

STUDY THE EFFECT OF PARAMETERS TO LOAD BALANCING IN CLOUD COMPUTING

Tran Cong Hung and Nguyen Xuan Phi

Posts and Telecommunications Institute of Technology, Vietnam

ABSTRACT

The rapid growth of users on the cloud service and number of services to the user increases the load on the servers at cloud datacenter. This issue is becoming a challenge for the researchers. And requires used effectively a load balancing technique not only to balance the resources for servers but also reduce the negative impact to the end-user service. The current, load balancing techniques have solved the various problems such as: (i) load balancing after a server was overloaded; (ii) load balancing and load forecast for the allocation of resources; (iii) improving the parameters affecting to load balancing in cloud. The study of improving these parameters have great significance to improving system performance through load balancing. From there, we can propose more effective methods of load balancing, in order to increase system performance. Therefore, in this paper we researched some parameters affecting the performance load balancing on the cloud computing.

KEYWORDS

Load balancing; Virtual Mmachines;Lload Balancing Parameters;Cloud Computing.

1. INTRODUCTION

Cloud computing is a rapidly growing technology in today's Internet world. With the advent of cloud computing, a new age of the Internet has really started. Along with the use of cloud computing in excess of any applications or services running on the Internet so far. Although the number of users on the Internet is huge compared to those who used the devices on the cloud, but the growth rate of users in cloud computing is growing. The Internet giants like Google, Microsoft, Amazon, and many other small-scale businesses, have started using Cloud and gives the user base for their Cloud experience and serve other purposes such as: work, business, research, ... cloud computing allows users to use software applications without having to install them on their computers. End-users pay a subscription fee to use Cloud services. That means that businesses only pay a service fee to use the software without hardware infrastructure equipment, Storage, ... for your business purposes. All the rest is due to the cloud service provider is responsible. Businesses focus on their main objective is business. In cloud computing, software is stored directly from the server of the service provider end user access through the Internet environment, this has increased the efficiency of their business over and over again. These include a number of cloud services typically such as Gmail (Google), EC2 (Amazon), Azure (Microsoft), Bluemix Cloud platform (IBM), ... With so many amenities, the popularity, the user to use cloud computing will increase. The increase in the number of users is synonymous with the increasing number of requests for cloud service providers at any given time. The number of users can be considered as the load of the server. Therefore, inevitably will lead to processing power of the server will be overloaded. The solution to this problem may be to increase the number of servers, increase the processing power of the server, or in many cases is scheduled for requesting access to the servers. The increase in processing power of servers is costly and is a temporary solution and in many cases unnecessary. So the solution is to study primacy of algorithms for

resource balancing (load) on cloud computing for businesses will help service providers reduce the considerable cost hardware infrastructure equipment.

With that approach, we studied the parameters affect the performance load balancing on the cloud and it is the basis to propose methods for improving high performance load balancing on the cloud. This paper is organized as follows: Part I: introduction. Part II is the related work. Section III studies the affect of the parameters by using the tool CloudSim. Part IV simulation results and Section V concludes.

2. RELATED WORK

In works [1], the P. Srinivasa Rao et al. refers to the elements to take a balanced approach to effective, the information nodes of all other nodes. When a node receives a job, it has to query the status of the other buttons to find out which node has less usage to forward the work, and when all nodes are query phenomenon overload happens. With the nodes broadcast a statement informed about its status will also cause huge load on the network. Next is the issue of time wasted at each node to perform query status. Besides, the current state of the network is also a factor affecting the performance load balancing. Because, in a complex network, with multiple subnets, the network node configured to locate all the other node is a fairly complex task. Thus, querying the status of nodes in the cloud will affect the performance load balancing.

The Agraj Sharma et al. [2] showed that factors response time will greatly affect the performance load balancing on the cloud. The work [2] outlined two outstanding issues of the previous algorithm are: i) load balancing occurs only when the server is overloaded; ii) continuous information retrieval resources available lead to increased computational cost and bandwidth consumption. So the authors have proposed an algorithm based on the response time of the request to assign the required decisions for servers appropriately, the algorithm approach has reduced the query information on available resources, reduced communication and computation on each server. The result is simulated with ECLIPSE software using JAVA IDE 1.6 has proven the correctness of the proposed algorithm.

According to algorithm Min – Min [3] minimizes the time to complete the work in each network node however algorithm has yet to consider the workload of each resource. Therefore, the authors offer algorithm Load Balance Improved Min-Min (LBIMM) to overcome this weakness. Failure to consider the workload of each resource leading to a number of resources are overloaded, some resources are idle; therefore, the work done in each resource is a factor affecting the performance load balancing on the cloud. Tradition Algorithm Min - Min is simple and is the basis for the current scheduling algorithm in cloud computing. After giving the algorithm to consider the workload of each resource to overcome the limitations of tradition algorithm Min - Min is algorithm LBIMM [3] for better experimental results, shown in Table 1.

Table1 . Comparison of the results of the algorithm Min - Min and LBIMM [3]

Scheduling Algorithm	Min-Min	LBIMM
Number of Tasks	10	10
Number of Resources	5	5
Makespan (Sec)	19.625	12.5
Average Resource Utilization Ratio	45.29%	82.29%

In this experiment, the same value the Number of Task and Number of Resources but LBIMM given Makespan (execution time in each resource) than less value and Average Resource Utilization Ratio (percentage of resource use average) higher than the algorithm Min - Min. Again shows, the study workload in the cloud resources that have great influence on the balance of resources in the cloud. The parameters mentioned here are: Average Resource Utilization Ratio and Makespan.

The study aims to maximize throughput, reduce latency in cloud computing is also the author Dhinesh Babu L.D et al. [4] concerned. Because the problem of balancing resources on cloud computing is an optimization problem NP-hard class, load balancing for independent work is a very important priority in the scheduling algorithm on cloud. With the proposed algorithm HBB-LB with the basic parameters are: through put, the waiting time in the queue, authors proved the effect of these parameters on performance job scheduling avoid overloading on the cloud environment.

In the works [5] Author Gaochao Xu and his team believe in environmental load balancing cloud computing has a significant impact on the performance of the system. Good load balancing makes cloud computing more effectively and improve the satisfaction of users. This work [5] introduced a load balancing model for the Public Cloud based on the concept of Cloud Partitioning with a conversion mechanism to choose different strategies for different situations. The division cloud into partitions will contribute to building load-balancing strategy is effective.

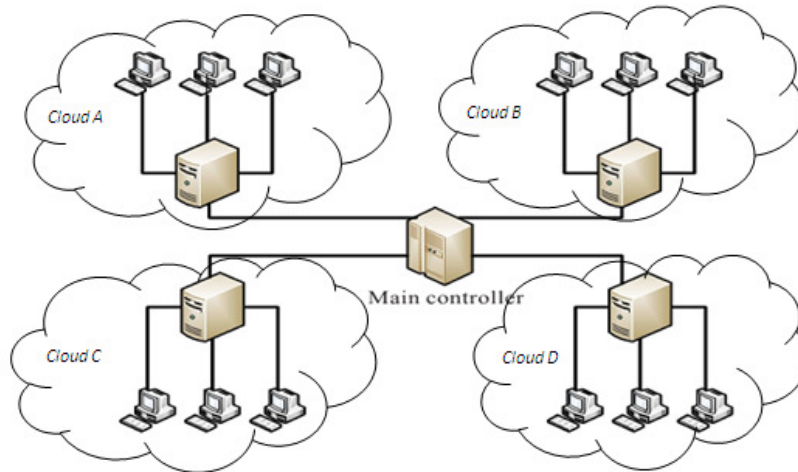


Figure 1. Relationship of the Cloud Partition [5]

Figure 1 shows the relationship of cloud partitions after having broken into separate partitions [5]. In each partition has its own set of load balancing and the balance is subject to the control of the Main Controller, select the partition decision cloud to serve, the information on the partition (state, load level, bandwidth, ...) management focus in the Main Controller. Now that we can see, in order to improve the efficiency of load balancing on the cloud, the division into separate partitions cloud will improve overall system performance servers.

Hiren H. Bhatt et al. [6] propose an algorithm enhanced load balancing using flexible load sharing algorithm (FLS). The author says, in the distributed environment, load balancing of the nodes and task scheduling is the main concern for the designers. In the Fig 2 show the nodes of domain in cloud scenario.

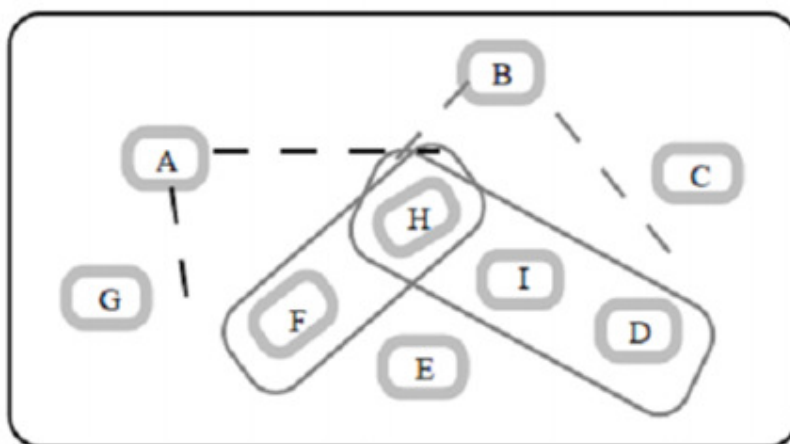


Fig. 2. Nodes of domain in cloud.

Domain of node A includes the node F and H, and B includes H, I, and D with FLS if H is the domain of A then the domain of node H will include node A. Then node will exchange the information from one node to another node. The third partition function will make the partition in to each domain so that using the remote location the information will be transferred to another domain. This research shows the basic technique for grouping the virtual machines. Divide the virtual machines to different groups and flexible load sharing will easily workload distributed for the virtual machines so that will reduce the overload VMs. The parameters in this study are: virtual grouping machines, load sharing.

Ritu Kapur [7] propose an algorithm LBRS (Load Balanced Resource Scheduling Algorithm). The idea of the algorithm is to consider the importance of resource scheduling policys and load balancing for resources in cloud. The main goal is to maximize the CPU Utilization, maximize the Throughput, minimize the Response Time, minimize the Waiting time, minimize the Turnaround Time, minimize the Resource Cost and obey the Fairness princip. In here, the paramaters mentioned are QoS: throughput, response time, waiting time,...

We found that the study of the parameters affecting to load balancing in cloud very practical sense, not only can increase performance processing by load balancing method the system but also the scientific basis to study make effective algorithms for improving performance of load balancing on the cloud computing. Some key parameters: throughput, response time, latency, execution time, ... Our main goal are methods to maximize throughput, reduce response time, minimum chemical timeout, reducing implementation time, ... to improve performance for data center cloud.

Table 1. Analysis of existing algorithms

Author's	Name of Method	Parameters	Findings
Srinivasa Rao et al [1]	Dynamic load balancing with a centralized monitoring capability	Node status, Processor, Job Requests	Minimizes the processing overhead on the processors and communication/network overhead on the network
Agraj Sharma et al [2]	Load Balancing by considering only the response time of the each request	Response Time, Threshold, Prediction Time	Considers the current responses and its variations to decide the allocation of new incoming requests, hence eliminates the need of collecting further data from any other source hence wasting the communication bandwidth
Huankai Chen et al [3]	Load Balance Improved Min-Min (LBIMM)	Makespan, Cloud Task Scheduling, Average Resource Utilization Ratio	Decreasing completion time of tasks, improving load balance of resources
Dhinesh Babu L.D et al. [4]	honey bee behavior inspired load balancing (HBB-LB)	Priorities of tasks, Honey bee behavior	Algorithm not only balances the load, but also takes into consideration the priorities of tasks that have been removed from heavily loaded Virtual Machines
Gaochao Xu et al [5]	load balance model for the public cloud based on the cloud partitioning concept with a switch mechanism to choose different strategies for different situations	Public cloud; Cloud partition	Division into separate partitions cloud will improve overall system performance servers
Hiren H. Bhatt et al. [6]	flexible load sharing algorithm (FLS).	Virtual grouping machines, Load sharing	Workload distributed for the virtual machines so that will reduce the overload VMs.
Ritu Kapur [7]	LBRS (Load Balanced Resource Scheduling Algorithm).	Throughput, Response time, Waiting time	To consider the importance of resource scheduling policies and load balancing for resources in cloud

3. CLOUDSIM: SIMULATION TOOL FOR CLOUD COMPUTING

Tools are the authors of CloudSim works [8] announced is very useful for simulating cloud computing environment. We used CloudSim Tool for simulation of our experiment.

3.1. THE BASIC COMPONENTS OF CLOUDSIM

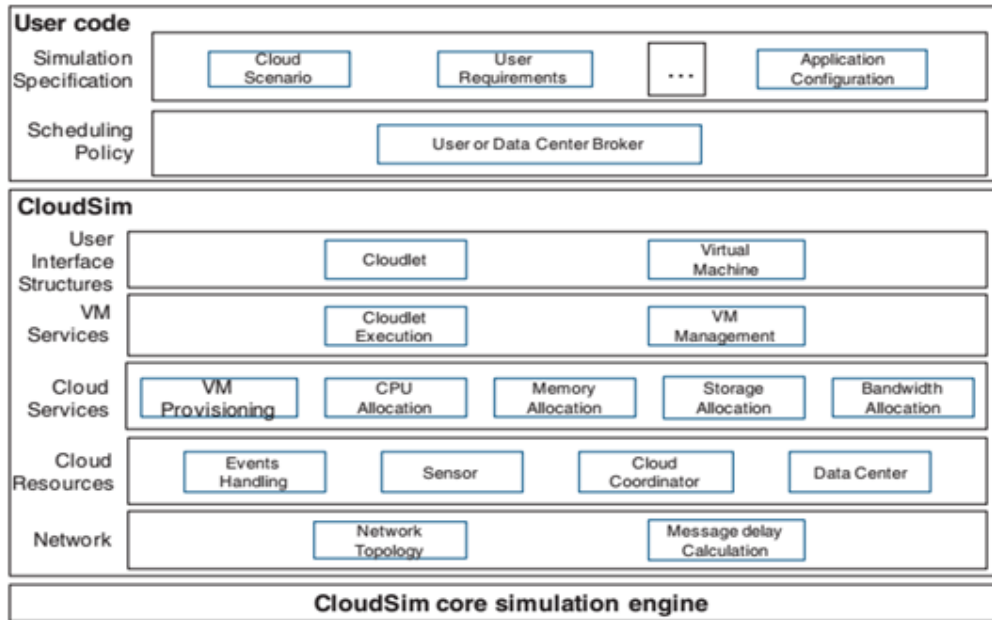


Figure 3. Layered CloudSim architecture [8].

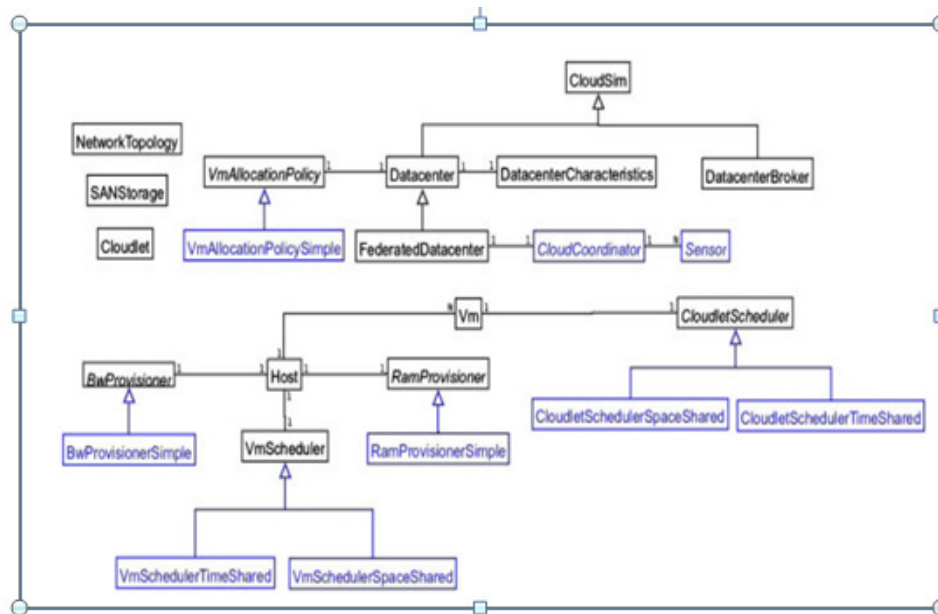


Figure 4. CloudSim class design diagram [8].

The function of the main class in CloudSim [8]:

- **Cloudlet:** Class of application services such as content delivery, social networking, business data, ...
- **Cloudlet Scheduler:** This abstract class is expanded by the implementation of different policies to determine the processing power of a VM Cloudlets. There are two policies: space-shared and time-shared.
- **Virtual Machines (VM):** The system handles the cloud computing work. It was created by the parameter before being handled work for the Datacenter. These parameters are: image size, virtual memory, MIPS, bandwidth, number of CPUs.
- **Vm Allocation Policy:** This class represents the allocation of virtual machines and servers. The main function is to select the server in the data center ready yet for memory, storage for virtual machines deployed on them.
- **VmScheduler:** this is an abstract class that is performed by space-shared Host, time-shared requirement to allocate processing cores to VMs
- **Datacenter:** This model core services layer infrastructure (hardware) provided by Cloud providers (eg Amazon, Azure, App Engine, ...), or in other words this is the layer that contains virtual servers to host different applications.
- **DatacenterBroker or Cloud Broker:** This layer acts as a broker, perform the negotiation between SaaS and Cloud Providers negotiations is controlled by QoS. Broker represent the SaaS Provider. It discovers the appropriate Cloud Providers by querying CIS layer (Cloud Information Service) and online negotiating commitments to allocate resources / services for the application QoS. The research and development system must extend this class to evaluate and test various brokerage policy. Difference between Broker and CloudCoordinator be understood as follows: Broker representing clients (ie decision components to increase efficiency), while CloudCoordinator behalf of the data center (ie trying to maximize the overall performance of the data center, without considering the specific needs of customers).
- **BwProvisioner:** The main role of this part is to implement the allocation of network bandwidth for a set of virtual machines compete on the data center. The cloud system development, and researchers can extend this class with its policy of prioritizing (QoS) to reflect the needs of the application. BwProvisioningSimple also allows a virtual machine can take up a lot of bandwidth on demand and use; however, it will be limited by the total bandwidth of the server.
- **CloudCoordinator:** This layer is responsible to periodically check the status of resources in the data center and on which to undertake decisive load. These components include special sensors and monitoring set up to handle the load. Monitoring data center resources is done by service updateDatacenter () by sending queries to the sensor. Services / Resource Discovery made by setDatacenter (), this method can be extended to implement the protocols and different mechanisms (multicast, broadcast, peer-to-peer). Besides, this component can also be extended to simulate cloud services such as Amazon EC2 Load Balancer-. The developers aim to extend this class to implement backup policies associated clouds (inter-cloud provisioning).
- **DatacenterCharacteristics:** This layer contains the configuration information of the resources in the data center
- **Host:** class physical resources (computer / host) include the following information: the amount of memory, the list of types of cores (for representing multi-core computers), shared policies the strength of the VMs, anticipated policy (provisioning) and memory bandwidth of VMs.
- **NetworkTopology:** This class reads the file to create a network topology Brite need simulation. This information is used to simulate the latency in CloudSim network.

- **RamProvisioner:** this is the class that represents the memory allocation policy the main (RAM) to the virtual machine. This class approval idle memory for the VM Host
- **SanStorage:** This class represents the network of Data Center storage (eg Amazon S3, Azure blob storage). SanStorage simulation can be used to store and extract information through the web interface without depending of time and space.
- **Sensor:** This interface is CloudCoordinator used to monitor specific parameters (energy consumption, use of resources, ...). This interface is defined by the method of: i) set the minimum and maximum thresholds for performance parameters; ii) periodically update the measurements. Therefore, this class is used to model the actual service by cloud vendors such as Amazon's CloudWatch, Microsoft Azure's FabricController. Each data center can create one or more sensors, each sensor is responsible for monitoring the various performance parameters.

3.2. NETWORK TRAFFIC FLOW IN SIMULATION

Modeling network topology to perform simulated cloud entities (host, storage capacity, users) has significant implications because the latency of the network architecture directly affect the experience customer services.

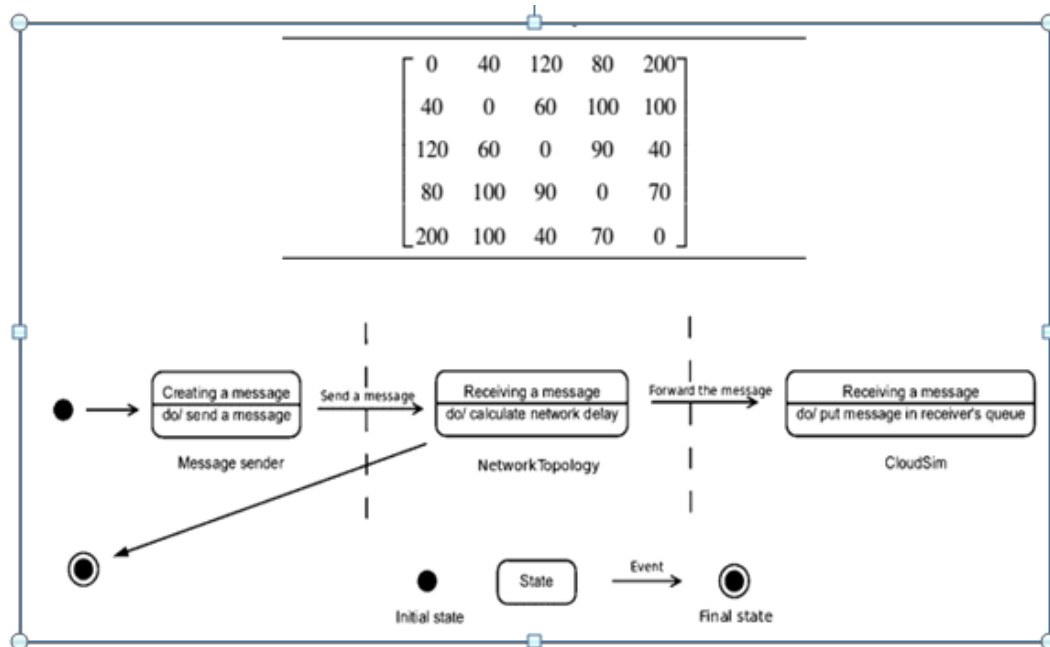


Figure 5. Network communication flow [8]

Looking at Figure 5 we can see, there is delay matrix in CloudSim. From entity i to entity j take some time $t + d$ with t is the time the message was sent, and d is the latency. The simulation using matrix latency easier when performing simulations of routers, switches and this matrix can be replaced by the weight or bandwidth between entities in data center. CloudSim taken from file Brite network topology include the following information: hosts, data centers, Cloud Broker, ...

4. SIMULATION RESULTS

In this scenario, we simulate a Datacenter only a physical host and create 10 virtual machines (VM) on the host. The aim is recorded and compared the time taken by a Cloudlet (considered Job) on the virtual machine when changing the parameter simulation, such as the magnitude of cloudlet, bw, ram, mips, ... of the virtual machine. Scenario simulation is done on CloudSim, here the Cloudlet can be considered as the work on job, depending on the magnitude of it that the virtual machine will handle fast or slow. Correspondingly, the Cloudlet in Figure 4 is the input of the system and we made a total measurement time (Time Execution) since CloudSim Cloudlet on until the end. Time Execution Unit of measure is milliseconds. Parameter mips represents the number of million instruction per second. The parameters of the VM and the Cloudlet used to simulate as follows (Table 3):

Table 3. Parameters of VMs and Cloudlet

VM description	
<i>mips = 20</i>	Million instructions per second
<i>size = 5120</i>	Image size (MB)
<i>ram = 128</i>	Vm memory (MB)
<i>bw= 1000</i>	Bandwith
<i>pesNumber = 1</i>	Number of cpus
<i>String vmm = "Windows"</i>	VMM name
Cloudlet properties	
<i>pesNumber=1</i>	Number of cpus
<i>length = 40000</i>	Length of this Cloudlet (size)
<i>fileSize = 300</i>	Input file size of this Cloudlet BEFORE submitting to a CloudResource
<i>outputSize = 300</i>	Output size of this Cloudlet AFTER submitting and executing to a CloudResource

4.1 THE MIPS PARAMETER AFFECTED

With 10 virtual machines created with the same parameter we obtained the results as shown in Table 4.

Table 4. Results of simulation when the parameter are equal

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
1	SUCCESS	2	1	2000	0.1	2000.1
2	SUCCESS	2	2	2000	0.1	2000.1
3	SUCCESS	2	3	2000	0.1	2000.1
4	SUCCESS	2	4	2000	0.1	2000.1
5	SUCCESS	2	5	2000	0.1	2000.1
6	SUCCESS	2	6	2000	0.1	2000.1
7	SUCCESS	2	7	2000	0.1	2000.1
8	SUCCESS	2	8	2000	0.1	2000.1
9	SUCCESS	2	9	2000	0.1	2000.1
10	SUCCESS	2	10	2000	0.1	2000.1

From Table 4 we see when the same input parameters, we obtained Time is equal. Now, we change the values of parameters mips (as Table 5), the results obtained as Table 6 and Figure 6.

Table 5. Change the parameter mips

VM 1	VM 2	VM 3	VM 4	VM 5	VM 6	VM 7	VM 8	VM 9	VM 10
mips*2	mips	mips*4	mips*2.5	mips*3	mips*5	mips*6	mips*3.5	mips*7	mips*4.5

Table 6. Simulation results when changed parameters mips

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
9	SUCCESS	2	9	285.71	0.1	285.81
7	SUCCESS	2	7	333.33	0.1	333.43
6	SUCCESS	2	6	399.99	0.1	400.09
10	SUCCESS	2	10	444.44	0.1	444.54
3	SUCCESS	2	3	500	0.1	500.1
8	SUCCESS	2	8	571.43	0.1	571.53
5	SUCCESS	2	5	666.66	0.1	666.76
4	SUCCESS	2	4	800	0.1	800.1
1	SUCCESS	2	1	1000	0.1	1000.1
2	SUCCESS	2	2	2000	0.1	2000.1

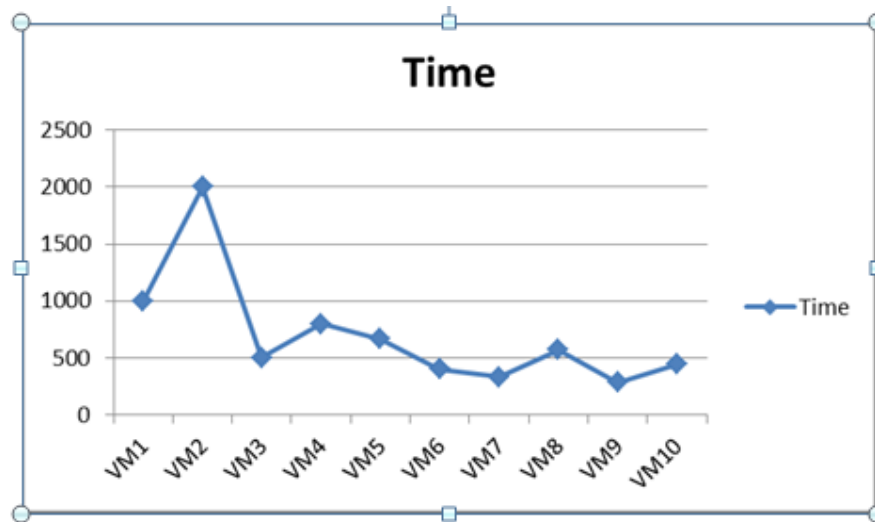


Figure 6. Execution time of VM when changed *mips*

From Figure 6 we see, VM9 (with $mips = mips * 9$) has highest $mips$ should be the first choice to Cloud carried out the work, therefore the smallest Time Execution, next the VMs has $mips$ descending is selected, VM2 should have the smallest $mips$ then Time Execution is the greatest. Figure 6 shows the correlation between $mips$ and Time Execution. The degree of influence of $mips$ parameter clearly affect the execution time of the virtual machines in the cloud. Therefore, the research to increase value of $mips$ will avoid the traffic deadlock or otherwise improve performance of load balancing in the cloud.

4.2 THE LENGTH PARAMETER AFFECTED

In CloudSim, $length$ parameter represents the size of the Cloudlet (job). In here, we will simulate split $length$ value for virtual machines, which simulate the tasks to be subdivided and assigned to virtual machines run in the Cloud.

Table 6. Length value for VMs

VM 1	$length/400$	VM 6	$length/600$
VM 2	$length/300$	VM 7	$length/800$
VM 3	$length/700$	VM 8	$length/200$
VM 4	$length/500$	VM 9	$length/100$
VM 5	$length/100$	VM 10	$length/50$

Since the parameters in Table 6 (the other parameters is unchange), we obtained the results as shown in Table 7 and Figure 7.

Table 7. Results of simulation parameters when changed length

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
7	SUCCESS	2	7	2.5	0.1	2.51
3	SUCCESS	2	3	2.85	0.1	2.95
6	SUCCESS	2	6	3.3	0.1	3.4
4	SUCCESS	2	4	4	0.1	4.1
1	SUCCESS	2	1	5	0.1	5.1
2	SUCCESS	2	2	6.65	0.1	6.75
8	SUCCESS	2	8	10	0.1	10.1
5	SUCCESS	2	5	20	0.1	20.1
9	SUCCESS	2	9	20	0.1	20.1
10	SUCCESS	2	10	40	0.1	40.1

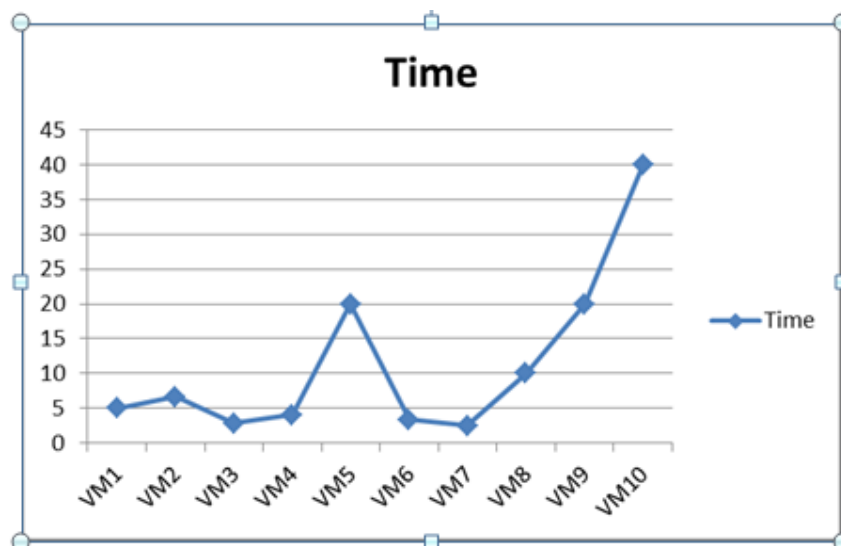
Figure 7. Execution time when *length* changed of VMs

Figure 7 show that, the virtual machine has length value greater (length = 40000, from Table 3) then the time to execute the greater. The length value represents the size of the job come to cloud. Here we see that, VM5, VM9 and VM10 corresponding length value is $\text{length}/100 = 400$, $\text{length}/100 = 400$, $\text{length}/50 = 800$ (with length = 40000, from Table 6), so corresponding time of execution is 20ms, 20ms and 40ms (Figure 7). The VM6 and VM7 have length value smaller length value of VM5, so corresponding time of execution is 3.3ms and 2.5ms (smaller execution time of VM5 is 20ms). Reason for time of executed on VM5, VM9 and VM10 larger than the other VMs is because it has a larger length value. Besides, VM7 has minimum length value (in Table 6) so there will be minimal execution time and vice versa VM10 greatest length value should have performed well greatest time. Now that we can see, the study of algorithms to be able to split the job before being assigned to the virtual machine VM is extremely important. This work has reduced the bottleneck phenomenon and increased the processing speed of the system, or in other words increase the performance for load balancing, balance of resources.

5. CONCLUSION

In the future, cloud computing is a promising technology, is a model for providing the resources needed for customer service oriented. Platform virtualization is one of the essential characteristics of cloud computing, virtualization refers to factors affecting business performance, such as information technology resources, hardware, software, operating system and service. Cloud computing is characterized by distributed computing and this is also the risk of bottlenecks that may occur during the allocation of resources for load balancing. The preventable risk overloading in cloud computing will make resources highly available status. The system access requests can occur simultaneously and will not avoid the "competitive" idle resources of the cloud. We have simulation and analysis of data on the impact of the parameters on cloud to performance of load balancing. From there, we discovered that, parameters makespan (runtime) is of great significance for the data center cloud. So the task of the researchers is to study the algorithms are effective load balancing to reduce the time makespan of virtual machines. Research suggest adjusting the parameters in order to improve performance of load balancing is also a research aimed at solving the imbalance on the cloud computing environment.

REFERENCES

- [1] P. Srinivasa Rao, V.P.C Rao and A.Govardhan, “Dynamic Load Balancing With Central Monitoring of Distributed Job Processing System”, International Journal of Computer Applications (0975 – 8887) ,Volume 65– No.21, March 2013.
- [2] Agraj Sharma, Sateesh K. Peddoju, “Response Time Based Load Balancing in Cloud Computing”, 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT).
- [3] Huankai Chen, Frank Wang, Na Helian, Gbola Akanmu, “User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing”, Parallel Computing Technologies, 2013 National Conference.
- [4] Dhinesh Babu ,Venkata Krishna P, “Honey bee behavior inspired load balancing of tasks in cloud computing environments”, Elsevier- Journal of Applied Soft Computing, no-13, 2013, pp-2292-2303
- [5] Gaochao Xu, Junjie Pang and Xiaodong Fu, “A Load Balancing Model Based on Cloud Partitioning for the Public Cloud”, Tsinghua Science and Technology , ISSN 1007-0214 04/12, pp 34-39, Volume 18, Number I, Febuary 2013.
- [6] Hiren H. Bhatt and Hitesh A. Bheda, “Enhance Load Balancing using Flexible Load Sharing in Cloud Computing”, 2015 1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India, 4-5 September 2015.
- [7] Ritu Kapur, “A Workload Balanced Approach for Resource Scheduling in Cloud Computing”, 2015 Eighth International Conference on Contemporary Computing (IC3), 20-22 Aug. 2015
- [8] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose and Rajkumar Buyya, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms” , Softw. Pract. Exper. 2011; 41:23–50, Published online 24 August 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/spe.995

AUTHORS

Tran Cong Hung was born in Vietnam in 1961. He received the B.E in electronic and Telecommunication engineering with first class honors from HOCHIMINH University of technology in Vietnam, 1987. He received the B.E in informatics and computer engineering from HOCHIMINH University of technology in Vietnam, 1995. He received the master of engineering degree in telecommunications engineering course from postgraduate department Hanoi University of technology in Vietnam, 1998. He received Ph.D at Hanoi University of technology in Vietnam, 2004. His main research areas are B – ISDN performance parameters and measuring methods, QoS in high speed networks, MPLS. He is, currently, Associate Professor PhD. of Faculty of Information Technology II, Posts and Telecoms Institute of Technology in HOCHIMINH, Vietnam.



Nguyen Xuan Phi was born in Vietnam in 1980. He received Master in Posts & Telecommunications Institute of Technology in Ho Chi Minh, Vietnam, 2012, major in Networking and Data Transmission. Currently, he is a PhD candidate in Information System from Post & Telecommunications Institute of Technology, Vietnam. He is working at the Information Technology Center of AGRIBANK in Ho Chi Minh, Vietnam.

