The CICA GRID

A Cloud Computing Infrastructure on Demand with Open Source Technologies

M. A. Alvarez[1], A. Fernandez-Montes[1], J. A. Ortega[1] and L. Gonzalez-Abril[2]
[1]*Computer Languages and Systems Department, University of Seville, Seville, Spain*
[2]*Applied Economics I Department, University of Seville, Seville, Spain*

Keywords: Cloud Computing, Cluster, Cobbler, IaaS, OpenNebula, Profile, Puppet, ReCarta, Virtual Machine.

Abstract: A new approach technology to enable the expansion and replication of resources on demand is presented in this paper. This approach is called CICA GRID and it provides service to research community in the Scientific Computer Centre of Andalusia (CICA). This approach is an alternative solution to the initial cost involved in building an own data center by public organizations for researches. This solution quickly provides resources with a minimal technical staff effort. Also, an architecture and user interface example called ReCarta was presented. This system supplies a private Cloud Computing system for non-technical end-users.

1 INTRODUCTION

In the last years, Cloud Computing has been launched as a concept that it has potential to transform the way in which computers are used and managed. This technology promises to realize the objective of transforming the computing resources into a single process. This process can use any quantity of resources during the needed time.

These features are especially interesting for the HPC/Grid Computing/Scientific area since they enable resources to be managed in a controlled environment. In this sense, researchers estimate their computing needs when a new project is considered. Therefore, researchers must spend their time to configure the available resources to satisfy their needs. Furthermore, the problem is bigger if researchers do not have technical computer knowledges.

In response to the needs of researchers and to improve the Andalusian Supercomputing Network (RASCI)[1], the Scientific Computer Center of Andalusia (CICA)[2] has implemented a technology solution called CICA GRID. It enables the expansion or replication of resources depending on research demands. CICA, in collaboration with the Spanish National Grid Initiative[3], features a high scalability cloud with a quick resources configuration: a GRID environment solution.

The developed approach incorporates three tools to carry out its function: Cobbler[4], Puppet[5] and OpenNebula[6]. It was called ReCarta[7] and it hides these tools to user by a web interface.

The paper is organized as follows: Cloud Computing technology is briefly presented in the next section. In Section 3, the project motivation and system architecture are analyzed and users' tools and examples are given in Section 4. Benchmarks and features are shown in Section 5. Section 6 provides a final discussion and concludes this paper.

2 CLOUD COMPUTING

Cloud Computing refers to hardware and software infrastructure which allows applications to be served across the web for end-users. Furthermore, it provides computational resources and virtual hosting to build their own applications for them and the hardware and software datacenter is called the Cloud.

There are two kinds of Cloud: Public Cloud (Armbrust et al., 2009) and Private Cloud. The first one is available for commercial purposes and pay-per-use (Stuer et al., 2007). The second one is found in an in-

[1]http://rasci.cica.es
[2]http://www.cica.es
[3]http://www.es-ngi.es
[4]http://cobbler.github.com
[5]http://reductivelabs.com/trac/puppet
[6]http://www.opennebula.org/doku.php
[7]http://trac.cica.es/recarta/

A. Alvarez M., Fernandez-Montes A., A. Ortega J. and Gonzalez-Abril L..
The CICA GRID - A Cloud Computing Infrastructure on Demand with Open Source Technologies.
DOI: 10.5220/0003992603010304
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 301-304
ISBN: 978-989-8565-11-2

dividual organization and the access is only allowed to authorized members. Also, Cloud Computing systems can be classified as IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service).

3 PROJECT MOTIVATION AND SYSTEM ARCHITECTURE

A cluster for HPC is supported by CICA and the applied model is basically IaaS inside a private cloud which is accessed only by users from RASCI. It uses SGE as the Local Resource Manager (LRM) with Sun Grid Engine. Furthermore, this cluster has about 30 machines ant it is part of the Spanish National Grid Initiative.

The main motivation is to increase the requirements for access to excess computational resources of a working scheme in a cluster with an LRM. Thus, a project where authenticated and authorized users could design their own computational infrastructure and then use and manage it across a comfortable and simple interface was initiated. It was called “Recursos a la carta” (À-la-carte resources).

To achieve these goals some technical issues must be solved: i) machine supply; and ii) how to distribute the available physical resources among these virtual machines which they need them. The presented proposal, called CICA GRID, is developed as follows.

3.1 Provisioning and Management of Large-scale Virtual Systems

The CICA GRID is a private cloud with 35 virtual machines. It is composed by gLite’s working nodes and services (Andreetto et al., 2008). It is essential to have a tool that enables easy and flexible administration of these machines. The management of this cloud will be easier and more automated with it. Also, it must support the production control of the features and services of each machine. Hence, the problem of building the machines demanded by users has been resolved using Cobbler/Koan.

This tool facilitates the provisioning of virtual machines according to options given by users when they select how they want to build their machines and establishes an object hierarchy which defines the configuration characteristics at the highest levels. From the highest to the lowest level, they are Distro, Profile, Subprofile and System.

The relationship between objects that can be defined with Cobbler and the actual supplied machines are shown in Figure 1.

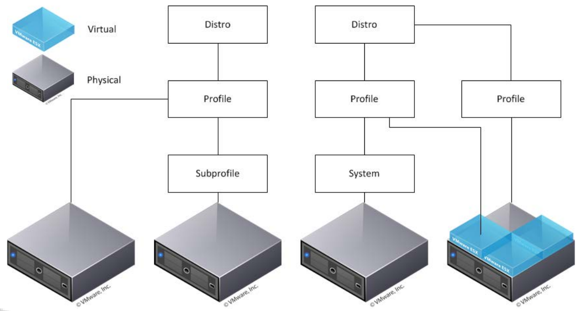

Figure 1: Cobbler object hierarchy.

At the time of computer installation, PXE boots the system while Cobbler shows a drop-down menu where the installation type can be chosen. If a virtual machine is going to be supplied, then it is possible to use the Koan command over the physical machine to specify what kind of machine is needed.

In the CICA GRID, users initiate a guided installation through a Cobbler profile. In answer to this request, the designated virtual machines are kept in a shared space (machine repository [see Figure 2]) where they are left available to OpenNebula for deployment.

Since provision and deployment of virtual machines does not resolve all infrastructure maintenance problems, a system for automating administration tasks is required and Puppet has been chosen. It provides a framework to simplify the work of system administrators, reusing the code as much as possible and allowing a modular system. Also, it is based on a client-server scheme and a declarative language that specifies administration tasks.

Puppet is used in the CICA GRID to configure and ensure that the NTP service of machines works correctly. Also, it must ensure that users are authenticated by LDAP and a basic backup configuration, security updates and certain file systems are set up. Through Cobbler profiles, each newly supplied virtual machine has a Puppet installed client.

Both Cobbler/Koan and Puppet have been proved to be capable of providing support for hundreds of machines.

3.2 Virtualized Systems

Open Nebula has been chosen to solve the problem of finding a system for an efficient deployment on virtual machine. It is an open-source virtual infrastructure engine and it enables dynamic deployment and re-placement of virtual machines using a pool of physical resources. It has achieved to decouple the server not only from the physical infrastructure, but also from the physical location.

Therefore, once Cobbler has provided machines requested by users and they have been saved in repository, the system will build needed files to enable OpenNebula to launch the deployment of machines as shown in Figure 2.

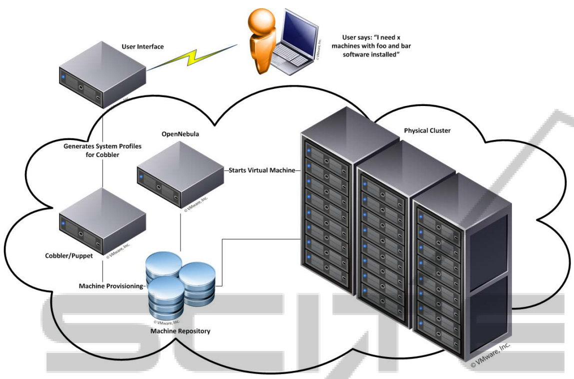

Figure 2: Architecture: connection scheme.

4 USER TOOL

The CICA GRID has a modular design. It facilitates its development and has been implemented in Python (Lutz and Van, 2001) language. Each module presents a well-defined interface so it can be easily used by other parts. They are Cobbler, DHCP and DNS management, Puppet, Open Nebula and User Interface modules.

4.1 ReCarta

A minimalist approach which attempts to show users the possible options is required. Therefore, the main focus is not on writing less code, but providing users with a useful system. This system is called ReCarta.

The created machine is composed for 2 steps. At first, users must define hardware and software features when they create a new group of machines. Later, users must indicate how many machines and the names of each have to be defined with these features.

At the end of this process, created system user data is shown along with the information needed for connection and start-up. Therefore, users have a project control panel at their disposal and they can see systems that they have been defined.

4.2 Code Example

A code example is given in order to illustrate the set of calls to defined API by different modules. They carry out tasks that a user has requested via web interface.

A new Cobbler profile (a new project in the user terminology) is created. It defines machines with 1 CPU, 512 MB of RAM, 4 GB of hard drive and Java language support.

```
import mod_cobbler
import mod_dhcpDns
import mod_puppet
import mod_nebula

miCobbler = mod_cobbler.Cobbler( 'john' )
miCobbler.setProfile( {
    'nombrePerfil':'project',
    'kickstart':'vm-kickstart.template',
    'diskSize':'4',
    'ram':'512',
    'cpus':'1',
    'comment':'project profile',
    'software':[ 'X-WINDOW', 'JAVA-SUPPORT' ]
} )
macs = miCobbler.setSystems( [
    { 'nombre':'project-vm',
        'comentario':'project VM',
        'perfil':'project' }
] )
mapIpNames = mod_dhcpDns.addSystemsDHCP(macs)
mod_dhcpDns.addEntryDns( mapIpNames )
miCobbler.provisionSystems( [ 'project-vm' ] )
```

It is important to note two ReCarta design features[8]. One is the high abstraction level offered by different methods. The kickstart template is modified to adapt it to different user requirements and it do not appear nowhere. Also, DHCP/DNS server configurations are modified to assign a place to new systems in network.

The other feature of ReCarta is the absence of system database to save information about defined projects and users, etc. ReCarta put the usernames as a prefix to data profiles and the project name to defined system names by users. This design decision has been taken to keep ReCarta as simple as possible.

5 BENCHMARKS AND FEATURES

The CICA GRID is a private cloud with 35 virtual machines with the following virtualized features: 1 core and 1 GB RAM. 6 physical servers are used to virtualize them with the following features: 2 cores and 4 GB RAM.

[8]http://trac.cica.es/recarta/wiki/RecartaDevel

Table 1: HPCC benchmarks.

	Intel 6400 Physical	Intel 6400 Virtualized
PTRANS(GB/s)	0.65	0.54
HPL(Gflops)	14.26	13.01
MPI Latency(ms)	0.00043	0.00053
MPI Bandwidth (ms)	1471.17	1477.64

Table 2: Bonnie++ write test - 2Gb blocks.

Server type	Sequential output (Kbs)	Sequential input (Kbs)	Random (seek/s)
Virtual	14014	34851	150.7
Physical	45678	49719	64.7

Table 3: Bonnie++ create test - 1Gb blocks.

Server type	Sequential Create (s)	Random Create (s)
Virtual	0.0000	0.0000
Physical	1621	891

Table 4: Consumption of 6 physical and 35 virtual servers.

Servers	Consumption (KWh/year)	Total (KWh/year)
35 virtual	222.6	7791
6 physical	516	3096

Nowadays, ReCarta creates systems compound of Xen (Barham et al., 2003) virtual machines. It has been used because it is proved that a paravirtualized virtual machine only loses 5-10% of CPU performances respect to equivalent physical machine.

Table 1 presents HPCC benchmark execution results. As expected, performances of the physical machine are better than virtualized machine. However, it is observed that the performance is about the same in both cases, so we can conclude that the proposal can be accepted as valid.

Table 2 and Table 3 show Bonnie++ execution results for virtual machine memory and equivalent physical machine. Also, a significant decrease in performance between virtual and physical machine can be seen for writing action to disc in these latter cases. In this case, the differences are slightly higher because the benchmark is performed on disk access. In this process, a virtual machine generates a very intense traffic on its virtual hard disk, especially reading.

The power consumption can see in Table 4. The use of virtualization allows the power consumption to be reduced to 39% can be seen in it.

6 CONCLUSIONS

Although the CICA GRID is still in its experimental phase, some case studies have been carried out. One of them is the creation of a small virtual cluster to be used with Apache Hadoop (Borthakur, 2007). Also, the project objectives have enabled that more job requests could be served without exceed the normal workload for a HPC cluster.

From the point of view of the energy-saving involved in virtualized environments, the CICA GRID renders research advances for the research groups with less cost by releasing electrical consumption.

We have learned during the launching of our pilot project of Cloud Computing that our users appreciate two advantages: i) the illusion of having a huge computing resource reserved exclusively for them; and ii) the possibility of increasing and decreasing the resources according to their needs.

ACKNOWLEDGEMENTS

This research is partially supported by the projects of the Spanish Ministry of Economy and Competitiveness ARTEMISA (TIN2009-14378-C02-01) and Simon (TIC-8052) of the Andalusian Regional Ministry of Economy, Innovation and Science.

REFERENCES

Andreetto, P., Andreozzi, S., Avellino, G., Beco, S., Cavallini, A., Cecchi, M., Ciaschini, V., Dorise, A., Giacomini, F., Gianelle, A., et al. (2008). The glite workload management system. In *Journal of Physics: Conference Series*. IOP Publishing.

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al. (2009). Above the clouds: A berkeley view of cloud computing. Technical report, Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley.

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. (2003). Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, 37(5):164–177.

Borthakur, D. (2007). The hadoop distributed file system: Architecture and design. *Hadoop Project Website*.

Lutz, M. and Van, G. (2001). *Programming Python: Object-Oriented Scripting*. O'Reilly & Associates, Inc., London, 2nd edition.

Stuer, G., Vanmechelen, K., and Broeckhove, J. (2007). A commodity market algorithm for pricing substitutable grid resources. *Future Generation Computer Systems*, 23(5):688–701.