ISSN 2237-8960
Brazilian Journal of Operations & Production Management

PROPOSING A LOAD BALANCING ALGORITHM WITH AN INTEGRATIVE APPROACH TO REDUCE RESPONSE TIME AND SERVICE PROCESS TIME IN DATA CENTERS

Sanaz Hosseinzadeh Sabeti
sanaz_hzs@yahoo.com
University of Applied Science and technology, Tehran, Iran.

Maryam Mollabgher
mollabagher@gmail.com
Faculty Member of University of Applied Science and technology, Tehran, Iran

ABSTRACT

Goal: Load balancing policies often map workloads on virtual machines, and are being sought to achieve their goals by creating an almost equal level of workload on any virtual machine. In this research, a hybrid load balancing algorithm is proposed with the aim of reducing response time and processing time.

Design / Methodology / Approach: The proposed algorithm performs load balancing using a table including the status indicators of virtual machines and the task list allocated to each virtual machine. The evaluation results of response time and processing time in data centers from four algorithms, ESCE, Throttled, Round Robin and the proposed algorithm is done.

Results: The overall response time and data processing time in the proposed algorithm data center are shorter than other algorithms and improve the response time and data processing time in the data center. The results of the overall response time for all algorithms show that the response time of the proposed algorithm is 12.28%, compared to the Round Robin algorithm, 9.1% compared to the Throttled algorithm, and 4.86% of the ESCE algorithm.

Limitations of the investigation: Due to time and technical limitations, load balancing has not been achieved with more goals, such as lowering costs and increasing productivity.

Practical implications: The implementation of a hybrid load factor policy can improve the response time and processing time. The use of load balancing will cause the traffic load between virtual machines to be properly distributed and prevent bottlenecks. This will be effective in increasing customer responsiveness. And finally, improving response time increases the satisfaction of cloud users and increases the productivity of computing resources.

Originality/Value: This research can be effective in optimizing the existing algorithms and will take a step towards further research in this regard.

Keywords: Cloud computing, load balancing, load balancing algorithms, Hybrid algorithm

ABEPRO
DOI: 10.14488/BJOPM.2019.v16.n4.a8

Brazilian Journal of Operations & Production Management
Volume 16, Número 4, 2019, pp. 627-637
DOI: 10.14488/BJOPM.2019.v16.n4.a8

1. INTRODUCTION

The structuring and implementation of the Industry 4.0 context is currently undergoing an evolution process and presents companies to the trend of a new business model format. Essentially, the Industry 4.0 environment has a high degree of technological development and collaborative structure, which is characterized mainly by the communication between different agents (hardware, software, data, people), allowing the exchange, storage, and interpretation of data in an intelligent system (Cordeiro et al., 2019)

Today, cloud computing has become common in IT, and is one step after the evolution of the Internet. Cloud computing provides an enormous amount of storage and computing services to users through the Internet. Cloud computing has emerged as a popular computing model for hosting large-scale computing systems and services. Recently, significant research on resource management techniques, focused on optimizing cloud resources among several users, has been provided. Resource management techniques are designed to improve the various parameters in the cloud (Dhanasekar et al., 2014).

The basic technology for cloud computing is the "virtualization" that separates resources and services from the underlying physical layer to provide multiple dedicated resources in the form of a virtual machine. The term cloud also refers to this basic concept (Barzegar et al., 2014). In the cloud environment, almost all virtualization resources are virtualized and shared among multiple users (Arianyan et al., 2015). "Virtualization is, in fact, the implementation of computer software that runs different programs just like a real machine. Virtualization has a close relationship with the cloud, because an end user can use cloud services through virtualization (Padhy and Rao, 2011).

Load balancing is an essential operation in cloud environments. Because cloud computing is growing fast and many customers all over the world are demanding more services and better results, load balancing is an important and necessary area of research. Many algorithms have been developed for allocating customer requests to available remote nodes. Effective load balancing ensures the efficient resource productivity of resources for customers according to demand" (Panwar and Mallick, 2015).

In this paper, cloud computing and load balancing were first identified as one of the methods for resource management in cloud computing. Then, by examining some load balancing algorithms, a load balancing algorithm that balances the workload on virtual machines using an integrative approach of available load balancing algorithms will be presented.

2. CLOUD COMPUTING

According to the definition of the National Institute of Standards and Technology (NIST), cloud computing is a model for providing easy access to a set of changeable and configurable computing resources, such as networks, servers, storage spaces, applications, and services, that is accessible through the network based on user request rapidly and with the least management operation and the least interaction, providing services (Sahu et al., 2013).

NIST cloud reference architecture components

- **Cloud Provider:** The person, organization or entity responsible for making a service available to cloud users. The architecture introduced by the NIST consists of four major components. The provider has six components: security, privacy, cloud services management, service layer, physical resources layer and control layer, and resource abstraction. Cloud services management includes business support, supply and configuration, and portability and collaboration. Also, in the service layer, SaaS, PaaS, and IaaS are the main and most commonly used models in cloud computing. Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Table 1 shows consumer activities and cloud providers in these service models.

Table 1. Activities in the service model (Bohn, 2011)

Service model	Consumer activities	Provider activities
Software as a Service (SaaS)	Using the program / service to conduct the business process	Installing, managing, maintaining, and supporting application on a cloud infrastructure
Platform as a Service (PaaS)	Developing, testing, creating and managing applications hosted by the cloud environment	Providing and managing cloud infrastructure and middleware for the platform user; provides development and management tools for platform users.
Infrastructure as a Service (IaaS)	Creating / installing, managing, and supervising services for infrastructure operations	Providing and managing physical processors, memory, network and host environments, and cloud structure for IaaS consumers.

Brazilian Journal of Operations & Production Management
Volume 16, Número 4, 2019, pp. 627-637
DOI: 10.14488/BJOPM.2019.v16.n4.a8

- **Cloud Consumer:** A person or organization that uses services of cloud providers to establish a business relationship

- **Cloud Auditor:** This is a component that can assess the behavior of cloud services, information system performance, efficiency, and security independently of cloud implementation. In the NIST architecture of security audit, the impact of privacy and efficiency are in this section. Regarding security, a cloud auditor can have an assessment of security controls in the information system in order to determine how well controls are implemented and how activities are taken into account and produce satisfactory results considering meeting the system needs.

- **Cloud Broker:** Is an entity that manages the use, performance and delivery of cloud services, and negotiates relationships between cloud providers and cloud consumers. In the NIST reference architecture, there are intermediary services, aggregation and connection, and transactions.

- **Cloud Carrier:** The intermediary that provides connectivity and transport of cloud services between cloud providers and cloud consumers.

Implementation models in cloud computing

The implementation models in cloud computing include:

- **Private Cloud:** In this type of cloud, corporate employees can access company or colleague data.

- **Community Cloud:** When multiple companies share their resources in the cloud, the created cloud is called community. This type of cloud is used by organizations with similar interests and common security needs.

- **Public Cloud:** Anyone from anywhere in the world can access it. Examples of these clouds are Google Cloud, which is open to everyone after a specific service level agreement (SLA) between the provider and the user. Public cloud is available based on a common ground.

- **Hybrid Cloud:** It is a combination of both public and private clouds.

3. LOAD BALANCING

Load balancing is the process of reallocating the entire load to the unique nodes of a collective system. Its objective is to effectively utilize resources, to improve the response time of a task, and to simultaneously eliminate a situation in which a number of nodes are strongly loaded, while others have a low load. Load balancing is a mechanism to increase service level agreement and better use of resources. The load considered can be the CPU load, the amount of memory used, the delay, or the network load. In fact, the goal of load balancing is to find a proper task mapping on system processors, so that the overall run time in each processor, almost a same amount of task, reaches its lowest amount (Kaur and Kaur, 2015).

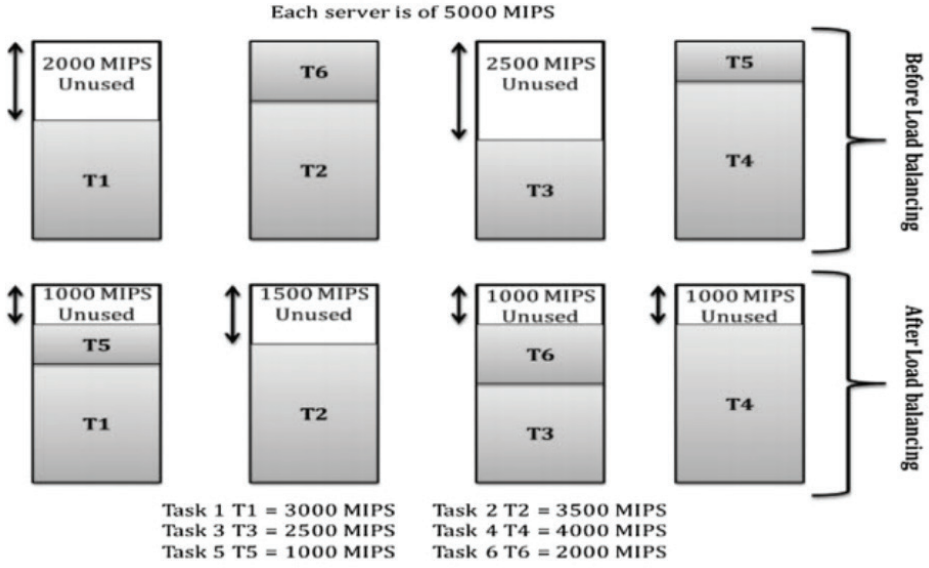

Figure 1. Load balancing of virtual machines (Mustafa et al., 2015)

Brazilian Journal of Operations & Production Management
Volume 16, Número 4, 2019, pp. 627-637
DOI: 10.14488/BJOPM.2019.v16.n4.a8

Load balancing is a new technique that provides high resource time and effective resource efficiency by allocating total load to different cloud nodes. Load balancing solves the overload problem and focuses on maximizing operational power, optimizing resource use, and minimizing response time. Load balancing is the prerequisite for maximum cloud performance and effective use of resources (Panwar and Mallick, 2015). By load balancing, one can balance the load by dynamically transferring the local task amount from a machine to another machine in a remote node or a less-used machine. This maximizes user satisfaction, minimizes the response time, increases resource utilization, reduces the number of task rejections (the tasks that are given back) and raises system performance (Khetan et al., 2013). In recent years, many researchers have come up with various ideas for solving the resource management problem through load balancing. In most load balancing methods, the migration approach between servers is used. Tasks have been migrated between virtual machines to balance load on servers, as shown in Figure 1 (Mustafa et al., 2015).

This figure shows, in the top section, the status of virtual machines before load balancing and, in the bottom section, the status of virtual machines after load balancing. As can be seen in the figure, before the load balancing, the two machines are fully loaded and have reached the over load status, while more than half of the capacity of the other two virtual machines is empty. In the second section of the figure, after the load balancing, all the virtual machines are almost in the same loading situation. In the load balancing at the level of virtual machines, the task load on the virtual machines is distributed. At this level, a task mapping to virtual machines is created. Load balancing at this level determines which task is allocated to which virtual machine.

Load Balancing Algorithms

In general, the load balancing algorithms design is performed taking into account the two main goals of providing and increasing the use of cloud resources. Scheduling algorithms for virtual machines require load balancing to effectively allocate virtual machines. In fact, load balancing algorithms decide which virtual machine will be allocated based on the cloud user request. So far, a large number of load balancing algorithms have been proposed; three popular algorithms used in the proposed approach of this paper are evaluated as follows:

1. Round Robin Algorithm

A round robin algorithm uses a simple technique to distribute all processes over all available processors. In this algorithm, the same task load is distributed on the processors. The algorithm also operates on the basis of random selection of the virtual machines, and the data center controller assigns requests to a list of virtual machines in a round way (Bhathiya, 2009).

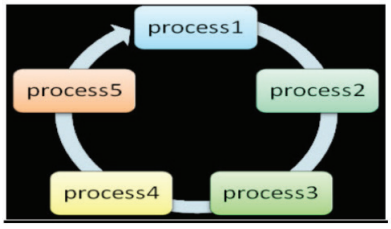

Figure 2. Round Robin Load Balancing (Pasha et al., 2014)

Throttled Algorithm

2. Throttled Algorithm

The equally spread current execution algorithm goes through some steps, taking into account the priorities (Bhathiya, 2009). The distribution of load equally with the load transfer from overloaded servers to light-loaded servers improves performance (Hu et al., 2010). In this algorithm, the load balancer consistently monitors the task queue for new activities and then assigns these tasks free virtual machines from the resource pool. The load balancer also uses the list of tasks allocated to virtual machines to help detect free machines and assign them to new tasks.

3. Equally Spread Current Execution Algorithm

The equally spread current execution algorithm goes through some steps, taking into account the priorities (Bhathiya, 2009). In this algorithm, the load balancer consistently monitors the task queue for new tasks and then assigns these tasks free virtual machines from the resource pool. The load balancer also uses the list of tasks allocated to virtual machines to help detect free machines and assign them to new tasks (Domanal and Reddy, 2013). The equally spread current execution algorithm is the same (optimized) Active Monitoring Load Balancer algorithm. The load distribution process is shown in Figure 4.

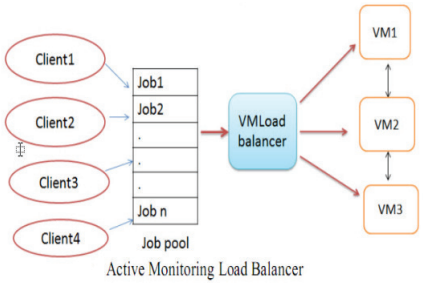

Figure 3. Active Monitoring Load Balancer (Pasha et al., 2014)

Brazilian Journal of Operations & Production Management
Volume 16, Número 4, 2019, pp. 627-637
DOI: 10.14488/BJOPM.2019.v16.n4.a8

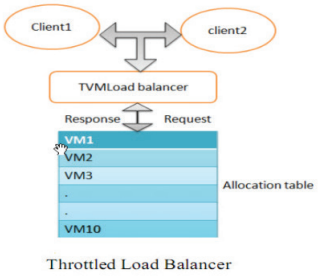

Figure 4. Load balancing (Pasha et al., 2014) Throttled

4. RESEARCH BACKGROUND

The method proposed by Hu et al. in 2010 has used the genetic algorithm for the load balancing between virtual machines, and it has also examined, in addition to the system current state, system changes and historical data. This method also calculates the effects of implementing virtual machines on host machines beforehand. Through this method, load balancing is achieved and the dynamic migration of virtual machines is reduced.

In 2013, Professor Soundarajan et al. have proposed a load balancing algorithm to optimize the use of resources in the cloud environment. The algorithm is a dynamic resource management method. In this algorithm, the goal is to efficiently distribute the load on accessible virtual machines that are not at the upper or lower limit. The simulation results show that this algorithm improves the use of resources and reduces response time.

Razali et al. presented a virtual machine classification according to their implementation time for load balancing. In this way, virtual machines migrated to two different classes of resources: high-power host and low-power host based on MIPS (Million Instructions per Second). Virtual machine migration is based on the CPU utilization in steady conditions. Using this method, the number of migrations is minimized and energy is saved in idle state (Razali et al., 2014).

Chen et al. (2017), in a study entitled "A novel load balancing architecture and algorithm for cloud services", a method for making dynamic balance was proposed for solving the problem of overload in cloud balancing. In this method, both server processing and computer loading are considered, and finally, the two algorithms to prove the proposed innovative approach were examined.

In 2018, Coutourier et al. investigated and introduced the best strategy for asynchronous iterative load balancing in cloud computing. The research purpose was to introduce a new strategy called the best attempt to balance the load of a node in all its loaded neighbors, while ensuring that all nodes involved in the load balancing step receive an equal amount. In the research using the SimGrid simulator, some of the bed test scenarios were considered and several QoS criteria were evaluated to demonstrate the utility of the proposed algorithm.

Rezaei et al. (2011) presented data center architecture for cloud computing that manages system resources to balancedly distribute load between data center resources and reduce power consumption. Failure to distribute the load balancedly can lead to reduction in terms of efficiency and vulnerability of the data center. Virtualization is a technology used at such centers and makes the live transmission of virtual machines possible. Moreover, in this research, an algorithm is presented that distributes the available load balancedly between different sources according to the productivity of the servers or hosts inside the data center. This system was evaluated based on the simulation and reallocation of virtual machines based on their productivity and the use of live transmission. The results show that the proposed algorithm causes load distribution and ensures SLA (Service level Agreement) properly (Rezaei et al., 2011).

In 2013, Mousavian Qalashqaei and Shiri optimized the load balancing on virtual machines at a rate of 20% by combining meta-heuristic methods. In this paper, a new method is proposed to find suitable solutions for mapping a set of requests to the available resources of the system, according to the conditions of cloud computing systems. In this method, they used the combination of the tabu search algorithm and the evolutionary algorithm mutation strategy (Mousavian Qalashqaei and Shiri, 2013).

Barani et al. (2015) performed load balancing to reduce virtual machine load. They provided an algorithm based on the processing power and the task load of virtual machines in cloud computing, comparing it against response time and Makespan with a number of other load balancing algorithms and, by performing the simulation, they found that the algorithm has an appropriate response time and makespan compared to previous algorithms. The makespan time is the time difference between the beginning and the end of a sequence of work or tasks in the system. This time is very important to measure the usefulness of the system. It is better to reduce this standard (Barani et al., 2015).

Chanaghlou and Dolati (2016), in a study entitled "Providing a Hybrid Multi-Objective Scheduling and Load Balancing in Cloud Computing", presented two algorithms for improving load balancing and task scheduling. The researchers concluded that the balance algorithm called Hypertext Markup Language (HMTL) has the ability to achieve load balancing goals and minimize overall runtime. It also uses the policy of reducing the number of task migrations. The scheduling algorithm, entitled LDTS (Linear Decision Trees), also assigns new tasks to system processing nodes by computing its cur-

Brazilian Journal of Operations & Production Management
Volume 16, Número 4, 2019, pp. 627-637
DOI: 10.14488/BJOPM.2019.v16.n4.a8

rent task load. Simulations have shown that the LDTS algorithm has improved load distribution. The HMTL algorithm has also improved parameters, such as moment load balancing, total load balancing, task load distribution, and overall runtime (Chanaghlou and Dolati, 2016).

In 2017, Derakhshanian et al. investigated the load balancing in cloud computing environment, taking into account the dependence between tasks and the use of adaptive genetic algorithm. Considering interactions between tasks, the purpose of this study was to provide a method for an optimal load balancing in the network, so that the total completion time and the idle time of the machines would be minimized. The experimental results showed that the localization of interactions would have a significant effect on reducing the total completion time (Derakhshanian et al., 2017).

In 2018, Mishkar et al. optimized task scheduling and load balancing in the cloud environment using the Ant Colony Algorithm. The purpose of this study was not merely to schedule tasks, but also to examine load balancing on machines. To do this, scheduling with the ant colony optimization algorithm was used, which provides effective solutions to many dynamic problems. In this research, the problem statement and the scheduling problem and related tasks were mentioned, and then definitions related to task scheduling and cloud environment were proposed, and then all the steps of the algorithm were followed, and, finally, load balancing was performed (Mishkar et al., 2018).

5. RESEARCH METHODOLOGY

The proposed algorithm is a hybrid algorithm, a combination of two techniques used in two other virtual algorithms, Throttled and ESCE. In the proposed algorithm, using the Throttled algorithm, the states of virtual machines are obtained. The ESCE algorithm is also used to monitor and assign tasks to virtual machines. Active load balancing algorithms always monitor the job queue so that they can assign them to free or idle machines. It also maintains a list of orders for allocation to any virtual machine. This list can determine the overloaded or low-loaded conditions in a time slice. Based on this information, the balancer is transmitted from overloaded machines to low-load machines so that the virtual machines reach a load balancing level.

The proposed algorithm is designed to improve response time and processing time. To achieve this goal, the virtual machine algorithm, with the least load, initially proposes reducing the search overload to find a machine that can do longer work and improve response time. In a data center, tasks and requests are received from user centers. The data center controller finds a virtual machine for each job that can do that. Figure 5 shows the conceptual model of the proposed algorithm. In the figure, the virtual machine (VM) hosts: cloud resources; Cloudlet are the same jobs; and DCC: Data Center Controller.

In each data center, there are a number of physical servers (Host), which include virtual machines (VMs). Jobs (Cloudlet) are received by the data center controller for execution and processing, and are allocated to virtual machines. In fact, users send their jobs to the data centers where the jobs are allocated to servers and to the virtual machines inside the servers.

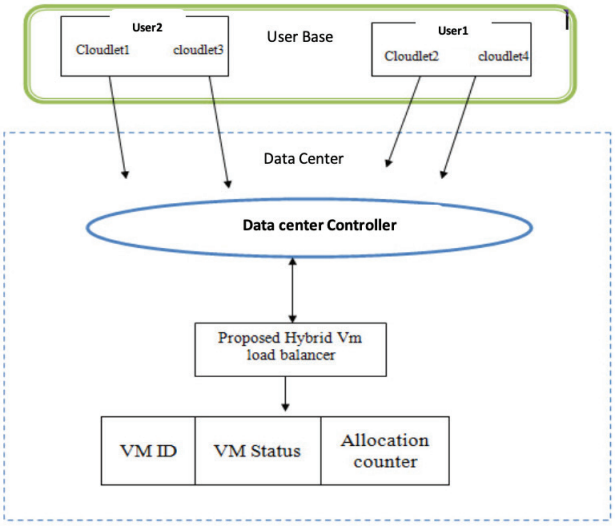

Figure 5. Conceptual model of the proposed algorithm

Introducing the steps of the proposed algorithm

- **Step 1:** The algorithm keeps a list of VMs, their status (occupied / free), and the tasks that are currently allocated to them.
- **Step 2:** The data center controller receives requests from cloud clients.
- **Step 3:** The data center controller asks the algorithm about available virtual machines.
- **Step 4:** The algorithm performs the following steps:

The next available virtual machine finds the status of virtual machines using the table. If the machine is idle, it goes to Step 5 and sends the machine ID to the controller. If the machine is not idle, it goes to the following stages.

- It examines whether the virtual machine capacity is greater than zero and the number of current allocations of that machine is lower than the number of machine allocations that is consid-

Brazilian Journal of Operations & Production Management
Volume 16, Número 4, 2019, pp. 627-637
DOI: 10.14488/BJOPM.2019.v16.n4.a8

ered among the VM list as the maximum number of allocation; it selects the VM as the virtual machine that has the minimum.

- Return of the VM ID of the virtual machine that has the minimum load.

- **Step 5:** The data center controller places the job on the VM sent from the algorithm.
- **Step 6:** If the virtual machine that has the minimum task load is overloaded.
- **Step 7:** The data center controller sends a reply to the job and transmits it to the pool of awaiting tasks.
- **Step 8:** The controller continues the job, restarting it from Step 2.

In the proposed algorithm, the information is first collected from the status quo, the VMs, their status (occupied / free), and the tasks that are currently allocated to them. The data center controller goes to the task list to allocate that job to a VM to do that. While the controller recognizes VMs by their IDs, it requests the algorithm to introduce a VM. The algorithm has a list of VMs and their status also includes the number of tasks that are being performed on each VM. For a VM, if the current allocation number is zero, it means that the machine is idle, whereas if the number is higher than zero, it means that the machine has not yet completed previous jobs. If the allocation number is lower than the maximum number of allocations, then this machine can do other jobs. Therefore, no machine will be idle and machines with the lowest allocation will be considered as the first option for allocating tasks. The ID algorithm sends the selected virtual machine to the data center controller and the controller checks if the selected machine can do the job, places the job on the VM, and announces the algorithm to update its table and add a task to the work being done by this machine. However, if this job cannot be done on this machine, since the machine has been selected with the minimum load, there is no other machine that can do it; thus, the data center controller returns the job to the pool to wait and receives another job from the user's requests list. This process continues until all jobs are done. This process is shown in Chart 1.

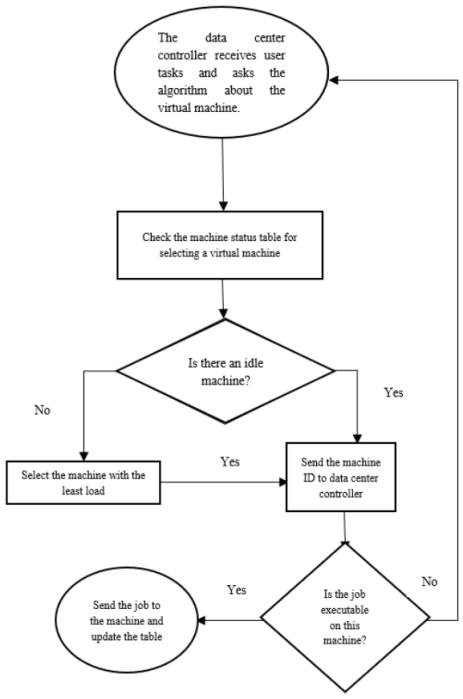

Chart 1. Flowchart of the proposed algorithm

Implementation

In the implementation, the CloudAnalyst simulator is used. This simulator is a CloudSim-based visual design and has been used in most of the studies on load balancing in cloud computing. The CloudAnalyst simulator easily covers any load balancing policy at the virtual machine level. The graphical user interface of this simulator can receive the settings in an interaction and, after implementing the load balancing policy, presents the results in the form of charts and tables. The code source for the proposed hybrid algorithm is added to the CloudAnalyst simulator code source set in Java via the NetBeans IDE 8.0 software and settings include configurations for the user base, configuration of program development, user grouping, data center settings, and the physical hardware of each data center, shown in Tables 1 to 5.

Table 1. User base settings

Name	Region	Requests per User Per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
UB1	0	60	1000	3	9	1000	100
UB2	1	60	1000	3	9	1000	100
UB3	2	60	1000	3	9	1000	100
UB4	3	60	1000	3	9	1000	100
UB5	4	60	1000	3	9	1000	100
UB6	5	60	1000	3	9	1000	100

Brazilian Journal of Operations & Production Management
Volume 16, Número 4, 2019, pp. 627-637
DOI: 10.14488/BJOPM.2019.v16.n4.a8

Table 2. Software deployment settings

Data Center	#VMs	Image Size	Memory	BW
DC1	10	10000	512	1000
DC2	10	10000	512	1000
DC3	10	10000	512	1000
DC4	10	10000	512	1000

Table 3. Data center

Name	Region	Arch	OS	VMM	Cost per VM $/Hr	Memory Cost $/s	Storage Cost $/s	Data Transfer Cost $/Gb	Physical Hw Units
DC1	0	X86	Linux	Xen	0.1	0.05	0.1	0.1	15
DC2	4	X86	Linux	Xen	0.1	0.05	0.1	0.1	15
DC3	2	X86	Linux	Xen	0.1	0.05	0.1	0.1	15
DC4	3	X86	Linux	Xen	0.1	0.05	0.1	0.1	15

Table 4. Details of physical hardware of each data center

Id	Memory (Mb)	Storage (Mb)	Available BW	Number of Processors	Processor Speed	VM Policy
1-15	204800	100000000	1000000	4	10000	TIME_SHARED

Table 5. Advanced settings

User grouping factor in User Bases	100
Request Grouping Factor in Data Centers	10
Executable instruction length per request (bytes)	100

6. RESULTS

The settings considered in the simulation model have been done on the algorithm. The purpose of this research is to reduce the response time and processing time in data centers. Therefore, the criteria for evaluation in the results include the response time and the processing time in the data centers. Subsequently, these criteria are evaluated on four algorithms, ESCE, Throttled, Round Robin and the proposed algorithm for this research (MyHybrid). The results on how to place user bases and data centers are shown in Figure 6.

Results in the response time criterion

The chart shows the response time to service in each data center in milliseconds for four algorithms. As shown in the chart, the average response time of the proposed algorithm is lower for each user base.

The average of this time is shown in Chart 2.

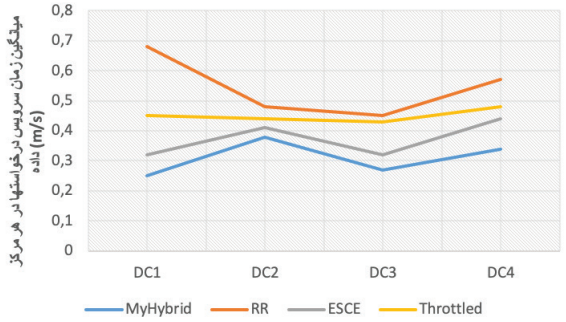

Chart 2. Average response time in milliseconds for each data center

Chart 3 shows the average response time in milliseconds for each user base for the four algorithms. As shown in the chart, the average response time of the proposed algorithm is lower in the user base.

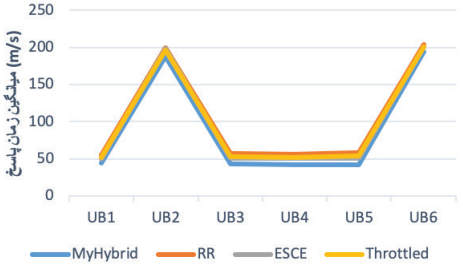

Chart 3. Average response time in milliseconds for each user base for the four algorithms

Brazilian Journal of Operations & Production Management
Volume 16, Número 4, 2019, pp. 627-637
DOI: 10.14488/BJOPM.2019.v16.n4.a8

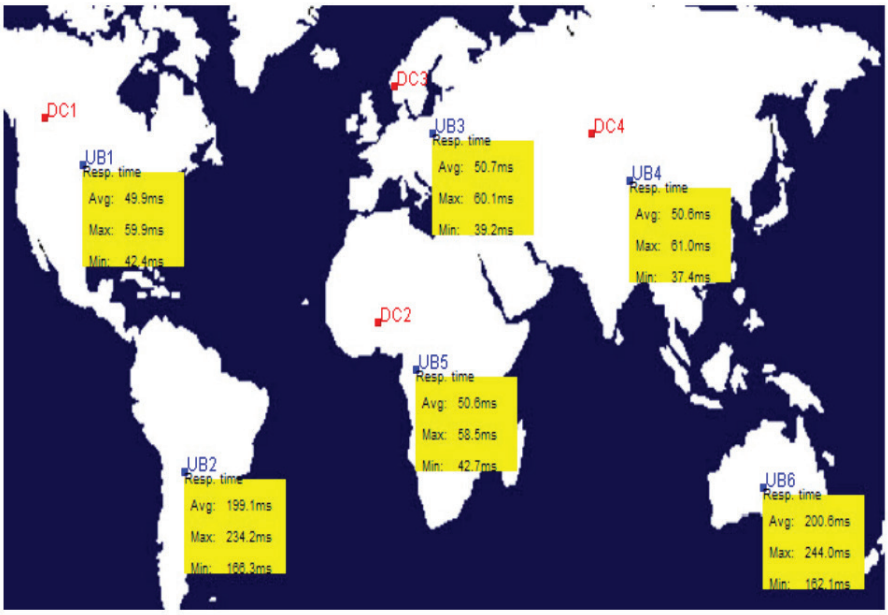

Figure 6. way of placing user bases and data centers

Table 6. Results in response time in milliseconds (ms)

MyHybridAlgo	Round Robin	Throttled	ESCE	User base
44.07	55.25	52.23	49.9	UB1
188.13	199.90	196.69	199.1	UB2
43.12	57.11	53.49	50.7	UB3
42.73	56.23	52.37	50.6	UB4
42.23	58.29	54.26	50.6	UB5
194.39	204.49	201.23	200.6	UB6

Table 7. Results in the overall data center processing time for the all algorithms in milliseconds (ms)

	ESCE	Throttled	RR	MyHybrid
Overall processing time	0.37	0.45	0.55	0.31

Results in the processing time criterion in data centers

The obtained results in the overall data center processing time for all algorithms are shown in Table 7.

The results obtained from the response time and processing time evaluation in data centers on the four algorithms, ESCE, Throttled, Round Robin and the proposed algorithm of this research (MyHybrid) show that the overall response time and data processing time in the data center for the proposed algorithm of the research is lower than the other algorithms compared.

7. CONCLUSION

This paper focuses on the task load balancing of the hosts and attempts to provide almost equal task loads for all hosts.

In this study, a combined load balancing algorithm was proposed from two existing ESCE and Throttled algorithms. The notice of the virtual machines' status and the number of assignments are the two main characteristics in the proposed hybrid algorithm. These two attributes, each of which is derived from an algorithm, are:

1. **Throttled algorithm:** It uses a table that holds the status indicators of virtual machines (free / occupied).

2. **ESCE algorithm:** It uses the task list assigned to virtual machines. This list specifies the number of virtual machines each assignment is assigned to.

The proposed algorithm is designed to improve response time and processing time. To this end, the algorithm recommends a virtual machine with the least load, at first, to reduce the search overhead to find a machine that can handle more length work and improve the response time.

Brazilian Journal of Operations & Production Management
Volume 16, Número 4, 2019, pp. 627-637
DOI: 10.14488/BJOPM.2019.v16.n4.a8

The proposed algorithm helps the data center controller to choose between the machines that can do it (machines available), a machine that is either idle or has the smallest load. This action reduces the processing and time overhead for looking for a virtual machine, especially for more work and improved processing time and response time.

In the implementation, after analyzing the CloudSim and CloudAnalyst simulators, the source code for the proposed hybrid algorithm was added to the Java language, and via the NetBeans IDE 8.0 software to the CloudAnalyst emulation source code set. Through the settings through the CloudAnalyst Home Page, the proposed algorithm was evaluated with three other algorithms: Round Robin, ESCE, and Throttled.

The results of the simulations performed according to the simulation model for the four algorithms show that the proposed algorithm has better response time and processing time than the other three algorithms.

The results of the overall response time for all algorithms show that the response time of the proposed algorithm is 12.28%, compared to the Round Robin algorithm, 9.1% compared to the Throttled algorithm, and 4.86% for the ESCE algorithm.

REFERENCES

Arianyan, E.; Taheri, H.; Sharifian, S. (2015), "Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data centers", Computers & Electrical Engineering, Vol. 47, pp. 222-240.

Barani, S.; Sattari, V.; Barani, F. (2015), "Load Balancing Based on Processing Power and Task Load of Virtual Machines in Cloud Computing", 7th International Conference on Information Technology and Knowledge, Urmia University, June 2015.

Barzegar, M.; Hori, A.; Dastghayebi Fard, G. H. (2014), "Presentation of methods based on Service Level Agreement for choosing a virtual machine in the dynamic combination of virtual machines", 20th Annual National Conference of Iran Computer Society, Ferdowsi University, pp. 448-453, March 2015.

Bhathiya, W. (2009), CloudAnalyst a Cloudsim-based tool for modelling and analysis of large scale cloud computing environments, MEDC project, *Cloud Computing and Distributed System Laboratory*, University of Melbourne, Australia, pp. 1-44.

Bohn, R. (2011), NIST Cloud Computing Reference Architecture & TaxonomyWorking Group, *Information Technology Laboratory*, June 21.

Chanaghlou, A.; Dolati, A. (2016), Providing a method for hybrid multi-objective scheduling and load balancing in cloud computing, Master's thesis, Kerman Graduate University of Advanced Technology.

Chen, S. L.; Chen, Y. Y.; Kuo, S.H. (2017), "CLB: A novel load balancing architecture and algorithm for cloud services", *Computers and Electrical Engineering*, Vol. *58*, pp. 154-160.

Cordeiro, G. A.; Ordónez, R. E. C.; Ferro, R. (2019), "Theoretical proposal of steps for the implementation of the Industry 4.0 concept", *Brazilian Journal of Operations & Production Management*, Vol. 16, No. 2, pp. 166-179.

Couturier , R.; Giersch, A.; Hakem, M. (2018), "Best effort strategy and virtual load for asynchronous iterative load balancing", *Journal of Computational Science*, Vol. 26, May 2018, pp. 118-127.

Derakhshanian, Y.; Mirabedini, S. J.; Haroon Abadi, A. (2017), "Load balancing in cloud computing environment, considering dependence between tasks and using adaptive genetic algorithm", Telecommunication Engineering Journal, Vol. 7, No. 24, Summer 2017, pp. 58-69.

Dhanasekar, P.; Senthilkumar, A.; Sairamprabhu, S. G. (2014), "Balancing the Virtual Machines Load by Dynamic Move for Cloud Environment", *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 2, No. 7.

Domanal, S. G. and Reddy, G. R. M. (2013), "Load Balancing in Cloud Computing Using Modified Throttled Algorithm", In Cloud Computing in Emerging Markets (CCEM), 2013 IEEE International Conference on (pp. 1-5). IEEE.

Hu, J.; Gu, J.; Sun, G. et al. (2010), "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", In *3rd IEEE International Symposium on Parallel Architectures, Algorithms and Programming*. DOI 10.1109/PAAP.2010.65.

Kaur, P. and Kaur, P. D. (2015), "Efficient and Enhanced Load Balancing Algorithms in Cloud Computing", *International Journal of Grid Distribution Computing*, Vol. 8, No. 2, pp. 9-14.

Khetan, A.; Bhushan, V.; Chand Gupta, S. (2013), "A Novel Survey on Load Balancing in Cloud Computing", *International Journal of Engineering Research & Technology (IJERT)*, Vol. 2, No. 2, pp. 1-5.

Mishkar, Y.; Seyed Bagheri, S. H.; Baghbani, A. et al. (2018), "Optimizing scheduling tasks and load balancing in the cloud environment using ant colony algorithm", In First National Conference on Research in Computer Engineering.

Mousavian Qalashqaei, N; Shiri, A, (2013), "Load balancing on virtual machines in cloud computing by combining meta--heuristic methods", In: National Conference on Computer Engineering to Information Technology Management.

Brazilian Journal of Operations & Production Management
Volume 16, Número 4, 2019, pp. 627-637
DOI: 10.14488/BJOPM.2019.v16.n4.a8

Mustafa, S.; Nazir, B.; Hayat, A. et al. (2015), "Resource management in cloud computing: Taxonomy, prospects, and challenges", *Computers and Electrical Engineering*, Vol. 47, pp. 1-344 (October 2015).

Padhy, R. P. and Rao, G. P. (2011), Load Balancing in Cloud Computing Systems, Bachelor thesis, Rourkela Orissa India, May 2011.

Panwar, R. and Mallick, B. (2015), "A Comparative Study of Load Balancing Algorithms in Cloud Computing", *International Journal of Computer Applications*, Vol. 117, No. 24, pp. 33-37, May 2015.

Pasha, N.; Agarwal, A.; Rastogi, R. (2014), "Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, No. 5, pp. 34-39.

Razali, R. A.; Rahman, R. A.; Zaini, N. et al. (2014), "Virtual machine migration implementation in load balancing for cloud computing", *5th International Conference on Intelligent and Advanced Systems (ICIAS)*, IEEE, pp. 1-4.

Rezaei, H.; Hajian, G. H.; Analooi, M. et al. (2011), Virtualized Resource Management for Load Balancing in the Cloud Computing Data Center, Masters Paper in Software Engineering, Iran University of Science and Technology, pp. 1-8.

Sahu, Y.; Pateriya, R. K.; Kumar, R. (2013), "Cloud server optimization with load balancing and green computing techniques using dynamic compare and balance algorithm", *5th International Conference on Computational Intelligence and Communication Networks*, IEEE-2013.

Soundarajan, D. and Subbiah, S. (2013), "Intelligent Redirection of Virtual Machine And Efficient Load Balancing In Private Cloud", *International Journal of Computer Networks and Wireless Communications (IJCNWC)*, pp. 441-444.

Received: 06 Jun 2019

Approved: 24 Jun 2019

DOI: 10.14488/BJOPM.2019.v16.n4.a8

How to cite: Sabeti, S. H.; Mollabgher, M. (2019), "Proposing a load balancing algorithm with an integrative approach to reduce response time and service process time in data centers", Brazilian Journal of Operations & Production Management, Vol. 16, No. 4, pp. 627-637, available from: https://bjopm.emnuvens.com.br/bjopm/article/view/864 (access year month day).