ANN. MULT. GPU PROG.

EECluster: An Energy-Efficient Tool for managing HPC Clusters

Alberto Cocaña-Fernández[1*], Jose Ranilla[2] and Luciano Sánchez[2]

[*]Correspondence:
cocanaalberto@gmail.com
[1] Departamento de Informática, Universidad de Oviedo, E-33204 Gijón, Spain
Full list of author information is available at the end of the article

Abstract

High Performance Computing clusters have become a very important element in research, academic and industrial communities because they are an excellent platform for solving a wide range of problems through parallel and distributed applications. Nevertheless, this high performance comes at the price of consuming large amounts of energy, which combined with notably increasing electricity prices are having an important economical impact, driving up power and cooling costs and forcing IT companies to reduce operation costs. To reduce the high energy consumptions of HPC clusters we propose a tool, named EECluster, for managing the energy-efficient allocation of the cluster resources, that works with both OGE/SGE and PBS/TORQUE Resource Management Systems (RMS) and whose decision-making mechanism is tuned automatically in a machine learning approach. Experimental studies have been made using actual workloads from the Scientific Modelling Cluster at Oviedo University and the academic-cluster used by the Oviedo University for teaching high performance computing subjects to evaluate the results obtained with the adoption of this tool.

Keywords: Energy-efficient of HPC cluster; Multi-criteria decision making; Evolutionary algorithms

1 Introduction

High Performance Computing (HPC) clusters have become a very important element in research, academic and industrial communities because they are an excellent platform for solving a wide range of problems through parallel and distributed applications [1]. Nowadays, HPC clusters are, in fact, the main architecture for supercomputers (as shown in Top500 architecture distribution[1]) due to the high performance of commodity microprocessors and networks, to the standard tools for high performance distributed computing, and to the lower price/performance ratio [2].

Nevertheless, this high performance comes at the price of consuming large amounts of energy. According to the U.S. Environmental Protection Agency [3], the consumption of data centers in USA was estimated at 61 billion kilowatt-hours (kWh) in 2006 for a total electricity cost of about $4.5 billion. Large energy consumptions combined with notably increasing electricity prices in both EU [4] and USA [5] also have an important economical impact for IT companies, driving up power and cooling costs and forcing them to reduce operation costs [6, 7]. Together with a very significant environmental impact, this economical impact is the main bottleneck constraining the expansion of supercomputing and data centers and, therefore, a powerful motivation to maximize the efficiency of clusters.

Many methods have been proposed within the field of energy-efficient cluster computing following both static and dynamic approaches. Example of dynamic approaches are the Dynamic Voltage and Frequency Scaling (DVFS) technique, used in [8, 9, 10, 11, 12, 13, 14, 15], the software frameworks

[1] June 2014 — TOP500 Supercomputer Sites, http://www.top500.org/lists/2014/06/

to develop energy-efficient applications, such as [16, 17, 18, 19, 20], energy-efficient job schedulers [21, 22] and thermal-aware methods [23, 24].

However, the most relevant technique for this paper is the adaptive resource cluster, which consists mainly in switching on and off cluster compute nodes, adapting to the requested resources at every moment and, therefore, saving energy. First introduced in [25] for Load-Balancing clusters, and also used in [26, 27, 28, 29, 30, 31] and in VMware vSphere[2] and Citrix XenServer hypervisors [3], has also been applied to HPC clusters in [32, 33] or [34].

In these works, the decision-making mechanism for determining the adequate resources (e.g. number of compute node slots) at every moment is based on a simple Knowledge based System (KBS) comprised of *if-then* rules. The KBS constantly monitors requested, idle and available resources. The rule base governing this system is made to depend on certain configuration parameters such as the time of inactivity to shutdown nodes. These parameters are tuned by hand, according to the experience of the administrator.

According to our own experience, these systems are not location-agnostic. In order to obtain the best energy saving, both the set of rules defining the system and the parameters on which the rules depend must be optimized for the actual load scenario.

Otherwise, the results either would interfere with the desired operation of the cluster or would not save as much energy as it could be possible. Because of this, we proposed in [35] a cluster management system, that works with both OGE/SGE and PBS/TORQUE Resource Management Systems (RMS), whose decision-making mechanism shares the same rule set proposed in [33], but whose numerical parameters were obtained by means of a multiobjective evolutionary algorithm in a machine learning approach. The results of the KBS implemented in [35] are suitable for many practical situations and capable of yielding good results in terms of compliance with administrator preferences in all QoS, energy saved and node reconfigurations.

Since the publication of [35], the question has been raised whether the mentioned rule base was optimal or, on the contrary, there exist an alternate definition of the rule base for which the behaviour of the system could be further improved. In this respect, a learning algorithm was recently proposed in [36] that elicits the linguistic definition of a part of the aforementioned knowledge base from data, making it to depend on the cluster behaviour. The learned rules were combined with expert knowledge to form an enhanced cluster management system, whose results improved on reference [35] in a set of benchmark problems. In this paper, the arquitecture of a system that implements this principles is designed, and details about its practical setup are given.

The remainder of the paper is as follows. Section 2 explains the architecture of the solution proposed. Section 3 discuss some use cases. Section 4 shows the experimental results. Section 5 concludes the paper and discusses the future work.

2 Architecture

The solution proposed consists on a service and an administration dashboard, coupled with a Database Management System (DBMS), and deployed over an HPC cluster running a Resource Management System such as OGE/SGE or PBS/TORQUE. The underlying architecture of these clusters combines a master node and several computing nodes. Cluster users access the master node through a remote connection such as SSH and they submit jobs to the RMS. The RMS schedules jobs execution and

[2] VMware Distributed Power Management Concepts and Use,
http://www.vmware.com/files/pdf/Distributed-Power-Management-vSphere.pdf

[3] Citrix XenServer - Efficient Server Virtualization Software,
http://www.citrix.com/products/xenserver/overview.html

when dispatched, jobs are assigned slots among the compute nodes, which are the ones actually running the job. Each slot represents a resource in the cluster, and depending on the RMS configuration the size of the resource ranges from a single CPU core to an entire host.

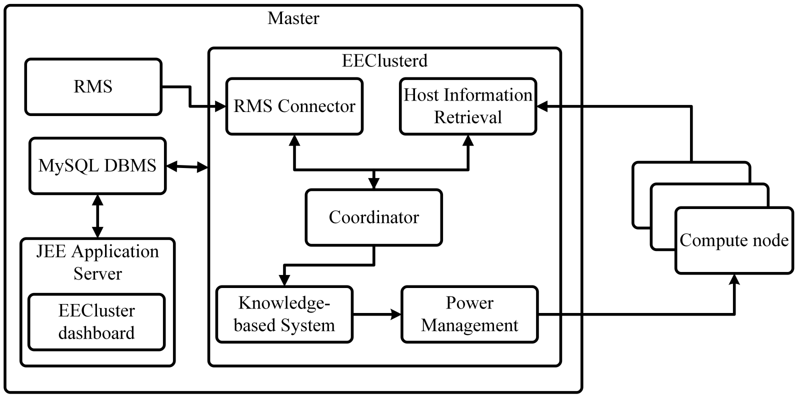


Figure 1 EECluster Tool: System components overview

Figure 1 provides a high-level overview of the system components. The working cycle of the EEClusterd service is the following:

1 Synchronization with the system current status
2 Use the Knowledge-based System to determine if any reconfiguration of the compute nodes must be performed
3 Select the target nodes to reconfigure
4 Power on/off the selected nodes through the Power Management module

In the following subsections the functionality of each system module are explained.

2.1 Synchronization

The synchronization task of the service collects and keeps updated records of the RMS and of every compute node. RMS data includes the cluster parallel environments (OGE/SGE), queues, hosts, users, and completed, queued and running jobs. The service retrieves this information through the RMS connector, which uses multiple command line applications, which vary depending on whether the underlying RMS is OGE/SGE or PBS/TORQUE. In the case of OGE/SGE the synchronization module uses

- *qhost*: node info (architecture, number of processors, sockets, cores, load, total memory, memory in use, total swap memory, swap memory in use) and its relation with each queue (slots in use, reserved slots, state)
- *qconf*: queue info (name, type, number of slots, parallel environments) and parallel environment info (name, allocation rule, number of slots)
- *qacct*: user accounting data (wallclock, utime, stime, cputime, memory, i/o) and job accounting data (jobnumber, queue, jobname, owner, priority, submission time, start time, end time, parallel environment, exit status, utime, stime, cputime)
- *qstat*: running and queued jobs status (jobnumber, priority, jobname, owner, state, submission time, queue, start time, slots)

As for PBS/TORQUE

- *pbsnodes*: node info (name, state, number of processors, properties, type, status)

- *qstat*: running and queued jobs status (job ID, username, queue, jobname) and queue info (type, priority, total jobs, max running jobs, max job walltime, slots)
- *PBS/TORQUE accounting records*: accounting records for completed jobs in the TORQUEROOT/server_priv/accounting/TIMESTAMP directory (jobnumber, user, queue, jobname, slots, walltime, start time, end time)

Regarding each host, the Host Information Retrieval EECluster module collects information about

- CPUs (model, frequency, cache) through the */proc/cpuinfo* file
- RAM memory through the */proc/meminfo* file
- GPUs (name, % utilization, temperature, fan speed, power usage, etc.) through NVIDIA System Management Interface
- Intel Xeon Phi coprocessor (model, active cores, frequency, memory, temperature) through *micinfo*
- PSU power usage through the IPMI interface
- Host MAC address through *arp*

Additionally, user information is accessed through the */etc/passwd* file.

2.2 Node states

In order to identify the situation of each compute node in regard to both the EEClusterd service and the RMS, a number of states are defined. These states, along with the transitions between them, are represented in Figure 2.

When a node is sent a power-on command by the EEClusterd service, it changes its state to Starting. As soon as the node is powered on, it changes its state to Running idle. Once a job is dispatched to a given node, that node is set as Executing job, returning to Running idle whenever the node completes every assigned job. When a node is appointed to be powered off, its state changes to Unavailable, meaning that the RMS must attempt to disable the node prior to the ultimate power-off command. This is done in order to assure that no job is assigned to the node while it shuts down. If the RMS cannot disable the node, the shutdown is rolled back, having the node powered-off otherwise.

Whenever a shutdown or power-on command is issued by the EEClusterd service, a timeout period starts. If the timeout expires before the operation is completed, the node state is changed back to the previous state.

2.3 Knowledge-based System

The key component of this architecture is a KBS implementing the decision-making mechanism that determines how many of the cluster resources must be on at every moment. The KBS used is a Hybrid Genetic Fuzzy System (HGFS) that combines both a fuzzy and a non-fuzzy set of rules and where the fuzzy part is learned by means of a genetic-based machine learning (GBML) multiobjective evolutionary algorithm (MOEA). Further information on this system can be found in [36].

2.4 Node selection

Once determined how many slots must be powered on/off, the next step is determine which specific nodes will be reconfigured. It is important to remark that only idle nodes would be powered off. The selection process involves two values: the node efficiency and the node timestamp of the last timed out.

The first one is calculated as $\frac{\mathrm{GFLOPS}}{\mathrm{Watts}}$, and the latter indicates the time of the last failure to power on/off upon request. In the first place, hosts are split by whether they succeeded or failed to comply with the last order. Those that succeeded are sorted according to their efficiency so that powered-on

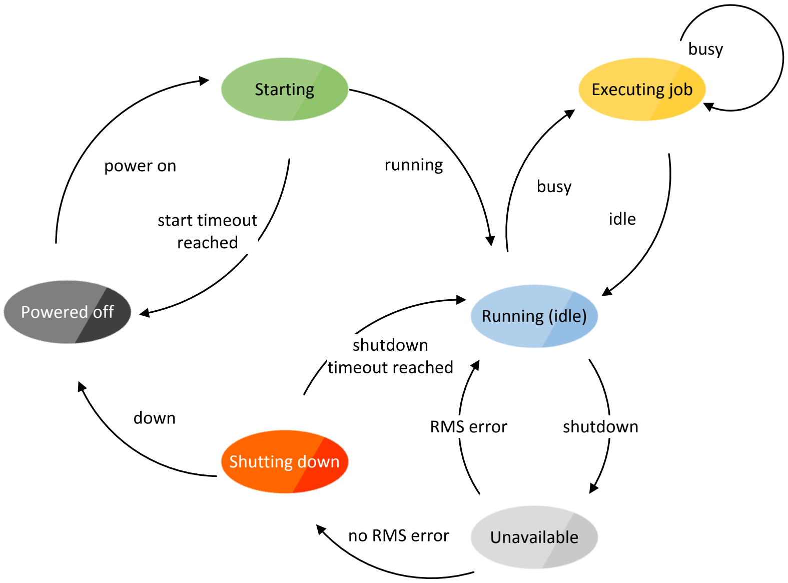


Figure 2 Node state diagram

nodes are the most efficient and powered-off nodes are the least efficient ones. Conversely, those that failed are sorted according to the timestamps of their failures; those with the earliest values are always chosen. This mechanism allows the system to continuously iterate through the potentially malfunctioning nodes, thus increasing the possibility of finding a repaired one.

2.5 Power Management

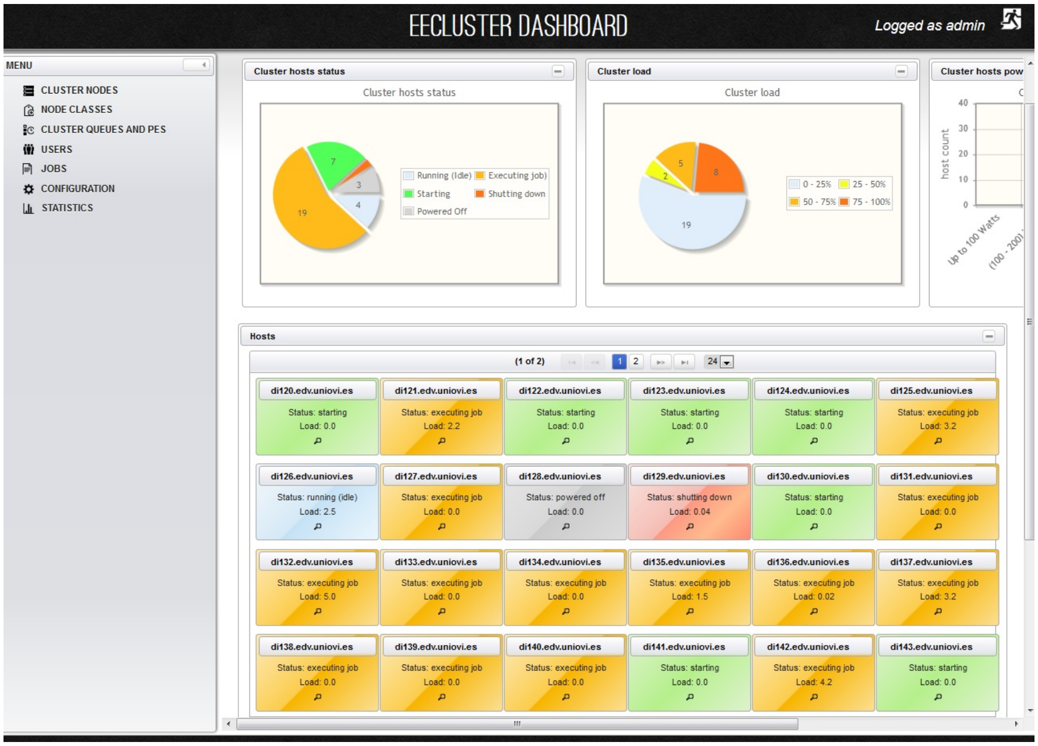
The Power Management module is the responsible for switching on/off the nodes appointed by the Knowledge-based System. This can be done either using Ethernet cards or IPMI cards (Intelligent Platform Management Interface). With Ethernet cards, the power on order is carried out by sending the Ethernet WOL (Wake On Lan) *magic packet* using the *ether-wake* program. It is important to point out that not all compute nodes will necessarily be in the same network, so the Power Management module must choose the correct network interface when sending the *magic packet*. This is configured in the dashboard. In order to shutdown a node, this can be done by simply executing the command *poweroff*. Another important remark is that for WOL to work, it must be enabled in the Ethernet card, or it will ignore the packet. In order to assure that a powered off host can be powered on again, prior to each power off, the *ethtool* is used to enable WOL. If the host has an IPMI card the Power Management module can use it to power it on/off. This is done using tools such as *ipmiutil*.

2.6 Administration dashboard

The administration dashboard is a Web application that displays current cluster status including nodes, queued and running jobs, host information, node classes, queues, parallel environments, jobs, users, statistics, charts... (see, for example, Figure 3), and also allows the cluster administrator to switch on/off nodes manually and configure the system.

3 Use cases

The EECluster tool has been tested in various environments including research, professional and academic clusters.



di120.edv.uniovi.es Status: starting Load: 0.0	di121.edv.uniovi.es Status: executing job Load: 2.2	di122.edv.uniovi.es Status: starting Load: 0.0	di123.edv.uniovi.es Status: starting Load: 0.0	di124.edv.uniovi.es Status: starting Load: 0.0	di125.edv.uniovi.es Status: executing job Load: 3.2
di126.edv.uniovi.es Status: running (idle) Load: 2.5	di127.edv.uniovi.es Status: executing job Load: 0.0	di128.edv.uniovi.es Status: powered off Load: 0.0	di129.edv.uniovi.es Status: shutting down Load: 0.04	di130.edv.uniovi.es Status: starting Load: 0.0	di131.edv.uniovi.es Status: executing job Load: 0.0
di132.edv.uniovi.es Status: executing job Load: 5.0	di133.edv.uniovi.es Status: executing job Load: 0.0	di134.edv.uniovi.es Status: executing job Load: 0.0	di135.edv.uniovi.es Status: executing job Load: 1.5	di136.edv.uniovi.es Status: executing job Load: 0.02	di137.edv.uniovi.es Status: executing job Load: 3.2
di138.edv.uniovi.es Status: executing job Load: 0.0	di139.edv.uniovi.es Status: executing job Load: 0.0	di140.edv.uniovi.es Status: executing job Load: 0.0	di141.edv.uniovi.es Status: starting Load: 0.0	di142.edv.uniovi.es Status: executing job Load: 4.2	di143.edv.uniovi.es Status: starting Load: 0.0

Figure 3 EEcluster dashboard

The academic cluster consists on 34 nodes arranged in three queues using OGE/SGE as RMS. The nodes include both PCs with Intel Core i3-2100 CPUs @ 3.10 GHz and 4 GB of RAM, and PCs with Intel Core i7 930 CPUs @ 2.80 GHz, 12 GB of RAM and CUDA-enabled NVIDIA GeForce GTX 480 cards. These nodes do not include an IPMI card so Ethernet interface is used to switch them on. Also, the nodes in each queue are in different networks which requires explicit configuration in the dashboard to specify the network interface that must be used in the master node to send the WOL packet to each node. The main purpose of this cluster is to allow students to learn and experiment with multicore, distributed and GPU computing.

The research cluster is used by the Parallel Computing and Information Retrieval group [4] at the University of Oviedo and consists on 4 computational nodes arranged in two queues running OGE/SGE as RMS. The nodes include two PowerEdge 1950 servers and one PowerEdge 2950, all with one Intel Xeon CPU E5420 @ 2.5 GHz and 16 GB of RAM, and one ASUS server with two Intel Xeon CPU E5-2650 @ 2.0 GHz, 64 GB of RAM, one NVIDIA Tesla K40m, one NVIDIA Tesla K20m and one Intel Xeon Phi 5110P. Every node in this cluster uses Scientific Linux as Operating System and features an IPMI card. Also, every hosts is within the same network. The main purpose of this cluster is to support research in the field of algorithm parallelization in multicore, distributed, GPUs and Intel MIC environments, and also chemical computational modelling.

The professional, and reseach, cluster is the Scientific Modelling Cluster of the University of Oviedo (CMS) which consists of three independent computing clusters and five transversal queues using PBS as a Resource Management Systems (RMS). A full description of CMS is given in its web site (http://cms.uniovi.es).

[4] pirweb.edv.uniovi.es

The deployment of EECluster tool in each case was quite plain and simple. It required the installation of MySQL DBMS and a GlassFish application server. Secondly the web application (EECluster dashboard) was deployed over the GlassFish server and init.d scripts were created to automatize the start-up upon system boot of both the EEClusterd service and the GlassFish server. Finally IPMI cards and network interfaces were configured through the dashboard for every node.

4 Experimental results

To measure the effect of the EECluster tool in a real world environment three metrics have been defined: the quality of service, the energy saved and the node recofigurations. These metrics are calculated by running a simulation of a cluster workload for a given set of n jobs, where the j-th job ($j = 1 \dots n$) is scheduled to start at time tsch$_j$, but effectively starts at time ton$_j$ and stops at time toff$_j$, the quality of service in a HPC cluster reflects the amount of time that each job has to wait before is assigned its requested resources. Once the job starts its execution, it will not be halted, thus we focus only on its waiting time. Because jobs do not last the same amount of time, their waiting in the queue is better expressed as a ratio considering their execution time. Finally, due to the potential existence of outlier values, the 90 percentile is used instead of average:

$$\mathrm{QoS} = \min \left\{ p : ||\{ j \in 1 \dots n : \frac{\mathrm{ton}_j - \mathrm{tsch}_j}{\mathrm{toff}_j - \mathrm{ton}_j} \leq p \}|| > 0.9\, n \right\} \qquad (1)$$

where $||A||$ is the cardinality of the set A.

The energy saved is measured as the sum of the amount of seconds that each node has been powered off. Let c be the number of nodes, let state(i, t) be 1 if the i-th node $(i = 1 \dots c)$ is powered at time t and 0 otherwise. Lastly, let the time scale be the lapse between tini$=\min_j\{\mathrm{sch}_j\}$ and tend$=\max_j\{\mathrm{toff}_j\}$. Then,

$$\text{Energy saved} = c \cdot (\mathrm{tend} - \mathrm{tini}) - \sum_{i=1}^{c} \int_{\mathrm{tini}}^{\mathrm{tend}} \mathrm{state}(i, t)\mathrm{d}t. \qquad (2)$$

The node reconfigurations is the number of times that a node has been powered on or off. Let nd(i) the number of discontinuties of the function $\mathrm{state}(i, t)$ in the time interval $t \in (\mathrm{tini}, \mathrm{tend})$:

$$\text{Reconfigured nodes} = \sum_{i=1}^{c} \mathrm{nd}(i) \qquad (3)$$

In particular, the experimental setup is based on actual workload of the aforementioned Scientific Modelling Cluster of the University of Oviedo spanning 22 months with a total of 2907 jobs. For both training and testing, a cluster simulator has been developed so that every model can be evaluated in the criteria previously described.

Three configurations for the KBS have been tested each one corresponding to a different set of administrator preferences. The first (labelled as HGFS QoS 0.0) priorities QoS above all other criteria, using energy savings just to break ties between QoS and node reconfiguration to break ties between both energy savings. The second (labelled as HGFS QoS 0.1) seeks the best energy savings as long the QoS value is below or equal to 0.1. The third one (labelled as HGFS QoS 0.5) is similar to the second, but rising the QoS boundary to 0.5. The holdout method was used for validation, with a 70-30% split in training and test.

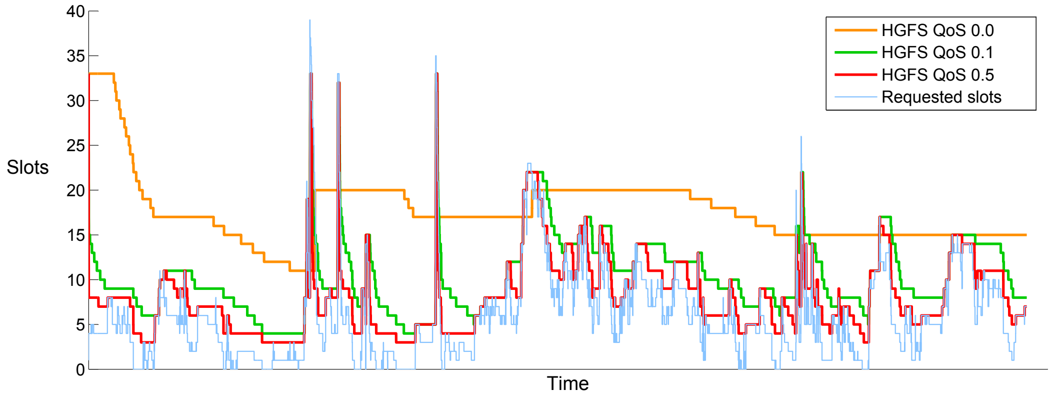


Figure 4 Cluster simulation trace obtained in the experiment for the test set

	Training set		
	QoS	Energy saved(s)	Reconfigurations
HGFS QoS 0.0	0.00	8.84E+08	75
HGFS QoS 0.1	0.09	1.13E+09	627
HGFS QoS 0.5	0.48	1.19E+09	929

Table 1 Experiment results for the training set

As shown in Tables 1 and 2, the HGFS used as the decision-making mechanism in the EECluster tool can produce very different behaviours depending on the preferences of the cluster administrator, from having no impact on the QoS to a controlled increase on the jobs waiting times and achieving extraordinary energy savings. In other words, the higher impact on the QoS is allowed, the higher energy savings are reached. This is graphically represented in Figure 4, which shows the evolution over time of the aggregated requested slots by the jobs and the slots powered on by each configuration.

	Test set		
	QoS	Energy saved(s)	Reconfigurations
HGFS QoS 0.0	0.00	2.41E+08	42
HGFS QoS 0.1	0.07	3.54E+08	361
HGFS QoS 0.5	0.19	3.90E+08	590

Table 2 Experiment results for the test set

5 Concluding remarks and future work

HPC clusters need large amounts of energy to obtain their high performance. Among the different approaches that have been proposed within the field of energy-efficient computing, intelligent cluster management systems have been successfully applied to determine the number of compute node slots at every moment on the basis of requested, idle and available resources. The system studied in this paper is based upon a compact set of decision rules. Some of the parameters defining this knowledge base are tuned by hand, according to the experience of the administrator, and others are learned from logged data. An arquitecture of a cluster management system implementing this structure has been proposed. Details about the practical setup were given, including some use cases and a numerical assessment. It was concluded that sensible energy savings can be achieved though the use of intelligent cluster management systems.

It is also expected that these savings are subjected to the predictability degree of the cluster load. In future works this point will be addressed. A balance will be sought between a purely reactive strategy, appropriate for erratic loads, and a predictive management, best for foreseeable loads.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

HPC clusters are the main architecture of supercomputing. These clusters need large amounts of energy to obtain their high performance. Therefore, many approaches have been proposed within the field of energy-efficient computing. One technique is the adaptive resource cluster, which consists in switching on and off cluster compute nodes. Here the decision-making mechanism is based on a simple Knowledge based System comprised of *if-then* rules that depend on some parameters. According to our own experience, these systems are not location-agnostic so in this article we propose a novel approach where both the set of rules and their parameters are automatically optimized. To do this a Hybrid Genetic Fuzzy System, which combines both a fuzzy and a non-fuzzy set of rules, is used. The fuzzy part is learned by means of a genetic-based machine learning multiobjective evolutionary algorithm.

Acknowledgements

This work has been partially supported by "Ministerio de Economía y Competitividad" from Spain/FEDER under grants TEC2012-38142-C04-04 and TIN2011-24302.

Author details

[1] Departamento de Informática, Universidad de Oviedo, E-33204 Gijón, Spain. [2] Departamento de Informática, Universidad de Oviedo, E-33204 Gijón, Spain.

References

1. Buyya, R., Jin, H., Cortes, T.: Cluster computing. Future Generation Computer Systems **18**(3), (2002). doi:10.1016/S0167-739X(01)00053-X
2. Yeo, CheeShin and Buyya, Rajkumar and Pourreza, Hossein and Eskicioglu, Rasit and Graham, Peter and Sommers, F.: Cluster Computing: High-Performance, High-Availability, and High-Throughput Processing on a Network of Computers. In: Zomaya, A. (ed.) Handbook of Nature-Inspired and Innovative Computing, pp. 521–551. Springer, ??? (2006). doi:10.1007/0-387-27705-6_16. http://dx.doi.org/10.1007/0-387-27705-6_16
3. U.S. Environmental Protection Agency: Report to Congress on Server and Data Center Energy Efficiency Public Law 109-431. Technical report, ENERGY STAR Program (2007). http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf?7e9c-bbd7
4. Eurostat: Electricity and natural gas price statistics - Statistics Explained (2013). http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Electricity_and_natural_gas_price_statistics#Further_Eurostat_information Accessed 07/04/14
5. EIA: Electric Power Monthly - Energy Information Administration. http://www.eia.gov/electricity/monthly/ Accessed 07/04/14
6. Ebbers, Mike Archibald, M., da Fonseca, C.F.F., Griffel, M., Para, V., Searcy, M.: Smarter Data Centers: Achieving Greater Efficiency. Technical report, IBM Redpaper (2011). http://www.redbooks.ibm.com/abstracts/redp4413.html
7. The Economist Intelligence Unit: IT and the environment A new item on the CIOs agenda? Technical report, The Economist (2007). http://www-03.ibm.com/services/ca/fr/green/pdf/SOLUTION_IT_it_and_the_environment.pdf
8. Hsu, C.-H., Kremer, U.: The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction. ACM SIGPLAN Notices **38**(5), 38 (2003). doi:10.1145/780822.781137
9. Hsu, C.-H., Feng, W.-c.: A Power-Aware Run-Time System for High-Performance Computing. In: ACM/IEEE SC 2005 Conference (SC'05), pp. 1–1. IEEE, ??? (2005). doi:10.1109/SC.2005.3. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1559953
10. Freeh, V.W., Lowenthal, D.K., Pan, F., Kappiah, N., Springer, R., Rountree, B.L., Femal, M.E.: Analyzing the Energy-Time Trade-Off in High-Performance Computing Applications. IEEE Transactions on Parallel and Distributed Systems **18**(6), 835–848 (2007). doi:10.1109/TPDS.2007.1026
11. Lim, M., Freeh, V., Lowenthal, D.: Adaptive, Transparent Frequency and Voltage Scaling of Communication Phases in MPI Programs. In: ACM/IEEE SC 2006 Conference (SC'06), pp. 14–14. IEEE, ??? (2006). doi:10.1109/SC.2006.11. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4090188
12. Cheng, Y., Zeng, Y.: Automatic Energy Status Controlling with Dynamic Voltage Scaling in Power-Aware High Performance Computing Cluster. In: 2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 412–416. IEEE, ??? (2011). doi:10.1109/PDCAT.2011.24. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6118547
13. Ge, R., Feng, X., Feng, W.-c., Cameron, K.W.: CPU MISER: A Performance-Directed, Run-Time System for Power-Aware Clusters. In: 2007 International Conference on Parallel Processing (ICPP 2007), pp. 18–18. IEEE, ??? (2007). doi:10.1109/ICPP.2007.29. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4343825
14. Huang, S., Feng, W.: Energy-Efficient Cluster Computing via Accurate Workload Characterization. In: 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 68–75. IEEE, ??? (2009). doi:10.1109/CCGRID.2009.88. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5071856
15. Chetsa, G.L.T., Lefrvre, L., Pierson, J.-M., Stolf, P., Da Costa, G.: A Runtime Framework for Energy Efficient HPC Systems without a Priori Knowledge of Applications. In: 2012 IEEE 18th International Conference on Parallel and Distributed Systems, pp. 660–667. IEEE, ??? (2012). doi:10.1109/ICPADS.2012.94. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6413638
16. Alonso, P., Badia, R.M., Labarta, J., Barreda, M., Dolz, M.F., Mayo, R., Quintana-Orti, E.S., Reyes, R.: Tools for Power-Energy Modelling and Analysis of Parallel Scientific Applications. In: 2012 41st International Conference on Parallel Processing, pp. 420–429. IEEE, ??? (2012). doi:10.1109/ICPP.2012.57. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6337603

17. Schubert, S., Kostic, D., Zwaenepoel, W., Shin, K.G.: Profiling Software for Energy Consumption. In: 2012 IEEE International Conference on Green Computing and Communications, pp. 515–522. IEEE, ??? (2012). doi:10.1109/GreenCom.2012.86. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6468359
18. Freeh, V.W., Lowenthal, D.K.: Using multiple energy gears in MPI programs on a power-scalable cluster. In: Proceedings of the Tenth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming - PPoPP '05, p. 164. ACM Press, New York, USA (2005). doi:10.1145/1065944.1065967. http://dl.acm.org/citation.cfm?id=1065944.1065967
19. Li, D., Nikolopoulos, D.S., Cameron, K., de Supinski, B.R., Schulz, M.: Power-aware MPI task aggregation prediction for high-end computing systems. In: 2010 IEEE International Symposium on Parallel & Distributed Processing (IPDPS), pp. 1–12. IEEE, ??? (2010). doi:10.1109/IPDPS.2010.5470464. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5470464
20. Xian, C., Lu, Y.-H., Li, Z.: A programming environment with runtime energy characterization for energy-aware applications. In: Proceedings of the 2007 International Symposium on Low Power Electronics and Design - ISLPED '07, pp. 141–146. ACM Press, New York, USA (2007). doi:10.1145/1283780.1283811. http://dl.acm.org/citation.cfm?id=1283780.1283811
21. Zong, Z., Ruan, X., Manzanares, A., Bellam, K., Qin, X.: Improving Energy-Efficiency of Computational Grids via Scheduling. In: Antonopoulos, N., Exarchakos, G., Li, M., Liotta, A. (eds.) Handbook of Research on P2P and Grid Systems for Service-Oriented Computing. IGI Global, ??? (2010). Chap. 22. doi:10.4018/978-1-61520-686-5. http://www.igi-global.com/chapter/improving-energy-efficiency-computational-grids/40816/
22. Zong, Z., Nijim, M., Manzanares, A., Qin, X.: Energy efficient scheduling for parallel applications on mobile clusters. Cluster Computing **11**(1), 91–113 (2007). doi:10.1007/s10586-007-0044-5
23. Bash, C., Forman, G.: Cool job allocation: measuring the power savings of placing jobs at cooling-efficient locations in the data center, p. 29. USENIX Association, ??? (2007). http://dl.acm.org/citation.cfm?id=1364385.1364414
24. Tang, Q. and Gupta, S. K S and Varsamopoulos, G.: Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers: A Cyber-Physical Approach. IEEE Transactions on Parallel and Distributed Systems **19**(11), 1458–1472 (2008). doi:10.1109/TPDS.2008.111
25. Pinheiro, E., Bianchini, R., Carrera, E.V., Heath, T.: Load balancing and unbalancing for power and performance in cluster-based systems. In: Workshop on Compilers and Operating Systems for Low Power, vol. 180, pp. 182–195 (2001). Barcelona, Spain
26. Das, R., Kephart, J.O., Lefurgy, C., Tesauro, G., Levine, D.W., Chan, H.: Autonomic multi-agent management of power and performance in data centers, 107–114 (2008)
27. Elnozahy, E.N., Kistler, M., Rajamony, R.: Energy-efficient server clusters, 179–197 (2002)
28. Berral, J.L., Goiri, I.n., Nou, R., Julià, F., Guitart, J., Gavaldà, R., Torres, J.: Towards energy-aware scheduling in data centers using machine learning. In: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking - e-Energy '10, p. 215. ACM Press, New York, USA (2010). doi:10.1145/1791314.1791349. http://dl.acm.org/citation.cfm?id=1791314.1791349
29. Lang, W., Patel, J.M., Naughton, J.F.: On energy management, load balancing and replication. ACM SIGMOD Record **38**(4), 35 (2010). doi:10.1145/1815948.1815956
30. Garcia, D.F., Entrialgo, J., Garcia, J., Garcia, M.: A self-managing strategy for balancing response time and power consumption in heterogeneous server clusters. In: 2010 International Conference on Electronics and Information Engineering, vol. 1, pp. 537–541. IEEE, ??? (2010). doi:10.1109/ICEIE.2010.5559691. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5559691
31. Llamas, R.M., Garcia, D.F., Entrialgo, J.: A Technique for Self-Optimizing Scalable and Dependable Server Clusters under QoS Constraints. In: 2012 IEEE 11th International Symposium on Network Computing and Applications, pp. 61–66. IEEE, ??? (2012). doi:10.1109/NCA.2012.29. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6299127
32. Alvarruiz, F., de Alfonso, C., Caballer, M., Hernández, V.: An Energy Manager for High Performance Computer Clusters. In: 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications, pp. 231–238. IEEE, ??? (2012). doi:10.1109/ISPA.2012.38. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6280297
33. Dolz, M.F., Fernández, J.C., Iserte, S., Mayo, R., Quintana-Ortí, E.S., Cotallo, M.E., Díaz, G.: EnergySaving Cluster experience in CETA-CIEMAT. In: 5th Iberian GRID Infrastructure Conference, Santander (2011)
34. Xue, Z., Dong, X., Ma, S., Fan, S., Mei, Y.: An Energy-Efficient Management Mechanism for Large-Scale Server Clusters. In: The 2nd IEEE Asia-Pacific Service Computing Conference (APSCC 2007), pp. 509–516. IEEE, ??? (2007). doi:10.1109/APSCC.2007.54. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4414502
35. Cocaña-Fernández, A., Ranilla, J., Sánchez, L.: Energy-Efficient Allocation of Computing Node Slots in HPC Clusters through Evolutionary Multi-Criteria Decision Making. In: Proceedings of the 14th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2014, pp. 318–330 (2014)
36. Cocaña-Fernández, A., Ranilla, J., Sánchez, L.: Energy-Efficient Allocation of Computing Node Slots in HPC Clusters through Parameter Learning and Hybrid Genetic Fuzzy System Modelling. The Journal of Supercomputing (2014). doi:10.1007/s11227-014-1320-9