Dynamic Reconfigurable Component for Cloud Computing Resources

Omar SEFRAOUI
ENSAO UMP
BP 669 Al Qods,
OUJDA, MOROCCO.

Mohammed AISSAOUI
ENSAO UMP
BP 669 Al Qods,
OUJDA, MOROCCO.

Mohsine ELEULDJ
ENSAO UMP
BP 669 Al Qods,
OUJDA, MOROCCO.

ABSTRACT

In recent years a new concept of IT organization emerged, the Cloud Computing. With This new concept, the resources are dynamically scalable, virtualized and provided to users as a service on the Internet. It is primarily intended to meet the demands of users and allow them access to virtually unlimited resources. This model motivates many academic institutions and non-academics as well to develop open-source solutions to improve performance. Among these techniques, dynamic reconfiguration of cloud resources has to take an interest.

In this paper an approach for optimization resources is presented, based on dynamic reconfiguration techniques. In fact, a Dynamic Reconfigurable Component (DRC) is proposed to be added to the cloud system, that optimize the use of cloud resources and enable dynamic resource allocation. Then, the implementation of this DRC component is provided.

General Terms:

Component, Resource optimization

Keywords:

Cloud computing, Dynamic reconfiguration, Openstack, Dynamic Reconfigurable Component, Autoscaling, KPI

1. INTRODUCTION

In recent years, the Cloud Computing has emerged as a new concept of IT organization, that delivers platform, software application and hardware infrastructure as a service over Internet. It allows the users to utilize the service by on-demand and pay-per-use models.

Currently, the number of cloud computing users is constantly increasing. Leading the providers to furnish more resources to overcome these growing user demands. Providers should intervene over their platforms to ensure a good performance of cloud services. The performance of cloud represents here how the resources must be optimized. It can be measured in terms of scalability, cpu load, throughput, response time, latency, QoS, cost, security, and so on.

In the cloud IaaS (Infrastructure as a Service), the equipment is provided in the form of virtual machines running by a hypervisor software. Each virtual machine is characterized by a set of
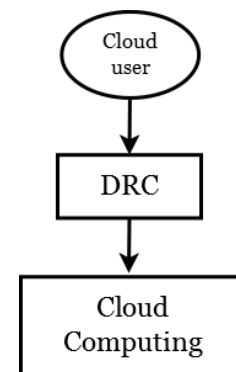

Fig. 1. **DRC integration in Cloud Computing system.**

hardware resources, consisting essentially of CPU, memory and external storage network.

The provisioning of virtual machines is on-demand and dynamically allocated to users.

The general architecture of cloud shows the important place that takes virtualization software in the overall structure of the system. So any optimization at this level reflects positively on the overall performance of the cloud.

Several resource optimization techniques are available, among these, especially there is:

- Live migration [1].

- Load balancing [2].

- Dynamic reconfiguration [3]

The live migration process allows moving virtual machines from one physical node to another without service interruption and with a completely transparent way for the user [1, 4].

Load balancing can generally allocate workloads services while reducing the number of servers and improving performance [4].

Dynamic reconfiguration of virtual machines provide the ability to modify the CPU power, the size of the memory associated with a virtual machine (VM) without stopping its execution [3].

It appears as a new attractive solution particularly for resource optimization techniques.

In cloud system, the clients order resources in the form of a lease, but in general they uses less resources than requested. This is a loss

for the client because of unused resources. and also can be a loss for the provider.

The use of a component is proposed in this paper, based on dynamic reconfiguration techniques. In figure 1 the block diagram of proposed solution is presented. The approach is to provide a system with resizing capabilities allowing the adaptation of desired resources to real needs of clients. Its principal role is to act on the resources delivered to users and to provide an optimized environment. Cloud users can define their own strategy and policy for managing cloud resources according to the established rules. The component is designed for the optimization of material resources and others types of resources(e.g bandwidth, number of connection,...). The component can communicate with different solutions of cloud platforms deployment, using standard REST API.

In the rest of this paper, the cloud computing concept is discussed. After, a review of the related work done in this topics is presented. Then, a presentation of DRC operation is given. Followed by an overview of cloud KPI metric. The DRC implementation and experimental results is presented. Finally, a conclusion and future work are outlined.
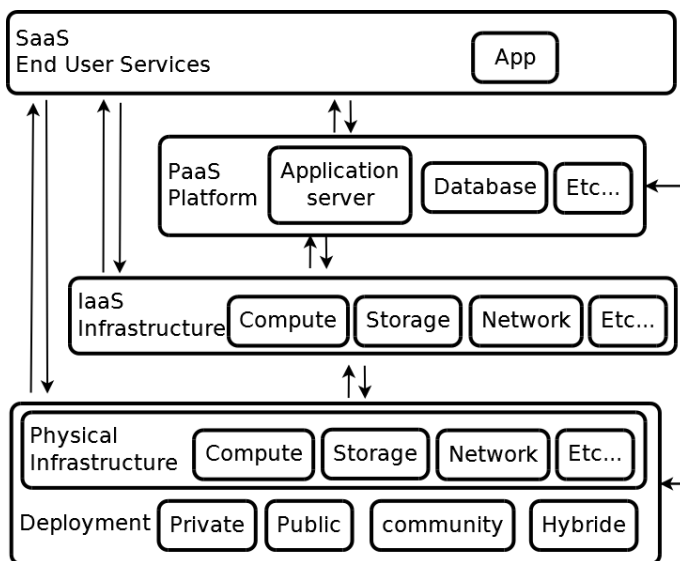
2. CLOUD COMPUTING

Fig. 2. **CLOUD COMPUTING SYSTEM.**

Cloud computing is relatively a new concept that brings together all the disciplines, technologies (Web services, virtualization, SOA: service oriented architecture, grid computing,...) and business models used to deliver IT capabilities (software, platforms, hardware) as a service request, scalable and elastic [3]. This is the new trend of computing where IT resources are dynamically scalable, virtualized and exposed as a service on the Internet [1].

Cloud computing is often associated with the supply of new mechanisms that allow providers to give users access to a virtually unlimited number of resources (*Resource Outsourcing*). It also uses billing mechanisms to use these resources on the basis of their consumption, allowing on-demand model: *pay-per-use* [1]. Warranties are offered by the infrastructure provider through tailored service contract: *Service Level Agreements* (SLA) [1].

Today, all major industry players offer cloud solutions, especially Amazon EC2 [5], Microsoft Azure [6], Google Apps [7] and IBM blue cloud [8].

Cloud computing consists of three levels of offerings: [1]

(1) Infrastructure as a Service (IaaS), where the equipment is provided in the form of virtual machines. The client maintains the applications, runtimes, integration SOA (Service Oriented Architecture), databases, server software while the supplier maintains the Cloud virtualization, hardware server, storage, networks. Among the main actors of IaaS there are Amazon EC2, Rackspace, GoGrid.

(2) Platform as a Service (PaaS), you can develop your own applications using the services provided. The client maintains only those applications while the supplier maintains the runtimes Cloud, SOA integration, databases, server software, virtualization, server hardware and the storage networks. There are among the key players: Google Apps Engine, Windows Azure.

(3) Software as a Service (SaaS), the entire applications are available remotely. Among the providers there are GoogleApps, salesforce, facebook.

The three levels of cloud offering are shown in figure 2, the lower level is the computer hardware resources (computing, storage, network), and mechanisms called virtualization hypervisor, which virtualize access to the material resources of a physical machine (processor, memory and other devices). The interest of a hypervisor is to dynamically add or remove instances of virtual servers on one physical server. This is done using the tools of services and interfaces management. The upper level represents the interactions between the users of the services and the cloud.

There are four different types of deployment of cloud computing [9], which are shown in figure 2:

(1) Public cloud

(2) Private Cloud

(3) Community Cloud

(4) Hybrid Cloud

In the public cloud model, as described in figure 2, the cloud infrastructure is leased to any class of customers, and the infrastructure is owned by the provider. In the model of private cloud infrastructure cloud is operated solely for a single organization with limited access with multiple consumers. In the hybrid cloud model, the cloud infrastructure is a composition of two or more forms of clouds (private, community or public) that enable the portability of data and applications. Finally, in the model of community cloud also called Outsourced private cloud, the cloud infrastructure is provisioned for exclusive use and shared for the specific community (eg the government). It can be owned, managed and operated by one or more organizations in the community.

3. RELATED WORKS

Recent works deal with cloud resource optimization. A new architecture using auto-scaling technology on batch jobs based system is proposed in [10],

In [11], the authors present an Online Resource Management Decision Support System that addresses both tasks scheduling and resource management optimization in a unique system.

Many algorithms have been proposed for optimizing resources. The authors present in [12] the economy based leasing algorithm,

developed and integrated with Haizea [13]. This economy takes care of incentives of customer and service provider.

Another approach consist of an optimal cloud resource provisioning algorithm that is proposed by formulating a stochastic programming model described by [14].

The authors in [15] present a policy based resource allocation in IaaS cloud, proposing a dynamic planning based scheduling algorithm, it is implemented in Haizea that can admit new leases and prepare the schedule whenever a new lease can be accommodated.

Various resource optimization methods were published. The paper [2] propose a Task-based System Load Balancing method using Particle Swarm Optimization. This achieves system load balancing by only transferring extra tasks, from an overloaded VM instead of migrating the entire overloaded VM.

Another appraoch named eviction-free memory based on memory pre-copy algorithm described by [16], to achieve the rapid migration of VM.

iAware, a lightweight interference-aware VM live migration strategy is presented in [17], it empirically captures and understands the relationships between VM performance interference and key influential factors that are practically accessible.

Another algorithm named honey bee behavior inspired load balancing were developed by [18], which aims to achieve well balanced load across virtual machines for maximizing the throughput. The proposed algorithm also balances the priorities of tasks on the machines in such a way that the amount of waiting time of the tasks in the queue is minimal.

4. CLOUD KPI METRICS

The metering process allows the cloud to collect information. With Key Performance Indicators (KPI) measuring tool, the cloud can read the resources used in real time.

The evaluation of overall performance of the cloud is characterized by indicators, that will be measured and evaluated.

The choice of these indicators is very important in the assessment process. The inappropriate choice of these indicators will not really reflect the real performance of the cloud system.

Cloud Computing Services can be evaluated on the basis of KPI, it can be measured using the software and hardware monitoring tools. Here some examples of definitions of the most important KPI are given, particularly in the context of VM instances. [19, 20].

(1) CPU, or CPU_util is defined as the average CPU utilization. by percent %The meter is related to the guest machine.

(2) Network, is the number of incoming and outgoing bytes or packets on the network for a VM interface.

(3) Memory, the volume of RAM in MB.

(4) Ip.floating, represent the duration of floating ip, creating and updating requests for this floating ip.

(5) Disk, Size of root and ephemeral disk in GB.

5. DRC COMPONENT PRINCIPLE

Cloud client typically lease virtual machines that include a fixed amount of resources, such as the number of cores, memory size, and so on. These resources are generally stable throughout the life of the virtual machines.

During use of the cloud, the resources used may be upper or lower to the resources leased by the client. It can cause a gap between
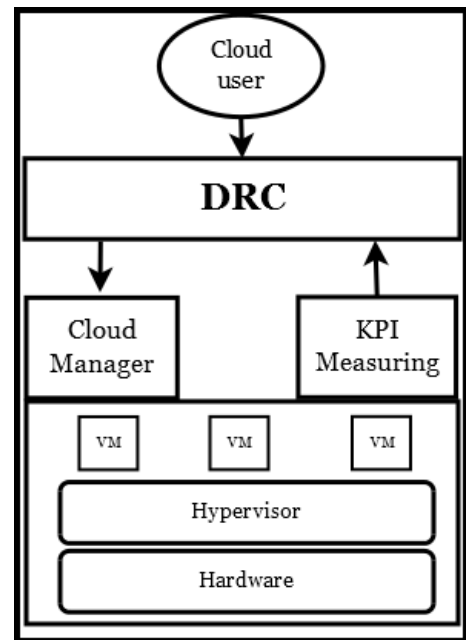

Fig. 3. **ARCHITECTURE CLOUD WITH DRC.**

the needs and provisions. The DRC component comes into play to solve this problem.

The implementation of the Dynamic Reconfigurable Component (DRC) is required to optimize the use of cloud computing resources.

The DRC component is intended to correct the users resources requests by removing unused resources, which results in the difference between the resources requested and used.

The DRC component as shown in figure 3 will read the resources used in real time, these resources are measured in accordance with given frequency by KPI measuring tool. After, the component will assign new values of resources depending on DRC optimization process and inject them to the cloud manager.

The KPI measuring will be used to collect measurements data, and will be transferred to the component for apply policy.

The component is intended not to be a part of the cloud manager. It is acting on the outside of cloud manager. The component is designed to be multiplatforms, it is independent of cloud management tools such as Openstack[21], Eucalyptus [22], Cloudstack[23] and OpenNebula[24]. The DRC component is designed for the optimization of material resources and others types of resources.

The component could be used for the client as well as for the provider.

6. CASE STUDY

The DRC component is implemented as a standalone python application. The DRC interact with the underlying cloud manager via RESTful API, to guarantee the cloud independence and interoperability.

Openstack [25], [21] is the cloud Manager solution selected [26] for deploying cloud infrastructure as shown in figure 4. Ceilometer [27], the OpenStack metering component, is used to collect measurements data. As shown in the figure 4, ceilometer
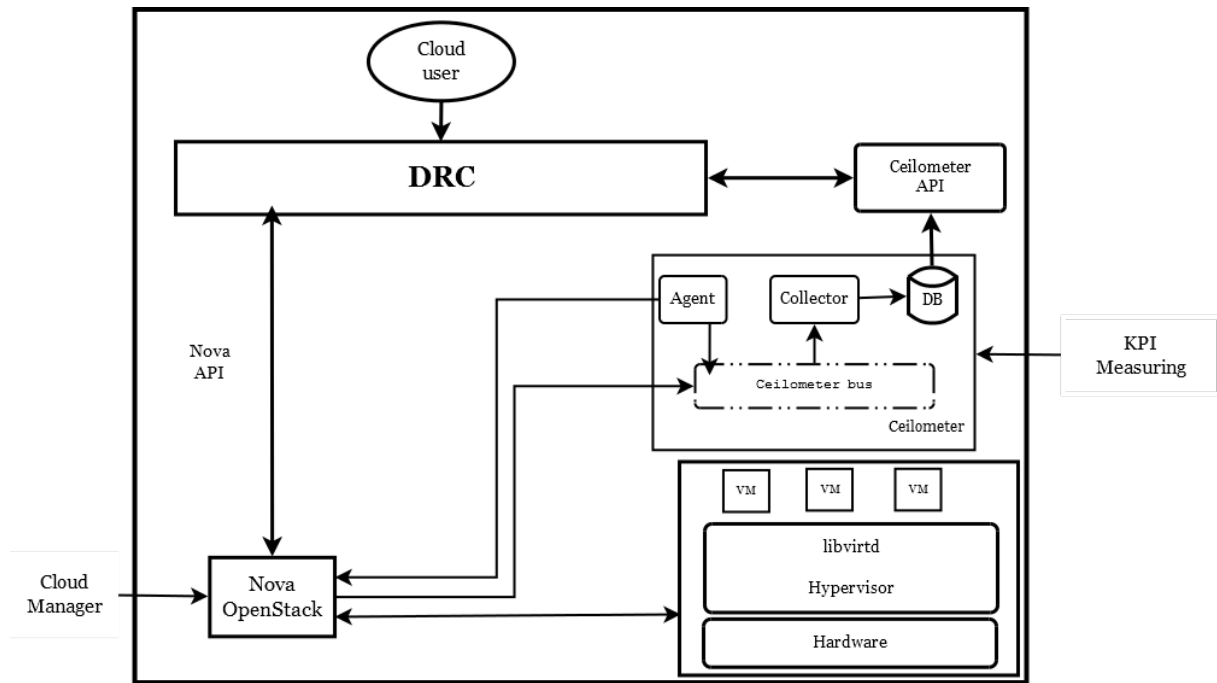
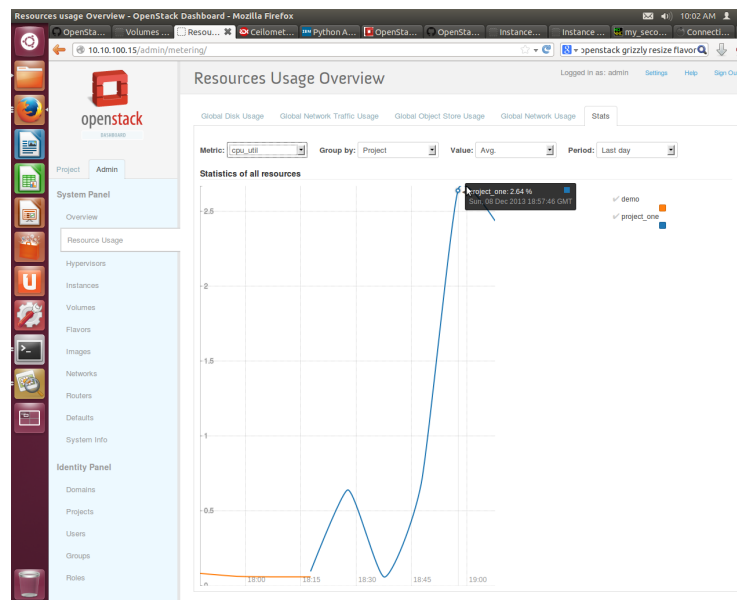Fig. 4. **EXPERIMENTAL ARCHITECTURE OF DRC IMPLEMENTATION.**

Fig. 5. **CEILOMETER CPU LOAD MEASUREMENTS.**

metering includes a compute agent, central agent, collector and data store.

The information about cloud activities are collected by ceilometer module and sent to DRC. The component DRC determine the optimal solutions depending on policy and strategy defined by users. DRC transmit orders to Nova node which act on VM parameters.

In this test the CPU load is taken as an example of KPI, as shown in figure 5. The measurement of CPU load is transferred to the component for apply optimization policy.

Then the CPU load is compared to the threshold defined in DRC. When the threshold is reached, a command is executed to decrease or increase resources with ”resize flavor” directive.

Flavor is an available hardware configuration for a server. It defines the size of a virtual server that can be launched. In OpenStack,

flavors are virtual hardware templates, defining sizes for RAM, disk, number of cores, and so on.

The resize operation converts an existing instance to a different flavor, in essence, scaling the instance up or down. The original instance is saved for a period of time to allow rollback if a problem occurs. When the resize is confirmed, the original instance is removed.

The threshold is represented in two states, up to 70 percent and down to 30 percent of CPU load. CPU load threshold that should not exceed.

7. CONCLUSION AND PERSPECTIVES

In this paper an approach for cloud optimization resources is presented, based on dynamic reconfiguration techniques. The implementation of a DRC component is proposed to act on the cloud resources demanded and to provide an optimized environment based on rules defined by the users.

The DRC is also intended to interact with different cloud platform solutions in transparent way, leading to the independence versus cloud providers. It can be easily amended to include other functionalities, rules and strategies for a better adaptation and integration of the users. The implementation of DRC shows a clear improvement, that brings us to propose to integrate the DRC to the cloud system.

As continuity to this work, to enhance the DRC functionality many artificial intelligence tools will be used to develop an intelligent DRC component, and additional KPI's will be studied and integrated to the cloud measuring.

8. REFERENCES

[1] Lutz Schubert, Keith G Jeffery, and Burkard Neidecker-Lutz. *The Future of Cloud Computing: Opportunities for European Cloud Computing Beyond 2010:–expert Group Report*. European Commission, Information Society and Media, 2010.

[2] Fahimeh Ramezani, Jie Lu, and Farookh Khadeer Hussain. Task-based system load balancing in cloud computing using particle swarm optimization. *International Journal of Parallel Programming*, pages 1–16, 2013.

[3] Luis M Vaquero, Luis Rodero-Merino, and Daniel Morán. Locking the sky: a survey on iaas cloud security. *Computing*, 91(1):93–118, 2011.

[4] Muhammad Atif and Peter Strazdins. Adaptive parallel application resource remapping through the live migration of virtual machines. *Future Generation Computer Systems*, 2013.

[5] Amazon elastic compute cloud, [url]. http://aws.amazon.com/, access on Dec. 2013.

[6] Windowsazure Project [url]. http://windowsazure.com. access on Sep. 2012.

[7] Google App Engine. Cloud google. https://cloud.google.com/, access on Dec. 2012.

[8] Patrícia Takako Endo, Glauco Estácio Gonçalves, Judith Kelner, and Djamel Sadok. A survey on open-source cloud computing solutions. In *Brazilian Symposium on Computer Networks and Distributed Systems*, 2010.

[9] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, 2010.

[10] Yuan Yuan Guo, Jing Li, Xin Chun Liu, and Wei Wei Wang. Batch job based auto-scaling system on cloud computing platform. *Advanced Materials Research*, 756:2386–2390, 2013.

[11] Fahimeh Ramezani, Jie Lu, and Farookh Hussain. An online fuzzy decision support system for resource management in cloud environments. In *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, pages 754–759. IEEE, 2013.

[12] Hemant Kumar Mehta and Eshan Gupta. Economy based resource allocation in iaas cloud. *International Journal of Cloud Applications and Computing (IJCAC)*, 3(2):1–11, 2013.

[13] Haizea project [url]. http://haizea.cs.uchicago.edu/,access on Sep. 2013.

[14] Sivadon Chaisiri, Bu-Sung Lee, and Dusit Niyato. Optimization of resource provisioning cost in cloud computing. *Services Computing, IEEE Transactions on*, 5(2):164–177, 2012.

[15] Amit Nathani, Sanjay Chaudhary, and Gaurav Somani. Policy based resource allocation in iaas cloud. *Future Generation Computer Systems*, 28(1):94–103, 2012.

[16] Pei Han and Qianrang Gong. Optimized live vm migration with eviction-free memory approach. In *Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 1*, pages 185–191. Springer, 2014.

[17] Fei Xu, Fangming Liu, Linghui Liu, Hai Jin, and Bo Li. iaware: Making live migration of virtual machines interference-aware in the cloud. 2013.

[18] LD Babu and P Venkata Krishna. Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied Soft Computing*, 2013.

[19] Saurabh Kumar Garg, Steve Versteeg, and Rajkumar Buyya. A framework for ranking of cloud computing services. *Future Generation Computer Systems*, 29(4):1012–1023, 2013.

[20] Ceilometer measurements. http://docs.openstack.org/developer/ceilometer/measurements.html.

[21] Openstack project [url]. http://www.opnstack.org/, access on Apr. 2012.

[22] Eucalyptus project [url]. http://www.eucalyptus.com/, access on Sep. 2012.

[23] Cloudstack project [url]. http://cloudstack.org/, access on Dec. 2012.

[24] Opennebula project [url]. http://opennebula.org/, access on Sep. 2012.

[25] Omar Sefraoui, Mohammed Aissaoui, and Mohsine Eleuldj. Openstack: Toward an open-source solution for cloud computing. *International Journal of Computer Applications*, 55(3):38–42, October 2012. Published by Foundation of Computer Science, New York, USA.

[26] Omar Sefraoui, Mohammed Aissaoui, and Mohsine Eleuldj. Comparison of multiple iaas cloud platform solutions. *Proceedings of the 7th WSEAS International Conference on Computer Engineering and Applications, (Milan-CEA 13)*. ISBN: 978-1-61804-150-0, 2013.

[27] Openstack component: Ceilometer. http://docs.opnstack.com/ceilometer/, access on Apr. 2013.