CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

PARAM SHAVAK

User Manual

PARAM SHAVAK USER MANUAL

www.cdac.in

Copyright Notice

Trademarks

Intended Audience

This document is meant for PARAM Shavak users. This document assumes that they have the following skills depending on the target platform:

- Basic Linux Commands

Typographic Conventions

Symbol	Meaning
Blue underlined text	A hyperlink or link you can click to go to a related section in this document or to a URL in your web browser.
Bold	The names of menus, menu items, headings, and buttons.
Italics	Variables or placeholders or special terms in the document.
`Courier font`	Linux commands and options

Getting help

For technical assistance or license renewal, please send an email to hpcs@cdac.in.

Give us your feedback

We value your feedback. Kindly send your comments on content of this document to hpcs@cdac.in. Please include the page number of the document along with your feedback.

DISCLAIMER

The information contained in this document is subject to change without notice. C-DAC shall not be liable for errors contained herein or for incidental or consequential damages in connection with the performance or use of this manual.

Table of Contents

Introduction 1
Overview 1
System Requirements 3
Hardware Overview 3
Software Overview 4
HPC Applications 4
PARAM Shavak System Architecture 5
Accessing PARAM Shavak 6
ssh Login from Linux 6
ssh Login from PuTTY 7
PARAM Shavak - User Interface 10
PARAM Shavak Desktop 10
Accessing PARAM Shavak Tools through GUI 11
System Administration 12
User Management 12
User, Group and Quota Management using Command Line 12
Adding a New User 12
Adding a Group 14
Assigning Quotas for User 14
User and Group Management using GUI 16
Adding a New User 18
Modifying User Properties 20
Adding a New Group 22
Network Configuration 23
Establishing an Ethernet Connection 23
MIC Administration on Param Shavak 27
Environment Variables 30
PARAM Shavak Security 32
Additional Linux Commands 33
The uptime Command 33
The w Command 33
The users Command 34

The who Command 34
The whoami Command 34
The ls Command 35
The crontab Command 35
The less Command 35
The more Command 36
The cp Command 36
The mv Command 36
The cat Command 36
The cd Command (change directory) 37
The pwd Command (print working directory) 37
The sort Command 37
The vi Command 37
The ssh Command (Secure Shell) 37
The ftp and sftp Commands 38
The service Command 38
The free command 38
The top Command 39
The tar Command 39
The grep Command 39
The find Command 40
The lsof Command 40
The last command 40
The ps command 41
The kill command 41
The rm command 41
The mkdir command 42
File Commands 42
Process Management 43
ssh 43
Searching 43
System Info 43
Compression 44
Network 44
Shortcuts 45
Application Usage Manual 46
Intel® Parallel Studio XE 2015 Cluster Edition Usage 46
Introduction 46
Intel® Advisor XE 48

Intel® C++ Compiler XE and Intel® Fortran Compiler XE 48
Compilation and Execution of C, C++, and Fortran Programs 48
Compiling and running 49
Compilation and Execution of OpenMP Application (C, C++ and Fortran Program) 49
Compiling and running 49
Intel® Inspector XE 49
Intel® Integrated Performance Primitives (Intel® IPP) 50
Intel® Math Kernel Library (Intel® MKL) 51
Intel® MPI Benchmarks 51
Intel® Threading Building Blocks 52
Intel® Trace Analyzer and Collector 52
Intel® VTune™ Amplifier XE 53
Intel® MPI Library 54
Running an MPI Program Using Intel® MPI Library 55
Building an MPI Application 55
Running an MPI Application 55
References: 56
Torque Resource Manager 57
Job submission through torque 57
Serial C/C++/Fortran Program 58
Ganglia Monitoring System 61
Using the ganglia 61
System Health 61
ONAMA 64
Onama Execution Model Interface 64
CHReME 65
Features 65
HPC Applications 66
Reference 68
Index 69

Chapter 1

Introduction

World over, there is a growing recognition for information technology (IT) and in particular high performance computing (HPC) for computational simulation and modeling. These are key technology resources for economic growth, environmental understanding, scientific and engineering research breakthroughs, and for sustaining competitiveness in strategic areas.

Several nations in the world are taking initiatives to provide high-end computing resources for their scientists and engineers for educational and research purposes.

Overview

In a bid to contribute to the research and development of India, C-DAC has developed a low cost and an energy efficient solution, which is as good as a supercomputer in a box namely – PARAM Shavak. This supercomputer, PARAM Shavak, has been equipped with advanced technologies to perform the high-end computations for the scientific, engineering and academic community. This new facility will be able to address and catalyze the research using modeling and simulation and promises to be a boon to the academic and scientific community. Additionally, it will help in creating a workforce that is aware of HPC skills (capacity building) and promote research and teaching by integrating leading-edge emerging technologies.

PARAM Shavak - Supercomputer in a box with unbound performance features:

- HPC system in a table top model.
- Powered with minimum two multicore CPUs, each with at least 10 cores along with either one or two cores or GPU accelerator cards.
- Phenomenal for academic, scientific and research institutions that are on the verge of adopting high performance computing culture.
- Equipped with indigenously developed software technologies by C-DAC for HPC applications in academic and scientific domains.
- Easy to deploy solution with minimal data-centre infrastructure.

- Pre-loaded with parallel programming development environment.
- Customizable as per the user hardware and software requirements.
- 2.3 TF and above computing power.
- Scalable model.
- Access to C-DAC PARAM Yuva II at National PARAM Supercomputing facility for computations on a larger scale as per the NPSF usage policy.
- Support for C-DAC's Reconfigurable Computing System technology to speed up applications through hardware.
- Resource for parallel programming, training and workshops.
- Affordable computing environment for the faculty, students – both undergraduate and post graduate PhD scholars

Chapter 2

System Requirements

This chapter provides the details of the configuration details.

Hardware Overview

Following are the hardware configuration details of PARAM Shavak –

Node	
Processor	Dual socket Intel Xeon E5-2600 v2 series/ with 10 cores each with minimum 2.2 GHz clock speed with min. specfp2006_rate of 580
RAM	64 GB ECC DDR3 1866 MHz RAM in balanced configuration
Network	Two 1GbE network ports
Accelerator's Slot	2 x16 PCI-E Gen3 slots for GPU/co-processors
DVD	One number internal DVD RW/DVD combo
HDD	4 x 2 TB SATA/NL-SAS disks with support for hardware RAID 0,1 and 5 Configured with RAID 5 Total usable space 5.456TB ○ 1.455 TB for System, applications ○ 4.0 TB for user home
Graphics	On board or add-on card for visualization
Monitor	19" TFT LCD monitor along with USB keyboard and mouse
Accelerator	i. Intel Xeon Phi 3120A/3120P card (Max 2) or ii. nVidia K20/K40(Max 2) series

Software Overview

Following are the software requirements for PARAM Shavak.

System Software Details

OS	CentOS 6.5
Drivers	MPSS/CUDA
Tools	ONAMA, CHReME
Libraries	mpich2, Intel®
Compilers	GCC, Intel®
SDK	Netbeans
Job Scheduler	Torque
Monitoring Tool	Ganglia

HPC Applications

Bio-informatics	mpiBLAST
Molecular Dynamics	GROMACS
Weather Forecasting and Oceanography	WRF, MOM
Quantum Chemistry	Abinit, NWChem
Materials Science	Quantum Espresso
CFD	OpenFOAM
Visualization Tools	GrADS, Ferret
Intel Xeon Phi	Mdynamics, Clustalw

PARAM Shavak System Architecture

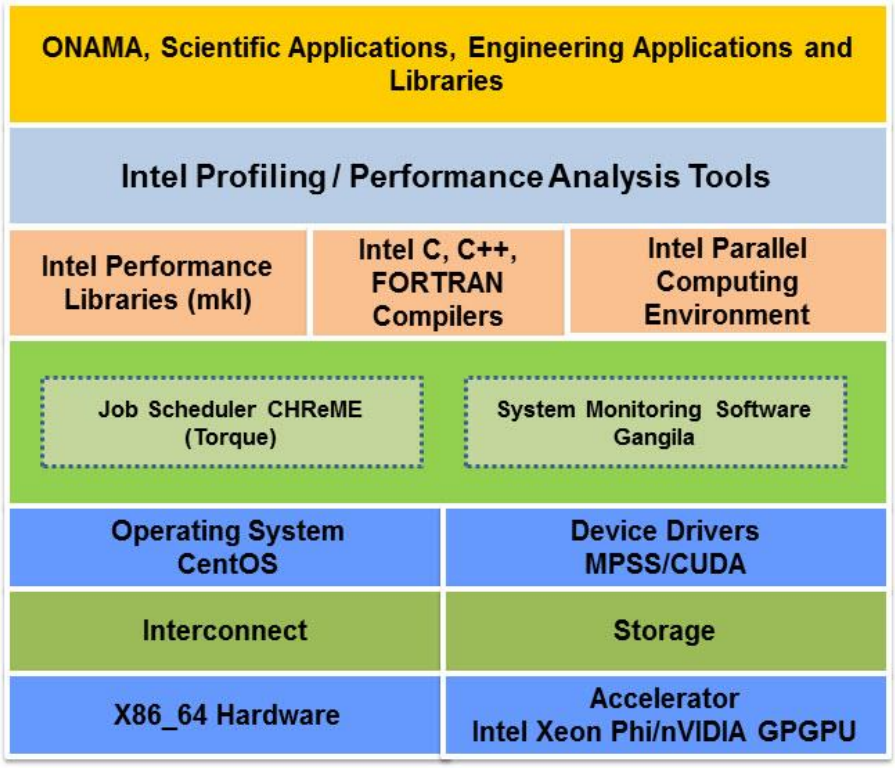


Figure 1 - PARAM Shavak Software Architecture Block Diagram

Chapter 3

Accessing PARAM Shavak

Access to the supercomputers is available through encrypted connections, such as ssh. The ssh program allows you to open a text console session on a remote computer. Thus ssh is essentially an encrypted version of telnet. The scp and sftp commands are used for transferring files between computers. The sftp program works like an encrypted form of ftp.

When you connect to a computer, you must enter the `user_id` and the appropriate password. This is done from a shell or terminal prompt on a Linux or UNIX system or Macintosh, from a UNIX shell on a PC (using a Unix-in-windows tool, such as Cygwin or MKS), or using a graphical `ssh` program under windows, such as `PuTTY`.

ssh Login from Linux

Connections to the supercomputer via `ssh` can be made from a terminal window. On a Macintosh computer, the `Terminal.app` program is in the Utilities section. In some Linux distributions, the terminal icon is on the menu bar, while others have a menu pick to open it. For instance in `Ubuntu, Applications > Accessories > Terminal`. A Windows computer with `Cygwin` installed have a `Cygwin` icon on the desktop. Once the terminal window is open, the commands syntax is usually the same on all systems. Typical variations on the `ssh` command line syntax are:

```
ssh<user_id>@hostname
ssh -l <user_id> hostname
```

Where hostname is the name of the login node and `<user_id>`is replaced by your account name.

Examples:

```
local>sshtest@shavak
local>ssh -l test@shavak
```

The captcha is displayed on the screen as shown below. Enter the correct string and press Enter.

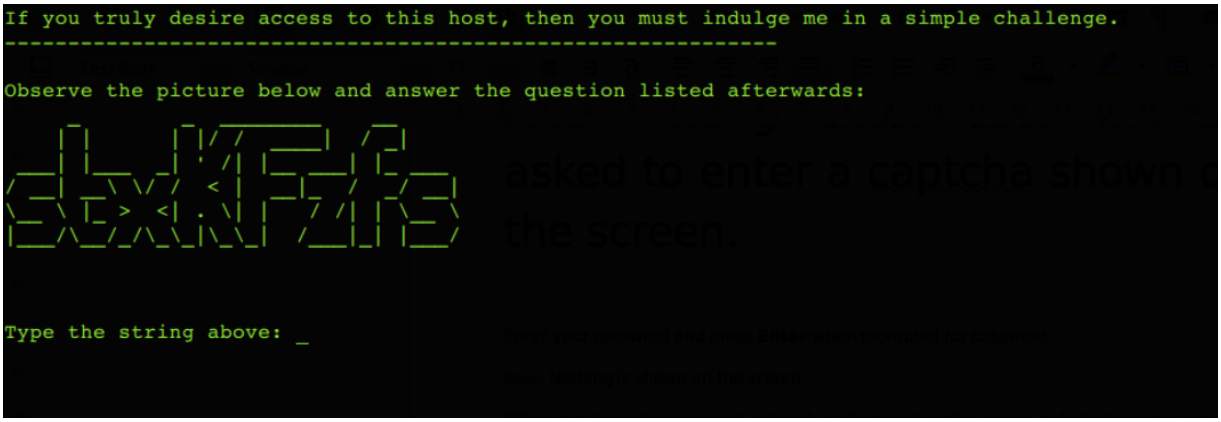


Figure 2 – Captcha Screen

Now enter your password and press **Enter**.

Note: Nothing is shown on the screen.

When you type the password, asterisk is displayed on the screen (which shows an onlooker how many characters are in your password). If you get a message about password database being too restrictive, it means that you typed an invalid password.

Note: You must type the provided password in the same case (lower or upper).

Once you enter the correct password, a message displays with current announcements. You are now logged on to the supercomputer and can use any of the Linux modules. To logout from the supercomputer, type exit and press `Enter`.

ssh Login from PuTTY

For Windows operating system, you need to additionally use a `ssh` utility such as putty. You can download the free version of the Putty utility from the http://putty.en.softonic.com/download site.

It can be used in conjunction with some X-window clients, such as X-Win32.

1. Download `putty.exe` from the website `putty.exe`. Unlike most software packages, putty.exe is not a package with an installation program. It is a single executable file that needs to be run to use `PuTTY`.

2. Save putty.exe directly to the desktop. Another option is to save putty.exe to a directory of your choice, then create a desktop shortcut to it.

3. To connect to the supercomputers with `PuTTY`, double click the `PuTTY` icon on the Windows desktop. Windows may pop up a security warning message that requires you to click on `Allow`, or `Run`, or `Continue` in order to allow the

PuTTY software to run. This opens the PuTTY Configuration window, shown in Figure.

4. Enter the host name in the Host Name (or IP Address) box. The rest of the settings should be correct with the defaults.

Note: These are Port 22, SSH, and keyboard-interactive under Connection -> SSH -> Auth.

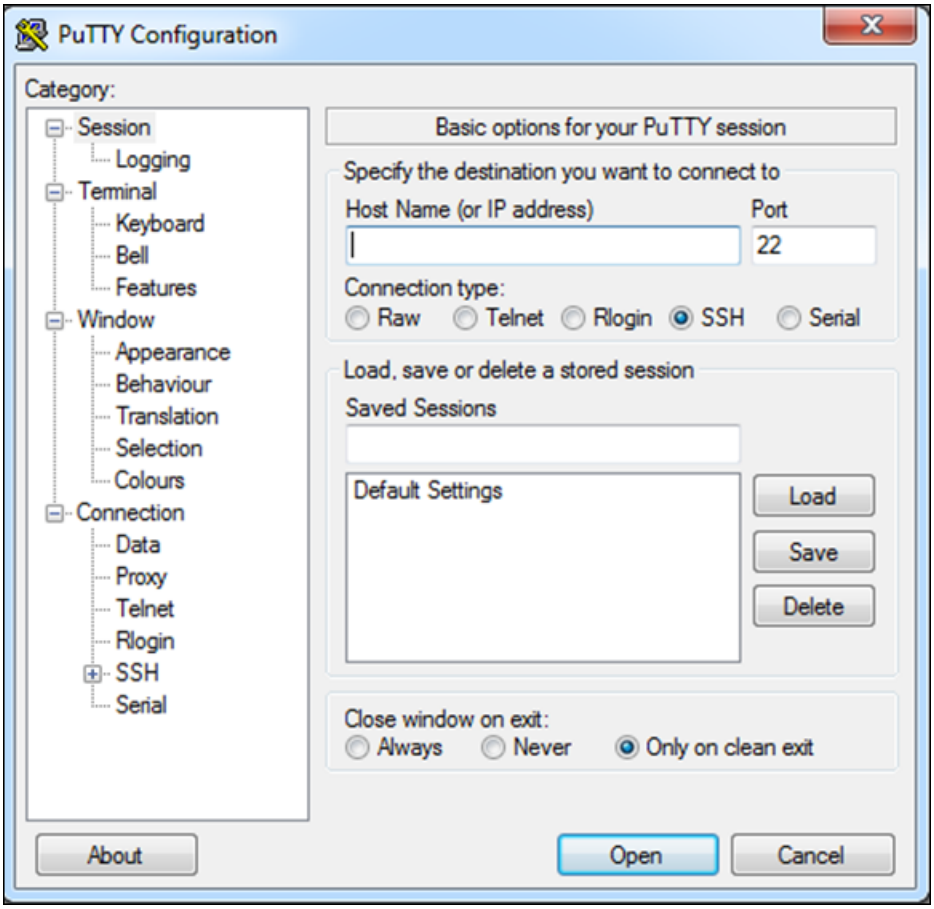


Figure 3 - PuTTY Configuration Dialog Box

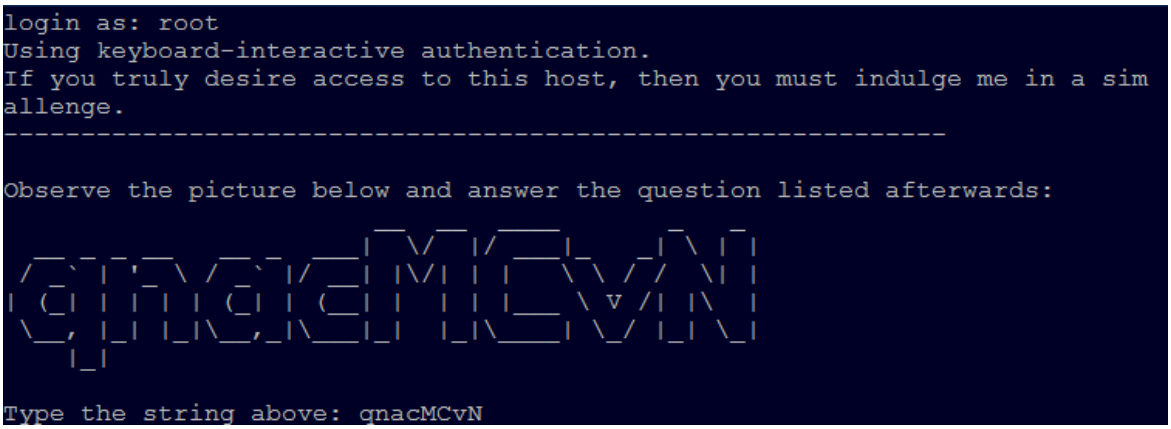
When using the commands involving a host name, such as ssh or scp, you should specify the full name, such as paramshavak.

5. Click Open. The initial window is replaced by the PuTTY terminal window shown in figure.

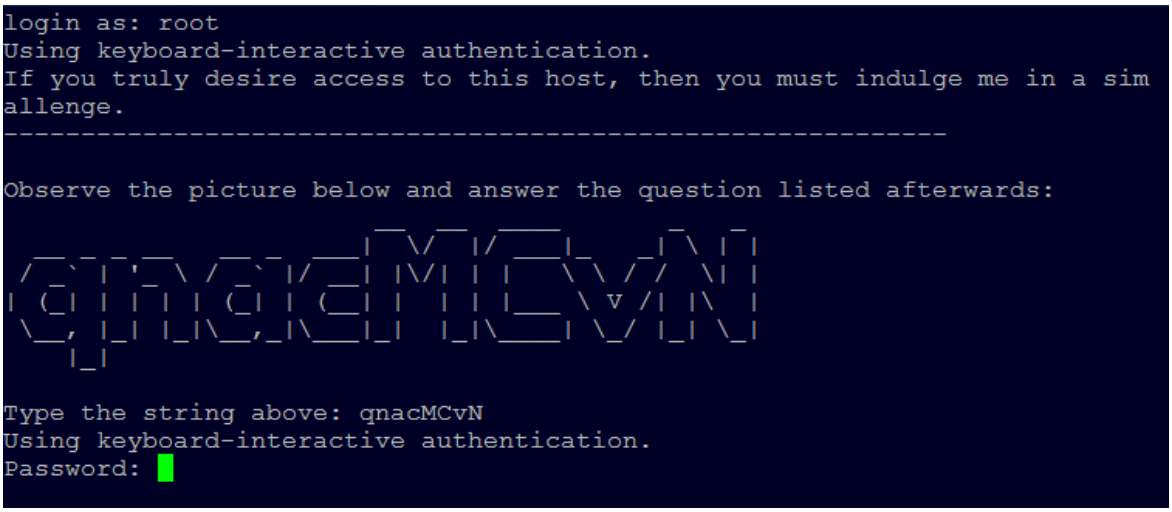
6. The PuTTY terminal window appears with the login as: prompt.

7. Enter your supercomputer account name and press Enter.

8. Enter the correct string for the Captcha.



9. Enter the password, and press Enter.



After you enter the correct password, a message displays with the current announcements. You are now logged on to the supercomputer, and can use any of the Linux, module, or queue system commands described in this manual.

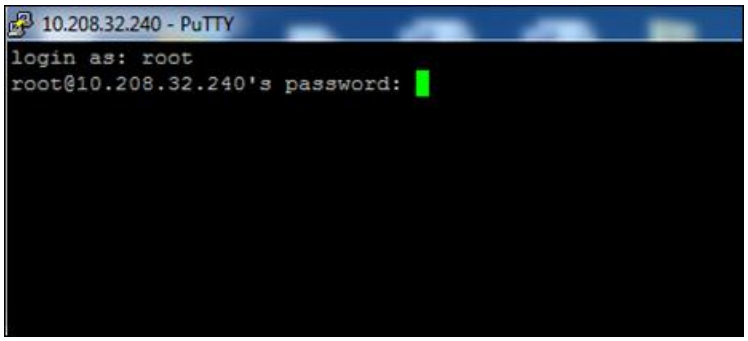


Figure 4 - PuTTY Terminal Window

Note: Nothing is shown on the screen when you type the password, not even asterisks (which show an onlooker of how many characters are in your password). If you get a message about password database being too restrictive, it means that you mistyped the password. The password must be typed exactly as sent to you including the use of upper and lower case characters.

PARAM Shavak - User Interface

PARAM Shavak Desktop

After successful login, the following Desktop environment appears.

Figure 5 - PARAM Shavak Desktop View

Accessing PARAM Shavak Tools through GUI

For accessing PARAM Shavak tools:

- Click the **Applications** tab > PARAM Shavak Tools.

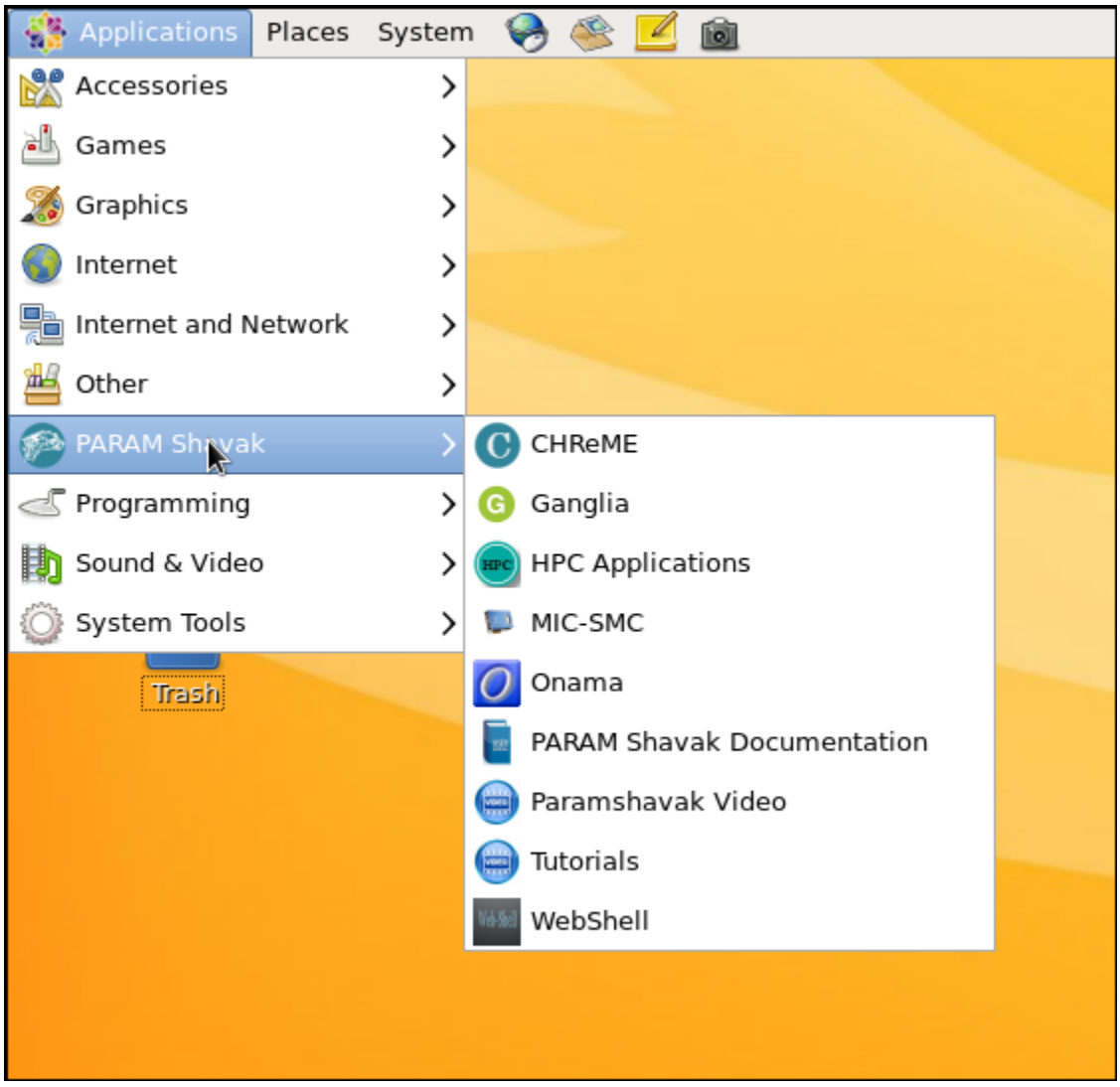


Figure 6 - PARAM Shavak Application Menu

Here, you can find the customized tools such as **HPC Applications**, **Chreme**, **Ganglia**, **HPC Applications**, **MIC-SMC**, **Onama**, **PARAM Shavak Documentation**, **PARAM Video**, and **Turorials**.

Chapter 4

System Administration

This chapter describes the type of users you can create and manage.

User Management

This section describes the methods to create user and user groups using GUI and CLI.

User, Group and Quota Management using Command Line

This section describes steps to add and manage a user or user group through command line interface.

Adding a New User

To add a user to the system:

1. Issue the `useradd` command to create a locked user account:
2. `useradd`<username>
3. Unlock the account by issuing the `passwd` command to assign a password and set password aging guidelines:
4. `passwd`<username>

Table 1- Add User Command's Options List

Option	Description
-c <comment>	<comment> can be replaced with any string. This option is generally used to specify the full name of a user.
-d<home-dir>	Home directory to be used instead of default /home/<username>/
-e<date>	Date for the account to be disabled in the format YYYY-MM-DD
-f<days>	Number of days after the password expires until the account is disabled. If 0 is specified, the account is disabled immediately after the password expires. If -1 is specified, the account is not to be disabled after the password expires.
-g<group-name>	Group name or group number for the user's default group. The group must exist prior to being specified here.
-G<group-list>	List of additional (other than default) group names or group numbers, separated by commas, of which the user is a member. The groups must exist prior to being specified here.
-m	Create the home directory if it does not exist.
-M	Do not create the home directory.
-n	Do not create a user private group for the user.
-r	Create a system account with a UID less than 500 and without a home directory
-p<password>	The password encrypted with crypt
-s	User's login shell, which defaults to /bin/bash
-u<uid>	User ID for the user, which must be unique and greater than 499

Note: The password must be of 8 characters. It should combination of alphabets, numbers and special characters. You must include at least one upper case character and one lower case character.

Adding a Group

To add a group to the system, use the following command.

`groupadd`<group-name>

Table 2 - Add Group Command's Options List

Option	Description
-g<gid>	Group ID for the group, which must be unique and greater than 499
-r	Create a system group with a GID less than 500 When used with -g<gid> and <gid> already exists, groupadd will choose
-f	Another unique<gid> for the group.

Assigning Quotas for User

To configure the quota for a user, as root in a shell prompt, execute the command:

`edquota`<username>

Perform this step for each user who needs a quota.

For example:

If a quota is enabled in /etc/fstab for the /home partition ((/dev/VolGroup00/LogVol02 in the example below) and the command `edquota testuser` is executed, the following is shown in the editor configured as the default for the system:

```
Disk quotas for user testuser (uid 501):
 Filesystem                   blocks  soft  hard  inodes   soft  hard
/dev/VolGroup00/LogVol02     440456     0     0     376745     0     0
```

The text editor defined by the EDITOR environment variable is used by `edquota`.

Note: To change the editor, set the EDITOR environment variable in your `~/.~/.bash_profile` file to the full path of the editor of your choice.

This section describes the details of the `bash_profile` file.

- First column: Name of the file system that has a quota enabled for it.

- Second column: Shows how many blocks you are currently using.
- Next two columns: Used to set soft and hard block limits for the user on the file system.
- inodes column: Shows how many inodes you are currently using.

Last two columns: Used to set the soft and hard inode limits for the user on the file system.

Note: The hard block limit is the absolute maximum amount of disk space that a user or group can use. Once this limit is reached, no further disk space can be used.

Note: The soft block limit defines the maximum amount of disk space that can be used. However, unlike the hard limit, the soft limit can be exceeded for a certain amount of time. That time is known as the grace period.

The grace period can be expressed in seconds, minutes, hours, days, weeks, or months.

If any of the values are set to 0, that limit is not set. In the text editor, change the desired limits.

For example:

```
Disk quotas for user testuser (uid 501):
Filesystem              blocks   soft   hard   inodes   soft   hard
/dev/VolGroup00/LogVol02 440456 470000 490000   376745    0      0
```

To verify that the quota for the user has been set, use the command:

```
quota testuser
```

User and Group Management using GUI

The User Manager allows you to view, modify, add, and delete local users and groups.

To use the **User Manager**, the following conditions must be met:

- You must be running the X Window System
- Have root privileges
- The system-config-users RPM package must be installed.

To start the User Manager from the desktop:

- Click **System** (on the panel) >**Administration**>**Users & Groups**.

Note: You can also type the command system-config-users at a shell prompt (for example, in an XTerm or a GNOME terminal).

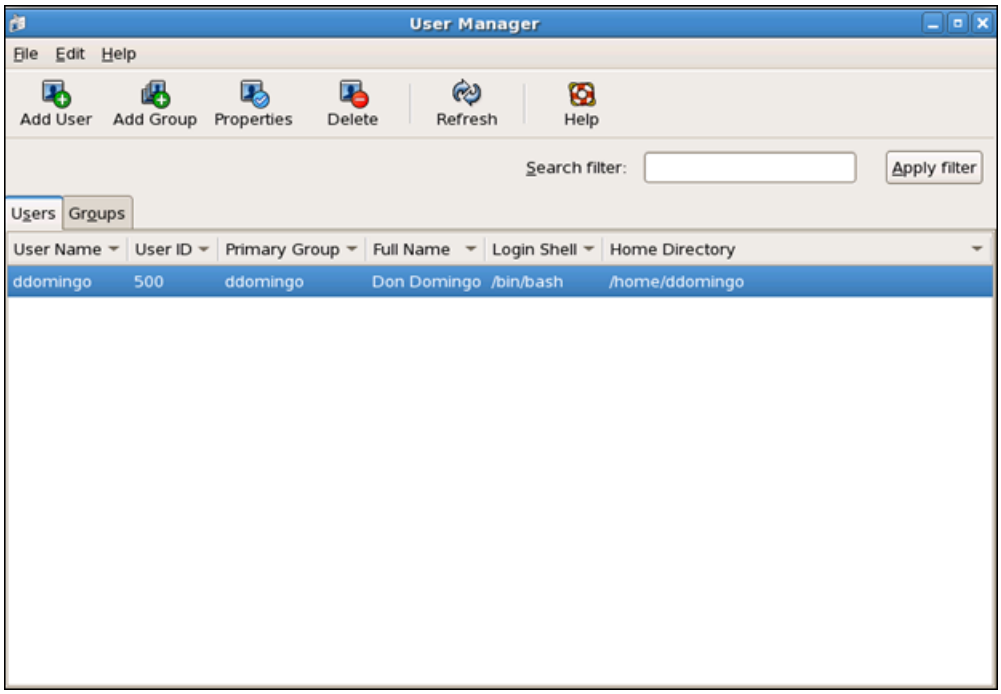


Figure 7- User Manager

- To view a list of local users on the system, click the **Users** tab.
- To view a list of local groups on the system, click the **Groups** tab.
- To find a specific user or group, type the first few letters of the name in the **Search** filter field. Press **Enter** or click the **Apply** filter button. The filtered list is displayed.

- To sort the users or groups, click on the column name. The users or groups are sorted according to the value of that column.

- Red Hat Enterprise Linux reserves user IDs below 500 for system users. By default, **User Manager** does not display system users. To view all users, including the system users, go to **Edit** >**Preferences** and uncheck **Hide system users and groups** from the dialog box.

Adding a New User

To add a new user:

1. Click the **Add User** button. A window as shown in the **Create New User** figure appears.

Figure 8 - Create New User Window

2. Type the username and full name for the new user in the appropriate fields.

3. Type the password in the **Password** and **Confirm Password** fields. The password must be at least six characters.

Note: The password must be of 8 characters. It should combination of alphabets, numbers and special characters. You must include at least one upper case character and one lower case character.

4. Select a login shell. If you are not sure which shell to select, accept the default value of `/bin/bash`. The default home directory is `/home/<username>/`. You can change the home directory that is created for the user, or you may choose not to create the home directory by unselecting **Create home directory**.

Note: If you select to create the home directory, default configuration files are copied from the `/etc/skel/`directory into the new home directory.

Red Hat Enterprise Linux uses the user private group (UPG) scheme. The UPG scheme does not add or change anything in the standard UNIX way of handling groups; it offers

a new convention. Whenever you create a new user, by default, a unique group with the same name as the user is created. If you do not want to create this group, unselect **Create a private group for the user**.

5. To specify a user ID for the user, select **Specify** user ID manually. If the option is not selected, the next available user ID above 500 is assigned to the new user. As Red Hat Enterprise Linux reserves user IDs below 500 for system users, it is not advisable to manually assign user IDs from 1-499.

6. Click **OK** to create the user.

Modifying User Properties

To view the properties of an existing user:

1. Click on the **Users** tab.
2. Select the user from the user list.
3. Click **Properties** from the menu (or choose File > Properties from the pull down menu). A window similar to Figure, **User Properties** appears.

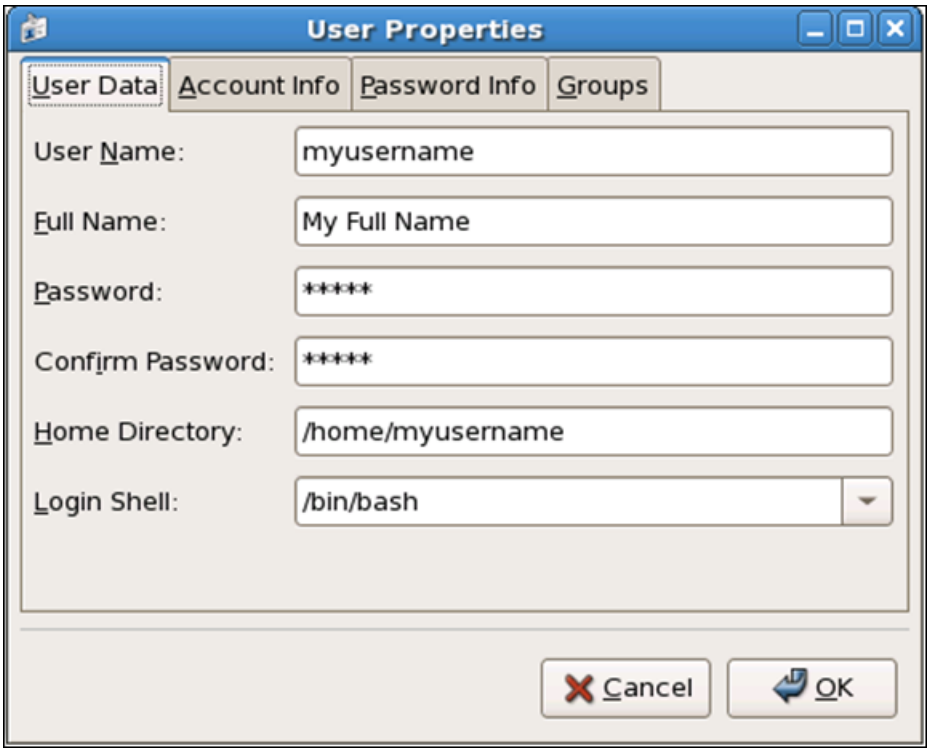


Figure 9 - User Properties Window

The **User Properties** window is divided into multiple tabbed pages:

- User Data — Shows the basic user information configured when you added the user. Use this tab to change the user's full name, password, home directory or login shell.
- Account Info — Select **Enable account expiration** if you want the account to expire on a certain date. Enter the date in the provided fields. Select **Local password is locked** to lock the user account and prevent the user from logging into the system.
- Password Info — Displays the date when the user's password was last changed. To force the user to change passwords after a certain number of days, select

Enable password expiration and enter a desired value in Days before changing the required field. You can configure the following:

- The number of days before the user's password expires
- The number of days before the user is warned to change passwords
- Days before the account becomes inactive can also be changed.
- Groups — Allows you to view and configure the **Primary Group of the user**, as well as other groups that you want the user to be a member of.

Adding a New Group

To add a new user group:

1. Click the **Add Group** button. A window similar to **Figure New Group** appears.
2. Type the name of the new group to create one.
3. To specify a group ID for the new group, select **Specify group ID** manually and select the GID.

Note: Red Hat Enterprise Linux reserves group IDs lower than 500 for system groups.

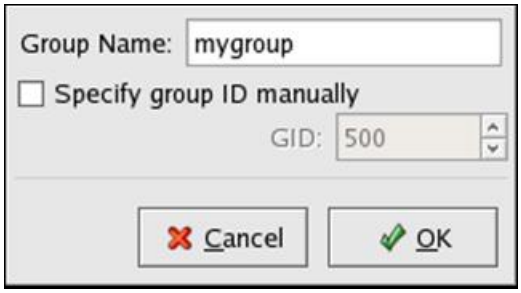


Figure 10 - New Group Dialog Box

4. Click **OK** to create the group. The new group appears in the group list.

Network Configuration

This section describes the steps to configure the network.

Establishing an Ethernet Connection

Different networks are configured to use different network speeds; make sure your network interface card (NIC) is compatible with the network to which you want to connect. To establish an Ethernet connection, you need the following:

- A network interface card (NIC)
- A network cable (usually a CAT5 cable)
- A network to connect to.

To add an Ethernet connection:

1. Click the **Devices** tab.
2. Click the **New** button on the toolbar.
3. Select **Ethernet connection** from the **Device Type** list, and click **Forward**.
4. If you have already added the network interface card to the hardware list, select it from the **Ethernet card** list. Otherwise, select **Other Ethernet Card** to add the hardware device.

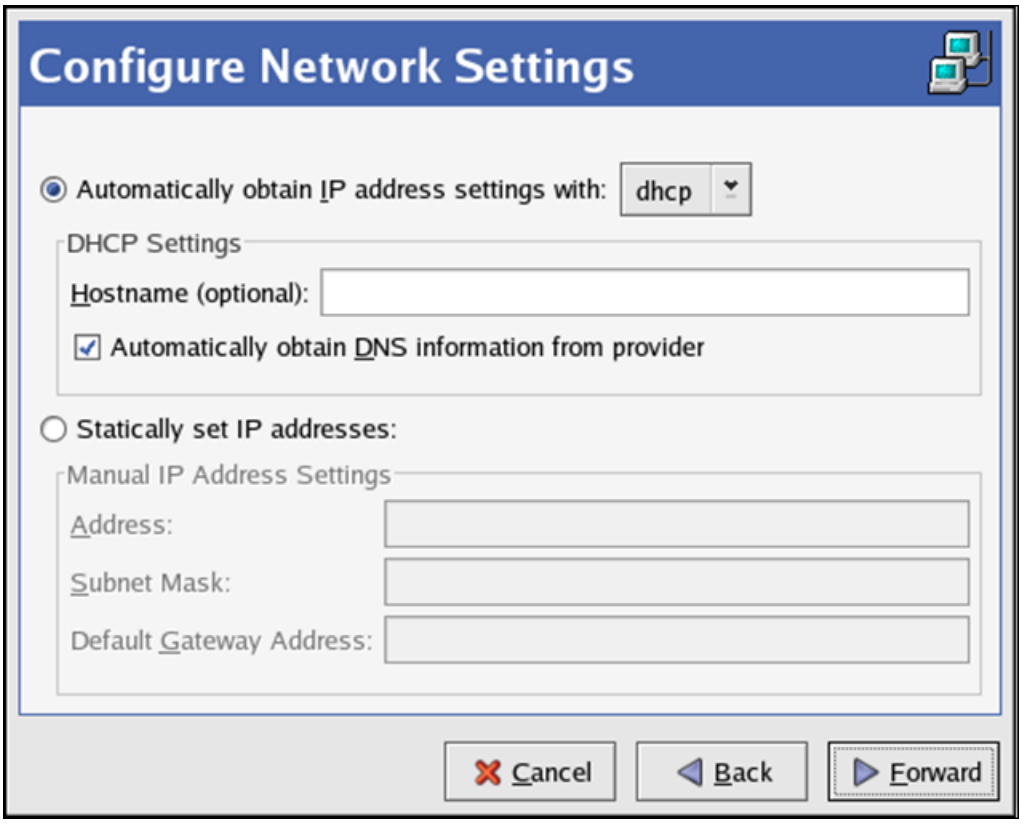


Figure 11 - Network Configuration Dialog Box

NOTE: The installation program detects supported Ethernet devices and prompts you to configure them. If you configured any Ethernet devices during the installation, they are displayed in the hardware list on the **Hardware** tab.

5. If you selected **Other Ethernet Card**, the **Select Ethernet Adapter** window appears.

    - Select the manufacturer and model of the Ethernet card.

    - Select the device name. If this is the system's first Ethernet card, select **eth0** as the device name; if this is the second Ethernet card, select **eth1** (and so on). The **Network Administration** tool also allows you to configure the resources for the NIC.

    - Click **Forward** to continue.

6. In the **Configure Network Settings** window shown in Figure **Ethernet Settings**, choose DHCP or a static IP address. If the device receives a different IP address each time the network is started, do not specify a hostname. Click **Forward** to continue.

7. Click **Apply** on the **Create Ethernet Device** page.

After configuring the Ethernet device, it appears in the device list as shown in the figure, **Ethernet Device**.

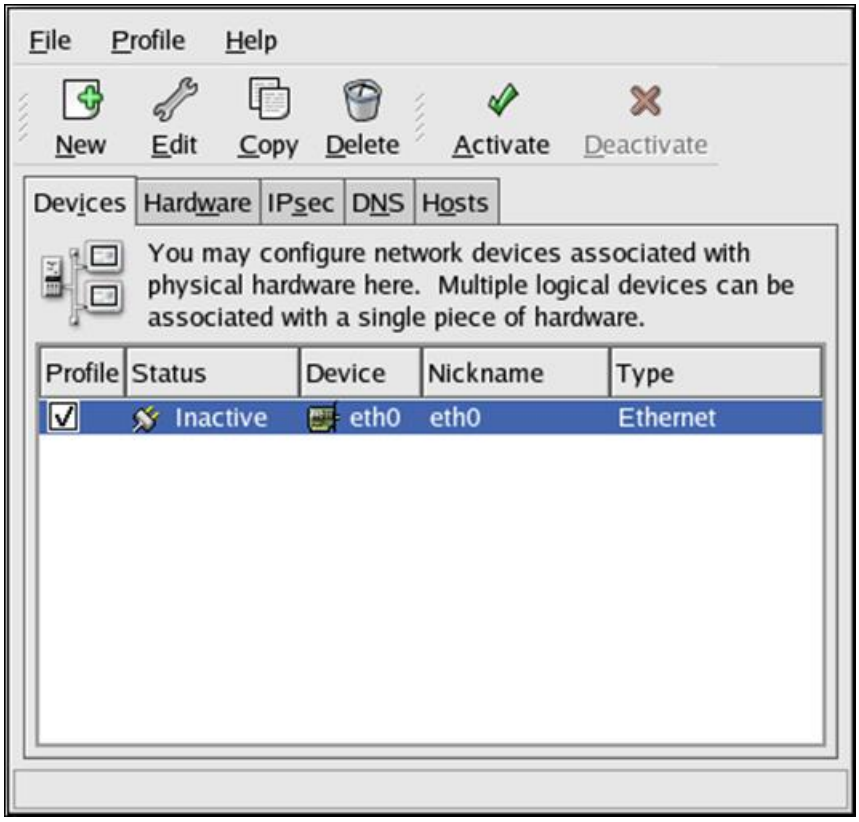


Figure 12 - Ethernet Devices

Note: To save changes, select **File > Save** to save the changes.

After adding the Ethernet device, you can edit its configuration by selecting the device from the device list and click **Edit**. For example, when the device is added, it is configured to start at boot time by default. To change this setting, select Edit and modify the Activate device when computer starts value and save the changes.

When the device is added, it is not activated immediately and shows the status as inactive. To activate the device, select it from the device list and click the **Activate** button. You need not repeat this step again once it is activated.

If you associate more than one device with an Ethernet card, the subsequent devices are device aliases. A device alias allows you to setup multiple virtual devices for one physical device, thus giving one physical device more than one IP address. For example, you can configure an eth1 device and an eth1:1 de vice.

MIC Administration on Param Shavak

1. Create user on Intel ® Xeon Phi™ as well as system.

```
mic_useradd
```

Note: user must not exist on the system.

Syntax:

```
mic_useradd [username]
```

2. Change or create user password on system and Intel® Xeon Phi™.

```
mic_passwd
```

Syntax:

```
mic_passwd [username]
```

3. Delete user from Intel® Xeon Phi™ as well as system.

```
mic_userdel
```

Syntax:

```
mic_userdel [username]
```

4. Sync user on Intel® Xeon Phi™

Syntax:

```
mic_sync [username]
```

Note: Before creating account on Xeon Phi, create a user on the host machine with a password.

Example:

```
[root@paramshavak ~] # mic_sync  test
```

5. How to login to Intel® Xeon Phi™

```
[root@paramshavak ~]# ssh mic0
```

```
[root@paramshavak-mic0 ~]#
```

```
[root@paramshavak ~]# ssh mic1
```

```
[root@paramshavak-mic1 ~]#
```

6. How to mount directory on Intel Xeon Phi™

Syntax:

```
micctrl --addnfs=ip address [br0]:/directory -dir=/directory
```

Example:-`micctrl --addnfs=10.10.10.254:/test --dir=/test`

Note: In PARAM Shavak /home and /opt is already mounted on Intel Xeon Phi card.

7. Add the entry to the /etc/exports file that you want to export.

```
[root@paramshavak~]# vim /etc/exports
```

```
/test       10.10.10.0/24 (rw, no_root_squash)
```

8. Intel® Xeon Phi™ Coprocessor Status

The **status** option displays the status of the Intel® Xeon Phi™ coprocessors in the system. It the status is "online" or "booting" it also displays the name of the associated boot image.

Example:

```
[root@paramshavak ~]# micctrl-s
```

```
mic0: online (mode: linux image: /usr/share/mpss/boot/bzImage-
knightscorner)
mic1: online (mode: linux image: /usr/share/mpss/boot/bzImage-
knightscorner)
```

9. micsmc

`micsmc` is the binary executable for the Intel® Xeon Phi™ coprocessor Platform Status Panel. The `micsmc` tool monitors core utilization, temperature, memory usage, power usage statistics, and error logs, among other features.

```
[root@paramshavak ~]# micsmc
```

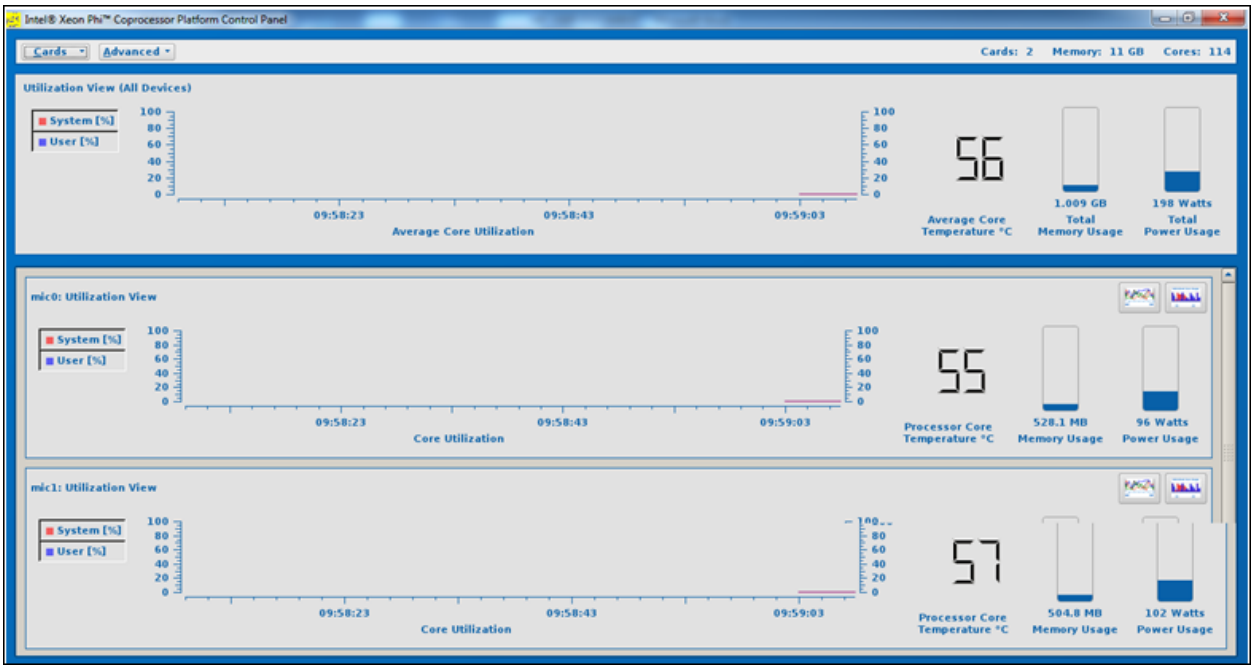


Figure 13 - The micsmc Output of PARAM Shavak

10. Waiting for Intel® Xeon Phi™ Coprocessor State Change

The **wait** (or w) option waits for the status of the Intel® Xeon Phi™ coprocessor to be either "online" or "ready". It also allows for a brief pause to the "ready" state during **mpssd** startup. It is intended for users to verify the **mpssd** daemon startup, shutdown, or reset procedure is complete. It has a built-in timeout value of 300 seconds.

Syntax:

```
micctrl -w [mic card list]
```

Example:

```
[root@paramshavak ~]$ micctrl -w mic0 mic1
```

```
mic0: online
mic1: online
```

For detailed information, follow the MPSS_Users_Guide which is located in the location /opt/manual/ MPSS_Users_Guide.pdf.

Environment Variables

Environment variables are values that are visible to all of the software running on a computer. These are often used to tell the operating system how it should behave, and to point software packages to the required directory location to find important files.

To view the environment variables:

- Login to your Linux account and type the command `env`. This shows a list of all the environment variables that your account currently sees.
- Some are found in all Linux systems, some are specific to the bash shell, and some are specific to a given computer program. The following table lists a selection of the environment variables that are amongst the most important ones for account configuration.
- Some environment variables are redundant. For example, the variables `LIBRARY_PATH`, `LD_LIBRARY_PATH`, `LIBPATH`, and `SHLIB_PATH` point the operating system to static libraries and shared object files (so files are dynamic linked libraries, equivalent to .dll files in Windows).
- There are multiple environment variables doing the same job because some are used by different shells, or Linux distributions. Since different programs look at different ones, a redundant configuration keeps all of the programs finding the paths to the libraries.

The value of an environment variable can be displayed with the `echo` command. For example, the PATH environment variable tells the operating system where to look for programs to run. You can see where the run script program resides by typing which run_script.

Table 3 - Environment Variables

Variable	Definition
CLASSPATH	Java programs uses this to find their libraries
DISPLAY	Tells X Windows where to display graphics
HISTSIZE	Number of commands displayed by the history command
HOME	Your home directory
HOST HOSTNAME	The computer (or cluster node) you are logged i
INCLUDEDIR, INCLUDE	Paths to header files
LD_LIBRARY_PATH, **LIBRARY_PATH, LIBPATH, SHLIB_PATH**	Paths to static linked and dynamic linked libraries
LS_COLORS	Allows customizing colors used by the ls command
MANPATH	Paths to data for the man command
PATH PATH	paths to find executable files
PS1	Changes the bash command prompt
PWD Current Working directory	PWD Current Working directory
TERM	Terminal display settings
USER, LOGNAME	User name

Use the following `run_script` command to see all of the directories that the operating system is looking for.

```
echo $PATH
```

You can set new environment variables, or add data to existing environment variables. For example, you may want to create some of your own programs and scripts. In order to find specific when you run them, you can put them in a new sub-directory, typically

named /home/MYNAME/bin. In order to tell the operating system to look for your programs in this directory, you could add a line into the .bashrc.local file. For example:

```
export PATH="$PATH:/home/MYNAME/bin"
```

Note that by including $PATH: in the new value of PATH, you are appending a new directory onto the existing path list. The directory names are separated by colons. If you leave out this $PATH: part of the line you would be taking away all of the paths to the operating system commands, thus breaking most of the functionality of your account.

Environment variables can be used in shell scripts. For example, if you want a shell script to create a directory with your user name, you can use the following line:

For example:

```
mkdir /scratch/$USER
```

Environment variables are accessible within most compiled computer languages. There is a mechanism for accessing them.

Note: Some of variables have been exported and can be found in /etc/bashrc file

PARAM Shavak Security

PARAM Shavak incorporates following measures of security.

- **TripWire**: PARAM Shavak incorporates TripWire which serves as a host based intrusion detection system. It is useful for detecting intrusions after the event, it can also serve many other purposes, such as integrity assurance, change management and policy compliance.

- **SSH:** Terminal CAPTCHA is implemented for better security of PARAM SHAVAK against brute force attacks.

- **Strong Password policy:** Following password policy is implemented in PARAM Shavak

    - Minimum 8 characters
    - Combination of alphabets, special characters and numbers.
    - At least one upper case character (A-Z)
    - At least one lower case character (a-z)
    - At least two special characters (!@#$%^&*)
    - At least 2 numbers (0-9)

- Operating System and services are hardened in order shield the system against Common Vulnerabilities and Exposures.

Chapter 5

Additional Linux Commands

This chapter provides information on additional Linux commands that can be used for PARAM Shavak

The uptime Command

In Linux, the `uptime` command shows how long the system is running and the number of users that are currently logged in. It also displays load average for 1, 5 and 15 minutes intervals.

```
# uptime
16:39:28 up 13 days, 21:15, 2 users, load average: 0.00, 0.00,
0.00
```

The w Command

The `w` command displays users currently logged in and their process along-with shows load averages. It also shows the login name, tty name, remote host, login time, idle time, JCPU, PCPU, command and processes.

```
#w
16:49:00 up 13 days, 21:24,  2 users,  load average: 0.00, 0.00,
0.00
USER    TTY     FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
test     pts/0   test123.shavak 15:44    59:18    0.52s  0.24s sshd:
test123.shavak [priv]
root     pts/2    test123.shavak 16:39    0.00s  0.11s  0.01s w
```

Following is a list of available options:

`-h`: Displays no header entries.

`-s`: Without JCPU and PCPU.

`-f`: Removes from field.

`-V`: (upper letter) – Shows versions.

The users Command

The `users` command displays currently logged in users. This command does not have parameters other than help and version.

```
# users
test123 root
```

The who Command

The `who` command simply returns user name, date, time and host information. The `who` command is similar to `w` command. Unlike `w`, the `who` command does not print what users are doing. Let's illustrate and see the difference between `who` and `w` commands.

```
# who
test123  pts/0        2014-07-22 15:44 (test123.shavak)
root     pts/2        2014-07-22 16:39 (test123.shavak)
# w
16:49:00 up 13 days, 21:24,  2 users,  load average: 0.00, 0.00,
0.00
USER   TTY    FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
test   pts/0    test123.shavak 15:44    59:18    0.52s  0.24s sshd:
test123.shavak [priv]
root   pts/2  test123.shavak 16:39    0.00s  0.11s  0.01s w
```

Following is a list of the who command Options:

`-b`: Displays last system reboot date and time.

`-r`: Shows current runlet.

`-a`, –all: Displays all information cumulatively.

The whoami Command

The `whoami` command prints the name of the current user. You can also use the `whoami` command to display the current user. If you are logged in as a root using the `sudo` command, the `whoami` command returns root as current user. Use the `whoami` command if you want to know the exact user logged in.

```
# whoami
root
```

The ls Command

The `ls` command displays list of files in human readable format.

```
# ls -l
total 270
dr-xr-xr-x.   2 root root   4096 Jul  8 14:52 bin
dr-xr-xr-x.   5 rootroot   1024 Jun 24 09:47 boot
drwxr-xr-x.  20 rootroot   3820 Jul 10 11:29 dev
drwxr-xr-x. 140 root root  12288 Jul 21 19:16 etc
```

Sort file as per last modified time.

```
# ls -ltr
total 270
drwxr-xr-x.   2 root root   4096 Sep 23  2011srv
drwxr-xr-x.  14 rootroot   4096 Feb 27 08:16 usr
drwx------.   2 root root  16384 Jun 11 11:44 lost+found
drwxr-xr-x.   2 root root   4096 Jun 23 11:06 NFS
drwxr-xr-x.   4 rootroot   4096 Jun 23 15:27 share
```

The crontab Command

List schedule jobs for current user with the `crontab` command and `-l` option.

```
# crontab -l
00 10 * * * /bin/ls>/ls.txt
```

Edit the `crontab` with the `-e` option. The following example opens schedule jobs in VI editor. Make necessary changes and quit pressing the: wq keys which saves the setting automatically.

```
# crontab -e
```

The less Command

The `less` command allows viewing the file promptly. You can do page up and down. Press `q` to quit from fewer windows.

```
# less install.log.syslog
<86>Jun 11 11:48:52 groupadd[2102]: group added to /etc/group:
name=dbus, GID=81
<86>Jun 11 11:48:52 groupadd[2102]: group added to /etc/gshadow:
name=dbus
<86>Jun 11 11:48:52 groupadd[2102]: new group: name=dbus, GID=81
<86>Jun 11 11:48:52 useradd[2106]: new user: name=dbus, UID=81,
GID=81, home=/, shell=/sbin/nologin
<86>Jun 11 11:49:09 groupadd[2171]: group added to /etc/group:
name=usbmuxd, GID=113
```

```
<86>Jun 11 11:49:09 groupadd[2171]: group added to /etc/gshadow:
name=usbmuxd
```

The more Command

The `more` command allows to quickly view file and shows details in percentage. You can do page up and down. Press `q` to quit out from more windows.

```
# more install.log.syslog
<86>Jun 11 11:50:05 groupadd[2564]: group added to /etc/group:
name=rtkit, GID=497
<86>Jun 11 11:50:05 groupadd[2564]: group added to /etc/gshadow:
name=rtkit
<86>Jun 11 11:50:05 groupadd[2564]: new group: name=rtkit, GID=497
<86>Jun 11 11:50:05 useradd[2569]: new user: name=rtkit, UID=499,
GID=497, home=/proc, shell=/sbin/nologin
<86>Jun 11 11:50:28 groupadd[2662]: group added to /etc/group:
name=avahi-autoipd, GID=170
<86>Jun 11 11:50:29 groupadd[2662]: group added to /etc/gshadow:
name=avahi-autoipd
<86>Jun 11 11:50:29 groupadd[2662]: new group: name=avahi-autoipd,
GID=170
```

The cp Command

The `cp` command copies file from source to destination preserving same mode.

```
# cp -p fileA fileB
```

You will be prompted before overwriting any file.

```
# cp  ifileA fileB
```

The mv Command

Rename fileA to fileB. The `-i` option prompts before overwrite. Asks for confirmation if exist already.

```
# mv -ifileA fileB
```

The cat Command

The `cat` command used to view multiple file at the same time.

```
# cat fileA fileB
```

You can combine more and less command with cat command to view file content if that does not fit in single screen / page.

```
# cat install.log | less

# cat install.log | more
```

The cd Command (change directory)

The `cd` command (change directory) takes you to the `fileA` directory.

```
# cd /fileA
```

The pwd Command (print working directory)

The `pwd` command return with present working directory.

```
# pwd
/root
```

The sort Command

Sorting lines of text files in ascending order. You can use the `-r` options to sort in descending order.

```
#sort fileA.txt

#sort -r fileA.txt
```

The vi Command

`vi` is the most popular text editor available in most of the UNIX-like OS. Following examples open file in read only with `-R` option. Press: q to quit from the `vi` window.

```
# vi -R /etc/shadows
```

The ssh Command (Secure Shell)

The `ssh` command is used to login into remote host. For example, the following `ssh` command connects to remote host (192.168.1.1) using user as narad.

```
# ssh narad@192.168.1.1
```

The ftp and sftp Commands

The `ftp` or `sftp` commands are used to connect to remote `host.ftp` is (file transfer protocol) and `sftp` is (secure file transfer protocol). For example the following command connects to ftp host (192.168.50.2).

```
# ftp 192.168.50.2
# sftp 192.168.50.2
```

Putting multiple files in remote host with `mput` similarly you can do `mget` to download multiple files from remote host.

```
# ftp>mput *.txt

# ftp>mget *.txt
```

The service Command

The `service` command calls the script located at `/etc/init.d/` directory and executes the script. There are two ways to start any service. For example, you start the service called `httpd` with the following service command.

```
# service httpd start
```

or

```
# /etc/init.d/httpd start
```

The free command

The `free` command shows free, total and swap memory information in bytes.

```
# free
total       used       free     shared buffers   cached
Mem:      1922428    1848872    73556    0       243180    1180768
-/+ buffers/cache:      424924    1497504
Swap:       4128760     105628    4023132
```

The `free` command with `-t` options shows total memory used and available memory in bytes.

```
# free -t
total        used        free      shared    buffers      cached
Mem:         1922428     1841648        80780           0      243184
1182600
-/+ buffers/cache:      415864    1506564
Swap:       4128760     105628    4023132
Total:      6051188    1947276    4103912
```

The top Command

The `top` command displays processor activity of the system and also displays tasks managed by kernel in real-time. It shows details of the processor and memory that are being used. Use the `top` command with `u` option. This displays specific user process details as follows. Press `O` (uppercase letter) to sort as desired by you. Press `q` to quit from top screen.

```
# top -u test123
top - 17:03:29 up 13 days, 21:39,  2 users,  load average: 0.07,
0.19, 0.09
Tasks: 196 total,    1 running, 195 sleeping,    0 stopped,    0
zombie
Cpu(s):   0.2%us,   0.3%sy,   0.0%ni, 99.5%id,   0.0%wa,   0.0%hi,
0.0%si,  0.0%st
Mem:   1922428k total,  1842152k used,    80276k free,   243240k
buffers
Swap:  4128760k total,   105628k used,  4023132k free,  1182912k
cached
  PID USER        PR   NI   VIRT   RES   SHR S  %CPU  %MEM     TIME+
COMMAND
 1437 test123   20   0  100m 2384 1384 S  0.0  0.1   0:00.26 sshd
 1438 test123   20   0  100m 1992  984 S  0.0  0.1   0:00.01 sshd
 1439 test123   20   0 57680 2296 1632 S  0.0  0.1   0:00.06 sftp-
server
 1454 test123   20   0  105m 1872 1408 S  0.0  0.1   0:00.04 bash
```

The tar Command

The `tar` command is used to compress files and folders in Linux. For example the following command creates an archive for /home directory with file name as archive-name.tar.

```
# tar -cvf archive-name.tar /home
```

To extract tar archive file, use the option as follows.

```
# tar  -xvf  archive-name.tar
```

The grep Command

The `grep` command searches for a given string in a file. Only test123 user displays from `/etc/passwd` file. You can use the `-i` option for ignoring case sensitive.

```
# grep test123 /etc/passwd
test123:x:500:500::/home/test123:/bin/bash
```

The find Command

The `find` command used to search files, strings and directories.

test123 word in '/' partition and return the output.

```
# find / -name test123
/var/spool/mail/test123
/home/test123
/home/test123/oofem/oofem-2.1/targets/home/test123
/home/test123/oofem/oofem-2.0/targets/home/test123
/home/test123/oofem/oofem-1.9/targets/home/test123
/home/test123/oofem20/oofem-2.0/targets/home/test123
```

The lsof Command

The `lsof` command means list of all open files. Following is a list of the `lsof` commands to open files by user test123.

```
# lsof -u test123
COMMAND    PID    USER   FD TYPE   DEVICE SIZE/OFF     NODE NAME
sshd      1437 test123  cwd DIR   253,0     4096        2   /
sshd      1437 test123  rtd DIR   253,0     4096        2   /
sshd         1437  test123   txt  REG     253,0     546680     149215
/usr/sbin/sshd
sshd        1437  test123   DEL  REG       0,4                5301422
/dev/zero
sshd         1437  test123   mem  REG     253,0      18592     530764
/lib64/security/pam_limits.so
sshd         1437  test123   mem  REG     253,0      10224     530762
/lib64/security/pam_keyinit.so
sshd         1437  test123   mem  REG     253,0      10240     530767
/lib64/security/pam_loginuid.so
sshd         1437  test123   mem  REG     253,0      18664     530779
/lib64/security/pam_selinux.so
sshd         1437  test123   mem  REG     253,0      41088     138057
/usr/lib64/libcrack.so.2.8.1
```

The last command

With the `last` command you can watch the user's activity in the system. This command can execute normal user also. It displays complete user's info such as terminal, time, date, system, re-booter, boot, and kernel version. It is a useful command for troubleshooting.

```
# last
root     pts/1  10.208.33.31     Thu Jun 12 11:41 - 14:34  (02:52)
root     pts/3  param.shavak Wed Jun 11 17:10 - 17:53  (00:43)
root     pts/3  param.shavak Wed Jun 11 16:58 - 16:58  (00:00)
root     pts/1  param.shavak Wed Jun 11 16:50 - 19:20  (02:29)
```

```
root     pts/7  param.shavak Wed Jun 11 15:18 - 19:01  (03:42)
```

You can use `last` with username to know for specific user's activity as follows.

```
# last test123
test123  pts/2  param.shavak Fri Jul  4 08:24 - 11:24  (02:59)
test123  pts/15 localhost:14.0   Thu Jul  3 16:55 - 16:55  (00:00)
test123  pts/8  localhost:14.0   Thu Jul  3 16:55 - 16:55  (00:00)
test123  pts/6  localhost:14.0   Thu Jul  3 16:55 - 15:21  (22:25)
test123  pts/4  10.208.32.221    Thu Jul  3 16:47 - 14:49  (22:01)
```

The ps command

The `ps` command displays the details of the processes running in the system. Following example, show test123 only.

```
# ps -ef |grep test123
root      1429 2402  0 15:44 ?     00:00:00 sshd: test123 [priv]
root      1433 2402  0 15:44 ?     00:00:00 sshd: test123 [priv]
test123   1437 1429  0 15:44 ?     00:00:00 sshd: test123@pts/0
test123   1438 1433  0 15:44 ?     00:00:00 sshd: test123@notty
```

The kill command

Use the `kill` command to terminate a process. First find the process id using the `ps` command as follows and kill the process with `kill -9` command.

```
# ps -ef |grep test123
root      1429  2402  0 15:44 ?    00:00:00 sshd: test123 [priv]
root      1433  2402  0 15:44 ?    00:00:00 sshd: test123 [priv]
test123   1437  1429  0 15:44 ?    00:00:00 sshd: test123@pts/0
test123   1438  1433  0 15:44 ?    00:00:00 sshd: test123@notty

# kill- 9 1437
```

The rm command

The `rm` command is used to remove or delete a file without prompting for confirmation.

```
# rm filename
```

Use the `-i` option to get confirmation before removing it. Using the options `-r` and `-f` removes the file forcefully without confirmation.

Note: Do not use this command until or unless you are sure what you are doing.

```
# rm -i test.txt
```

```
rm: remove regular file `test.txt'?
```

The mkdir command

The `mkdir` command is used to create directories under Linux.

```
# mkdir directoryname
```

File Commands

- ls – directory listing
- ls -al – formatted listing with hidden file
- cd dir - change directory to dir
- cd – change to home
- pwd – show current directory
- mkdirdir – create a directory dir
- rm file – delete file
- rm -r dir – delete directory dir
- rm -f file – force remove file
- rm -rfdir – force remove directory dir
- cp file1 file2 – copy file1 to file2
- cp -r dir1 dir2 – copy dir1 to dir2; create dir2 if it doesn't exist
- mv file1 file2 – rename or move file1 to file2 if file2 is an existing directory, moves file1 into directory file2
- ln -s file link – create a symbolic link link to a file
- touch file – create or update file
- cat > file – places standard input into file
- more file – output the contents of file
- head file – output the first 10 lines of file
- tail file– output the last 10 lines of file

- tail -f file – output the contents of file as it grows, starting with the last 10 lines

Process Management

- ps – display your currently active processes
- top – display all running processes
- kill pid – kill process id pid
- killallproc – kill all processes named proc *
- bg – lists stopped or background jobs; resume a stopped job in the background
- fg – brings the most recent job to foreground

ssh

- ssh user@host – connect to host as user
- ssh -p port user@host – connect to host on port
- ssh-copy-id user@host – add your key to host for user to enable a keyed or password less login

Searching

- grep pattern files – search for pattern in files
- grep -r pattern dir – search recursively for pattern in dir
- command | grep pattern – search for pattern in the output of command
- locate file – find all instances of file

System Info

- date – show the current date and time
- cal – show this month's calendar
- uptime – show current uptime
- w – display who is online
- whoami– who you are logged in as

- finger user– display information about user
- uname -a – show kernel information
- cat /proc/cpuinfo – cpu information
- cat /proc/meminfo – memory information
- man command– show the manual for command
- df – show disk usage
- du – show directory space usage
- free – show memory and swap usage
- whereis app – show possible locations of app
- which app – show which app will be run by default

Compression

- tar cf file.tar files – create a tar named file.tar containing files
- tar xf file.tar – extract the files from file.tar
- tar czf file.tar.gz files – create a tar with Gzip compression
- tar xzf file.tar.gz – extract a tar using Gzip
- tar cjf file.tar.bz2 – create a tar with Bzip2 compression
- tar xjf file.tar.bz2 – extract a tar using Bzip2
- gzip file – compresses file and renames it to file.gz

gzip -d file.gz – decompresses file.gz back to file

Network

- ping host – ping host and output results
- whois domain – get whois information for domain
- dig domain – get DNS information for domain
- dig -x host– reverse lookup host
- wget file – download file

- wget -c file – continue a stopped download

Shortcuts

- Ctrl+C – halts the current command
- Ctrl+Z– stops the current command, resume with fg in the foreground or bg in the background
- Ctrl+D – log out of current session, similar to exit
- Ctrl+W – erases one word in the current line
- Ctrl+U – erases the whole line
- Ctrl+R – type to bring up a recent command
- !! - repeats the last command
- exit – log out of current session

Chapter 6

Application Usage Manual

This chapter describes the methods to use the application.

Intel® Parallel Studio XE 2015 Cluster Edition Usage

Introduction

Intel® Parallel Studio XE Cluster Edition provides a software tools environment for hybrid parallel programming (message passing and threading).

Intel® Parallel Studio XE Cluster Edition supports hybrid parallel programming application development using Intel® MPI Library with optimized parallel libraries, performance analysis, and benchmarks. Intel® Parallel Studio XE Cluster Edition saves software developers time and improves performance on distributed computing systems.

Intel® Parallel Studio XE 2015 Cluster Edition consists of the following components:

- Intel® Advisor XE
- Intel® C++ Compiler XE
- Intel® Fortran Compiler XE
- Intel® Inspector XE
- Intel® Integrated Performance Primitives (Intel® IPP)
- Intel® Math Kernel Library (Intel® MKL)
- Intel® MPI Benchmarks
- Intel® MPI Library
- Intel® Threading Building Blocks (Intel® TBB)
- Intel® Trace Analyzer and Collector

- Intel® VTuneTM Amplifier XE
- GNU* Project Debugger (GDB*) (GPL licensed)

Intel® Parallel Studio XE Cluster Edition for Linux* OS accelerates parallel software development on cluster systems based on Intel® 64 architectures, as well as Intel® Many Integrated Core Architecture (Intel® MIC Architecture) on Linux* OS. For Intel® MIC Architecture, only Intel® Xeon Phi™ coprocessor (codename: Knights Corner) is supported.

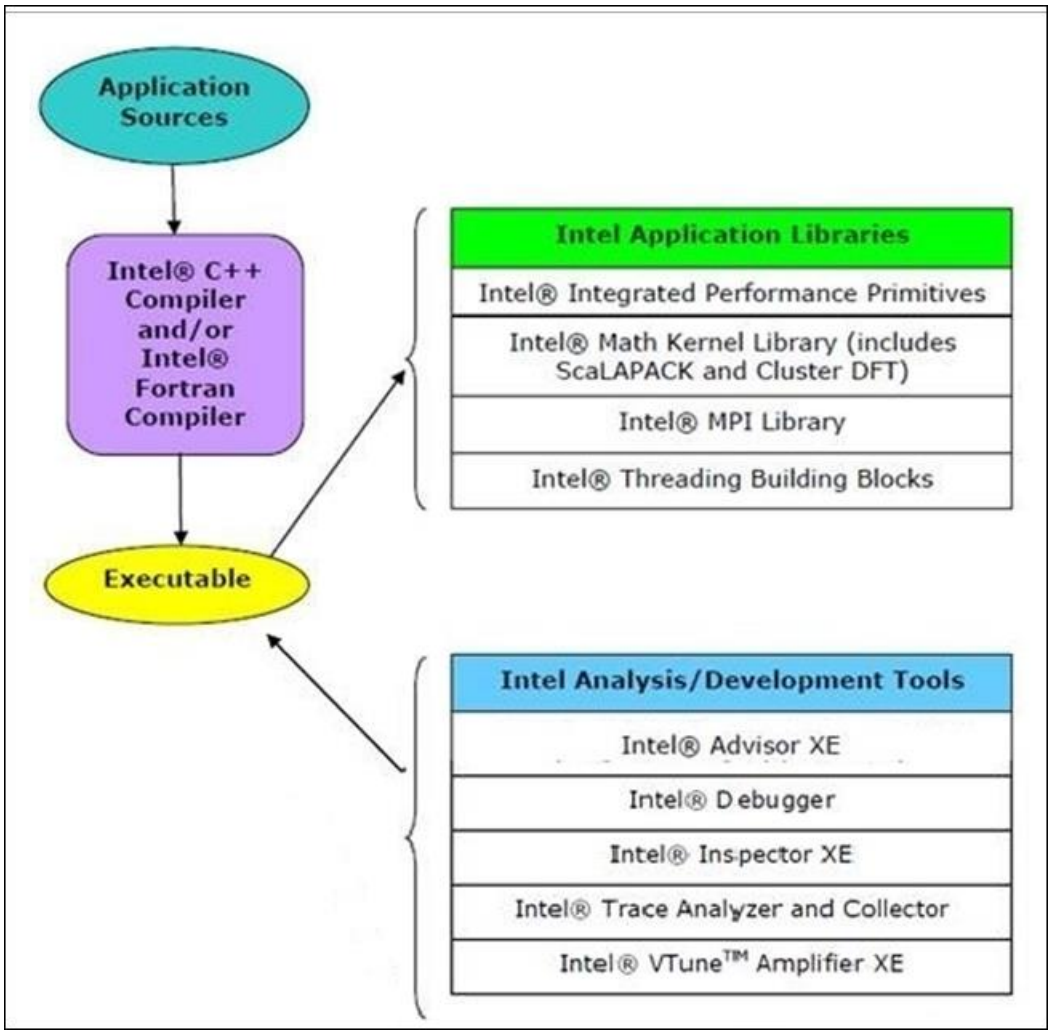


Figure 14 - Software Architecture of the Intel® Cluster Studio XE

Intel® Advisor XE

Intel® Advisor XE 2015 guides developers to add parallelism to their existing C/C++ or Fortran programs.

Intel Advisor XE provides a standalone graphical user interface (GUI) as well as a command line interface (CLI). You can use the Intel Advisor XE to:

- Find the most time-consuming serial code regions in your program.
- Insert Intel Advisor XE annotations to identify these as possible parallel code regions.
- Predict the approximate parallel performance characteristics of the proposed parallel code regions.
- Check for data sharing problems that could prevent the application from working correctly when parallelized.

More Information about this application you can find from following link:

https://software.intel.com/sites/default/files/article/251103/release-notes-advisor-xe-lin-rtm.pdf

or

```
<install-
dir>/advisor_xe/documentation/en/documentation_advisor_xe.htm
```

Note:<install-dir> used in this section refers to /opt/intel

Intel® C++ Compiler XE and Intel® Fortran Compiler XE

Compilation and Execution of C, C++, and Fortran Programs

To compile C, C++ and/or Fortran programs using INTEL compilers, do the following:

Ensure that the compiler path should be in your PATH variable, the INTEL compiler library directories should be in LD_LIBRARY_PATH variable and the required header files should be in INCLUDE environment variable.

This setting is enabled by default for all user accounts on PARAM Shavak.

This can be done manually by the following command:

```
source       /opt/intel/composer_xe_2015.0.090/bin/compilervars.sh
intel64
```

Compiling and running

Intel C, C++, Fortran compiler is installed on `/opt/intel/`

Compile Serial Application (Example):

```
ifort  <program_name.f90>  -o  <output_file_name>(For  the  Fortran
program)
icc <program_name.c>  -o <output_file_name> (For the C program)
icpc <program_name.c>  -o <output_file_name> (For the C++ program)
```

It will create a.out default executable file, if you not provide any <output_file_name>.

Compilation and Execution of OpenMP Application (C, C++ and Fortran Program)

Compiling and running

Intel C, C++, Fortran compiler is installed on `/opt/intel/`

Compile OpenMP Application (Example):

```
ifort   -openmp  <program_name.f90>   -o  <output_file_name>(For the
Fortran program f77,f90)
icc  -openmp <program_name.c>  -o <output_file_name>(For the C program)
icpc  -openmp  <program_name.c>   -o  <output_file_name>(For  the  C++
program)
```

It will create default executable file a.out, if you not provide any <`output_file_name`>.

For More Information See:

For C/C++:

<*install-dir*>/composerxe/Documentation/en_US/beginusing_lc.htm

For Fortran:

<*install-dir*>/composerxe/Documentation/en_US/beginusing_lf.htm

Intel® Inspector XE

Intel® Inspector 2015 for Systems helps developers identify and resolve memory and threading correctness issues in their C, C++ and Fortran programs.

Intel Inspector XE is a static and dynamic error checking tool for developing multithreaded applications on Windows* or Linux* operating systems. Intel Inspector XE maximizes code quality and reliability by quickly detecting memory, threading, and source code security errors during the development cycle. Intel Inspector XE is an easy, comprehensive solution that delivers rapid results in isolating memory and multithreading errors.

Intel Inspector XE has a standalone graphical user interface (GUI) as well as a command line interface (CLI).

More information can be found at:

https://software.intel.com/sites/default/files/managed/3c/d2/Release_Notes_Inspector_Linux_0.pdf

Or

```
<install-
dir>/inspector_xe/documentation/en/documentation_inspector.htm
```

Intel® Integrated Performance Primitives (Intel® IPP)

Intel® Integrated Performance Primitives (Intel® IPP) to improve performance of multimedia, enterprise data, embedded, communications, and scientific/technical applications. The primitives are a common interface for thousands of commonly used algorithms. Using these primitives enables you to automatically tune your application to many generations of processors without changes in your application.

Intel IPP library provides high performance implementations of signal, image, and data processing functions for several hardware/instruction set generations. Code written with Intel IPP automatically takes advantage of available CPU capabilities. This can provide tremendous development and maintenance savings. You can write programs with one optimized execution path, avoiding the alternative of multiple paths (Intel® Streaming SIMD Extensions 2, Supplemental Streaming SIMD Extensions 3, Intel® Advanced Vector Extensions , etc.) to achieve optimal performance across multiple generations of processors.

The goal of the Intel IPP software is to provide algorithmic building blocks with

- a simple "primitive" C interface and data structures to enhance usability and portability
- faster time-to-market
- scalability with Intel® hardware

More information can be found at:

https://software.intel.com/en-us/intel-ipp

Or

```
<install-
dir>/composerxe/Documentation/en_US/ipp/ipp_documentati
on.htm
```

Intel® Math Kernel Library (Intel® MKL)

Intel MKL is installed as part of the Intel® Parallel Studio 2015, under the mkl sub-directory. The Intel MKL installation directory is further referred to as

<*install-dir*>/composer_xe_2015_version/mkl, where <*install-dir*> is the Intel Parallel Studio XE 2015 Cluster Edition installation directory.

Application tuning with optimized mathematical library functions from Intel® Math Kernel Library (Intel® MKL) that includes ScaLAPACK* solvers and Cluster DFTs (Discrete FourierTransforms). Intel® MKL for Linux* OS support Intel® Many Integrated Core Architecture (Intel® MIC Architecture).

A script located in the < install directory>\mkl\bin\ directory can be used to set the MKLROOT, LIB, INCLUDE and any required system-specific environment variables to point to the appropriate MKL library directories. Use of this script is an optional, but convenient, means by which to configure your development system for compiling and linking with intel® MKL.

You run the script with a single input that specifies the primary processor architecture of interest.

For example, *mklvars.sh intel64* configures the environment variables for compiling and linking a 64-bit Intel® MKL application.

The documentation index page that links to various documents associated with Intel® MKL is located at:

```
<install-dir>/composer_xe_2015_version/Documentation/en_US/mkl.
```

Intel® MPI Benchmarks

- Provide a concise set of benchmarks targeted at measuring the most important MPI functions.
- Set forth a precise benchmark methodology.

- Report bare timings rather than provide interpretation of the measured results. Show throughput values if and only if these values are well defined.

For More Information See:

http://www.hpc.ut.ee/dokumendid/ips_xe_2015/imb_latest/doc/IMB_Users_Guide.pdf

Or

```
<install-dir>/imb_latest/doc/IMB_Users_Guide.htm
```

Intel® Threading Building Blocks

Intel® Threading Building Blocks (Intel® TBB) is a widely used C and C++ library for creating high performance, scalable parallel applications.

It is designed to promote scalable data parallel programming. Additionally, it fully supports nested parallelism, so you can build larger parallel components from smaller parallel components. To use the library, you specify tasks, not threads, and let the library map tasks onto threads in an efficient manner.

Features:

Enhance Productivity and Reliability - Rich set of components to efficiently implement higher-level, task-based parallelism

Gain Performance Advantage- Future-proof applications to tap multicore and many-core processing power

Portability and Compatibility –Open source and commercial licensing supports Windows*, Linux*, OS X*, and Android* (additional with open source).It's compatible with multiple compilers and Intel compatible processors including Intel® Atom™, Core™, Xeon® processors, and Intel® Xeon Phi™ coprocessors.

For More Information See:

```
<install-
dir>/composerxe/Documentation/en_US/tbb/tbb_documentation.htm
```

Intel® Trace Analyzer and Collector

Intel® Trace Collector provides event - based tracing in cluster applications through an instrumentation library that ensures low overhead in execution. The trace information provides performance data, statistics, and multi-threaded events on Intel® 64 and Intel® Many Integrated Core Architecture (Intel® MIC Architecture).

Intel® Trace Analyzer provides visual analysis of application activities gathered by the Intel Trace Collector.

Intel® Trace Analyzer and Collector enables you to understand MPI application behavior, quickly find bottlenecks and achieve high performance for parallel cluster applications.

Use this tool to do the following:

- Evaluate profiling statistics and load balancing
- Learn about communication patterns, parameters, and performance data
- Identify communication hotspots
- Decrease time to solution and increase application efficiency

For More Information See:

```
<install-dir>/itac_latest/Doc_Index.htm
```

Intel® VTune™ Amplifier XE

The Intel® VTune™ Amplifier XE 2015 provides an integrated performance analysis and tuning environment with graphical user interface that helps you analyze code performance on systems with IA-32 or Intel® 64 architectures.

Intel® VTune™ Amplifier XE analyzes complex code and helps us rapidly identify bottlenecks. By using it and other Intel® Software Development Tools, we were able to improve our code performance.

Use VTune Amplifier to locate or determine the following:

- The most time-consuming (hot) functions in your application and/or on the whole system
- Sections of code that do not effectively utilize available processor time
- The best sections of code to optimize for sequential performance and for threaded performance
- Synchronization objects that affect the application performance
- Whether, where, and why your application spends time on input/output operations
- The performance impact of different synchronization methods, different numbers of threads, or different algorithms

- Thread activity and transitions
- Hardware-related bottlenecks in your code

For More Information See:

https://software.intel.com/sites/default/files/managed/23/42/release-notes-amplifier-xe-linux.pdf

Or

```
<install-
dir>/vtune_amplifier_xe/documentation/en/documentation_
amplifier.htm
```

Intel® MPI Library

Intel MPI Library is installed as part of the Intel® Parallel Studio XE 2015 Cluster Edition.

Compiling and Linking with Intel® MPI Library

To compile and link an MPI program using the Intel MPI Library, do the following:

Ensure that the underlying compiler, MPI libraries, and the related software appears in your PATH environment variable. If you are using Intel® compilers, ensure that the compiler library directories appear in the LD_LIBRARY_PATH environment variable. For example, when using the Intel® C++ Compiler, Intel® Fortran Compiler, and the Intel® MPI Library, run the appropriate set-up scripts.

This setting is enabled by default for all user accounts on PARAM Shavak.

This can be done manually by using following commands:

```
<install_dir>/composer_xe_<version>/bin/compilervars.[c]sh intel64
<install_dir>/impi/<version>/bin64/mpivars.[c]sh
```

In these commands, version is the installation package number.

Compile your MPI program using an appropriate compiler command with the mpi prefix as shown in the table below. For example, for C program you can use the mpiicc command as follows:

```
mpiicc <name of C program>  -o  <name of output file>
```

For example, for C++program you can use the mpiicc command as follows:

```
mpiicpc <name of C++ program>  -o  <name of output file>
```

For example, for Fortran program you can use the mpiicc command as follows:

```
mpiifort <name of Fortran program>  -o  <name of output file>
```

Running an MPI Program Using Intel® MPI Library

Use the mpirun/mpiexec.hydra command to launch MPI programs, for example:

```
mpirun -n <# of processes> ./myprog.exe
```

OR

```
mpiexec.hydra  -n <# of processes> ./myprog.exe
```

For More Information See:

```
<install-dir>/impi_latest/Doc_Index.html
```

Using Intel® MPI Library on Intel® Xeon Phi™ Coprocessor

Intel® MPI Library for the Intel® Many Integrated Core Architecture (Intel® MIC Architecture) supports only the Intel® Xeon Phi™ coprocessor (codename: Knights Corner).

Building an MPI Application

To build an MPI application for the host node and the Intel® Xeon Phi™ coprocessor, follow these steps:

Establish the environment settings for the compiler and for the Intel® MPI Library:

```
(host)$ . <install-dir>/compiler/bin/compilervars.sh intel64
(host)$ . <install-dir>/impi/intel64/bin/mpivars.sh
```

Build your application for Intel® Xeon Phi™ coprocessor:

```
(host)$ mpiicc -mmicmyprog.c -o myprog.mic
```

Build your application for Intel® 64 architecture:

```
(host)$ mpiiccmyprog.c -o myprog
```

Running an MPI Application

To run an MPI application on the host node and the Intel® Xeon Phi™ coprocessor, do the following:

Ensure that NFS is properly set up between the hosts and the Intel® Xeon Phi™ coprocessor(s). For information on how to set up NFS on the Intel® Xeon Phi™ coprocessor(s), visit the Intel® Xeon Phi™ coprocessor developer community at http://software.intel.com/en-us/mic-developer.

Use the I_MPI_MIC_POSTFIX environment variable to append the .mic postfix extension when running on the Intel® Xeon Phi™ coprocessor.

```
(host)$ export I_MPI_MIC_POSTFIX=.mic
```

1. Make sure your ~/mpd.hosts file contains the machine names of your Intel® Xeon® host processors and the Intel® Xeon Phi™ coprocessor(s). For example:

```
(host)$ cat ~/mpd.hosts
clusternode1
clusternode1-mic0
```

2. Launch the executable file from the host.

```
(host)$ export I_MPI_MIC=on
(host)$ export mpirun -n 4 -hostfile ~/mpd.hosts ./myprog
```

NOTE: You can also use the –config file and -machine file options.
To run the application on Intel® Xeon Phi™ coprocessor only, follow the steps described above and ensure that mpd.hosts contains only the Intel® Xeon Phi™ coprocessor name.

References:

http://software.intel.com

Torque Resource Manager

Torque Resource Manager provides control over batch jobs and distributed computing resources. It is an advanced open source product based on the original PBS project and incorporates the best of both community and professional development. It incorporates significant advances in the areas of scalability, reliability and functionality and is currently in use at tens of thousands of leading government, academic and commercial sites throughout the world. TORQUE may be freely used, modified, and distributed under the constraints of the included license.

Job submission through torque

PBS can be used to submit Serial and Parallel JOBS. For Serial and OpenMPjobs there is no need to run MPDBOOT. But for MPI Jobs it is mandatory to run MPDBOOT before submitting PBS script.

Advantages of using PBS:

- Users need not worry about which nodes are free and which are already utilized for submitting their jobs.
- PBS finds out whether the resources needed by the different jobs are available or not and on which nodes.
- It submits the jobs accordingly or places them in the queue depending upon the resource availability.
- It takes care of Load Sharing on its own.

Some useful PBS commands

Commands	Description
qsub<pbs script>	To submit PBS script.
qstat –a	To see all the jobs submitted by you. It shows you the name of the Job along with a unique Job ID.
qstat -n1 <Job ID>	It shows you on which nodes your Job having the particular Job ID is running.
pbsnodes –a	To see how many nodes are available through PBS.
pbsnodes -l free	It shows you the nodes that are not fully utilized or are free.

Serial C/C++/Fortran Program

Compiling and submitting through Torque:

Intel® C, C++, Fortran compiler is installed on /opt/intel

Compile Serial Application (Example)

- ifort program_name.f90 (For the Fortran program)
- icc program_name.c (For the C program)
- icpc program_name.C (For the C++ program)

It creates `a.out` default executable file.

Job submission through torque:

Create a pbs script as:

```
vipbs_script /*SCRIPT NAME IS USER DEPENDENT, HERE IT IS
pbs_script*/
#!/bin/bash
#PBS N
<filename> /*THE ERROR AND OUTPUT OF THE PROGRAM WILL
BE STORED IN <FILENAME>*/
## PATH OF THE DIRECTORY WHERE WE ARE RUNNING THE PROGRAME.
cd<absolute path of the Executable>
## THE PATH OF THE EXECUTABLE
<absolute path of the executable>/a.out
```

Save the `pbs_script` file.

Submit jobs as:

```
qsub pbs_script
```

OpenMP C/C++/Fortran Program

Compiling and submitting through Torque

Intel® C, C++,Fortran compiler is installed on /opt/intel

Compile OpenMP Application (Example):

- ifort –openmp program_name.f90 (For the Fortran program f77, f90)
- icc –openmp program_name.c (For the C program)
- icpc –openmp program_name.C (For the C++ program)

It creates `a.out` default executable file

Job submission through torque:

1. Create a pbs script as:

```
vipbs_script /*SCRIPT NAME IS USER DEPENDENT, HERE IT IS
pbs_script*/
#!/bin/bash
#PBS N
<FILENAME> /*THE ERROR AND OUTPUT OF THE PROGRAM
WILL BE STORED IN <FILENAME>*/
#### THE ABSOLUTE PATH OF THE EXECUTABLE
cd<absolute path of the Executable>

### TO DEFINE THE NUMBER OF THREADS
export OMP_NUM_THREADS=n /*where n=no. of threads*/
##THE ABSOLUTE PATH OF THE EXECUTABLE WITH THE EXECUTABLE
NAME
<absolute path of the Executable>/a.out
```

absolute path of the executable with executable name

Save the `pbs_script` file.

2. Submit jobs as:

```
qsub pbs_script
```

MPI C/Fortran Program Compiling and submitting through Torque

Intel® C, C++, Fortran MPI compiler is installed on /opt/intel

Compile MPI Application (Example):

- mpif77 program_nam.f77 (For the Fortran F77 program)
- mpif90 program_nam.f90 (For the Fortran F90 program)
- mpiifort program_nam.f (For the both Fortran F77 and F90 program)
- mpicc program_nam.c (For the C program)
- mpiicpc program_nam.C (For the C++ program)

It creates `a.out` default executable file

Submitting MPI C/FORTRAN program through scheduler:

3. Create a pbs script as:

```
vipbs_script  /*SCRIPT  NAME  IS  USER  DEPENDENT,  HERE  IT  IS
pbs_script*/
#!/bin/bash
#PBS N
<FILENAME> /*THE ERROR AND OUTPUT OF THE PROGRAM
WILL BE STORED IN <FILENAME>*/
#PBS l
nodes=4:ppn=4 /*Total 4*4=16 instances to be run*/
#PBS m
abe /*Email me when job 'a'borts,'b'egins and 'e'nds.*/
cd<absolute path of the Executable>

/rhome/intel/impi/3.2.1.009/bin64/mpiexecmachinefile
$PBS_NODEFILE -n
16
<absolute path of the executable>/a.out 1>prl1.log2 2>prl1.err2
/*Total 16 instances (n 16)of the executable to be run.*/
```

Save the `pbs_script` file.

**Description: It run 16 Instances of the Program on 4 Nodes using 4

processors specified by the line "#PBS l

nodes=4:ppn=4" in the pbs_script file.

1>prl1.log2 saves the output in prl1.log2

2>prl1.err2 saves the error in prl1.err2

4. Submit jobs as:

```
qsub pbs_script
```

Ganglia Monitoring System

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and grids. It is based on a hierarchical design targeted at federations of clusters. It leverages widely used technologies such as XML for data representation, XDR for compact, portable data transport, and RRD tool for data storage and visualization.

It uses carefully engineered data structures and algorithms to achieve very low per-node overheads and high concurrency. The implementation is robust, has been ported to an extensive set of operating systems and processor architectures, and is currently in use on thousands of clusters around the world. It has been used to link clusters across university campuses and around the world and can scale to handle clusters with 2000 nodes.

Using the ganglia

System Health

The webpages available from this link (hostname or ip address/ganglia) provide a graphical interface to live cluster information provided by Ganglia monitors running on each cluster node. The monitors gather values for various metrics such as CPU load, free memory, disk usage, network I/O, operating system version, etc. These metrics are sent through the private cluster network and are used by the front-end node to generate the historical graphs.

In addition to metric parameters, a heartbeat message from each node is collected by the Ganglia monitors. When a number of heartbeats from any node are missed, this web page will declare it dead. These dead nodes often have problems which require additional attention, and are marked with the Skull-and-Crossbones icon, or a red background.

For accessing system report which is generated by Ganglia you have to type <hostname or ip address>/ganglia in the browser as shown in the following figure.

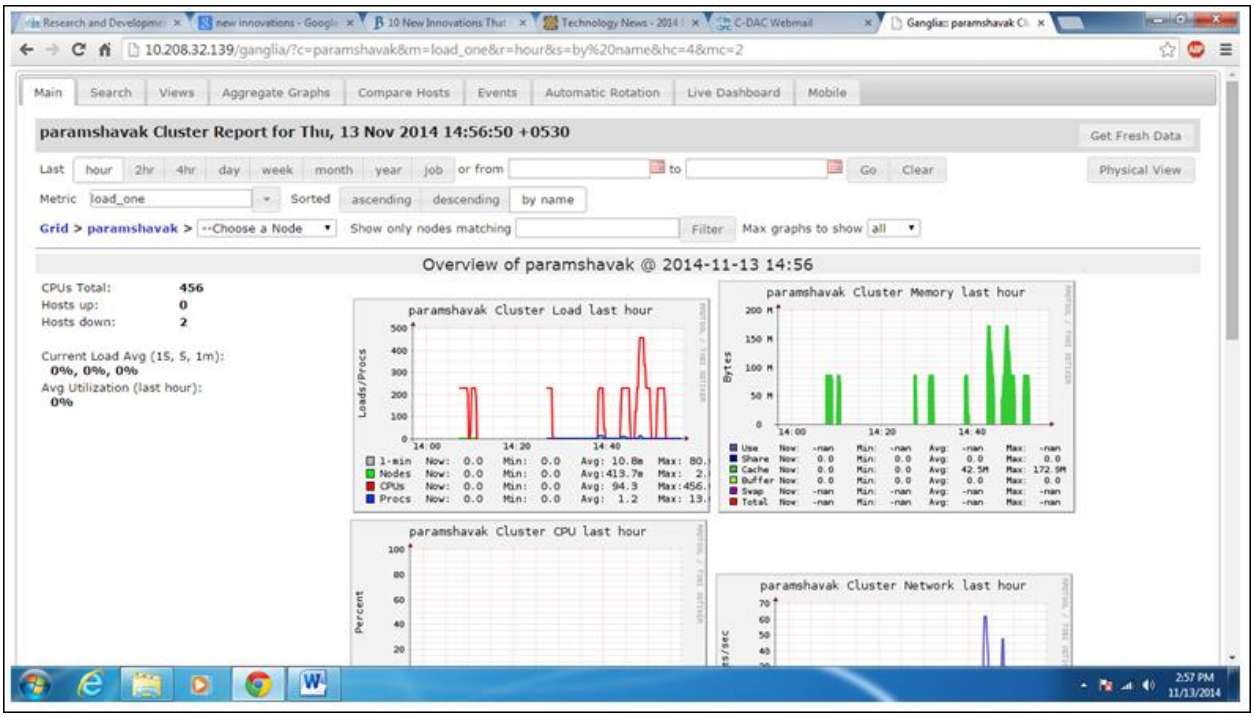


Figure 15 - PARAM Shavak load_ one last hour sorted by name

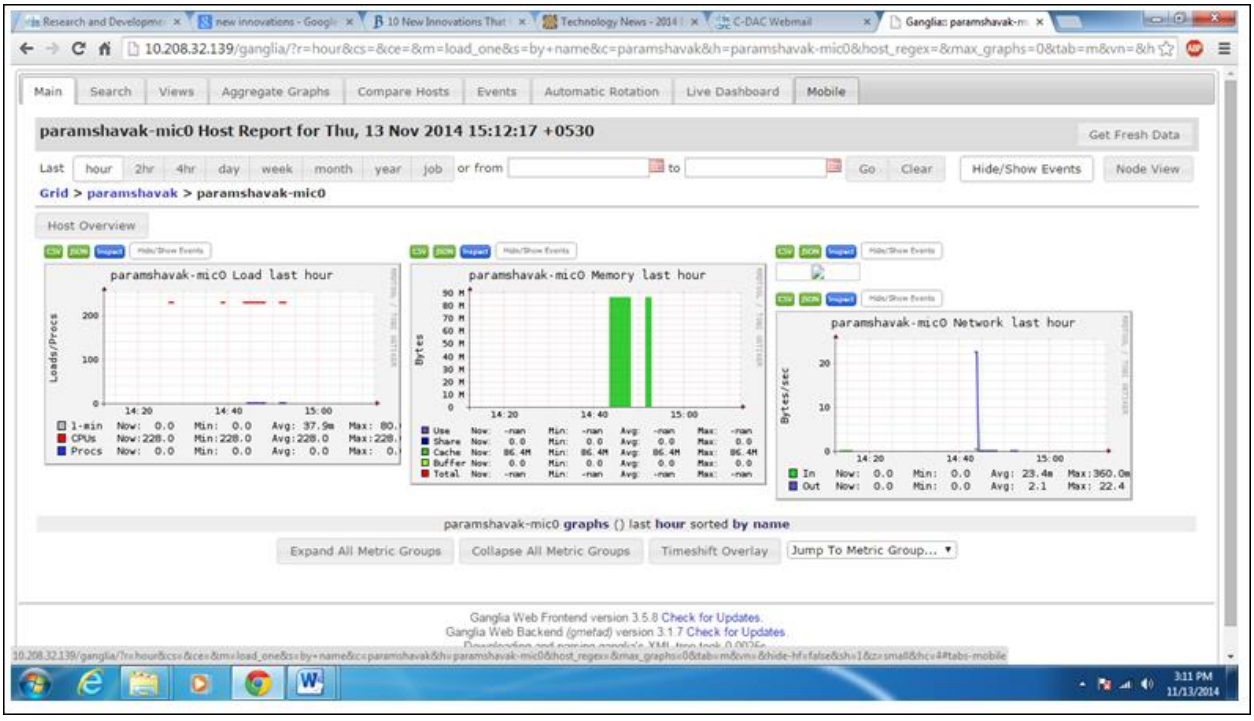


Figure 16 - ParamShavak-mic0 graphs last hour sorted by name

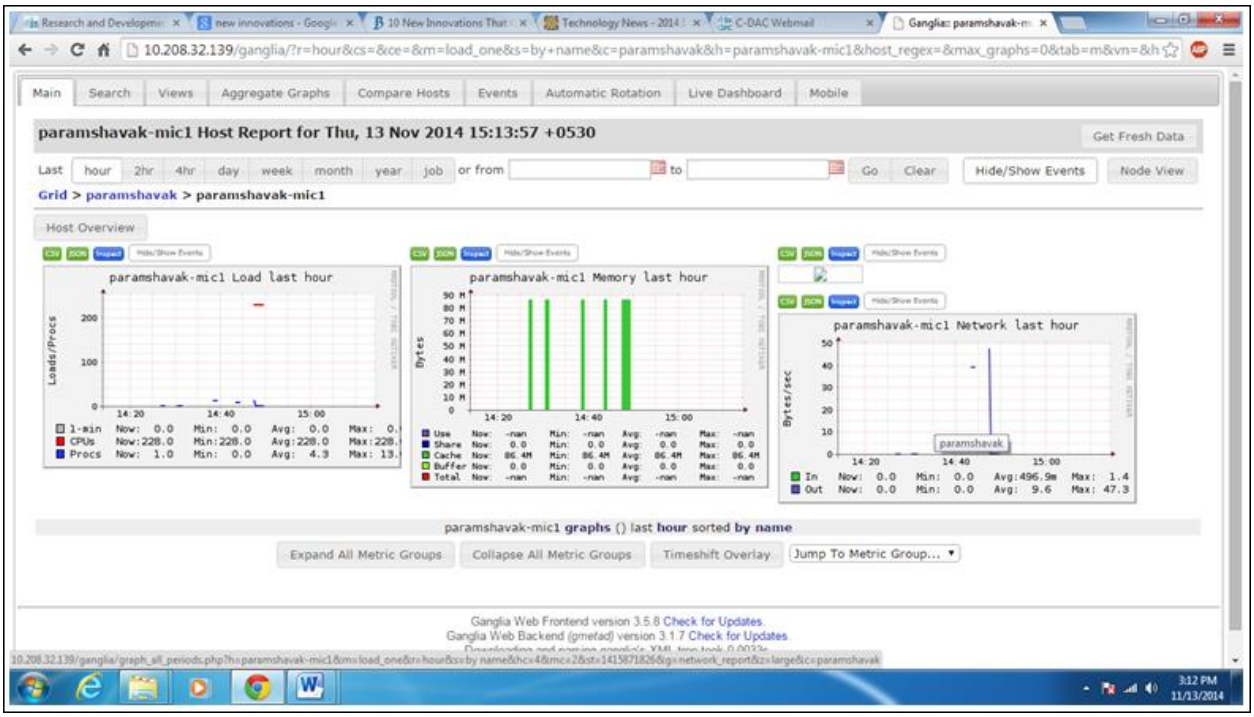


Figure 17 - ParamShavak-mic1 graphs last hour sorted by name

Figure 18 - The Ganglia Main Page

ONAMA

ONAMA is an integrated software and hardware solution for future technocrats/engineers, providing them a platform for a quantum leap to gain an in-depth understanding of High Performance Computing (HPC). This is an integrated package comprising of a well-selected set of parallel as well as serial applications of scientific and engineering domain and is useful across various engineering disciplines.

ONAMA provides a great opportunity for networking amongst industry experts, professors, faculties, research scholars and students to work in multi-disciplinary areas of engineering and to incorporate the latest technologies in high-end computing.

Onama Execution Model Interface

Figure 19 - ONAMA Introduction

After you click on the icon appearing on the desktop, ONAMA Execution Model interface appears. Initially, it locates the executables and finds which applications are installed. The interface automatically enables the installed applications, so that one can select and execute the application with the click of mouse. Above figure shows the front panel of ONAMA Execution Model. Some domains have two sections, viz serial and parallel, containing respective domain application/s.

On selection of application, a small text box appears on the main interface describing the application. The ONAMA Manual opens a manual containing description of applications, ONAMA interfaces and utilities.

CHReME

To access Linux-based HPC environments, scientists and researchers require expertise in Linux and HPC, which is something many do not have. CHReME empowers users with an intuitive GUI to exploit HPC resources and provides a layer of abstraction to shield them from the complexity of accessing HPC resources. This enables them to concentrate on their core research/scientific work. CHReME's Web Interface makes clusters of different magnitude easy to manage and monitor, which makes things easier for the HPC systems administrator while enabling researchers and scientists of varied domains to carry out their scientific simulation with minimal efforts.

Features

Comprehensive monitoring: Through CHReME, administrator can collect, monitor, visualize and analyses data pertaining to resource consumption, jobs, users, etc., thus empowering admins with actionable information.

Extensive runtime environment: This provides an extensive HPC runtime environment for parallel applications including compilers, scientific libraries, etc.

Security: Users access remote cluster resources from any location using a Web browser through HTTPS encryption and the SSH network protocol, which provides a secure data communication channel between the client machine and the HPC system.

Workload manager integration: CHReME is integrated with Torque, a resource manager, and with Maui, a cluster scheduler that helps users get optimal and efficient cluster resources.

Resource registry and management: The portal allows the creation; management and monitoring of various cluster resources like queues, parallel environments, users, nodes, etc.

Alerts and reporting: Users can receive automated notifications via e-mail, based on events such as job completion, changes in resource availability, job suspension, etc.

Credential management: This function is used to create new roles, new users, to assign/un-assign roles to users, and to disable/enable users from the HPC system.

Uploading and downloading files and folders: The portal offers a Web interface for file uploading and downloading from the client or HPC system. Errors and output files of the applications and jobs can be downloaded onto the client machine.

More detailed information about CHReME can be found in the user manual of CHReME in the path /opt/manual of PARAM Shavak.

Figure 20 - The CHReME main page

HPC Applications

The term high performance computing (HPC) refers to any computational activity requiring more than a single computer to execute a task. Computer clusters are used to solve advanced computation problems. Computational Science – and with it the associated computational resources and HPC technology – has now established itself as the third pillar of scientific enquiry alongside theory and experiment.

HPC has the capacity to handle and analyze massive amounts of data at high speed. Tasks that can take months using normal computers can be done in days or even minutes. It can be used to model and solve highly complex problems across a range of high value sectors.

Uses are diverse and examples include DNA sequencing, weather forecasting, quantum chemistry, fluid dynamics, etc…

- HPC applications available on PARAM Shavak include:
- Bio-informatics: mpiBLAST
- Molecular Dynamics: GROMACS, LAMMPS, NAMD
- Weather forecasting and Oceanography: WRF, MOM
- Quantum Chemistry: NWChem, ABINIT
- Materials Science: Quantum Espresso
- CFD: OpenFOAM

For more detailed information about HPC Applications, refer the HPC Applications user manual in the path /opt/manual of PARAM Shavak.

Reference

1. Intel® Manycore Platform Software Stack (MPSS)

https://software.intel.com/en-us/articles/intel-manycore-platform-software-stack-mpss

2. Intel® Xeon Phi Coprocessor

https://software.intel.com/en-us/mic-developer#pid-12599-231

3. Intel® Cluster Studio XE 2013

https://software.intel.com/en-us/intel-cluster-studio-xe

4. TORQUE RESOURCE MANAGER

http://www.adaptivecomputing.com/products/open-source/torque/

5. Ganglia Monitoring System

http://ganglia.info/

6. MPICH2

http://www.mpich.org/about/overview/

7. nVIDIA TESLA GPGPU

http://www.nvidia.com/object/tesla-workstations.html

Index

Accelerator, 9
Account Info, 27
Activate the device, 32
Add a new user, 25
Add a new user group, 29
Add Ethernet connection, 30
bash_profile file, 21
CHReME, 71
CPU, 7
DVD, 9
Environment variables, 35
Ethernet device, 32
Features, 7
Ganglia, 66
GPU accelerator cards, 7
Graphics, 9
Hardware configuration, 9
HDD, 9
HPC applications, 7
micsmc tool, 34
Monitor, 9
Network, 9
Network cable, 30
Network interface card, 30
ONAMA, 69
PARAM Shavak, 7
Password Info, 27
Processor, 9
PuTTY, 12
RAM, 9
Root privileges, 23
Serial and Parallel JOBS, 62
sftp commands, 12
ssh program, 12
System-config-users, 23
Terminal.app, 12
Torque Resource Manager, 62
User Data, 27