# Open Source Solutions for Installation and Management of PC Clusters under Linux for ATLAS*

Rafael Ángel García Leiva, Jose del Peso
Experimental High Energy Physics Group
Department of Theoretical Physics
Universidad Autónoma de Madrid

16th December 2002

ATL-SOFT-2003-001
15 February 2003

## Abstract

In this report the principal Open Source software solutions currently available for the installation and management of clusters of PCs under the operating system Linux are reviewed and evaluated. The software packages analyzed are: NPACI Rocks, OSCAR, OpenMosix and EU-DataGrid Fabric Management Project. In order to evaluate and compare the considered solutions, a set of criteria are proposed about the minimal desirable functionality that any software for the installation and management of clusters must provide. The proposed questions are organized into four different groups: installation, configuration, monitoring and management. How the selected software solutions deal with those questions is analyzed.

## 1 Introduction

The analysis of the ATLAS data constitutes a challenge for the computing. A production rate of about 100 events per second, each of 1 Mbyte in size, is foreseen after the third level trigger. The CPU time needed to proccess these events for analysis exceeds the capacity of any of the existing computer centers. In addition, it is fundamental to generate a larger amount of Monte Carlo simulated events in order to compare the data with various Particle Physics models. This generation of Monte Carlo events may use programs which requires more CPU time than the data proccess itself. Moreover the data has to be accessible from many institutes spread all over the world (members of the ATLAS Collaboration). This leads to the construction of a system of distributed computing formed by computer centers of the different ATLAS institutes (GRID approach). Since the events are independent of each other, solutions based on highly replicated systems of commodity components are well suited for each computer center, such as "Farms" of Linux PCs interconnected by Gigabit Ethernet, which when arranged in different configurations can be applied to simulation as well as to data analysis. Such "Farms" require automation techniques for software installation and system management. In this report a comparison between some existing Open Source solutions is performed.

With *Open Source* software is meant the software packages that can be downloaded free of charge from Internet, and it is permitted to read, to modify and to redistribute the source code (for more information about Open Source and its advantages see the home page of the Open Source Initiative [1]). *Linux* [2] is a free Unix-type operating system developed by a team of volunteers around the world. With commodity off the shelf hardware is meant the cheap PC class machines connected using standard networking technologies like Ethernet. With *dedicated* comput-

*Work supported by Ministerio de Ciencia y Tecnologia

ers is meant that the computers are only used as part of the cluster, in contrast of a network of workstations, where the nodes are configured to be used as individual computers as well. And with High-Performance Computing (HPC) is meant that the cluster has been designed to provide greater computational power than isolated computers can provide, in contrast to High Availability clusters, designed to provide reliable services.

In the next subsections the criteria to analyze and compare the different software solutions are introduced and the hardware used for the tests is described. The next four sections proceed with the analysis of each selected solution, namely: NPACI Rocks [3], Oscar [4], OpenMosix [5], and the European DataGrid Fabric Management Project [6]. In the next section other important solutions available are reviewed briefly. And finally some conclusions of the work.

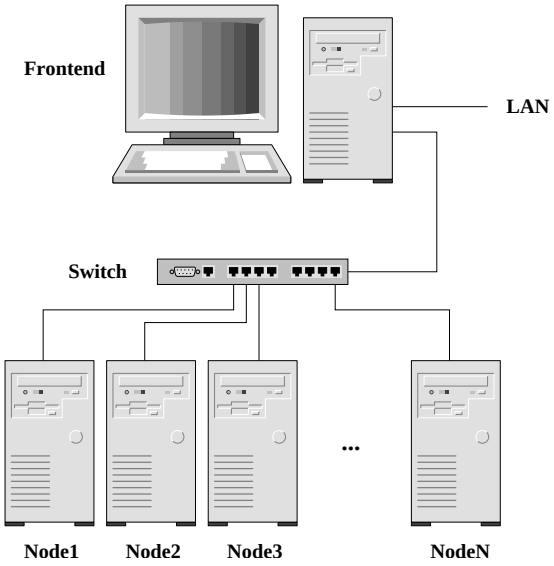


Figure 1: General Structure of a Cluster

## 1.1 General Structure of a Cluster

The most simple and general configuration of a HPC cluster consists of two or more PC boxes, in this case running Linux, connected through a high speed LAN, for example though a Fast Ethernet switch (see Figure 1). One special node is distinguished, called *frontend*, that has two network interface cards, one of them connected to a local LAN, and the other connected to a private network for the cluster. The frontend has a keyboard, a monitor and a mouse, and it is used as a gateway to the world, as login node for the users, and as fileserver. The rest of the boxes, called *computing nodes* or *nodes*, are usually headless, that is, they have no keyboard, no mouse, or monitor, and they are used for running applications.

## 1.2 Evaluation Criteria

In order to evaluate and compare the different software solutions, a set of questions are proposed about the most common problems and tasks that any system manager has to deal with when manages a Linux cluster. it will be analysed how the considered software packages solve these questions. The questions have been organized into four different groups: installation, configuration, monitoring and maintenance.

**Installation:**

- *general questions*: can we use a standard Linux distribution, or is it included? Can we use our favorite Linux distribution? Are multiple configurations or profiles in the same cluster supported? Can we use heterogeneous hardware in the same cluster, for example, nodes with IDE disks and nodes with SCSI disks, or nodes with different network interface cards?
- *frontend installation*: how is the clustering software installed? Do we need to install additional software, for example, a web server? Which services must be running and configured on the frontend (DHCP, NFS, DNS, etc.)?
- *nodes installation*: can we install the nodes in an automatic way? Is it necessary to introduce information about nodes manually (MAC addresses, node names, hardware elements, etc.), or is it collected automatically? Can we boot the nodes from a diskette, or from NIC? Do we

need a mouse/keyboard/monitor physically attached to nodes?

**Configuration:**

- *frontend configuration:* how do we configure the services in the frontend machine (configuration of DHCP server, creation of a PBS node file, creation of /etc/hosts, etc)?
- *nodes configuration:* how do we configure the operating system of the nodes, for example, keyboard layout (English, Spanish, etc.), disk partitioning, timezone, etc.? How do we create and configure the initial users, for example, how do we generate `ssh` keys? Can we modify the configuration of individual software packages? Can we modify the configuration of the nodes once they are installed?

**Monitoring:**

- *nodes monitoring*: Can we monitor the status of the individual cluster nodes, for example, load, amount of free memory, swap in use, etc? And the status of the cluster as a whole? Do we have a graphical tool to display all this information?
- *detecting problems*: How do we know if the cluster is working properly? How can we detect that there is a problem with one node, for example, the node has crashed, or its disk is full? Can we have alarms? Can we define actions for certain events, for example, mail to the system manager if a network interface is down?

**Maintenance:**

- *general questions*: is it easy to add or remove a node? And to add/remove a user? And to add/remove a software package? In case of installing a new software package, must it be in RPM format, or can be a `tar.gz` file?
- *reinstallations*: how do we reinstall a node? Is it possible, and easy, to reinstall the complete cluster?
- *upgrading*: can we upgrade the operating system? Can we upgrade individual software packages? Can we upgrade the nodes hardware?
- *solving problems*: do we have tools to solve problems in a remote way, or is it necessary to `telnet` to nodes? Can we restart a node remotely?

## 1.3 Hardware Used for Tests

For the tests, a small cluster composed of five PCs with different hardware characteristics (different hard disks, network interface cards, etc.) have been chosen in order to check the hardware compatibility of the software. All nodes have been connected using a single Fast Ethernet Switch (although two different brand of switches have been tested). As said in Section 1.1, a special node is distinguished, called frontend, and four computing nodes.

**Frontend:**

| | |
|---|---|
| Mother Board: | Asus TUV-4 |
| Proccessor: | Intel Pentium III 1GHz |
| Memory: | Hynix PC133V (256Mb) |
| Hard Disk: | Fujitsu MAJ3182MP (20Gb) |
| Video Card: | nVidia Riva TNT 2 |
| SCSI Controller: | Adaptec AHA-29160 |
| CDROM: | Asus CD S520/A (52x) |
| Network Card 1: | 3Com 3C905CX-TX-NM |
| Network Card 2: | 3Com 3C905C-TXM |

**Node 1:**

| | |
|---|---|
| Mother Board: | Asus TUV-4 |
| Proccessor: | Intel Pentium III 1GHz |
| Memory: | Hynix PC133V (256Mb) |
| Hard Disk: | Seagate ST340810A (40Gb) |
| Video Card: | nVidia Riva TNT2 |
| SCSI Controller: | No |
| CDROM: | LG CDR-8521B |
| Network Card: | 3Com 3C905C-TXM |

**Node 2:**

| | |
|---|---|
| Mother Board: | Asus CUV4X-E |
| Proccessor: | Intel Pentium III 1GHz |
| Memory: | 1x 256MB PC133 |
| Hard Disk: | Seagate ST340016A (40Gb) |
| Video Card: | nVidia Riva TNT2 |
| SCSI Controller: | No |
| CDROM: | No |
| Network Card: | 3Com 3C905C-TXM |

**Node 3 (Asus Terminator Tualatin):**

| | |
|---|---|
| Mother Board: | ASUS TUSC |
| Proccessor: | Intel Pentium III Tualatin |
| Memory: | 2 x 128Mb PC133 |
| Hard Disk: | Seagate ST340810A (40Gb) |
| Video Card: | SiS AGP 300 |
| SCSI Controller: | No |
| CDROM: | No |
| Network Card: | SiS 900 |

**Node 4 (SuperMicro Super Server 5011E):**

| | |
|---|---|
| Mother Board: | SuperMicro P3TSSE |
| Proccessor: | Intel Pentium III 1GHz |
| Memory: | 256Mb |
| Hard Disk: | Seagate ST340810A (40Gb) |
| Video Card: | ATI Mach 64 VT |
| SCSI Controller: | No |
| CDROM: | Matshita CR-177 |
| Network Card 1: | Intel Ethernet Pro 100 |
| Network Card 2: | Intel Ethernet Pro 100 |

**Network Elements:**

| | |
|---|---|
| Switch 1 | 3Com OfficeConnect 3C16791 |
| Switch 2 | SMC-EZ108DT |

# 2 NPACI Rocks

NPACI Rocks [3] is a Linux distribution, based on Red Hat Linux [7], that has been specifically designed and created for the installation, administration and use of Linux clusters. NPACI Rocks is the result of a collaboration between several research groups and private companies, led by the Cluster Development Group at San Diego Supercomputing Center [8]. For this report, Rocks version 2.2.1 (that it is based on Red Hat Linux 7.2 distribution) has been evaluated.

## 2.1 Included Software

The software packages for cluster management included within Rocks are:

- *ClusterAdmin:* a software package with tools for cluster administration, it includes programs like `shoot-node` to reinstall a node,
- *ClusterConfig:* a software package with tools for cluster configuration, it includes software like an improved DHCP server,
- *eKV:* keyboard and video emulation through Ethernet cards,
- *Rock-dist:* an utility to create the Linux distributions used for nodes installation, this package includes a modified version of the Red Hat `kickstart` installation program with support for eKV,
- *Ganglia:* a real-time cluster monitoring tool (with a web-based graphical user interface) and remote execution environment, and
- *Cluster SQL:* the SQL database schema to record the configuration of the nodes.

In addition to previous software packages, Rocks also includes the following standard packages to program and use of clusters:

- portable batch system PBS and Maui scheduler,
- a modified version of the secure network connectivity suite OpenSSH, integrated with the rest of the Rocks software,
- parallel programming libraries MPI and PVM, with modifications to work with OpenSSH, and
- support for Myrinet GM networks interface cards.

## 2.2 Installation

The NPACI Rocks distribution comes in three CD-ROMs (ISO images can be downloaded from Internet), that contains a full Red Hat Linux distribution. The installation procedure of Rocks is based on the Red Hat `kickstart` installation program and a DHCP server.

### 2.2.1 Frontend Installation

Before to proceed with the installation of Rocks in the frontend machine, it is necessary to create a configuration file for the Red Hat `kickstart` installation program. This file will contain information about the characteristics and configuration parameters of the cluster (domain name, language, disk partitioning, root password, etc). In the Rocks web page [3] there is an utility form to help to create this file.

Once the `kickstart` configuration file has ben created, the installation of the operating system on the frontend machine is automatic: one needs to insert the first Rocks CD, a floppy disk with the `kickstart` file, and reset the machine. When the installation completes, the CD is ejected and the machine reboots. At this point the Rocks frontend machine is completely installed and configured.

### 2.2.2 Installation of the nodes

Once the frontend machine has been installed, one proceeds with the installation of nodes by executing the program `insert-ethers` in the frontend machine. This program is responsible to gather information about the nodes, by capturing DHCP requests, and to configure the necessary services in the frontend (NIS maps, MySQL database, DHCP configuration file, etc).

Afterwards, the first ode is started using a Rocks CD (or a boot floppy) and the installation of the system is performed by the `kickstart` program without user intervention. Once the installation is finished, the CD is ejected and the node is rebooted automatically. At this point one can continue with the installation of the next node.

The installation of a node can be followed from the frontend by telneting to node port 8080.

## 2.3 Configuration

Rocks uses a MySQL database to store information about cluster global configurations parameters and node specific configuration parameters (for example, node names, Ethernet addresses, IP addresses, etc). The information for this database is collected in an automatic way when the cluster nodes are installed for the first time. Rocks uses this database to configure the frontend machine (DHCP server, `/etc/hosts`, etc.).

The nodes configuration is done at installation time using the Red Hat `kickstart` program. The `kickstart` configuration file for each node is created automatically with a CGI script on the frontend machine. At installation time, the nodes request their `kickstart` files via HTTP. The CGI script uses a set of XML-based files to construct a general `kickstart` file and then it applies the node-specific modifications by querying the MySQL database with the requesting node IP address.

## 2.4 Monitoring

The monitoring process of a cluster installed under Rocks can be done via standard Unix tools or using the *Ganglia Cluster Toolkit* [9].

Ganglia is a software package for real-time monitoring and remote execution of programs on Linux machines. It has been developed at the Berkeley Computer Science Division of the University of California, and it is included with the Rocks software distribution. The monitoring part is performed by `gmond`, the Ganglia monitoring daemon. This demon must be running on each node of the cluster, and it is responsible to collect information about more than twenty metrics describing the nodes state (for example, CPU load, free memory, free swap, etc.) This information is collected and stored by the frontend machine. A web browser can be used to output in a graphical way the actual state of the cluster as a whole, or the state of any particular node. In Figure 2 it is shown an example of the information provided by the Ganglia web interface.

Besides Ganglia the nodes run SNMP. The process status on any given node can be inquired using a

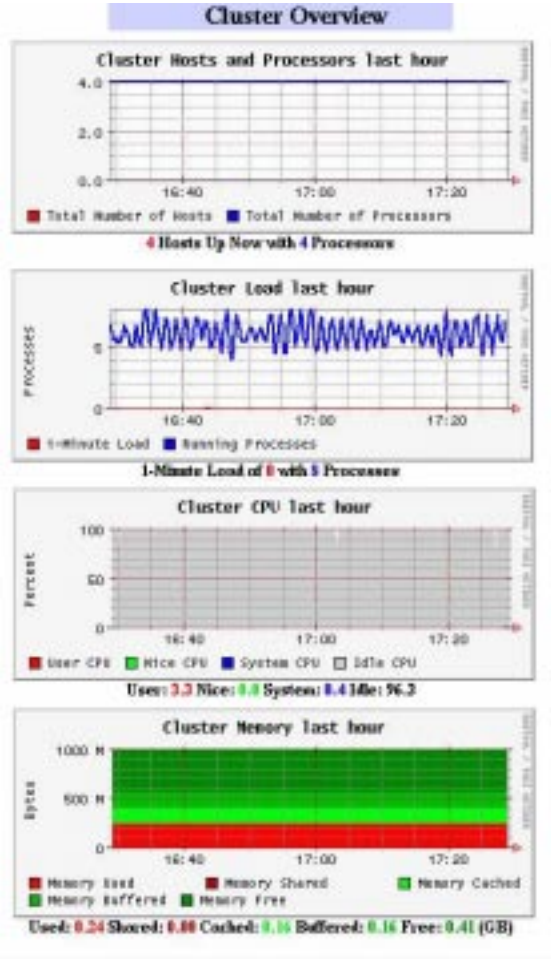


Figure 2: Ganglia Web Interface

script called `snmp-status`. Rocks also lets all the `syslog` messages of the computing nodes be forward to the frontend machine.

## 2.5 Maintenance

The principal mechanism that Rocks has for cluster maintenance is reinstallation: whenever we want to install, upgrade or remove a software package, or whenever we want to change the operating system configuration or a software package configuration, we have to perform a complete reinstallation of all the cluster nodes to propagate the changes.

For example, to install a new software package in the cluster, one has to copy the package (it must be in RPM format) into the Rocks software repository, to modify the corresponding XML configuration file and to execute the program `rocks-dist` in order to make a new Linux distribution containing the new software. Once this new distribution is created, one needs to reinstall all the nodes to apply the changes.

The Rocks program `shot-node` can be used to reinstall a node. This program receives as an argument the name of the node to be reinstalled. The evolution of the reinstallation can be followed from the frontend machine by a telnet to port 8080. The program `shot-node` can be used as well with a list of nodes to reinstall, which may be the complete cluster.

Rocks uses a NIS map to define the users of the cluster, and the user home directories are mounted from the frontend using NFS.

The tool `insert-ethers may be run to add or remove a node of the cluster.`

## 2.6 Discussion

Probably, the strongest point of NPACI Rocks is how easy it results to install a cluster under Linux. With Rocks it is not necessary to have a deep knowledge of cluster architecture or systems administration to be able to install a cluster from scratch.

Monitoring is another area in which Rocks provides a good solution. With Ganglia software, and specially with its web based user interface, it is very easy to monitor the status of the cluster or individual nodes.

Also, the Rocks philosophy for cluster configuration and maintenance, that is, *"it becomes faster to reinstall all nodes to a known configuration than it is to determine if nodes were out of synchronization in the first place"[3],* results an adequate strategy for the nodes of a cluster, because a complete reinstallation of a cluster takes only a few minutes.

Other good points of Rocks worth to mention are:

- Because the installation is based on the `kickstart` program instead of a hard disk image cloning, Rocks provides a good support for

clusters made of heterogeneous hardware nodes.

- The eKV software makes almost unnecessary the use of monitor or keyboard physically attached to nodes.
- The node configuration parameters are stored on a MySQL database. This database can be queried to make reports about the cluster, or to program new monitoring and management scripts.
- It is very easy to add or to remove a user of the cluster, we only have to modify the corresponding NIS map.

Among the bad points of Rocks we can mention the following:

- Rocks is an extended Red Hat Linux 7.2 distribution. This means that it can not work with other vendor distributions, or even with other versions of Red Hat Linux.
- In general, the documentation provided by Rocks is scarce and low quality.
- All the software installed under Rocks must be in RPM. If we have a `tar.gz` software package we have to create the corresponding RPM by ourselves.
- The monitoring program does not issue any alarm when something goes wrong in the cluster.
- User homes are mounted by NFS, which it is not an scalable solution.
- PXE is not supported to boot nodes from network.

# 3 OSCAR

OSCAR (*Open Source Cluster Application Resources* [4]) is a software package that includes a set of fully integrated and easy to install Open Source tools for the installation, administration and use of clusters based on Linux. OSCAR has been developed by the *Open Cluster Group* [10], an informal group of people with the objective to make cluster-computing practical for high performance computing. For this report we have evaluated OSCAR version 1.3 installed under Red Hat Linux 7.2.

It is worth to mention that there are companies like MSC Software that have developed Linux distributions (see for example MSC Linux [11]) specifically designed to install clusters under Linux, and that are based on OSCAR (these distributions have not been evaluated for this document).

## 3.1 Included Software

The software packages included in OSCAR are:

- SIS, *System Installation Suite*, a collection of software packages designed to automate the installation and configuration of networked workstations,
- LUI, *Linux Utility for cluster Installation*, a utility for installing Linux workstations remotely over an Ethernet network,
- C3, *Cluster Command and Control* tool suite, a set of command line tools for cluster management,
- *Ganglia:* a real-time cluster monitoring tool (with a web-based graphical user interface) and remote execution environment,
- *pfilter*, a network packet filtering system and firewall compiler,
- OPIUM, a password installer and user management tool, and
- *switcher*, a tool for user environment configuration.

In addition to the previous packages, OSCAR also includes the following standard packages to program and use of clusters:

- parallel programming libraries MPI (LAM/MPI and MPICH implementations), and PVM,

- secure network connectivity suite OpenSSH, and
- portable batch system OpenPBS and Maui scheduler.

## 3.2 Installation

The installation of OSCAR must be done on an already working Linux machine, and it is based on SIS and a DHCP server.

### 3.2.1 Frontend Installation

Previous to the installation of the OSCAR software on our frontend machine, we has to install and correctly configure one of the supported Linux distributions. Moreover the network interface card to the cluster intranet must be properly configured (for example, using a private IP subrange), and the hard disk must have at least 2 Gbytes of free space on a partition with the `/tftpboot` directory, and another 2 Gbytes for the `/var` directory.

Once the frontend machine has been correctly configured, one has to copy into the `/tftpboot/rpm` directory all the RPMs that compose a standard Linux distribution (optionally, we can include the corresponding software updates as well). Those RPM will be used to create the operating system image that will be needed during the installation of the cluster nodes.

Finally, the program `install_cluster` is executed. This program starts by adapting the frontend configuration to become the cluster server (it updates system configuration files like `/etc/host` or `/etc/exports`, creates system startup scripts at `/etc/rc.d/init.d`, restarts affected services, and so on). Then, it shows a graphical wizard (see Figure 3) that will help during the installation procedure of the cluster nodes.

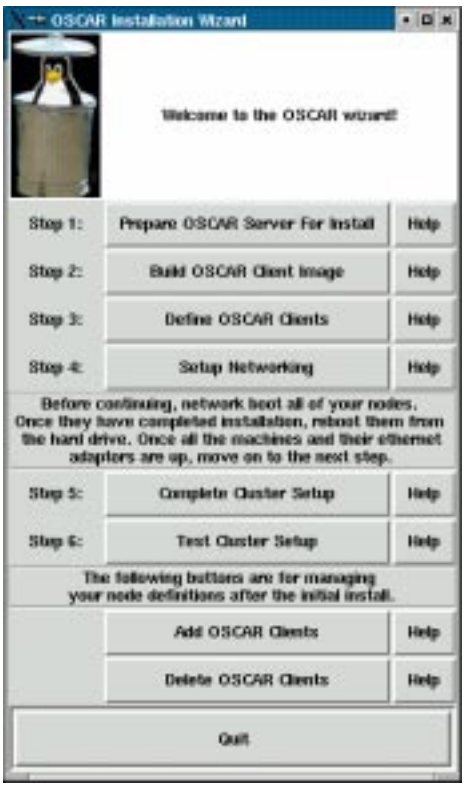


Figure 3: Oscar Installation Wizard

### 3.2.2 Installation of the nodes

The installation of the cluster nodes is done in three phases: *cluster definition*, *nodes installation* and *cluster configuration*.

During the *cluster definition* phase, one builds the image with the operating system to install into the nodes. In order to build this image, it is needed to provide a list with the software packages to install, a description of the disk partitioning, and to choose a method to assign IP addresses (static or dynamic). Once the image is ready, some information about the nodes is required, namely: node names, subnet mask, default gateway, etc. Finally, one has to collect all the Ethernet addresses and match then to their corresponding IP address.

The gathering of Ethernet addresses is the most time consuming part. OSCAR provides a graphical tool, based on DHCP requests, to help during this process. Each computing node, when it is started for first time, generates a DHCP request that it is captured by the frontend machine. This request can be generated directly by the network interface card or using a boot floppy disk. OSCAR informs about

the addresses being captured and let us to assign the corresponding IP addresses. Once all the Ethernet address has been collected and the corresponding IP addresses assigned, OSCAR configures and restarts the DHCP server.

After the cluster definition is complete, the next step is the *nodes installation*. During this phase, the nodes are boot from the network interface and are automatically installed and configured.

Finally the frontend and nodes are configured to work together as a cluster. This procedure is done automatically by OSCAR (*cluster configuration*).

### 3.3 Configuration

Frontend configuration is managed by the OSCAR wizard running the script `install_cluster`. The initial nodes configuration is done by the OSCAR wizard as well, but in this case with the help of the SIS tool suite.

The SIS tool suite is composed of three packages: *SistemInstaller*, that generates on the frontend the image with the operating system to install into nodes, *SistemImager*, that is responsible to copy the image into nodes, and *SistemConfigurator*, that performs a post-install configuration of the nodes once the images have been installed. SIS uses a database to store the information about the cluster nodes from which it obtains the cluster configuration information. The SIS tools can be used by the system manager without the wizard to update the nodes configuration.

### 3.4 Monitoring

In version 1.3, OSCAR includes an RPM package with the *Ganglia Cluster Toolkit* [9] for cluster monitoring. See Section 2.4 for a review of Ganglia functionality.

### 3.5 Maintenance

OSCAR includes C3 (*Cluster Command and Control tool suite*), a set of tools for cluster maintenance. C3 is a set of text oriented programs that are executed from the command line of the frontend, and take effects on the cluster nodes. All tools have two execution modes: sequential, where all nodes are affected in order, and parallel, where all nodes are affected in parallel.

The C3 tool suite includes the following programs:

- `cexec`: remote execution of commands,
- `cget`: file transfer to the frontend,
- `cpush`: file transfer from the frontend,
- `ckill`: similar to Unix `kill` but using process names instead of PIDs,
- `cps`: similar to Unix `ps`,
- `cpushimage`: reinstalls a node,
- `crm`: similar to the Unix `rm`,
- `cshutdown`: shut down individual nodes, or the complete cluster.

Apart of the C3 suite, OSCAR has other utilities for cluster maintenance. The OSCAR installation wizard can be used to add or remove a node of the cluster. The SIS suite can add/ remove software packages to/from cluster nodes. The OPIUM and `switcher` packages performs the management of users and their environments.

Finally, OSCAR has the program `start_over`, that let us to undo the installation of the cluster, removing all the configuration files, and to reinstall the complete cluster from scratch again.

### 3.6 Discussion

The OSCAR package provides solutions for all the aspects of cluster management: installation, configuration, monitoring and maintenance. In this respect, we can say that OSCAR is a rather complete and integrated solution.

Other good points of OSCAR worth to mention are:

- In general the documentation is clear and enough. There is an *User Guide* and a *Installation Guide*, and every software package included with OSCAR has its own documentation.

- The installation wizard is very helpful to install a cluster.
- OSCAR is a set of software packages which can be installed on any Linux distribution, although only Red Hat and Mandrake are supported at present.

Among the bad points of OSCAR one can mention the following:

- Despite the fact that OSCAR comes with an installation wizard, the installation procedure needs an experienced system manager with some knowledge of cluster architecture.
- Since the installation of Oscar is based on images of the operating system residing on the hard disk of the frontend, for each kind of hardware of the nodes, a different image has to be created. For example, it is necessary to create an image for the nodes with IDE hard disks, and another image for the nodes with SCSI disks.
- It does not provides a solution that lets us to follow the nodes installation without using a physical keyboard and monitor connected to the nodes.
- The management tool C3 is rather basic, and it has not all the desirable functionality for this kind of applications.

# 4 OpenMosix

Mosix (Multicomputer Operating System for UnIX [13]) is a clustering software that allows a set of Linux computers to work like a single system. With Mosix there is no need to modify or to link applications with any parallel programming library (like MPI or PVM), or even to explicitly migrate processes to different nodes (for example, it is not necessary to use a batch system like PBS), Mosix does it automatically.

Mosix is a project led by professor Amnon Barak of the Hebrew University at Jerusalem, and it has been developed over a period of more than 20 years.

Actually there exists an alternative (and incompatible) implementation of Mosix called OpenMosix (see [5]). This implementation is maintained by Moshe Bar, one of the original founders of the Mosix project. In their current versions, both implementations offer almost the same functionality.

In this report, OpenMosix version 2.4.17-2 is evaluated, running under Red Hat Linux 7.2.

## 4.1 Included Software

OpenMosix is not a Linux distribution nor a group of software packages, OpenMosix is just a patch that it is applied to the Linux kernel. In addition to this patch, OpenMosix comes with an optional package called `openmosix-user`. This package contains a set of tools for management of clusters running OpenMosix.

There are other software packages and solutions developed by third party groups that are based on or use OpenMosix. Among them, the most important are:

- MosixView [14]: a graphical environment for the configuration and monitoring of a cluster running OpenMosix,
- ClumpOS [15]: a CD-based Linux/OpenMosix mini-distribution, and
- LTSP [16]: the *Linux Terminal Server Project* can be combined with OpenMosix to improve the efficiency of diskless terminals.

## 4.2 Installation

Since OpenMosix is a patch for the Linux kernel, one needs first to install a Linux distribution (Red Hat Linux 7.2 for this report).

### 4.2.1 Frontend Installation

The OpenMosix patch can be installed using the standard Linux `patch` utility and recompiling the kernel with OpenMosix support enabled, or alternatively, it is possible to download a RPM package with a OpenMosix enabled kernel, and install it.

OpenMosix has a configuration file (`/etc/mosix.map`) that must contain all the IP addresses of the machines that belongs to the cluster. The administrator has to edit manually this file.

To use MFS, the OpenMosix distributed filesystem, a mount point has to be created (for example `/mfs`) and the corresponding entry in the file `/etc/fstab` must be added.

Finally, one can install other additional OpenMosix utilities, like the `user-tools` package or the MosixView program. These utilities come in standard RPM package format.

#### 4.2.2 Nodes Installation

OpenMosix does not provide an easy or automatic way to install and configure the computing nodes of a cluster. We have to install each node manually, in the same way we did with the frontend machine.

### 4.3 Configuration

Included with the `openmosix-tools` package there is the utility `mosctl` for the configuration of OpenMosix behavior. This tool let to control how the process migration is performed, for example, to force the migration of a process from one node to another, to disable all migrations for a specific node, and so on. Moreover, the MosixView program provides a graphical environment for this configuration tool (see Figure 4).

However, all the operating system configuration parameters (both, at frontend machine and computing nodes) must be modified by hand by the system manager. OpenMosix does not provide any cluster configuration tool.

### 4.4 Monitoring

The `openmosix-tools` package includes several utilities for cluster monitoring. There are modified versions of the standard Unix programs `ps` and `top`, to show the current cluster status. There is also a program, called `mon`, that displays a text based bar-char showing nodes load, speed and free memory.

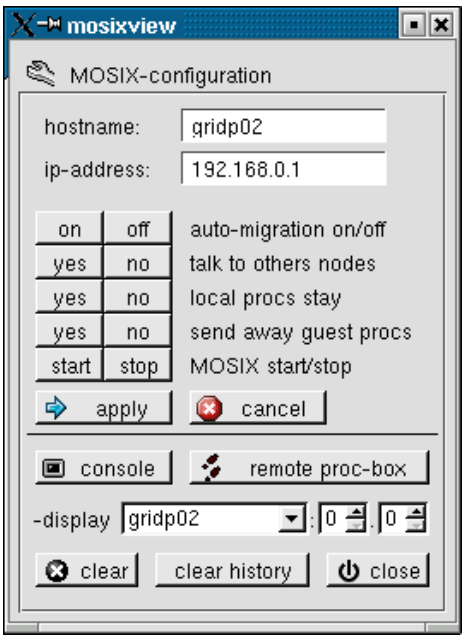


Figure 4: MosixView Node Configuration

All this information can be visualized graphycally using the MosixView utility. A graphical overview of the memory usage and load of any particular node or the average load and global memory available for the cluster can be obtained. This utility also allows to set time checkpoints and to change graph colors depending on nodes load.

### 4.5 Maintenance

OpenMosix does not provide any tool specifically designed for cluster maintenance. In general, the system manager has to perform manually all the cluster maintenance tasks. The only utility that OpenMosix has, and that can be helpful for cluster maintenance, is `mosrun`, a tool for remote execution of programs.

### 4.6 Discussion

The strongest point of OpenMosix is its ability to migrate process, and to keep load balanced among cluster nodes, even in the case the cluster is composed of heterogeneous hardware. Moreover, this

process migration is done in a way that it is completely transparent to the final user of the cluster. Another good point is that OpenMosix can be installed easyly under any Linux distribution, since it is just a kernel patch. Concerning the monitoring, OpenMosix includes the `openmosix-tools` package, which in combination with the MosixView program provides a good solution for cluster load and memory monitoring.

The main drawbacks of OpenMosix concerns that it does not provide any tool for the automatic nodes installation, it has no tools for cluster configuration or cluster maintenance, and the monitoring part does not provide any way to monitor other operating system parameters apart from cluster load and memory.

# 5 European Union DataGrid Fabric Management Project

DataGrid [6] is a project funded by European Union, with the aim to build the next generation of computing infrastructure, providing intensive computation and analysis of shared large-scale databases, across widely distributed scientific communities. The DataGrid project is organized into *work packages*, each one with a defined task to perform. The interest for this report is the job of the work package number four (WP4): *fabric management*.

The goal of WP4 is to develop a new set of tools and techniques for the development and deployment of very large computing fabrics (up to tens of thousands of processors), constructed from commodity components, and with a reduced system administration and operation cost.

Although the software developed by WP4 has been designed with the DataGrid project in mind, it can be used with general purpose clusters as well. Unfortunately, at the time of writing this report, WP4 has not released a fully functional version yet.

## 5.1 Included Software

The tools developed by the WP4 project are classified mainly into four areas:

- *installation:* automatic software installation and maintenance,
- *configuration:* tools for configuration information gathering, databases to store configuration information, and protocols for configuration management and distribution,
- *monitoring:* software for gathering, storing and analyzing information (performance, status, environment) about nodes, and
- *fault tolerance:* tools for problems identification and solution.

## 5.2 Installation

Software installation and configuration it is done with the aid of LCFG, the *Local ConFiGuration system* [12]. LCFG is a set of tools to install, configure, upgrade and uninstall software for the operating system and the applications. LCFG lets to monitor the status of software, to perform version management, and to manage application installation dependencies. LCFG supports policy-based upgrades and automated scheduled upgrades.

### 5.2.1 Frontend Installation

Our frontend machine must be installed and configured to become what LCFG calls an *installation server*. An installation server is a machine running a standard Red Hat 6.2 distribution (with Red Hat's updates), and where the LCFG installation server software has been installed.

Once the LCFG software has been installed, one has to create a software repository directory (with at least 6 Gbytes of free space) that will contain all the RPM packages necessary to install the cluster nodes. LCFG provides a set of utility Makefiles to fill in this software repository. These Makefiles are responsible to copy the Red Hat distribution RPMs from CD, to copy the RPMs updates from Internet, to extract package headers, and to create the installation root filesystem used by nodes during installation.

In addition to a working Red Hat 6.2 distribution and the LCFG installation server software, the fron-

tend machine must have the following services configured and running:

- a DHCP server, that will contain the Ethernet addresses of the nodes,
- a NFS server, to export the installation root directory,
- a Web server, that the nodes will use to fetch their configuration profiles,
- a DNS server, to resolve node names given their IP addresses[1], and
- a TFTP server, if we want to use PXE to boot nodes (instead of a boot floppy).

Finally, we need the package list files, containing the name of the packages to be installed, and the LCFG profile files, that will be used to create the individual XML profiles containing all the configuration parameters used by the nodes to install and configure themselves. LCFG brings a set of default package lists and profile files which can be modified according to the needs.

#### 5.2.2 Nodes Installation

Once we have the frontend machine properly configured, we can proceed with the installation of the client nodes. For each node of the farm, we need to add the following information in the frontend:

- a new entry to the configuration file of the DHCP server, with the node Ethernet address,
- a new entry to the configuration files of the DNS server (or `/etc/host` file), with the new name and the new IP address.

Apart from the DHCP and DNS servers, the following modifications in the LCFG server are needed:

- to create a new source file with the profile of the node (one can copy an existing file, and modify it), and
- to invoke the program `mkxprof` to make a XML file with the configuration parameters of the node,

Once the new node information is added to the frontend, one continues with the installation of the operating system. In order to do that, one needs to boot the node with a boot disk (PXE boot is supported as well). The root filesystem will be mounted by NFS from the frontend, and the node will be installed automatically.

### 5.3 Configuration

The configuration of a cluster installed under the WP4 software is managed by LCFG. The configuration of the cluster is described in a set of *source files*, held on the frontend machine and created using a *High Level Description Language*. Each "source file" describes a global aspect of the cluster configuration which will be used as a profile for the cluster nodes (linux system parameters, network parameters, etc). In addition a "node source file" is needed for each node and contains an include of the corresponding source files and the specific configuration parameters of the node (for example, hostname, Ethernet device driver, etc). These node source files must be validated and translated into node *profile files* (with the `rdxprof` program). The profile files use a *Low Level Description Language*, that it is based on the XML language, to describe nodes configuration. Finally, the profile files are published using the frontend web server.

The cluster nodes fetch their XML profiles files with the HTTP protocol, and they reconfigure themselves using a set of *components scripts*. These scripts are responsible of reading the configuration parameters and taking the appropriate actions necessary to implement the configuration. The cluster nodes can reconfigure themselves automatically whenever there is a change on the profiles files, or alternatively, whenever the system manager requests the reconfiguration of the nodes.

[1] alternatively we can use the `/etc/hosts` file of the frontend machine

## 5.4 Monitoring

The monitoring of a cluster installed under the WP4 software is done with the aid of a set of *Monitoring Sensors* that run on cluster nodes. The information generated by these Monitoring Sensors is collected at each node by a *Monitoring Sensor Agent*, and then, it is sent to the frontend machine, to a central *Monitor Repository Collector*.

A Monitoring Sensor (MS) is a process that runs on a node and measures a group of metrics (for example, CPU load, disk usage, network throughput, etc.) Many MS may run simultaneously on the same machine, each one monitoring a set of parameters. It is even possible to have sensors that sample metrics remotely (for example, we can monitor a SNMP enabled switch). However, there is only one MS implemented at present, namely `sensorLinuxProc` that uses the Linux `/proc` filesystem to measure various basic quantities (CPU load, network, etc). It is possible to implement new Monitoring Sensors using a defined sensor API.

On each node a Monitoring Sensor Agent (MSA) has to be running. This MSA is responsible to launch the Monitoring Sensors, to collect sensors measures, to store this information in a local cache database, and to send it to the frontend machine. The MSA launches the MS according to a schedule defined by the system administrator.

And finally, in the frontend machine there must be a process called Monitoring Repository Collector (MRC). The MRC collects all the information gathered by the MSAs, and stores it in a text file database (one file per metric, per node and per day).

A text-based utility can be used to query the monitoring database, or alternatively, a web browser to navigate through this information and plot the result of the query.

## 5.5 Maintenance

There are LCFG tools to install, remove or upgrade a software package in the cluster nodes. In order to do that we have to modify the corresponding RPM package list file and to notify the changes to the nodes. For example, to remove a software package, first we have to remove the RPM file from the software repository, then we have to remove the corresponding entry in the package list file, and finally we have to call the `updaterpms` program to update the nodes.

In the same way, with the LCFG tools we can add, remove or modify users and groups from nodes. In this case we have to modify the nodes LCFG profile files, and populate changes into nodes with the `rdxprof` program. Alternatively, we can run the `rdxprof` program as a daemon, in this way the changes will we automatically distributed to nodes. Moreover, if we run this program daemon, we can use a web browser to display the LCFG status of nodes.

WP4 also has developed a *Fault Tolerance* system to automatically detect node faults and performance problems. The system is composed of a *Fault Tolerance Correlation Engine*, that it is responsible to detect critical states on cluster nodes, and to decide if it is necessary to dispatch any action, and an *Actuator Dispatcher*, that receives orders to start some *Fault Tolerance Actuators*, that are independent software modules responsible to take the actions necessary to control or to solve the problem. Unfortunately, due to the lack of information about this components, they could not be evaluated for this report.

## 5.6 Discussion

The EU-Datagrid WP4 project has produced a set of tools for the installation and management of clusters that is very powerful. The LCFG software provides a good solution for the installation and configuration of a cluster. With the package lists files it is possible to control the software to install on cluster nodes, and with the source files one has full control over the configuration of nodes. The design of the monitoring part is very flexible. We can implement our own monitoring sensor agents, and we can even monitor remote machines (for example, network switches).

Other good points of the WP4 software are:

- LCFG supports heterogeneous hardware clusters.
- Multiple configuration profiles can be defined in the same cluster.

- We can define alarms when something goes wrong, and trigger actions to solve these problems.

The most important problem of the WP4 software is that it is still under development. Some of the software components are not very mature (for example, there is only one monitoring sensor currently available), and others do not work properly (there are still some bugs). Also, the documentation is incomplete (sometimes confusing). The following drawbacks have been observed in the present beta release of the WP4 software during the test:

- It only works with Red Hat Linux 6.2.
- A DNS server is needed or, alternatively, to modify by hand the `/etc/hosts` file on the frontend machine.
- There is not an automatic way to configure the frontend DHCP server.
- The nodes can be boot from the network using the PXE facility, but the solution provided by WP4 is rather artificial.
- It was not possible to install the Asus Terminator node using the boot disk, because the kernel does not includes a driver for the SIS 900 network card.
- The web-based monitoring interface is rather basic

# 6 Other Solutions

In this section we are going to mention very briefly other solutions that are available for cluster installation and management, but that have not been evaluated.

## 6.1 SCore

The *SCore Cluster System Software* is a high-performance parallel programming environment for workstation and PC clusters. SCore is currently maintained by the *PC Cluster Consortium* [17], an interest group with the aim to contribute to the PC cluster market through the development, maintenance, and promotion of cluster system software.

SCore release 5 is based on Red Hat Linux 7.2, and it has the following features: a modified kernel that turns the cluster into a single system image cluster, graphical tools for easy installation and a real-time load monitoring, and parallel programming libraries and tools with support for heterogeneous hardware and for multiple programming paradigms.

## 6.2 Scalable Cluster Environment

SCE, the *Scalable Cluster Environment* [18], is a set of integrated Open Source tools that enable users to easily build and deploy Linux clusters. SCE provides its own Linux distribution to build a diskless cluster or convert temporarily a group of free machines into a cluster without disrupting any previously installed software. Alternatively, we can install SCE on top of NPACI Rocks distribution.

SCE has the following features: a cluster installation tool, web based monitoring, a cluster management system, system administration tools, parallel programming libraries and batch scheduling system.

## 6.3 CPlant

CPlant, the *Computational Plant* [19], is a software package developed at Sandia National Laboratories, with the aim to build scalable clusters of commodity computing and networking components. The CPlant software package includes among other utilities: a modified Linux kernel, a set of management tools, a parallel version of NFS, job launch, monitoring and termination utilities, MPI and PBS (with an improved scheduler), and support for Myrinet GM networks.

# 7 Conclusion

NPACI Rocks is the easiest solution for the installation and management of a cluster under Linux. With Rocks it is not necessary to have high experience in

Linux systems administration or any previous knowledge of clusters architecture. Rocks also provides solutions for the most important problems that appears during the installation, configuration and monitoring of clusters.

To install and manage a Linux cluster under OSCAR is more difficult than with Rocks. With OSCAR it is necessary to have some experience in Linux system administration, and some knowledge of cluster architecture. However, in general, the solutions provided by OSCAR are more flexible and complete than those provided by Rocks.

OpenMosix is an ideal solution for time sharing, multiuser clusters, due to its ability for load balancing and process migration. Unfortunately, OpenMosix does not provide any solution for the installation and management of Linux clusters. To solve this problem, OpenMosix can be used together with other clustering software, like OSCAR or Rocks.

Finally, the EU-DataGrid WP4 project offers the most flexible, complete and powerful solution of the analyzed packages. However, the WP4 software is today still under development (no stable release yet), and so the WP4 solution is the most difficult solution to understand (the documentation is confusing), to install, and to use. You need to be a highly experienced system manager in order to install and manage a Linux cluster under the WP4 software. It is important for these aspects to be improved if the potential of WP4 is to be realised.

## References


[1] Open Source Initiative http://www.opensource.org

[2] The Linux Operating System http://www.linux.org

[3] NPACI Rocks http://rocks.npaci.edu/index.php

[4] OSCAR http://oscar.sourceforge.net

[5] OpenMosix http://openmosix.sourceforge.net

[6] European Union DataGrid Project http://www.eu-datagrid.org

[7] Red Hat Linux http://www.redhat.com

[8] San Diego Supercomputing Center http://www.sdsc.edu

[9] Ganglia http://ganglia.sourceforge.net

[10] The Open Cluster Group http://www.openclustergroup.org

[11] MSC Linux http://www.msclinux.com

[12] Local Configuration System http://www.lcfg.org

[13] Mosix http://www.mosix.com

[14] MoxisView http://www.mosixview.com

[15] ClumpOS http://labs.psoftware.org

[16] Mosix with LTSP http://people.nl.linux.org/~jelmer/ltsp-mosix.html

[17] PC Cluster Consortium http://www.pccluster.org

[18] Scalable Cluster Environment http://www.opensce.org

[19] CPlant http://www.cs.sandia.gov/cplant