An Architecture for Developing Cyber Environments using Multiple HPC Infrastructures for e-Science Applications

Felipe Maciel, Carina T. Oliveira[3], Renato Juaçaba Neto
João Marcelo Alencar, Paulo Rego, Ronaldo Bezerra
Danielo G. Gomes, Rossana M. C. Andrade[*], José Neuman de Souza[†]

[1]Group of Computer Networks, Software Engineering, and Systems - GREat
Federal University of Ceará - UFC

[2]National Center of High Performance Computing - CENAPAD
Federal University of Ceará - UFC

[3]Computer Networks and Distributed Systems Laboratory - LAR
Federal Institute of Education, Science and Technology of Ceará - IFCE

{felipemaciel,renatojuacaba,ronaldolima}@great.ufc.br

{carina.oliveira}@ifce.edu.br

{joao.marcelo,pauloalr,rossana,danielo,neuman}@ufc.br

Abstract. *In this paper, we propose a novel architecture to allow the implementation of a cyber environment composed of different High Performance Computing (HPC) infrastructures (i.e., clusters, grids and clouds). To access this cyber environment, scientific researchers do not have to become computer experts. In particular, we assume that scientific researchers provide a description of the problem as an input to the cyber environment and then get their results without being responsible for managing the computational resources. We provide a prototype of the architecture and introduce an evaluation which studies a real workload of scientific applications executions. The results show the advantages of the proposed architecture. Besides, we highlight this work provides guidelines for developing cyber environments focused on e-Science.*

1. Introduction

High Performance Computing (HPC) can be defined as a computing environment which deals with complex computational requirements, and with a wide range of data processing constraints and/or inelastic applications. In the last few years, we have seen an interesting growth in e-Science applications, which rely heavily on HPC infrastructures[1].

Clusters, grids and public/private clouds are suitable for e-Science but each of these HPC infrastructures has its own strengths and weaknesses in terms of capacity, capability, resource heterogeneity, security, interoperability [Mateescua et al. 2011].

In this context, research scientists need an interface to access the available HPC infrastructures to submit and manage jobs of scientific applications. The HPC access

[*]CNPq Research Productivity in Technological Development and Innovative Extension Scholarship
[†]CNPq Research Productivity Scholarship
[1]http://www.top500.org/statistics/list

solutions range from the most basic and flexible Command-Line Interface (CLI) to high-level solutions based on scientific Web portals. Regarding the CLI solution, to access an Unix cluster infrastructure, for example, research scientists have to know/learn Unix commands, as well as how to interact with queue management systems via command line. On the other hand, scientific portals simplify the user experience through the interface abstraction with the computing infrastructure. In this way, it is possible to reduce the research scientists efforts in relation to the infrastructure interaction and leave them focusing preferentially on the object of research [Bastos et al. 2013]. However, despite scientific portals facilitate the jobs submission, it decreases users flexibility once portals usually depend on the application/software.

In this paper, we propose an integration of heterogeneous HPC infrastructures in order to allow better problem-solving in e-Science. More specifically, we present an architecture to implement a cyber environment for e-Science applications. The main goal of this cyber environment is to keep the strengths of each HPC technology as well as to reduce their weaknesses.

The contribution of this work is twofold. First, we propose an architecture that details four main modules and how they communicate to design the proposed cyber environment. Second, we provide a proof of concept: we have developed a prototype of the proposed architecture through the implementation of a scientific portal integrated with two different HPC infrastructures (cluster and cloud).

2. The Proposed Architecture

In this section, we introduce the four modules of the proposed architecture.

2.1. Scientific Portals Module

The Scientific Portals Module is responsible for the direct interaction of scientific researchers with the cyber environment. It aims at providing scientific researchers a friendly Web interfaces that facilitates the access to e-Science applications. In this way, scientific researchers can concentrate their efforts on the subject of research, and not on the technical peculiarities of the other modules. Therefore, e-Science applications provided by the Scientific Portals Module can be seen as Software-as-a-Service (SaaS) [Zhang et al. 2010].

The Scientific Portals Module provides secure access to e-Science applications and their respective outputs. These functionalities are achieved through the use of friendly Web pages that allow scientific researchers: (i) to authenticate through the interaction of the Scientific Portals Module with the Authentication Module; (ii) to interact with HTML forms, in which parameters of the e-Science application would be collected and; (iii) to manage the job from a simple HTML interface: job status, download outputs, cancel jobs, etc.

When a scientific researcher submits a job through one of the applications forms (i.e., scientific portal), a description of this job is created and passed to the Decision Module that, in turn, will decide how to handle with the new job request. Scientific researchers can access the scientific portal at any time to manage jobs. It is possible due to the communication of the Scientific Portals Module with the Environment Module.

Tabela 1. Resource Comparison

Type	Computer Power	Network	Scalability	Security
Cluster	TeraFlops	Low latency, high bandwidth	Low	Restricted access
Grid	TeraFlops/PetaFlops	High bandwidth	Medium	Controlled access
Cloud	Unlimited (in theory)	Regular bandwidth	High	Public access

2.2. Authentication Module

The Authentication Module is responsible for implementing the scientific researchers' authentication to the scientific portals, as well as guaranteeing that scientific researchers can access and then use the resources of all computing environments. For this reason, the Authentication Module presents a connection with the Scientific Portals Module and Environment Module.

Since it is desirable a cyber environment to support heterogeneous computing environments such as cluster, grids and public/private clouds, it becomes necessary a centralized solution to perform authentication and also avoid multiple databases of users/credentials.

Clusters generally run a modified version of existing operating systems, mostly UNIX variants. These systems also offer ways of authenticating users through directory services such as LDAP with little configuration effort. Since the creation of an user is not an action that occurs many times during a day, the overhead is small, and our architecture can support both modern and legacy computing resources.

2.3. Decision Module

The proposed architecture aims to integrate heterogeneous computing environments as diverse as clouds, grids and clusters. Since we are considering HPC applications, ensuring good performance of these applications is one of our main driving goals. After an user submission, our cyber infrastructure must decide where the application code will actually run. The Decision Module is responsible for this selection, based on user requirements and component infrastructures characteristics.

The resources in a cyber environment can be clusters, grid and clouds. Clusters are the systems with more predictable performance of the 3 resource types. They are a concise set of powerful machines, with a fast low latency interconnect network and plenty of storage attached. Grids are more dynamic and have geographically dispersed components, but still with a broadband network and strict access rules. Clouds are a totally different option, with virtual machines and unpredictable network behavior, but with elasticity guaranteed to enable a huge number of virtual machines. Table 1 summarizes these resources characteristics.

Unfortunately, even a simple submission to a restricted set of resources may lead to a multi-variable optimization problem that is not easy to solve. Even in the case of a single machine with multiple processors, the allocation problem has been shown to be NP [Fernandez-Baca 1989]. Then, in this work, we propose an architecture for our Decision Module in which the administrator of the portal is able to define the scheduling scheme according to whatever is better for the current domain. The basic module simply loads a mapping with application/resource pairs. This way, the Decision Module knows to which

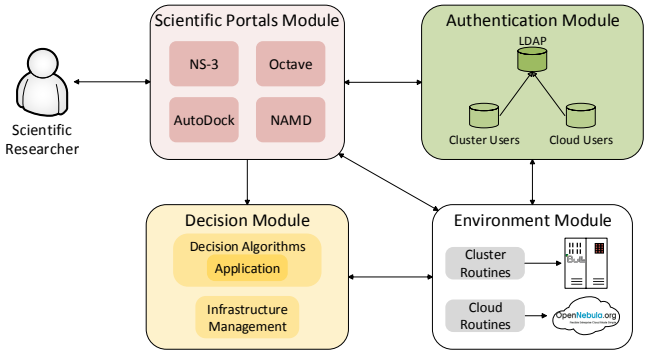

Figura 1. Prototype of the proposed architecture.

resource it should send the job created by the job description. An extended module may provide an heuristic to aid the decision process or apply statistical analysis. These two approaches are just examples of what is used in existing solutions for scheduling in grids [De Alencar et al. 2012].

2.4. Environment Module

The Environment Module is composed of differents HPC infrastructures (e.g., cluster, grid and cloud), which will execute the jobs of scientific applications according to the choice of the Decision Module, and their routines. Such routines can be seen as plugins that translate the requests and interactions for each HPC infrastructure, implementing executions according to the particularities of each one.

If the Decision Module determines that a job will be executed in a cluster, for example, the routines implemented will interact with job schedulers that generally manage the computing resources in this kind of infrastructure. The submission of a job is made through a *bash* script that informs the job scheduler the parameters of the execution, including the instruction to the application requested by the scientific researcher and the input file, when required. After submission, the execution is unattended and controlled by the job scheduler in background. Such proceeding is similar when the job is executed in grids, since they use job schedulers as well.

Environment Module shows a computing infrastructures heterogeneity, which emphasizes a key feature of cloud computing: the elasticity [Zhang et al. 2010]. Elasticity provides seemingly infinite computing resources available on-demand, and it adapts quickly enough to follow load surges, thereby eliminating the need for cloud computing users to plan far ahead for provisioning [Armbrust et al. 2010]. Decision Module only considers active computing resources on its evaluation process. Thus, it constantly monitors the Environment Module to relocate job submission for active infrastructures.

3. Prototype and Proposal Evaluation

Based on the architecture detailed in Section 2, we have implemented a prototype integrated with two different HPC infrastructures: Bullx Cluster and OpenNebula Cloud Platform[2]. Figure 1 illustrates the implemented prototype. The cluster and cloud compu-

[2]http://opennebula.org/

ting infrastructures illustrated in the Environment Module of Figure 1 are in operation at the CENAPAD-UFC[3].

Scientific Portals Module was based in four applications: the Network Simulator 3 is a widely-used discrete-event network simulator for Internet systems, the GNU Octave is a high-level language, primarily intended for numerical computations, the AutoDock is a suite of automated docking tools and the NAMD is a parallel molecular dynamics code designed for high-performance simulation of large bio-molecular systems.

In our prototype[4], the decision algorithm of the Decision Module takes into account the amount of resources used by the task to decide the ideal infrastructure for execution. According to this criteria, a task can be classified into serial or parallel.

In order to assess the viability of the architecture detailed in Section 2, we conducted an analysis of how a workload from a traditional HPC infrastructure performs on an heterogeneous environment composed of a public cloud and a cluster.

The workload used consists of a real trace of the jobs executed on the cluster through the months of April/2014 to November/2014. For each job, we collected the submit time, the queue interval, and the amount of resources (cores and computing nodes) utilized. We noticed that a considerable number of jobs are serial code that uses only one CPU core at a time, no matter how many cores are available. This happens because users are performing parameter sweeping computations or simply because they have not updated their applications to make use of parallelism. The main benefit for them for using the cluster is the guaranteed uptime. Besides serial jobs, there are parallel jobs that employ several computing nodes at the same time and benefit the most from the fast interconnect. Table 2 describes the workload regarding the number of serial and parallel jobs.

Tabela 2. Workload Description.

Job Type	Number	Total Running Time
Serial	19357	1631 days 16h 59 min 2s
Parallel	970	614 days 2h 10min 35s

One common situation in the workload was the presence of parallels jobs with high queue waiting intervals because the cluster had a lot of serial jobs executing. The parallel jobs benefit from the cluster architecture, but the serial jobs do not. There are no penalties from running serial jobs at a public cloud [Expósito et al. 2013]. In our architecture, the infrastructure maintainer may define scheduling policies in the Decision Module (see section 2.3). A possible policy for decreasing the queue interval for parallel jobs is to submit all serial applications to a public cloud. This would not affect the performance of the serial jobs and overall user satisfaction would increase.

To evaluate the proposed policy, the first step for the analysis was the submission of the obtained workload to a simulated environment consisted of a cluster with the same resources as the cluster at CENAPAD-UFC and a public cloud without limit for the number of virtual machines. We used the GridSim[5]. The impact of the policy enforcement on

[3]http://www.cenapad.ufc.br/

[4]https://a2cc.cenapad.ufc.br/

[5]http://www.buyya.com/gridsim/

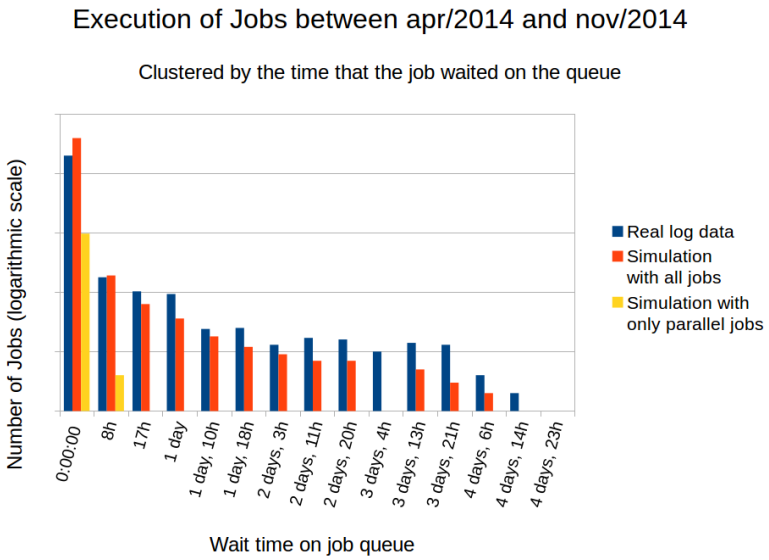

Figura 2. Queue intervals with and without serial jobs.

the queue intervals is depicted in Figure 2. In the y-axis we can see the number of jobs with average queue interval in the x-axis. For example, the majority of jobs wait less than 8 hours in the queue and none of them waited more than 5 days.

The real log data and the simulation log with all jobs are very similar. This shows that the simulator is fine tuned with reality. We can see the policy effect when the serial jobs were removed from the workload. The simulation with only parallel jobs shows that no jobs wait more than 8 hours in the queue. This leads to the conclusion that removing the serial jobs will decrease queue interval for parallel jobs.

But what to do with the serial jobs? Sending them to a public cloud is an option. A public cloud generally offers almost limitless resources, or at least a limit that is way beyond regular user needs. There is an overhead for creating virtual machines, but it takes seconds or minutes, while the queue interval in the cluster may reach days. For the researcher there would be no decrease in performance and the queue interval would also be smaller for the serial jobs. But public cloud resources are not free, and we must consider the cost of sending all the serial jobs to the cloud.

We estimated the price using instances with the smallest number of cores. Using the cheapest instances of two different providers (Amazon EC2[6] and Microsoft Azure[7]), the total cost sums close to 5.000 dollars. With intermmediate instance types (better processors), the cost increases considerably. A detailed view for the cost/instance ratio is presented in Figure 3.

This analysis shows that maintainers of such cyber environments that have access to a public cloud may steer the scheduling according to a monetary budget and the queue interval requirements.

[6]http://aws.amazon.com/ec2/pricing/
[7]http://azure.microsoft.com/en-us/pricing/details/virtual-machines/

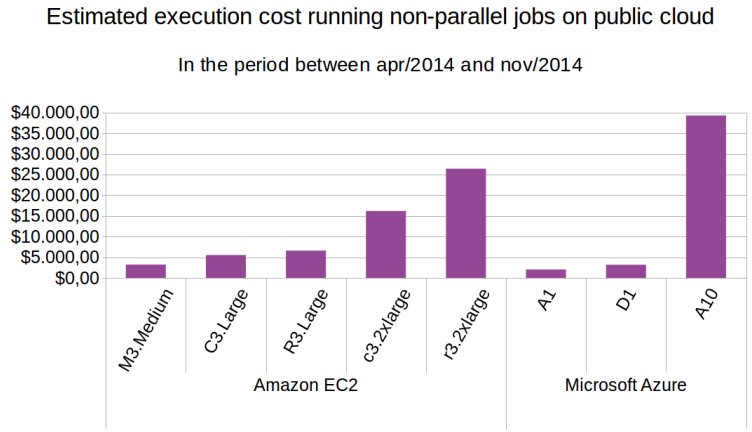

Figura 3. Total cost for serial jobs according to instance types.

4. Related Work

The a^2c system is a solution developed as a trade-off between the flexibility of the CLI and the easiness provided by scientific portals [Maciel et al. 2013]. In this system, users interact with a Web form through the communication protocol HTTPS, which makes the connection encrypted. In the a^2c server, features are implemented to eliminate the need of CLI and execute Linux/SLURM commands in the cluster. In turn, the Web server interacts with the cluster via the cryptographic network protocol SSH. When performing login in the a^2c system, the user information and password are sent (encrypted by SSH) to the cluster. After that, the authentication is performed and the user can access the computational resources provided by the CENAPAD-UFC. As login authentication is performed in the cluster itself, we eliminate the register of information in the a^2c server.

Falfushinsky *et al.* [Falfushinsky et al. 2013] describe the integration of a private cloud within grid sites for accelerating the application deployment and supporting multiple virtual organizations by grid sites. This cloud in grid approach has been implemented and tested in Ukrainian National Grid, a part of European Grid Infrastructure. The authors developed their own cloud management system using the Oracle VirtualBox hypervisor and the users authentication is performed under the standard grid rules using proxy certificates.

Zissis and Lekkas [Zissis and Lekkas 2012] also propose the use of LDAP operating in concert with Public Key Infrastructure to address several identified threats in cloud computing environments, and to ensure the authentication, integrity and confidentiality of involved data and communications.

5. Conclusion

In this paper, we propose an architecture that allows the implementation of a cyber environment composed of different computing infrastructures such as clusters, clouds and grids. We have presented a prototype of the proposed architecture that validates it. Also, we introduce preliminary results that compare the performance of a scientific portal in two different HPC infrastructures (cluster and cloud). The discussions and results provide guidelines for scientific applications developers and practitioners in planning efficient cyber environments.

The contributions presented in this paper bring up interesting perspectives for future research. For instance, we plan to evaluate the cyber environment when different policies are adopted for the Decision Module. Also, we plan to implement new cloud platforms and evaluate their overall performance.

6. Acknowledgements

This work is a product of the Group of Computer Networks, Software Engineering, and Systems (GREat), Associated to the National Institute of Science and Technology – Medicine Assisted by Scientific Computing (INCT-MACC). The NS-3 e-Science portal is a partial result of the project MCT/CNPq 14/2013-Universal 486287/2013-0.

Referências

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2010). A view of cloud computing. *Commun. ACM*, 53(4):50–58.

Bastos, B. F., Moreira, V. M., and Gomes, A. T. A. (2013). Rapid Prototyping of Science Gateways in the Brazilian National HPC Network. In *International Workshop on Science Gateways (IWSG)*.

De Alencar, J. M. U., Andrade, R. M., Viana, W., and Schulze, B. (2012). P2PScheMe: a P2P scheduling mechanism for workflows in grid computing. *Concurrency and Computation: Practice and Experience*, 24(13):1478–1496.

Expósito, R. R., Taboada, G. L., Ramos, S., Touriño, J., and Doallo, R. (2013). Performance analysis of HPC applications in the cloud. *Future Generation Computer Systems*, 29(1):218 – 229. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures.

Falfushinsky, V., Skarlat, O., and Tulchinsky, V. (2013). Cloud Computing Platform within Grid Infrastructure. In *IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*.

Fernandez-Baca, D. (1989). Allocating modules to processors in a distributed system. *IEEE Transactions on on Software Engineering*, 15(11):1427–1436.

Maciel, F. A. O., Cavalcante, T. M., Neto, J. Q., Alencar, J. M. U., Oliveira, C. T., and Andrade, R. M. C. (2013). Uma Arquitetura para Submissão e Gerenciamento de jobs em Infraestruturas Computacionais de Alto Desempenho. In *Workshop de Computação em Clouds e Aplicações (WCGA)/Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*. SBC.

Mateescua, G., Gentzsch, W., and Ribbens, C. J. (2011). Hybrid Computing—Where HPC meets grid and Cloud Computing. *Future Generation Computer Systems*, 27(5):440–453.

Zhang, Q., Cheng, L., and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18.

Zissis, D. and Lekkas, D. (2012). Addressing cloud computing security issues. *Future Generation Computer Systems*, 28(3):583 – 592.