

---

## Deploying Red Hat HPC Cluster in Al-Neelain University Labs

**Mohamed Mahagoub Abbas Mohamed**

Bachelor of Electronics & Communication Engineering, Mashreq University, Sudan  
mohamed\_abbas6@yahoo.com

**Noha Ahmed Ishaq Saleh**

Master of Computer Science, Computer Science, Al-Neelain University, Sudan

### Abstract

By using HPC Cluster, users can expedite their HPC workloads on elastic resources as needed and save money by choosing from low-cost pricing models that match utilization needs. The researchers preferred the red hat open-source solution because it is the easiest pre-integrated, pre-tested, and certified configurations. Most HPC software forgets about keys like storage and data integrity also I/O performance, So the researchers recommend using third party tools and monitors to reduce this weakness.

**Keywords:** HPC Cluster, Red Hat, Al-Neelain University

### Introduction

Indeed, HPC has moved from a selective and expensive endeavour to cost-effective technology within reach of virtually every budget.

HPC is a powerful technique, but it is not available at the corner office supply centre (yet) nor can it solve every product or process problem you have. It requires a dedicated effort and commitment to new and exciting technologies. Finding the right partners and technologies is critically important.

---

HPC will undoubtedly lead to future breakthroughs. Along with these successes, expect that there will be novel ways in which to derive a business advantage using HPC.

### **Research Problem**

Many scientists and engineers suffer from testing and modelling their application on expensive special purpose machines.

Application with high load and special needs on networking and computational units and special pre environment, cooling system, power and space also all these things let the cost increases dramatically with the development and the growth of applications.

### **Research Objective**

Design and deploy a complete solution can achieve the balance between application growth and taking care from the cost, count, easy of management, throughput, performance, space, power, cool system and another component.

### **Research Domain**

This research focus on the usage of this solution and the deployment ways the project and the persons who can gain benefit of this solution.

With an entry fee of at least seven figures, supercomputing was for the serious scientists and engineers who needed to crunch numbers as fast as possible.

In today's HPC world, it is not common for the supercomputer to use the same hardware found in Web servers and even desktop workstations. The HPC world is now open to almost everyone because the cost of entry is at an all-time low.

To many organizations, HPC considered an essential part of business success.

---

## 1- The Forms of HPC

There seem to be four modes in which you can obtain the cycles needed for typical HPC problems:

- The commodity HPC cluster: Built from standard off-the-shelf servers and high speed interconnects, a typical HPC system can deliver industry-leading, cost-effective performance.

A typical cluster can employ hundreds, thousands, and even tens of Thousands of servers all working together on a single problem (this is the high tech equivalent of a “divide and conquer” approach to solving large problems). Because of high performance and low cost, the commodity cluster is by far the most popular form of HPC computing. In addition, keep in mind the compatibility advantage — x86 commodity servers are ubiquitous.

- Dedicated supercomputer: In the past, the dedicated supercomputer was the only way to throw a large number of compute cycles at a problem. Supercomputers are still produced and use specialized non-commodity components. Depending on your needs, the supercomputer may be the best solution although it does not offer the commodity price advantage
- HPC cloud computing: This method is relatively new and employs the Internet as a basis for a cycles-as-a-service model of computing. The compute cycles in question live in the cloud somewhere allowing a user to request remote access to cycle’s on-demand. An HPC cloud provides dynamic and scalable resources (and possibly virtualization) to the end-user as a service. Although clouds can be cost effective and allow HPC to be purchased as an expense and not a capital asset, it also places some layers between the user and hardware that may reduce performance.

- Grid computing: Grid is similar to cloud computing, but requires more control by the end-user. Its main use is academic projects where local HPC clusters are connected and shared on a national and international level. Some computational grids span the globe while others are located within a single organization.

## 2 -Who Uses HPC Today?

The following is a list of major areas where HPC has a significant presence:

- Bio-sciences and the human genome.
- Computer aided engineering (CAE).
- Chemical engineering.
- Digital content creation (DCC) and distribution.
- Economics/financial.
- Electronic design and automation (EDA).
- Geosciences and geo-engineering.
- Mechanical design and drafting.
- Defence and energy.
- Government labs.
- University/academic.
- Weather forecasting.

The list could be longer as more and more areas are finding HPC useful as a tool to better understand their science, market, products, and customers.

As pioneers, the government and academic sectors have been successfully using and validating HPC methods for well over a decade.

---

## Proposed Approach

HPC systems are built from many components. There are some common elements, but clusters may employ a wide range of hardware and software solutions.

### Choosing Cluster Hardware

Obviously if you are looking into HPC, you are going to have to make sure you get hardware that can handle it! This section goes over what you need to make sure you have in place in order to set up an HPC system.

#### 1- Processors and Nodes

The processor is the workhorse of the cluster. And, keeping the workhorse busy is the key to good performance. Parallel programs are often distributed across many nodes of the cluster. However, multi-core has changed this situation a bit. Cluster nodes may now have 8 or even 16 cores per node (for example, the Sun Fire X4440 server with four Quad-Core AMD Opteron™ processors). It is Possible for whole HPC applications to fit on a single cluster node.

The choice of processor is very important because cluster installations often rely on scalable processor performance. Advances in the x86 architecture such as simultaneous 32/64 bit operation (as can be found in AMD64 technology), integrated memory controllers (as can be found in AMD Opteron processors), and technologies similar to AMD Hyper Transport™ technology have propelled commodity processors to the fore- front of HPC.

Depending on the design of the cluster, nodes can be fat (lots of cores, disk, and memory), thin (small number of cores and memory), or anything in between. Some applications work well in either type of node while others work best with a particular configuration.

---

Pay attention to the amount of memory. In general, “more cores per node” means more memory per node because each core could be asked to run a totally separate program. Many HPC cores require a large amount of memory. Check your application(s) memory requirements and size your nodes appropriately. This is something sometimes overlooked when upgrading from dual- to quad-core processors if you don’t increase the memory capacity, you effectively reduce the amount of memory per core, which could in turn limit some of the advantages from upgrading in the first place.

Another feature to consider for nodes is system management. The Intelligent Platform Management Interface (IPMI) specification defines a set of common interfaces to a computer system. System administrators can use IPMI to monitor system health and manage the system. Often IPMI features can be managed over a network interface, which is very important with large numbers of nodes often located in a remote data centre.

## 2- Co-processors

Back in the supercomputer days, there were products called array processors that were designed to do certain mathematic operations very quickly. Recently, there has been a resurgence of this concept with GP-GPU (General Purpose Graphical Processing Units) or as they are some- times called, video cards. Interestingly, this trend is not new in HPC, where it is not uncommon for practitioners to use new technologies in innovative ways to solve complex problems.

As with array processors, these devices can accelerate certain types of mathematical operations. In the case of GP-GPUs, however, dual-use as a commodity product has made this a relatively low-cost solution. One word of advice, take note of the precision, FLOPS reported for real applications, and

---

remember existing parallel applications must be re- programmed to use these types of devices. There are efforts to help address this problem.

### 3- The Communication: Interconnects

In order to keep all those nodes busy, a good interconnect (a port that attaches one device to another) is needed. As always, it all depends on your application set. In general, most high performance cluster systems use a dedicated and fast interconnect.

High performance interconnects are usually rated by latency, the fastest time in which a single byte can be sent (measured in nanoseconds or microseconds), and bandwidth, the maximum data rate (measured in Megabytes or Gigabytes per second).

The smaller the number, the more bandwidth (speed) that small packets will achieve. A final number to look at is the messaging rate. This tells you how many messages per second and interconnect can send and is important for multi-core nodes because many cores must share the interconnect.

Although the numbers listed are good for sizing up an interconnect, the ultimate test is your application(s). In most cases, your application will communicate using MPI (Message Passing Interface) libraries. This software is a communications layer on top of the hardware. MPI implementations vary and the true test is to run a few benchmarks.

In terms of available technology, high performance computing networks are generally being deployed using two major technologies; InfiniBand (IB) and 10 Gigabit Ethernet. Of course, if your applications don't require a large amount of node-to- node communication, standard Gigabit Ethernet (GigE) is a good

---

solution. GigE is often on the motherboard and there are very high density/performance GigE switches available.

Infini Band has been successfully deployed in both large and small clusters using blades and 1U servers. . Another key feature of Infini Band is the availability of large multi-port switches like the Sun Data Centre Switch 3456, which provides low latency. A large port density allows for less cabling and fewer sub-switches when creating large clusters.

#### 4- The Storage

Storage is often the forgotten in the HPC cluster. Almost all clusters require some form of high performance storage. The simplest method is to use the head node as an NFS server. Even this simple solution requires that the head node have some kind of RAID sub-system. Often, NFS doesn't scale to large numbers of nodes. For this reason, there are alternative storage designs available for clusters. Most of these designs are based defines the type of storage hardware that will be required.

HPC applications notoriously create large amounts of data, and ensuring that an archiving system is available is crucial to many data centers. Moving data to tried-and-true backup technologies (magnetic tape) is an important step in protecting your HPC investment. A good archiving system will automatically move data from one storage device to another based upon policies set by the user.

Flash modules can be integrated directly onto motherboards, as is the case with the Sun Blade X6240, or integrated using the PCIe bus. Keep in mind that a poor storage sub-system can slow cluster down as much, if not more, than a slow interconnect. Underestimating your storage needs and assuming NFS “will just handle it” may be a mistake.



---

A final aspect of the storage hardware is making sure it will work and integrate with the other cluster hardware. Sometimes storage sub-systems are purchased from separate vendors, causing incompatibilities that can often be traced to software driver issues or even physical incompatibilities. Many users avoid these issues by utilizing a top-tier vendor with well-designed and tested sub-systems for both their compute and storage needs. The key here is to manage the data throughout its entire life cycle, from the cluster to home directories to bulk storage to tape.

### **5- Racking**

Compute nodes can take the form of 1U servers or blade systems the choice is largely one of convenience. Blade systems often have a slightly higher acquisition cost but can offer much more manageability, better density, and some power and cooling redundancy. Regardless of which form of computer node you choose, you should ensure that all cluster equipment is “rack mountable” and can be placed in standard equipment racks. . It is very useful to map out your rack chassis space for all equipment and allow some extra for future enhancements.

### **6- Power and Cooling**

Power and cooling has gone from an overlooked expense to a critical factor in cluster procurements. A general rule of thumb for forecasting power and cooling cost is that the yearly cost to keep a cluster powered and cooled will equal roughly one third of the purchase price of the cluster itself. For this reason, looking for a green solution makes both environmental and economic sense. For instance, choosing a cluster that uses 45nm Quad-Core AMD Opteron™ HE processors, each with 55-watt ACP (Average CPU Power) rating can be a smart move. Factoring in up to 20 percent, lower idle power compared to similarly configure competing systems may just make it a brilliant move.

---

## Cluster Software

To get your HPC system up and running, you need something to run on it. That is where software comes in. Linux is by far the most common operating system that HPC users choose and any other software needs to work well with Linux. Other options include Solaris, which is “open” and has full support for Linux binary compatibility, and Microsoft HPC Server.

### 1- Operating Systems

Linux represents a plug-and-play alternative and does not add any licensing fees for the compute nodes (which can be quite large in number).

In addition to the Linux kernel, much of the important supporting software has been developed as part of the GNU project.

The GNU/Linux core software is open-source and can be freely copied and used by anyone. There are, however, requirements to ensure source code is shared.

The openness and “share ability” of GNU/Linux has made it an ideal HPC operating system. It has allowed HPC developers to create applications, build drivers, and make changes that would normally not be possible with closed source.

### 2- Cluster Software

There are several types of software tasks that are needed to run a successful cluster. These tasks include administration, programming, debugging, job scheduling, and provisioning of nodes.

From a user’s perspective, programming is perhaps the most important aspect of the cluster.

The most important HPC tool for programming is probably MPI (Message Passing Interface). MPI allows programs to talk to one another over cluster networks (this allows individual nodes to coordinate their participation in the overarching task). Without this software, creating parallel programs would be, and was in the past, a very custom (and likely time-consuming) process. Today, there are both open and commercial MPI versions. The two most popular open MPIs are MPICH2 from Argonne Lab and the Open MPI project.

In addition to MPI, programmers need compilers, debuggers, and profilers. The GNU software includes very good compilers and other programming tools; however, many users prefer to use professional compiler/debugger/profiler packages such as those offered by Sun Microsystems (Sun Studio 12 for Linux), the Portland Group (PGI), and Intel.

### 3- File Systems

Almost all clusters use the standard NFS file system to share information across nodes. This is a good solution; however, NFS was not designed for parallel file access (for instance, multiple processes reading and writing to the same file). This limitation had become a bottleneck for HPC systems. For this reason, parallel file systems were developed.

One of the areas where the open GNU/Linux approach has served the HPC community is with file systems. There are multitudes of choices, all of which depend on your application demands. HPC file systems are often called “parallel file systems” because they allow for aggregate (multi-node) input and output. Instead of centralizing all storage on a single device, parallel file systems spread out the load across multiple separate storage devices. Parallel file systems are very often “designed” because they must be matched to a particular cluster.

---

One popular and freely available parallel file system is Luster from Sun Microsystems. Luster is a vetted, high-performance parallel file system. Other options include PVFS2, which is designed to work with MPI.

Cluster file systems cover a large area. In addition to massive amounts of input, scratch, and checkpoint data, most HPC applications produce large amounts of output data that are later visualized on specialized systems.

#### 4- HPC Resource Schedulers

Clusters usually have lots of cores and lots of users. Sharing all these resources is no trivial matter. Fortunately, the allocation of cores is done by scheduling software and not users (thus avoiding the ensuing chaos that could occur). Depending on the application, a scheduler may closely pack the cores (for instance, keeping them all together on a small number of nodes) or distribute them randomly across the cluster.

A cluster schedule works as follows. Like the old mainframe days, all users must submit their jobs to a work queue. As part of the submission processes, the user must specify the resources for the job (for instance, how many cores, how much memory, how much time, and so on). The resource scheduler then determines, based on site-wide policies, whose job gets to run next. Of course, it depends on when the resources become available and thus, as users often find out, being first in line doesn't necessarily mean being first to execute.

The resource scheduler is a critical part of the cluster because it would be almost impossible to share resources without some form of load balancing tool. One important feature of the scheduling layer enables administrators to take resource nodes "off-line" for repair or upgrade and users are none the wiser users rarely have a say in which nodes they're assigned. Additionally, if a node fails, the running jobs that use that node may fail, but other nodes keep working and the

---

scheduler will work around the failed node. There are several popular and freely available resource schedulers. One popular choice is Sun Grid Engine from Sun Microsystems. Others include Torque, Lava, and Maui. In the commercial sector there are fully supported versions of Sun Grid Engine, Moab, Univa UD Uni Cluster, and Platform LSF.

### **5- Ready to Run Application Software**

There are plenty of “cluster aware” software packages in both commercial and open source form. Many of the commercial packages may require specific GNU/Linux distributions, but open source packages can be built within your specific environment. Commercial packages may or may not have better performance or feature sets; however, they all have professional support options that are not always available with open source packages. Due diligence will pay off in this area.

### **6- Provisioning: Creating the Cluster**

Turning raw hardware into a functioning cluster is not as difficult as it was in the past. This process is called provisioning and it involves installing and configuring the head node and worker nodes of the cluster. There are two main methods that are used:

- Local: This type of provision involves placing a full operating system image on each node. Obviously, this method requires that each node have a hard disk drive (or other persistent local storage). Care must be taken to keep the node images synchronized (which consumes valuable time and resources).

The advantages to this method are that the nodes can be started independently and they don't require a central server; there can also be advantages in quicker

boot times and reduced peak network traffic as a large number of nodes are brought on-line.

- Remote: This method configures each node across the network when it boots using standard protocols such as DHCP (Dynamic Host Configuration Protocol) and TFTP (Trivial File Transfer Protocol). These methods often are called diskless because they don't require a hard drive on each node; however, they don't preclude using node hard drives for local and temporary storage. The advantage of remote provisioning is that the software environment for each node is centralized and a single change can be easily propagated throughout the cluster.

In addition, in cases where nodes don't have hard drives, this is the only way to bring up the nodes.

## 6- Cluster Tool Kits

Most GNU/Linux distributions lack many of the key cluster software packages. To remedy this situation, there are now cluster distributions that provide turn-key provisioning and ready-to-run software. Many of these are also freely available and based on some of the above mentioned distributions. The freely available options include

- Sun HPC Software, Linux Edition: [www.sun.com/software/products/hpcsoftware](http://www.sun.com/software/products/hpcsoftware)
- Rocks Clusters: [www.rocksclusters.org](http://www.rocksclusters.org)
- Oscar (Open Source Clusters Application Resources)

<http://oscar.openclustergroup.org>

---

There are commercial versions as well. These are based on GNU/Linux and often provide extra commercial applications and support as part of the cluster package:

- Scyld Clusterware
- ClusterCorp Rocks+
- Platform OCS
- Red Hat HPC Solution

## **Applying the Proposed Approach**

### **Installation Prerequisites**

Installing Red Hat HPC Solution (Red Hat HPC) requires one system to be designated as an installer node. This installer node is responsible for installing the rest of the nodes in the cluster.

Prior to installing Red Hat HPC, confirm that the designated machine has Red Hat Enterprise Linux 5.3 installed and meets the following requirements:

- Root partition with at least 40 GBytes free.
- Disable SELinux.
- One or more network interfaces use a statically defined IP addresses. Connect these to the networks where the machines are provisioned.
- Red Hat Enterprise Linux Version 5.3 installation media
- A valid subscription to Red Hat Network is required including an entitlement to Red Hat HPC Channel

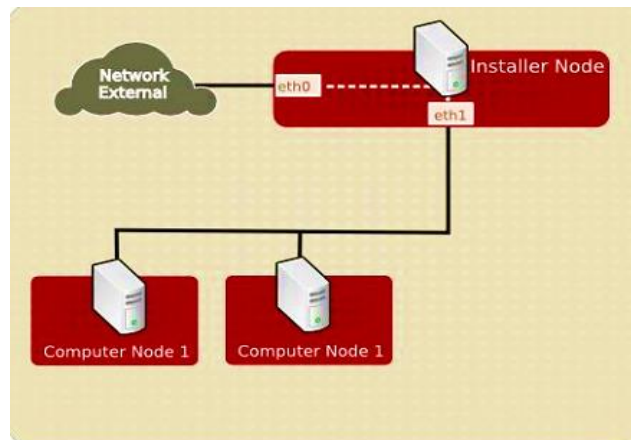
- The firewall (iptables) must be configured to permit the services needed for installation on all networks used to provision nodes (HTTP, HTTPS, TFTP, DNS, NTP, BOOTPS, etc).
- A script is provided to appropriately configure the firewall.
- Red Hat HPC creates a private DNS zone for all machines under its control. The name of this zone must NOT be the same as any other DNS zone within the organization where the cluster is installed

### Installation Procedure

- Verify that the installer node meets the prerequisites.
- Register on Red Hat Network and subscribe to the appropriate channels.

### Recommended Network Topology

In its default configuration, the Red Hat HPC Solution treats one Network interface of the installer node as a public interface on which it imposes a standard firewall policy, while other interfaces are treated as trusted, private interfaces to the cluster nodes.





---

## Starting the Install

Log into the machine as root and install the Red Hat HPC bootstrap RPM:

```
# yum install ocs mod_ssl
```

After installing the ocs RPM, source the OCS environment

```
# source /etc/profile.d/kusuenv.sh
```

Run the installation script:

```
# /opt/kusu/sbin/ocs-setup
```

The script detects your network settings and provide a summary per NIC:

```
NIC: eth0
```

```
=====
Device = eth0           IP = 172.25.243.44
Network = 172.25.243.0  Subnet = 255.255.255.0
mac = 00:0C:29:C4:61:06 Gateway = 172.25.243.2
dhcp = False           boot = 1
```

Red Hat HPC creates a separate DNS zone for the nodes it installs. The tool prompts for this zone. The Red Hat HPC Solution stores a copy of the OS media and installation images. The OCS installer prompts for the location of the directory to store the operating system. The default is /depot. A

Symbolic link to /depot is created if another location is used.

The OCS installer builds a local repository using the OS media. This repository is by OCS when provisioning compute nodes.

The OCS installer asks for the physical DVD or CDs (in the optical drive physically connected to the installer host), a directory containing the contents of the OS media, or an ISO file providing the media.

---

After the OS media is successfully imported (approximately 5-10 minutes when importing from a physical optical drive) and the local OCS repository created, a sequence of scripts runs to configure the OCS cluster for the installation.

The default firewall rules for a RHEL installation blocks the ports needed to provision nodes. The script provided configures the firewall to allow these ports. When the script runs, it opens the ports necessary for provisioning the nodes. It also configures Network Address Translation (NAT) on the installer node, so that the provisioned nodes can access the non-provisioning networks connected to the installer on other interfaces.

To run the script to configure the firewall as root, run:

```
# /opt/ kusu/ bin/ kusurc/ opt/kusu/ etc/ S02KusuIptables.rc.py
```

Once the installation has completed the following message appears:

Congratulations! The base kit is installed and configured to provision on:

Network 1.2.3.4 on interface ethX

The installer node is ready to begin installing other nodes in the cluster.

Prior to installing the compute nodes, it is best to add all the desired kits, and customize the node groups. If the kits are added after the Compute Nodes have been installed it is necessary to run the following command to get Nagios® and Cacti® to display the nodes in their respective web interfaces:

```
# addhost -u
```

This causes re-generation of many of the application configuration files.

---

## Updating an Existing Installation

Updating an existing Red Hat HPC cluster is a two-step process. The installer node contains a Red Hat repository for RHEL 5.

This repository must first be updated prior to updating the kits or running a yum.

Update on the master installer. If the master installer contains packages that are newer than the packages in the Kusu repository, there can be dependency problems when installing some kits.

Once the repository is updated, the kits on the installer can be updated. Before the base kit can be updated, the existing addon kits in the RHHPC system must be removed. This is required, as some of the older kits are not guaranteed to be compatible with RHHPC 5.3. The updated versions of the addon.

Kits for RHHPC 5.3 must be installed.

To update kits follow the instructions below.

### 1. Removing Incompatible Kits Prior to Updatin \_the Base Kit

1. Remove the kit components from the nodegroup. Run ngedit and select the installer node group to edit. Go to the component screen. De-select the components of the kits you wish to upgrade.

Continue and apply the changes.

2. Run the above step for all nodegroups.
3. Remove the kit associations from the repository.

```
#repoman -e -kkitname -rreponame
```

Optionally, to list repositories and associated kits, the following command can be used:

```
#repoman -l
```

4. Update repository after removing kit associations:

```
#repoman -u -rreponame
```

5. Remove older kits from the system.

```
#kitops -e -kkitname
```

Optionally use this command to list Installed kits.

```
#kitops -l
```

The base kit must be updated prior to reinstalling the other kits. The steps below outline how to update the base kit on the installer.

## 2. Updating the Base Kit Prior to Other Kits

- Ensure that the installer node can connect to Red Hat Network (RHN).
- Update the ocs package

```
# yum update ocs
```

- Source the environment:

```
# source /etc/profile.d/kusuenv.sh
```

- Run the OCS upgrade script. This will update the base kit from RHN, and rebuild the repository for installing nodes.

```
# ocs-setup -u
```

The base kit is now updated. If desired the other kits can be updated. Use the following procedure to update the kits:

---

### 3. Updating Other Kits

1. Update the kit downloader's by running the following command for the downloader you wish to Upgrade

```
#yum update ocs-kit-kitname
```

### 4. Updating the Installer Node and the Compute Node Repository

Red Hat HPC manages updates to the installer nodes differently from all other nodes in the cluster. The RPM packages and updates to the Operating System Repository for all nodes provisioned by the installer (and that includes compute nodes and diskless nodes) are managed independently from updating the installer node.

To update the installed packages on the installer node, use the following command:

```
# yum update
```

The yum tool downloads all of the required updates for the operating system and installs them on the installer node. Since updating installer nodes and compute nodes is separate you can choose to update the installer node – and either choose to update the compute nodes or not update the compute nodes.

Prior to updating the repository it is recommended that a snapshot (copy) of the repository be made. If there are any application issues with the updates the copy can be used:

```
# repoman -r rhel5_x86_64 -s
```

To update the compute nodes in a Red Hat HPC cluster use the following command:

```
# repopatch -r rhel5_x86_64
```

The repopatch tool downloads all of the required updates for the operating system and installs them into the repository for the compute nodes. repopatch displays an error if it is not properly configured.

For example.

```
# repopatch -r rhel5_x86_64
Getting updates for rhel-5-x86_64. This may take awhile...
Unable to get updates. Reason: Please configure
/opt/kusu/etc/updates.conf
```

Edit the /opt/kusu/etc/updates.conf file adding your username and password for Red Hat. Network to the [rhel] section of the file, for example:

```
[fedora]
url=http://download.fedora.redhat.com/pub/fedora/linux/

[rhel]
username=
password=
url=https://rhn.redhat.com/XMLRPC
yumrhn=https://rhn.redhat.com/rpc/api
```

After configuring the /opt/kusu/etc/updates.conf file, repopatch downloads all of the updates from Red Hat Network and creates an update kit which is then associated with the rhel-5-x86\_64 repository using ngedit.

repopatch automatically associates the update kit with the correct repository. View the list of update kit components from ngedit on the Components screen and list the available update kits with the kitops command for example



Once repopatch has retrieved the updated packages and rebuilt the repository, the compute nodes can be updated.

This is done by either reinstalling them using:

```
# boothost -r -n {Name of Node group}
```

Or without reinstalling by using:

```
# cfmsync -u -n {Name of Node group}
```

The cfm sync command causes the compute nodes to start updating packages from the repository they installed from.

## 5. Installing Additional Red Hat HPC Kits

Additional software tools such as Nagios® and Cacti are packaged as software kits. Software packaged as a kit is easier to install onto a Red Hat HPC Cluster. A kit contains rpms for the software, rpms for meta-data and configuration files.

---

To install Cacti® onto the Red Hat HPC cluster use the following commands:

```
# yum install ocs-kit-cacti  
# /opt/kusu/sbin/install-kit-cacti
```

To install Nagios® onto the Red Hat HPC cluster use the following commands:

```
# yum install ocs-kit-nagios  
# /opt/kusu/sbin/install-kit-nagios
```

To see what kits are available use:

```
# yum search ocs-kit
```

The yum commands above download the respective kit downloaders from Red Hat Network. The kit downloaders are distinguished by the ocs- kit-\* prefix. In the case of a download problem the kit downloaders can be safely re-run.

Included in the kit downloader RPM is an installation script that adds the kit to the Red Hat HPC cluster repository and rebuilds the cluster repository.

Every kit that is downloaded from Red Hat Network has a corresponding script used to install the kit into the cluster repository.

Verifying the Red Hat HPC install

Once the installer node is successfully configured the next step is to verify that all software components are installed and working correctly.

The following steps can be used to verify the Red Hat HPC Install.

## 6. Viewing Available Red Hat HPC Kits

Use the following command to query the kits available from Red Hat Network:

```
# yum list ocs-kit-\*
```



The following kits are available:

Name	Description
ocs-kit-cacti	A reporting tool
ocs-kit-lava	Open source LSF, a batch scheduling and queuing system
ocs-kit-nagios	A network monitoring tool
ocs-kit-ntop	A network monitoring tool
ocs-kit-rhel-java	The Java Runtime
ocs-kit-hpc	A collection of MPIS (MPICH 1.2, MVAPICH 1.2 and OpenMPI), math libraries (ATLAS, BLACS, SCALAPACK), and benchmarking tools.
ocs-kit-ganglia	Another system monitoring tool
ocs-kit-rhel-xfed	The XFED stack

## 7. Verifying the HPC Install

1. Start the web browser (Firefox). The cluster homepage is displayed.
2. Use the `dm esg` command to check for hardware issues.
3. Check all network interfaces to see if they are configured and up.

```
# ifconfig -a | more
```

4. Verify the routing table is correct.

```
# route
```

Make sure the following system services are running:

Service	Command
Web Server	<code>service httpd status</code>
DHCP	<code>service dhcpd status</code>
DNS	<code>service named status</code>
Xinetd	<code>service xinetd status</code>
MySQL	<code>service mysqld status</code>
NFS	<code>service nfs status</code>

---

5. Run some basic Red Hat HPC commands.

List the installed repositories

```
# repoman -l
```

List the installed kits

```
# kitops -l
```

Run the Node Group Editor

```
# ngedit
```

Run the Add Host tool

```
# addhost
```

Adding Nodes to the Cluster

The addhost tool adds nodes to a Red Hat HPC cluster.

addhost listens on a network interface for nodes that are PXE booting and adds them to a specified node group.

Node groups are templates that define common characteristics such as network, partitioning, operating system and kits for all nodes in a node group.

Open a terminal window or login to the installer node as root to add nodes.

## 8. Adding Nodes to the Cluster

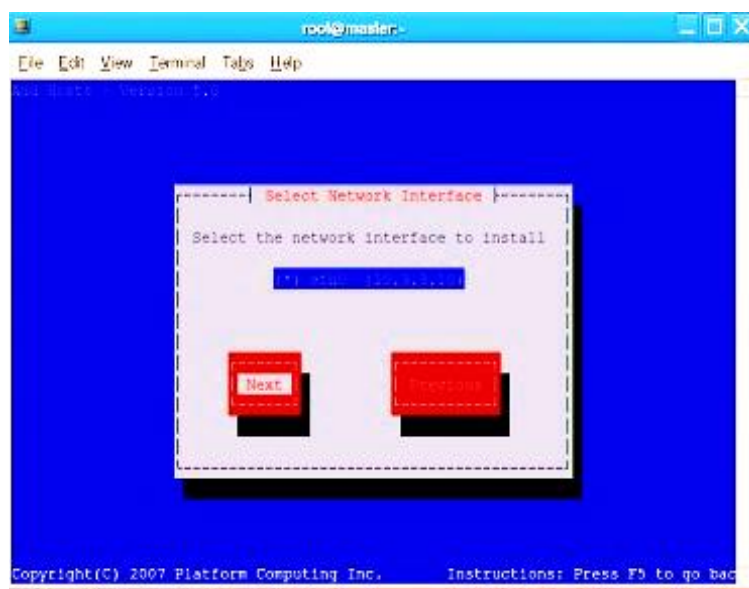
1. Run addhost

```
# addhost
```

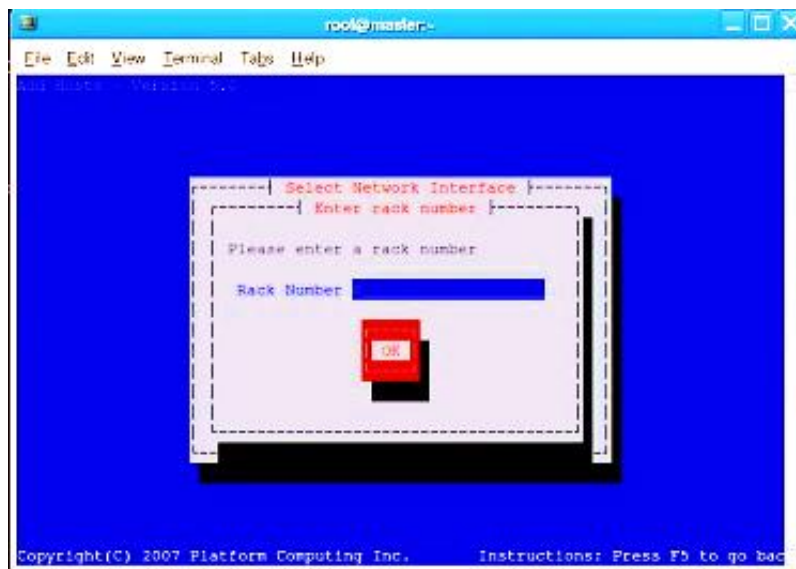
2. Select the node group for the new nodes. Normally compute nodes are added to the computerhel node group:



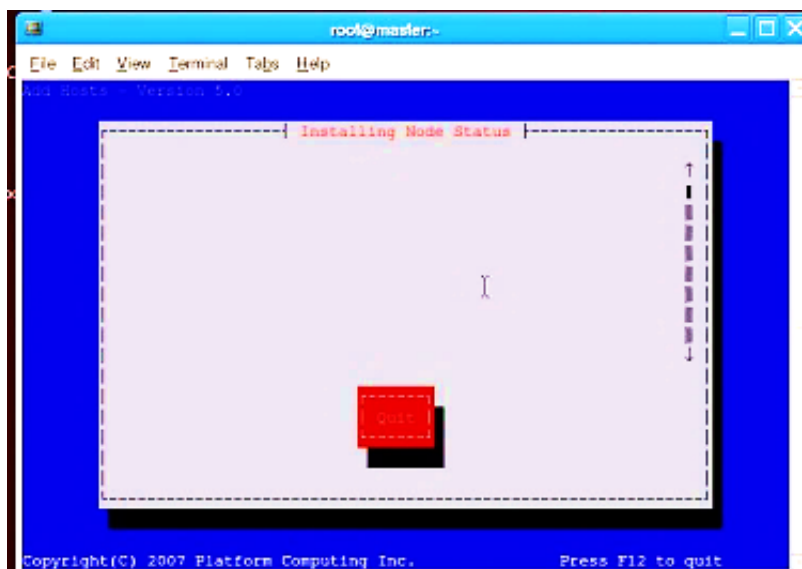
3. Select the network interface to listen on for new PXE booted node



4. Indicate the rack number where the nodes are located



5. addhost waits for the nodes to boot



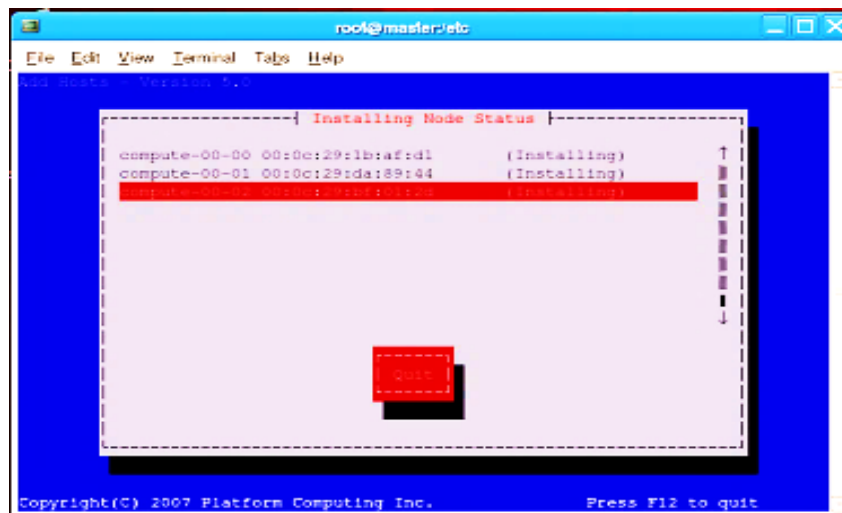
6. Boot the nodes you want to add to the cluster. Wait a few seconds between powering up nodes so that the machines are named sequentially in the order they are started.

```
Network boot from Intel E1000
Copyright (C) 2003-2005 VMware, Inc.
Copyright (C) 1997-2000 Intel Corporation

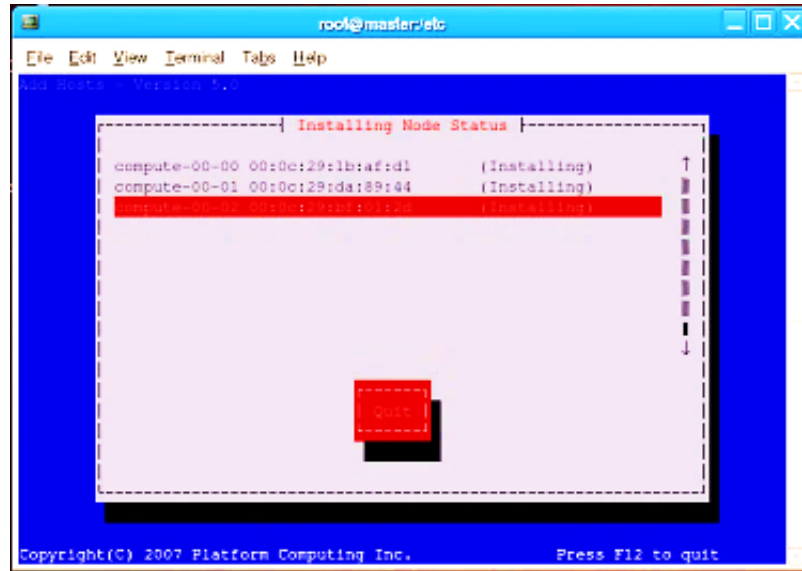
CLIENT MAC ADDR: 00 0C 29 1B AF D1 GUID: 564069CB-A5FB-7C2B-E3EB-4050E610AFD1
CLIENT IP: 10.3.3.1 MASK: 255.255.255.0 DHCP IP: 10.3.3.10
GATEWAY IP: 10.3.3.10

PXELINUX 3.11 2005-09-02 Copyright (C) 1994-2005 H. Peter Anvin
JNDI data segment at: 000990F0
JNDI data segment size: 4068
JNDI code segment at: 0009E950
JNDI code segment size: 0BB0
PXE entry point found (we hope) at 9E95:0106
My IP address seems to be 0A030301 10.3.3.1
ip=10.3.3.1:10.3.3.10:10.3.3.10:255.255.255.0
TFTP prefix:
Trying to load: pxelinux.cfg/01-00-0c-29-1b-af-d1
Loading kernel-rhel-5-x86_64.....
Loading initrd-rhel-5-x86_64.img.....
Ready.
```

7. When a node is successfully detected by addhost a line appears in the installing node status window.



8. Exit addhost when Red Hat HPC has detected all nodes. The Installing node status screen does not update to indicate that the node has installed.



## 9. Managing Node Groups

Red Hat HPC cluster management is built around the concept of node groups. Node groups are a powerful template mechanism that allows the cluster administrator to define common shared characteristics among a group of nodes.

A node group for compute nodes makes it easy to configure and manage 1 or 100 nodes all from the same node group.

The `ngedit` command is a graphical TUI (Text User Interface) run by the cluster administrator to create, delete and modify node groups.

The `ngedit` tool modifies cluster information in the Red Hat HPC database and also automatically calls other tools and plugins to perform actions or update configuration.

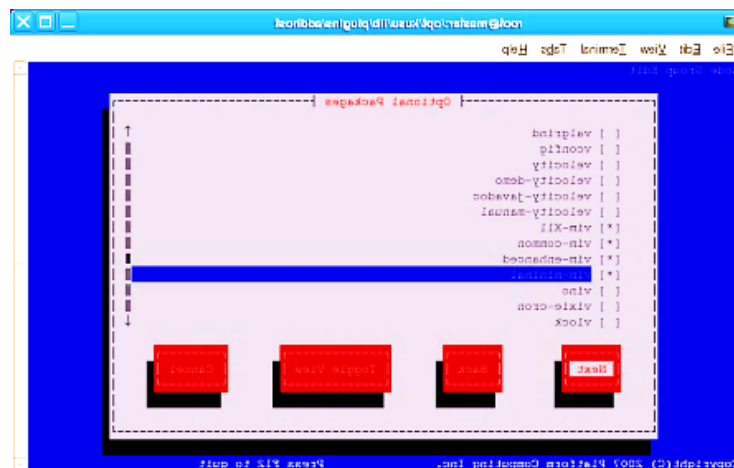
The Red Hat HPC database keeps track of the node group state, thus several changes can be made to a node group simultaneously and the physical nodes in the group can be updated immediately or at a future time using the `cfm sync` command.

### Adding RPM Packages in RHEL to Node Groups

Open a Terminal and run the node group editor as root.

```
# ngedit
```

Select the `compute-rhel` node group and move through the Text User Interface screens by pressing `F8` or by choosing `next` on the screen. Stop at the Optional Packages screen.



Additional RPM packages are added by selecting the package in the tree list. Pressing the space bar expands or contracts the list to display the available packages.

- **Adding RPM Packages Not in RHEL to Node Groups**

Red Hat HPC maintains a repository containing all of the RPM packages that ship with Red Hat Enterprise Linux. This repository is sufficient for most customers. RPM packages that are not in Red Hat Enterprise Linux can also be added to a Red

---

Hat HPC repository by placing the RPM packages into the appropriate contrib directory under /depot. For example:

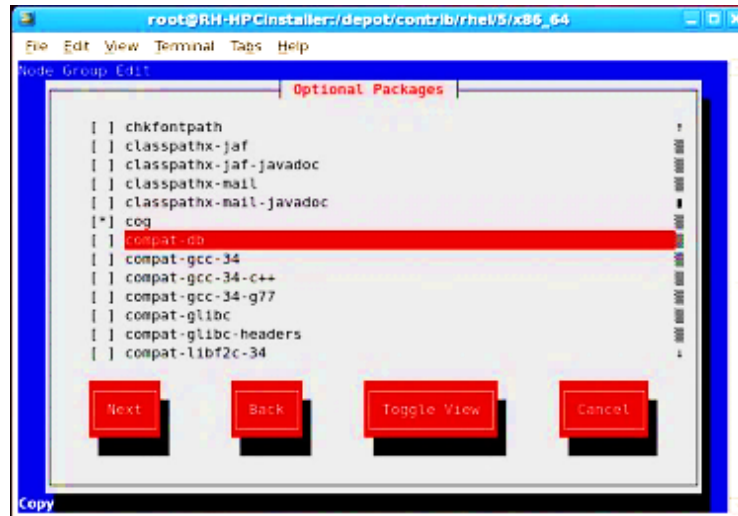
1. Start with the RPMs that are not in Red Hat Enterprise Linux or in a Red Hat HPC Kit
2. Create the appropriate subdirectories in /depot/contrib:  

```
# mkdir -p /depot/contrib/rhel/5/x86_64  
# cp foo.rpm /depot/contrib/rhel/5/x86_64/foo.rpm
```
3. Rebuild the Red Hat HPC repository with repom an:  

```
# repoman -u -r rhel5_x86_64
```
4. It takes some time to rebuild the repository and associated images.
5. Run ngedit and navigate to the Optional Packages screen.
6. Select the new package by navigating within the package tree and using the spacebar to select.
7. Continue through the ngedit screens and either allow ngedit to synchronize the nodes immediately or perform the node synchronization manually with cfm sync -p at a later time.

Example: selecting a RPM package that is not included in Red Hat Enterprise





Linux Contributions can be added to more than one Red Hat HPC repository, the directory structure is:

`/depot/ contrib/<os_name>/<version>/<architecture>`

### • Adding Kit Components to Node Groups

Adding kit components to nodes in a node group is very similar to adding additional RPM packages.

1. Open a Terminal and run `ngedit`.
2. Press F8 (or choose Next) and proceed to Components screen.
3. Enable components on a per-node group basis.

Each Red Hat HPC kit installs an application or a set of applications. The kit also contains components, which are meta-RPM packages designed for installing and configuring applications within the cluster.

By enabling the appropriate components, it is easy to configure all nodes in a node group.

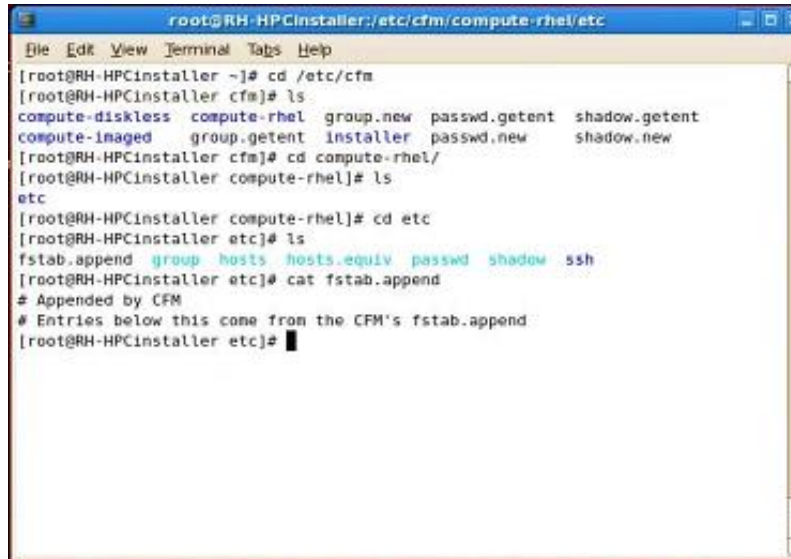


## 10. Synchronizing Files in the Cluster

HPC clusters are built from individual compute nodes and all of these nodes must have copies of common system files such as /etc/passwd/ etc/ shadow/ etc/ group and others.

Red Hat HPC contains a file synchronization service called CFM (Configuration File Manager).

CFM runs on each compute node in the cluster and when new files are available on the installer node a message is sent to all of the nodes notifying them that files are available. Each compute node connects to the installer node and copies the new files using the HTTP protocol. All files to be synchronized by CFM are located in the directory tree /etc/cfm /<node group>



```
root@RH-HPCInstaller:/etc/cfm/compute-rhel/etc
Die Edit View Terminal Tabs Help
[root@RH-HPCInstaller ~]# cd /etc/cfm
[root@RH-HPCInstaller cfm]# ls
compute-diskless  compute-rhel  group.new  passwd.getent  shadow.getent
compute-imagcd   group.getent  installer  passwd.new     shadow.new
[root@RH-HPCInstaller cfm]# cd compute-rhel/
[root@RH-HPCInstaller compute-rhel]# ls
etc
[root@RH-HPCInstaller compute-rhel]# cd etc
[root@RH-HPCInstaller etc]# ls
fstab.append  group  hosts  hosts.equiv  passwd  shadow  ssh
[root@RH-HPCInstaller etc]# cat fstab.append
# Appended by CFM
# Entries below this come from the CFM's fstab.append
[root@RH-HPCInstaller etc]#
```

In the screenshot above /etc/cfm directory contains several node group directories such as compute-diskless and compute-rhel. In each of those directories is a directory tree where the /etc/cfm /<node group> directory represents the root of the tree. The /etc/cfm /compute-rhel/etc directory contains several files or symbolic links to system files.

Creating symbolic links for the files in CFM allows the compute nodes to be automatically synchronized with system files on the installer node. /etc/passwd and /etc/shadow are two examples where symlinks are used.

Adding files to cfm is simple. Create all of the directories and subdirectories for the file then place the file in the appropriate location.

Existing files can also have a <filename>.append file. The contents of a <filename>.append file are automatically appended to the existing <filename> file on all nodes in the node group.

Use the cfm sync command to notify all of the nodes in all node groups or nodes in a single node group. For example:

---

```
# cfmsync -f -n compute-rhel
```

Synchronizes all files in the compute-rhel node group.

```
# cfmsync -f
```

Synchronizes all files in all node groups

### Known Issues

Summary: ocs-setup -u fails to update the system, with the message OCS setup script does not seem to have run in this machine, cannot upgrade.

Details: RHHPC 5.1 used to utilize a lockfile mechanism to control if the system had been installed.

This has been moved to the database where the state is stored. However, when upgrading from OCS 5.1, this file still needs to be tested. If this file is removed, then ocs-setup -u will not correctly trigger.

Work around: Run the following command before re-running ocs-setup -u:

```
#touch /var/lock/subsys/ocs-setup.
```

Summary: After updating the system then removing and installing the updated cacti kit, the graphs do not display properly.

Details: The cacti user's home directory was not created properly with RHHPC 5.2's cacti kit. This has a knock on effect when updating the cacti kit because the rpms do not recreate the user, if the user already exists.

Work around: Run the following command prior to running the updated install-kit-cacti kit installer.

script:

```
# userdel cacti
```

---

---

Summary: The ganglia user can sometimes not be created when installing ganglia, causing the services to fail.

Details: An interaction with the other addon kits can sometimes cause the ganglia user to be not created.

Symptoms: Running gmond and gmetad fail, user ganglia does not exist.

Workaround: Run the following commands to create the ganglia user and permission the directories correctly:

```
#useradd -d/ var/ lib/ ganglia -s/ sbin/ nologin ganglia
```

```
#cd/ var/ lib/ ganglia/
```

```
#chown ganglia: ganglia rrds
```

```
#service gmond restart
```

```
#service gmetad restart
```

## Results

1. Cluster design should be influenced strongly by Applications that will be used on cluster.
2. With the right tools, a cluster installation will not require much more time and effort than installing and Fine-tuning a desktop OS on a notebook PC.
3. A complete and easy to use HPC user environment allows users to get their work done quickly and smoothly.
4. Good cluster management environment is a key for keeping cluster operational over time.

5. For IT departments that rely on paid employees or consultants to deploy and manage HPC clusters, the Red Hat HPC Solution dramatically reduces the level of effort required to install the cluster and configure all of the layered services.
6. Most organizations depreciate their capital assets over a fixed period of time. For example, a department with a three-year capital depreciation policy that takes six months to get a cluster operational realizes a reduction of 17% in terms of financial return on assets and this is before they account for the cost of people's time or periods of avoidable outages. The Red Hat HPC Solution maximizes financial return on assets by ensuring that clusters are immediately usable.
7. Some organizations may account for system and network monitoring and management costs outside of the context of the HPC deployment itself. These costs are real, however, and the tools included in the Red Hat HPC Solution often come at an additional cost with other cluster management solutions.
8. By relying on pre-integrated, pre-tested, and certified configurations fully backed by both Red Hat and Platform Computing, customers are assured that in the event of problems, issues can be addressed quickly and efficiently without the need for additional on-site consultants or support expertise.

## Future Work

1. Incomplete attention to end-to-end data integrity.

While no HPC solution exists today providing end-to-end (client to system and back to client) data integrity, the community contributes and supports the T10PI standard. The standard is currently being adopted for components used in the HPC environment.

2. Storage system software is not resilient enough.

---

An otherwise normal hardware error can cause a storage system outage due to current software limitations. Today this impacts the choice of hardware and the amount of facilities infrastructure required to sustain reliable storage systems.

3. Provide monitoring and diagnostic tools that allow users to understand file system Configuration and performance characteristics and troubleshoot poor I/O performance.

Today there exist several tools for monitoring file system performance:

Darshan, IPM, and LMT.

However, these do not have broad deployment across HPC sites and need further support for code maintenance and feature improvement to be more widely accepted.

4. Communication between storage systems users and storage system experts needs improvement.

There are few storage experts at HPC sites, and it is rare to identify individuals on user projects who are responsible for focusing on storage or I/O. Further, there are rarely consultants at HPC centers who focus on or have the skill set to manage storage or I/O issues. As storage and I/O increasingly become focal issues in HPC projects, this will improve. Sites are working on identifying efficient and effective user education mechanisms (e.g., online videos, onsite workshops).

5. Need tools and mechanisms to capture provenance and provide lifecycle management for data.
6. Ensure storage systems are prepared to support data-intensive computing and new workflows.

Storage systems are designed for computational system needs today. However, new instruments such as genomic sequencers and next generation light sources have tremendous bandwidth and capacity requirements alone that will strain existing systems. As well, the push towards data intensive computing will result in different system architectures than exist at most HPC facilities today.

7. Measure and track trends in storage system usage to the same degree we measure and track flops and cycles on computational systems.

This will improve understanding of the capabilities and limitations of the system in use, and will help with improving quality of service for users in a shared storage environment.

8. Tools to provide scalable performance in interacting with files in the storage system. This is focused on tools users require to interact with files (cp, tar, gzip, grep, etc.)

## References

1. Charles Severance, Kevin Dowd, High Performance Computing (RISC Architectures, Optimization & Benchmarks), Publisher O'Reilly Media Edition Second Publication Date July 12, 1998.
2. Mark Black, Kailash Sethuraman, Daniel Riek, Red\_Hat\_HPC\_Solution-5.3-Installation\_Guide-en-US, Copyright © 2008 Red Hat and Copyright © 2008 Platform Computing Inc.
3. Wiley Publishing, High Performance Computing For Dummies®, Published by Inc. by Douglas Eadline 2007.
4. AMD, 4TH GEN AMD EPYC PROCESSOR, Fourth Edition, May 2024.
5. ARCHITECTURE Phil Merkey, Beowulf History, <https://www.beowulf.org/overview/history.html>, Last visited May 2024.
6. Luster, <https://www.lustre.org/documentation/>, Last visited May 2024.



- 
7. Robert G. Brown, <https://www.clustermonkey.net/Newbie/matching-cluster-hardware-to-your-application.html>, published: 21 May 2007, Last visited May 2024.