2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing

A Compartive Study of Cloud Computing middleware

Chaker El Amrani, Kaoutar Bahri Filali, Kaoutar Ben Ahmed, Amadou Tidiane Diallo, Stéphano Telolahy
Department of Computer Engineering
Abdelmalek Essaadi University organization
Tangier, Morocco
ch.elamrani@fstt.ac.ma

Tarek El-Ghazawi
Department of Electrical and Computer Engineering
The George Washington University
Washington DC, USA
tarek@gwu.edu

***Abstract*—Cloud computing is an emerging IT technology that is being used increasingly in industry, government and academia. There are several Cloud Computing middleware solutions available in the market. This paper proposes an approach and set of characteristics and metrics for comparing Cloud computing middleware based on functionality. Three popular open source middleware: Nimbus, Eucalyptus and OpenNebula, are also analyzed and evaluated on the basis of the proposed parameters. Future work will include more systems and an experimental benchmarking to study the relative performance.**

Keywords-component; cloud middleware; functional characteristics; conceptual metrics; Nimbus; Eucalyptus; OpenNebula.

I. INTRODUCTION

The development of Cloud Computing systems has gone through enormous advances in recent years. Cloud Computing is a model for enabling ubiquitous, on-demand network access to a shared pool of configurable computing resources. Consumers have access to large amounts of data and computational infrastructures through resource management middleware. The users pay based on their usage [1, 2].

Several commercial and open source Cloud computing middleware are available and comparing and contrasting them is not easy. In order to better choose the appropriate solution, a methodology is based on a set of functional characteristics has been developed, and a detailed comparison between three middleware systems has been presented in this paper.

II. CLOUD COMPUTING ARCHITECTURE

A Cloud Platform is composed of physical or virtual servers, storage systems, datacenters, networking devices, hypervisors and middleware. The architecture of cloud services is based on a real-time scalable approach. Cloud computing is the delivery of computing resources as a service, whereby shared resources, software, and information are provided to computers and other devices as a utility over a network [3]. Cloud Computing thus provides to users on-demand services, which can be Software as a Service (SaaS), Platform as a Service (PaaS) or Infrastructure as a Service (IaaS), as shown in "Fig. 1".

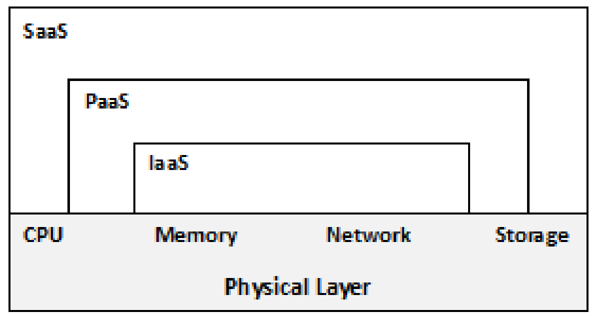

Figure 1. Cloud Computing Overview

IaaS provides machine instances, PaaS provides a programming environment that abstracts machine instances and other technical details, and SaaS, which is the most popular Cloud Computing services, consists on applications offering online resources and storage.

The main techniques used in Cloud Computing are [4]:

- Virtualization: provides on-demand flexible and scalable platforms.
- Workflow orchestration: Cloud should be able to automatically orchestrate services from various sources and of different types to form a workflow for users.
- Web service and SOA: make Cloud available on various distributed platforms and can be accessed through networks.
- Web 2.0: is related to the development of Web-based communities and business models.
- Large distributed storage system: Cloud should include a network storage system and a distributed data system.

Clouds can be divided into three categories: Private Clouds where the infrastructure is dedicated to a particular organization, Public Clouds where resources are shared between multiple organizations and a combination of both called Hybrid Clouds where some resources are in-house and others are provided externally [5].

This work was supported in part by the I/UCRC Program of the National Science Foundation under Grant No. IIP-0706352.

978-0-7695-4691-9/12 $26.00 © 2012 IEEE
DOI 10.1109/CCGrid.2012.129

III. CLOUD MIDDLEWARE FUNCTIONALITIES

Cloud middleware is an abstraction layer that hides system complexity and enables seamless communication mechanism between the cloud computing components [6,7,8].

The most important functionalities and characteristics of Cloud middleware and related definition and benefits are presented in Table I.

TABLE I. MIDDLEWARE FUNCTIONALITIES

Middleware functionalities/ Characteristics	Definition	Benefits
Data replication	storing the same data in multiple devices	- improving performance - increasing availability
Scheduling jobs	Resource allocation and job execution	- Managing the job queue for a cluster - Run-time optimization
Fault tolerance	A transparent takeover of a failed machine by another with continuity of service	- system recovery after outages
Load balancing	Optimizing job distribution on multiple computers.	- Increased availability and data accesses
Interoperability	Support interoperation among different Cloud services	- Easy access to more external services - Easy data exchanges
Security	Support to security (passwords, SSH, X509 certificates, LDAP, etc.)	- Protection against intruders - Securing data exchange
Image disk management	Storing, transferring and cloning image disk	- Avoids loss of data
Resources monitoring	Enabling viewing information about the use of resources in real time	- Controlling the resource usage
Virtualization management	Creation of virtual versions of the system, such as OS, storage device or networks.	- Enabling resource sharing and reliability
User management	adding, deleting or modifying user's account and authorizations	- Controlling user access to cloud services - Managing privileges
Control panel	User interface used to execute tasks	- Easy manipulation

Middleware performance criteria are not considered here and will be treated in future work that will include experimental benchmarking.

IV. CLOUD COMPUTING MIDDLEWARE

Nowadays, there are many commercial and open source middleware. This paper focuses on the comparison between three popular open source solutions: Nimbus, Eucalyptus and OpenNebula.

Eucalyptus (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) uses existing infrastructure to create a scalable, secure web services layer that abstracts CPU, network and storage to provide IaaS. Eucalyptus web services are designed for hybrid clouds. It is composed of several components that interact with one another, enabling developers to modify existing modules. Eucalyptus is an open source implementation of Amazon EC2 (Elastic Compute Cloud) and is compatible with most business interfaces [9].

The OpenNebula is a toolkit to build private, public and hybrid clouds. It has been designed to be integrated with heterogeneous distributed data center infrastructures. OpenNebula orchestrates storage, network and virtualization technologies to deploy multi-tier services, such as groups of interconnected virtual machines, on distributed infrastructures, combining both data center resources and remote cloud computing resources, according to allocation policies [10].

Nimbus is known as a cloud computing solution providing IaaS. It consists on a set of software to be installed to one service node and a separate piece of software to be installed on any number of virtual machine monitor (VMM) nodes. Nimbus provides an implementation of Amazon's Elastic Compute Cloud (EC2) that allows using clients developed for the real EC2 system against Nimbus based clouds. It also allows users to create auto-configuring clusters. The solution provided enables users to even store Virtual Machines (VMs) with no private credentials on-board [11].

V. ESTABLISHMENT OF METRICS FOR MIDDLEWARE COMPARISON

On the basis of the above listed functionalities (Table I), we defined some conceptual metrics to compare cloud middleware, and we assigned a weight to each of them based on its perceived importance from experience. The metrics are the following:

- Virtualization management support: working in a scalable manner under XEN, KVME and VMwre hypervisors.
- Scheduling: support for round robin, first-fit, using batch scheduler such as Haizea, PBS and SGE, and enabling integration of new algorithms.
- Security standards: support of security technologies such as VPN tunnel, SSH authentication and ACL.
- Cloud interfaces: use of standardized interfaces for public cloud such as EC2 and S3.
- Control panel: managing and administering resources with user-friendly and rich interfaces, including command line and web interface, support for users' management and accounting.

Each metric is evaluated based on the number of technologies included. Accordingly, a metric's score is equal to 1 if all the above listed solutions are available and provided.

The comparison of the three open source cloud computing middleware, based on the five metrics is represented in "Fig. 2".

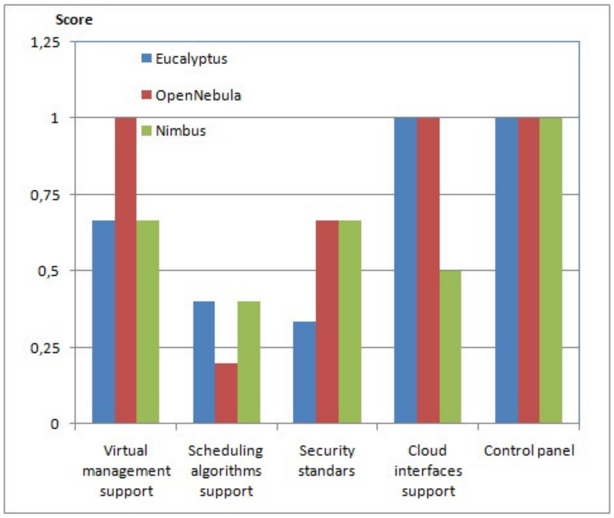

Figure 2. Middleware comparison

From "Fig. 2", it can figure out that the three cloud middleware are different but there are some common between them. For example all provide the same Control panel technologies. However, there are also some differences, for instance in the way each middleware manages the virtualization and security.

According to the cloud architecture and above functionalities, we assigned weights to metrics as shown in Table II.

TABLE II. METRICS' WEIGHTS

Metric	**weight**
Virtualization management support	0.22
Scheduling algorithms support	0.28
Security standards	0.28
Cloud interfaces support	0.05
Control panel	0.17

Among the five metrics, we consider Scheduling algorithms support and Security standards the most relevant ones, followed by Virtualization management support. The Control interfaces support is ranked last, and is preceded by Control panel. The total of weights is equal to 1.

The following formula:

$$Final\_Score = \sum_{i=1}^{5} Score_i * Weight_i \qquad (1)$$

gives a final score, based on the scores found in the previous evaluation "Fig. 2", and the metrics' weights proposed in Table II.

"Fig. 3" shows a comparison between the three open source cloud computing middleware based on the formula (1).

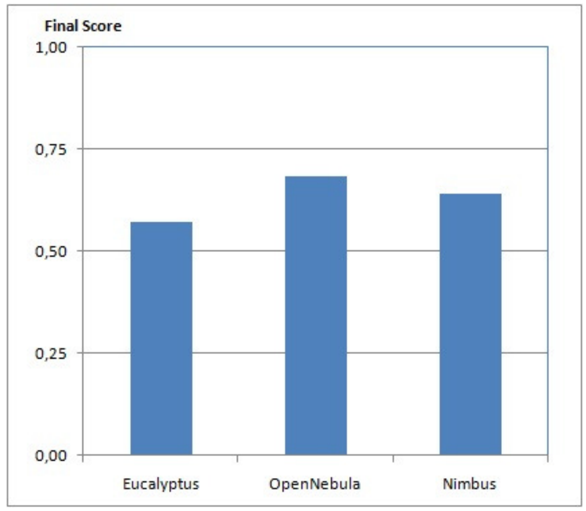

Figure 3. Representation of middleware's final score

According to "Fig. 3", OpenNebula obtained the highest score, followed by Nimbus and Eucalyptus. This means that OpenNebula provides to users the most important and relevant cloud technologies related to Virtualization, Scheduling, Security and Cloud interfaces and panel, comparing to the two other middleware.

This classification was made according to the conceptual metrics.

VI. CONCLUSION AND FUTURE WORK

In order to evaluate different middleware solutions, a set of functionalities, metrics and related weights were presented. Three open source middleware were compared. In spite of its limited scheduling options, OpenNebula seems to consistently compare favorably with the other paradigms. All of the three paradigms provide acceptable basic functionalities, however.

This conceptual study will be completed by a future experimental work with extensive benchmarking and whereby metrics will capture among other variables, average turn-around, response-time, throughput, and CPU and network utilization [12].

ACKNOWLEDGMENT

The authors are grateful to the Fulbright Program for enabling this study to be undertaken by Professors and graduate students from The George Washington University in the United States and Abdelmalek Essaadi University in Morocco.

REFERENCES

[1] F. Feldhaus, S. Freitag and C. El Amrani, “State-of-the-Art Technologies for Large-Scale Computing”, Ch. 1, pp. 1-17, in Werner Dubitzky, Krzysztof Kurowski and Bernhard Schott, Large-Scale Computing Techniques for Complex System Simulations, Wiley-IEEE Computer Society Pr, 2011.

[2] L. Wang, M. Kunze, J. Tao, G. von Laszewski, “Towards building a cloud for scientific applications”, Advances in Engineering Software, Vol 42, pp. 714–722, 2011.

[3] G. Reese, “Cloud Application Architectures: Building Applications and Infrastructure in the Cloud”, O'Reilly Media, 2009.

[4] T. Velte, A. Velte, R. Elsenpeter, “Cloud Computing, A Practical Approach”, McGraw-Hill Osborne Media, 2009.

[5] J. W. Rittinghouse and J. F. Ransome, “Cloud Computing Implementation, Management, and Security”, CRC Press, 2009.

[6] A. Nathani, S. Chaudharya, G. Somani, “Policy based resource allocation in IaaS cloud”, Future Generation Computer Systems, Vol 28, pp 94–103, 2012.

[7] J. Gallard, A. Lebre, C. Morina, T. Naughtonb, S. L. Scott, G. Vallee, “Architecture for the next generation system management tools”, Future Generation Computer Systems, Vol 28, pp 136–146, 2012.

[8] T. Mather, S. Kumaraswamy, S. Latif, “Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice)”, O'Reilly Media, 2009.

[9] http://www.eucalyptus.com/

[10] http://opennebula.org/

[11] http://www.nimbusproject.org/

[12] T. El-Ghazawi1, K. Gaj, N. Alexandridis, F. Vroman, N. Nguyen, J. R. Radzikowski, P. Samipagdi, S. A. Suboh, “A performance study of job management systems”, Concurrency and Computation: Practice and Experience,Vol 16, Issue 13, pp. 1229–1246, November 2004.