

HPC Environment Management: New Challenges in the Petaflop Era

Jonas Dias, Albino Aveleda

Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
{jonas, bino}@nacad.ufrj.br

Abstract. High Performance Computing (HPC) is becoming much more popular nowadays. Currently, the biggest supercomputers in the world have hundreds of thousands of processors and consequently may have more software and hardware failures. HPC centers managers also have to deal with multiple clusters from different vendors with their particular architectures. However, since there are not enough HPC experts to manage all the new supercomputers, it is expected that non-experts will be managing those large clusters. In this paper we study the new challenges to manage HPC environments containing different clusters with different sizes and architectures. We review available tools and present LEMMing [1], an easy-to-use open source tool developed to support high performance computing centers. LEMMing integrates machine resources and the available management and monitoring tools on a single point of management.

Keywords: HPC management, cluster management, management tools, resource management

1 Introduction

High performance computing (HPC) systems are typically found on universities and research centers. Broadly used to process large scale scientific experiments and complex simulations, they are also very popular among enterprises. About 62.40% of the 500 most powerful supercomputers in the world are in the industry sector [2].

The millionaire supercomputing market is very active. The competition between governments, research centers and huge enterprises encourages investments to break scientific computing barriers. Recently, the so called PetaFlop barrier has been broken [3]. In other words, it is the capacity to sustain more than 10^{15} floating point operations per second. That was achieved using supercomputers with more than 250,000 processing cores. They are on the top of the list of the 500 biggest supercomputers in the world. Since 1993, the TOP500.org portal collects information about the 500 world's most powerful systems. It publishes a ranking twice a year with the performance of many supercomputers measured by the LINPACK benchmark [4]. Observing the TOP500 statistics, it is possible to follow the evolution and trends of the HPC technology.

Figure 1 shows the historical growth of the number of processors per system to obtain more powerful computers. Today, the vast majority of the TOP500 supercomputers have more than two thousand processors. Nowadays, the

development of new technologies produces better hardware. However, the greater the number of nodes in a supercomputer, the greater is the chance of a failure in any of these nodes. Thus, management processes need to be faster and easier to handle failures quickly.

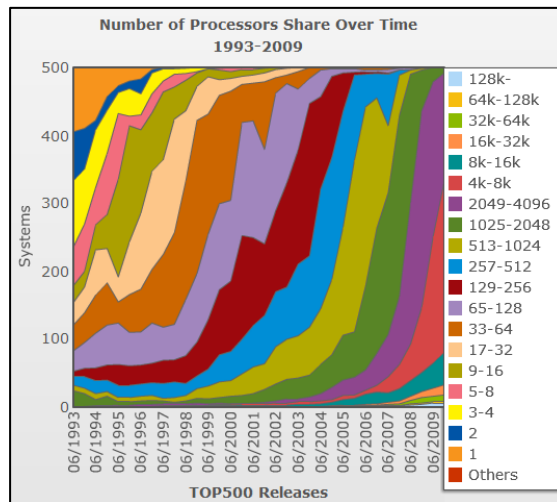


Figure 1 - Growth of the number of processors on supercomputers. (Source: TOP500.org historical charts)

Most cluster management and monitoring tools can handle huge supercomputers. However, they usually do not present very organized information and they are not user friendly. For example, most tools present the listing of the compute nodes of a cluster sequentially. On small clusters, it is not a problem, since it is easy to check several nodes listed on a screen. But on huge supercomputers, to find a single node on a sequential list with hundreds or even thousands of nodes is not practical. To solve this problem, the nodes can be easily organized on a hierarchy, but no tool has ever presented such solution. The user interface usability may not be so important for an expert manager, but with the spread of HPC, many supercomputers are managed by non-expert administrators, which may need an easy-to-use interface.

To manage a supercomputer includes ensuring the full access to its functionalities and resources, handling defects and problems as fast as possible. A queue system may be offered to distribute the access between users equally or due to some internal policy. For example, on an engineering company, the engineer group may have greater priority over the development group. It is also interesting to have monitoring tools to show the status of the machines over time and store critical scenarios that may be repeated. HPC centers usually have a complex and non uniform infrastructure with different types of supercomputers. They use a variety of tools, including their own, to manage their HPC environment. It may be very hard to keep control of many resources using multiple tools. Each tool has its own user interface, web address and features. Often the system administrator may have to deal with more than two different tools to accomplish a single management process. For example, the network monitoring software alerts that a cluster is reporting an error regarding some of its nodes being down. The manager may need to access the specific monitoring cluster

tool and, after detecting where the problem is, access the cluster server through a secure shell to solve the problem through command lines.

There are few software for analysis and administration of HPC centers in the market. The available solutions are mostly proprietary and usually do not integrate with software from other vendors. Therefore, a center with heterogeneous machinery may have to deal with multiple software systems. Some products also lacks in usability, especially when it is necessary to manage a huge number of nodes.

We believe that the development of an open source tool for HPC environments with improved usability and integration is needed. Thus, we propose LEMMING, an open source tool that let you administrate a HPC environment with multiple clusters and servers through a single Rich Internet Application (RIA). LEMMING is meant to be a single point of management. It supports deployment, management and monitoring of clusters and servers focused, mainly, on high performance computing, taking advantage of other available tools. It also lets the manager to customize the interface with the preferred resources, widgets and external tools.

The remainder of this paper is organized as follows. Section 2 provides an overall comparison between different types of management and monitoring tools for supercomputers. Section 3 describes the LEMMING project and section 4 concludes and discusses future work.

2 Available Tools

Since we are proposing a new tool that takes advantage of other available tools to deploy, manage and monitor supercomputers, it is necessary to discuss these tools in order to show the role of our proposal. It is not possible to give many details and discuss every tool. We choose the ones we believe are the most important, according to our experience. Proprietary software is not referenced directly in accordance to trademark law restrictions.

2.1 Deployment Tools

There are many possibilities to setup a cluster but there are few tools that offer an easy way to do it. The most popular open source ones are OSCAR (Open Source Cluster Application Resource) [5] and Rocks [6]. Both integrate other free tools and offer an easy manner to install the cluster system. OSCAR is installed after the operating system and runs a graphical user interface (GUI) that configures and deploys the cluster in several steps. Alternatively, Rocks installation involves the Linux operating system and a customizable set of other packages. OSCAR and Rocks install useful tools for management and monitoring. Rocks has a group of tools and commands that simplifies management tasks like: components listing, node addition and removal and change in properties like IP address, hostname, gateway and DNS server. OSCAR supports these operations on the GUI or through scripts.

Another important open source solution for cluster deployment is xCAT (Extreme Cloud Administration Toolkit) [7]. It provides support for the deployment and administration of high scale systems and has plenty of useful scripts. The focus of xCat is not, actually, the ease of installation and configuration but robustness and scalability. It does not mean that OSCAR and Rocks are not robust and do not have

great scalability. It just means that xCAT can be harder to install and OSCAR and Rocks may need some adjustments after installation for improved results. Table 1 summarizes a comparison among OSCAR, Rocks and xCAT involving important characteristics for cluster deployment.

Table 1: Comparison of different cluster deployment tools.

	<i>Cluster Installation</i>	<i>Node Adding</i>	<i>MPI</i>	<i>Queuing System</i>	<i>Monitoring Tool</i>
OSCAR	GUI	GUI + Network listening	Yes	Yes	Yes
Rocks	GUI	UI + Network listening	Yes	Yes	Yes
xCAT	Command Line	Command Line + Manual Adding	No	No	No

Both OSCAR and Rocks provide a GUI to deploy the cluster environment. OSCAR launches a GUI with a step-by-step installation window on the graphical interface of the cluster Linux server. Rocks installation is integrated with the operating system graphical installer. xCAT is completely installed through command line and requires the manual editing of SQLite tables to set up cluster configurations and available resources. OSCAR installation GUI includes a node adding step. Rocks provides an ASCII user interface to handle node adding. OSCAR and Rocks have the option to listen the network to automatically add new booted nodes. On xCAT the nodes must be manual added editing a set of database tables using command line.

OSCAR and Rocks can install MPI libraries (OpenMPI [8] or MPICH [9]), a workload and resource-management tool (shortened as Queuing System on the table) such as Sun Grid Engine [10] or TORQUE [11]. They both can also install a monitoring tool, usually Ganglia [12]. On xCAT, these tools must be installed and integrated separately. All the mentioned tools provide good documentation.

2.2 Monitoring Tools

There are some popular network monitoring tools that are also very useful for HPC environment management like Nagios [13] and Cacti [14]. Nagios monitors the status of the environment hosts and possibly some of their services. If some services or a host stop working, Nagios can alert the administrators. Cacti uses RRDTool [15] to show statistical data on graphics about CPU load, network bandwidth, packages transmission, temperatures and others metrics. Together, they provide a global view of the available resources and services on the whole network. Both softwares are prepared to be extended and customized through plugins with specialized functionalities. This level of customization is very interesting since different HPC centers may have specific necessities, like the monitoring of its critical services or some specific cluster.

Another useful and very popular monitoring tool is Ganglia [12]. Ganglia shows many monitoring metrics about the cluster nodes on a web page. It also uses

RRDTool for data storage and visualization. It plots graphics of the metrics over time. Ganglia uses XML to report the available metrics for each node of the cluster or grid, so it is easy to integrate Ganglia with other applications. Table 2 gives a comparison among Cacti, Ganglia, and Nagios monitoring tools.

Table 2: Comparison of the mentioned monitoring tools

	Web Based	RIA	Send Alert	Plugins	Monitoring focus
Cacti	Yes	No	No	Yes	Network
Ganglia	Yes	No	No	No	Cluster/Grid
Nagios	Yes	No	Yes	Yes	Network

All the described monitoring tools are web based, but none is a rich internet application (RIA). Only Nagios send alerts if something goes wrong on the network. However, Cacti has plug-ins that also do it. Nagios and Cacti can be extended by plug-ins. Ganglia only lets the user to add extra metrics to monitor the cluster. Ganglia monitoring focus differs from Nagios and Cacti, since it is designed to monitor a single cluster or grid, while Nagios and Cacti are prepared to manage a whole network. However, a supercomputer or a grid can also be considered a private LAN. Anyway, this focus difference implies that Ganglia provides more detailed information about the cluster or grid nodes while Nagios and Cacti may show an overview of the network status.

2.3 Proprietary Solutions

Proprietary cluster solutions normally use a set of open source systems and tools (Linux Operating System, Ganglia monitoring tool, free compilers and MPI Libraries). Many vendors use OSCAR, Rocks or xCAT to handle its base cluster installation. In addition, they tune the cluster configuration and install their proprietary environments to administrate the cluster. These tools are designed for machines from a specific vendor and do not deal with other types of hardware. Actually, it is quite common that different models of machines, from the same vendor, came with distinct software to handle administration tasks. Sometimes there is a separate tool for each part of the system, for example: a tool to handle the storage system, other to handle the nodes of the cluster and yet others to handle users, queue systems and system modules.

There is some proprietary software that provides integration. They offer the management of varied equipment through a single graphical user interface. The difficulties found on this software are, commonly, the usability and incompatibility with hardware from other vendors. Since they are all proprietary, there is usually no collaboration. These tools do not have a web interface and commonly have usability issues. As an illustration, one of these integrated management software understands each node of a cluster as an isolated computational resource. It then lists the nodes as items in the same level of a tree on a side panel. For example: there is a tree item called MyCluster and all the nodes are appended under this item as leaves. There is no problem if the cluster has few nodes, but on a large scale system, it is impractical to find a specific node in the middle of a huge list of names.

There are good proprietary solutions promising easy cluster deployment, management, usage and monitoring. Many provides a rich internet application or a user friendly GUI. However, on a HPC center with machines from different vendors, it may be difficult to use a single proprietary tool to manage all the available resources. No product offers a single point of management for an HPC environment, integrating the management of different resources and the aggregation of any other desired tool on a single web application.

Many of the discussed software are not prepared to handle the new challenges to manage large scale supercomputers. These challenges are related to the increasing number of processors and compute nodes on supercomputers, the heterogeneous environments that may use resources from different machines, the particular tools an environment needs or wants to use, and the different type of administrators, with varied level of knowledge, managing these clusters. HPC centers with huge supercomputers need a tool that presents all the available resources as a whole but presents it organized and customized by the user. It must be prepared to show thousand of nodes smartly, highlighting problems and warnings and providing an easy mapping to the hardware position on the equipment room. The software needs to be easy to use, to configure and to customize, being prepared to be integrated with other tools easily. It should make easier for a non expert manager to handle the main operations of a HPC center environment. Integration, flexibility and great usability are the most important features to deal with these new challenges.

The need to have a free tool that integrates many resources and tools providing an easy-to-use web interface for easy HPC environment administration made us idealize LEMMING, described below. Our main purpose is not to reinvent, but to use the open source collaboration model to integrate the HPC management and monitoring software stack with improved usability.

3 The LEMMING Project

The LEMMING project is inspired on the Zimbra Collaboration Suite[16]. Zimbra is a groupware [17] centered on e-mail and contact exchanging. It combines a set of open source tools, configuring them together. Zimbra offers an easy way to deploy and manage an e-mail server and web client with improved usability. Taking advantages of AJAX (Asynchronous JavaScript And XML) [18] features, it offers a rich web interface that behaves like a desktop application.

We decided to implement LEMMING instead of extending another tool because we need an application with great usability enhanced by a RIA. No cluster management and monitoring open source tool is developed as a RIA. Extending another tool would also make LEMMING dependent on it. For example: neither OSCAR nor Rocks installs Nagios as a monitoring tool; if LEMMING was a Nagios plug-in, it would compel the administrators to install Nagios to use LEMMING on a OSCAR or Rocks cluster system. But administrators may not want Nagios to monitor the cluster, they may prefer Ganglia. OSCAR and Rocks install Ganglia to monitor the cluster and LEMMING, actually, depends on Ganglia.

LEMMING is an acronym of Linux Enterprise Management and Monitoring but also makes reference to a species of small rodents that live near the Arctic region. The

lemming population increases drastically on a period of the year and the little animals start searching for a new habitat. They normally die on this process and the population is reduced to few specimens. Since the size of clusters is also increasing quickly, raising the probability of failure, we found the analogy interesting.

The hardware technology evolves fast and computers are more robust. Nevertheless, considering a hypothetical failure probability of one component in a thousand, a cluster with thousands of nodes will have many failures. If these defects are not noticed and corrected quickly, the cluster machine may be just like the lemming population – reduced to few nodes. This problem needs a solution, since the number of processing units per cluster is increasing, as shown on statistics from the TOP500 portal [2].

The main objectives of LEMMING are flexibility, easiness to add features and great usability to solve detected problems quickly. It was also designed as a web application to keep the remote access aspect of clusters for management tasks. To attend the usability requirement, we decided to develop a rich Internet application (RIA) using the AJAX model. Among many toolkits and frameworks, after some research, we decided to use the ZK [19] framework. ZK was chosen since it provides an easy development model based on interface components and lets you produce AJAX content very easily without programming Javascript. It means that both server and client side of the web application can be written in Java. Since the AJAX model, alone, does not enhance web page usability, it was also necessary to study and discuss a group of user interface models and components that fit the application necessities but that are also familiar to users.

LEMMING is designed to handle multiple supercomputers on a single interface. Its features include:

- Being freeware
- Web Service based
- AJAX interface design
- Integration of other tools
- Single point of management
- Tested with Rocks clusters
- Support for many cluster topologies organization
- Integrated with workload management
- Parallel shell tools
- Customizable Dashboard

A single point of management means that LEMMING concentrates the management of many components of many supercomputers and servers on a single interface. It also concentrates other tools that may be associated to a specific resource (the Ganglia page of a cluster, for example, or a local particular tool) or may represent the whole status of the HPC environment (the Nagios or Cacti web pages). LEMMING also reduces complexity of cluster management, providing a quick highlight of the status of available resources.

LEMMING is designed to be flexible, i.e., to handle multiple clusters and its components with different architectures and installed systems. It is not feasible to

build a huge tool that can support all the available cluster and server systems solution out of the box. Thus, we decided to split LEMMing in two modules. The first module is inevitably associated with a cluster system and is designed to implement a determined application programming interface (API) using the available tools in the cluster system. The second module is the web application that uses the API to manage the machines. Thus, to support a new system, only the first module may need to be adapted. The cluster system remains transparent to the second module that only has to deal with the known API.

We used web services technology to implement the idealized architecture. The module within the cluster exposes a set of web services that work as an API to the web module. Figure 2 shows the relationship between LEMMing modules. The administrator access the web application (LEMM-GATE) through a browser and the web module access the web services (LEMM-WS) to get cluster information and to execute operations on it. This architecture lets the LEMM-GATE server to be accessible through an external network while the clusters may remain accessible only through internal network. This approach prevents the cluster servers from external attack.

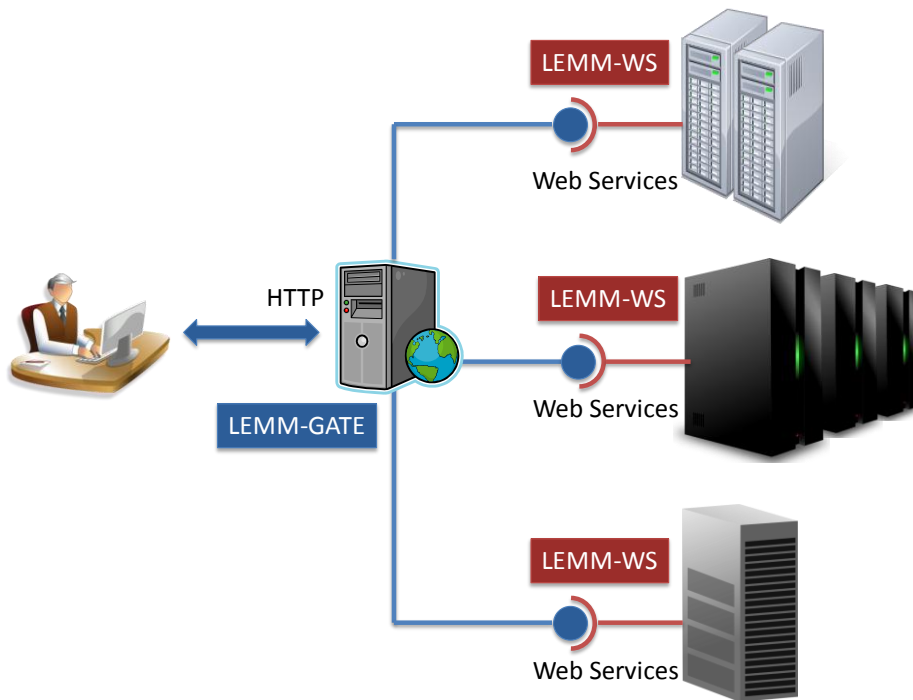


Figure 2 – LEMMing modules relationship

The idea of LEMMing is to take advantage of existent and available tools, before obtaining data manually. Thus, the web services deployed inside the cluster system explores the reports from Ganglia, from the queue system, like Torque [11] or SGE (Sun Grid Engine) [10] and available proprietary systems. Another web service takes advantages of macro commands and scripts available to execute operations on the

cluster. It may also need to connect to a database or a configuration file. Consequently, this service is very dependable of the installed cluster system, since these commands usually differ between architectures. To amortize this problem, there are some generic packages of classes inside LEMM-WS that supports shell command execution, database access, XML parsing, and file access. These packages were designed to reduce the effort while developing LEMM-WS for a new cluster system. Currently, LEMM-WS fully supports Rocks clusters, but we are already testing it on other cluster systems.

LEMM-WS can be easily deployed on a web server like Apache Tomcat [20]. To access management operations locally, the web services must be installed inside the cluster system, on the access server of the cluster, which is casually referenced as the frontend node.

The web application module is responsible to access the web services, show obtained data and work as the graphical interface to access the operations to handle the cluster system. It is a rich Internet application, i.e., it is a web page but behaves like a desktop application. LEMM-GATE user interface was designed to be familiar to the user. It is also prepared to organize the cluster according to its topology, which means that the compute nodes are organized on a hierarchy that divides the whole set into racks, chassis and blades. Resources reporting errors or warnings are highlighted visually to be examined by the administrator. The organization of the topology as a hierarchical tree is designed to enhance the navigation over the cluster structure. On machines with a huge number of nodes, it is especially necessary because the cluster is displayed in parts, making it easier to inspect. This organization is also very important on a cluster expansion, since it does not require renaming the old nodes. However, if the nodes are named sequentially, but the rack is not full, when new nodes are added on the empty spaces, there may be a name inconsistency.

LEMMing interface displays the overview of the domain with different supercomputers on the left side. They are arranged on a tree and the cluster items can be expanded to show their servers and nodes organized hierarchically. When an item of the tree is selected, LEMMing interface shows specific information about the selected item. The information displayed is defined by the user that associate tabs to the item. A tab may display a whole web page from another tool (for example, the Nagios page), or it may display some selected widgets. LEMMing offers some widgets by default, like the cluster or node load monitoring widget, the CPU usage share widget and the cluster queue viewer widget. It is also easy to create new widgets according to local needs. The user may customize a different set of tabs for each different type of item. For example, items representing a whole HPC environment may have tabs with Nagios and Cacti; items representing a cluster may have tabs with Ganglia and widgets monitoring cluster load and queue; and compute node items may also display monitoring widgets and the Ganglia page for that specific node. Other customizations are possible adding content to items representing racks, chassis and blades.

Since it is possible to integrate different tools inside LEMMing, the interface complexity may grow. However, since the interface is customizable and each administrator may have its own LEMMing account, an administrator can build its own workspace on LEMMing on a complexity level that is adequate to his skills.

The management operations on cluster components can be executed accessing the context menus of the item. The action calls the operator web service on LEMM-WS to execute the operation. For example, to reboot a node, the user right clicks it on the item and selects the 'Reboot' option on the context menu. There are other details about the interface that are not presented here. A video demonstration of LEMMING can be found at the project page [1]. Figure 3 shows two sample screens of LEMMING to give an idea of its interface.

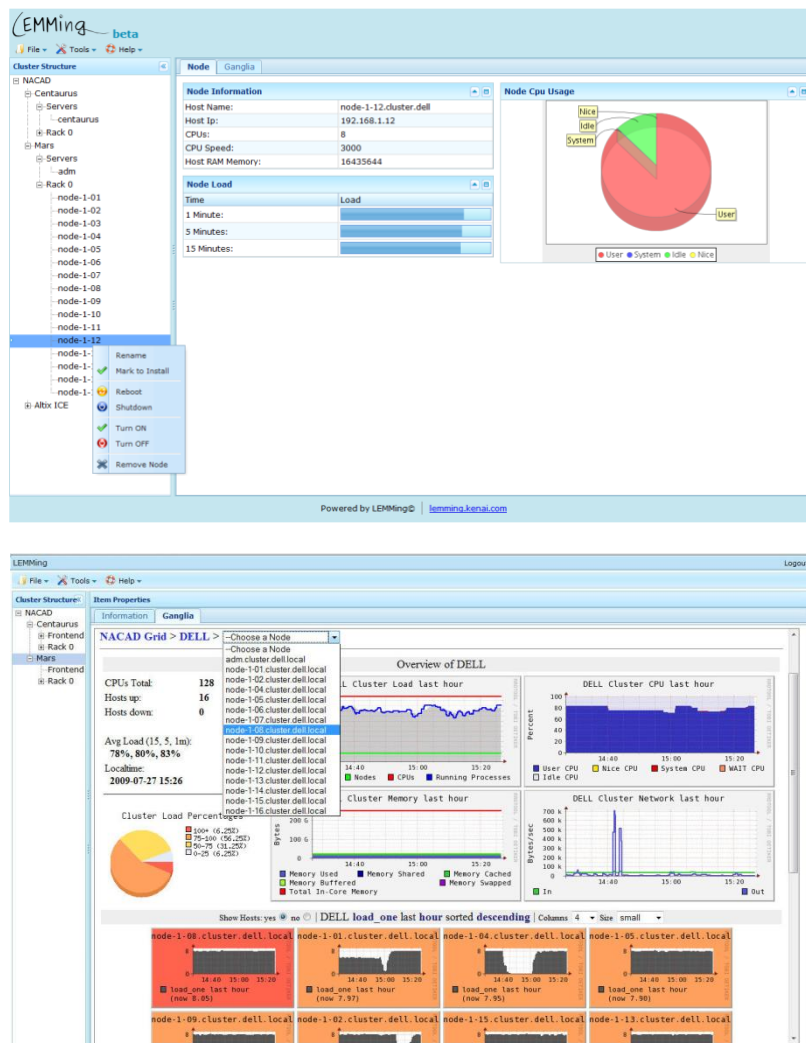


Figure 3 – LEMMING web interface

The web services module being developed detached from the AJAX interface allows the building of some specific deployment scenarios. As said, LEMM-GATE and LEMM-WS are independent modules and can be deployed on different servers.

However, they can also be put in the same server on the same web container. On a simple environment with only one cluster, for example, it is reasonable to deploy both LEMMing modules on the cluster frontend. However, on big HPC environments, the web application should be deployed on a web server and the web services inside each cluster. Therefore the single web application will access all the LEMM-WS modules deployed on every cluster and manage the clusters through them.

4 Conclusion

The new challenges to manage HPC environments with high scale supercomputers require new solutions to support multiple and heterogeneous resources on a single point of management with improved usability. We are proposing LEMMing, a tool to integrate the cluster management and monitoring software stack and operate it through a rich internet application.

We believe that a tool designed to attend a huge variety of systems and architectures should not be developed by a single group or company. Collaboration is essential to take LEMMing to other platforms. We expect that, being open source, it grows up with the community and support a great number of environments. LEMMing uses great available tools (Rocks, Torque/SGE, Ganglia) through a flexible architecture and a smart and easy-to-use web interface.

We are now working to extend LEMMing support to different cluster systems, while adding features like an integrated IPMI [21] support, users administration and queuing system management. We believe this solution will attend most HPC management scenarios, even with multiple supercomputers with thousands of processors. More information and a video demonstrating LEMMing can be found on the LEMMing project web page at <http://lemm.sf.net/> [1].

Acknowledgments

The authors thank the High Performance Computing Center (NACAD-COPPE/UFRJ) and Professor Alvaro Coutinho for supporting LEMMing Project. They also thank Dell Brazil for partially supporting the project.

References

- [1] J. Dias e A. Aveleda, *LEMMing Project*, <http://lemm.sf.net/>, 2010.
- [2] TOP500, *TOP500 Supercomputing Sites*, <http://www.top500.org/>, 2010.
- [3] K.J. Barker, K. Davis, A. Hoisie, D.J. Kerbyson, M. Lang, S. Pakin, e J.C. Sancho, "Entering the petaflop era: the architecture and performance of Roadrunner," *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, Austin, Texas: IEEE Press, 2008, pp. 1-11.
- [4] A. Petitet, R. Whaley, J. Dongarra, e A. Cleary, *HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers*, 2010.
- [5] B. des Ligneris, S. Scott, T. Naughton, e N. Gorsuch, "Open Source Cluster Application Resources (OSCAR): design, implementation and interest for the

- [computer] scientific community,” *Proceedings of the 17th Annual International Symposium on High Performance Computing Systems and Applications and the OSCAR Symposium: May 11-14, 2003, Sherbrooke, Quebec, Canada, 2003*, p. 241.
- [6] P.M. Papadopoulos, M.J. Katz, e G. Bruno, “NPACI Rocks: Tools and Techniques for Easily Deploying Manageable Linux Clusters,” *Cluster Computing, IEEE International Conference on*, Los Alamitos, CA, USA: IEEE Computer Society, 2001, p. 258.
- [7] xCAT, *Extreme Cloud Administration Toolkit*, <http://xcat.sourceforge.net/>, 2010.
- [8] E. Gabriel, G.E. Fagg, G. Bosilca, T. Angskun, J.J. Dongarra, J.M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R.H. Castain, D.J. Daniel, R.L. Graham, e T.S. Woodall, “Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation,” *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, 2004, pp. 353-377.
- [9] W. Gropp, “MPICH2: A New Start for MPI Implementations,” *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, 2002, pp. 37-42.
- [10] W. Gentsch, “Sun Grid Engine: Towards Creating a Compute Power Grid,” *IN CCGRID '01: PROCEEDINGS OF THE 1ST INTERNATIONAL SYMPOSIUM ON CLUSTER COMPUTING AND THE GRID*, 2001, p. 35.
- [11] G. Staples, “TORQUE resource manager,” *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, Tampa, Florida: ACM, 2006, p. 8.
- [12] M.L. Massie, B.N. Chun, e D.E. Culler, “The ganglia distributed monitoring system: design, implementation, and experience,” *Parallel Computing*, vol. 30, Jul. 2004, pp. 817-840.
- [13] W. Barth, *Nagios: System and Network Monitoring*, No Starch Press, 2008.
- [14] P. Shivakumar, N.P. Jouppi, e P. Shivakumar, “CACTI 3.0: An Integrated Cache Timing, Power, and Area Model,” 2001.
- [15] T. Oetiker, *RRDtool - Round-Robin Database Tool*, <http://oss.oetiker.ch/rrdtool/>, 2010.
- [16] M. Gagné, “Zimbra collaboration suite, Version 4.5,” *Linux J.*, vol. 2007, 2007, p. 14.
- [17] C.A. Ellis, S.J. Gibbs, e G. Rein, “Groupware: some issues and experiences,” *Commun. ACM*, vol. 34, 1991, pp. 39-58.
- [18] L.D. Paulson, “Building Rich Web Applications with Ajax,” *Computer*, vol. 38, 2005, pp. 14-17.
- [19] H. Chen e R. Cheng, *ZK Step-By-Step: Ajax without JavaScript Framework*, Apress, 2007.
- [20] Tomcat, *Apache Tomcat*, <http://tomcat.apache.org/>, 2010.
- [21] Intel Corporation, Dell, Hewlett-Packard, e NEC Corporation, *IPMI - Intelligent Platform Management Interface*, White Paper: <http://www.intel.com/design/servers/ipmi/>, 2010.