ISBN 978-93-84422-71-4
Proceedings of 2016 International Conference on Image Processing, Production and Computer Science (ICIPCS'2016)
London (UK), March 26-27, 2016 pp.62-69

A Genetic Algorithm for Resource Allocation with Energy Constraint in Cloud Computing

Mezache chaabane[1], Kazar Okba[2], and Samir Bourekkache[2]

[1] Kenchela University
[2]Biskra University

***Abstract*:** *The Cloud computing is a novel approach for the deliverance of IT services on the World Wide Web. Where companies are able to rent resources from cloud for storage and other computational purposes so that their infrastructure cost can be reduced significantly. However, one of the major pitfalls in cloud computing is related to optimizing the resources being allocated. Because of the uniqueness of the model, resource allocation is performed with the objective of minimizing the costs associated with it. The other challenges of resource allocation are meeting customer demands and application requirements(SLA). In this paper, we propose an adaptive resource allocation algorithm for the cloud system. . Our algorithms adjust the resource allocation adaptively based on the updated of the actual task executions. And the experimental results show that our algorithms work significantly in the situation where resource contention is fierce.*

Keywords: *Cloud computing, SLA, Virtual machine, resource allocation*

1. Introduction

"***cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction "NIST [1].***

This cloud model aims to promote the availability and it is composed of three service models: Cloud Software as a Service(SaaS), Cloud Platform as a Service (PaaS) and Cloud Infrastructure as a Service(IaaS).

In an IaaS cloud, resources or services are provided to users in the form of leases. The users can control the resources safely thanks to the free and efficient virtualization solutions. While we applaud the numerous opportunities the cloud offers to providers and users, data security and confidentiality, availability of service and effective resource management techniques are some obstacles and challenges to the growth of cloud computing. In order to meet up with both client Service Level Agreement (SLA) for Quality of Service (QoS) and their own operating cost, cloud providers are faced with the challenges of under provisioning (a starvation or saturation of VM resources that leads to service degradation) and over-provisioning (underutilization and subsequent waste of VM resources).

Presently, several works have been done on scheduling of applications in Clouds [2],[3],[4]. These approaches are usually considering one single SLA objective such as cost of execution, execution time, etc. Due to combinatorial nature scheduling algorithm with multiple SLA objective for optimal mapping of workload with multiple SLA parameters to resources is found to be NP-hard [5]. The available solutions are based on the use of heuristics. When a job is submitted to the clouds, it is usually partitioned into several tasks. Following two questions are to be consider when applying parallel processing in executing these tasks: 1) how to allocate resources to tasks; 2) task are executed in what order in cloud; and 3) how to schedule overheads when VMs prepare, terminate or switch tasks. Task scheduling and resource allocation can solve these three problems.

Typically, efficient provisioning requires two distinct steps or processes: (1) initial static planning step: the initially group the set of VMs, then classify them and deployed onto a set of physical hosts; and (2) dynamic resource provisioning: the allocation of additional resources, creation and migration of VMs, dynamically responds to varying workload. Step 2 runs continuously at production time but in contrast Step 1 is usually performed at the initial system set up time and may only be repeated for overall clean-up and maintenance on a monthly or semi-annually schedule. In this paper we focus on dynamic resource provisioning as mentioned above in step 2. In order to attain the agreed SLA objective our proposed algorithm dynamically responds to fluctuating work load by preempting the current executing task having low priority with high priority task and if preemption is not possible due same priority then by creating the new VM form globally available resources.In section II, we discuss works related to this topic. In section III, models for resource allocation and task scheduling in IaaS cloud computing system are presented. We propose our algorithms in section IV, followed by experimental result in section V. Finally, we have mentioned the conclusion in section VI.

2. Related Work

Cloud computing is a new paradigm attracting a growing number of customers through benefits such as economy, security, ease of deployment, improved management of resources (distribution of VMs), etc. In this context, in the Cloud service providers must provide the skills necessary to guarantee customers access to natural resources and meet the requested QoS. Many programming methods have been used to solve the VM allocation problem on a set of servers taking into account the resources provided and consumed by different elements. The remainder of this section presents several works addressing this problem.

In [6] the authors proposed a multi-agent scheduling approach that is based on the inter communication nodes where each node agent tent reserved all the resources available on the nodes of neighborhoods (the neighborhood changes randomly), if all the nodes neighborhood are not reserved the agent becomes responsible for scheduling all the MVS nodes of its neighborhood. In the contrary case, a neighborhood nodes is reserved by another fails and goes round to the other nodes (agents). After the agent takes responsibility for ordering it starts the information gathering operation on the available resources and VMs to replace MVs. He once scheduled MVs agent release all neighboring nodes already booked for that can be reserved by other agents. The evaluation of this approach was achieved through simulations on virtual machines 6048 and 1008 compute nodes.

This approach is partially decentralized, fault tolerant. If any node fails it does not put a stop scheduling; but in fact if the head of scheduling on a neighborhood node fails this puts a scheduling breakpoint on that neighborhood; and nodes are reserved cannot be reserved for another scheduling in other neighborhoods, which will defeat any attempt to schedule these neighborhoods. This approach takes into consideration the actual consumption of resources of the MVs. But the drawback that scheduling periodically triggered every 30 seconds if it is not possible to migrate the VMs must wait 30 seconds to correct the error (limited neighborhood) and make another scheduling.

In this article [7] the authors proposed an approach that aims to optimize energy consumption infrastructure. The approach uses the concept of SMAs for scheduling or each node has an agent that can send one or more scouts who travel to neighboring nodes to obtain information about available resources. Once back on its original node. The officer then decides whether it is possible to reduce a portion of the load virtual machines on its node. This is more likely to occur even if the node is lightly loaded (and vice versa). In case of migration, the agent chooses as destination node that consumes the least power and resources are most used. When the original node is not hosting any virtual machine node is turned off. The simulations were performed on over 1,000 compute nodes but the number of virtual machines is not mentioned in the article.

The work cited in this paper proposes a dynamic scheduling approach completely decentralized virtual machines where each agent order the machines hosted on its node because the agent has a partial knowledge of the state of resources in the system. If a node fails the scout who is there to be regenerated by the originating node to continue scheduling, this mechanism makes the tolerant approach to failure. To minimize the number of

information circulated on the network, the authors emphasize the need to limit the number of nodes visited by a scout. One can notice on this technique scheduling that while scouts are in phase harvest information changes have places on the nodes of origin (virtual machines can be moved, the status of available resource also changes because of the lack agent between synchronization mechanism before moving virtual machines from one node to another), so the Scouts can send invalid data "outdated."

The authors of [8] propose a meta-scheduler that uses a multi-objective genetic algorithm (MO-GA) to find the best scheduling based on three goals: reducing energy consumption and emissions and (QoS) for the client, all with on-time termination applications and the constraints of the model. Their approach uses the geographic distribution of data centers to find the best scheduling, since energy, CO2 levels and different electricity prices from one area to another. The meta-scheduling algorithm proposed uses a multi-objective genetic algorithm in order to find the best scheduling for applications over time. As before each scheduling, the meta-scheduler waits for a fixed period called scheduling cycle. This period brings together a set of applications to have more choice and thus optimize future scheduling results. Once this phase is completed, the application is processed by the group MO-GA to find the best possible schedules on the cloud data center distributes. The result of this calculation is stored in Pareto archive.

This approach was designed to do load balancing, it is partially centralized, and the most critically point is how to fix the scheduling cycle for it is not too small so the information gathering phase is insufficient or very long makes the information gathered on the state of infrastructure resources on winning.

The authors in [9] propose a multi-agent approach that is characterized by: 1-all agent communicate periodically with all the neighborhood, the neighborhood of 2 each agent changes every definite period and randomly .3- when the node agent detects that the node to the least number of virtual machines in neighboring node sends a number of MV as much as it can migrate. This approach has been evaluated through simulations with 40000 and 10000 virtual machine computing nodes.

This approach offers a fully decentralized scheduling mechanism. If a node goes, other nodes can continue to share virtual machines, so it is also fault tolerant. This approach does not address the case where a node resources are over-utilized and does not deal as the rate of energy consumption that affects MVS for travel costs.

The load-balancing algorithms virtual machines are used to assign the different user tasks to a heterogeneous set of virtual machines. These are created in the host data center by the "datacentercontroler" [10] .The balancing algorithms commonly used fillers are "Round Robin Load Balancer," "ThrottledLoad Balancer", "Active Monitoring Load Balancer" and "EffecientLoad Balancer" [11] [12]. The load-balancing algorithms "Round Robin", "ActiveMinoriting" and "Throttled" does not take into account the current load virtual machines for assignments, the heterogeneity of nodes and variability of workloads (in size tasks) [13]. Meenakshi Sharma et al [11] proposed an algorithm by modifying the algorithm "Throttled", this algorithm uses the response time metric as allowance. Calculating the response time is in the initialization phase once for all virtual machines.

In [13] the authors proposed an improvement of the algorithm proposed by Sharma et al [11]. They took into consideration the current state of the workload of the virtual machines, and as a metric allocation, time to treatment and not the response time because this time is calculated based on the transfer time, while that period does not depend on the choice of the virtual machine. The basic idea of the algorithm is to assign tasks to virtual machines in the data center so that the task is allocated to the virtual machine, the runtime expected to be minimal. This approach was designed to do load balancing, it is centralized, when the data center controller fails the scheduling operation stops. But she considers overuse of the MV .the validation of the approach was performed on a simulator with 30 CloudAnalysit physical machine and 50 virtual machine and a simulation time from one hour to six hours.

3. Proposed System

The data center is particularly greedy in electricity consumption if resources are constantly turn are be used. According to [14] an ideal data center consumes 70% of its total power. This loss of power considered a major

cause of energy inefficiency. So it's important to put a medium which allows for efficient energy management of cloud environments through resource allocation planning.

This work is a contribution to the reduction of excessive power consumption by using a resource allocation algorithm (we use the genetic algorithm) with constraint of energy. The purpose of this algorithm to minimize the number of hosts turn by minimizing the number of virtual machine turn on and to have the maximum number of hot to put on standby. Cloud Computing Vision 2015 Intel [15] also highlights the need for such dynamic planning approaches of energy resources to improve the consumption of data center

3.1. General System Architecture

The main component of the proposed system may be located in different data centers as they can be centralized. The main resources are virtual machines, host representing IaaS for deployment of cloud applications. Our architecture system make that each data center has its own resource management system. This means that each data center is able to run all the appropriate customer requests.

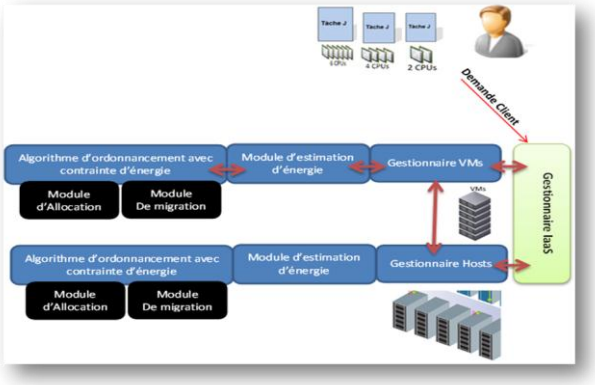

Fig. 1: Global architecture of the proposed system

3.2. Modeling of the Proposed Spproach :

A genetic algorithm is by definition a meta-heuristic can be adapted to many kinds of problems. A clear description of the chosen modeling in the context of this paper is therefore necessary:

• A chromosome represents an investment solution of tasks of virtual machines, respectively, as shown in Fig2.

• A gene represents a task respectively a virtual machine.

• The value for a gene represents the number of the VM the physical machine on which the task respectively the virtual machine was allocated respectively.

• In parallel, task characteristics, virtual machines and physical machines are stored so as to calculate the values of each metric and the fitness value of the chromosome

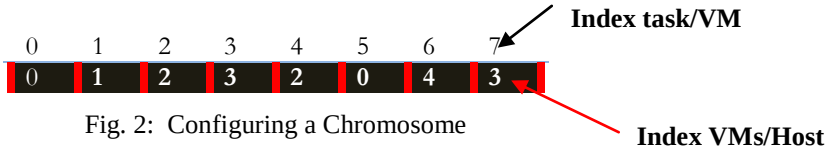

Fig. 2: Configuring a Chromosome

3.3. Cloudlets Model (task):

A Cloudlet (spot) noted J, works in a single virtual machine. its modeling allows you to define the characteristics presented in Table 4.1 and the influence thereof on the virtual machine responsible for its execution. In addition, a cloudlet is characterized by:

• Length expressed in MIPS
• Number of Processors Pes
• RAM size.
• Budget

$$\forall j, v \quad e_{jv} \in \{0, 1\} \tag{01}$$

$$\forall v,j \quad rjv \tag{02}$$

$$\forall v,j \quad mjv \tag{03}$$

•The task is represented by "j", the MV is represented by "v".
• EJV is a binary variable that describes the mapping of job j on the VM v. If EJVs j = 1 task is allocated to the VM v
• Rational variables rjv mjv and represent the percentage amount of fluid resource (respectively rigid) that the task j requires the VM V.

3.4. Virtual Machine Model:

$$\forall j, h \quad e_{jh} \in \{0, 1\} \tag{04}$$
$$\forall J \quad V_j \in \{0, 1\} \tag{05}$$
$$\forall h,j \quad rjh \tag{06}$$
$$\forall h,j \quad mjh \tag{07}$$

The VM is represented by "j", the host is represented by "h".
• EJH is a binary variable that describes the mapping of the VM on the host h j. If EJH = 1 MV j is allocated to the host h.
• Vh is a binary variable that can describe the state of the VM V. If V = 1 V VM is on, if V = 0 V VM is off.
• Rational variables rjh mjh and represent the percentage amount of fluid resource (rigid respectively) j requires that the VM on the Hot h. Here, we will have rjh representing the percentage of CPU that the application VM j on the hot hours, and in the same way mjh the memory percentage required by theVM j on the machines of hot h.

3.5. Physical Machine Model:

$$\forall h, k \quad E_{hk} \in \{0, 1\} \tag{08}$$
$$\forall h \quad P_h \in \{0, 1\} \tag{09}$$
$$\forall h,j \quad \alpha jh \tag{10}$$

The physical machine (in) is represented by "h" and the cluster is represented by k.
• Ehk is a binary variable that describes the mapping of the hot "h" on the "k" cluster. If ehk = 1 belongs to the host cluster K.
• pH is a binary variable that can describe the state of the host h. If ph = 1 h host is on, if ph = 0, host h is off.
• Finally, we have the αjh variable representing the fluid resource amount actually allocated to the vm j on host h. Note here that we will not have similar variable for memory, it is considered rigid, and therefore can not be different from mjh.

Static allocation:

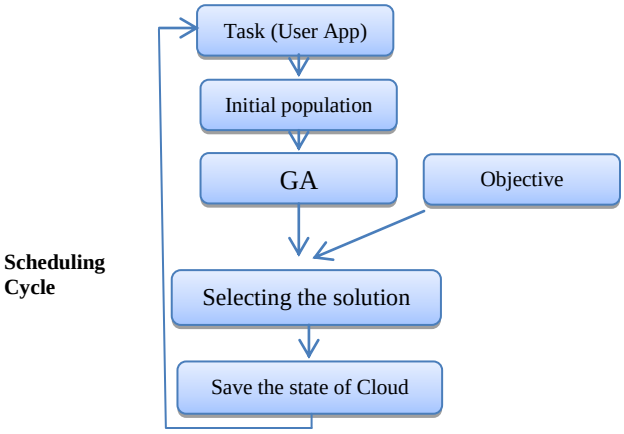

Fig. 3: Flow chart of the allocation algorithm

Where Objective Function is:

$$F_1 = \min\left(\frac{\sum_{j=1}^{v} MIPS_{Alouer}}{TotalMIPS}\right)\} \tag{11}$$

$$F_2 = MinNbrVms(F_1) \tag{12}$$

$$F_3 = Min\left(\frac{\sum_{h=1}^{H} E(h)}{H}\right) \tag{13}$$

$$E(h) = C_h^{\min} H + (C_h^{\max} - C_h^{\min}) \sum_{j=1}^{h} \alpha_{jh} \tag{14}$$

3.6. Dynamic Allocation

The goal for the algorithm is to select all the VMs that need to be migrated from a set of machines called sources (that the algorithm is to put them off) to a destination other set of machines that the ability to receive these VMs.

Collaboration between the genetic algorithm VMs investment and the migration algorithm

Figure 4-10 summarizes how the algorithms of allocation and migration have combined to achieve a min energy consumption in infrastructure nodes and therefore the data center.

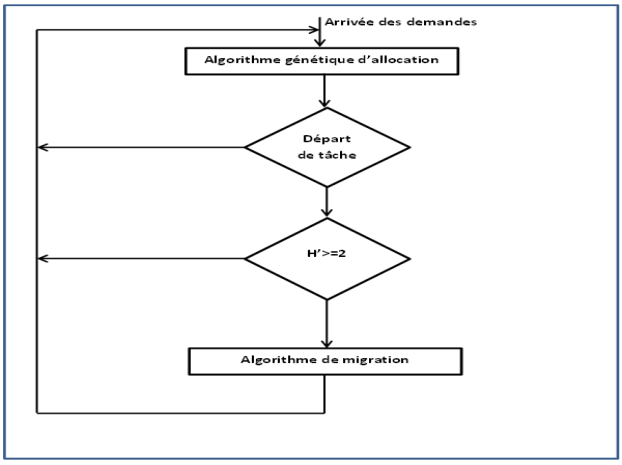

Fig. 4: collaboration allocation algorithm and migration vms

4. Results

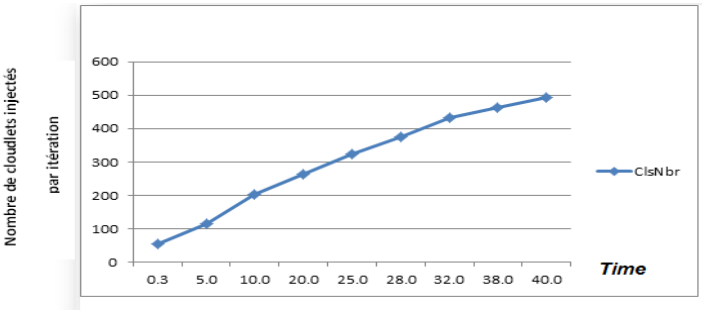

Fig.5: Number of cloudlets injected for each iteration

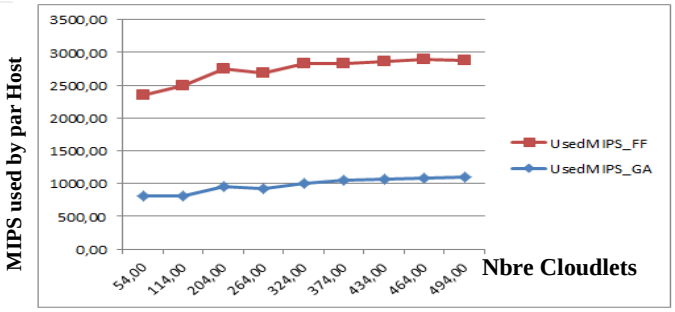

Fig 6: Free MIPS By host

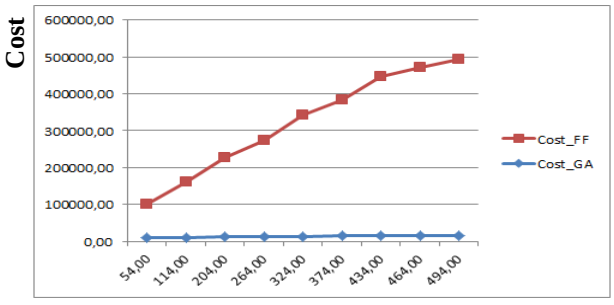

Fig 7: Evaluation of energy cost for each algorithm

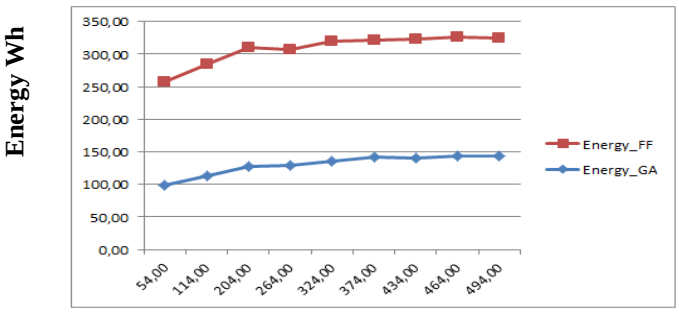

Fig 8: Evaluation of energy consumption for each algorithm

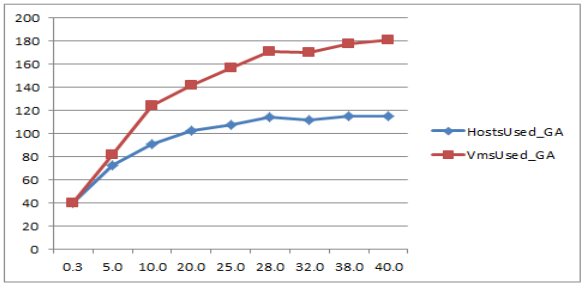

Fig 9: Number of VM and Host are used

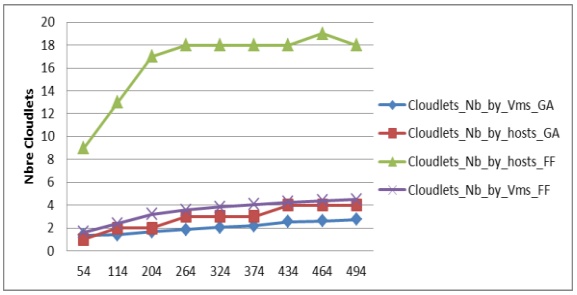

Fig 10: Number of cloudlets Evaluation by Host and VM

5. Conclusion

Simulation results obtained with the genetic algorithm applying optimization showed the influence of energy model of each physical machine on the overall cost of consumption and the ratio of the number of cloudlets and charge of the virtual machine which influences on the number of lit physical machine.

Throughout the production of this work, we have identified various potential routes extension of our work.

• Implementation of the management model of energy on any device. One of our prospects is to apply the broad principles of resource manager of design for other devices (disk, RAM, network card).

• Taking into account the constraints related to the topology of the application, in particular the link capacity between virtual machines. We also plan to study the effects of different network architectures and organization of physical machines on the real-time service quality (machine placement and robust network fault, etc.). This generally leads to large size optimization problems for which appropriate optimization methods should be designed.

6. References

[1] Timothy Grance, Peter Mell, "The nist definition of cloud computing (Draft) ," January 2011.

[2] S. K. Garg, R. Buyya, and H. J. Siegel, “Time and cost trade off management for scheduling parallel applications on utility grids,” Future Generation. Computer System, 26(8):1344–1355, 2010.

http://dx.doi.org/10.1016/j.future.2009.07.003

[3] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, “A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments,” in AINA ’10: Proceedings of the 2010, 24th IEEE International Conference on Advanced Information Networking and Applications, pages 400–407, Washington, DC, USA, 2010, IEEE Computer Society.

http://dx.doi.org/10.1109/aina.2010.31

[4] M. Salehi and R. Buyya, “Adapting market-oriented scheduling policies for cloud computing,” In Algorithms and Architectures for Parallel Processing, volume 6081 of Lecture Notes in Computer Science, pages 351–362. Springer Berlin/ Heidelberg, 2010.

http://dx.doi.org/10.1007/978-3-642-13119-6_31

[5] J. M. Wilson, “An algorithm for the generalized assignment problem with special ordered sets,” Journal of Heuristics, 11(4):337–350, 2005.

http://dx.doi.org/10.1007/s10732-005-3208-6

[6] Armel Esnault, Eugen Feller, Christine Morin, "A case for fully decentralized dynamic vm consolidation in clouds," In CloudCom ’12: 4th IEEE International Conference on Cloud Computing Technology and Science, Washington, DC, USA,IEEE Computer Society, 3-6 December 2012.

[7] Elisabetta Di Nitto, Daniel Dubois, Raffaela Mirandola, Donato Barbagallo, "A bio-inspired algorithm for energy optimization in a self-organizing data center," In Danny Weyns, Sam Malek, Rogério de Lemos, and Jesper Andersson,editors,Self-OrganizingArchitectures Springer Berlin/Heidelberg, Germany, vol. 6090 , 2010.

[8] Nouredine Melab, El-Ghazali Talbi, Yacine Kessaci, "Optimisation multi-critère pour l’allocation de ressources sur clouds distribués avec prise encompte de l'énergie," In Rencontres Scientifiques France Grilles , 2011.

[9] Ozalp Babaoglu, Fabio Panzieri, Moreno Marzolla, "Server consolidation in Clouds through gossiping," InWoWMoM'11: Proceedings of the 12th IEEE International Symposium on a World ofWireless, Mobile and Multimedia Networks,Washington IEEE Computer Society, June 2011.

[10] Rodrigo N. Calheiros, Rajkumar Buyy, Bhathiya Wickremasinghe, "Cloud Analyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and applications," In 24th International Conference on Advanced Information Networking and Applications (AINA), IEEE Computer Society, p. 446-452, 2010.

[11] Pankaj Sharma, Dr. Sandeep Sharma, Meenakshi Sharma, "Efficient load balancing algorithm in vm cloud environment ," In International Journal on Computer Science and Technology, vol. 3, pp. 439-441, 2012.

[12] Sarabjit Singh, Meenakshi Sharma, Sandeep Sharma, "Performance analysis of load balancing algorithms," In International Journal of Advanced Computer Science and Applications, vol. 3, pp. 86-88, 2012.

[13] Mohammed Oumsis, Abdelkader ElMahdaouy, "Evaluation et amélioration de performances des algorithmes d'équilibrage de charges dans un environnement Cloud Computing," In JDTIC 2012 : Les 4èmes Journées Doctorales en Technologies de l'Information et de la Communication, Casablanca, Morocco, Oct 25-27 2012.

[14] Erica Naone, "Conjuring clouds ," Technology Review, 112(4):54–56, 112(4):54–56, 2009. Intel's cloud computing 2015 vision. http://www.greenit.fr/article/acteurs/hebergeur/data-center-56-de-consommation-electrique-entre-2005-et-2010-3890.