

Nikola Pavković, Tibor Skala, Valentin Vidić

# Automatic Enlarge and Deployment of Computer Cluster Using Dual-Boot Approach

DOI 10.7305/automatika.54-2.103

UDK 004.27.032.21.057.8

IFAC 2.8.3

Original scientific paper

The paper presents an innovative and simple way of creating computer clustering and obtaining HPC clusters using computers in the classroom, Ethernet facilities and open source software. Automatic enlarge of a computer cluster is a cost-effective way to increase available computing power. This is achieved by forming a computer cluster from the computers in the classroom. The main aim of this paper is to present a solution that will use existing resources in the computer classroom, for applying complex computer services/jobs under the Linux operating system. The execution of these services/jobs programs is performed at the time when the computing resources are not in used in learning under the Windows operating system. A real-life example of applying computers in dual-boot Windows/Linux work mode along with the developed software application support is presented in a teaching environment. The implementation also includes logistics and support for automatic computer clustering and for service/job programs execution. The primary goal is to use existing resources for useful applications in education in the image programming, simulations and volume rendering.

**Key words:** Computer clustering, Dual boot cluster, Empower computing resources, Debian GNU/Linux clustering, Debian cluster components

**Automatsko proširenje i primjena računalnog grozda korištenjem dual-boot principa.** U radu je prikazan inovativan i jednostavan način stvaranja računalnih klastera i dobivanja HPC klastera korištenjem postojećih računala u učionici, Ethernet objekata, i open source softvera. Automatsko oblikovanje računalnog klastera je ekonomičan način da se poveća raspoloživa računalna snaga. To se postiže formiranjem računalnog klastera od postojećih računala u učionici. Glavni cilj ovog rada je prikazati rješenje koja će koristiti postojeće resurse u okruženju računalne učionice za izvođenje složenih računalnih poslova koji zahtijevaju veću računalnu snagu pod Linux operativnim sustavom. Izvršenje tih poslova se obavlja u vrijeme kada se računalni resursi ne koriste za obrazovanje pod Windows operativnim sustavom. Prikazan je primjer primjene računala u dual-boot Windows/Linux modu rada uz razvijene originalne programske podrške za aplikacije u obrazovnom okruženju. Provedba također uključuje logistiku i podršku za automatsko klasteriranje računala i izvršavanje poslova. Primarni cilj je iskorištavanje postojećih resursa za korisne aplikacije u obrazovanju primarno u području programiranja slike, simulacije i renderiranja. Rezultati provedenog principa su prikazani.

**Ključne riječi:** računalni klaster, računalni grozd, povećanje računalne snage, Debian GNU/Linux klasteriranje, Debian bazirana grozd distribucija DCC

## 1 INTRODUCTION

Linux clusters are more and more used as high performance (supercomputing) infrastructure facilities in scientific research laboratories, as well as in other kind of research work where high computing power is essential. Linux clusters are taking over the market share of specialized supercomputers due to their price/performance ratio. In order to build a "Linux cluster" from a number of standalone PCs, a standard Linux distribution must be upgraded with some extra functionality which will pro-

vide easy installation, administration, enforcing the security policy and monitoring of elementary resources within a Linux cluster. Although some of the tools already exist, there are a very few complete distributions of Linux targeting high-performance users. The real problem is that all cluster-targeted distributions are based on the Red-Hat as a base Linux distribution. There are many Linux distributions available on the market today, each one of them having some advantages and disadvantages. But, some quality characteristics make Debian GNU/Linux a

prime choice when selecting a platform for intensive and mission-critical applications.

From practical experience, it is very well known that Debian handles the security patches in a very unique, admin-friendly way. Debian is often the first Linux distribution that releases a patched package when a security problem occurs. In order to keep the system's security level high, this is a very important issue. Additionally, more technical and legal issues make Debian very appropriate for automatic Cluster deployment.

We integrate some existing technologies (like LDAP, System Installation Suite, Torque, C3, etc.) and develop a production-grade toolset for easier cluster management, based on Debian GNU/Linux distribution by name Debian Cluster Distribution, DCC [1]. This involves development of automation mechanisms that provide a flexible platform for high-performance computation tasks, but also provide a system-administrator to have a secure, easy to maintain, reliable and good supported cluster administration toolbox, based on Debian/GNU Linux. Our DCC is the basic distribution for the proposed automatic dual boot computer clustering solutions. As the network and system complexity grow, especially with the recent advent and acceptance of cloud computing and virtualization in general, a unified method for the simplification and automation of cluster configuration management, using a high-level description language, becomes necessary. This is our main motivation for the proposed clustering solution, which is based on a unique multiple platform solution, system program realization, easy workload tools that generally enlarge solutions with better utilized resources.

## 2 RELATED WORKS

Cluster computing has emerged as a result of the convergence of several trends, including the availability of inexpensive high performance microprocessors and high speed networks, the development of standard software tools for high performance distributed computing, and the increasing need of computing power for computational science, education and commercial applications [2].

Although early clusters [3] were built from specialized components, with dedicated specially designed interconnection circuits, most modern clusters are designed around cheap (some prefer the word "commodity" or "commercial") off-the-shelf (COTS) computers, with a dedicated, isolated Ethernet providing the inter-connection. This design is often referred to as a Beowulf cluster, whose advantages are low cost, simple creation and easy upgrade. The cluster may be built from available personal computers, or may use specialized circuit boards termed as blades. A good, though slightly dated, review of the hardware, software, and networking components used in Beowulf clustering is found in Sterling, et al. (1999) [4]. There are

other solutions, based on multiple network interfaces per computer [5-6], which enable much more effective communication in a mesh based clusters, however, at a higher price.

General coverage of clusters may be found in Bookman (2003) [7] and Vrenios (2002) [8]. Clusters tend to be classified by their primary usage, e.g., clusters intended for large-scale transaction processing, with an emphasis on rapid access and easy scalability; clusters designed to provide "fail-safe" and high-availability processing; and clusters that provide massive amounts of distributed, parallel processing power for large-scale applications amenable to this capability (often referred to as "supercomputing"). This categorization is somewhat artificial, since the technology is similar in most cases, differing mostly in configuration and choice of software.

There are few papers attempted to integrate computer clustering concepts into the school's curriculum. Mostly, the focus is in various aspects of clustering and parallel processing, (Montante, 2002) [9]. A substantial general discussion of clustering in the classroom end curriculum may be found in Apon et al., (2001) [10].

Most clusters and software support for clusters are using the Linux operating system (OS). Since the computer classroom teaching is based mainly on Windows applications, we propose in this paper a new approach for enlarging an existing Linux OS based cluster with Windows OS based classroom nodes based on dual-boot mode and automatic computer clustering procedures, using our custom developed original algorithms and software support solutions. The dual-boots provides a certain level of efficiency when using two operating systems on a single machine and applies automatic clustering. A literature reference based on a similar solution was not found.

## 3 AUTOMATIC DEPLOYMENT OF THE COMPUTER CLUSTER

### 3.1 Problem Description

A cluster of computers is a group of computers interconnected by a very fast network that operate together as a single computer with multiple processors. Alternatively, a cluster of computers is defined as a group of independent computers also known as nodes cooperating as a single system of computers intended to provide availability of resources and mission-critical applications.

The cluster of computers, shown in Fig. 1, consists of a control node and at least one computer, operating as a compute node. The users run their jobs through the control node, which in turn ensures that the jobs are processed by available compute nodes. If the compute nodes are not available, the control computer holds the jobs until

the resources needed to perform a specific job become available. There are several reasons for unavailability of a compute node for a specific job at a given time

- compute node is busy processing another job
- compute node is shut-off due to a technical fault
- compute node is shut-out because its resources are engaged in other tasks.

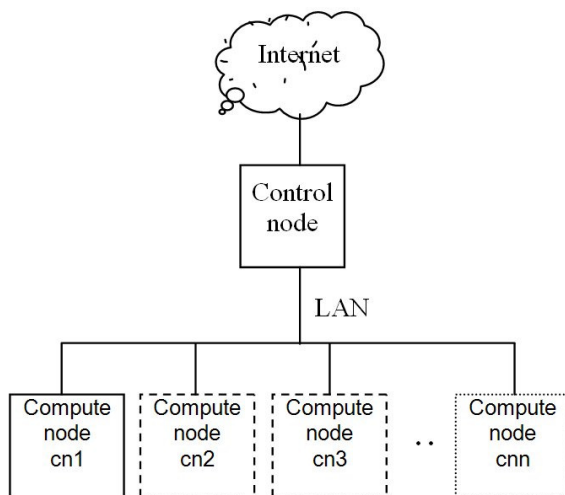


Fig. 1. Cluster of computers - architecture

The last reason for unavailability in the above list occurs when a cluster of computers uses resources of personal computers otherwise used for other purposes. Such computers are incorporated into a cluster only when they are not used for their usual purpose. Such a time may be:

- strictly defined, e.g., off hours in a specific organization
- during a randomly occurring period, e.g., upon activation of a screensaver, which indicates that the computer has not been used for a certain period prior the activation
- during a randomly occurring period of low level of resource utilization, e.g., cluster processes assigned by the control node are given the lowest priority level.

The subject of this paper is the recurring automatic inclusion of personal computers in a cluster at a time when these are not used for their regular purpose.

A computer cluster in a sense of infrastructure used for high-performance scientific/education computing is a set of computers dedicated for that purpose and - due to

its closed architecture - it is featured by very strict limitations to the range of services available to users. The *automatic enlarge of the cluster* term implies the development of models of automatic inclusion of computers, which are not always a part of the cluster, into the established cluster at a time when they are not in use. An ideal example is a computer classroom with powerful computers connected by a fast network and mostly inactive during off hours.

### 3.2 Problem Analysis

In tackling the problem, we may take two different approaches: system approach and application approach [11]. Specifically, the system approach to the extension of the computer cluster would be logical inclusion of workstations as compute nodes of the computer cluster during their period of inactivity. After it has been thus extended, the cluster could use the newly incorporated nodes for processing any job. Considering that the nodes of the computer cluster are operating under Linux operating system, for instance, the workstations of the classroom could not simultaneously be used as compute nodes if they run Windows OS or applications.

On the other hand, if we take the application approach to the extension of an already existing cluster, the operating system of the workstations used for expansion is not dealt with. Unlike the system extensions, application-level extensions involve High Throughput Computing. Applications, suitable for such a mode of processing, are featured by their readiness for a trivial parallelization, most often in terms of input data sets. Since a very few applications are suitable for such mode of processing, the application approach of extension is not useful for all purposes. Looking at the problem from that point of view, such an automatic clustering solution is suitable for educational applications in the case of Faculty of Graphic Arts (FGA), where the primary interest was to build the infrastructure resource for an advanced and complex graphical structures rendering. It is obvious that the cluster extension is more interesting on a system level because of the flexibility and uniformity. However, there is also a question of procedures needed to logically add computers that are outside the local network, to the cluster. Simple solution to the problem is that all workstations run Linux OS. The cluster extension procedure has to ensure that the workstations are able to run services, which enable batch system of the parent cluster, to create processes.

Taking in consideration the fact that the most commonly used OS in computer classrooms is still MS Windows, it is obvious that the workstations must be rebooted to Linux prior to their inclusion in the parent cluster, and further, a boot schedule should be created in order to shut down current OS and reboot to the desired operating system at a predefined time schedule.

## 4 TEST INSTALLATION

No well-known Linux cluster distribution directly supports the extension of a computer cluster by adding user workstations. In order to achieve the functionality of adding user workstations automatically, the actual Linux distribution needs to be slightly modified in terms of reconfiguration of networking and system services. We address the problems to be solved when the cluster is automatically extended, possible solutions for the problems observed and implementation issues of the test installation at the FGA Zagreb.

### 4.1 Problems to be Solved in the Automatic Cluster Enlarge

#### 4.1.1 Network Configuration

The workstations that should be included in the automatically extended computer cluster during the nighttime are connected by networks that may have very diverse configurations (topology, DHCP/static IP, routing etc). It is therefore almost impossible to define a single solution that would be generally applicable. We shall therefore assume that the cluster is using a typical network configuration: front end will be connected to the local network and compute nodes by first network interface card (NIC) and to the Internet by the second NIC. Compute nodes receive their private IP via DHCP from the front end, and they access the Internet through the front end using NAT (Network Address Translation).

The user workstations that will be included in the cluster also need to access the front end. The simplest way to provide such an access is by connecting the front end to the local network of user workstations by an additional NIC. The workstations of the network may contain other computers as well (DHCP servers, routers etc., see Fig. 2). The network configuration should be discussed with the network administrator and if necessary, certain network configuration changes have to be performed. It is possible to use the front end computer of the cluster instead of the DHCP server and the router. If the existing DHCP server is kept in the network configuration, it should lease static IP addresses to user workstations for easier cluster configuration.

The FGA uses the front end equipped only with a single NIC, because there are no compute nodes that are strictly dedicated to the HPC cluster permanently and the Internet access is provided through the router of the user workstation local network. The existing DHCP server is used to assign IP addresses to workstations. DHCP server installed on the front end is used only during the boot of the workstations over the network.

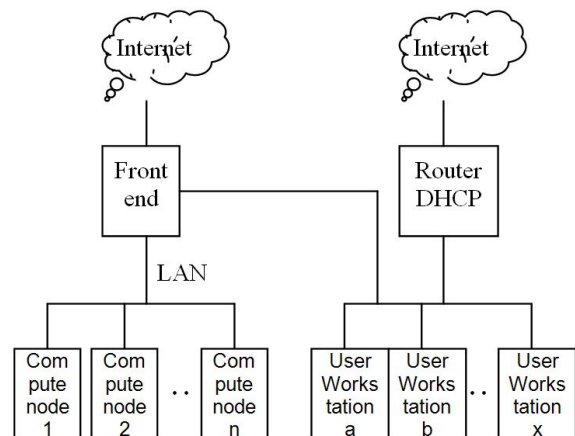


Fig. 2. Extended cluster

The DHCP server on the front end is configured as follows:

```
class "PXEclients" {
    match if substrstring(option
vendor-class-identifier, 0, 9) =
"PXEClient";
}

# This entry serves PXE requests on
eth0...
subnet 172.16.0.0 netmask 255.255.0.0 {
    pool {
        max-lease-time 60;
        filename "/pxelinux.bin";
        deny unknown clients;
        allow members of "PXEclients";
        range 172.16.100.1 172.16.100.1;
        host node1 { hardware
ethernet 00:0c:6e:22:35:6d; }
        host node2 { hardware
ethernet 00:0c:6e:22:35:6e; }
        host node3 { hardware
ethernet 00:0c:6e:22:35:6f; }
    }
}
```

The DHCP server responds to requests from known user workstations and does so only during PXE boot. Pre-Boot Execution Environment (PXE) allows a workstation to boot from a server on a network prior to booting the operating system on the local hard drive. After a workstation has booted, it is assigned an IP address by the existing DHCP server, not the one existing on the front-end. After a workstation has booted, it is assigned an IP address by the existing DHCP server.

#### 4.1.2 Installing Linux on the User's Workstations

If a workstation has necessary disk space available (>1GB), a Linux OS installation is recommended. Precautions should be taken in order to prevent damages to the existing OS (presumably Windows). The workstation is (1) booted over the network, (2) installation procedure (re)partitioning the hard disk is started and (3) Linux OS is installed on the newly created partition. The installation procedure is controlled by scripts that run from the front end computer. The scripts may be modified so that they reduce the size occupied by currently installed OS and use the freed up space for a new partition, which will later hold the Linux OS installation. It is also advisable to use the local hard drives storage of intermediate results. The advantage of such a strategy is the speedup of the process because data and files are accessed using the local hard drive. The disadvantage of the strategy is a need for manual modification and testing of the installation scripts in order to successfully re-partition the disks.

If the user workstations do not have enough free disk space, they can completely be booted to the Linux OS from the network. Such a computer can also access a complete file system, except the kernel, over the network. The advantage of this approach is that no changes are required on the user workstation (except for a modification needed to allow booting from the network), so the whole cluster can be installed faster. The disadvantage is the slower access of the file system over the network, which may become a "bottleneck problem" of such a procedure.

Due to its simplicity, the implementation of the cluster extension at the FGA followed the second procedure, using network boot. The compute nodes scheme created is modified so that it allows mounting of root file system from the network. This primarily means that a separate copy of certain system directories (*/etc*, */tmp*, */var*, */dev*) has been created. It is also necessary to use the kernel and *initrd* (a scheme for loading a temporary file system into memory in the boot process of the Linux kernel) with "root over NFS" capability.

#### 4.1.3 User Workstations Scheduled Reboot

User workstations need to be rebooted at a scheduled time in order to switch between the OSs. The reboot procedure is usually run in the morning in order to boot the workstations to the locally installed OS (usually Windows) and in the evening in order to switch to nighttime Linux cluster. Each workstation status at any given time may be in the one of the following states:

- Inactive: if the user workstation, i.e., NIC, motherboard and BIOS, support Wake on LAN functionality, which enable that it can be turned on by sending a special network frame.

- Windows OS - the workstation can be rebooted at scheduled time in either of the two ways:
  - Using the Task Scheduler - "shutdown-r" command is executed at determined time. This method is somewhat inflexible because every workstation must be separately manually configured (and eventually reconfigured) in order to be rebooted to Linux OS, but it might be recommended from the security reasons, because it is not necessary to enable remote access from the front end.
  - Remote reboot - using the "net rpc shutdown" command issued from the front end. Appropriate authorizations on the Windows computer (username and password) are required to run this command over the network. However, the changes in the reboot schedule are reflected only in front end computer.
- Linux cluster - the nodes can be rebooted easily because they can be accessed securely by using SSH (authentication is done by RSA keys).

On our test installation the reboot is initiated as a cron job at 7 am and 9 pm. By the morning reboot, the workstations are in Linux OS cluster mode and can be consecutively accessed by SSH to initiate the reboot procedure. By the evening reboot, the workstations are running the Windows OS. The reboot is initiated using the "net rpc shutdown" issued from the front end.

Cron script performing the morning workstations' reboot from the Linux OS to the Windows OS is as follows:

```
ln -sf
/tftpboot/pxelinux.cfg/{syslinux.cfg,
localboot,default}
USER=root cexec /sbin/reboot
```

Cron script performing the evening workstations reboot from the Windows OS to the Linux OS is as follows:

```
ln -sf
/tftpboot/pxelinux.cfg/{nfsboot,default}
while read ip user pass; do
    net rpc shutdown -I $ip -U $user%
    $pass -r -f -t 300
done < /root/windows.hosts
```

#### 4.1.4 OS Sensitive Scheduled Workstation Reboot

After the workstation boot has been started, an appropriate OS should be started. The most flexible solution is to perform the boot by using PXE. The network boot must

be enabled and set as the primary boot medium (before the local disk) in the BIOS of the workstation. When such a station is booting, it uses the DHCP and TFTP server to retrieve a configuration of the appropriate OS and boot method.

If Linux boot is required, the configuration data will include the name of the kernel file (retrieved over the network) and the name of the root partition (if Linux OS is installed locally, on the hard drive), respectively, and NFS server, if "root over NFS" method is used. If the workstation needs to boot the Windows OS, the PXE configuration file will contain a local (i.e. from hard drive) boot command. Consequently, the front end computer should have two configurations containing parameters for either reboot to the Linux OS and to the Windows OS and should place them in the right time to a location accessible by workstations when configuration parameters need to be retrieved (/tftpboot/pxelinux.cfg/default). The advantages of the PXE are in its flexibility (changes in the configuration are done centrally, on the front end) and in the fact that the local workstation bootloaders need not be modified.

If a workstation does not support the PXE, a bootloader that allows booting from the network may also be used. One example is the Grub bootloader [12]. Apart from the network boot, the Grub is able to retrieve the configuration (menu) from the TFTP server. Two configurations are kept on the server: first is used for the Linux OS boot and the second for the Windows OS boot. A configuration forcing the workstations to boot the Windows OS is used in the daytime. In the evening (e.g., 9 pm), a cron job, responsible for reboot of the Windows workstations, sets up a new configuration, which boots the Linux OS (both local and "root over NFS" boots are available). In the morning, the workstations are rebooted from the Linux to the Windows OS and a new Grub configuration for booting Windows is set. This method is as flexible as the PXE boot, but requires modifications of the local bootloaders. If, for any reason, it is not possible to boot a computer from the network, there is also a possibility of using advanced bootloader (e.g., Smart Boot Manager) to boot different operating systems depending on the local clock. This is the least flexible solution because each of the configuration changes (such as time of boot for a particular OS) must be done on each workstation bootloader.

The user workstations use the PXE boot. If the PXE is given a configuration that requires that the boot must be local, the Windows OS will be started for the time interval from 7 am to 9 pm. In the same way, the PXE can perform the Linux boot. The cron service that initiates the workstation reboot also performs modifications of the PXE configuration file. The PXE configuration file, used for the Linux boot, consists of the names of files containing kernel and initrd (retrieved via TFTP) and "root over NFS"

parameters (the name and location on the NFS server and IP assign method (DHCP)).

The PXE configuration to be used for booting the Linux OS is:

```
DEFAULT nfsboot

LABEL nfsboot
PROMPT 0
TIMEOUT 50
KERNEL kernel.nfs
APPEND initrd=initrd.img.nfs
root=/dev/nfs
nfsroot=172.16.1.96:/var/lib/systemimager
/images/node ip=dhcp
```

The PXE configuration to be used for booting the Windows OS is:

```
DEFAULT localboot

LABEL localboot
LOCALBOOT 0
```

#### 4.1.5 Job Scheduling

For the scheduling in the extended cluster, it is desirable to use a more advanced scheduler, which takes into account the fact that workstations are not available between 7 am to 9 pm. The Maui scheduler was used in the FGA test cluster implementation. The configuration of the scheduler allows node resources to be reserved at the time the workstation is running Windows OS:

```
SRDAYS[windows] MON TUE WED THU FRI
SRSTARTTIME[windows] 07:00:00
SRENDTIME[windows] 21:00:00
SRTASKCOUNT[windows]
number_of_nodes_in_cluster
```

If such a reservation were not made, for instance, the Maui scheduler might start a job needing five hours of processing, for example at 5 am. Such a job would be processed for two hours and it would be interrupted at 7 am when the workstations reboot is scheduled. If jobs are still running and workstations need to be used by their regular users, which mean that they should be rebooted to Windows OS, it unfortunately means that those jobs, which happen to run at the specified OS-switch time, are killed. There is still an open research area with many unresolved problems directed to the following discussion: what happens if jobs are checkpointed and moved to another location. However, the above configuration is not entirely correct. Assuming that the workstations will be running Linux

OS over the weekend, they may be allowed to start operations at any given processing time. The Maui is automatically adding scheduled reservations just, however, only a single day ahead, i.e., if a reservation expires on the Friday night, new reservations will not be added until Monday. The Maui will allow processing of arbitrary long jobs during the weekend.

For the described example above, the Monday reservations must be inserted in cron using the following script:

```
DATE=`/bin/date --date="+3 days" +%m/%d`
/opt/maui/bin/setres -n windows.MON -s
06:50:00_$DATE -e 21:10:00_$DATE ALL
```

Similar functionality can be achieved with other schedulers such as: SGE, TORQUE C/FIFO (dedicated time) etc. We were using only Torque C/FIFO scheduler and the functionality of SGE was explored only theoretically.

#### 4.1.6 Remote Access to a Cluster

The users should be allowed to access to front end remotely in order to ensure a wide accessibility of the cluster. A minimum requirement for the remote access is the SSH functionality of the front end that is easily achievable if a public IP address can be assigned to the front end. However, the front end node and the workstations usually have only private IP addresses, hidden from the Internet behind NAT. In this case, the front end must create a tunnel to any publicly accessible computer. The tunnel must go through NAT barrier and allowing the remote user to communicate back (send information to the front end). The test installation at the FGA cluster tunnels the SSH traffic only. Thus, the TCP port 2200 on publicly available computers is tunneled in a SSH session to port 22 of the front end. In this way, the users connected to the port 2200 of the publicly accessible computer are in fact communicating with the front end computer, consequently with the entire cluster. If a large number of ports need to be tunneled, it is easier to implement a PPP connection, thus assigning a public IP address to the front end.

The above text describes our methodology for enabling automatic enlarge functionality to an existing cluster installation based on the DCC (Debian Cluster Components) distribution, which was chosen because authors are most experienced with it. The described procedures can be applied in the same way also to any other Linux cluster distribution.

## 4.2 DCC Description

The DCC is a set of additional packages installed on an already active Debian system, rather than a complete Linux

OS distribution. In this way, the system uses the existing packaging system (apt) and other packages existing in the standard Debian distribution, which significantly reduces the DCC maintenance costs. Most of the DCC packages re-configure common Debian packages for functioning in the cluster environment (LDAP server, DHCP, TFTP etc.). In addition to the packages used to reconfigure the existing system, the DCC contains a small number of binary packages, e.g., Torque, C3, Ganglia, etc., for the creation and maintenance of the cluster. The most important components of the DCC cluster implemented at FGA are listed below:

- TORQUE - job scheduler (the successor of the OpenPBS)
- SIS - used for easy creation of a node image (debootstrap) and network installation of nodes (PXE)
- LDAP, CPU - used to remotely manage of user accounts within the cluster
- C3 - cluster maintenance tools
- Ganglia - cluster control through a web interface
- Shorewall - firewall configuration.

The common installation method of a Debian package (apt-get install package-name) can be used to install any additional Debian package (MPICH, LAM MPI, PVM, PovRay etc.) that might be needed by the users of the extended cluster.

## 4.3 Test Installation Results

Our installation of the FGA extended cluster was preliminary tested by repetitive usage of a graphical rendering program, which is a CPU intensive application. The testing has shown that "root over NFS" approach does not affect significantly the time needed to process such applications. The test cluster was controlled by the scalable distributed Ganglia monitoring system [13], which is a standard and robust implementation, ported to Linux/Windows OS and various processor architectures. In Fig. 3, obtained from the Ganglia cluster monitoring system, we can see a clear difference between the night and the day operation mode on a three workstation from the extended cluster. In Fig. 4, the corresponding network load at the front node is shown, for the same time period. We see increased network traffic only during the network boot to Linux/Windows OS. In the meantime, the traffic is quite constant on the value of about 50 kB/s, with three compute nodes that were present at the time. After the cluster applications have been started on the compute nodes, at 1 am, the network load is only slightly increased.

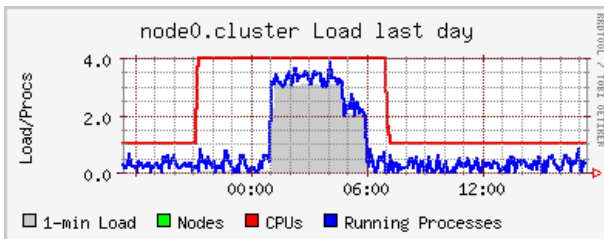


Fig. 3. Daytime vs. nighttime mode

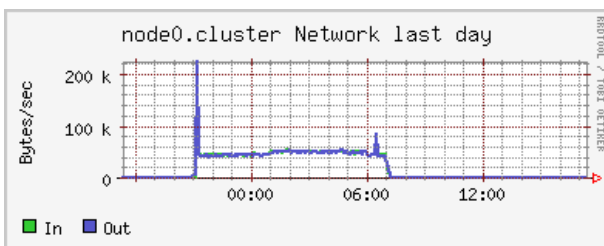


Fig. 4. Front node load

#### 4.4 Spatial and Temporal Separation

Eventhough the two functions of the computer classroom or lab workstations could be dynamically combined, this is not recommended and therefore we strictly use spatial and temporal separations. The spatial separation consists of a physical separation of OS images on different hard disk partitions and of using separate network segments; while temporal separation implies that the two operation modes will be active in different time periods.

### 5 ORIENTATION TO APPLICATIONS

Based on our practice work, we found out that image processing, using a computer cluster for scientific, technical or educational visualizations, is a useful applied orientation. The greatest motivation to empower existing resources with those from the classroom was to justify previous investments and to produce immediate result for users/students that were not able to obtain their own grid certificate. Such a profile of users is to be expected, because the FGA is an institution that is not primarily focused to computer science, however, it relies in great extent on the ability of the computer graphics.

Our choice for the POV-Ray rendering agent was motivated by the open source and free license of the program. Besides, it is popular also in cluster/grid community and compatible with the Linux environment [14].

We want to popularize the new technology in a way that it will be friendly for graphic design professionals and for student community. Initial implementation includes Debian Linux and MS Windows mixture with rendering

agents in a farm. (Cygwin offers POSIX environment on the top of Windows architecture.) The render farm was distributed across the classroom and across the different hardware and OS architectures with different machine performances. Such a heterogeneous system made the selection of the FCFS simple scheduler a logical alternative to complex schedulers, used by grid resource brokers.

Even that our focus application is oriented to the volume rendering we have proven the usefulness, generality, and applicability of the implemented system. Based on automatic dual-boot, the enlarge computer cluster effectively uses the educational resources, while idle, for scientific and technical applications.

### 6 DISCUSSION AND CONCLUSION

The underlying idea of this research and development was that the computer classroom/laboratory should be used in two operation modes. One operation mode is educational work during daytime and the other mode is active during the night when the free processing power is engaged in cooperation with an already existing computing structure thus serving as an auxiliary computing power. It is interesting that the primary OS used for computing is Debian GNU/Linux, while Windows OS is first choice for education purposes. Automatic clustering is achieved in a simple and interesting way: boot-server configuration is modified at a predefined time and reboots process is initiated on the client workstations.

Automatic enlarge of a computer cluster is a cost-efficient method of increasing available computing power. The enlarge method presented in active state is an implementation of our existing solution "Debian Cluster Components". A modification of DCC in a form of additional Debian package enabling diskless cluster operation was necessary. "Diskless" mode is very convenient, because it is a non-invasive implementation of a complete solution to an existing computer classroom facility.

Presented solution is creating significant HPC computing resources in the classroom or laboratory educational environment. Obtained by expansion of infrastructure in the educational purposes gives a large gain without any investments made. The computer cluster system in the phase of system and application testing has shown to be a steady and reliable operation. Like any complex process, and system solution always shows the advantages and disadvantages or functional limitation.

In this work communication bottlenecks influence on the empowered cluster, because there is just a single communication line from classroom to the dedicated cluster.

So, the bottleneck problem in our case is found in a local area network (LAN) and its performances. On the



topology and bandwidth of active equipment used in specific LAN, depends what will be the network bandwidth between nodes.

Regarding classroom node homogeneity there is no need to have management of heterogeneity. It is solely the responsibility of the user, possibly the local scheduler can be configured in a manner that groups of nodes of similar characteristics are in the common queue.

When talking about concrete jobs and type of HPC applications that are expected to run efficiently on the empowered cluster, we can conclude that this kind of cluster, since limited duration of performing a job and lack of checkpointing is applicable only for short time jobs (up to 10 hours of walltime). Most education programs in the field of simulation and rendering with the typical classroom resource (up to 30 nodes) belong to that class's of jobs.

Application testing showed very reliable operation and major improvements in efficiency of resource utilization. Such an approach could increase resources available for scientific research oriented education on any given educational environment.

## ACKNOWLEDGMENT

The authors acknowledge financial support from EU FP7 SEE GRID SCI and the state budget by the Croatian Ministry of Science, Education and Sport, under grant 098 2562-2567, and would also like to thank for all our colleagues at FGA for their support in setting-up our testing environment.

## REFERENCES

- [1] <http://dcc.irb.hr/>
- [2] R. Buyya (ed.), *High Performance Cluster Computing Systems and Architectures*, Prentice Hall, 1999.
- [3] T. Sterling, T. Cwik, D. Becker, J. Salmon, M. Warren and B. Nitzberg, "An assessment of Beowulf-class computing for NASA requirements: Initial findings from the first NASA workshop on Beowulf class clustered computing", in *Proceedings book of the IEEE Aerospace Conference*, (Aspen CO), IEEE, March, 1998.
- [4] T. Sterling, J. Salmon, D.J. Becker & D.F. Savarese, *How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters*. MIT Press, Cambridge, MA, 1999.
- [5] R. Trobec, "Two-dimensional regular d-meshes", *Parallel Computing* 26 (13-14), 1945-53, 2000.
- [6] J. M. Camara, M. Moreto, E. Vallejo, R. Beivide, J. Miguel-Alonso, C. Martinez, and J. Navaridas, "Twisted torus topologies for enhanced interconnection networks", *IEEE Transactions on Parallel and Distributed Systems* 21, 1765-1778, 2010.
- [7] C. Bookman, *Linux Clustering: Building and Maintaining Linux Clusters*, New Riders, Indianapolis, 2003.
- [8] A. Vrenios, *Linux Cluster Architecture*, Sams, Indianapolis, 2002.
- [9] R. Montante, "Beowulf and Linux: an Integrated Project Course", *Journal of Computing Sciences in Colleges*, no. 17(6), pp. 10-18, 2002.
- [10] A. Apon, R. Buyya, H. Jin & J. Mache, Cluster Computing in the Classroom: Topics, Guidelines, and Experiences. *1st IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001)*, Brisbane, Australia, May, 2001. at [www.buyya.com/papers/cc-edu.pdf](http://www.buyya.com/papers/cc-edu.pdf)
- [11] E. Afgan, P. Bangalore, & T. Skala, "Scheduling and planning job execution of loosely coupled applications", *The Journal of Supercomputing*, DOI: 10.1007/s11227-011-0555-y, 2011.
- [12] <http://www.gnu.org/software/grub/>
- [13] <http://ganglia.sourceforge.net/>
- [14] <http://www.povray.org/>



**Nikola Pavković** received B.Sc. degree in 2005. at the Faculty of Electrical Engineering and Computer Science at Zagreb University. Since year 2000. he works as Linux-cluster system administrator. He was Debian Cluster Components project leader and become head of Information Systems Department at CIR at Ruder Bošković Institute in 2008. where he still works today. Since 2008. he is assistant lecturer at Faculty of Electrical Engineering and Computing, Zagreb. He has more than ten scientific and professional papers published in international and national journals and proceedings in national and international conferences. He conducts scientific research and participates in national and EU FP6 and FP7 projects.



**Tibor Skala** received a title of electrical engineer in 2001. from the Faculty of Electrical Engineering and Computer Science at Zagreb University. In the period from 2004. he works as a teaching assistant at the Department of Multimedia and Information Systems at the Faculty of Graphic Arts in Zagreb for the group of collegiums in module Multimedia. He enrolled postgraduate study in 2004. year, and received PhD degree for doctoral dissertation titled "Efficiency of the procedure of generating graphic content on distributed computing systems," in 2010. He has around thirty scientific and professional papers published in international and national journals and proceedings in national and international conferences. He conducts scientific research and participates in national and EU FP6 and FP7 projects. He collaborates on research projects on European SEE-GRID and SEE GRID-SCI 2008-2010. and the project of Ministry of Science, Education and Sports of the Republic of Croatia named: Evaluation of quantitative and qualitative criteria, the process of graphic reproduction and the bilateral Croatian - Slovenian project: Intelligent Agents in educational programs, the Faculty of Education in Maribor and Graphic Arts in Zagreb, Slovenia, 2003.



**Valentin Vidić** studied computer science at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. His first job following the graduation in 2004. was building secure wireless network on the Ruđer Bošković Institute. He was then involved in several national and European projects aimed at building a scalable computational grid infrastructure: CRO-GRID, SEE-GRID and EGEE. During this time he spent one year collaborating on the LCG project in CERN, Geneva. He is also one of the authors of the Debian Cluster Components suite. He is currently employed as a Linux system administrator in Croatian Academic and Research Network.

#### **AUTHORS' ADDRESSES**

**Nikola Pavković**  
**Information Systems Department,**  
**Ruđer Bošković Institute,**  
**Bijenička 54, 10000 Zagreb, Croatia**  
**email: nix@irb.hr**

**Asst. Tibor Skala, Ph.D.**  
**Department of Multimedia and Information Systems,**  
**Faculty of Graphic Arts,**  
**University of Zagreb,**  
**Getaldiceva 2, 10000 Zagreb, Croatia**  
**email: tibor.skala@grf.hr**

**Valentin Vidić**  
**CARNet**  
**Josipa Marohnića 5, 10000 Zagreb, Croatia**  
**email: valentin.vidic@carnet.hr**

Received: 2011-08-25

Accepted: 2012-09-24