Journal of Theoretical and Applied Information Technology
31st May 2015. Vol.75. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THE TASKS ALLOCATION BASED ON THE PRE-ESTIMATION OF THE PROCESSING TIME IN THE CLOUD ENVIRONMENT

YOUSSEF FAHIM[1], ELHABIB BEN LAHMAR[1], EL HOUSSINE LABRIJI[1], AHMED EDDAOUI[1]
[1]Hassan II University - Mohammedia - Casablanca, Faculty of Sciences Ben M'sik
Laboratory of Information Technology and Modeling Casablanca Morocco

E-mail: Fahimyoussef87@gmail.com, h.benlahmer@gmail.com, labriji@yahoo.fr, ahmed_eddaoui@yahoo.fr

ABSTRACT

The Cloud computing has reduced the performance and profitability of the materiel and software resources. These challenges were solved by several load balancing algorithms between the virtual machines of the data centers. In fact, we find three main points of research in this article:
Data center: It receives queries from clients of Cloud service to run them, using its own virtual machines in a sequential or parallel way. Sharing resources, is one of the main characteristics of the data center in the Cloud computing.
Virtual machine (VM): Is a virtual device that has some performances determined by the Cloud provider. Such as memory storage capacity and the performance of its processor for the processing of tasks desired by the data center.
The load balancing algorithms: they select the virtual machines in a data center for a new allocation of tasks, according to the predetermined metrics in each algorithm. Generally, we find two types of algorithms, static and dynamic. But, there is a problem when it comes to the current load of each virtual machine.
In this paper, we propose a new improvement of the load balancing by our algorithm «pre-estimated processing time», that allows cloud service providers, to improve the performance, availability and maximize the use of virtual machines in their data centers. The key point of our improvement, it's that take into account, the current load of the virtual machine of a data center and the pre-estimation of the processing time of a task before any allocation.

Keywords: *Cloud computing; Virtual machine; Load balancing; pre-estimated processing time.*

1. INTRODUCTION

Cloud computing allows the providers to offer their customers: platforms, infrastructures and software as services [7, 13].

The data centers are physical and heterogeneous units of calculation, in a form of several virtual machines aggregations, interconnected via heterogeneous networks [1, 5, and 6]. This variety of interveners in the execution of a user tasks, make the providers of Cloud services face the challenge of the quality of services.

Cloud computing has reduced the performance and profitability of the materiel and software resources [1, 8]. Our algorithm «pre-estimated processing time » is one of the solutions suggested to meet these needs taking into account some performances parameters of the load balancing algorithms; parameters like: stability, error tolerance, overloads detection, cooperation, tasks migration, allocations and compatibility of the calculated results with the values generated after the execution [2, 10].

In this paper, precisely in section 2, we will present a state of art for the load balancing algorithms. This section will be divided into 2 sub-sections. The first, illustrates the static algorithms. The second deals with the dynamic algorithms. By the way, our synthesis will be mentioned in section 3, but the sub-section 3.1 will show the characteristics of our algorithm « pre-estimated processing time ». The section 4 is at the same time, the conclusion and a glance at our perspectives.

2. STATE OF ART

The performance parameters are parameters with which we measure and test the profitability and the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

performances of several static or dynamic load balancing algorithms.

The static load balancing algorithms, share the client query between virtual machines in a data center for the processing. But, there is a problem when it comes to the current load of each virtual machine.

The dynamic algorithms as "Efficient Response Time Load Balancer" and "Mini time processing load balancer" prove to be a solution to respond to this problem. The advantage of these algorithms, before allocating a task, is that they search in the allocation tables on the virtual machine, with a metric which is quite inferior (be it the response time or the processing time) [13].

2.1 Statics Algorithms

Before every allocation of tasks to the virtual machine, the static load balancing algorithms determine the performances and the distribution of the loads, and the data center controller receives the results at the end of every processing.

One of the disadvantages of these algorithms is that they do not take into account the current load of the virtual machine during the implementation.

Examples of these algorithms:

A. Round Robin Load Balancer: The queries in this algorithm are distributed between virtual machines with one after the other, relying on the help of the data center controller [3].

The allocation order of the tasks takes place in each virtual machine locally and independently of the other remote machine, on the basis of the available number of tasks and the number of virtual machines [10].

B. Throttled Load: This algorithm gets information about the state of the virtual machine to decide whether it is valid or not for the new allocation, then the algorithm sends the identifier (ID) of the valid virtual machine to the data center, for a new allocation and implementation of the task. If the processing of the task is completed, the virtual machine sends the result to the data center controller which notifies the algorithm for dis-allocation [9].

C. Active Minoriting Load Balancer: This is an algorithm which counts the minimum number of tasks assigned to each virtual machine. It sends the ID of the machine to the data center controller. This controller informs the algorithm to start the allocation and the incrementation of its table by the new number of tasks assigned to the machine having the ID negotiated [9].

D. Threshold Algorithm: The algorithm determines the status of virtual machines in two types "tunder" or "tupper".

Each virtual machine has private load copy of the whole system. This copy can be presented as follows:

- Under load: if the load is below the status "Tunder".
- Medium: if the load is between the status "Tunder" and "tupper".
- Overload: if the load of the virtual machine is superior than the status "tunder" [10].

At the beginning, all virtual machines are in the status “under load”, when the virtual machine exceeds this status, the algorithm sends a message to the data center controller to inform it, and as a result, the controller directs the allocation towards a machine which has an inferior status.

The major drawback of this algorithm is that it does not take into consideration the exceptional case where all the virtual machines get simultaneously to the “overload” status.

2.2 Dynamic Algorithms

The most interesting point about these algorithms, before any allocation of tasks, is that they search in their allocation tables on the virtual machine with an inferior metric (a short processing or response time compared with the other algorithms) [4], In contrast to static algorithms, the load balancing takes place dynamically during the execution and the tasks are allocated to machines with the inferior metric [10]. In cloud computing, the dynamic load balancing algorithm is executed by all nodes present in the system and the task of load balancing is shared among them. The interaction among nodes to achieve load balancing can take two forms: cooperative and non-cooperative [15].

A. Efficient Response Time Load Balancer: This algorithm is based on an inferior response time for the allocation of the virtual machine by the data center. First, « Efficient Response Time Load Balancer » detects the machine which has the inferior response time. Second, it returns the ID of the virtual machine to the data center controller for the new allocation. Last, when the task is finished, the controller notifies the algorithm to update the allocation table [9].

B. Central Queue Algorithm: The data center controller has a central queue, in which the tasks

Journal of Theoretical and Applied Information Technology
31st May 2015. Vol.75. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

are classified in a FIFO order (first in first out). If a virtual machine has passed into the status "under load", it sends a request of a new allocation task to the data center controller, then, this controller removes the task from the queue and sends it directly to the requester machine [10].

C. Local Queue Algorithm: With the help of this algorithm, all virtual machines will have local queues. When one of these virtual machines turns to the under-load mode, the local load manager searches for other tasks from the other remote virtual machines.

The advantage of this algorithm is the dynamic migration and allocation of all tasks loaded into the data center controller towards the virtual machines [11].

D. Mini time Processing Load Balancer: The developer of this algorithm has improved the design of the « Efficient Response Time Load Balancer » [9], by a new algorithm "Mini Time Processing "[4], which takes into account the current state of the virtual machine workload, thanks to the time processing, which is a main metric in this algorithm [4].

E. Biased Random Sampling: in this algorithm a virtual graph is constructed, each server is symbolized as a node in the graph and the connectivity of each node representing the load on the server [14]. The most keys points about this algorithm are:

- Whenever a node executes a job, it deletes an incoming edge, which indicates reduction in the availability of free resource.
- After completion of a job, the node creates an incoming edge, which indicates an increase in the availability of free resource.

The walk starts at any one node and at every step a neighbor is chosen randomly.

3. SYNTHESIS

After this careful study of the different load balancing algorithms, the comparison between its performances parameters and according to the figure 1 below we notice that:

- For the dynamic algorithms like "Efficient Response Time Load Balancer" or "Mini processing time load balancer", the updating of the allocation table by the new status of the virtual machines is made only after the end of current tasks processing [4, 9].

- The load balancing, that is based only on the status of the virtual machines is insufficient to measure the degree of imbalance (DI) among VMs [12].

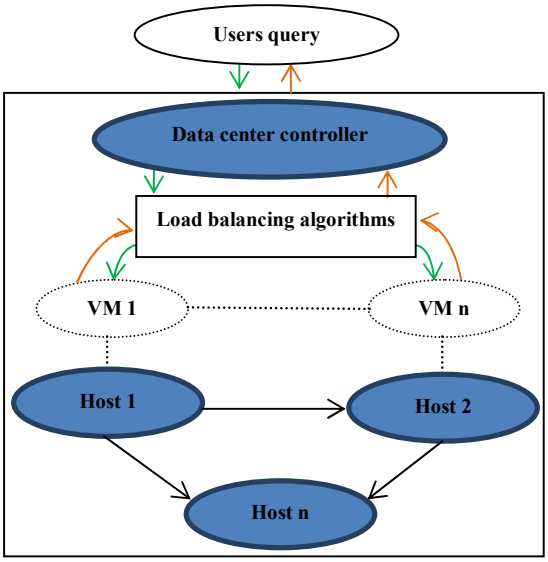

Figure 1. The data center core

In the figure 2 below, we suggest probable states, which show, clearly, the weakness of the dynamic algorithms that are concentrated in each new allocation, only, on the current virtual machine workload, and not also, on the task characteristics, in order to minimize the waiting time of tasks in queue and decrease the degree of imbalance among the virtual machines.

Journal of Theoretical and Applied Information Technology
31st May 2015. Vol.75. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

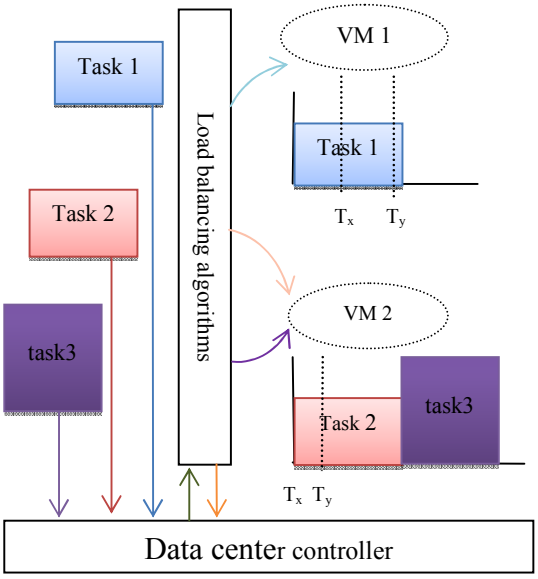

Figure 2. Probable states of allocation

State 1: according to the metric stored in the allocation table; In a certain time T_X, the dynamic algorithms direct the allocation of tasks 2 towards another virtual machine 2, which has a metric that is inferior to the machine 1 [4,9].

State 2: The virtual machine 1 (VM1) continues the processing of the tasks related to it, and the new machine 2 (VM2) starts the processing. The data center controller informs the algorithm about the new metric of the virtual machines at every end of task.

State 3: In a time T_y, the virtual machine 2 has a inferior metric according to the allocation table, so, the algorithm allocates the task 3 to the VM2, but really, the virtual machine 1 is most suitable for this allocation compared to MV2, because this allocation of the task 3 is based on the old parameters of the allocation table before its update. To update the allocation table, the algorithm must wait the end of processing of task 1.

3.1 Pre-estimated Processing Time

The state 3 of the figure 2 pushed us to think of the improvement of algorithms that are proposed by Sharma [9] and El mehdaouy [4], using our dynamic algorithm « pre-estimated processing time ». We take into consideration, the estimate of processing finish time of tasks (T.f.es) before each allocation, like an essential metric for the best load balancing among the virtual machines. The figure 3 below, explains this in detail.

The steps of our algorithm are as follows:

a) Randomly assign to each virtual machine a task and initialize the processing time in the allocation table.

b) At the arrival of new queries and based on its characteristic: the algorithm classifies the tasks and estimates its finish processing time (T.f.es). After a negotiation among the virtual machines, the algorithm « pre-estimated processing time » finds the machine that gives the shortest estimated time of task finish. If not, step a.

c) The algorithm returns the ID of the machine identified to the data center controller.

d) The controller, after receiving the ID, starts the new allocation.

e) The algorithm updates the allocation table by incrementing the number of tasks assigned to the virtual machine, then, it returns to step b.

f) If the virtual machine finishes processing the query, and the controller of the data center receives the response, the latter notifies the algorithm to return to step b. If not, step

g) with : $T.f.es_{n,i} = \sum_{1}^{i-1} T.es_{n,i} + T.es_{n,i}$

If $T.arr_{n,i} + T.f.es_{n,i} < T.act+\epsilon$ so step h
Else : step f.

h) The reallocation of the task i of the machine n, towards another virtual machine (n+1) according to step b.

With $T.es_{n,i}$ is the estimated time for the processing finish of task i in the virtual machine n. $T.arr_{n,i}$ is the starting time of the task i processing, in the virtual machine n. $T.f.es_{n,i}$ is the estimated final time for the finish of task i processing in the virtual machine n. T.act+ ϵ is the current time + the constant ϵ which represents a short time of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

prolongation in the form of a predefined constant. MV (n) is the virtual machines with a number n.

According to what we suggest, in the figure 3 below, our algorithm « pre-estimated processing time » will be able to minimize the waiting time of tasks in queue and decrease the degree of imbalance (DI) among the virtual machines, thanks to the calculation of the estimated final time for the finish of task processing, before each allocation. While:

$$DI = (T.f.es_{max} - T.f.es_{min}) / T.f.es_{moy}$$

Where $T.f.es_{max}$ and $T.f.es_{min}$ are the maximum and minimum final time for the finish of task processing among all VMs, $T.f.es_{moy}$ is the average final time for the finish of task processing among all VMs.

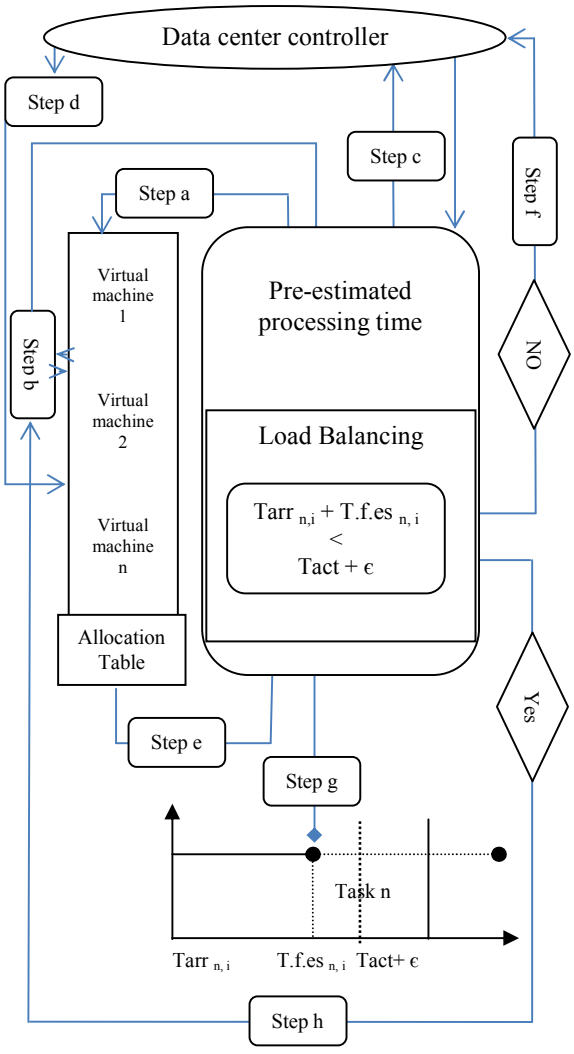

Figure 3. Diagram of pre-estimated processing time

4. CONCLUSION

The improvement, which we proposed, allows the Cloud services providers to improve the performances, the availability and to maximize the use of the virtual machines of their data centers. It permanently controls the current load of the virtual machine and the characteristics of the task during the processing and allocation, in order to avoid a probable blocking of tasks in queue. The most interesting point about our algorithm, before any allocation of tasks, is that it calculates and estimates the finish time of task processing.

Our improvement of load balancing in cloud computing has many perspectives, in terms of implementation and validation. Mainly, to classify the tasks according to its characteristics ,to calculate the estimated finish time of the task processing before allocating it to the virtual machine and ,finely, to provide the proof of our improvement by an experimental results in our next works.

REFRENCES:

[1] Yousri Kouki, Damián Serranoy, Thomas Ledoux, Pierre Sensz, Sara Bouchenak 2013, " SLA et qualité de service pour le Cloud Computing " : ComPAS'2013 (RenPar/ SympA/ CFSE) Grenoble, France, from 16 to 18 January 2013.

[2] Ghislain Charrier, Yves Caniou 2009, " Ordonnancement et réallocation de tâches sur une grille de calcul " : RenPar'19 / SympA'13 / CFSE'7 Toulouse, France, From 9 to 11 September 2009.

[3] Jaspreet kaur, "Comparison of load balancing algorithms in a Cloud": Jaspreet kaur/ International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com, Vol. 2, Issue 3, May-Jun 2012, pp.1169-1173.

[4] Abdelkader El Mahdaouy, Mohammed Oumsis, " Evaluation et amélioration de performances des algorithmes d'équilibrage de charges dans un environnement Cloud Computing ".

[5] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya, "CloudSim: AToolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", Software Practice &

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Experience, Vol 41 Issue 1, pp. 446-452, January 2011.

[6] K.Keahey, M.Tsugawa, A.Matsunaga, andJ.A.B.Fortes, "Sky computing", IEEE Internet Computing, vol.13, pp.43–51, October 2009.

[7] Gatien Gaspard, Rémi Jachniewicz, Julien Lacava, Vincent Meslard, "Équilibrage de Charge et Haute Disponibilité", LP ASRALL ,22 April 2009.

[8] pascale vicat-blanc, olivier glück, cyril otal, françois echantillac, "emulation of a grid network cloud: ewan", institut national de recherche en informatique et en automatique, n° 5449, December 2004, issn 0249-6399.

[9] Meenakshi Sharma, Pankaj Sharma, Dr. Sandeep Sharma,"Efficient Load Balancing Algorithm in VM CloudEnvironment", International Journal on Computer Science and Technology, vol. 3, pp.439-441, Jan. - March 2012.

[10] Sandeep Sharma, Sarabjit Singh, Meenakshi Sharma,"Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology 38 2008.

[11] William Leinberger, George Karypis, Vipin Kumar, "Load Balancing Across Near-Homogeneous Multi-Resource Servers", 0-7695-0556-2/00, 2000 IEEE.

[12] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang," Cloud Task scheduling based on Load Balancing Ant Colony Optimization", College of Computer Science and Technology, Sixth Annual ChinaGrid Conference 2011.

[13] Ram Prasad Padhy (107CS046), PGoutam Prasad Rao (107CS039)."Load balancing in cloud computing system" Department of Computer Science and Engineering National Institute of Technology, Rourkela Rourkela-769 008, Orissa, India May, 2011

[14] Martin Randles, Enas Odat, David Lamb, Osama Abu- Rahmeh and A. Taleb-Bendiab,"A Comparative Experiment in Distributed Load Balancing", 2009 Second InternationalConference on Developments in eSystems Engineering.

[15] Ali M. Alakeel, "A Guide to Dynamic Load Balancing in Distributed Computer Systems", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.