International Journal of Engineering & Technology, *7 (4.12) (2018)* 63-68

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Comparative Study of Different Load Balancing Techniques

Jyoti Parashar [1]*, Dr. Avinash Sharma[2]

[1] *Research Scholar,Maharishi Markandeshwar (Deemed to be University), Ambala, Haryana*
[2]*Maharishi Markandeshwar (Deemed to be University), Ambala, Haryana*
**Corresponding author E-mail:* jyoti.parashar123@gmail.com

Abstract

Cloud computing is a new technology used to manipulate, configure and can be used to access distributed computing applications in the network. It implements the load balancing approach which is used to distribute all of its workload to every node connected in the network. By using this technique resource utilization is done properly. It can also used to achieve user satisfaction and computing resources. If load balancing is used properly then it can efficiently and properly implement the fail-over, scalability, over- provisioning techniques. It can also minimize the resources used and avoid the bottleneck. In my research, review of different load balancing techniques, its usage, limitations, applications and various performance metrics are described..

Keywords: *Cloud Computing; Load Balancing; Different Load Balancing algorithm*

1. Introduction

The word "cloud" is originated from the telecommunication network. It is used by Virtual Private Network (VPN) for communication of data. Cloud computing can be used at the demand of the consumer. Shared resources, different types of information, various software and other devices are provide at their demand.

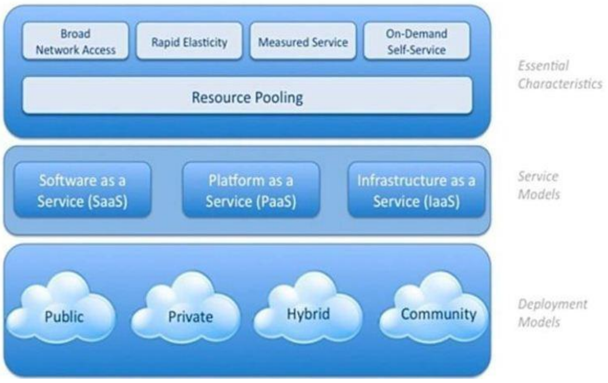

Fig. 1: Architecture of cloud computing [13]

Cloud computing is a distributed design that brings together server resources on an acceptable stage in order to provide on request figuring resources and administrative data. A cloud specialist organization (CSP's) provides the different stages to their customers to use the services and make the web administrative control. This service is similar to a broadband connection offered by the service provider for the Internet connection [3]. Cloud computing provides the services through the Internet; these service belongs to hardware and software both. Due to huge amount of user data on the cloud network resources are not utilized properly and in results delay and service failure is occur. To avoid this problem load balancing method is used in which request from the user is distributed among different resources.

2. Load Balancing

Load balancing is a process of distributing loads and computers resources in cloud computing environment. It allows managing the load demands by divide the resources into multiple computers. Work load is divided into multiple systems to maximize throughput and minimize the cost. Load Balancing can be done by scheduling task, propose resource allocation and task migration [6].

2.1. Methods of load balancing

1. Generic hash method
2. Least Time, Optionally Weighted
3. IP Hash
4. Round Robin
5. Least connected, Optionally Weighted

2.2. Pros of Cloud load balamcing

- **Increase Scalability:** Load Balancing increase the scalability by balancing the traffic of the source. By using effective load balancers it is easy to increase the traffic of the users on the network server. It is very useful on the e-commerce sites where the users are in high rate.
- **Traffic handling:** Load balancing performs a great role when the traffic is increased suddenly on the specific resource, site or a server. It occurs when the number of users is increased for the same data at the same time. For handling this situation Load balancing is performed.
- **Failure management:** It helps in identifying the faults in the early stage and provides the effective solution to manage them properly before the failure occurs. Due to multiple data center availability it helps to reduce the failure rate by bypass the failure using distributed re-

sources the nodes that are unaffected and causing minimum disturbance on the network.

3. Clousim Simulator

CloudSim is a framework which provides the simulation and experimentation model in cloud computing. CloudSim is introduced by the CLOUDS laboratory in 2012. It provides the simulation for the large data centers of the cloud computing. By using these components it is easy to implement the new policies, algorithms and scheduling approaches. This tool is also used for various strategies like cost, time and application execution time.

1. User Interface Structure: In this part of CloudSim user interface machines are connected to each other for user requests.
2. Virtual machine services: This level deal with the services provided by the virtual machine and virtual management of the task performed.
3. Cloud Service: This level of the simulator manages the memory allocation, CPU allocation, Storage allocation and the bandwidth allocation
4. Cloud Resources: This level link with the host and data center of the cloud simulator.

4. Load balancing different approach

Dasgupta, Kousik, et al.[1]: authors in present the effective resources in cloud infrastructure a genetic algorithm is introduced which is based on load balancing approaches for cloud computing. This algorithm is used to reduce the make span of a given tasks set. The simulation of proposed strategy is done by using cloud analyst simulator. The simulated result shows that proposed algorithm not only performs better than other techniques but also gives the guarantee to the requirement of customer jobsIt has been assumed that all jobs have the same priority, which is not the case; it is housed in the JUV and carefully taken in function of fitness.

The algorithm proposed in authors is given below:

GA Load Balancing Algorithm

Step1: Processing units are randomly initialized after the encoding process into strings.

Step 2: Calculate the fitness values by using eq 3 [Fitness].

Step3: While when maximum number of iteration are increased or optimum solution is found Do:

Step3 (a): Selects the chromosomes which have lowest fitness twice and eliminate those chromosomes which have highest fitness value for constructing the mating pool [selection].

Step3 (b): Perform the single-point cross by randomly selecting an offspring that you select based on the crossover point.

Step3 (c): Mutation of the new offspring with a mutation probability of (0.05) [Mutation].

Step3 (d): Place a new offspring as a new population and use this population for the next step Iteration [Accept].

Step 3(e): End condition test [Test].

Step 4: End

Wen, Hui, et al. [2]: the researchers presented a cloud-based multimedia system, an effective load-balancing approach called "Multimedia Load Balancing Stress" (CMBL). First, discussions on multimedia applications and cloud-based work. This algorithm provides the effective solution to the problem related to multimedia applications, computational complexity. The information provides an effective LBA for CBMS. It can fully consider the load servers and network status. The effectiveness of the approach is given in the experimental results.

Proposed Algorithm

```
for all p   P do
 2: Evaluate network link latency from client p to individual landmark
Calculate landmark order of client p to landmark bp
Set of available nodes are acquired, R
for all r   R do
Evaluate network link latency from node r to each landmark
Calculate landmark order of node R to landmark br
If  landmark br = landmark bp then
Evaluate network proximity, arp
 for all b € B do
Measure server utilization srb
 end for
Compute traffic load trp
compute cost Erp
else
Erp =  ƆC
end if
end for
end for
choose optimal node op Node(r. p) minimizing Erp
Add op Node(r. p) to op L List(p)
 for all p   P do
if |op L List(p)| > 1 then
Assign client p to the first item of op Node (p)
else
Assign client p to op Node (p)
 end if
end for
```

Zhao, Yi, et al. [3]: authors in present a adapting live migration of virtual machines researcher about the intra issue cloud load balancing. To reduce the immigration time of virtual machines used to frame and implementing a load balancing model and proposed an algorithm to obtain equilibrium solution of the problem. The experimental result shows that the introduced technique converges fast but slow in migration of open VZ virtual. The given algorithm is shown below.

Algorithm COMPARE_AND_BALANCE

```
Every individual VM on present host cur do
   for all host p   [k] do
          l1         number of VMs on host p
   end for
     l1
    Select B   [K] instantly with P[B] =  /l
     e'          EVALAUTED_COST (B)
     e            EVALUATE_COST (cur)
     If e' < e Then
          with probability e-e' transfer to b
    end if
end for
```

Soni, et al [4]: a proposed to stable the load over virtual machines in cloud data center an techniques termed as central load balancer is introduced. This technique ignore over and under loading of virtual machine. This technique helps to stable the distributed load in virtual machines. The result shows that proposed algorithm gives better load balancing for a heavy CC environment while comparison with other techniques.

Algorithm of Central Load Balancer

Step 1: at starting point all the virtual machines are in available states. CLB used to maintain a table which consists of virtual machine id (VMid) and priority of VMs.
Step 2. New request is received from data center controller.
Step 3. For next allocation CLB are queried by Data Center Controller.
Step 4. CLB parses the table from top to determine the availability of the highest priority virtual machine and the state of that virtual machine's.
If found:
a) The CLB returns the VM id back to the Data Center Controller.
b) VMid received the request from data center controller.
c) all the notifications regarding new allocation is forwarded by data center controller to CLB
d) On the basis of notification CLB upgrades the table.
If not found:
e) The CLB returns -1.
f) The request in queues is done by data center controller.
Step 5. Data center also used to notify the CLB about the VM deallocation while VM finishes processing of requests response cloudlet is received by data center controller.
Step 6. Proper check is done by data center controller regarding waiting request in queue. IF thre exist any waiting request it used to process from step 3.
Step 7. Continue from step 2.

5. Load balancing distributed system

Garg, et al. [9]: authors present there is an algorithm proposed which is inspired from ant colony optimal technique. This algorithm is designed on a semi-distributed system where there is a formation of cluster and these cluster contains number of nodes. Then there is an introduction of ant on the cluster as per the need of environment. The main objective is to improve the performance of system by developing load balancing technique. The result shows that semi-distribution approach gives better use of resources as compare to other approaches.

Semi Distributed System Algorithm

Step 1: Server recieved the request by client

Step 2: Active ant and passive ant are created

Step 3: Selection of nodes randomly.

Step 4: Go to step 5 if node is under loaded otherwise go to step 3.

Step 5: We create return ant when active ant place in its destination and got o step 9 otherwise go to step6.

Step 6: If active ant terminated than passive ant used to go to starting point and there it request for new active ant
Else, if PA gets destroyed than active ant by its own create new passive ant.

Step 7: End if

Step 8: Go to step 3

Step 9: Exit

Siar, et al. [10]: researchers present to a stable or balance the load in a distribution computing system algorithm which is proposed on the basis of game theory. The game theory models the load balancing problem as a non-cooperative game over the users. The main aim of this proposed algorithm is to distinguish fairness in system as well in level of user. By utilizing GS and hybrid population approximation is done to the solution of optimal and near optimal games. Simulated result shows optimality of algorithm in terms of fairness.

Load Balancing Game Algorithm

I. Initialization Phase
1. Initialize k, user's number.
2. Initialize l, computer number.
3. Initialize the 1×k, vector μ, rate of processing in computer.
4. Initialize the 1×l, vector λ, rate of job arrival of users.
5. Randomly generate the initial profile of players, on the basis of initial preferences.
II. Estimation on optimal solutions
6. While No elimination –condition.
7. For individual player in particular game.
8. Execute the algorithm on profile of player to estimate its optimal response, where strategies of other player are fixed.
9. Replace the estimated optimal strategy of players into overall strategy profile.
10. End for
11. End while
III. Returning on the estimated optimal solution
12. Return the strategy profile of game as solution.

Beniwal et al. [11] proposed an approach of semi distributed framework to enhance the web performance. The author used server load balancing method in which clients request are divided among the group of servers. Static and Dynamic method of load balancing approaches were used for load balancing in distributed network. Dynamic load balancing scheme works at the rum time. In the proposed method semi- distributed approach is used and tested on two different applications with various parameters.

Semi- Distributed Approach

Step1. Request sent by client.

Step2. Request accepted by process distribution module.

Step3. Forwarded to cluster.

Step4. Central node receives request.

Step5. Status of subordinate node validated.

Step6. If subordinated node is under loaded.

Go to step 8

Else repeat the step 5

Else, if go to step 7

Step7. Transfer the node to the neighboring cluster.

go to step 4.

Step8. Execute the request

Step9. Respond back to the client.

Data sets: In below approaches use workflows. In this paper use genome, LIGO and SIPHT workflow which increases complexity and dependency.

Approaches: In review paper analysis the normal load balancing, optimize load balancing, probabilistic load balancing and machine learning base load balancing approaches.

Pak, Ilchol, et al. [12] proposed a load balancing approach for multi-layer web server cluster. Every individual layer contains multiple load balancers to balance the client's request that occurred simultaneously. In this proposed methodology upgraded load balancing algorithm is proposed which evaluate the weights according to the use of the resources and divide the load according to status of load of the target server. The results of this algorithm provide the better and effective result.

```
Calculate weight ( P[r], RN, s1,m)
Input: P[r]: number of available resources.
RN: Request Number
s1 :Consumption of resources by per request
m: server numbers
outcome :
Min_Res←min(P[r]);
Max_Res←min(P[r]);
Idle_Res←Max_Res-Min_Res;
α←RN/m
β=Idle_Res/s1
Ass_Rate=( P[r]-Min_Res)/Idle_Res;
While(T<=Th)
    If (P[r]== Max_Res) then ω[r]←α+β/2;
       else if(P[r]== Min_Res) then ω[r]←α+β/2;
       else if ω[r]←Ass_Rate*α
   else if
 default: ω[r]←0
return ω[r];
```

6. Result and Analysis

Table 1: Comparitive Analysis Of Load Balancing Algoritms

Algorithms And Parameters	Ga Load Balancing Algorithm	Constraint Multimedia Load Balancing (Cmlb)	Compare And Balance	Central Load Balancer
Minimizing Tasks	Yes	**No**	**No**	**Yes**
Scheduling	**No**	**Yes**	**No**	**No**
Distributed	**No**	**Yes**	**Yes**	**Yes**
Load Balance	**No**	**Yes**	**No**	**Yes**
Over Load	**No**	**No**	**No**	**Yes**
Stability	**No**	**Yes**	**No**	**Yes**
Priority	**Yes**	**No**	**No**	**No**

Response time: Figure 1.1 shows the comparison of different approaches based on response time delay. The Response time is an important approach in multitasking scenario. In this paper, web based distributed scenario is used. The multi-task scenario is used in this system which can cause the task switching. Out of given approaches, which approach will take less time for task allocation in first time on virtual machines. In figure 1.2 CLB shows significantly less time for response time.

Table 1.1: Table of response time

No of Nodes	CMLB	CB	CLB
5	*5*	*9*	*4*
10	*6*	*11*	*8*
15	*7*	*13*	*10*
20	*9*	*15*	*12*
25	*11*	*19*	*15*

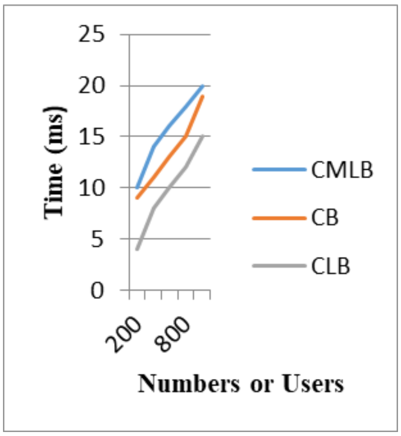

Fig. 1.1: Graph of response time

Throughput: Figure 1.2 represents the analysis of throughput based on task computation on virtual machine in one-time unit. Throughput indicates the optimized utilization of resources. In above figure 1.1 number of nodes represented by X-axis and throughput represented by Y-axis. There is significant improvement in CLB in optimize utilization of resources therefore it increases the throughput. This analysis also related to the Figure 1.2, in which response time reduced and hence number of task increased in that time span.

Table 1.2: Table of Throughput

No. of nodes	CMLB	CB	CLB
5	3	4	1
10	5	7	3
15	7	8	5
20	6	8	10
25	7	9	10
30	8	10	12

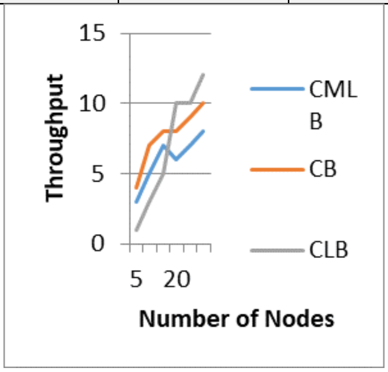

Fig. 1.2: Graph of throughput

Cost: Figure 1.3 shows the Cost of resources. Cost represents the use of resources such as Memory, Processor and Virtual machine. If task is not increased but cost is increased, then there will be negative improvement. On the other hand, if task is increased and cost is also increased then it shows neutral improvement. If task is increased and cost is not increased, then there will be positive improvement in algorithm. In the above figure, in CLB cost is increased for same number of task.

Table 1.3: Table of Cost

No of Users	CMLB	CB	CLB
10	36	38	42
30	42	48	50
50	66	70	73
70	95	98	105
90	113	118	124

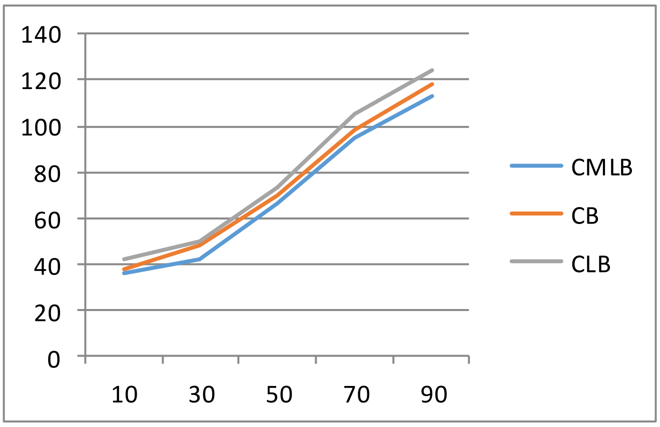

Fig. 1.4: Graph of cost

Throughput in Distributed System: In Figure 1.4 & 1.5analysis of different distributed systems, semi distributed and Game algorithm is done based on throughput and Response time respectively. In figure 1.5, throughput of distributed system based on Game algorithm shows significant improvement as compare to other distributes system. While in Figure 1.6 the distributed system based on game algorithm shows less response time as compare to other distributed system.

Table 1.4: Table of throughput in distributed systems

No. of real server in Web Server Cluster	Normalized Throughput (Distributed)	Normalized Throughput (Semi-Distributed)	Normalized Throughput (Game Algorithm)
8	0.4	0.5	0.6
12	0.47	56	0.7
16	0.49	0.68	0.8

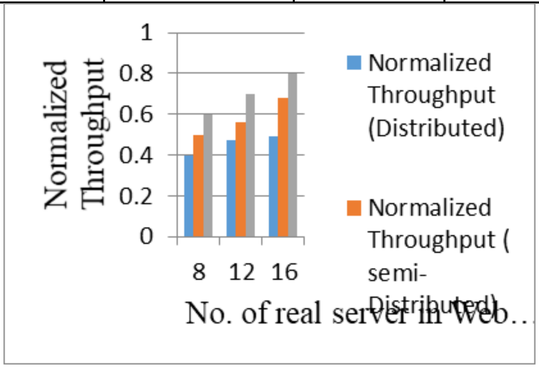

Fig. 1.4: Graph of throughput in distributed systems

Table 1.5: Table of Response in distributed systems

No. of Concurrent Request	Distributed	Semi distributed	Game Algorithm
100	78	77	75
200	135	125	120
300	155	145	130
400	180	160	140
500	190	168	165
600	185	179	177
700	400	320	230
800	550	480	350
900	700	615	425
1000	810	680	554

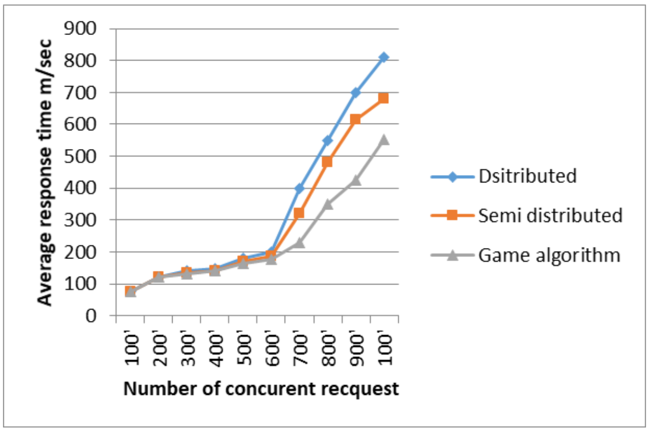

Fig. 1.5: Graph of Response in distributed systems

6. Conclusion

The principle accessible definitions removed from the writing have been broke down to give both an integrative and a fundamental Cloud definition. Despite the fact that our incorporating definition is covered with numerous framework ideas our shared factor definition features the significant highlights of Clouds in web services distributed environment.

References

[1] Dasgupta, Kousik, et al. "A genetic algorithm (ga) based load balancing strategy for cloud computing." *Procedia Technology*10 (2013): 340-347.

[2] Wen, Hui, et al. "Effective load balancing for cloud-based multimedia system." *Electronic and mechanical engineering and information technology (EMEIT), 2011 international conference on*.Vol. 1.IEEE, 2011.

[3] Zhao, Yi, and Wenlong Huang. "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud." *INC, IMS and IDC, 2009.NCM'09.Fifth International Joint Conference on*.IEEE, 2009.

[4] Soni, Gulshan, and Mala Kalra. "A novel approach for load balancing in cloud data center." *Advance Computing Conference (IACC), 2014 IEEE International*.IEEE, 2014.

[5] Bo, Zhang, GaoJi, and Ai Jieqing. "Cloud loading balance algorithm." *Information Science and Engineering (ICISE), 2010 2nd International Conference on*.IEEE, 2010.

[6] Randles, Martin, David Lamb, and A. Taleb-Bendiab. "A comparative study into distributed load balancing algorithms for cloud computing." *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*.IEEE, 2010.

[7] Zhao, Yi, and Wenlong Huang. "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud." *INC, IMS and IDC, 2009.NCM'09.Fifth International Joint Conference on*.IEEE, 2009.

[8] Al Nuaimi, Klaithem, et al. "A survey of load balancing in cloud computing: Challenges and algorithms." *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on*. IEEE, 2012.

[9] Garg, Atul. "A Framework to Optimize Load Balancing to Improve the Performance of Distributed Systems." *International Journal of Computer Applications* 122.15 (2015).

[10] Siar, Hajar, KouroshKiani, and Anthony T. Chronopoulos. "An effective game theoretic static load balancing applied to distributed computing." *Cluster Computing* 18.4 (2015): 1609-1623.

[11]Beniwal, Payal, and AtulGarg. "Proposed Semi-Distributed Framework to Enhance Web Performance." *International Journal of Computer Applications* 141.12 (2016).

[12] Pak, Ilchol, et al. "An efficient load balancing approach for n-hierarchical web server cluster." *Wuhan University Journal of Natural Sciences* 20.6 (2015): 537-542.

[13] Tushar Desai and JigneshPrajapati." A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing."*International Journal Of Scientific & Technology Research*, Volume 2, Issue 11, November 2013

Authors:

Jyoti Parashar received degree of Master of Computer Applications from Maharshi Dayanand University, Rohtak **in 2013 and completed his M.phil degree from Maharishi Markandeshwar University, Mullana (Ambala) in 2015.**

Dr.Avinash Sharma is working as Professor in computer science & engineering department in Maharishi Markandeshwar Maharishi Markandeshwar engineering college.