A Matchmaking Strategy Of Mixed Resource On Cloud Computing Environment

Wisam Elshareef, Hesham A. Ali, Amira Y. Haikal

Abstract: Today, cloud computing has become a key technology for online allotment of computing resources and online storage of user data in a lower cost, where computing resources are available all the time, over the Internet with pay per use concept. Recently, there is a growing need for resource management strategies in a cloud computing environment that encompass both end-users satisfaction and a high job submission throughput with appropriate scheduling. One of the major and essential issues in resource management is related to allocate incoming tasks to suitable virtual machine (matchmaking). The main objective of this paper is to propose a matchmaking strategy between the incoming requests and various resources in the cloud environment to satisfy the requirements of users and to load balance the workload on resources. Load Balancing is an important aspect of resource management in a cloud computing environment. So, this paper proposes a dynamic weight active monitor (DWAM) load balance algorithm, which allocates on the fly the incoming requests to the all available virtual machines in an efficient manner, in order to achieve better performance parameters such as response time, processing time and resource utilization. The feasibility of the proposed algorithm is analyzed using Cloudsim simulator, which proves the superiority of the proposed DWAM algorithm over its counterparts in literature. Simulation results demonstrate that proposed algorithm dramatically improves response time, data processing time and more utilized of resource compared Active monitor and VM-assign algorithms.

Index Terms: Cloud Computing; Resource management; Matchmaking; Load balance

1 Introduction

Cloud Computing is an emerging trend in IT environment. Cloud computing, as figure 1 depicts, is a style of computing that involves on-demand access to a shared pool of computing resources such as (network, servers, storage, applications and services), delivering hosted services over the Internet and storing data online, there are so many complex calculations and concepts implemented to achieve better and better use of resources and performance.

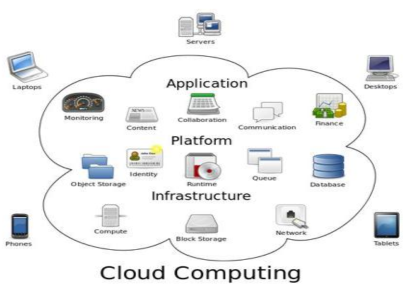

Fig 1: *cloud computing [1]*

Cloud computing allows each user to use the software and computing services on demand at any time, in any place and anywhere through the Internet. Cloud computing mainly deals with computing, software, data access and storage services that may not require knowledge of the end-user's geographical location and system configuration, which is to provide services [2]. Clouds exhibit varying demands, system sizes, supply patterns and resources (hardware, software, and network); users have heterogeneous, dynamic, and Quality of Service (QoS) requirements; and applications have varying performance, workload, and dynamic application scaling requirements [3]. The main objective of cloud computing is to provide easy, scalable way to computing resources and IT services. Fast development of cloud computing appears through many organizations such as GoGrid, Google, Rack space, Microsoft, Amazon EC2 cloud computing and Apple to provide cloud services to various consumers. The cloud system dynamically allocates computing resources for the customer/ user in response to customers' resource reservation requests and in accordance with customers' QoS requirements [4]. The characteristics of cloud computing are multi-tenancy, rented services delivery model, on-demand usage, external data storage, transparent, rapid elasticity, a broad network access, resource pooling and measured service[5, 6]. An important part of the cloud is the resource management. The resource management strategy in cloud should effectively utilize the pool of resource and achieve a high system performance. Resource management can be achieved through some sort of load balancing among the participating nodes. On that point are some metrics that will serve to measure the efficiency of each load balancing techniques (LBT). LBT in a cloud environment; consider various parameters [7, 8] such as response time, throughput, scalability, reliability, QoS, resource utilization and fault tolerance. Resource management affects three basic criteria for system evaluation, They are the performance, functionality and cost. Inefficient management of resources has a direct negative effect on performance and cost. You can also indirectly affect system functionality [9]. Matchmaking and scheduling are important issues performed by resource managers in the cloud [10]. Resource allocation in a cloud environment involves two phases, matchmaking is the first phase and scheduling is the second phase. Matchmaking is defined as, the method of allocating jobs associated with user requests to resources designated from the obtainable resource pool..Load balance means distribute load of multiple resources to achieve maximum throughput, minimize the response time and to avoid the overloading at a certain node. Both matchmaking and scheduling need to satisfy users' QoS requirements defined in a service level agreement (SLA).

1.1 Motivation

Cloud computing is a computing paradigm that can provide on demand, dynamic and scalable virtual resources through the Internet service to users. These resources are considered the backbone of the cloud. Recently, there are several techniques for the management of these resources, including matchmaking which means map incoming request to different resource, and thus exposed to the problems of satisfying the requirements of users, better utilization of different types of resources and dividing the load equally to maximize the throughput and minimize response time. Load balancing is required to distribute the dynamic local workload evenly across all the nodes. It helps to achieve user satisfaction and maximize the resource utilization ratio by ensuring an efficient and equitably for all computing resources allocation.. The important issue here is how to achieve load balance on VM and maintain QoS so that satisfying the requirements of users. Although literature is plentiful with huge number of researches that provide numerous load balancing strategies, there is still a crucial need for an efficient load balancing strategy to satisfy user requirements and simultaneously maximizing resource utilization. In this paper, we will propose a new technique for matchmaking between arrival request and heterogeneous resources in cloud environment. The innovation of the proposed technique is to distribute load between the heterogeneous resources, and on the other hand to satisfy the requirements of end-users. Without load balancing, users could experience delays, timeouts and possible long system responses, and our goal is to improve performance metrics such as response time, processing time, resource utilization and avoid overload. The rest of this paper is arranged as follows. Section 2 explores cloud computing challenges and emphasizes on resource management challenges. Section 3 related work is discussed. Section 4 the proposed technique. Section 5 experimental result. Finally, we present some concluding remarks.

2. Cloud Computing Challenges

This section lists the key research issues and articulate future research directions that have not boon fully addressed related to cloud computing arena. Some of the research challenges in cloud computing can be categorized as shown in Figure 2.

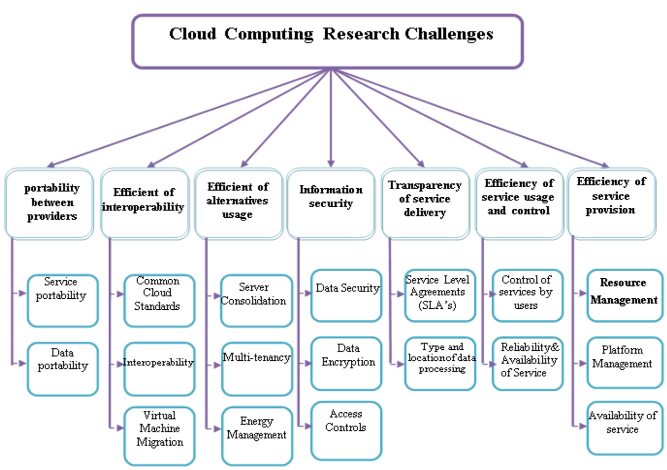

***Fig 2.**, Taxonomy of challenges in cloud computing*

Resource management is considered one of the Vertis research area in the cloud environment. The previous efforts in cloud computing resource management can be summarized as presented in table 1 with the advantages and disadvantages of each approach.

Table 1

Cloud Computing Resource Management Previous

	Resource management strategy	Pros	Cons
1	Linear Scheduling Strategy [11]	Improved throughput Response time Improved resource utilization	Not suitable for interactive real time application
2	PRE-Copy Approach for Scheduling [12]	Page level protection hardware	Long forwarding chains Delayed user experiences
3	Matchmaking and Scheduling [10]	Less the delay Economic	The uncertainties associated with such type of “matchmaking”. Unequal distribution of the load on the various resources. Lack of knowledge regarding local resource management policies.
4	Just-In-Time Resource allocation [13]	Cost effective	Prediction error Use of recursive data structures
5	MiyakoDori [14]	Memory reuse Shorter migration time	Efficient only in cases where migration back to the same system
6	A Two Tired On- Demand Resource Allocation Mechanism for VMBased Data Centers[15]	It addresses the problems of availability and scalability. If a failure of overall resource allocation occurs, then the local resource assignment will work conversely.So no failure of resource allocation is occurred	Application workload scheduling is not considered. Mismatch between the on demand resource and workload dispatch.

Given the table 1 there some problem in matchmaking that is unequal distribution of the load on the various resources, strive to overcome them to submit our proposal. In literature, there are many existing load balancing techniques that mainly focus on reducing associated overhead, service response time and improving performance. Table 2 lists some of these techniques in conjunction with the advantages and disadvantages of each technique.

TABLE 2:
LOAD BALANCING TECHNIQUE OF CLOUD COMPUTING

	Load Balancing Technique	Pros	Cons
1	Round Robin[16] [17] The first request is allocated to a randomly picked VM. Subsequent requests They are assigned Circular order.	Equality distribution workloads for all the nodes	Job processing time is not considered. Decrease Resource Utilization Does not save the state of the previous allocation of a VM to a request
2	Round Robin with Server Affinity: A VM Load Balancing [18] save the state of the previous allocation of a VM to a request and VM state (available/busy)	Improved Response time Improved Data center processing time Compared with Round robin	It is not clear for the best use of the resources and utilize them.
3	Active VM Load Balancer [20] Maintained number of requests currently allocated to each VM. Request is allocated to the least loaded VM.	Request is allocated to the least loaded VM.	Processing power and capacity hardware of VM is not considered.
4	Weighted Active Monitoring LB Algorithm [16] An assigned weight to VMs. Task is assigned to the least loaded and the highest weight VM.	Considers both the load Weight of available VMs. Increase response time and processing time	It did not take into account resource utilization.
5	ESCE Algorithm [19] If there is an overloaded VM then distributes some of the tasks to some idle VM	Improver response time and data processes time	Not fault tolerant because of single point of failure.
6	Throttled load balancer [18] Maintained the state of each VM. Request is accepted if found in the table otherwise the request is queued	tries to distribute the load evenly among the VMs. Response time improved	other parameters are not taken into account such as: weight of VM, processing time, etc

Given some of the problems of previous strategies as in the table 2, we strive to overcome some of the problems by submitting a proposal, which improves the performance metric. Problems such as processing power of VM and resource utilization, that means optimal use of resources. Based on the current status of the system, the load balancing algorithms can be divided into two categories as static and dynamic load balancing algorithms:

- **Static load balance algorithms**: In this algorithm distributed the traffic equally between servers. By this approach, decide how to distribute the workload accordance with prior knowledge of the problem and characteristics of the system. These algorithms such as Round Robin, Ant Colony optimize, Threshold and Randomized algorithms.
- **Dynamic load balance algorithms**: In this algorithm does not depend prior knowledge of the problem and characteristics of the system. Dynamic algorithms use status information to make decisions during the execution of the program. An important advantage of this approach is that its decision for balancing load based on the current status of the system. These algorithms such as Min-Min, Central Queue, Active Monitor load balancer and Throttled load balancer algorithms.

The proposed algorithm based on the second type of load balance algorithms that is a dynamic load balance algorithm, specifically conducting improvement on Active Monitor load balancer algorithm.

3. RELATED WORK

Matchmaking has received considerable attention from researches mostly in the context of on-demand requests. Many studies and analysis have been performed on matchmaking and load balancing for the cloud environment. The emerging needs and loads in cloud environment had driven research community to developed various load balancing strategies. Shikharesh Majumdar et al [30] proposed Any-Schedulability Criterion to deal with problem of matchmaking in an environment that comprise opaque resources that its local schedule policies are not know. In this environment, deal with advance reservation requests, each request has an earliest start time and deadline. However, the proposed criterion did not take into account the priority utilization of resources. Jasmin James et al [16] proposed weighted active monitoring load balancing algorithm as an improvement over the Active VM Load Balancer [20] via assigning a weight for each VM. Experimental result showed that their proposed algorithm achieved better processing time and response time; however, the proposed algorithm didn't consider process duration for each individual request. Komal Mahajan et al [18] deployed Round Robin (RR) approach [25] for VM Load Balancing. Their proposed algorithm introduced an improvement over the RR algorithm, as it included the state of previous allocation of VM to a request, as a result of experiences that gives better results than the RR algorithm. Tejinder Sharma et al. [21] proposed an enhanced load balance algorithm which lives migration of load is done in a virtual machine to avoid the under utilization and hence improving data transfer cost, this algorithm adopted on round robin and throttled algorithms to improve performance metric such as data processing time and response time, but some variables are not taken into account such as weight of VM. Zaouch et.al [22] presented a study about load balancing techniques in the cloud computing and now these affects Qos. Domanal et.al [23] proposed VM-assign Load Balance algorithm. This algorithm specifies the incoming requests to the available resources. Their proposed algorithm is a modified version of Active Monitoring Load Balance algorithm (AMLB)[20] which maintains information about all VMs and number of current requests allocated to VM. When a new request arrives, load balancer identified least load VM by id. Load balancer return VM id to data center controller. Data center controller sends the request to VM identified by id. Data center notify load balancer of the new allocation. In VM-assign algorithm the first allocation of VM is similar to

AMLB but next allocation put the condition that is not used in previous assignments, however, the least loaded VM which will not chosen in the next iteration may have good processing speed[17]. VM-assign algorithm chooses the VM without the knowledge of his processes power that can do the job or not. Shahapure et.al [24] proposed time sliced and priority load balancing algorithm, it's based on the principle of time scheduling and priority of requests and it considered an enhancement round robin algorithm. Experimental results showed that the algorithm reduced the waiting time and turnaround time. Rashmi et.al [26] proposed Shortest Job First Scheduling with threshold (SJFST) algorithm, using various threshold values, which is considered as a Timer determines the period for the execution of the job. Experimental results showed that the algorithm reduced job rejection rate compared Shortest Job First Scheduling (SJFS) Pan, J.-S. et al [27] proposed Interaction Artificial Bee Colony (IABC) load balance algorithm, it's based on principle of task scheduling to VM.Proposed algorithm makes all task scheduling by using parameter value calculated according to gravity formulation compared to Artificial Bee Colony (ABC) [28], which used this parameter random number during 0 to 1. Experimental results prove that IABC is more efficient compared to ABC. Zhan, Z.-H. et al [29] Proposed a load balance aware genetic algorithm (LAGA) with Min-min and Max-min to solve task scheduling problems, so that used Time Load Balance (TLB) strategy to help establish the fitness function with makespan. Experimental results prove the LAGA algorithm improved several task-scheduling problems compared with another algorithm that not used TLB.

3.1 problem formulation

With the ever increasing number of incoming requests of users, there is an urgent need to matchmaking requests for resources, in order to matchmaking required distributed the workload evenly between the different resources, which causes some of resources not use the more than the others, which means increasing the utilization of resource. That necessitates achieve load balancing between those resources. However, there are only a limited number of VM load balancing having improved the performance, such as response time and resource utilization. Accordingly, this paper proposes new strategy to achieve load balance between heterogeneous resources and improve performance compared to previous strategies in this area.

4. PROPOSED MATCHMAKING FRAMEWORK

Figure 3 depicts the proposed layered model for cloud computing matchmaking. The proposed framework consists of three layers namely, user layer, core middleware layer and system level layer. Each layer performs a specific function and consists of different module. The innovation of the proposed framework is using load balancing algorithm to achieve user's Qos requirements and better resource utilization.

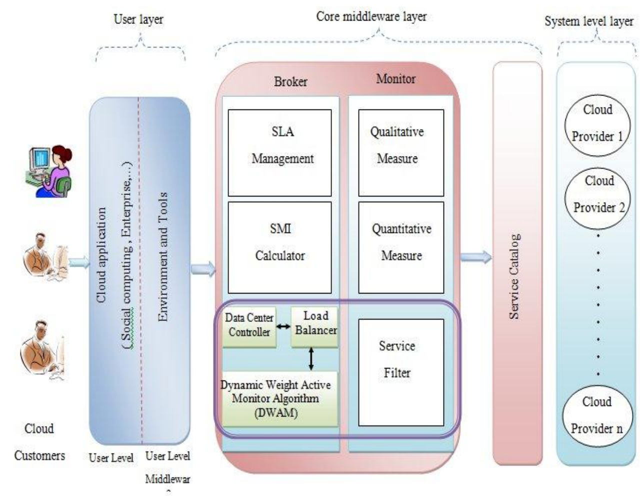

Fig3.*Proposed Framework of Cloud Computing*

The components of the framework are as follows:

1. The user layer: this layer consists of two sub layers as follows:
- User level: is used by the user to deal with the services provided by cloud.
- User level middleware: This component provides environments and tools simplifying the development and the deployment of applications in the Cloud and is constitutes the access point of applications to the Cloud.
- Core middleware layer: This component is responsible for providing a suitable run-time environment for applications and to exploit the physical resources. This layer consists of:
- Broker: This component is responsible for interaction with clients and understanding their application needs. It performs discovery and classification of appropriate services using other components such as the Ranking systems. The cloud broker contains:
- SLA Management:
  Is the component that keeps track of customers' SLAs with cloud providers and their fulfillment history.
- SMI Calculator:
  Is the component that calculates the various Key Performance Indicators (KPI's) which they are used by the classification system to prioritize cloud services.
- Monitoring: this component first discovers Cloud services that can meet users essential QoS requirements. Then, it monitors the performance of the Cloud services. It also keeps track of the formal requirements of SLA previous customers are being satisfied by the Cloud provider.
- Service Catalogue: Catalog: stores the services and their characteristics advertised by various Cloud providers.
- System level layer: This component is characterized by the physical resources such as clusters, datacenters, and spare desktop machines.

4.1 The proposed DWAM Algorithm

Some load balancing technique distributes the load among all nodes without node configuration. The Proposed framework will distribute the load with node configuration like based on weight allocated to the server node. Form 4 it explains the mechanism of load balance algorithm's action in cloud computing, each load balance algorithm has load balancer that identifies VM and send VM IDs to the data center controller, which is responsible for management allocate request to VM.

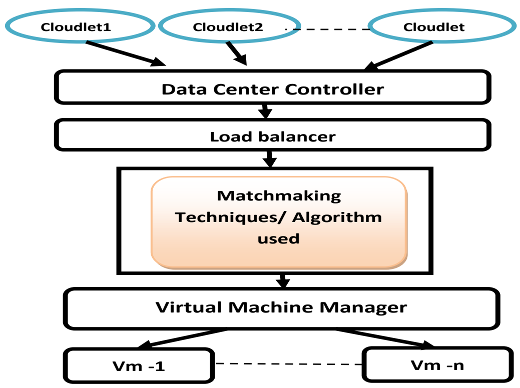

Fig 4: *Load Balance Algorithms Execution*

Dynamic Weight Active Monitor (DWAM) Load Balance algorithm is a modification for the VM-assign Load Balance [23] and Active Monitoring Load Balancer [20] by assigning a dynamic weight to each VM. Unlike previous algorithms which calculated weight in a static way. The proposed DWAM algorithm introduces the concept of dynamic weights with active monitoring. Each VM is assigned a dynamic weight and according to the highest weight, they receive more connections. In a situation, when all the weights become equal, VM will receive balanced traffic. The VM is assigned a varying amount of the available processing power of VMs. To these VMs of different processing powers; the requests are allocated to the most powerful VM and then to the lower and so according to their weight and its availability. Hence, optimizing the given performance parameters. After selecting the VM with the least load and with least process duration is identified, so that if VM was used the last time we choose another, it allocates requests to the most powerful VM according to the weight assigned. The main objective of the proposed algorithm is to achieve better response time, processing time and resource utilization. The weight of VM is calculated as shown in equation 1:

$$VM_w = C_s * (1 - CPU_{ut}) * WC_{cpu} + \left(1 - \left(\frac{T_m - U_m}{T_m}\right) * WC_m\right) - \left(F - A + \frac{D_s}{B.W} + N_{lat}\right) * WC_{net} \qquad (1)$$

Where

Cs = Clock speed
CPU_{uti} = CPU Utilization
WC_{cpu} = Weight constant of CPU
T_m = Total Memory
U_m = Used Memory
WC_m = Weight constant of memory
F = Finish time of user request
A = Arrival time of user request
Ds = data size of single request
Bw = Band width
N_{lat} = Network latency
WC_{net} = Weight constant of network

Here the weight constant assumption parameter into scale between 0 and 1; divide to CPU, Memory and Network, so that the total is currently just 1. The CPU has greater impact on the execution of a VM comparing in memory or net. So, it has a maximum weight constant. The use of a constant value in the equation, because there are different units of measurement for each of memory, CPU and network. Proposed DWAM algorithm VM load balancer and the related flowchart are given in Figures 5 and 6 respectively. Main ideas, that upon the proposed DWAM algorithm are assigned weight in a dynamical way by equation for each VM, depending on process completion time (process duration), a VM with least load and observance if VM with least load used in the last iteration.

DWAM algorithm

Input: No of incoming request r1, r2,………rn.
Available VM vm1, vm2,…… vmn.

Output: Mapping request of VM's such that Qos parameters are fulfill and achieve load balance.

Steps:

1. Create VM's on data center with appropriate memory, storage, bandwidth, ect.
2. Calculate weight factor (Wi) for all VM according to equation (1).
3. DynamicWeightActiveVMLoadBalancer maintains an index table of VMs, associated weighted count and the number of requests currently allocated to the VM. At start all VM's have 0 allocations.
4. When requests arrive at the data center it passes to the load balancer, it parses the table; least loaded VM and with least process duration is selected for execution.

Case I: if found

Check whether the chosen least loaded VM is used immediately in the last iteration

If YES

goto step 4 to find next least VM

If NO

Least loaded VM is chosen

5. Identifying the least loaded VM with least process duration, then load balancer return VM id to data center.
6. Data center send request to VM (identified by id) and it notify the load balancer about new allocation.
7. Load balancer updates the table increasing the allocate count for that VM.
8. When VM finishing processing the request, data center notify the load balancer of VM de-allocate.
9. Load balancer updates the table by decreasing the allocation count for the VM by one.
10. Continue from **step 4** for the next request.

Fig 5. *DWAM proposed algorithm*

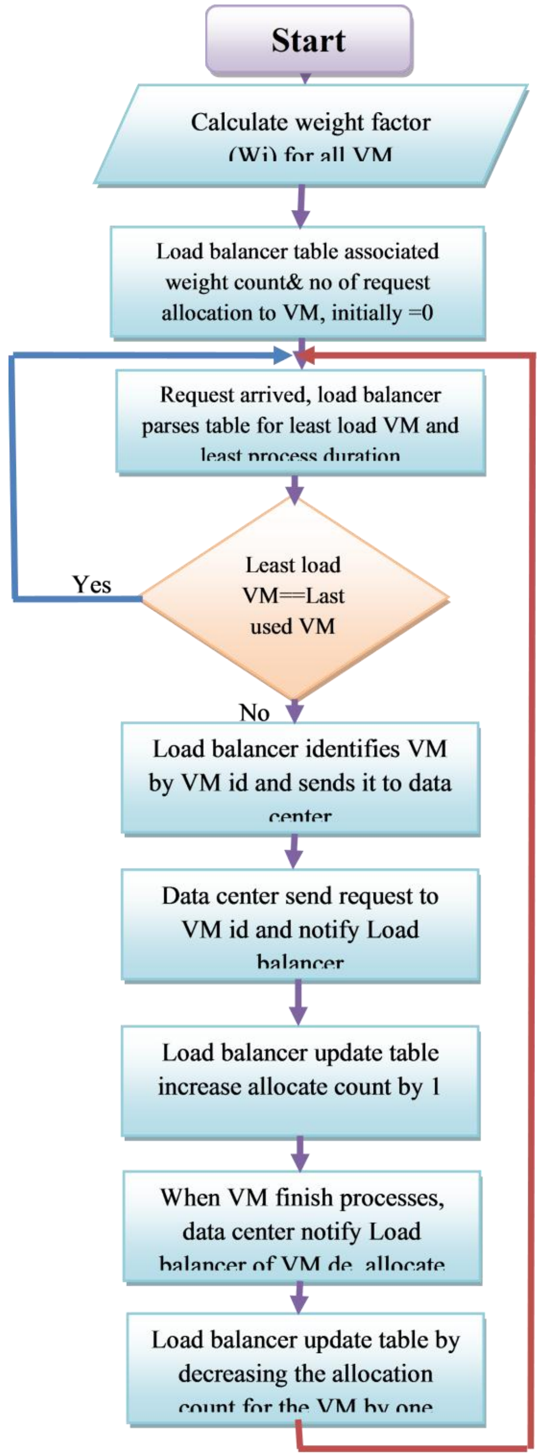

Fig 6. *Flowchart for proposed algorithm*

5. EXPERIMENTAL

The proposed algorithm is implemented in cloudsim simulation toolkit. Java language is used for implement new DWAM load balancing algorithm. To analyze result and compare them with existing algorithm, we used CloudAnalyst tool for simulating proposed algorithm.

5.1 Experimental setup

Cloud Analyst tool gives the real time scenario with six geographical locations. The reservation data centers at the same time. The users of each continent the assumed that a small percentage of the total Internet users is during peak hours and off-Peak hours, users are ten peak hours. For experimentation Internet users in six different continents considered six user bases and peak hours and off- peak hours are given a in Table 3. We have studied Internet users in different continents from month of May 2013.

TABLE 3:
SIMULATION CONFIGURATION

S.No	User Base	Region	Simultaneous Online Users During Peak Hrs	Simultaneous Online Users During Off- peak Hrs
1	UB1	0-N. America	470000	80000
2	UB2	1-S. America	600000	110000
3	UB3	2- Europe	350000	65000
4	UB4	3- Asia	800000	125000
5	UB5	4- Africa	115000	12000
6	UB6	5- Oceania	150000	30500

The number of host and data center, storage of host and VM given in the Table4.

TABLE 4:
SIMULATION CONSIDERATION

Parameter	Value
Simulation toolkit	Cloudsim
Number of host	100
Number of Datacenter	5
Host storage	100GB
VM storage	10GB

Application is deployed in five Data centers located in different parts of the world (six different regions: R0, R1, R2, R3, R4 and R5) as figure 7.

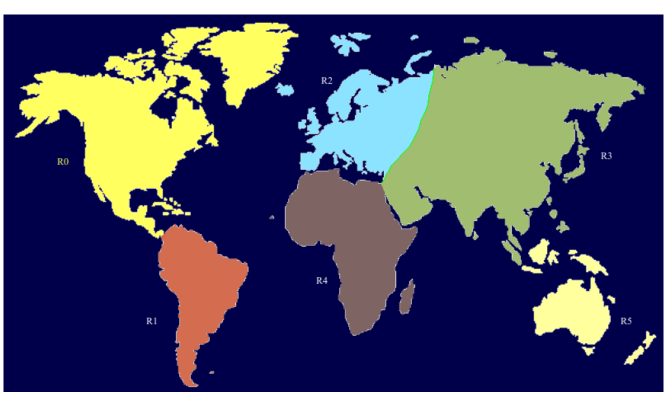

***Fig 7.** Data centers in different regions*

The cloud environment set up generated was having following configuration; data centers configuration as presented in table 5 .Each data center has one VM. We used five VM in the experimental and these have four-type different configuration as in table 6.

TABLE 5:
DATA CENTERS CONFIGURATION

Parameter	Value
Data Center OS	Linux
Data Center Architecture	X64

TABLE 6:
VM CONFIGURATIONS

CPU No. Of processors	Memory	Storage space	Bandwidth
1	1 GB	1TB	1000000 bps
2	4 GB	2 TB	1000000 bps
4	8 GB	4 TB	1000000 bps
8	16 GB	8 TB	1000000 bps

5.2 Result and analysis

Here compared the proposed DWAM load balance algorithm with Active Monitor and VM- assign load balance algorithms. As shown in table 7, fig 8, fig 9 the response time is significantly better than other algorithms. From table 7 can be seen maximum response time decrease from 249.6 to 203.6, that means 18.4% improvement of response time compared VM- assign algorithm and 44.8% improvement compared Active Monitor algorithm. Due to the assigned weights to each VM, which means the request are assigned to most powerful VM and with least process duration. In addition, improvement appears dramatically compared Active Monitor algorithm, because it chooses VM based only with least load.

TABLE 7:
RESPONSE TIME (MS) RESULTS

VM#	ACTIVE MONITOR	VM- ASSIGN	DWAM
VM0	369.36	236.34	90.54
VM1	362.93	249.6	75.6
VM2	339.64	124.51	168.64
VM3	278.19	181.69	203.6
VM4	300.96	221.68	193.5
MAX	369.36	249.6	203.6
MIN	278.19	124.51	75.6
AVG	1651.08	1013.82	731.88

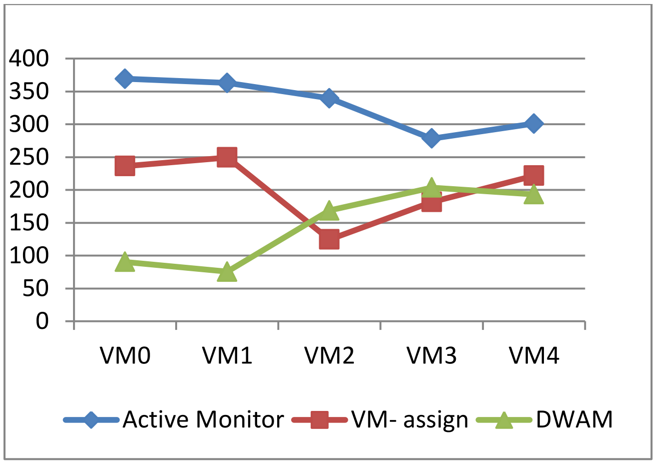

Fig 8. *Response Time*

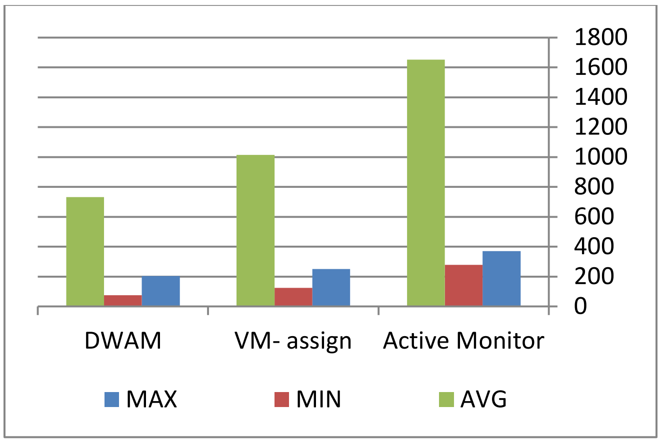

Fig 9. *Response Time results comparison*

The proposed DWAM algorithm distributes the incoming requests to all VM's. Therefore, it more utilization of the resources compared to Active load balancer and VM-assign as shown table 8, fig 10, fig 11. Proposed algorithm will not allow the VM, which was allocated in its previous step, so that take chance to other least load VM that achieved more utilization of the resource. From table 8 we can note proposed algorithm more utilized of resource by 59.7% compared Active Monitor algorithm and 11% compared VM-assign algorithm. This proves that the proposed contributed to the improvement of resource utilization when it assigned weight to each VM and VM with least process duration. In additional, it ruled out the least load VM that used in last iteration.

TABLE 8:
RESOURCE UTILIZATION RESULTS

VM#	ACTIVE MONITOR	VM- ASSIGN	DWAM
VM0	353.94	180.32	160.32
VM1	212.75	177.32	155.65
VM2	398.39	146.3	157.89
VM3	183.38	90.36	70.65
VM4	196.44	178.61	130.97
MAX	398.39	180.32	160.32
MIN	183.38	90.36	70.65
AVG	1344.9	772.91	675.48

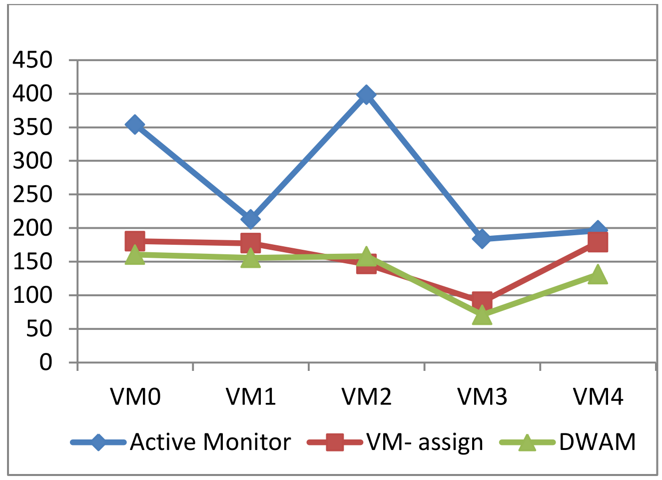

Fig 10. *Resource utilization analysis*

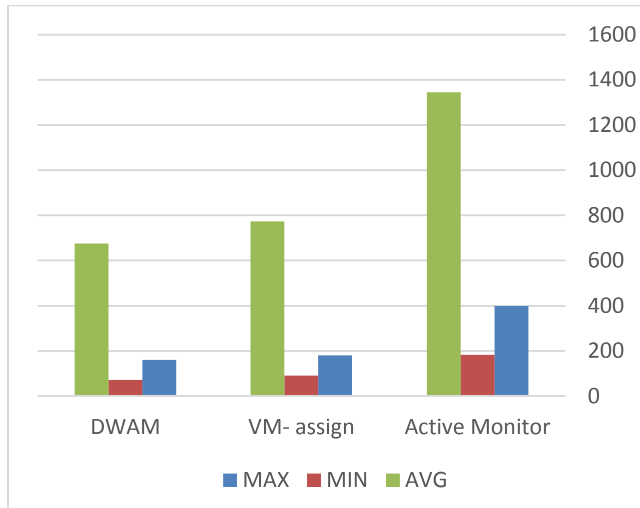

Fig 11. *Resource utilization results comparison*

The results in table 9,fig 12, fig 13 show that proposed algorithm reducing data center processing time compared two algorithms. To adopt the proposed algorithm to assigned weight for each VM play an important role in improvement of data center processing time. Therefore, the request allocated to highest weight of VM, which makes it implement request in shortest possible time. As table 9 we notice 17% improvement process time compared Active Monitor algorithm and 11% compared VM-assign algorithm.

TABLE 9:
DATA PROCESS TIME (MS) RESULTS

VM#	ACTIVE MONITOR	VM- ASSIGN	DWAM
VM0	0.965	0.9	0.7
VM1	0.854	0.765	0.64
VM2	0.4	0.7	0.8
VM3	0.48	0.58	0.44
VM4	0.8	0.45	0.3
MAX	0.965	0.9	0.8
MIN	0.4	0.45	0.3
AVG	3.499	3.395	2.88

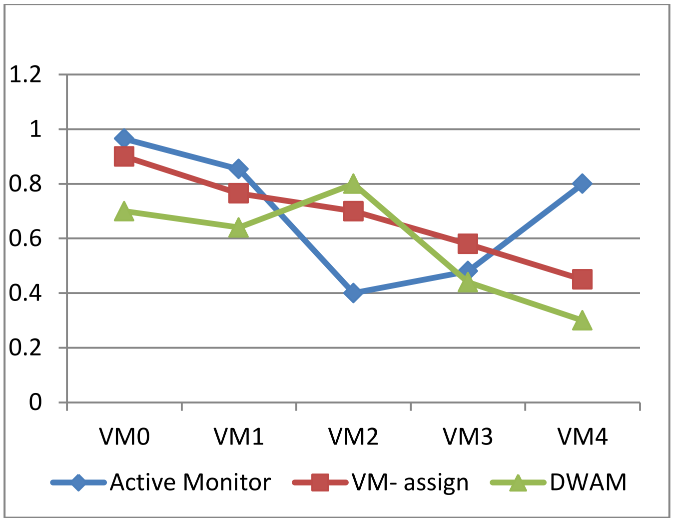

***Fig 12.** comparison analysis between proposed algorithm Vs other algorithm*

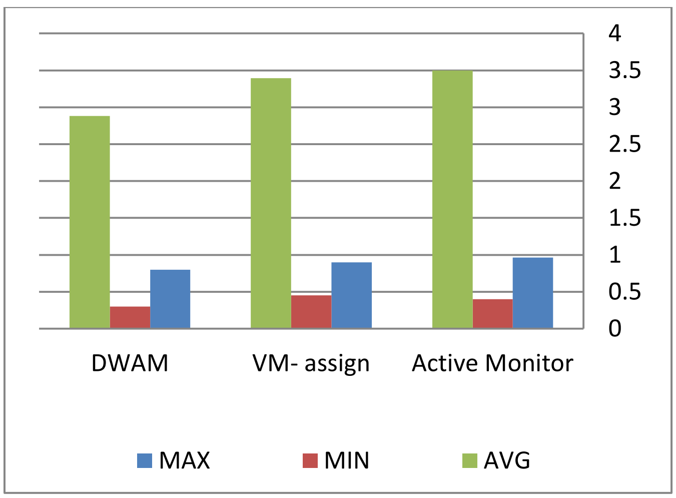

***Fig 13.** Data Process Time comparison*

The above shown tables and figures clearly indicates that the parameters: response time, data processing time and resource utilization are bit improvement in proposed DWAM algorithm compared Active Monitor and VM-assign algorithms.

6. Conclusion

In this paper proposed new strategy of load balance and then implemented in cloud environment using CloudSim toolkit. In proposed DWAM algorithm the requests are located to the most power full VM, that is by assign the weights dynamically for each VM. From the results in a table 7, we can say Improvement largely on response time compared other algorithms. According to the experiment, we conclude that if select VM with least load, with least process duration and highest weight for execution of request then it also increase the performance of the system also decrease the response time of the requests and data center processing time. From table 8, we can say that the proposed algorithm is more utilization of the resource. When compared proposed DWAM algorithm to Active monitor and VM-assign algorithms, we noticed response time reduced, data processing time is also decrease and more utilized of resource. The improvement in the performance can be noted when the algorithm run on more number of data centers. We also conclude that proposed DWAM load balance algorithm is best among other.

References:

[1] Shroff, G., 2010. Enterprise cloud computing. Enterprise Cloud Computing Technology, Architecture, Applications, pp 241–242.

[2] Kalagiakos, P. & Karampelas, P., 2011. Cloud Computing learning. 2011 5th International Conference on Application of Information and Communication Technologies (AICT), pp 7–11.

[3] Vahora, S. & Patel, R., 2015. CloudSim-A Survey on VM Management Techniques. IJARCCE International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 1, January, pp 128–133.

[4] Sidana, S., & Suri, B. 2013. Cloud Computing: A Review International Journal of Advanced Computer Science and Software Engineering, Vol. 3, issue 4, April

[5] Chung, M. & Hermans, J. 2010. From Hype to Future. KPMG's 2010 Cloud Computing Survey. KPMG.

[6] Peter, M. & Timothy, G. 2011. The NIST Definition of Cloud Computing . National Institute of Standards and Technology, Gaithersburg, MD, Sept.

[7] Manvi, S.S. & Shyam, G.K., 2014. Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. Journal of Network and Computer Applications, pp 424–440.

[8] Kansal, N. J. & Chana, I. 2012. Cloud Load Balancing Techniques: A Step Towards Green Computing. Environment International Journal of Advanced Computer Science Issues, Vol. 9 (1).

[9] Sanchari, S. & Abhilash, K.V., 2014, A Survey on Resource Management in Cloud Computing. International Journal of Computer and Information Technology, Vol. 5 (3) , 3887-3889

[10] Hoang, P. et al., 2014. Resource management techniques for handling uncertainties in user estimated job execution times. International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2014).

[11] P, A.S., 2012. Linear Scheduling Strategy for Resource Allocation in Cloud Environment. International Journal on Cloud Computing: Services and Architecture IJCCSA, Vol.2, No.1, February, pp 93–17.

[12] Soni, S.K. & Kapoor, R.K., 2013, A Survey on Pre-copy Based live Migration of Virtual Machine in Cloud Environment International Journal of Advanced Computer Science and Software Engineering, Vol. 3, issue 10, October.

[13] Roy, N., Dubey, A. & Gokhale, A., 2011. Efficient Autoscaling in the Cloud Using Predictive Models for Workload Forecasting. 2011 IEEE 4th International Conference on Cloud Computing, pp. 500–507.

[14] Akiyama, S. et al., MiyakoDori: A Memory Reusing Mechanism for Dynamic VM Consolidation. 2012 IEEE Fifth International Conference on Cloud Computing, pp. 606-613.

[15] Song, Y., Sun, Y. & Shi, W., 2013. A Two-Tiered On-Demand Resource Allocation Mechanism for VM-Based Data Centers. IEEE Transactions on Services Computing IEEE Trans. Serv. Comput., pp 116–129.

[16] James, J. & Verma, B., 2012. Efficient VM Load Balancing Algorithm for a Cloud Computing Environment. International Journal of Advanced Computer Science & Engineering, Vol. 4, pp 1658-1663, ISSN: 0975-3397, September

[17] Shaw, S.B. & Singh, A.K., A survey on scheduling and load balancing techniques in cloud computing environment. 2014 International Conference on Computer and Communication Technology (ICCCT), pp 87–95.

[18] Mahajan, K., Makroo, A. & Dahiya, D., 2013. Round Robin with Server Affinity: A VM Load Balancing Algorithm for Cloud Based Infrastructure. Journal of Information Processing Systems, Vol.9, No.3, September, pp 379–394.

[19] Mohapatra, S., Rekha, K.S. & Mohanty, S., 2013. A Comparison of Four Popular Heuristics for Load Balancing of Virtual Machines in Cloud Computing. International Journal of Computer Applications IJCA,Vol. 68- No. 6, pp 33–38.

[20] Hemant, S.M. & Parag R.K. & Vinay, C. 2013. Load Balancing On Cloud Data Centres. International Journal of Advanced Computer Science and Software Engineering, Vol. 3, issue 1, January.

[21] Sharma,T. & Banga, V.K. ,2013, Efficient and Enhanced Algorithm in Cloud . International Journal of Advanced Computer Science and Applications IJACSA, ISSN: 2231-2307, Vo. 3,pp 385-390, Issue-1, March.

[22] Zaouch, A. & Benabbou, F., 2015. Load Balancing for Improved Quality of Service in the Cloud. International Journal of Advanced Computer Science and Applications IJACSA, Vol. 6,No.7,pp 184-189.

[23] Domanal, S.G. & Reddy, G.R.M., 2014. Optimal load balancing in cloud computing by efficient utilization of virtual machines. 2014 Sixth International Conference on Communication Systems and Networks (COMSNETS) IEEE. Bangalore

[24] Shahapure, N.H. & Jayarekha, P., 2015. Time sliced and priority based load balancer. 2015 IEEE International Advance Computing Conference (IACC).

[25] Mustafa, E. et al., 2015. Load Balancing Algorithms Round-Robin (RR), Least-Connection and Least Loaded Efficiency. International Journal of Computer and Information Technology. Vol 04 – Issue 02, March. pp 255-257.

[26] Srinivasan, R.K., Suma, V. & Nedu, V., 2013. An Enhanced Load Balancing Technique for Efficient Load Distribution in Cloud-Based IT Industries. Advances in Intelligent Systems and Computing Intelligent Informatics Springer, Heidelberg , Vol. 182, pp 479–485.

[27] Pan, J.-S. et al., Interaction Artificial Bee Colony Based Load Balance Method in Cloud Computing.2015. Advances in Intelligent Systems and Computing Genetic and Evolutionary Computing.Vol. 329, pp 49–57.Switzerland,Springer.

[28] Zhao, J., Lv, L. & Sun, H., Artificial Bee Colony Using Opposition-Based Learning. 2015. Advances in Intelligent Systems and Computing Genetic and Evolutionary Computing, Vol.329, pp 3–10. Switzerland,Springer.

[29] Zhan, Z.-H. et al., Load Balance Aware Genetic Algorithm for Task Scheduling in Cloud Computing.2014. Lecture Notes in Computer Science Simulated Evolution and Learning, Vol. 8886. pp 644–655. Switzerland,Springer.

[30] Melendez, J.O. & Majumdar, S., Matchmaking on distributed systems with limited knowledge of resources. 2010. Ottawa ON: IEEE Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010 International Symposium on.