EFFICIENT POWER MANAGEMENT IN HIGH PERFORMANCE COMPUTER CLUSTERS

Carlos de Alfonso, Miguel Caballer and Vicente Hernández
Instituto Universitario I3M, Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia, Spain

Keywords: Green computing, Cluster.

Abstract: Nowadays the computer cluster infrastructures are very common not only in research centers, but also in the business environment. These computer infrastructures were traditionally used to perform complex mathematic calculations using parallel programs, but currently it is extended the use of them as virtualization platforms and shared by means of grid middlewares. Anyway, this kind of infrastructures usually has many idle periods due to different issues such as the lack of intensive calculations, holydays, weekends, etc. Even having reduced the power consumption of different components, and accomplishing energy saving initiatives, it would be better to not to waste energy when it is not needed. This paper shows that green computing techniques can be applied in these infrastructures, powering-off the nodes which are not being used and switching them on when needed, to enable reducing power consumption considerably. These techniques must be applied trying to cause a minimum impact to the user who is using the infrastructure, and without interference in running applications.

1 INTRODUCTION

Green computing or green IT, refers to environmentally sustainable computing or IT. It is "the study and practice of designing, manufacturing, using, and disposing of computers, servers, and associated subsystems - such as monitors, printers, storage devices, and networking and communications systems - efficiently and effectively with minimal or no impact on the environment. Green IT also strives to achieve economic viability and improved system performance and use, while abiding by our social and ethical responsibilities. Thus, green IT includes the dimensions of environmental sustainability, the economics of energy efficiency, and the total cost of ownership, which includes the cost of disposal and recycling. It is the study and practice of using computing resources efficiently." (Murugesan, 2008)

Nowadays is usual to perform experiments consisting on the model simulations using computers. Moreover these models are more and more complex and require greater computer infrastructures to perform the calculations efficiently. So, it is common to use singular infrastructures of cluster of computers, to compute these models.

On the other hand, it is extended the use of computer infrastructures as virtualization platforms. These platforms are usually managed by some specific middlewares as OpenNebula (Fontan, 2008), Eucalyptus (Nurmi, 2008), etc. that enable to manage virtual machines (power on, stop, check the state, etc.) efficiently in a cluster infrastructure. Nevertheless, these middle-wares try to have only the needed virtual machines actives. This way, in many times, the platform is not completely used.

Maintain these infrastructures working have an important power impact. Furthermore is important to consider that the computer power source have efficiencies lower than 80% (Liang, 2008) (although there are many current initiatives to improve this ratio as (The 80 PLUS Program, 2009) or (Hoelzle, 2010)). The rest of the enery is transformed into heat, which may be dissipated by coolers which, indeed, need energy.

Many other previous green computing works, realted to high performance cluster systems, were focused on reducing the power consumption and heat disipation of the different components of the computer (mainly the CPU) (von Laszewski, 2010), (Feng, 2008), (Feng, 2007).

This paper describes techniques which aim at reducing power consumption of high performance

de Alfonso Laguna C., Caballer M. and Hernández V. (2010).
EFFICIENT POWER MANAGEMENT IN HIGH PERFORMANCE COMPUTER CLUSTERS.
In *Proceedings of the Multi-Conference on Innovative Developments in ICT*, pages 39-44
DOI: 10.5220/0003036500390044

cluster systems, which run batch jobs. That working pattern match many scientific computing clusters which apply queue managers such as Portable Batch System (PBS), Load Sharing Facility (LSF), or others, using similar ideas than shown in (Da-Costa, 209).

These techniques can be also applicable to clusters shared using Grid middlewares (Globus Toolkit (Globus Alliance, 2009), gLite (EGEE, 2009)), since the job submission is performed using the same queue managers.

The next section shows the special features of the working modes of the clusters in the described cases. Then some approaches to reduce the power consumption are proposed. Finally it is show a case analyzing the impact of the raised measures in a real cluster.

2 CLUSTER BEHAVIOUR

A cluster is composed by, besides the passive elements (screws, cables, etc.), the following elements: working nodes, administration nodes, front-end node, storage systems, internal network switches, external switches, firewalls, etc. (Lucke, 2004).

The power needed to feed this infrastructure is great and it is convenient to have a power saving plan, to minimize the consumption. This plan must take into account the way that the cluster is used, and should try to reduce the impact in the throughput of the applications and the system as a whole.

A common way to use a cluster consist in having a front-end which is in charge of coordinating the execution of user's jobs by a Local Resource Manager System (LRMS), such as a queue system (Torque, OpenPBS, LSF, Sun Grid Engine, etc.). In this case, the users submit jobs to a queue, and they are executed as they have enough nodes to satisfy the job's requirements.

In the last years, a common way of using a cluster was by sharing it by a Grid Computing middleware such as Globus Toolkit. The way of running jobs in the cluster by Grid techniques is partly integrated with the LRMS. So the usage of the cluster consists in submitting the jobs to the local queue. The main difference in this case is that it is needed an extra information reporter, which must provide the number of free online nodes (among other characteristics of the cluster).

A recent manner of managing a cluster consists in having it as a virtualization platform. Some examples of Virtual Appliance management platforms are VMWare, Eucalyptus or Open Nebula. Using these middlewares, the Virtual images are started in the form of running virtual machines, in the internal nodes of the cluster.

A power saving plan must take into account the way of using the clusters, but also the hardware issues. The most common way to reduce the power consumption is to power off the machines, but the problem can arise when trying to power on the nodes again. In this sense, there are mainly two approaches: the usage of Power Distribution Units (PDU), to manage the power access to the nodes, or the usage of Wake-on-Lan alternative.

Each of the approaches need that the nodes are configured in a specific way: using a PDU requires that the node powers on when power is restored, and WOL alternative needs that the network card is well tuned by the operating system.

Regarding any of the alternatives, there is always a residual consumption associated to the PDU controlling system or to the network card WOL monitoring. So it will impossible to achieve the zero consumption.

3 TECHNIQUES

Different techniques can be used to save energy in the computer cluster environment. These techniques are based in powering-off idle nodes, or alternatively, hibernating them or putting them in standby mode.

These two last approaches may have problems in Linux environments since (1) many devices do not support the standby mode and (2) in many cases (depending on the memory used in the machine, as the main factor) hibernating the node and starting up the machine again implies far more time than switching-off and on the computer.

On the other hand, the basic criterion for applying any of these techniques is that the node must be idle. Furthermore, the process of switching-on a working node in a cluster is quite fast, as during the boot process only the minimum applications are needed to be loaded. So the approach of powering off an idle node and powering it on when it is needed is an appropriate solution.

In any case, the aim of using these techniques is to achieve the maximum power saving, trying to maintain the response times for the users, and searching for the easier measures to apply.

In order to achieve those objectives the following aspects must be considered:

- To select the power-on/off block size: If a group of nodes are idle, they can be switched

off. Nevertheless, it is interesting not to power down all the cluster nodes (when possible), considering an eventual increase in the demand. Reciprocally when a demand of a set of nodes is detected, instead of switching on only the number of needed machines, a block of nodes can be activated preventing a future demand of more resources. The "block size" can be defined as the number of nodes to be deactivated when the cluster is idle, or to be activated when some resources are needed. This size can change from one node to the whole cluster, but it is important to select the proper one, depending on the cluster and the user needs.

The usage of a small block size enables a maximum power reduction, since only the needed nodes will be activated. On the other hand, it would increase the response time for the users, since in many cases the launched jobs will wait for the cluster to be switched on before getting their jobs started.

In case of using a large block size, the power consumption is greater, since the number of activated machines would be more than needed by the users. But, on the other hand, as users usually send consecutive executions, they only should wait for the first one, while the other jobs will be launched immediately.

According to the usage of the cluster, it should be applied an intermediate approach, which reduces the waiting times to the users, but also reduces the power consumption. Some mixed solution can be also chosen so that it can be used different heuristics to change the block size according to different factors: the job requirements, the time when the job is submitted, the number of connected users, etc.

- To select the cluster inactivity time to deactivate: In a HPC cluster environment it is quite simple to find patterns in the the usage of the nodes, due to the use of Local Resource Management Systems (LRMS). A first approach consist in checking the LRMS logs files to get the maximum and minimum inactivity times, to obtain the appropriate values to the system usage.

  To work in a correct way for estimating the switching off time, the jobs can be grouped in the switched-on nodes, correlating with the power-on/off block sizes, to obtain the waiting times and thus determining the most suitable time to wait before suspending a node.

- In addition, other heuristics can be introduced, considering the working hours, the holidays, etc.

To get the best solution with the most appropriate values for the analyzed parameters, a good approach is to make some simulations, varying the different parameter to evaluate the system behavior and selecting the best values.

Anyway, the behavior of the system may change in the future, according to different projects, users interests, etc., so it will be necessary to re-evaluate the parameter values to adequate them to the new system usage and user needs.

4 GREEN COMPUTING APPROXIMATION

We have developed a system which, taking into account the considerations in previous sections, powers-on and off the nodes in a cluster.

The solution is divided into two subsystems: the core engine, which is in charge of orchestrating the power-on and off of the system, and the plug-ins which must evaluate which nodes have been inactive for a long time, and so they are likely to be powered off.

The core engine consists in a script which is periodically launched by common linux utilities (in our case, the cron daemon). The pseudo code of the body of this script is stated below:

```
inactive_nodes=
for each plugin in plugin_dir
 candidate_nodes = exec_monitor
 inactive_nodes = inactive_nodes
        intersect candidate_nodes
end for
poweroff inactive_nodes
```

In our approximation, there must be implemented a plug-in for each way of using the cluster. Each plug-in has two parts: the first one will be invoked when checking the state of usage of the cluster, and will return a list containing the nodes which are inactive. The second part will be in charge of on demand powering-on the nodes of the cluster, and must be integrated with the running subsystem.

Currently we have developed the Torque-LRMS plug-in. The plug-in in charge of listing the inactive nodes uses the "pbsnodes" to get the list of the resources and their states. The script in charge of getting the information about the demand uses the "qstat" command in order to get the jobs launched into the system, and the resources needed by them.

To switch-on the nodes the "Job Submission Filter" option of the "qsub" command is used to enable launching a script before the effective submission of a job into the queue system. This script checks the availability of the resources needed by the submitted job, and switches-on a set of nodes, if necessary, using the selected block size. This solution minimizes the impact to the user since it will use the same commands to submit the jobs to the system.

This cluster also is shared using the Globus Toolkit grid middleware. The service Globus Resource Allocation Manager (GRAM) provides the capability to submit jobs to the cluster LRMS. The GRAM service as well as the users, uses the "qsub" command to submit jobs to the queue system, so the integration is immediate. On the other hand the Monitoring and Discovery System (MDS) requires a simple modification since it considers the powered-off nodes as unavailable resources. The script that provides the information about the Torque/PBS system to the Globus information system must be modified to publish the switched-off nodes as available resources.

5 USE CASE

Some of the described techniques to reduce power consumption have been tested in a high performance cluster, in order to check the real impact in the power consumption and in the response time of the user submitted jobs.

The cluster is composed by 51 bi-processor nodes Intel Xeon CPU 2.80GHz, interconnected by a SCI network in a 10x5 2D Torus topology. Each node has 2 GB of RAM memory. The front-end node is the access point to the cluster, and the other (50) are used as the working nodes.

The SCI network has an important restriction when applying the green computing techniques, since it is necessary to have all the nodes activated to get the best parallel performance of the network. If one of the nodes is inactive, some of the paths between nodes are lost, and thus decreasing the network performance. This restriction makes necessary switching on/off the whole cluster for avoiding the interference on the user or the running applications.

5.1 Cluster Analysis

To study the impact of the described techniques, it has been made a complete analysis of the cluster, not only at the power consumption level, but also at the usage of the system by the users. Taking the results of this analysis it is possible to get an estimation of the different key factors for the technique described in this paper: the power consumption, the economic saving and the final impact on the user interactivity or running applications.

5.1.1 Power Consumption

In the target cluster, we have measured the power consumption in the main states of the cluster: powered off, idle, and calculating with one and two processors per node.

Figure 1 shows the power on cycle of the 50 working nodes (blue), the execution of a parallel process in all the nodes using one processor per node (green) and finally the execution using two processors per node (red).

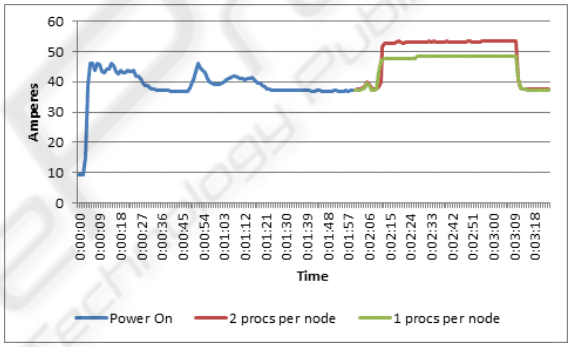

Figure 1: Cluster Power consumption.

In the initial state (all the working nodes switched off) the power consumption is 9.4A. This consumption corresponds to the front-end node, the switches, the KVMs, etc. Once the working nodes are powered on and the consumption arrives a stable state, it reaches 37.2 A of consumption. When the parallel process is started the consumption gets to 48.5 A; but when a parallel process uses two processors per node the consumption increases until 53.3 A.

5.1.2 Usage

To evaluate the cluster usage, the torque LRMS accounting files have been analyzed for about four months.

The total number of jobs was 8806, which have been launched in a 133 days period. The jobs average time was 15'32''.

The cluster usage percentage (at least one job running in the LRMS) was only the 17.38% of the

time. Therefore the remaining 82.62% it is consuming resources without any throughput.

Table 1: Jobs that have to wait the cluster to power on.

Wait Time	4 h	3 h	2 h	1h
Jobs	54	63	72	85

To obtain the best value for the cluster inactivity time to deactivate, it has been analyzed the impact of using a set of values for the user wait time, and but also the power consumption. Table 1 shows the jobs that have to wait the cluster to power on, when considering different values of the time to wait. In any case the number of jobs that have to wait is around 1% of the total jobs. On the other side, the power-on time of the whole cluster is less than 1 min and 30 seconds (as shown in Figure 1).

Table 2: Percentage of time in each state.

Wait Time	Off	Idle	Used
4 h	75,86%	6,77%	17,38%
3 h	76,70%	5,92%	17,38%
2 h	78,11%	4,51%	17,38%
1 h	79,96%	2,66%	17,38%

Table 2 shows the percentages of time that the cluster would have been in the different states, applying the different values for the cluster inactivity time to deactivate. It shows that there is no significant difference among the different values.

5.2 Results

To estimate the power consumption, the behavior of the cluster will be split into three states: powered off, idle, and in use. For the first two cases, the power consumption has been measured in the previous section. In the last case the power consumption has been estimated to be a usage of the 75% of the cluster nodes, using only one processor. The diary power consumption is shown in Table 3.

Table 3: Diary power consumption.

State	**Ampere**	**Volts (mean)**	**kW/Day (A*V*24/1000)**
Off	9,4	230	51,88
Idle	37,2	230	205,34
Used	41	230	226,32

Using this data and the current usage percentage of the cluster, the total yearly power and economic consumption is show in Table 4.

With the restrictions imposed by the usage of the SCI network, in the performed tests it has been applied an algorithm which powers off the whole cluster, with a period of inactivity of 4 hours. The 4 hours period has been selected due to the results obtained in the last section, in order to avoid the maximum number of cluster activations (it restricts the component stress due to the power-on and power-off cycles). Table 5 shows an economic and power consumption estimation applying the described green computing techniques, showing an economic saving of 3.754€, which means a 55% of the whole current expenses.

Table 4: Current power consumption.

State	**Pct**	**kW/Year**	**€/Year***
Off	0%	0	0
Idle	82,62%	59.927	5.453
Used	17,38%	14.353	1.306
TOTAL	100%	74.280	**6.759**

* Cost: 0,091 €/kw. Data obtained from the Ministerio de Industria, Turismo y Comercio del Gobierno de España.

Other important issue to consider is the impact of the cluster activation in the wait time for the users. Some test has been performed to evaluate this time, getting an average of 1'58'', with a maximum time of 2'10''. This time is only the 12.6% of the average time of the measured jobs. Furthermore, as it has been analyzed in the previous section only will affect to the 0.61% of the total jobs, so the effective impact to the user will be minimal.

Table 5: Estimated power consumption.

	Pct	kW/Year	€/Year
Off	75,86%	13.755	1.252
Idle	6,77%	4.908	447
Used	17,38%	14.353	1.306
TOTAL	100%	33.016	**3.005**

Not only it is important the power consumption of the cluster but also it is needed to consider the heat dissipation produced by it, that must be counteracted by a cooling system equipment.

The cooling system of the cluster room in the use case is composed by two compressors, each one consuming about 10 A. Approximately in the warm months (from May to October) the cooling system is working with both compressors whether the studied cluster is switched-on or not. So in these 6 months there is no possibility of decreasing the power consumption of the cooling system. In the cool months (from November to April) we have measured that if the main cluster is switched off only one of the compressors is needed to maintain the temperature of the cluster room. But when the cluster is activated both of them are needed. Using those observations it can be estimated the power

consumption saving produced in the cooling system when applying the green computing techniques selected.

Table 6 shows the cooling system power consumption in the cold months, in the warm months, and the total consumption and compares them with the current expenses. It is produced a saving of 695€, a 19% of the current total cost.

Table 6: Cooling system power consumption comparison.

	Estimated					Curr.
	cold months		warm months			
	Kw	€	Kw	€	€/Year	€/Year
Off	7.641	695	15.283	1.391	2.086	0
Idle	1.363	124	1.363	124	248	3.030
Used	3.500	319	3.500	319	637	367
Total	12.506	1.138	20.093	1.834	**2.972**	**3.667**

6 CONCLUSIONS AND FURTHER WORK

This paper describes techniques to reduce the power consumption on cluster infrastructures, reducing the impact on users. Different aspects to be considered are commented: the selection of the block size to power on/off the cluster, the time to detect the inactivity of the system to deactivate nodes, etc.

In this kind of environments the usage of LRMS enables to easily manage the information about the usage of cluster systems.

The real case described, exposes good results in the implantation of this kind of techniques in a real cluster, decreasing the power consumption in 55% of the total expenses.

We are currently working in the development of other plug-ins, which will address emerging uses of clusters, such as Cloud Computing. In our case, we are debugging plug-ins for Open Nebula and Eucalyptus. In this case, the aim is to maintain a quality of service for virtual machines, with the minimum number of host nodes. In order to determine which nodes are candidate to be powered off and on, we manage concepts such as virtual memory per node, virtual cores per real cores and disk image size.

Our approximation is oriented to cluster management, in which nodes are homogeneous and are interconnected by a private network. But we are also considering the extension to heterogeneous nodes (with different number of cores per node, memory size, disk size, etc.), and evaluating the impact of managing machines in local area networks.

ACKNOWLEDGEMENTS

The authors wish to thank the financial support received from The Spanish Ministry of Education and Science to develop the project "ngGrid - New Generation Components for the Efficient Exploitation of eScience Infrastructures", with reference TIN2006-12890. This work has been partially supported by the Structural Funds of the European Regional Development Fund (ERDF).

REFERENCES

Murugesan S., 2008. Harnessing Green IT: Principles and Practices. In *IEEE IT Professional*, pp 24-33.

Fontan, J. et al, 2008. OpenNebula: The Open Source Virtual Machine Manager for Cluster Computing. In *Open Source Grid and Cluster Software Conference.* San Francisco, CA, USA.

Nurmi, D. et al., 2008. The Eucalyptus Open-source Cloud-computing System. In *Proceedings of Cloud Computing and Its Applications*.

Liang, S. A., 2008. Low cost and high efficiency PC power supply design to meet 80 plus requirements. In *Industrial Technology IEEE International Conference*, pp 1-6.

The 80 PLUS Program, 2009. viewed 1 December 2009. <http://www.80plus.org/>.

Hoelzle, U., Weihl, B., 2010. High-efficiency power supplies for home computers and servers, Google Inc. viewed 22 February, 2010) <http://static.googleusercontent.com/external_content/untrusted/services.google.com/blog_resources/PSU_white_paper.pdf>.

von Laszewski, G. et al, 2009. Power-Aware Scheduling of Virtual Machines in DVFSenabled Clusters. In *IEEE Cluster 2009*. New Orleans.

Feng, W. and Feng, X., 2008. Green supercomputing comes of age. *IT Professional*, vol. 10, no. 1, pp. 17–23.

Feng, W. and Cameron K., 2007. The Green500 List: Encouraging Sustainable Supercomputing. In *IEEE Computer*, pp. 50–55.

Da-Costa, G. et al, 2009. The green-net framework: Energy efficiency in large scale distributed systems. In *High Performance Power Aware Computing Workshop*.

Globus Alliance, 2009. Globus Toolkit, viewed 1 December 2009, <http://www.globus.org/toolkit>.

EGEE, 2009, gLite Lightweight Middleware for Grid Computing. viewed 1 December 2009, <http://www.glite.org>

Lucke, R. W., 2004. Building Clustered Linux Systems, *Prentice Hall*. ISBN - 978-0-13-144853-7.