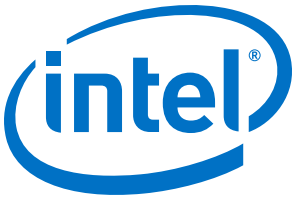

Intel® Scalable System Framework (Intel® SSF) Reference Design

Cluster installation based on OpenHPC* for systems with Intel® Xeon® Processor E5-2699 v4 and Intel® Omni-Path Fabric.

Version 1.1.3

November 8, 2016

Summary

This Reference Design is part of the Intel® Scalable System Framework series of reference collateral.

The Reference Design is a verified implementation example of a given Reference Architecture, complete with hardware and software Bill of Materials information and cluster configuration instructions. It can confidently be used “as is”, or be the foundation for enhancements and/or modifications.

Additional Reference Designs are expected in the future to provide example solutions for existing Reference Architecture definitions and for utilizing additional Intel® SSF elements. Similarly, more Reference Designs are expected as new Reference Architecture definitions are introduced.

This Reference Design is developed in support of the classic HPC cluster Reference Architecture listed below using certain Intel® SSF elements:

- Intel® Scalable System Framework Architecture Specification
- Servers with Intel® Xeon® Processor E5-2699 v4
- Intel® Omni-Path Fabric and Software Stack
- Software stack based on OpenHPC* v1.1

Legal Notices

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

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an older number and are referenced in this document may be obtained by calling 1-800-548-4725 or visiting www.intel.com/design/literature.htm.

Intel and the Intel logo 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.

Contents

Chapter 1

This Reference Design

Section 1.1: Hardware Bill of Materials

Hardware Bill of Materials for the Head Node

Quantity	Item	Manufacturer	Model
1	Intel® Server Chassis	Intel	R2000WT
1	Intel® Server Board (w/10Gb Intel® Ethernet Controller)	Intel	S2600WTT
	(2x) Intel® Xeon® Processor	Intel	E5-2699 v4
	(8x) 8GB ECC DDR4 2133MHz	Micron	MTA18ASF1G72PZ
	Intel® SSD 800GB, 2.5-inch SATA	Intel	S3700 Series
1	Low Latency Gigabit[1] Ethernet Switch	Hewlett-Packard	J2724A ProCurve Switch
1	Intel® Omni-Path Edge Switch	Intel	100 Series

Hardware Bill of Materials for the Compute Node

Quantity	Item	Manufacturer	Model
4	Intel® Server Chassis	Intel	R2000WT
4	Intel® Server Board (w/10Gb Intel® Ethernet Controller)	Intel	S2600WTT
	(2x) Intel® Xeon® Processor	Intel	E5-2699 v4
	(8x) 8GB ECC DDR4 2133MHz	Micron	MTA18ASF1G72PZ

Section 1.2: Software Bill of Materials

Software	Version
CentOS* Linux* Installation DVD	7.2.1511
Intel® Parallel Studio XE Cluster Edition	2017.1
Intel® SSF meta-RPMs for EL7	2016.0
OpenHPC*	1.1
Intel® Omni-Path Fabric Software (Including Intel® Omni-Path Host Fabric Interface Driver)	10.2.0.0.158
Intel® Open Fabrics Interfaces (OFI) Software for Intel® Omni-Path	10.2.0.0.158-93

[1] Cluster tested using a 1GbE switch. It is highly recommended to use a 10GbE switch for optimal performance. An updated reference design will be released in the near future.

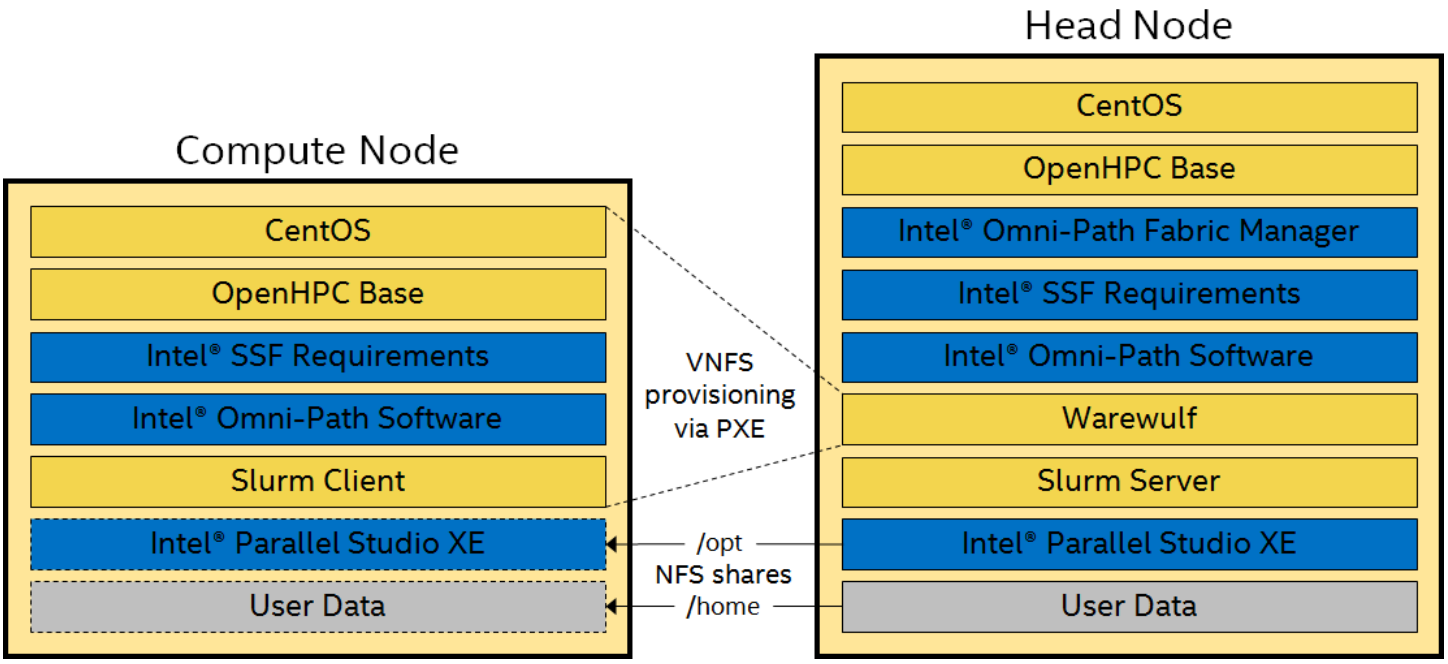

Figure 1.1: An overview of system configuration.

Section 1.3: Conventions

Certain conventions used in this reference design are contingent upon the hardware and software listed in the bill of materials, the document date of release, and developer preference. These conventions are distinguished by red text throughout this document. In theory, other suitable values may be used in place of these conventions.

Cluster/Subcluster

- Internal Subnet (Ethernet): `192.168.1.0`
- Internal Netmask (Ethernet): `255.255.255.0` (/24)
- Internal Subnet (Intel® Omni-Path): `192.168.5.0`
- Internal Netmask (Intel® Omni-Path): `255.255.255.0` (/24)

Head Node

- Hostname: `frontend`
- Domain Name: `localdomain`
- Internal IP Address (Ethernet): `192.168.1.254`
- Internal Network Device (Ethernet): `enp3s0f0`
- External Network Device (Ethernet): `enp3s0f1`
- Hostname (Intel® Omni-Path): `frontend-ib0`
- Internal IP Address (Intel® Omni-Path): `192.168.5.254`
- Internal Network Device (Intel® Omni-Path): `ib0`

Compute Node

Hostname representation in this document varies to best fit the situation. When used in hostnames, `NNN` and `XXX` are meant represent a strictly 3-digit number (e.g. `node001`). When used in IP addresses, however, leading zeroes are **not** used (e.g. `192.168.1.1`). As a means of disambigutation, `NNN` is used to specify the last host, and `XXX` used to denote a single host.

The prescribed class C network supports up to 254 hosts including the head node. These settings allow for a convenient pairing of node hostnames to IP addresses – `nodeXXX` to `192.168.1.XXX` – but at the cost of scalability past 253 compute nodes. To build a cluster for which `NNN` is greater than 253, the network conventions set forth must be reconsidered.

- Hostnames: `nodeXXX`
- Internal IP Addresses (Ethernet): `192.168.1.XXX`
- Internal Network Device (Ethernet): `enp3s0f0`
- Internal IP Addresses (Intel® Omni-Path): `192.168.5.XXX`
- Internal Network Device (Intel® Omni-Path): `ib0`

Chapter 2

Preparation

Section 2.1: Assembly

The system is a simple Beowulf style cluster, consisting of a single head node managing all cluster functions and one or more compute nodes for processing.

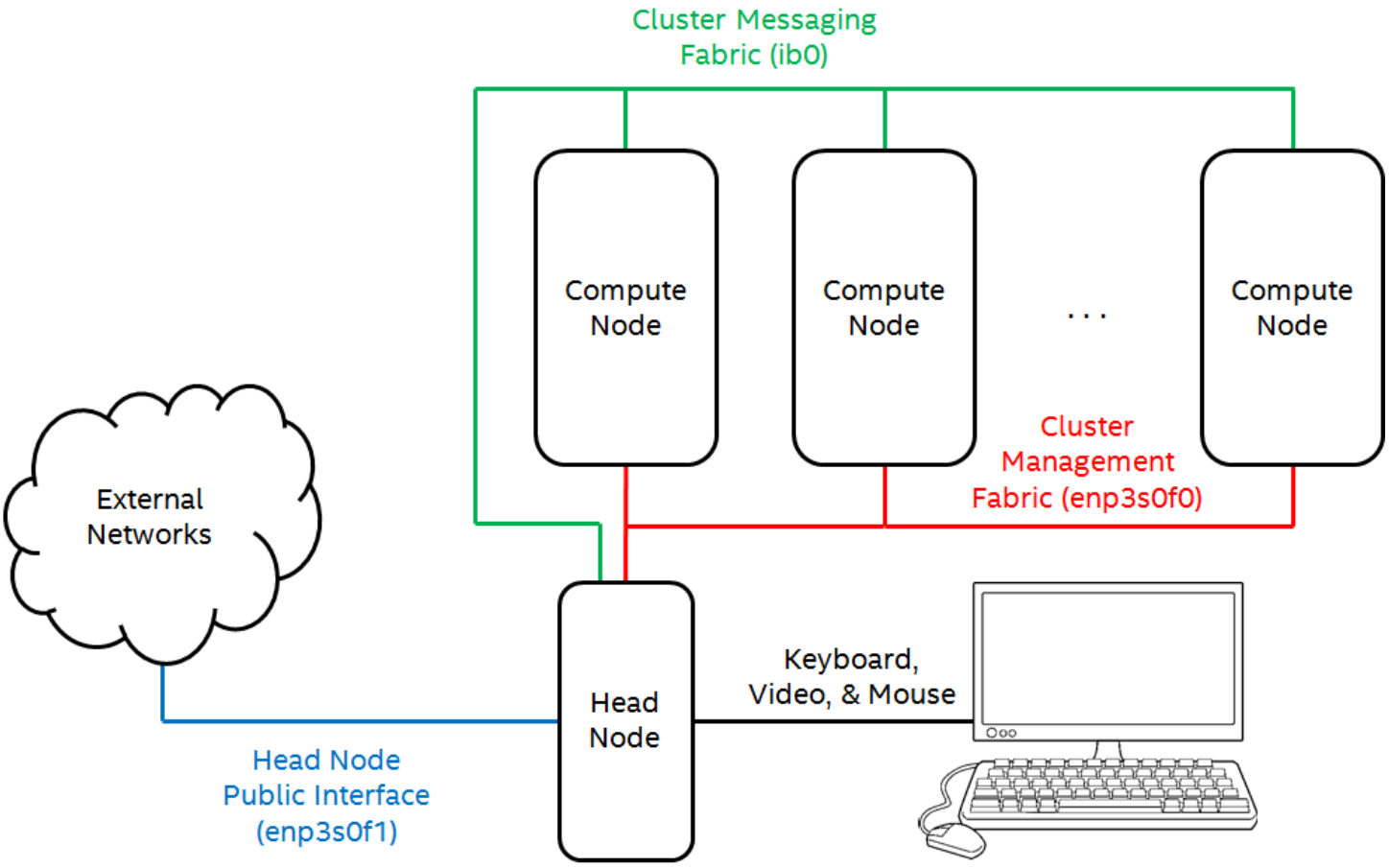

Figure 2.1: A visualization of system networks.

Section 2.2: Collect MAC Addresses

Warewulf* provides a method to gather MAC addresses as nodes are booted. However, this method is inefficient for large clusters. The preferred method to identify nodes is to pre-collect MAC addresses. Before following these instructions, record the MAC address for the first Ethernet port of each node for later use.

Chapter 3

Install Software on the Head Node

The head node is configured as the primary node in the cluster and is set up to manage and install all compute nodes.

Section 3.1: Install the Linux* Operating System from Disc

1. Insert the CentOS* 7.2 install disc. Boot from disc and select "Install CentOS 7".

2. Select "English" as the language and click "Continue".

3. Select "DATE & TIME".

a. Select the timezone using the geographical location dropdown menus.

b. Click "Done" to return home.

4. Select "SOFTWARE SELECTION".

a. In the box labeled "Base Environment" on the left side, select "Infrastructure Server".

b. Click "Done" to return home.

5. If necessary, select "INSTALLATION DESTINATION".

a. Select the automatic partitioning option.

b. Click "Done" to return home. Accept all defaults for the partitioning wizard if prompted.

6. Select "NETWORK & HOST NAME".

a. Enter "`frontend.localdomain`" as the hostname.

b. Select "Ethernet `enp3s0f0`" and click "Configure" to setup the internal cluster interface.

(1) From the "General" section, check "Automatically connect to this network when it is available".

(2) From the "IPv4 Settings" section, select the "Manual" method and add the address `192.168.1.254` with netmask `24`. Save and exit.

c. Select "Ethernet `enp3s0f1`" and click "Configure" to setup the external cluster interface.

(1) From the "General" section , check "Automatically connect to this network when it is available".

(2) Configure the external interface as necessary. Save and exit.

d. Set the toggle to "ON" for both interfaces.

e. Click "Done" to return home.

7. Click “Begin Installation”.

8. While waiting for the installation to finish, set the root password. User creation is optional.

9. Click “Reboot” when the installation is complete.

10. Boot from the primary drive.

11. Login as root.

ADVISORY: Suppress the HFI messages.

On booting the head node, there might be messages in the local terminal saying:

```
LNI failure last states: local 0x00001000 remote 0x00000000
8051 info error: Unknown frame received
```

These messages are displayed because CentOS* does not provide the hfi drivers. The drivers are introduced at a later stage by installing the Intel® Omni-Path Fabric Software. To suppress the messages now, issue the following command:

```
dmesg -n 1
```

Section 3.2: Setup Command Line Tools

The remaining steps in this document are all designed to be completed from a command line interface unless otherwise stated. The command line interface is the default local login method since no GUI is provided in the prescribed operating system installation. Remote SSH login is also available by default. It is suggested to use the method that makes it easiest to copy and paste commands from this document.

TIP: Configure the yum command line tool.

The `yum` command line tool is a primary means of installing software in this reference design. By default, `yum` prompts the user before making software changes. When prompted by `yum`, accept the changes by pressing **Y** followed by **Enter**.

It is possible to bypass interaction on a per-command basis by running `yum` with the long flag `--assumeyes` or the short flag `-y`. A global bypass may be applied by creating an alias for the command. To allow the alias to persist on reboot, it is advised to add it to `/root/.bashrc`.

a. Open the `/root/.bashrc` file for editing.

b. Add the following line to the end of the file:

```
alias yum='yum -y'
```

c. Save the file and exit.

d. Source the updated file to realize the alias in the current session.

```
source /root/.bashrc
```

An alternative method of setting a global bypass is to edit the `yum` configuration file.

a. Open the `/etc/yum.conf` file for editing.

b. Under the `[main]` section, add the following line:

```
assumeyes=1
```

c. Save the file and exit.

TIP: Configure the wwsh command line tool.

The `wwsh` command line tool is a primary means of configuring Warewulf* in this reference design. Just like `yum`, the default behavior of `wwsh` is to prompt the user before making changes. When prompted by `wwsh`, accept the changes by pressing **Y** followed by **Enter**.

It is possible to bypass interaction on a per-command basis by running `wwsh` with the long flag `--yes` or the short flag `-y`. A global bypass may be applied by creating an alias for the command. To allow the alias to persist on reboot, it is advised to add it to `/root/.bashrc`.

a. Open the `/root/.bashrc` file for editing.

b. Add the following line to the end of the file:

```
alias wwsh='wwsh -y'
```

c. Save the file and exit.

d. Source the updated file to realize the alias in the current session.

```
source /root/.bashrc
```

Section 3.3: Configure YUM Proxy

TIP: If necessary, configure an Internet proxy.

If the public network implements a proxy server for Internet access, YUM must be configured to use it.

a. Open the `/etc/yum.conf` file for editing.

b. Under the `[main]` section, append the following line:

```
proxy=http://<address>:<port>
```

Where `<address>` is the address of the proxy server and `<port>` is the HTTP port.

c. Save the file and exit.

Section 3.4: Enable Software Repositories

The OpenHPC* community provides a release package that includes GPG keys for package signing and YUM repository configuration. This package can be installed directly from the OpenHPC* build server. In addition, the head node must have access to the standard CentOS* 7.2 and EPEL* repositories. Mirrors are readily available for both repositories.

The public EPEL* repository is enabled automatically by the ohpc-release package. This requires that the CentOS* Extras repository is enabled, which is default.

12. Enable the OpenHPC* repository.

```
yum install http://build.openhpc.community/\
OpenHPC:/1.1/CentOS_7.2/x86_64/ohpc-release-1.1-1.x86_64.rpm
```

Section 3.5: Lock Kernel Version

Certain procedures in this reference design require packages to be built against the kernel. A future kernel update may break the compatibility of these built packages with the new kernel, so the kernel version is locked to provide further longevity to this reference design.

13. Install the versionlock plugin for YUM.

```
yum install yum-plugin-versionlock
```

14. Lock the kernel version.

```
yum versionlock *-3.10.0-327.36.1.el7.x86_64
```

Section 3.6: Update Packages

Packages provided by the initial infrastructure server installation are updated from the online repositories to ensure they are at the latest version.

15. Update all packages.

```
yum update
```

Section 3.7: Install Additional Packages from CentOS*

The development tools package group provides the tools needed to build certain software packages later in this guide. The file server package group provides the tools necessary to configure an NFS server.

16. Install the development tools package group.

```
yum groupinstall development
```

17. Install the file server package group.

```
yum groupinstall file-server
```

Section 3.8: Install OpenHPC* Base and Warewulf* Provisioning System

To add support for provisioning services, install the OpenHPC* base and Warewulf* provisioning system package groups.

18. Install the OpenHPC* base package group.

```
yum groupinstall ohpc-base
```

19. Install the Warewulf* package group.

```
yum groupinstall ohpc-warewulf
```

Section 3.9: Install SLURM Server

The SLURM Workload Manager provides a means of scheduling jobs and allocating cluster resources to those jobs. The server component is installed on the head node.

20. Install the SLURM server package group.

```
yum groupinstall ohpc-slurm-server
```

Section 3.10: Install Intel® SSF Compliance Packages

ADVISORY: This step requires additional software components not provided via enabled YUM repositories:

- `ssf-meta-el7-2016.0-1.x86_64.rpm`
- `ssf-meta-el7-advisory-2016.0-1.x86_64.rpm`

The Intel® SSF meta-RPMs are installed on the head node to meet Intel® SSF software stack requirements and advisories.

21. Install the Intel® SSF compliance packages.

```
yum install ssf-meta-el7-2016.0-1.x86_64.rpm \
ssf-meta-el7-advisory-2016.0-1.x86_64.rpm
```

Section 3.11: Reboot the Head Node

The head node is rebooted so that any kernel updates are activated.

22. Issue a reboot command.

```
init 6
```

23. Login as root.

Section 3.12: Install Intel® Parallel Studio XE 2017 Cluster Edition

ADVISORY: This step requires additional software components not provided via enabled YUM repositories:

- `parallel_studio_xe_2017_update1.tgz`

Intel® Parallel Studio XE provides compilers and performance libraries to build fast code, analysis tools to aid in debugging and tuning applications, and MPI cluster tools for developing parallelizable software. The included Intel® MPI runtimes enable the Intel® Omni-Path Fabric if present, and the Intel® Cluster Checker will assist in validating compliance to the Intel® SSF Reference Architecture.

24. Install soft prerequisite packages.

```
yum install gtk2 redhat-lsb
```

25. Extract the installer.

```
tar -zxvf parallel_studio_xe_2017_update1.tgz -C /usr/src
```

26. Install Intel® Parallel Studio XE 2017 Cluster Edition.

a. Start the installer.

```
/usr/src/parallel_studio_xe_2017_update1/install.sh
```

b. Press **Enter** to continue.

c. Read the EULA. Press **Space** to scroll through each page and continue to the next prompt.

d. Type the word **accept** and press **Enter**.

e. Wait for the prerequisite check to finish. This check may take several minutes.

f. Follow the prompts to activate the license. Activation may take several minutes. Press **Enter** to continue.

g. Accept or decline involvement in the Intel® Software Improvement Program. Press **Enter** to continue.

h. Press **Enter** to begin configuring the installation.

i. Press **Space** to scroll, type **2**, then press **Enter**.

j. Press **1** to deselect IA-32 architecture, then press **Enter**. It is possible to install with IA-32 selected, though it is outside the scope of this reference design.

k. Press **Enter** once to proceed, then press **Enter** again to begin the installation.

l. If the prompt shown below appears, select **y** and press **Enter**.

```
---------------------------------------------------------------------------
Please type a selection or press "Enter" to accept default choice [1]:
WARNING: Destination directory already exists.
---------------------------------------------------------------------------
Do you want to continue?
---------------------------------------------------------------------------
n. No
y. Yes
---------------------------------------------------------------------------
Please type a selection or press "Enter" to accept default choice [n]:
```

m. Wait for the installation to finish. Installation may take several minutes. When prompted, press **Enter** to complete the installation.

Section 3.13: Install Packages To Enable Environment Modules

The Intel® SSF Reference Architecture demands that a number of software components be controllable via environment modules. OpenHPC* provides versions of these software components with module files pre-configured and also scripts for generating module files for software components from Intel® Parallel Studio XE.

27. Install the GNU C compiler.

```
yum install gnu-compilers-ohpc
```

28. Install the OpenHPC* compatibility package for Intel® Parallel Studio XE.

ADVISORY: The Intel® Parallel Studio XE compatibility package in OpenHPC* 1.1 is compatible with previous versions of Intel® Parallel Studio XE. Using this package with Intel® Parallel Studio XE 2017 Cluster Edition, however, requires circumventing a package dependency and manually generating the environment modules.

```
yumdownloader intel-compilers-devel-ohpc
rpm -i intel-compilers-devel-ohpc*.rpm --nodeps --noscripts
```

Section 3.14: Install Intel® Omni-Path Fabric Software

ADVISORY: This step requires additional software components not provided via enabled YUM repositories:

- `IntelOPA-IFS.RHEL72-x86_64.10.2.0.0.158.tgz`

The IFS package installs the Intel® Omni-Path Fabric Host Software components needed to set up compute, I/O, and service nodes with drivers, stacks and basic tools for local configuration and monitoring. It also installs the Intel® Omni-Path Fabric Suite FastFabric Toolset and the Intel® Omni-Path Fabric Suite Fabric Manager.

29. Install CentOS* distribution supported fabric software.

```
yum groupinstall infiniband
```

30. Install prerequisite packages.

```
yum install \
atlas infinipath-psm qperf perftest sysfsutils opensm-devel \
libibumad-devel libibverbs-devel libibmad-devel librdmacm-devel \
openssl-devel libuuid-devel infinipath-psm-devel valgrind-devel
```

31. Extract the installer.

```
tar -zxvf IntelOPA-IFS.RHEL72-x86_64.10.2.0.0.158.tgz -C /usr/src
```

32. Install the Intel® Omni-Path fabric software.

a. Change directories.

```
cd /usr/src/IntelOPA-IFS.RHEL72-x86_64.10.2.0.0.158
```

b. Run the installer, specifying a sufficient subset of available components to install.

```
./INSTALL -i opa_stack -i ibacm -i intel_hfi -i oftools -i fastfabric \
-i delta_ipoib -i opafm
```

In the final lines of the installer output, a warning is printed:

```
A System Reboot is recommended to activate the software changes
```

Despite this warning, it is **NOT** necessary to reboot the system at this time. The system is rebooted in a later step.

c. Enter into the fabric delta directory.

```
cd /usr/src/IntelOPA-IFS.RHEL72-x86_64.10.2.0.0.158/\
IntelOPA-OFED_DELTA.RHEL72-x86_64.10.2.0.0.169/RPMS/redhat-ES72
```

d. Install the package that provides `ibstat` from the fabric delta directory.

```
yum install infiniband-diags-1.6.7-2.el7.x86_64.rpm
```

e. Change directories back to the root home.

```
cd ~
```

Section 3.15: Install Open Fabrics Interface (OFI) Software for Intel® Omni-Path Fabric

ADVISORY: This step requires additional software components not provided via enabled YUM repositories:

- `OFI-for-IntelOPA-RHEL7_2-x86_64.10.2.0.0.158-93.tar.gz`

The Open Fabrics Interface (OFI) Software for Intel® Omni-Path provides OFI support for Intel® Omni-Path Fabric.

33. Extract the installer.

```
tar -zxvf OFI-for-IntelOPA-RHEL7_2-x86_64.10.2.0.0.158-93.tar.gz -C /usr/src
```

34. Install the Open Fabrics Interface (OFI) Software for Intel® Omni-Path.

a. Enter into the OFI directory.

```
cd /usr/src/OFI-for-IntelOPA-RHEL7_2-x86_64.10.2.0.0.158-93
```

b. Install libfabric packages from the OFI directory.

```
yum install libfabric-1.3.0-93c095d.el7.x86_64.rpm \
libfabric-devel-1.3.0-93c095d.el7.x86_64.rpm
```

c. Change directories back to the root home.

```
cd ~
```

Chapter 4

Configure the Head Node

Section 4.1: Configure Warewulf*

Warewulf* is configured to provision nodes using the conventions of this guide.

1. Set the provisioning interface.

The provisioning interface is the internal device configured for the `192.168.1.254` IP address.

a. Open the `/etc/warewulf/provision.conf` file for editing.

b. Modify the line containing `network device = eth1` so that it reads as follows:

```
network device = enp3s0f0
```

c. Save the file and exit.

2. Update the default settings for provisioned nodes.

a. Open the `/etc/warewulf/defaults/node.conf` file for editing.

b. Add or modify the following lines:

```
cluster = default
groups = compute
netmask = 255.255.255.0
network = 192.168.1.0
netdev = enp3s0f0
```

c. Save the file and exit.

3. Update the default node provisioning settings.

a. Open the `/etc/warewulf/defaults/provision.conf` file for editing.

b. Modify the file so that it contains these lines:

```
bootstrap = 3.10.0-327.36.1.el7.x86_64
vnfs = compute-xeon
files = dynamic_hosts, passwd, group, shadow, gshadow
```

c. Save the file and exit.

4. Update the VNFS settings.

a. Open the `/etc/warewulf/vnfs.conf` file for editing.

b. In the `EXCLUDE` section, modify the file so that the following lines, and only the following lines, are included and uncommented.

```
exclude += /tmp/*
exclude += /var/log/*
exclude += /var/tmp/*
exclude += /var/cache/*
exclude += /home/*
exclude += /opt/*
```

c. In the `HYBRIDPATH` section, modify the file so that the following line, and only the following line, is included and uncommented.

```
hybridpath = /opt/ohpc/admin/images/%{name}
```

d. In the `HYBRIDIZE` section, modify the file so that the following lines, and only the following lines, are included and uncommented.

```
hybridize += /usr/src
hybridize += /usr/lib/locale
hybridize += /usr/lib64/locale
hybridize += /usr/include
hybridize += /usr/share
```

e. Save the file and exit.

5. Update the bootstrap settings to include kernel updates.

a. Open the `/etc/warewulf/bootstrap.conf` file for editing.

b. Add the following line:

```
drivers += updates, updates/kernel
```

c. Save the file and exit.

6. Enable the tftp service for compute node image distribution.

a. Open the `/etc/xinetd.d/tftp` file for editing.

b. Modify the line containing `disable = yes` so that it reads `disable = no`.

c. Save the file and exit.

7. Restart the xinetd service to allow tftp changes to take effect.

```
systemctl restart xinetd
```

8. Enable HTTP access using the new Apache web server syntax.

a. Open the `/etc/httpd/conf.d/warewulf-httpd.conf` file for editing.

b. Locate the section beginning with `<Directory /usr/libexec/warewulf/cgi-bin>`. At the beginning of this section, insert the following line:

```
Require all granted
```

c. Locate the section beginning with `<Directory /usr/share/warewulf/www>`. In this section replace

```
Allow from all
```

with

```
Require all granted
```

d. In this same section, delete the following line:

```
Order allow,deny
```

e. When complete, the updated section reads as follows:

```
<Directory /usr/libexec/warewulf/cgi-bin>
    Require all granted
    SetHandler perl-script
    PerlResponseHandler ModPerl::Registry
    PerlOptions +ParseHeaders
    Options +ExecCGI
</Directory>

<Directory /usr/share/warewulf/www>
    Options Indexes MultiViews
    AllowOverride None
    Require all granted
</Directory>
```

f. Save the file and exit.

9. Enable the database and web services to start automatically and restart the services.

```
systemctl enable mariadb.service
systemctl enable httpd.service
systemctl restart mariadb
systemctl restart httpd
```

10. Import files containing user and group information.

Warewulf* includes functionality to import file from the head node and distribute these to other nodes. This is used to synchronize user and group information with compute nodes.

```
wwsh file import /etc/hosts --name=dynamic_hosts
wwsh file import /etc/passwd
wwsh file import /etc/group
wwsh file import /etc/shadow
wwsh file import /etc/gshadow
```

Section 4.2: Configure SLURM Server

11. Create a user group that is granted unrestricted SSH access.

Normal user SSH access to compute nodes is restricted by PAM (pluggable authentication module) via SLURM. Since this may not be ideal in certain instances, a user group is created for which these restrictions do not apply.

```
groupadd sshwhitelist
```

12. Create the SLURM user account.

The default SLURM configuration requires a SLURM user for the resource management daemons. The user is added as a system user with no home directory and no shell.

```
useradd --system --shell /sbin/nologin slurm
```

13. Update the SLURM server configuration.

a. Open the `/etc/slurm/slurm.conf` file for editing.

b. Modify the line containing `ControlMachine` so that it reads:

```
ControlMachine=frontend
```

c. Update the node resource information. Locate the line beginning with `NodeName=` and modify it so that it reads:

```
NodeName=node[001-NNN] Sockets=2 CoresPerSocket=22 ThreadsPerCore=2 State=UNKNOWN
```

d. Update the partition setup. Locate the line beginning with `PartitionName=` and modify it so that it reads:

```
PartitionName=cluster Nodes=node[001-NNN] Default=YES MaxTime=24:00:00 State=UP
```

e. Save the file and exit.

14. Import files for SLURM configuration.

Warewulf* is used to synchronize the SLURM configuration file and cryptographic key that is required by the MUNGE authentication library to be available on every host in the resource management pool.

```
wwsh file import /etc/slurm/slurm.conf
wwsh file import /etc/munge/munge.key
```

15. Update the list of provisioned files.

a. Open the `/etc/warewulf/defaults/provision.conf` file for editing.

b. Modify the line beginning with `files` so that it reads:

```
files = dynamic_hosts, passwd, group, shadow, gshadow, slurm.conf, munge.key
```

c. Save the file and exit.

16. Enable and restart the MUNGE and SLURM controller services.

```
systemctl enable munge.service
systemctl enable slurmctld.service
systemctl restart munge
systemctl restart slurmctld
```

Section 4.3: Configure Environment Modules

Setting up modules for software components, many of which supplied by Intel® Parallel Studio XE, provides both a simple means of loading environments and complying with Intel® SSF Reference Architecture.

17. Create module files for Intel® Parallel Studio XE components.

a. Build the module for the Intel® C++ Compiler.

```
mkdir -p /opt/ohpc/pub/modulefiles/intel/icc
/opt/ohpc/admin/compat/modulegen/mod_generator.sh \
/opt/intel/bin/iccvars.sh -arch intel64 -platform linux \
>/opt/ohpc/pub/modulefiles/intel/icc/2017.1
```

b. Build the module for the Intel® Fortran Compiler.

```
mkdir -p /opt/ohpc/pub/modulefiles/intel/ifort
/opt/ohpc/admin/compat/modulegen/mod_generator.sh \
/opt/intel/bin/ifortvars.sh -arch intel64 -platform linux \
>/opt/ohpc/pub/modulefiles/intel/ifort/2017.1
```

c. Build the module for the Intel® Math Kernel Library.

```
mkdir -p /opt/ohpc/pub/modulefiles/intel/mkl
/opt/ohpc/admin/compat/modulegen/mod_generator.sh \
/opt/intel/mkl/bin/mklvars.sh intel64 \
>/opt/ohpc/pub/modulefiles/intel/mkl/2017.1
```

d. Build the module for Intel® Threading Building Blocks.

```
mkdir -p /opt/ohpc/pub/modulefiles/intel/tbb
/opt/ohpc/admin/compat/modulegen/mod_generator.sh \
/opt/intel/tbb/bin/tbbvars.sh intel64 \
>/opt/ohpc/pub/modulefiles/intel/tbb/2017.1
```

e. Build the module for Intel® Cluster Checker.

```
mkdir -p /opt/ohpc/pub/modulefiles/intel/clck
/opt/ohpc/admin/compat/modulegen/mod_generator.sh \
/opt/intel/clck/2017.1.016/bin/clckvars.sh \
>/opt/ohpc/pub/modulefiles/intel/clck/2017.1
```

f. Build the module for the Intel® MPI Library.

```
mkdir -p /opt/ohpc/pub/modulefiles/intel/mpi
/opt/ohpc/admin/compat/modulegen/mod_generator.sh \
/opt/intel/impi/2017.1.132/intel64/bin/mpivars.sh \
>/opt/ohpc/pub/modulefiles/intel/mpi/2017.1
```

18. Alter the skeleton bashrc so modules are loaded by default.

a. Open the `/etc/skel/.bashrc` file for editing.

b. Append the following lines to the end of the file. Some lines are commented so that each user may conveniently select which modules to load for his or her account.

```
#module load gnu/5.3.0
#module load intel/icc/2017.1
#module load intel/ifort/2017.1
#module load intel/mkl/2017.1
```

```
#module load intel/tbb/2017.1
module load intel/clck/2017.1
module load intel/mpi/2017.1
```

Save the file and exit.

Section 4.4: Add Head Node to Hosts File

The head node is not added automatically to the hosts file, so it is done manually.

19. Update the hosts file.

a. Open the `/etc/hosts` file for editing.

b. Add the following line to the end of the file:

```
192.168.1.254 frontend.localdomain frontend
```

c. Save the file and exit.

Section 4.5: Configure Firewall

Provisioning services use DHCP, TFTP, and HTTP network protocols. Default firewall rules may block these services. However, disabling the firewall on an Internet-accessible device is insecure. The following changes allow all connections within the cluster while maintaining the firewall on the external interface.

20. Secure the external connection.

The zone can be set to `internal` or `public` if more security is required.

```
nmcli connection modify enp3s0f1 connection.zone work
```

21. Disable the firewall on the internal network.

```
nmcli connection modify enp3s0f0 connection.zone trusted
```

22. Commit the changes to the current configuration.

```
nmcli connection reload
```

23. Restart the firewall to realize changes.

```
systemctl restart firewalld
```

Section 4.6: Configure Chrony (NTP)

HPC systems typically rely on synchronized clocks throughout the system and the chrony system service can be used to facilitate this synchronization.

24. Configure the chrony server.

a. Open the `/etc/chrony.conf` file for editing.

b. Delete or comment any lines to servers that are not used.

c. For each external time server to be used, add the line:

```
server <hostname>
```

Where <hostname> is the name or IP address of an external time server.

d. Allow unrestricted NTP access to compute nodes on the cluster subnet by adding the following line:

```
allow 192.168.1.0/24
```

e. Save the file and exit.

25. Enable and restart the chrony service.

```
systemctl enable chronyd.service
systemctl restart chronyd
```

Section 4.7: Configure NFS

The /home directory is shared as read/write, and the /opt directory is shared read-only across the cluster.

26. Update exports.

a. Open the /etc/exports file for editing.

b. Add the following lines:

```
/home 192.168.1.0/24(rw,no_subtree_check,fsid=10,no_root_squash)
/opt 192.168.1.0/24(ro,no_subtree_check,fsid=11)
```

c. Save the file and exit.

27. Enable and restart the NFS server.

```
systemctl enable nfs-server.service
systemctl restart nfs-server
```

Section 4.8: Configure System Log Forwarding

System logging for the cluster can be consolidated to the head node to provide easy access and reduce the memory requirements on the diskless compute node.

28. Configure the head node to receive messages.

a. Open the /etc/rsyslog.conf file for editing.

b. Uncomment these lines by removing the #:

```
# $ModLoad imudp
# $UDPServerRun 514
```

c. Save the file and exit.

29. Enable and restart the rsyslog service.

```
systemctl enable rsyslog.service
systemctl restart rsyslog
```

Section 4.9: Configure Intel® Cluster Checker

Intel® Cluster Checker (Intel® CLCK) is one of many tools installed as part of Intel® Parallel Studio XE. It is a powerful tool designed to assess cluster health.

30. Add the Intel® CLCK user.

```
useradd --groups sshwhitelist clck
```

31. Set the Intel® CLCK user password.

Execute the following command. You must enter the desired password once to set and then again to confirm.

```
passwd clck
```

Section 4.10: Configure Intel® Omni-Path Fabric Software

32. Configure IP over IB for Intel® Omni-Path.

a. Change directory into the unpackaged Intel® Omni-Path source.

```
cd /usr/src/IntelOPA-IFS.RHEL72-x86_64.10.2.0.0.158
```

b. Run the installer.

```
./INSTALL
```

c. Press **2** at the main menu to configure IP over IB.

d. Press **Y** followed by **Enter** to continue.

e. Press **Enter** to configure, by default, a single interface.

f. Press **Enter** to use the default interface name.

g. Enter `192.168.5.254` as the IPv4 address for `ib0` then press **Enter**. Press **Y** followed by **Enter** to confirm.

h. Enter `255.255.255.0` as the IPv4 netmask for `ib0` then press **Enter**. Press **Y** followed by **Enter** to confirm.

i. Press **X** to exit the installer.

j. Enable autostart for the drivers.

```
./INSTALL -s
```

k. Change directories back to the root home.

```
cd ~
```

33. Disable the firewall on the IP over IB interface.

a. Open the `/etc/sysconfig/network-scripts/ifcfg-ib0` file for editing.

b. Add or modify the following lines:

```
NAME=ib0
TYPE=Infiniband
```

```
NM_CONTROLLED=yes
```

c. Save the file and exit.

d. Restart the Network Manager to have `ib0` as a Network Manager Controlled Interface.

```
systemctl restart NetworkManager
```

e. Disable the firewall on the `ib0` network.

```
nmcli connection modify ib0 connection.zone trusted
```

34. Enable and restart the fabric manager service.

```
systemctl enable opafm.service
systemctl restart opafm
```

35. Add IP over IB names to the hosts file.

a. Open the `/etc/hosts` file for editing.

b. Add the following line to the end of the file:

```
192.168.5.254 frontend-ib0.localdomain frontend-ib0
```

c. Save the file and exit.

36. Import files for Intel® Omni-Path Fabric Software configuration.

A template network script for Intel® Omni-Path interface that is available that is specifically designed to be deployed via Warewulf* for compute nodes to use.

```
wwsh file import /opt/ohpc/pub/examples/network/centos/ifcfg-ib0.ww
wwsh file set ifcfg-ib0.ww --path=/etc/sysconfig/network-scripts/ifcfg-ib0
```

37. Update the list of provisioned files to include the IP over IB network script.

a. Open the `/etc/warewulf/defaults/provision.conf` file for editing.

b. Modify the line beginning with `files` so that it reads:

```
files = dynamic_hosts, passwd, group, shadow, gshadow, slurm.conf, munge.key, ifcfg-ib0.ww
```

c. Save the file and exit.

38. Update the bootstrap settings to include hfi1 drivers.

a. Open the `/etc/warewulf/bootstrap.conf` file for editing.

b. Add the following lines:

```
drivers += hfi1
firmware += updates/hfi1*
```

c. Save the file and exit.

Section 4.11: Reboot the Head Node

As per the directive issued by the Intel® Omni-Path Fabric Software installer, the head node is now rebooted.

39. Issue a reboot command.

```
init 6
```

40. Login as root.

Chapter 5

Build the Compute Node Image

Section 5.1: Initialize Node Image

The build of Warewulf* includes enhancements and enabling for CentOS* 7.2. The `wwmkchroot` command creates a minimal chroot image for use with Warewulf*. This image is created in `/opt/ohpc/admin/images/compute-xeon`.

1. Define chroot location.

For simplicity, the **chroot** location is made permanent by adding it to the `/root/.bashrc` file.

a. Open the `/root/.bashrc` file for editing.

b. Add the following line to the end of the file:

```
export CHROOT=/opt/ohpc/admin/images/compute-xeon
```

c. Save the file and exit.

2. Source the .bashrc file.

```
source /root/.bashrc
```

3. Build the base chroot image.

```
wwmkchroot centos-7 $CHROOT
```

Section 5.2: Update Packages

Update the intial packages from the online repositories to ensure they are at the latest version.

4. Update all packages.

```
yum --installroot=$CHROOT update
```

Section 5.3: Install Additional Packages from CentOS*

5. Install a kernel in the chroot image.

The kernel package is not installed to the **chroot** image automatically, so it is done manually. Note the use of the installroot flag in this command. This instructs YUM to install to the specified **chroot** as opposed to the functional filesystem on the head node.

```
yum --installroot=$CHROOT install kernel
```

6. Install the base package group.

The base package group provides basic tools from CentOS*.

```
yum --installroot=$CHROOT groupinstall base
```

Section 5.4: Install OpenHPC* Base

7. Install the OpenHPC* base package group.

```
yum --installroot=$CHROOT groupinstall ohpc-base
```

Section 5.5: Install SLURM Client

8. Install the SLURM client package group.

```
yum --installroot=$CHROOT groupinstall ohpc-slurm-client
```

Section 5.6: Install Intel® SSF Compliance Packages

ADVISORY: This step requires additional software components not provided via enabled YUM repositories:

- `ssf-meta-el7-2016.0-1.x86_64.rpm`
- `ssf-meta-el7-advisory-2016.0-1.x86_64.rpm`

The Intel® SSF meta-RPMs are installed on the **chroot** to meet Intel® SSF software stack requirements and advisories.

9. Install the Intel® SSF compliance packages.

```
yum --installroot=$CHROOT install ssf-meta-el7-2016.0-1.x86_64.rpm \
ssf-meta-el7-advisory-2016.0-1.x86_64.rpm
```

Section 5.7: Install Intel® Omni-Path Fabric Software

Only the main kernel module RPMs are required. The `-devel` and `-debug` RPMs are not required.

10. Install the CentOS* distribution fabric software.

```
yum --installroot=$CHROOT groupinstall infiniband
```

11. Install additional prerequisite packages.

```
yum --installroot=$CHROOT install infinipath-psm
```

12. Remove conflicting software packages.

```
yum --installroot=$CHROOT remove ibacm
```

13. Install the Intel® Omni-Path fabric software.

a. Enter into the fabric delta directory.

```
cd /usr/src/IntelOPA-IFS.RHEL72-x86_64.10.2.0.0.158/\
IntelOPA-OFED_DELTA.RHEL72-x86_64.10.2.0.0.169/RPMS/redhat-ES72
```

b. Install packages from the fabric delta directory.

```
yum --installroot=$CHROOT install \
libpsm2-10.2.42-1.x86_64.rpm libpsm2-compat-10.2.42-1.x86_64.rpm \
ifs-kernel-updates-3.10.0_327.36.1.el7.x86_64-153.x86_64.rpm \
libhfi1-0.5-24.el7.x86_64.rpm \
hfi1-firmware-0.9-46.noarch.rpm \
hfi1-firmware_debug-0.9-46.noarch.rpm \
hfi1-uefi-0.2-28.x86_64.rpm \
hfidiags-0.8-27.x86_64.rpm opa-scripts-1.0-1.0.noarch.rpm \
infiniband-diags-1.6.7-2.el7.x86_64.rpm \
srptools-1.0.3-1.el7.x86_64.rpm
```

c. Enter into the Intel® Omni-Path tools directory.

```
cd /usr/src/IntelOPA-IFS.RHEL72-x86_64.10.2.0.0.158/\
IntelOPA-Tools-FF.RHEL72-x86_64.10.2.0.0.154/RPMS/x86_64
```

d. Install packages from the Intel® Omni-Path tools directory.

```
yum --installroot=$CHROOT install \
opa-address-resolution-10.2.0.0-154.x86_64.rpm \
opa-basic-tools-10.2.0.0-154.x86_64.rpm \
opa-mpi-apps-10.2.0.0-154.x86_64.rpm
```

e. Change directories back to the root home.

```
cd ~
```

Section 5.8: Install the Open Fabrics Interface (OFI) Software for Intel® Omni-Path Fabric

14. Install the Open Fabrics Interface (OFI) Software for Intel® Omni-Path.

a. Enter into the OFI directory.

```
cd /usr/src/OFI-for-IntelOPA-RHEL7_2-x86_64.10.2.0.0.158-93
```

b. Install libfabric packages from the OFI directory.

```
yum --installroot=$CHROOT install \
libfabric-1.3.0-93c095d.el7.x86_64.rpm \
libfabric-devel-1.3.0-93c095d.el7.x86_64.rpm
```

c. Change directories back to the root home.

```
cd ~
```

Chapter 6

Configure the Compute Node Image

Section 6.1: Configure SLURM Client

1. Enable the SLURM client service.

```
systemctl --root=$CHROOT enable slurmd.service
```

2. Allow unrestricted SSH access for the sshwhitelist group.

NOTE: Users not belonging to `sshwhitelist` will be subject to the SSH restrictions.

a. Open the `$CHROOT/etc/security/access.conf` file for editing.

b. Append the following lines, in order, to the end of the file.

```
+ : sshwhitelist : ALL
- : ALL : ALL
```

c. Save the file and exit.

3. Enable SSH control via the SLURM resouce manager.

Enabling PAM (pluggable authentication module) in the **chroot** environment limits SSH access to only those nodes on which the user has active jobs.

a. Open the `$CHROOT/etc/pam.d/sshd` file for editing.

b. Add the following lines, in order, to the end of the file:

```
account sufficient pam_access.so
account required pam_slurm.so
```

c. Save the file and exit.

Section 6.2: Configure Domain Name Resolution

To access the remote repositories by hostname (and not IP address), the **chroot** environment also needs to be updated to enable DNS resolution. The head node has a working configuration in place which is copied to the **chroot** image.

4. Update chroot name resolution.

Note that the copy command is preceded by a backslash so as to override the alias in `/root/.bashrc` that defaults to interactive mode.

```
\cp -p /etc/resolv.conf $CHROOT/etc/resolv.conf
```

Section 6.3: Configure SSH Keys

Add the cluster key to the node **chroot** image to allow passwordless root access to all nodes.

5. Create SSH keys.

```
wwinit ssh_keys
```

6. Authorize the new root key.

```
cat /root/.ssh/cluster.pub >> $CHROOT/root/.ssh/authorized_keys
```

Section 6.4: Configure Chrony (NTP)

On the **chroot** image, add Network Time Protocol (NTP) support and identify the head node as the NTP server.

7. Enable the chrony service.

```
systemctl --root=$CHROOT enable chronyd.service
```

8. Configure the chrony client.

a. Open the `$CHROOT/etc/chrony.conf` file for editing.

b. Delete or comment any lines to servers.

c. Add the following line:

```
server 192.168.1.254
```

d. Save the file and exit.

Section 6.5: Configure NFS

9. Add the NFS client mounts.

a. Open the `$CHROOT/etc/fstab` file for editing.

b. Add the following lines to the end of the file:

```
192.168.1.254:/home /home nfs nfsvers=4,rsize=1024,wsize=1024,cto 0 0
192.168.1.254:/opt /opt nfs nfsvers=4 0 0
```

c. Save the file and exit.

Section 6.6: Configure System Log Forwarding

Disable logging on compute nodes except for emergency and boot logs.

10. Update the system log forwarding configuration.

a. Open the `$CHROOT/etc/rsyslog.conf` file for editing.

b. Add the following line:

```
*.* @192.168.1.254:514
```

c. Comment the following lines by adding a # at the beginning of each line:

```
*.info;mail.none;authpriv.none;cron.none
authpriv.*
mail.*
cron.*
uucp,news.crit
```

d. Save the file and exit.

Section 6.7: Configure Intel® Omni-Path Fabric Software

The Intel® Omni-Path Fabric Software installer script configures many settings on the head node. These settings need to be propagated to the **chroot** image.

11. Copy configuration files from the head node to the chroot image.

Note that the copy command is preceded by a backslash so as to override the default alias in `/root/.bashrc` that forces interactive mode.

```
\cp -p /etc/security/limits.conf $CHROOT/etc/security/
\cp -p /etc/rdma/rdma.conf $CHROOT/etc/rdma/
\cp -p /etc/sysconfig/irqbalance $CHROOT/etc/sysconfig/
\cp -p /etc/udev/rules.d/05-opa.rules $CHROOT/etc/udev/rules.d/
\cp -p /etc/udev/rules.d/60-ipath.rules $CHROOT/etc/udev/rules.d/
\cp -p /etc/udev/rules.d/70-persistent-ipoib.rules $CHROOT/etc/udev/rules.d/
```

Chapter 7

Provision the Compute Nodes

Warewulf* employs a two-stage process for provisioning nodes. First, a bootstrap is used to initialize the kernel and install process, then an encapsulated image containing the full system is loaded on the node.

Section 7.1: Assemble Bootstrap Image

The bootstrap image includes the runtime kernel and associated modules, as well as simple scripts for provisioning.

1. Clear BASH_ENV.

If the environment variable `BASH_ENV` is set, it conflicts with Warewulf* scripts that use Perl with the `-T` switch.

```
unset BASH_ENV
```

2. Create the bootstrap image.

```
wwbootstrap --chroot=$CHROOT 3.10.0-327.36.1.el7.x86_64
```

Section 7.2: Assemble the Virtual Node File System (VNFS) Image

The `wwvnfs` command creates a VNFS capsule from the **chroot** image defined for the compute node instance.

3. Create the VNFS image.

```
wwvnfs --chroot=$CHROOT
```

Section 7.3: Register Nodes for Provisioning

Default associations for nodes, bootstrap images, and VNFS images are already configured in the Warewulf* config files.

4. Add compute nodes to the Warewulf* database.

ADVISORY: In the step below, XXX is used in both IP addresses and hostnames. Omit leading zeroes for IP addresses otherwise Warewulf* will interpret the first zero to mean the number is in octal base. It is safe to include leading zeroes in hostnames.

Repeat the following command for each compute node in the cluster:

```
wwsh node new nodeXXX --ipaddr=192.168.1.XXX --hwaddr=<mac_address>
```

Section 7.4: Configure Fabric IP Addresses

5. Add IP over IB addresses to the Warewulf* database.

ADVISORY: In the step below, XXX is used in both IP addresses and hostnames. Omit leading zeroes for IP addresses otherwise Warewulf* will interpret the first zero to mean the number is in octal base. It is safe to include leading zeroes in hostnames.

Repeat the following command for each compute node in the cluster:

```
wwsh node set nodeXXX -D ib0 --ipaddr=192.168.5.XXX
```

Section 7.5: Prepare Warewulf*

6. Re-synchronize provisioned files on the head node.

```
wwsh file resync
```

7. Restart the dhcpd service.

```
systemctl restart dhcpd
```

8. Update the PXE files.

```
wwsh pxe update
```

Section 7.6: Deploy Compute Nodes

The head node is ready to boot compute nodes remotely.

9. Boot or reboot each compute node.

Power cycle each compute node. Methods may include manually resetting the power button or issuing IPMI commands. Boot each node over the network from the enp3s0f0 interface.

TIP: Monitor the status of compute nodes.

Forward a command to all compute nodes to assess how many have booted.

```
pdsh -w node[001-NNN] uptime
```

Chapter 8

Post Installation

Section 8.1: Add Users and Synchronize Files

Warewulf* automatically synchronizes imported files from the head node at five minute intervals. If a new user is created, several files are immediately outdated. To expedite file synchronization, manually update the database and have the compute nodes pull the updated files.

1. Re-synchronize provisioned files on the head node.

```
wwsh file resync
```

2. Send a directive to expedite file propagation.

Note that this command requires each compute node to be running.

```
pdsh -w node[001-NNN] /warewulf/bin/wwgetfiles
```

Section 8.2: Make Nodes Available in SLURM

3. Allow compute nodes to be available for SLURM jobs.

To put every node into an idle state, thus making it available for resource allocation, issue the following commands:

```
scontrol reconfig
scontrol update NodeName=node[001-NNN] State=idle
```

Section 8.3: Configure Intel® Cluster Checker

A few changes to the Intel® CLCK config file are necessary.

Since this reference design installs diskless compute nodes, the sign `iozone_ran_no_bandwidth` should be suppressed.

We must change the `network_interface` option to the head node private interface.

It is necessary to specify the full path to the `ibstat` binary and to the `tmi.conf` file so that Intel® CLCK may collect the proper data for Intel® Omni-Path Fabric.

a. Open the `/opt/intel/clck/2017.1.016/etc/clck.xml` file for editing.

b. Modify the file so that it looks like this (without comments):

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
<analyzer>
  <config>
  </config>
  <suppressions>
    <suppress>
      <id>iozone-ran-no-bandwidth</id>
    </suppress>
  </suppressions>
</analyzer>
<collector>
  <network_interface>enp3s0f0</network_interface>
  <provider>
    <ibstat>
      <binary>/usr/sbin/ibstat</binary>
    </ibstat>
    <tmiconf>
      <config_file>
        /opt/intel/compilers_and_libraries_2017/linux/mpi/etc64/tmi.conf
      </config_file>
    </tmiconf>
  </provider>
</collector>
<database>
</database>
</configuration>
```

c. Save the file and exit.

Section 8.4: Run Intel® Cluster Checker

4. Switch to the clck user.

To run Intel® Cluster Checker, switch to the `clck` user.

```
su - clck
```

5. Create the nodelist.

The application makes use of the shared home directory for storing temporary files. The application requires a nodefile – a file containing node and node group information.

a. Open the `nodelist` file for editing. This is a new file.

b. Add the head node by appending the following line:

```
frontend # role:head
```

c. For each compute node, add the following line, incrementing `XXX` each time.

```
nodeXXX # role:compute
```

d. The file should look similar to this when completed.

```
frontend # role:head
node001 # role:compute
node002 # role:compute
node003 # role:compute
node004 # role:compute
```

e. Save the file and exit.

6. Run the collector.

Intel® Cluster Checker separates the tasks of data collection and data analysis into two separate tools. Run this command to collect the data:

```
clck-collect -f nodelist -a
```

This command instructs Intel® Cluster Checker to collect information for all checks for every node in the nodelist. To see a more comprehensive list of collector options, run the command:

```
clck-collect --help
```

ADVISORY: Define the environment variable that identifies the license file.

Depending on the method of installing Intel® Parallel Studio XE, it may be necessary to define the `INTEL_LICENSE_FILE` environment variable prior to running the analyzer. The variable should be set to the absolute path of the license, usually located in `/opt/intel/licenses`.

```
export INTEL_LICENSE_FILE=/opt/intel/licenses/l_1234ABCD.lic
```

ADVISORY: Run the analyzer in health mode.

Prior to running the analyzer in validation mode, it is recommended to run in health mode to identify general health issues with the cluster. This may be done by running this command:

```
clck-analyze -f nodelist -m health -p diagnosed_signs
```

The analyzer returns the list of checks performed, the list of nodes checked, and also the results of the analysis. The results are essentially a synopsis of the cluster health.

7. Run the analyzer in validation mode.

Run the analyzer to validate against the Intel® Scalable System Framework Reference Architecture Specification. This may be done by running the following command:

```
clck-analyze -f nodelist -m validation -S compat-hpc -C 2016.0 \
-p diagnosed_signs
```

In validation mode, the analyzer results highlight all instances of non-compliance against Intel® Scalable System Framework Reference Architecture Specification. Analyis of this reference design ends in a “PASS” result indicative of validation to that specification.

Similar to the collector, to see a more comprehensive list of analyzer options, run the command:

```
clck-analyze --help
```