORM_VERSION` must also contain the `core-2018.0` section identifier and meet all corresponding requirements.

1.2.2. Operating System and Kernel

Advisory

The operating system on user accessible nodes should materially conform to the Linux* Standard Base 5.0 core specification[4].

1.2.3. Programming Interfaces

Advisory

Some applications may require additional or newer versions of these runtimes for compatibility.

A. A materially conformant POSIX.1-2008 API[5] shall be provided on user accessible nodes.

B. The following Linux Standard Base* (LSB) 5.0 runtime libraries[6] shall be provided on user accessible nodes and recognized by the dynamic loader for the Intel® 64 architecture:

LIBRARY	RUNTIME NAME
libc	libc.so.6
libcrypt	libcrypt.so.1
libdl	libdl.so.2
libgcc_s	libgcc_s.so.1
libm	libm.so.6

[4] Linux Standard Base 5.0: http://refspecs.linuxfoundation.org/LSB_5.0.0/allspecs.shtml

[5] POSIX.1-2008: http://pubs.opengroup.org/onlinepubs/9699919799/

[6] Linux Standard Base Core 5.0 Core Specification: http://refspecs.linuxfoundation.org/LSB_5.0.0/LSB-Core-generic/LSB-Core-generic/requirements.html, http://refspecs.linuxfoundation.org/LSB_5.0.0/LSB-Core-AMD64/LSB-Core-AMD64/requirements.html
Linux Standard Base 5.0 Desktop Specification: http://refspecs.linuxfoundation.org/LSB_5.0.0/LSB-Desktop-generic/LSB-Desktop-generic/requirements.html

libncurses	libncurses.so.5
libncursesw	libncursesw.so.5
libpam	libpam.so.0
libpthread	libpthread.so.0
librt	librt.so.1
libstdcxx	libstdc++.so.5, libstdc++.so.6
libutil	libutil.so.1
libz	libz.so.1
proginterp	/lib64/ld-linux-x86-64.so.2, /lib64/ld-lsb-x86-64.so.3

Advisory

Where explicitly required for compatibility with existing practice or current program binaries, other runtime libraries and ABIs may be provided in addition to Intel® 64 architecture.

The following additional runtime libraries should be provided on all user accessible nodes and recognized by the dynamic loader for the Intel® 64 architecture:

- libBrokenLocale.so.1
- libSegFault.sols
- libanl.so.1
- libacl.so.1
- libattr.so.1
- libbz2.so.1
- libcap.so.2
- libcrypto.so.6
- libnsl.so.1
- libnss_compat.so.2
- libnss_dns.so.2
- libnss_files.so.2
- libnss_hesiod.so.2
- libnss_ldap.so.2
- libnss_nis.so.2
- libnuma.so.1
- libpanel.so.5
- libpanelw.so.5
- libresolv.so.2
- libthread_db.so.1

C. The LP64 version of the following runtime libraries shall be provided on user accessible nodes and with runtime environments configurable using user environment management:
   a. ANSI* standard C/C++ language runtime of the GNU* C Compiler version 4.8 or later
   b. ANSI* standard C/C++ language runtime of the Intel® C++ Compiler version 18.0 or later
   c. Standard Fortran language runtime of the Intel® Fortran Compiler version 18.0 or later
   d. Intel® Math Kernel Library version 2018.0 or later
   e. Intel® Threading Building Blocks version 2018 or later
   f. Intel® MPI Library Runtime Environment version 2018 or later
   g. The Intel® Distribution for Python* scripting language version 2018 or later

Advisory

User environment management should load these runtime libraries by default.

For each component, the runtimes are defined to include all of the runtime libraries distributed with the component. E.g., the OpenMP* runtime library is included as part of the Intel® C++ Compiler runtime.

The identified Intel runtime components above are provided without fee.

1.2.4. Command System and Tools

A. The following subset of the Linux Standard Base* (LSB) 5.0 command system[7] shall be provided on all user accessible nodes:

[	date	find	ls	printf	tee
ar	dd	fold	mkdir	ps	test
awk	diff	fuser	mkfifo	pwd	time
basename	dirname	getconf	mktemp	rm	touch
bc	du	grep	more	rmdir	tr
cat	echo	head	mv	sed	true
chmod	ed	hostname	nice	seq	uname
chown	egrep	iconv	nl	sh	uniq
cksum	env	id	nohup	sleep	vi
cmp	ex	join	od	sort	wc
comm	expr	kill	paste	split	xargs
cp	false	killall	patch	strings	
csplit	fgrep	ln	pathchk	tail	
cut	file	logname	pidof	tar	

Advisory

A complete and materially conformant POSIX.1-2008* command system[8] should be provided on all user accessible nodes.

A complete and materially conformant Linux Standard Base* (LSB) 5.0 command system[7] should be provided on all user accessible nodes.

B. The Perl* scripting language version 5.16 or later shall be provided on user accessible nodes.

C. The Tcl scripting language version 8.5 or later shall be provided on user accessible nodes.

Advisory

The requirement for Tcl scripting language may be removed in future versions of the specification.

[7] Linux Standard Base 5.0 Core Specification: http://refspecs.linuxfoundation.org/LSB_5.0.0/LSB-Core-generic/LSB-Core-generic/command.html

[8] POSIX.1-2008 Utilities: http://pubs.opengroup.org/onlinepubs/9699919799/idx/utilities.html

Chapter 2.
Base Reference Architectures

2.1. Classic High Performance Compute Cluster

The “classic” High Performance Compute cluster typically uses COTS components to form a parallel computing platform. Applications that target this type of system typically use the Message Passing Interface (MPI) for parallel execution.

Typical clusters often employ a head node that serves to manage the system, is the primary login/interface for users, and provides numerous system-wide functions and utilities. Compute nodes based on Intel® 64 architecture processors are the primary computational resource. A user workload running on one or more allocated compute nodes is known as a job. Additional nodes may provide specific, non-compute cluster services, such as login nodes, storage nodes, etc. Typically only the login node(s) are directly accessible by users.

These clusters have at least three distinct communications needs: application messaging, system management, and cluster-wide storage. From an architectural standpoint, each of these networks has distinct logical requirements. However, implementations may choose to combine one or more of these logical networks in a single physical fabric.

2.1.1. Classic High Performance Compute Cluster Base

A. Configuration Information and Compliance
   a. The `INTEL_HPC_PLATFORM_VERSION` identifier for this section is `hpc-cluster-2018.0`.
   b. If an implementation claims compliance to this section, then `INTEL_HPC_PLATFORM_VERSION` must also contain the `core-2018.0` and `core-intel-runtime-2018.0` section identifiers and meet all corresponding requirements.

B. Operating System and Kernel

Advisory

In most configurations, the hardware components and software environment of compute nodes should be uniform. Files that specify unique identification or configuration of the compute nodes may be different as needed.

In cases where specialized nodes are desired, a resource manager should be provided and should comprehend any differences.

C. Network Fabrics
   a. Each compute node’s network host name shall be consistently resolved to its network address.
   b. The management fabric shall be accessible using the standard IP network stack.
   c. All login nodes shall be externally accessible via SSH.

Advisory

Login nodes are generally connected to an externally accessible network. Stand-alone systems that are not connected to an externally accessible network, that is, systems that only allow console access to the login node, are considered to meet this requirement.

D. Authentication and Access Control
   a. For non-privileged users, all nodes shall operate under a single authentication domain, i.e., once authenticated, one set of credentials shall permit access to the cluster.

Advisory

This requirement is applicable to the resources allocated to an individual user or user's jobs. Local policies control user access to the system, and this requirement is not meant to dictate those polices. This requirement does not apply to privileged users.

b. Privileged users shall be able to execute commands on all nodes.
c. Unprivileged users or their jobs shall be able to execute commands on all currently allocated compute nodes.

Advisory

Unprivileged users are not required to be able to access a compute node unless they have been allocated resources on that node.

Unprivileged users are not required to have interactive access to a compute node even if they have been allocated resources on that node.

d. Unprivileged users shall be able to access their data stored locally on currently allocated compute nodes.

Advisory

Unprivileged users are not required to be able to access data stored locally on a compute node unless they have been allocated resources on that node.

2.1.2. High Performance Compute Cluster Application Compatibility

An ecosystem of highly compatible "classic" HPC clusters provides a consistent application target for application developers. Solutions that comply with this section present a known interface to the application layer. In turn, application developers and vendors can compile and distribute binaries for this target platform, enabling application binary mobility.

While this section is intended primarily for MPI applications distributed as binaries, it is also applicable for MPI applications built from source on the intended system as well as non-MPI workloads.

A. Configuration Information and Compliance
   a. The `INTEL_HPC_PLATFORM_VERSION` identifier for this section is `compat-hpc-2018.0`.
   b. If an implementation claims compliance to this section, then `INTEL_HPC_PLATFORM_VERSION` must also contain the `hpc-cluster-2018.0` section identifier and meet all corresponding requirements.

B. Hardware

Advisory

Minimal hardware requirements are described to ensure functional systems are built from this platform definition. This specification does not guarantee that specific implementations built only to these minimal requirements will provide optimal application performance. Implementers must determine when additional hardware resources beyond this set of minimums may be required to provide optimal application performance.

   a. Each compute node shall have a minimum of 64 gibibytes of random access memory.

      Advisory

      A total of 96 gibibytes of random access memory or more per node is recommended.

      All memory channels of a server should be populated using identical DIMMS to provide optimal performance. This may correspond to an amount of memory that is higher than the requirement.

   b. Each compute node shall have access to at least 80 gibibytes of persistent storage.

      Advisory

      The storage may be implemented as direct access local storage or available over a network.

      The storage may be globally visible or node private.

   c. Login nodes shall have at least 200 gibibytes of persistent storage.

      Advisory

      The storage may be implemented as direct access local storage or available over a network.

C. Programming Interfaces

Advisory

Some applications may require additional or newer versions of these runtimes for compatibility.

a. The following Linux Standard Base* (LSB) 5.0 runtime libraries[9] shall be provided on user accessible nodes and recognized by the dynamic loader for the Intel® 64 architecture:

LIBRARY	RUNTIME NAME
libGL	libGL.so.1
libGLU	libGLU.so.1
libICE	libICE.so.6
libSM	libSM.so.6
libX11	libX11.so.6
libXext	libXext.so.6
libXft	libXft.so.2
libXi	libXi.so.6
libXrender	libXrender.so.1
libXt	libXt.so.6
libXtst	libXtst.so.6
libfontconfig	libfontconfig.so.1
libfreetype	libfreetype.so.6
libjpeg	libjpeg.so.62
libxcb	libxcb.so.1

Advisory

Where explicitly required for compatibility with existing practice or current program binaries, other runtime libraries and ABIs may be provided in addition to Intel® 64 architecture.

The following additional runtime libraries should be provided on all user accessible nodes and recognized by the dynamic loader for the Intel® 64 architecture:

- libXau.so.6
- libXcursor.so.1
- libXfixes.so.3
- libXinerama.so.1
- libXmu.so.6
- libXp.so.6
- libXrandr.so.2
- libXxf86vm.so.1
- libpng12.so.0

D. Storage and File System

a. Cluster file systems shall provide at least the consistency and access guarantees provided by NFS version 3.0[10].

Advisory

[9] Linux Standard Base 5.0 Desktop Specification: http://refspecs.linuxfoundation.org/LSB_5.0.0/LSB-Desktop-generic/LSB-Desktop-generic/requirements.html

[10] Network File System (NFS) version 3 Protocol (RFC 1813): https://www.ietf.org/rfc/rfc1813.txt

Cluster file systems should provide at least the consistency and access guarantees provided by NFS version 4.0[11].

Cluster file systems should support POSIX.1-2008 file semantics[12].

[11] Network File System (NFS) version 4 Protocol (RFC 3530): https://www.ietf.org/rfc/rfc3530.txt

[12] POSIX.1-2008 System Interfaces: http://pubs.opengroup.org/onlinepubs/9699919799/functions/V2_chap02.html

Chapter 3. Capabilities

3.1. High Performance Fabric

High performance network fabric is the interconnect technology that connects systems through one or more network switches. While standard Ethernet network fabric continues to improve, high performance fabric is defined in this context as technology providing capabilities beyond the standard TCP/IP or UDP/IP protocol to deliver higher bandwidth, lower latency, or both. High performance fabric is a key technology used in high performance computing, but can be desirable in many other types of solutions. This section defines minimum requirements for solutions that contain high performance fabric technology.

3.1.1. Configuration and Compliance Information

A. The `INTEL_HPC_PLATFORM_VERSION` identifier for this section is `high-performance-fabric-2018.0.`

B. If an implementation claims compliance to this section, then `INTEL_HPC_PLATFORM_VERSION` must also contain the `core-2018.0` section identifier and meet all corresponding requirements.

3.1.2. Open Fabric Interface

A. Each node shall include the OpenFabric Interfaces* (OFI) `libfabric` library version 1.4.0 or greater.

B. The version of `libfabric` used by interconnected nodes shall be consistent.

Advisory

Each node should include `libfabric` version 1.6.0 or later.

3.1.3. Remote Direct Memory Access (RDMA)

Every node shall support remote direct memory access (RDMA) to interconnected nodes.
The following packages shall be installed on every node to support RDMA:

A. `rdma-core` version 13 or greater

Advisory

`rdma-core` is a recent package that is now the upstream location for "user space components for Linux Kernel's drivers/infiniband subsystem."[13]

B. `infiniband-diags` version 1.6.4 or greater

C. All nodes with interconnected fabric shall provide the same version of the packages listed above.

3.1.4. Firmware

The firmware shall be consistent across interconnected nodes in the system for the following hardware:

A. Host Fabric Network Device

B. Fabric switches

3.1.5. TCP/IP and UDP/IP Capabilities

The fabric shall support standard TCP/IP and UDP/IP capabilities, providing a functional protocol adhering to the following:

[13] https://github.com/linux-rdma/rdma-core

A. The standard IP network stack functionality as described in RFC1180[14]

B. Berkeley Sockets API (BSD Sockets)

Advisory

The fabric should have standard TCP/IP and UDP/IP capabilities configured and enabled.

3.1.6. Subnet Management

For fabrics that require subnet management, there shall be active subnet management visible to all host fabric network devices.

Advisory

Fabrics that require subnet management should have access to a backup subnet manager.

[14] RFC1180: https://tools.ietf.org/html/rfc1180

3.2. Software Defined Visualization

A node that provides capability for Software Defined Visualization (SDVis) typically contains off-the-shelf hardware components. Applications for SDVis use any or all of the following open source libraries as a basis: the Intel® Rendering Framework:Embree Ray Tracing Kernel Library, the Intel® Rendering Framework:OSPRay Distributed Ray Tracing Infrastructure library, and/or the Intel® Rendering Framework:OpenSWR OpenGL Software Rasterizer which is fully integrated into the Mesa libraries.

Advisory

A service or login node may provide the capabilities to drive at least 6 displays with a resolution of 4096x2160 pixels per display.

3.2.1. Software Defined Visualization Base Capabilities

A. Configuration and Compliance Information

a. The `INTEL_HPC_PLATFORM_VERSION` identifier for this section is `sdvis-core-2018.0`.

b. If an implementation claims compliance to this section, then `INTEL_HPC_PLATFORM_VERSION` must also contain the `core-2018.0` and `core-intel-runtimes-2018.0` section identifiers and meet all corresponding requirements.

B. Runtime Environment

Each node shall provide the following libraries contained in the listed implementation version or higher, which includes the list of Mesa libraries with integrated Intel® Rendering Framework:OpenSWR support.

IMPLEMENTATION	LIBRARY NAME	DESCRIPTION
`libdrm 2.4.92`	`libdrm.so.2.4.0`	Cross-driver middleware which allows user-space applications (such as Mesa) to communicate with the Kernel by the means of the DRI protocol.
	`libdrm_amdgpu.so.1.0.0`	
	`libdrm_intel.so.1.0.0`	
	`libdrm_nouveau.so.2.0.0`	
	`libdrm_radeon.1.0.1`	
	`libkms.so.1.0.0`	
`Mesa 18.1.0`	`libswrSKX.so.0.0.0`	Mesa libraries with integrated and enabled Intel® OpenSWR support. Mesa is an open-source implementation of various graphics APIs, e.g. OpenGL
	`libswrAVX2.so.0.0.0`	
	`libswrAVX.so.0.0.0`	
	`libglapi.so.0.0.0`	
	`libGL.so.1.5.0`	
	`libOSMesa.so.8.0.0`	

Advisory

It is advised to use the latest available version of the required projects.

C. Visualization Libraries

Visualization capabilities require that the following libraries be loaded by default in the environment of a user:

a. `embree` version 2.17.4 or later in the 2 series

b. `embree` version 3.2.0 or later in the 3 series

c. `ospray` version 1.6.0 or later.

Advisory

It is advised to use the latest available version of the required packages.

3.2.2. Software Defined Visualization Single Node Capabilities

This section defines requirements for a single node that serves as the primary interface for the users, providing all required system-wide functions, utilities, and software tools.

A. Configuration and Compliance Information
   a. The `INTEL_HPC_PLATFORM_VERSION` identifier for this section is `sdvis-single-node-2018.0`.
   b. If an implementation claims compliance to this section, then `INTEL_HPC_PLATFORM_VERSION` must also contain the `sdvis-core-2018.0` section identifier and meet all corresponding requirements.

B. Hardware requirements
   a. The single node shall use processors that support Intel® Hyper-Threading Technology and have that capability enabled.
   b. The node shall use processors that provide at least two 512-bit Fused-Multiply-Add (FMA) execution units per core.
   c. There shall be a minimum of one memory module per processor memory channel installed.
   d. A minimum of 3.5 gibibytes of random access memory per processor core and a minimum of 64 gibibytes of total random access memory shall be installed.
   e. A minimum of 4 tebibytes of persistent storage shall be available to the node.

3.2.3. Software Defined Visualization Multi-Node Capabilities

This section defines requirements for Software Defined Visualization multinode capabilities.
The capabilities provided by the Intel® Rendering Framework libraries are the basis for the advanced scientific visualization capabilities provided by VTK, ParaView and VisIt.

A. Configuration and Compliance Information
   a. The `INTEL_HPC_PLATFORM_VERSION` identifier for this section is `sdvis-cluster-2018.0`.
   b. If an implementation claims compliance to this section, then `INTEL_HPC_PLATFORM_VERSION` must also contain the `compat-hpc-2018.0`, `high-performance-fabric-2018.0` and `sdvis-core-2018.0` section identifiers and meet all corresponding requirements

B. Hardware requirements
   a. All nodes that support SDVis workloads shall use processors that support Intel® Hyper-Threading Technology and have that capability enabled.
   b. The login node shall have a minimum of 6 gibibytes of random access memory per processor core, with a minimum of 192 gibibytes of total random access memory.
   c. Each compute node shall have a minimum of 2.5 gibibytes of random access memory per processor core, with a minimum of 96 gibibytes of total random access memory.
   d. A minimum of 10 tebibytes of shared fault-tolerant storage shall be accessible to all nodes allocated to a job.

C. Software Requirements

   An implementation of this section shall provide the following packages in the default environment of a user:

a. VTK version 8.1.1 or greater
b. `ParaView` version 5.5.0 or greater
c. `VisIt` version 3.0.0 or greater

Advisory

It is advised to use the latest available version of the required packages.

3.3. System Management

The system management interface provides autonomous monitoring and control. System hardware is accessed independently of the operating system, processor(s), or memory on the system; this is referred to as out-of-band (OOB) management. This provides management and monitoring of the target system without impact to computational performance, as well as system control prior to boot and when the OS is non-functional.

Normally, management capability is provided by a dedicated management processor on the Managed System's mainboard. OOB system control must remain operational even when the main CPU is non-responsive, so that the system can be remotely reset.

While there are many server management implementations, the IPMI (Intelligent Platform Management Interface) is widely implemented by server hardware vendors. All requirements in this section are either mandatory or optional features of IPMI, but use of a management solution with similar capabilities is not precluded.

It is expected that future implementations of this section will specifically require conformance to Redfish*, a successor to IPMI, as defined by the Desktop Management Taskforce* (DMTF).

3.3.1. Configuration and Compliance Information

A. The `INTEL_HPC_PLATFORM_VERSION` identifier for this section is `system-management-2018.0`

B. If an implementation claims compliance to this section, then `INTEL_HPC_PLATFORM_VERSION` must also contain the `core-2018.0` section identifier and meet all corresponding requirements.

3.3.2. Management Control Services Requirements

A. There shall be a minimum of 1 node hosting management control services.

Advisory

Management control services should not be hosted on a compute or enhanced compute node.

Additional management control services may operate outside of the cluster. External management control services may duplicate or enhance, but not replace, required functionality.

Implementations on Managed Systems should conform to IPMI v2.0 revision 1.1, including Errata 7, published on April 21, 2015.

Managed Systems should conform to Redfish* 2017.2 (Redfish Specification 1.3.0). This will include the following:

a. Conformance to Host Interface Specification 1.0

b. A minimum of one TCP/IP-enabled interface supporting HTTP Redfish.

c. A recommendation that the Managed System support HTTP and SMBIOS Type 42.

d. Storage device inventory and management (Swordfish*)

e. Read system utilization

f. Support for translation of IPMI to Redfish (PSME)

B. Each node running Management Control Services shall be able to open a session and issue commands to any Managed System's primary channel.

C. Session control and commands shall be scriptable using UNIX V7Bourne Shell syntax.

Advisory

BASH (Bourne Again SHell) and POSIX shells meet this requirement.

3.3.3. Managed system requirements

A. There shall be 1 primary channel on each Managed System.

Advisory

There may be additional management channels on a Managed System.

A Managed System should provide direct console access, either for redirection of KVM (keyboard/video/mouse) or redirection of a serial (RS232) interface.

B. The primary channel on all Managed Systems shall exist on a single, logical TCP/IP v4 network.

C. The primary channel on a Managed System shall be accessible using the TCP/IP v4 protocol through at least one of the following:

a. A direct Ethernet or fabric interface on that system

b. An indirect Ethernet or fabric interface via a system acting as a proxy for the target Managed System. The proxy system shall also comply with the requirements in this section.

Advisory

Managed Systems should implement a dedicated network interface used only for out-of-band management traffic.

D. The primary channel on all Managed Systems shall provide the capability to configure each of the following parameters using the DHCP protocol.

a. IP address

b. subnet mask

c. default gateway

E. The primary channel on all Managed Systems is available when the system is attached to a power source, regardless of the power state of the system.

F. The primary channel on all Managed Systems shall provide the capability to open a session with the ability to perform all commands defined in this section.

Advisory

A Managed System should provide a minimum of three user accounts with the ability to set channel privilege level per user. Available privilege levels should include administrator, read-only (RO), and power-control-only.

The Managed System should support at least 4 simultaneous sessions.

The Managed System should support security options, including communication payload encryption, SHA256 authentication algorithms, and a firmware firewall.

G. A Managed System, through its primary channel, shall provide these capabilities to Management Control Services:

   a. Read identification of the managed system.

Advisory

Identification of the managed system may be any cluster-wide unique value, including hardware serial numbers or user-assigned values. The specific definition of identification is system dependent.

   b. Initiate a hard reset
   c. Initiate a soft reset
   d. Initiate a Power off (S5)
   e. Initiate a Power on (S0)
   f. Read the current ACPI state
   g. Read current power state
   h. Activate a visual or audio indicator that allows physical identification of the managed system
   i. Configure the setting for the default system state after a power loss (power restore policy)
   j. Set the default media option to be used the next time the managed system is booted
   k. Read the managed system firmware, uEFI, or BIOS event log.

Advisory

The Managed System should provide the capability to send SNMP (Simple Network Management Protocol) alerts over a management channel.

H. Each Managed System shall provide a report that includes minimally the following:

   a. An inventory of baseboard, processors, and system memory.
   b. Baseboard firmware versions, including BIOS, uEFI, BMC, and on-board Ethernet.

Advisory

The Read System Information capability should have the capability to report:

- An inventory of all Field-Replaceable Units, including storage devices, power supplies, and add-in cards.
- Active operating system name and version

I. Each Managed System shall provide the capability to read all of the following information for each managed sensor:

   a. Sensor Identification
   b. Sensor type and capabilities
   c. Current measurement, with units
   d. Current thresholds set, if thresholds are supported
   e. Sensor self-test results, if the sensor supports self-tests.

J. All functions on the primary channel shall be available on the Managed System that is powered on (S0), in sleep or hibernate state (S1-S4), or soft off (S5) regardless of operating system status.

Advisory

All configuration options for the Managed System primary channel should be readable and configurable directly through a command line shell on that Managed System.

Managed Systems should implement the capability to read a complete list of its supported management commands and capabilities.

Chapter 4. Components

4.1. Second-Generation Intel® Xeon® Scalable Processor

In order to optimally utilize the capabilities of the second-generation Intel® Xeon® Scalable Processors it is advised to use a runtime environment that is optimized for this processor family.

4.1.1. Configuration and Compliance

A. The `INTEL_HPC_PLATFORM_VERSION` identifier for this section is `second-gen-xeon-sp-2019.0.`

B. If an implementation claims compliance to this section, then `INTEL_HPC_PLATFORM_VERSION` must also contain the `core-2018.0` and `core-intel-runtime-2018.0` section identifiers and meet all corresponding requirements.

4.1.2. Programming Interfaces

A. The LP64 version of the following runtime libraries shall be provided on user accessible nodes, with the runtime environment being user configurable using user environment management:

a. ANSI* standard C/C++ language runtime of the Intel® C++ Compiler version 19.2 or later

b. Standard Fortran language runtime of the Intel® Fortran Compiler version 19.2 or later

c. Intel® Math Kernel Library version 2019.2 or later

d. Intel® Threading Building Blocks version 2019.2 or later

e. Intel® MPI Library Runtime Environment version 2019.2 or later

f. The Intel® Distribution for Python* scripting language version 2019.2 or later

Advisory

It is advised to use the latest available version of the required runtime libraries. User environment management should load these runtime libraries by default. For each component, the runtimes are defined to include all of the runtime libraries distributed with the component, e.g., the OpenMP* runtime library is included as part of the Intel® C++ Compiler runtime.
The identified Intel runtime components above are provided without fee.