# OpenHPC (v2.0)
# Cluster Building Recipes

## CentOS8.2 Base OS

## bright/SLURM Edition for Linux* (x86_64)


Document Last Update: 2020-11-18

Document Revision: 718c10e9e

## Legal Notice

# Contents

# 1 Introduction

This guide presents a simple cluster installation procedure using components from the OpenHPC software stack. OpenHPC represents an aggregation of a number of common ingredients required to deploy and manage an HPC Linux* cluster including provisioning tools, resource management, I/O clients, development tools, and a variety of scientific libraries. These packages have been pre-built with HPC integration in mind while conforming to common Linux distribution standards. The documentation herein is intended to be reasonably generic, but uses the underlying motivation of a small, 4-node stateless cluster installation to define a step-by-step process. Several optional customizations are included and the intent is that these collective instructions can be modified as needed for local site customizations.

**Base Linux Edition**: this edition of the guide highlights installation without the use of a companion configuration management system and directly uses distro-provided package management tools for component selection. The steps that follow also highlight specific changes to system configuration files that are required as part of the cluster install process.

## 1.1 Target Audience

This guide is targeted at experienced Linux system administrators for HPC environments. Knowledge of software package management, system networking, and PXE booting is assumed. Command-line input examples are highlighted throughout this guide via the following syntax:

```
[sms]# echo "OpenHPC hello world"
```

Unless specified otherwise, the examples presented are executed with elevated (root) privileges. The examples also presume use of the BASH login shell, though the equivalent commands in other shells can be substituted. In addition to specific command-line instructions called out in this guide, an alternate convention is used to highlight potentially useful tips or optional configuration options. These tips are highlighted via the following format:

**Tip**

Life is a tale told by an idiot, full of sound and fury signifying nothing. –Willy Shakes

## 1.2 Requirements/Assumptions

This installation recipe assumes the availability of a single head node *master*, and four *compute* nodes. The *master* node serves as the overall system management server (SMS) and is provisioned with CentOS8.2 and is subsequently configured to provision the remaining *compute* nodes with bright in a stateless configuration. The terms *master* and SMS are used interchangeably in this guide. For power management, we assume that the compute node baseboard management controllers (BMCs) are available via IPMI from the chosen master host. For file systems, we assume that the chosen master server will host an NFS file system that is made available to the compute nodes. Installation information is also discussed to optionally mount a parallel file system and in this case, the parallel file system is assumed to exist previously.

An outline of the physical architecture discussed is shown in Figure 1 and highlights the high-level networking configuration. The *master* host requires at least two Ethernet interfaces with *eth0* connected to the local data center network and *eth1* used to provision and manage the cluster backend (note that these

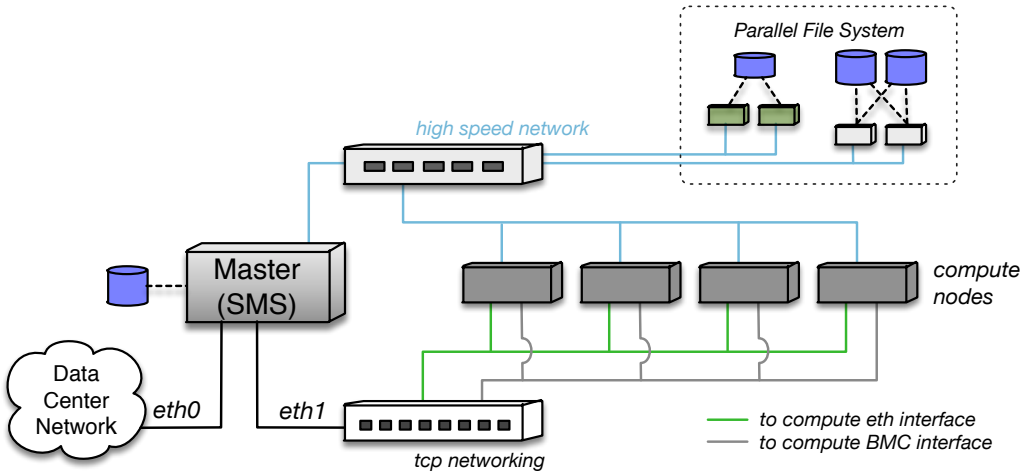


Figure 1: Overview of physical cluster architecture.

interface names are examples and may be different depending on local settings and OS conventions). Two logical IP interfaces are expected to each compute node: the first is the standard Ethernet interface that will be used for provisioning and resource management. The second is used to connect to each host's BMC and is used for power management and remote console access. Physical connectivity for these two logical IP networks is often accommodated via separate cabling and switching infrastructure; however, an alternate configuration can also be accommodated via the use of a shared NIC, which runs a packet filter to divert management packets between the host and BMC.

In addition to the IP networking, there is an optional high-speed network (InfiniBand or Omni-Path in this recipe) that is also connected to each of the hosts. This high speed network is used for application message passing and optionally for parallel file system connectivity as well (e.g. to existing Lustre or BeeGFS storage targets).

## 1.3 Inputs

As this recipe details installing a cluster starting from bare-metal, there is a requirement to define IP addresses and gather hardware MAC addresses in order to support a controlled provisioning process. These values are necessarily unique to the hardware being used, and this document uses variable substitution (`${variable}`) in the command-line examples that follow to highlight where local site inputs are required. A summary of the required and optional variables used throughout this recipe are presented below. Note that while the example definitions above correspond to a small 4-node compute subsystem, the compute parameters are defined in array format to accommodate logical extension to larger node counts.

- ${sms_name} # Hostname for SMS server
- ${sms_ip} # Internal IP address on SMS server
- ${sms_eth_internal} # Internal Ethernet interface on SMS
- ${internal_netmask} # Subnet netmask for internal network
- ${ntp_server} # Local ntp server for time synchronization
- ${bmc_username} # BMC username for use by IPMI
- ${bmc_password} # BMC password for use by IPMI
- ${num_computes} # Total # of desired compute nodes
- ${c_ip[0]}, ${c_ip[1]}, ... # Desired compute node addresses
- ${c_bmc[0]}, ${c_bmc[1]}, ... # BMC addresses for computes
- ${c_mac[0]}, ${c_mac[1]}, ... # MAC addresses for computes
- ${c_name[0]}, ${c_name[1]}, ... # Host names for computes
- ${compute_regex} # Regex matching all compute node names (e.g. "c*")
- ${compute_prefix} # Prefix for compute node names (e.g. "c")

Optional:

- ${sysmgmtd_host} # BeeGFS System Management host name
- ${mgs_fs_name} # Lustre MGS mount name
- ${sms_ipoib} # IPoIB address for SMS server
- ${ipoib_netmask} # Subnet netmask for internal IPoIB
- ${c_ipoib[0]}, ${c_ipoib[1]}, ... # IPoIB addresses for computes
- ${nagios_web_password} # Nagios web access password

# 2 Obtaining and downloading the Bright ISO

If you do not already have a Bright product key, you can get a quote by visiting the following page and Bright Computing will contact you with a quote: Get A Quote. Alternatively, Bright also offers Easy8, which offers installations of Bright on a cluster of up to 8 nodes free of charge. Please note that Easy8 does not have access to Bright Computing support. You can find more about Easy8 from the following link: Easy8.

Once you have your Bright product key, you can then proceed to download Bright from the Bright Customer Portal by selecting the "Download ISO" link in the left menu pane.

# 3 Install Operating System (OS)

In an external setting, installing the desired BOS and Bright on a *master* SMS host typically involves booting from a DVD ISO image on a new server. With this approach, insert the Bright/CentOS8.2 DVD, power cycle the host, and follow the Bright provided directions to install the BOS on your chosen *master* host. Alternatively, if choosing to use a pre-installed server, please verify that it is provisioned with the required CentOS8.2 distribution. If Bright is not already installed on the *master* host, please follow the steps in the Bright installation manual on performing an addon installation

In depth instructions of installing Bright/CentOS8.2 are covered in the Bright Installation Manual.

If the SLURM workload manager is not selected during the Bright *master* installation and you intend to install SLURM afterwards, please refer to the Installation of Workload Managers section of the Bright administration manual.

Prior to beginning the installation process of OpenHPC components, several additional considerations are noted here for the SMS host configuration. First, the installation recipe herein assumes that the SMS

host name is resolvable locally. Since Bright manages this part of the `/etc/hosts` file automatically, it should not be required to make manual modifications to the `/etc/hosts` file.

While it is theoretically possible to enable SELinux on a cluster provisioned with bright, doing so is beyond the scope of this document. Even the use of permissive mode can be problematic and we therefore recommend keeping SELinux disabled on the *master* SMS host. If SELinux components are installed locally, the `selinuxenabled` command can be used to determine if SELinux is currently enabled. If enabled, consult the distro documentation for information on how to disable.

Finally, the Bright node-installer provisioning rely on DHCP, TFTP, HTTP and rsync (or optionally rsync over SSH) network protocols. For the internal network, Bright is automatically configured to allow these protocols to the *master*, so out of the box, no additional manual configuration should be required.

# 4 Install OpenHPC Components

With the BOS installed and booted, the next step is to add desired OpenHPC packages onto the *master* server in order to provide provisioning and resource management services for the rest of the cluster. The following subsections highlight this process.

## 4.1 Enable OpenHPC repository for local use

To begin, enable use of the OpenHPC repository by adding it to the local list of available package repositories. Note that this requires network access from your *master* server to the OpenHPC repository, or alternatively, that the OpenHPC repository be mirrored locally. In cases where network external connectivity is available, OpenHPC provides an `ohpc-release` package that includes GPG keys for package signing and enabling the repository. The example which follows illustrates installation of the `ohpc-release` package directly from the OpenHPC build server.

```
[sms]# yum install https://support2.brightcomputing.com/ohpc/ohpc-reposetup-2.0_100008_cm9.0_590aab9b26.x86_64.rpm
```

After installing the ohpc-reposetup Bright package on the *master* node, the ohpc-reposetup module needs to be loaded, then the ohpc-reposetup.sh script run to enable OpenHPC in the software images and the *master* node

```
[sms]# module load ohpc-reposetup/2.0/ohpc-reposetup.module
[sms]# ohpc-reposetup.sh
```

### 4.2 Boot compute nodes

Prior to booting the *compute* hosts, we configure them to use PXE as their next boot mode. After the initial PXE, since Bright installs iPXE on the node disk drive, ensuing boots will continue to attempt to PXE boot from the network.

During the initial boot of the nodes, when the Bright node installer runs, you will be asked to assign each of the compute nodes an identity. This only needs to occur on the initial boot as Bright will record the node MAC address in it's database so that on subsequent boots, Bright will know which node is being provisioned. For more information regarding the node boot process, please refer to the Node-Installer section of the Bright admin manual.

If you wish to define additional compute nodes than what was defined during the *master* installation, the steps to add additional compute nodes to Bright can be found in the Adding New Nodes section of the Bright admin manual.

After the initial boot, providing the nodes BMC interfaces have been configured per section 3.7 of the Bright administration manual, the *master* server should be able to boot the compute nodes. This is done by using the `power` Bright command leverging IPMI protocol. The following power cycles a Bright node.

```
[sms]# cmsh
% device use <node_name>
% power reset
```

It is also possible to power cycle multiple nodes at once, for example, to power cycle all nodes in the default category, the following can be performed.

```
[sms]# cmsh
% device
% power reset -c default
```

## 5 Install OpenHPC Development Components

The install procedure outlined in §4 highlighted the steps necessary to install a *master* host, assemble and customize a *compute* image, and provision several compute hosts from bare-metal. With these steps completed, additional OpenHPC-provided packages can now be added to support a flexible HPC development environment including development tools, C/C++/FORTRAN compilers, MPI stacks, and a variety of 3rd party libraries. The following subsections highlight the additional software installation procedures.

### 5.1 Development Tools

To aid in general development efforts, OpenHPC provides recent versions of the GNU autotools collection, the Valgrind memory debugger, EasyBuild, and Spack. These can be installed as follows:

```
# Install autotools meta-package
[sms]# yum -y install ohpc-autotools

[sms]# yum -y install EasyBuild-ohpc
[sms]# yum -y install hwloc-ohpc
[sms]# yum -y install spack-ohpc
[sms]# yum -y install valgrind-ohpc
```

## 5.2 Compilers

OpenHPC presently packages the GNU compiler toolchain integrated with the underlying Lmod modules system in a hierarchical fashion. The modules system will conditionally present compiler-dependent software based on the toolchain currently loaded.

```
[sms]# yum -y install gnu9-compilers-ohpc
```

## 5.3 MPI Stacks

For MPI development and runtime support, OpenHPC provides pre-packaged builds for a variety of MPI families and transport layers. Currently available options and their applicability to various network transports are summarized in Table 1. The command that follows installs a starting set of MPI families that are compatible with both ethernet and high-speed fabrics.

Table 1: Available MPI variants

| | Ethernet (TCP) | InfiniBand | Intel® Omni-Path |
|---|---|---|---|
| MPICH (ofi) | ✓ | ✓ | ✓ |
| MPICH (ucx) | ✓ | ✓ | ✓ |
| MVAPICH2 | | ✓ | |
| MVAPICH2 (psm2) | | | ✓ |
| OpenMPI (ofi/ucx) | ✓ | ✓ | ✓ |

```
[sms]# yum -y install openmpi4-gnu9-ohpc mpich-ofi-gnu9-ohpc
```

Note that OpenHPC 2.x introduces the use of two related transport layers for the MPICH and OpenMPI builds that support a variety of underlying fabrics: UCX (Unified Communication X) and OFI (OpenFabrics interfaces). In the case of OpenMPI, a monolithic build is provided which supports both transports and end-users can customize their runtime preferences with environment variables. For MPICH, two separate builds are provided and the example above highlighted installing the ofi variant. However, the packaging is designed such that both versions can be installed simultaneously and users can switch between the two via normal module command semantics. Alternatively, a site can choose to install the ucx variant instead as a drop-in MPICH replacement:

```
[sms]# yum -y install mpich-ucx-gnu9-ohpc
```

In the case where both MPICH variants are installed, two modules will be visible in the end-user environment and an example of this configuration is highlighted is below.

```
[sms]# module avail mpich

-------------------- /opt/ohpc/pub/moduledeps/gnu9 ---------------------
   mpich/3.3.2-ofi  mpich/3.3.2-ucx (D)
```

If your system includes InfiniBand and you enabled underlying support in §?? and §??, an additional MVAPICH2 family is available for use:

```
[sms]# yum -y install mvapich2-gnu9-ohpc
```

Alternatively, if your system includes Intel® Omni-Path, use the (`psm2`) variant of MVAPICH2 instead:

```
[sms]# yum -y install mvapich2-psm2-gnu9-ohpc
```

An additional OpenMPI build variant is listed in Table 1 which enables PMIx job launch support for use with Slurm. This optional variant is available as `openmpi4-pmix-slurm-gnu9-ohpc`.

## 5.4 Performance Tools

OpenHPC provides a variety of open-source tools to aid in application performance analysis (refer to Appendix A for a listing of available packages). This group of tools can be installed as follows:

```
# Install perf-tools meta-package
[sms]# yum -y install ohpc-gnu9-perf-tools
```

## 5.5 Setup default development environment

System users often find it convenient to have a default development environment in place so that compilation can be performed directly for parallel programs requiring MPI. This setup can be conveniently enabled via modules and the OpenHPC modules environment is pre-configured to load an `ohpc` module on login (if present). The following package install provides a default environment that enables autotools, the GNU compiler toolchain, and the OpenMPI stack.

```
[sms]# yum -y install lmod-defaults-gnu9-openmpi4-ohpc
```

**Tip**

If you want to change the default environment from the suggestion above, OpenHPC also provides the GNU compiler toolchain with the MPICH and MVAPICH2 stacks:

- lmod-defaults-gnu9-mpich-ofi-ohpc
- lmod-defaults-gnu9-mpich-ucx-ohpc
- lmod-defaults-gnu9-mvapich2-ohpc

## 5.6 3rd Party Libraries and Tools

OpenHPC provides pre-packaged builds for a number of popular open-source tools and libraries used by HPC applications and developers. For example, OpenHPC provides builds for FFTW and HDF5 (including serial and parallel I/O support), and the GNU Scientific Library (GSL). Again, multiple builds of each package are available in the OpenHPC repository to support multiple compiler and MPI family combinations where appropriate. Note, however, that not all combinatorial permutations may be available for components where there are known license incompatibilities. The general naming convention for builds provided by OpenHPC is to append the compiler and MPI family name that the library was built against directly into the package name. For example, libraries that do not require MPI as part of the build process adopt the following RPM name:

`package-<compiler_family>-ohpc-<package_version>-<release>.rpm`

Packages that do require MPI as part of the build expand upon this convention to additionally include the MPI family name as follows:

```
package-<compiler_family>-<mpi_family>-ohpc-<package_version>-<release>.rpm
```

To illustrate this further, the command below queries the locally configured repositories to identify all of the available PETSc packages that were built with the GNU toolchain. The resulting output that is included shows that pre-built versions are available for each of the supported MPI families presented in §5.3.

**Tip**

OpenHPC-provided 3rd party builds are configured to be installed into a common top-level repository so that they can be easily exported to desired hosts within the cluster. This common top-level path (`/opt/ohpc/pub`) was previously configured to be mounted on *compute* nodes in §**??**, so the packages will be immediately available for use on the cluster after installation on the *master* host.

For convenience, OpenHPC provides package aliases for these 3rd party libraries and utilities that can be used to install available libraries for use with the GNU compiler family toolchain. For parallel libraries, aliases are grouped by MPI family toolchain so that administrators can choose a subset should they favor a particular MPI stack. Please refer to Appendix A for a more detailed listing of all available packages in each of these functional areas. To install all available package offerings within OpenHPC, issue the following:

```
# Install 3rd party libraries/tools meta-packages built with GNU toolchain
[sms]# yum -y install ohpc-gnu9-serial-libs
[sms]# yum -y install ohpc-gnu9-io-libs
[sms]# yum -y install ohpc-gnu9-python-libs
[sms]# yum -y install ohpc-gnu9-runtimes
```

```
# Install parallel lib meta-packages for all available MPI toolchains
[sms]# yum -y install ohpc-gnu9-mpich-parallel-libs
[sms]# yum -y install ohpc-gnu9-openmpi4-parallel-libs
```

## 5.7 Optional Development Tool Builds

In addition to the 3rd party development libraries built using the open source toolchains mentioned in §5.6, OpenHPC also provides *optional* compatible builds for use with the compilers and MPI stack included in newer versions of the Intel® Parallel Studio XE software suite. These packages provide a similar hierarchical user environment experience as other compiler and MPI families present in OpenHPC.

To take advantage of the available builds, the Parallel Studio software suite must be obtained and installed separately. Once installed locally, the OpenHPC compatible packages can be installed using standard package manager semantics. Note that licenses are provided free of charge for many categories of use. In particular, licenses for compilers and developments tools are provided at no cost to academic researchers or developers contributing to open-source software projects. More information on this program can be found at:

https://software.intel.com/en-us/qualify-for-free-software

**Tip**

As noted in §??, the default installation path for OpenHPC (`/opt/ohpc/pub`) is exported over NFS from the *master* to the compute nodes, but the Parallel Studio installer defaults to a path of `/opt/intel`. To make the Intel® compilers available to the compute nodes one must either customize the Parallel Studio installation path to be within `/opt/ohpc/pub`, or alternatively, add an additional NFS export for `/opt/intel` that is mounted on desired compute nodes.

To enable all 3rd party builds available in OpenHPC that are compatible with Intel® Parallel Studio, issue the following:

```
# Install OpenHPC compatibility packages (requires prior installation of Parallel Studio)
[sms]# yum -y install intel-compilers-devel-ohpc
[sms]# yum -y install intel-mpi-devel-ohpc
```

```
# Optionally, choose the Omni-Path enabled build for MVAPICH2. Otherwise, skip to retain IB variant
[sms]# yum -y install mvapich2-psm2-intel-ohpc
```

```
# Install 3rd party libraries/tools meta-packages built with Intel toolchain
[sms]# yum -y install ohpc-intel-serial-libs
[sms]# yum -y install ohpc-intel-io-libs
[sms]# yum -y install ohpc-intel-perf-tools
[sms]# yum -y install ohpc-intel-python3-libs
[sms]# yum -y install ohpc-intel-runtimes
[sms]# yum -y install ohpc-intel-mpich-parallel-libs
[sms]# yum -y install ohpc-intel-mvapich2-parallel-libs
[sms]# yum -y install ohpc-intel-openmpi4-parallel-libs
[sms]# yum -y install ohpc-intel-impi-parallel-libs
```

# Appendices

# A Package Manifest

This appendix provides a summary of available meta-package groupings and all of the individual RPM packages that are available as part of this OpenHPC release. The meta-packages provide a mechanism to group related collections of RPMs by functionality and provide a convenience mechanism for installation. A list of the available meta-packages and a brief description is presented in Table 2.

Table 2: **Available OpenHPC Meta-packages**

| Group Name | Description |
|---|---|
| ohpc-autotools | Collection of GNU autotools packages. |
| ohpc-base | Collection of base packages. |
| ohpc-base-compute | Collection of compute node base packages. |
| ohpc-gnu9-geopm | Global Extensible Open Power Manager for use with GNU compiler toolchain. |
| ohpc-gnu9-io-libs | Collection of IO library builds for use with GNU compiler toolchain. |
| ohpc-gnu9-mpich-io-libs | Collection of IO library builds for use with GNU compiler toolchain and the MPICH runtime. |
| ohpc-gnu9-mpich-parallel-libs | Collection of parallel library builds for use with GNU compiler toolchain and the MPICH runtime. |
| ohpc-gnu9-mpich-perf-tools | Collection of performance tool builds for use with GNU compiler toolchain and the MPICH runtime. |
| ohpc-gnu9-mvapich2-io-libs | Collection of IO library builds for use with GNU compiler toolchain and the MVAPICH2 runtime. |
| ohpc-gnu9-mvapich2-parallel-libs | Collection of parallel library builds for use with GNU compiler toolchain and the MVAPICH2 runtime. |
| ohpc-gnu9-mvapich2-perf-tools | Collection of performance tool builds for use with GNU compiler toolchain and the MVAPICH2 runtime. |
| ohpc-gnu9-openmpi4-io-libs | Collection of IO library builds for use with GNU compiler toolchain and the OpenMPI runtime. |
| ohpc-gnu9-openmpi4-parallel-libs | Collection of parallel library builds for use with GNU compiler toolchain and the OpenMPI runtime. |
| ohpc-gnu9-openmpi4-perf-tools | Collection of performance tool builds for use with GNU compiler toolchain and the OpenMPI runtime. |
| ohpc-gnu9-parallel-libs | Collection of parallel library builds for use with GNU compiler toolchain. |
| ohpc-gnu9-perf-tools | Collection of performance tool builds for use with GNU compiler toolchain. |
| ohpc-gnu9-python-libs | Collection of python related library builds for use with GNU compiler toolchain. |
| ohpc-gnu9-python3-libs | Collection of python3 related library builds for use with GNU compiler toolchain. |
| ohpc-gnu9-runtimes | Collection of runtimes for use with GNU compiler toolchain. |
| ohpc-gnu9-serial-libs | Collection of serial library builds for use with GNU compiler toolchain. |
| ohpc-intel-geopm | Global Extensible Open Power Manager for use with Intel(R) Parallel Studio XE software suite. |
| ohpc-intel-impi-io-libs | Collection of IO library builds for use with Intel(R) Parallel Studio XE software suite and Intel(R) MPI runtime. |
| ohpc-intel-impi-parallel-libs | Collection of parallel library builds for use with Intel(R) Parallel Studio XE toolchain and the Intel(R) MPI Library. |
| ohpc-intel-impi-perf-tools | Collection of performance tool builds for use with Intel(R) Parallel Studio XE compiler toolchain and the Intel(R) MPI runtime. |
| ohpc-intel-io-libs | Collection of IO library builds for use with Intel(R) Parallel Studio XE software suite. |
| ohpc-intel-mpich-io-libs | Collection of IO library builds for use with Intel(R) Parallel Studio XE software suite and MPICH runtime. |
| ohpc-intel-mpich-parallel-libs | Collection of parallel library builds for use with Intel(R) Parallel Studio XE toolchain and the MPICH runtime. |
| ohpc-intel-mpich-perf-tools | Collection of performance tool builds for use with Intel(R) Parallel Studio XE compiler toolchain and the MPICH runtime. |

Table 2 (cont): **Available OpenHPC Meta-packages**

| Group Name | Description |
|---|---|
| ohpc-intel-mvapich2-io-libs | Collection of IO library builds for use with Intel(R) Parallel Studio XE software suite and MVAPICH2 runtime. |
| ohpc-intel-mvapich2-parallel-libs | Collection of parallel library builds for use with Intel(R) Parallel Studio XE toolchain and the MVAPICH2 runtime. |
| ohpc-intel-mvapich2-perf-tools | Collection of performance tool builds for use with Intel(R) Parallel Studio XE compiler toolchain and the MVAPICH2 runtime. |
| ohpc-intel-openmpi4-io-libs | Collection of IO library builds for use with Intel(R) Parallel Studio XE software suite and OpenMPI runtime. |
| ohpc-intel-openmpi4-parallel-libs | Collection of parallel library builds for use with Intel(R) Parallel Studio XE toolchain and the OpenMPI runtime. |
| ohpc-intel-openmpi4-perf-tools | Collection of performance tool builds for use with Intel(R) Parallel Studio XE compiler toolchain and the OpenMPI runtime. |
| ohpc-intel-perf-tools | Collection of performance tool builds for use with Intel(R) Parallel Studio XE toolchain. |
| ohpc-intel-python3-libs | Collection of python3 related library builds for use with Intel(R) Parallel Studio XE toolchain. |
| ohpc-intel-runtimes | Collection of runtimes for use with Intel(R) Parallel Studio XE toolchain. |
| ohpc-intel-serial-libs | Collection of serial library builds for use with Intel(R) Parallel Studio XE toolchain. |
| ohpc-slurm-client | Collection of client packages for SLURM. |
| ohpc-slurm-server | Collection of server packages for SLURM. |
| ohpc-warewulf | Collection of base packages for Warewulf provisioning. |

What follows next in this Appendix is a series of tables that summarize the underlying RPM packages available in this OpenHPC release. These packages are organized by groupings based on their general functionality and each table provides information for the specific RPM name, version, brief summary, and the web URL where additional information can be obtained for the component. Note that many of the 3rd party community libraries that are pre-packaged with OpenHPC are built using multiple compiler and MPI families. In these cases, the RPM package name includes delimiters identifying the development environment for which each package build is targeted. Additional information on the OpenHPC package naming scheme is presented in §5.6. The relevant package groupings and associated Table references are as follows:

- Administrative tools (Table 3)
- Resource management (Table 4)
- Compiler families (Table 5)
- MPI families (Table 6)
- Development tools (Table 7)
- Performance analysis tools (Table 8)
- IO Libraries (Table 10)
- Runtimes (Table 11)
- Serial/Threaded Libraries (Table 12)
- Parallel Libraries (Table 13)

Table 3: **Administrative Tools**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| conman-ohpc | 0.3.0 | ConMan: The Console Manager. http://dun.github.io/conman |
| docs-ohpc | 2.0.0 | OpenHPC documentation. https://github.com/openhpc/ohpc |
| examples-ohpc | 2.0 | Example source code and templates for use within OpenHPC environment. https://github.com/openhpc/ohpc |
| genders-ohpc | 1.27 | Static cluster configuration database. https://github.com/chaos/genders |
| lmod-defaults-gnu9-impi-ohpc<br>lmod-defaults-gnu9-mpich-ofi-ohpc<br>lmod-defaults-gnu9-mpich-ucx-ohpc<br>lmod-defaults-gnu9-mvapich2-ohpc<br>lmod-defaults-gnu9-openmpi4-ohpc<br>lmod-defaults-intel-impi-ohpc<br>lmod-defaults-intel-mpich-ohpc<br>lmod-defaults-intel-mvapich2-ohpc<br>lmod-defaults-intel-openmpi4-ohpc | 2.0 | OpenHPC default login environments. https://github.com/openhpc/ohpc |
| lmod-ohpc | 8.2.10 | Lua based Modules (lmod). https://github.com/TACC/Lmod |
| losf-ohpc | 0.56.0 | A Linux operating system framework for managing HPC clusters. https://github.com/hpcsi/losf |
| mrsh-ohpc | 2.12 | Remote shell program that uses munge authentication. https://github.com/chaos/mrsh |
| nhc-ohpc | 1.4.2 | LBNL Node Health Check. https://github.com/mej/nhc |
| ohpc-release | 2 | OpenHPC release files. https://github.com/openhpc/ohpc |
| pdsh-ohpc | 2.34 | Parallel remote shell program. https://github.com/chaos/pdsh |
| prun-ohpc | 2.0 | Convenience utility for parallel job launch. https://github.com/openhpc/ohpc |
| test-suite-ohpc | 2.0.0 | Integration test suite for OpenHPC. https://github.com/openhpc/ohpc/tests |

Table 4: **Resource Management**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| openpbs-execution-ohpc | 20.0.0 | OpenPBS for an execution host. http://www.openpbs.org |
| openpbs-client-ohpc | 20.0.0 | OpenPBS for a client host. http://www.openpbs.org |
| openpbs-server-ohpc | 20.0.0 | OpenPBS for a server host. http://www.openpbs.org |
| pmix-ohpc | 3.1.4 | An extended/exascale implementation of PMI. https://pmix.github.io/pmix |
| slurm-devel-ohpc | 20.02.5 | Development package for Slurm. https://slurm.schedmd.com |
| slurm-example-configs-ohpc | 20.02.5 | Example config files for Slurm. https://slurm.schedmd.com |
| slurm-pam_slurm-ohpc | 20.02.5 | PAM module for restricting access to compute nodes via Slurm. https://slurm.schedmd.com |
| slurm-perlapi-ohpc | 20.02.5 | Perl API to Slurm. https://slurm.schedmd.com |
| slurm-contribs-ohpc | 20.02.5 | Perl tool to print Slurm job state information. https://slurm.schedmd.com |
| slurm-ohpc | 20.02.5 | Slurm Workload Manager. https://slurm.schedmd.com |
| slurm-slurmd-ohpc | 20.02.5 | Slurm compute node daemon. https://slurm.schedmd.com |
| slurm-slurmctld-ohpc | 20.02.5 | Slurm controller daemon. https://slurm.schedmd.com |
| slurm-slurmdbd-ohpc | 20.02.5 | Slurm database daemon. https://slurm.schedmd.com |
| slurm-libpmi-ohpc | 20.02.5 | Slurmś implementation of the pmi libraries. https://slurm.schedmd.com |
| slurm-torque-ohpc | 20.02.5 | Torque/PBS wrappers for transition from Torque/PBS to Slurm. https://slurm.schedmd.com |
| slurm-openlava-ohpc | 20.02.5 | openlava/LSF wrappers for transition from OpenLava/LSF to Slurm. https://slurm.schedmd.com |

Table 5: **Compiler Families**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| gnu9-compilers-ohpc | 9.3.0 | The GNU C Compiler and Support Files. http://gcc.gnu.org |
| intel-compilers-devel-ohpc | 2020 | OpenHPC compatibility package for Intel(R) Parallel Studio XE. https://github.com/openhpc/ohpc |

Table 6: **MPI Families / Communication Libraries**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| intel-mpi-devel-ohpc | 2020 | OpenHPC compatibility package for Intel(R) MPI Library. https://github.com/openhpc/ohpc |
| libfabric-ohpc | 1.10.1 | User-space RDMA Fabric Interfaces. http://www.github.com/ofiwg/libfabric |
| mpich-ofi-gnu9-ohpc<br>mpich-ofi-intel-ohpc | 3.3.2 | MPICH MPI implementation. http://www.mpich.org |
| mpich-ucx-gnu9-ohpc<br>mpich-ucx-intel-ohpc | 3.3.2 | MPICH MPI implementation. http://www.mpich.org |
| mvapich2-gnu9-ohpc<br>mvapich2-intel-ohpc<br>mvapich2-psm2-gnu9-ohpc<br>mvapich2-psm2-intel-ohpc | 2.3.2 | OSU MVAPICH2 MPI implementation. http://mvapich.cse.ohio-state.edu |
| openmpi4-gnu9-ohpc<br>openmpi4-intel-ohpc | 4.0.4 | A powerful implementation of MPI/SHMEM. http://www.open-mpi.org |
| ucx-ohpc | 1.8.0 | UCX is a communication library implementing high-performance messaging. http://www.openucx.org |

Table 7: **Development Tools**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| EasyBuild-ohpc | 4.3.0 | Software build and installation framework. https://easybuilders.github.io/easybuild |
| automake-ohpc | 1.16.1 | A GNU tool for automatically creating Makefiles. http://www.gnu.org/software/automake |
| autoconf-ohpc | 2.69 | A GNU tool for automatically configuring source code. http://www.gnu.org/software/autoconf |
| cmake-ohpc | 3.16.2 | CMake is an open-source, cross-platform family of tools designed to build, test and package software. https://cmake.org |
| hwloc-ohpc | 2.1.0 | Portable Hardware Locality. http://www.open-mpi.org/projects/hwloc |
| libtool-ohpc | 2.4.6 | The GNU Portable Library Tool. http://www.gnu.org/software/libtool |
| python3-scipy-gnu9-mpich-ohpc<br>python3-scipy-gnu9-mvapich2-ohpc<br>python3-scipy-gnu9-openmpi4-ohpc | 1.5.1 | Scientific Tools for Python. http://www.scipy.org |
| python3-numpy-gnu9-ohpc<br>python3-numpy-intel-ohpc | 1.19.0 | NumPy array processing for numbers, strings, records and objects. https://github.com/numpy/numpy |
| python3-mpi4py-gnu9-impi-ohpc<br>python3-mpi4py-gnu9-mpich-ohpc<br>python3-mpi4py-gnu9-mvapich2-ohpc<br>python3-mpi4py-gnu9-openmpi4-ohpc<br>python3-mpi4py-intel-impi-ohpc<br>python3-mpi4py-intel-mpich-ohpc<br>python3-mpi4py-intel-mvapich2-ohpc<br>python3-mpi4py-intel-openmpi4-ohpc | 3.0.3 | Python bindings for the Message Passing Interface (MPI) standard. https://bitbucket.org/mpi4py/mpi4py |
| spack-ohpc | 0.15.0 | HPC software package management. https://github.com/LLNL/spack |
| valgrind-ohpc | 3.15.0 | Valgrind Memory Debugger. http://www.valgrind.org |

Table 8: **Performance Analysis Tools**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| dimemas-gnu9-impi-ohpc<br>dimemas-gnu9-mpich-ohpc<br>dimemas-gnu9-mvapich2-ohpc<br>dimemas-gnu9-openmpi4-ohpc<br>dimemas-intel-impi-ohpc<br>dimemas-intel-mpich-ohpc<br>dimemas-intel-mvapich2-ohpc<br>dimemas-intel-openmpi4-ohpc | 5.4.2 | Dimemas tool.<br>https://tools.bsc.es |
| extrae-gnu9-impi-ohpc<br>extrae-gnu9-mpich-ohpc<br>extrae-gnu9-mvapich2-ohpc<br>extrae-gnu9-openmpi4-ohpc<br>extrae-intel-impi-ohpc<br>extrae-intel-mpich-ohpc<br>extrae-intel-mvapich2-ohpc<br>extrae-intel-openmpi4-ohpc | 3.7.0 | Extrae tool.<br>https://tools.bsc.es |
| geopm-gnu9-impi-ohpc<br>geopm-gnu9-mpich-ohpc<br>geopm-gnu9-mvapich2-ohpc<br>geopm-gnu9-openmpi4-ohpc<br>geopm-intel-impi-ohpc<br>geopm-intel-mpich-ohpc<br>geopm-intel-mvapich2-ohpc<br>geopm-intel-openmpi4-ohpc | 1.1.0 | Global Extensible Open Power Manager.<br>https://geopm.github.io |
| imb-gnu9-impi-ohpc<br>imb-gnu9-mpich-ohpc<br>imb-gnu9-mvapich2-ohpc<br>imb-gnu9-openmpi4-ohpc<br>imb-intel-impi-ohpc<br>imb-intel-mpich-ohpc<br>imb-intel-mvapich2-ohpc<br>imb-intel-openmpi4-ohpc | 2019.6 | Intel MPI Benchmarks (IMB).<br>https:<br>//software.intel.com/en-us/articles/intel-mpi-benchmarks |
| likwid-gnu9-ohpc<br>likwid-intel-ohpc | 5.0.1 | Performance tools for the Linux console.<br>https://github.com/RRZE-HPC/likwid |
| msr-safe-ohpc | 1.4.0 | Allows safer access to model specific registers (MSRs).<br>https://github.com/LLNL/msr-safe |
| omb-gnu9-impi-ohpc<br>omb-gnu9-mpich-ohpc<br>omb-gnu9-mvapich2-ohpc<br>omb-gnu9-openmpi4-ohpc<br>omb-intel-impi-ohpc<br>omb-intel-mpich-ohpc<br>omb-intel-mvapich2-ohpc<br>omb-intel-openmpi4-ohpc | 5.6.2 | OSU Micro-benchmarks.<br>http://mvapich.cse.ohio-state.edu/benchmarks |
| paraver-ohpc | 4.8.2 | Paraver.<br>https://tools.bsc.es |
| papi-ohpc | 5.7.0 | Performance Application Programming Interface.<br>http://icl.cs.utk.edu/papi |

Table 8 (cont): **Performance Analysis Tools**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| pdtoolkit-gnu9-ohpc<br>pdtoolkit-intel-ohpc | 3.25.1 | PDT is a framework for analyzing source code.<br>http://www.cs.uoregon.edu/Research/pdt |
| scalasca-gnu9-impi-ohpc<br>scalasca-gnu9-mpich-ohpc<br>scalasca-gnu9-mvapich2-ohpc<br>scalasca-gnu9-openmpi4-ohpc<br>scalasca-intel-impi-ohpc<br>scalasca-intel-mpich-ohpc<br>scalasca-intel-mvapich2-ohpc<br>scalasca-intel-openmpi4-ohpc | 2.5 | Toolset for performance analysis of large-scale parallel applications.<br>http://www.scalasca.org |
| scorep-gnu9-impi-ohpc<br>scorep-gnu9-mpich-ohpc<br>scorep-gnu9-mvapich2-ohpc<br>scorep-gnu9-openmpi4-ohpc<br>scorep-intel-impi-ohpc<br>scorep-intel-mpich-ohpc<br>scorep-intel-mvapich2-ohpc<br>scorep-intel-openmpi4-ohpc | 6.0 | Scalable Performance Measurement Infrastructure for Parallel Codes.<br>http://www.vi-hps.org/projects/score-p |
| tau-gnu9-impi-ohpc<br>tau-gnu9-mpich-ohpc<br>tau-gnu9-mvapich2-ohpc<br>tau-gnu9-openmpi4-ohpc<br>tau-intel-impi-ohpc<br>tau-intel-mpich-ohpc<br>tau-intel-mvapich2-ohpc<br>tau-intel-openmpi4-ohpc | 2.29 | Tuning and Analysis Utilities Profiling Package.<br>http://www.cs.uoregon.edu/research/tau/home.php |

Table 9: **Lustre**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| lustre-client-ohpc | 2.12.5 | Lustre File System.<br>https://wiki.whamcloud.com |

Table 10: **IO Libraries**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| adios-gnu9-impi-ohpc<br>adios-gnu9-mpich-ohpc<br>adios-gnu9-mvapich2-ohpc<br>adios-gnu9-openmpi4-ohpc<br>adios-intel-impi-ohpc<br>adios-intel-mpich-ohpc<br>adios-intel-mvapich2-ohpc<br>adios-intel-openmpi4-ohpc | 1.13.1 | The Adaptable IO System (ADIOS).<br>http://www.olcf.ornl.gov/center-projects/adios |
| hdf5-gnu9-ohpc<br>hdf5-intel-ohpc | 1.10.6 | A general purpose library and file format for storing scientific data. http://www.hdfgroup.org/HDF5 |
| netcdf-cxx-gnu9-impi-ohpc<br>netcdf-cxx-gnu9-mpich-ohpc<br>netcdf-cxx-gnu9-mvapich2-ohpc<br>netcdf-cxx-gnu9-openmpi4-ohpc<br>netcdf-cxx-intel-impi-ohpc<br>netcdf-cxx-intel-mpich-ohpc<br>netcdf-cxx-intel-mvapich2-ohpc<br>netcdf-cxx-intel-openmpi4-ohpc | 4.3.1 | C++ Libraries for the Unidata network Common Data Form.<br>http://www.unidata.ucar.edu/software/netcdf |
| netcdf-fortran-gnu9-impi-ohpc<br>netcdf-fortran-gnu9-mpich-ohpc<br>netcdf-fortran-gnu9-mvapich2-ohpc<br>netcdf-fortran-gnu9-openmpi4-ohpc<br>netcdf-fortran-intel-impi-ohpc<br>netcdf-fortran-intel-mpich-ohpc<br>netcdf-fortran-intel-mvapich2-ohpc<br>netcdf-fortran-intel-openmpi4-ohpc | 4.5.2 | Fortran Libraries for the Unidata network Common Data Form.<br>http://www.unidata.ucar.edu/software/netcdf |
| netcdf-gnu9-impi-ohpc<br>netcdf-gnu9-mpich-ohpc<br>netcdf-gnu9-mvapich2-ohpc<br>netcdf-gnu9-openmpi4-ohpc<br>netcdf-intel-impi-ohpc<br>netcdf-intel-mpich-ohpc<br>netcdf-intel-mvapich2-ohpc<br>netcdf-intel-openmpi4-ohpc | 4.7.3 | C Libraries for the Unidata network Common Data Form.<br>http://www.unidata.ucar.edu/software/netcdf |
| phdf5-gnu9-impi-ohpc<br>phdf5-gnu9-mpich-ohpc<br>phdf5-gnu9-mvapich2-ohpc<br>phdf5-gnu9-openmpi4-ohpc<br>phdf5-intel-impi-ohpc<br>phdf5-intel-mpich-ohpc<br>phdf5-intel-mvapich2-ohpc<br>phdf5-intel-openmpi4-ohpc | 1.10.6 | A general purpose library and file format for storing scientific data.<br>http://www.hdfgroup.org/HDF5 |

Table 10 (cont): **IO Libraries**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| pnetcdf-gnu9-impi-ohpc<br>pnetcdf-gnu9-mpich-ohpc<br>pnetcdf-gnu9-mvapich2-ohpc<br>pnetcdf-gnu9-openmpi4-ohpc<br>pnetcdf-intel-impi-ohpc<br>pnetcdf-intel-mpich-ohpc<br>pnetcdf-intel-mvapich2-ohpc<br>pnetcdf-intel-openmpi4-ohpc | 1.12.1 | A Parallel NetCDF library (PnetCDF).<br>http://cucis.ece.northwestern.edu/projects/PnetCDF |
| sionlib-gnu9-impi-ohpc<br>sionlib-gnu9-mpich-ohpc<br>sionlib-gnu9-mvapich2-ohpc<br>sionlib-gnu9-openmpi4-ohpc<br>sionlib-intel-impi-ohpc<br>sionlib-intel-mpich-ohpc<br>sionlib-intel-mvapich2-ohpc<br>sionlib-intel-openmpi4-ohpc | 1.7.4 | Scalable I/O Library for Parallel Access to Task-Local Files.<br>http://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/SIONlib/_node.html |

Table 11: **Runtimes**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| charliecloud-ohpc | 0.15 | Lightweight user-defined software stacks for high-performance computing. https://hpc.github.io/charliecloud |
| singularity-ohpc | 3.4.2 | Application and environment virtualization. https://www.sylabs.io/singularity |

Table 12: **Serial/Threaded Libraries**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| R-gnu9-ohpc | 3.6.3 | R is a language and environment for statistical computing and graphics (S-Plus like). http://www.r-project.org |
| gsl-gnu9-ohpc<br>gsl-intel-ohpc | 2.6 | GNU Scientific Library (GSL). http://www.gnu.org/software/gsl |
| metis-gnu9-ohpc<br>metis-intel-ohpc | 5.1.0 | Serial Graph Partitioning and Fill-reducing Matrix Ordering. http://glaros.dtc.umn.edu/gkhome/metis/metis/overview |
| openblas-gnu9-ohpc | 0.3.7 | An optimized BLAS library based on GotoBLAS2. http://www.openblas.net |
| plasma-gnu9-ohpc<br>plasma-intel-ohpc | 2.8.0 | Parallel Linear Algebra Software for Multicore Architectures. https://bitbucket.org/icl/plasma |
| scotch-gnu9-ohpc<br>scotch-intel-ohpc | 6.0.6 | Graph, mesh and hypergraph partitioning library. http://www.labri.fr/perso/pelegrin/scotch |
| superlu-gnu9-ohpc<br>superlu-intel-ohpc | 5.2.1 | A general purpose library for the direct solution of linear equations. http://crd.lbl.gov/~xiaoye/SuperLU |

Table 13: **Parallel Libraries**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| boost-gnu9-impi-ohpc<br>boost-gnu9-mpich-ohpc<br>boost-gnu9-mvapich2-ohpc<br>boost-gnu9-openmpi4-ohpc<br>boost-intel-impi-ohpc<br>boost-intel-mpich-ohpc<br>boost-intel-mvapich2-ohpc<br>boost-intel-openmpi4-ohpc | 1.73.0 | Boost free peer-reviewed portable C++ source libraries.<br>http://www.boost.org |
| fftw-gnu9-impi-ohpc<br>fftw-gnu9-mpich-ohpc<br>fftw-gnu9-mvapich2-ohpc<br>fftw-gnu9-openmpi4-ohpc | 3.3.8 | A Fast Fourier Transform library.<br>http://www.fftw.org |
| hypre-gnu9-impi-ohpc<br>hypre-gnu9-mpich-ohpc<br>hypre-gnu9-mvapich2-ohpc<br>hypre-gnu9-openmpi4-ohpc<br>hypre-intel-impi-ohpc<br>hypre-intel-mpich-ohpc<br>hypre-intel-mvapich2-ohpc<br>hypre-intel-openmpi4-ohpc | 2.18.1 | Scalable algorithms for solving linear systems of equations.<br>http://www.llnl.gov/casc/hypre |
| mfem-gnu9-impi-ohpc<br>mfem-gnu9-mpich-ohpc<br>mfem-gnu9-mvapich2-ohpc<br>mfem-gnu9-openmpi4-ohpc<br>mfem-intel-impi-ohpc<br>mfem-intel-mpich-ohpc<br>mfem-intel-mvapich2-ohpc<br>mfem-intel-openmpi4-ohpc | 4.1 | Lightweight, general, scalable C++ library for finite element methods.<br>http://mfem.org |
| mumps-gnu9-impi-ohpc<br>mumps-gnu9-mpich-ohpc<br>mumps-gnu9-mvapich2-ohpc<br>mumps-gnu9-openmpi4-ohpc<br>mumps-intel-impi-ohpc<br>mumps-intel-mpich-ohpc<br>mumps-intel-mvapich2-ohpc<br>mumps-intel-openmpi4-ohpc | 5.2.1 | A MUltifrontal Massively Parallel Sparse direct Solver.<br>http://mumps.enseeiht.fr |
| opencoarrays-gnu9-impi-ohpc<br>opencoarrays-gnu9-mpich-ohpc<br>opencoarrays-gnu9-mvapich2-ohpc<br>opencoarrays-gnu9-openmpi4-ohpc | 2.8.0 | ABI to leverage the parallel programming features of the Fortran 2018 DIS.<br>http://www.opencoarrays.org |
| petsc-gnu9-impi-ohpc<br>petsc-gnu9-mpich-ohpc<br>petsc-gnu9-mvapich2-ohpc<br>petsc-gnu9-openmpi4-ohpc<br>petsc-intel-impi-ohpc<br>petsc-intel-mpich-ohpc<br>petsc-intel-mvapich2-ohpc<br>petsc-intel-openmpi4-ohpc | 3.13.1 | Portable Extensible Toolkit for Scientific Computation.<br>http://www.mcs.anl.gov/petsc |

Table 13 (cont): **Parallel Libraries**

| RPM Package Name | Version | Info/URL |
|---|---|---|
| ptscotch-gnu9-impi-ohpc<br>ptscotch-gnu9-mpich-ohpc<br>ptscotch-gnu9-mvapich2-ohpc<br>ptscotch-gnu9-openmpi4-ohpc<br>ptscotch-intel-impi-ohpc<br>ptscotch-intel-mpich-ohpc<br>ptscotch-intel-mvapich2-ohpc<br>ptscotch-intel-openmpi4-ohpc | 6.0.6 | Graph, mesh and hypergraph partitioning library using MPI.<br>http://www.labri.fr/perso/pelegrin/scotch |
| scalapack-gnu9-impi-ohpc<br>scalapack-gnu9-mpich-ohpc<br>scalapack-gnu9-mvapich2-ohpc<br>scalapack-gnu9-openmpi4-ohpc<br>scalapack-intel-impi-ohpc<br>scalapack-intel-mpich-ohpc<br>scalapack-intel-mvapich2-ohpc<br>scalapack-intel-openmpi4-ohpc | 2.1.0 | A subset of LAPACK routines redesigned for heterogenous computing.<br>http://www.netlib.org/lapack-dev |
| slepc-gnu9-impi-ohpc<br>slepc-gnu9-mpich-ohpc<br>slepc-gnu9-mvapich2-ohpc<br>slepc-gnu9-openmpi4-ohpc<br>slepc-intel-impi-ohpc<br>slepc-intel-mpich-ohpc<br>slepc-intel-mvapich2-ohpc<br>slepc-intel-openmpi4-ohpc | 3.13.2 | A library for solving large scale sparse eigenvalue problems.<br>http://slepc.upv.es |
| superlu_dist-gnu9-impi-ohpc<br>superlu_dist-gnu9-mpich-ohpc<br>superlu_dist-gnu9-mvapich2-ohpc<br>superlu_dist-gnu9-openmpi4-ohpc<br>superlu_dist-intel-impi-ohpc<br>superlu_dist-intel-mpich-ohpc<br>superlu_dist-intel-mvapich2-ohpc<br>superlu_dist-intel-openmpi4-ohpc | 6.1.1 | A general purpose library for the direct solution of linear equations.<br>http://crd-legacy.lbl.gov/~xiaoye/SuperLU |
| trilinos-gnu9-impi-ohpc<br>trilinos-gnu9-mpich-ohpc<br>trilinos-gnu9-mvapich2-ohpc<br>trilinos-gnu9-openmpi4-ohpc<br>trilinos-intel-impi-ohpc<br>trilinos-intel-mpich-ohpc<br>trilinos-intel-mvapich2-ohpc<br>trilinos-intel-openmpi4-ohpc | 13.0.0 | A collection of libraries of numerical algorithms.<br>https://trilinos.org |

## B Package Signatures

All of the RPMs provided via the OpenHPC repository are signed with a GPG signature. By default, the underlying package managers will verify these signatures during installation to ensure that packages have not been altered. The RPMs can also be manually verified and the public signing key fingerprint for the latest repository is shown below:

```
Fingerprint:  5392 744D 3C54 3ED5 7847 65E6 8A30 6019 DA565C6C
```

The following command can be used to verify an RPM once it has been downloaded locally by confirming if the package is signed, and if so, indicating which key was used to sign it. The example below highlights usage for a local copy of the `docs-ohpc` package and illustrates how the *key ID* matches the fingerprint shown above.

```
[sms]# rpm --checksig -v docs-ohpc-*.rpm
docs-ohpc-2.0.0-72.1.ohpc.2.0.x86_64.rpm:
    Header V3 RSA/SHA1 Signature, key ID da565c6c: OK
    Header SHA256 digest: OK
    Header SHA1 digest: OK
    Payload SHA256 digest: OK
    V3 RSA/SHA1 Signature, key ID da565c6c: OK
    MD5 digest: OK
```