Two Level Load Balancing Strategy in Cloud

Amal ZAOUCH*, Faouzia BENABBOU, Naoufal ER-RAJI

Information Technology and Modeling Laboratory, Science Faculty Ben M'sik, Casablanca, Morocco

Abstract
Load balancing is the efficient distribution of incoming client requests in a server pool or virtual machines. For example, it allows high-audience website servers not to be overloaded. One of the challenging scheduling problems in Cloud load balancing is the choice of Virtual Machines (VM) when assigning tasks or migrating VM. This paper proposes a load balancing strategy for distributed cloud data center to be applied in two levels control: Physicals Machines and Clusters. Load balancing is done by two managers; they ensure exchange information and decide afterwards the level concerned with load balancing. Measurement of load is based on load information, including CPU utilization and memory utilization. Virtual Machines are allowed to be migrated between different federations to distribute loads while the communication costs are also incurred. Therefore, the objectives of this strategy are twofold: reducing the load of the overloaded physicals machines and decreasing the communication costs among different federations. The proposed method attempts not only giving a good Cloud balancing but also ensures reducing response time and communication cost and enhancing performance of the whole system.

Key words:
Cloud Computing, Load balancing, Hierarchical, Communication, Overhead

1. Introduction

The Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility over the Internet. A Cloud datacenter can be considered a hierarchical system in structure, which is composed of many clusters and each cluster contains one or more physical machines (PMs), in every PM runs some virtual machines (VMs). virtualization technology has emerged to facilitate data center infrastructure which has become a vital in cloud computing environments. This technology is more flexible, managing the resources is easy since resources can be utilized efficiently by more virtual machines on a physical Host. virtual machines can be increased or decreased based on user requirements. The key challenge of service providers is managing these physical and virtual resources in a holistic manner. Cloud computing delivers a pool of computing resources to process the huge tasks with minimum cost [1].

Load balancing is the mechanism of distributing the load among various nodes of any system [2]. Major goal of load balancing is optimal utilization of available resources. With virtualization, Cloud datacenters should have ability to balance the load at each level in a hierarchical manner. Such agility becomes a key in modern cloud computing infrastructures that aim to efficiently distribute the load among resources. Otherwise, virtualization and live migration of VMs on PMs are key enablers of efficient resource allocation in data centers. Live migration of a VM from one PM to another makes it possible to react to the changing resource requirements of the VMs. Therefore, it is important to limit the number of live VM migrations. For these reasons, VM migration is a key ingredient of load balancing problem [3]. Understanding the exact impact of live migration is a difficult problem on its own. Hence, VM migrations and task assignment must find the optimal balance between Quality of Services or user satisfaction and cost of communication.

Virtual machines (VMs) have become one of the basic building blocks of datacenters due to cost savings, elasticity, and ease of administration. They are used to provide Infrastructure-as-a-Service (IaaS) to support cloud computing and high performance-computing (HPC) applications [4, 5].

Day to day operation of datacenters relies upon live migration of VMs to deal with overload during peak hours by packing VMs into fewer physical machines, or to perform maintenance of the physical machines. The scale of live migration can range from migrating VMs across only a few physical machines in the same cluster to entire datecenter. Live migration is a network intensive activity that involves the transfer of tens to hundreds of gigabytes of memory over the network. This traffic can overload the core network links and switches within the datacenter Ethernet and degrade the performance of other network-bound applications whose packets traverse the core links. Live migration traffic also consumes the bandwidth at the source and target network interfaces and competes with the bandwidth requirements of applications running within the VMs [6].On one hand, an administrator may want to maintain application-level quality of service of VMs by accompllishing requests in a reasonable reposnse time. On the other hand, the administrator may also wish to decrease cost due to live migration by reducing the total number of migration of VMs .Both objectives can be aided if we can find the an optimum solution that meets this challenge.

The remainder of the paper is organized as follows: section 1 an introduction. Section 2 describe load balancing

Manuscript received August 5, 2019
Manuscript revised August 20, 2019

problem. Section 3 discusses related work. Section 4 contain detailed description of proposed strategy. In section 5 two algorithms are designed and finally, conclusion and future work are outlined in section 6.

2. Load Balancing Problem

The load is an abstract concept describing the busyness of the system, and load distribution of all resources of parallel system is called load balancing. Load balancing contributes to assure the high efficiency of task assignment algorithm, and it can always adjust the load assignment to keep all resources in the system in balanced state. There are two kinds of techniques applied for the load balancing: load assignment and load migration [7].

- The load assignment is to properly assign the user tasks to all resources to make the system load on all resources roughly equal.
- The load migration is to migrate the tasks from heavy-loaded resources to light-loaded, so that the system load gets balanced and the overall performance is improved.

The load balancing algorithm proposed in this paper uses load migration.

Based on spatial distribution of nodes, we can find three types of algorithms that specify which node is responsible for balancing load in cloud computing environment:

- Centralized load balancing technique, where all the allocation and scheduling decision are made by a single node.
- Distributed load balancing technique, where no single node is responsible for making resource provisioning or task scheduling decision; but every node in the network maintains local knowledge base to ensure efficient distribution of tasks in static environment and re-distribution in dynamic environment.
- Hierarchical load balancing which involves different levels of the cloud in load balancing decision. Such load balancing techniques mostly operate in master slave mode [7] and that is the approach we adopted here.

The hierarchical load balancing can be modeled using tree data structure wherein every node in the tree is balanced under the supervision of its parent node. Master or manager can use light weight agent process to get statistics of slave nodes or child nodes. Based upon the information gathered by the parent node provisioning or scheduling decision is made [7]. One of the challenges of load balancing algorithms is the overhead which determines the amount of overhead involved while implementing a load balancing system, it is composed of overhead due to VM migration or communication cost [8]. A well-designed load balancing algorithm should reduce overhead. The main objective of load balancing methods is to speed up the execution of applications on resources whose workload varies at run time in unpredictable way. Hence, it is significant to define metrics to measure the resource workload.

3. Related Works

There are quite many researchers conducted in the area of load balancing solution. Most of them focus on load balancing in one level of cloud computing. One of the challenges of load balancing in the cloud is to consider other levels for load distribution other than virtual machines, indeed physicals machines and clusters.

In [9] Tian et al introduced a dynamic and integrated resource scheduling algorithm (DAIRS) for balancing VMs in Cloud. They considered two level physical machines and virtual machine for load balancing operation. This algorithm treats CPU, memory and network bandwidth as integrated resource with weights. They also developed a new metric, average imbalance level of all the hosts, to evaluate the performance under multiple resource scheduling. The main drawback of DAIRS is that it ignores the communication cost of migrations Malhotra and al.[2] proposed an adaptive Framework for load balancing to improve the performance of cloud environment. This algorithm uses of intelligent agents for keeping record of load on virtual machines and for the balancing load two levels are considered: VM and data center. Predefined threshold value is used in allocation VMs decision. If the VM load is up the limit defined, the agent proposes another data center that has the minimum data transfer time. This work proposes an adaptive solution, which enhances the performance of the distributed systems, but it does not consider the number of migrations between nodes or data centers.

Hao and al. [10] proposed a load balancing scheme based on the minimum value of standard deviation, and is implemented at three levels: Datacenter, host and Processing Elements (PE). The results in CloudSim indicate that the method gives a good Cloud balancing and ensures makespan and communication overhead reduction and enhancing throughput of the whole the system. But they did not consider the load produced by memory occupancy. The authors fixed standard deviation of load distributions and searched all task assignments possibilities, and this may be costly. Furthermore, they did not pay attention to load migration.

In [11] Dhu and al presented a distributed algorithm for load balancing in the master-slave architecture, where masters correspond to datacenters and the slaves correspond to VMs. The load balancing strategy should be such that the load is equally distributed across all

datacenters. Simulations shown that the load is distributed efficiently when using cluster based load-balancing approach compared to the closest Datacenter strategy. However, the model may require more transfer time for execution of certain tasks; hence, cost communication will increase as well.

J. Hu et al. in [12] proposed the scheduling algorithms to balance the load of virtual machine resource using genetic algorithm. The authors have tried to found the least loaded virtual machine and tried to reduce the migration cost. Due to dynamic changes in user requests and the presence of large number of virtual machines, there are chances of load wastage. But this solution makes requests waiting longer in the queue list and response time will increase.

Xu et. al. [13] introduced a model for load balancing in cloud by using the game theory. This algorithm is based on cloud partionning. They divided the cloud into three categories idle, normal and overloaded based on load degree. The authors has introduced an agent based model using decision theory. The migration concept used in this architecture transfers the load from overloaded nodes to under loaded nodes, considering all the VMs in the different datacenters so the transfer will be costly.

In [14] tasks are migrated from overloaded to under loaded VMs to balance the load as well as fulfill the customer's expectation. The algorithm does not take into account the problem of overloading virtual machines at peak time; It ignores communication costs when demand increases.

In addition to the considered levels that contribute in load balancing operation; most reviewed papers have either focused on the VM load balancing at the initial placement stage without considering live migration or paid attention to load migration without considering tasks assignment. The efficient load balancing solution in real cloud environment must consider both of load assignment and load migration as well.

In this paper, we pay attention to the load balancing on cloud and propose a two levels control strategy. The two levels include clusters and physicals machines where a hierarchical load balancing strategy is developed. Unlike traditional load-balancing scheduling algorithms, which consider only Virtual Machines with one factor such as CPU load, our proposed model treats CPU, and memory integrated for both physical machines (PM) and clusters.

4. Proposed Cloud Load Balancing Strategy

Our proposition is based on two purposes that can be summarized as follows:

- To provide a balancing strategy at physical machines' level in the same cluster to reduce communication costs.
- To provide a load balancing at the clusters' level if the system is in a saturation state to decrease response time.

In the following section, we will describe the architecture of our system as well as the load balancing strategy.

4.1 Architecture Components

The figure 1 presents the cloud computing architecture based-on our strategy. The architecture is composed of a data center, local managers with physicals machines PM, and global manager with clusters. In every PM runs some VM. Physicals machines in different clusters are heterogeneous so the load in one cluster may be very high while the other clusters may have nothing running on it.

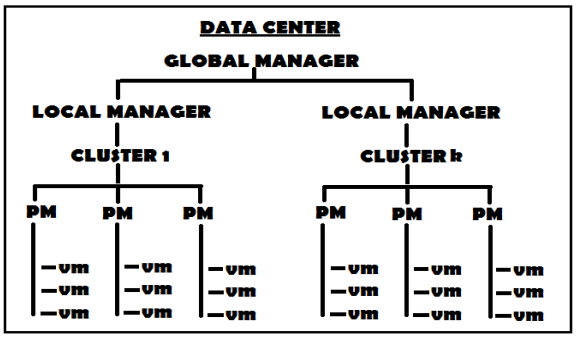

Fig. 1 Architecture of proposed strategy

Referring to the structure of proposed model in figure 1, the load balancing strategy is hierarchical and we present two load balancing levels based on two managers local and global.

Local manager: collect the load information from the physicals machines and balance the load in that local area. It balances the load for the physicals machines and this to avoid additional communication costs.

Global manager: collects periodically the load information from every cluster. The global manager balances the load for clusters and communicates periodically with local managers to collect information load of each cluster.

Throughput will be improved if load migration is done at first within physicals machines within a cluster then within a datacenter. So, more than one level is considered to improve the performance of the whole system. According to the architecture, we can achieve two level of load balancing:

- Intra-cluster load balancing: in this level, load balance concerns only one cluster. This process is achieved only if locals manager fail to load balance its workload among their respective virtual machines.
- Inter-cluster load balancing: the load balance or the load migration at this level is used only if some clusters fail to load balance their load among their

associated PMs. Clusters receivers will be selected according to the throughput.

This strategy is designed to improve the overall resources utilization of each cluster and to reduce the Inter-process communication overhead inter cluster. Therefore, the objectives of this algorithm are twofold: reducing the load of the overloaded machines and decreasing the communication costs by migrating load locally in the cluster at first and improve response time by reducing number of migrations among different federations if the system becomes saturated.

4.2 Load Estimation and Information Exchange Policy

Several load indices have been proposed in the literature, like CPU, queue length, average CPU etc. The success of a load-balancing algorithm depends from stability of the number of messages (small overhead), low cost update of the workload, and short mean response time, which is a significant measurement for a user [2]. It is also essential to measure the communication cost induced by a load balancing operation. Ideally, the load information should reflect the current CPU utilization, memory utilization and network traffic of a node. In the proposed work, CPU and memory utilization are used to estimate the load. In most cases a prerequisite is that VMs in a local PM can communicate much faster than VMs among different PMs [14]. In the proposed strategy, information exchange is reduced due the hierarchical structure of the architecture; local exchange is prioritized till maximum machine capacity utilization is reached in case of saturation and then, clusters' level is responsible to distribute load.

- Cost of communication calculation: It can be expressed by the following formula [14]:

$$\text{Comm.} = \frac{1}{t} * nbr\_msg \quad (1)$$

Where nbr_msg the amount of messages exchanged every time step t.

- Physical machines' utilization:

$Up(t)$ is the utilization of host pj at time t:

$$Up(t) = a * U\text{cpu}(t) + (1 - a) * U\text{mem}(t) \quad (2)$$

Where $U\text{cpu}(t)$ is CPU utilization of physical machine pj and $U\text{mem}(t)$ is memory utilization of machine pj and a is a coefficient representing the relative importance between CPU utilization and memory utilization. As both CPU and memory are equally important, a is set to 0.5.

So load on PM at time t is computed by the following equation [15]:

$$\text{l}(t) = U(t) * C\text{i} \quad (3)$$

Where Cj is the computation capacity of machine pi, which is the frequency of the PM's CPU mapped onto million instructions per second (MIPS) ratings of each core [15].

- Load on Cluster:

Suppose that one cluster k has n physicals machines as $\{pm_{k,1} \ldots. pm_{k,n}\}$. Suppose that the load of PM is l_i and $\bar{l}$ is the average load of each PM. When we calculate standard deviation of load, we pay attention to physicals machines, which is, belongs to the cluster. The standard deviation of load in one cluster is given by:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(li - \bar{l}\right)^2} \quad (4)$$

4.3 Load Measurement

Here we present the parameters used in the proposed strategy. If the standard deviation value is small, it means that the difference of each load is small. The small standard deviation tells that the load of the entire system is balanced. The lower value the standard deviation has, the more load balanced the system has [10]. The information exchange policy chosen for this research is a periodic policy with a time interval that will be set in the simulation experiments.

4.4 Process Transfer Policy

In the proposed strategy, this determination includes two steps. First the physicals machines are classified according to their load to overloaded, balanced or under loaded. Second, clusters are classified to saturated, overloaded, balanced or under loaded. Physicals machines composed one cluster can remain overloaded even if VM live migration is proceeded; in this case cluster is considered saturated and workload will be distributed in cluster's level.

5. Algorithms

This section presents the three distributed scheduling algorithms. The proposed work allows us to develop a hierarchical strategy at two levels and is designed: inter-cluster, intra-cluster.

5.1 Load Balancing Decision

After finding the workload and standard deviation, the system should decide whether to do load balancing or not. For this, there are two possible situations: (1) Finding whether the system is balanced (2) Finding whether the whole system is saturated or not (The whole group is overloaded or not). If overloaded, load balancing is

meaningless. We have to define threshold of balanced state noted T1 from which we can say that the cluster is in overloaded state; and T2 a threshold when we consider the cluster in saturated state. So, two thresholds values T1and T2 are considered to decide the process transfer policy and according to the formula (4); Cluster's state is as follow:

```
If σ ≤ T1
        Cluster is balanced
Else
If σ >T1 and σ<T2
        Cluster is overloaded.
Else
        Cluster is saturated.
End
End
```

Here two threshold value T1 and T2 are used to decide about level concerned with load balancing and then threshold values are calculated using standard deviation value of clusters load multiplied by representative weight coefficient and are:

$$T1 = \alpha * \sigma \quad (5)$$

$$T2 = \beta * \sigma \quad (6)$$

with $0 < \alpha < 1 - \beta$ *and* $\alpha < \beta \leq 1$

Generally, thresholds values are decided by a data center administrator based on the computing capabilities of each machine and dynamic behavior of applications and services, such as in Red Hat [16] and VMware [17].

5.2 Intra Cluster Load Balancing

In this level load balancing is launched only when some VM's managers fail to balance locally the overload of their VMs. Knowing the global state of each PM, the local manager can evenly distribute the global overload between its physical's machines.

```
For every PM of Cluster and periodically do
Calculate load of PM according to formula (1)
Update actual load l of PM
Send it to local manager
End For
For each cluster
Update actual load of Cluster
Send it to global manager
Receive Data center average load from global manager
Calculate the standard deviation of load on each Cluster
Calculate Thresholds Values T1&T2. according to formula (4) and (5)
If σ ≤ T1
    Cluster is balanced
Else
    If σ >T1 and σ<T2
    Cluster is overloaded.
    Algorithm sorts Physical Machines in ascending order of utilization
    Transfer load from overloaded PM to idle PM
  Else
      if σ> T2
          Cluster is saturated.

          Call intra data center load balancing algorithm
      Else
          System balanced
  End if
While PMs overloaded and idle PM exist do
  Sorts idles PMs in ascending order of workload l
  Sorts overloaded PMs in descending order of workload l
  Transfer load from the overloaded PM to the idle one for balancing load in one cluster
done
END
```

5.3 Inter-Cluster Load Balancing Algorithm

This algorithm performs a global load balancing among all clusters of a date center. It is executed only if the other level is failed to achieve a complete load balance.

Inter-cluster load balancing: in this second level it performs a global load balancing among all clusters of a cloud datacenter. It is executed only if the other level are failed to achieve a complete load balance. The main advantage of this strategy is to prioritize local load balancing first (within a cluster, then within a datacenter). The goal of this strategy is to decrease the amount of messages between physicals machines. As consequence of this goal, the overhead induced by our strategy is reduced.

```
For every cluster and periodically do
Calculate standard deviation of load of cluster according to formula (4)
Send it to global manager
End for
Update global datacenter workload
Compute datacenter average load
Send it to all clusters
If σ> T2
  Cluster saturated
  Datacenter overloaded
Else
  Return
End if
While clusters overloaded and idle exist do
  Sorts idles clusters in ascending order of workload
```

Sorts overloaded clusters in descending order of workload
Transfer load from the saturated cluster to the idle one for balancing load in whole structure
Call intra cluster load balancing algorithm
Done

6. Conclusion and Future Work

This paper proposed a hierarchical strategy load balancing for Cloud datacenters. The main objective is to achieve a significant benefit in response time and a minimum communication cost. A dynamic model is proposed with the global manager at higher level and the local manager at next. It's a technique that can be used to improve the performance of cloud computing by balancing the workload across all the nodes in the cloud with maximum resource utilization, in turn reducing overhead and communication cost. Since this work is a conceptual model, more work is needed to implement the algorithm and evaluate its performance. We intend to evaluate the proposed strategy in Cloudsim simulator and control variations of response time and communication cost. Furthermore, we will compare the proposed solution with others algorithms in terms of t metrics of quality of service.

References

[1] Zhang, J., Huang, H., and Wang, X. (2016) "Resource provision algorithms in cloud computing: a survey." Journal of Network Computing. Applications. 64: 23–42

[2] Malhotra, M., & Singh, A. (2015, February). Adaptive Framework for Load Balancing to Improve the Performance of Cloud Environment. In Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on (pp. 224-228). IEEE (2015)

[3] Mann, Z. Á . (2015). Allocation of Virtual Machines in Cloud Data Centers—A Survey of Problem Models and Optimization Algorithms. ACM Computing Surveys (CSUR), 48(1), 11. (2015)

[4] Amazon Elastic Compute Cloud.http://aws.amazon.com/ec2.

[5] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The eucalyptus open-source cloud-computing system. In Proc. of Cluster, Cloud, and Grid Computing, May 2009.

[6] Deshpande, Umesh, Unmesh Kulkarni, and Kartik Gopalan. "Inter-rack live migration of multiple virtual machines." Proceedings of the 6th international workshop on Virtualization Technologies in Distributed Computing Date. ACM, 2012.

[7] Katyal, Mayanka, and Atul Mishra. "A comparative study of load balancing algorithms in cloud computing environment." arXiv preprint arXiv: 1403.6918 (2014).

[8] Xu, Minxian, Wenhong Tian, and Rajkumar Buyya. "A survey on load balancing algorithms for virtual machines placement in cloud computing." Concurrency and Computation: Practice and Experience 29.12 (2017): e4123.

[9] Tian, W., Zhao, Y., Zhong, Y., Xu, M., & Jing, C. (2011, September). A dynamic and integrated load-balancing scheduling algorithm for Cloud datacenters. In Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on (pp. 311-315). IEEE (2011)

[10] Hao, Y., Liu, G., & Lu, J. (2014). Three Levels Load Balancing on Cloudsim.International Journal of Grid and DistributedComputing, 7(3), 71-88 (2014)

[11] Dhurandher, Sanjay K., et al. "A cluster-based load balancing algorithm in cloud computing." 2014 IEEE International Conference on Communications (ICC). IEEE, 2014.

[12] J. Hu, J. Gu, G. Sun, T. Zhao (2010), "Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", in Proc. PAAP, pp. 89-96.

[13] 3. G. Xu, J. Pang, X. Fu. (2013, Feb). A Load Balancing Model Based on Cloud Partitioning for the Public Cloud. Tsinghua Science and Technology.[Online].18(1),pp. 34-39

[14] Song, Xiao, Yaofei Ma, and Da Teng. "A load balancing scheme using federate migration based on virtual machines for cloud simulations." Mathematical Problems in Engineering 2015 (2015).

[15] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," Concurrency Computation Practice and Experience,vol. 24, no. 13, pp. 1397–1420, 2012.

[16] Red hat: Red hat enterprise virtualization 3.2technical reference guide 22015. URL https://access. redhat.com/site/documentation/enUS/Red_Hat_Enterprise_ Virtualization/3.2/ html/Technical_Reference_Guide/index.html

[17] Vmware distributed resource scheduling 2015. URL http://www.vmware.com/au/products/ vsphere/features/drs-dpm