# Analysis Of Alba Efficiency For Distributed Cloud Computing

**Madanmohan K[a], Dr. Tryambak Hirwarkar [b]**

[a] *Research Scholar, Dept. of Computer Science & Engineering,*
*Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal Indore Road, Madhya Pradesh, India*
[b] *Research Guide, Dept. of Computer Science & Engineering,*
*Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal Indore Road, Madhya Pradesh, India*

**Article History**: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

**Abstract:** Balancing the computational load of various simultaneous tasks on heterogeneous architectures is one of the basic prerequisites for efficient use of such systems. Load-imbalance is naturally present if the calculation load is disseminated Non consistently across different tasks or if execution time for similar sort of tasks changes from one class of handling component to the next. Load-imbalance may anyway likewise emerge from causes that are outside the ability to control of the client, as case operating system jitter, over-membership of the accessible specialists, impedance and asset conflict by simultaneous tasks, and so on composing a fair equal application requires cautious investigation of the issue and a decent downplaying of different hardware architectures of the computing nodes.

## Introduction

Cloud Computing is the delivery of computing administrations like workers, storage, databases, networking, programming, examination, insight, and that's only the tip of the iceberg, over the Cloud. Cloud Computing gives an option in contrast to the on-premises datacentre. With an on-premises datacentre, we need to oversee everything, like buying and introducing hardware, virtualization, introducing the operating system, and some other required applications, setting up the network, arranging the firewall, and setting up capacity for data. In the wake of doing all the set-up, we become liable for keeping up it through its whole lifecycle.

The US National Institute of Standards and Technology (NIST) describes cloud computing as ". . . a compensation examine model for empowering accessible, helpful, on-request network admittance to a common pool of configurable computing assets (for example networks, workers, storage, applications, benefits) that can be quickly provisioned and delivered with negligible administration exertion or specialist organization cooperation."

The cloud environment gives an effectively available online entrance that makes convenient for the client to deal with the process, storage, network, and application resources.

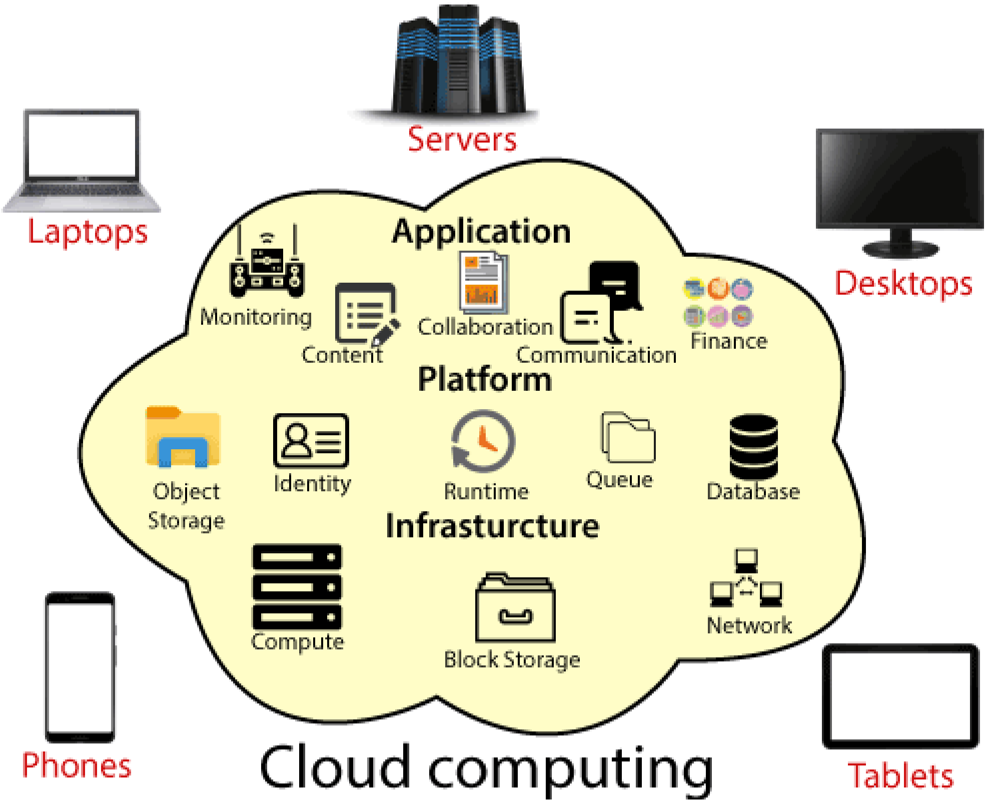


**Figure 1. Architecture of Cloud**

**Advantages of cloud computing**

Cost: It reduces the huge capital costs of buying hardware and software.

Speed: Resources can be accessed in minutes, typically within a few clicks.

Scalability: We can increase or decrease the requirement of resources according to the business requirements.

Productivity: While using cloud computing, we put less operational effort. We do not need to apply patching, as well as no need to maintain hardware and software. So, in this way, the IT team can be more productive and focus on achieving business goals.

Reliability: Backup and recovery of data are less expensive and very fast for business continuity.

Security: Many cloud vendors offer a broad set of policies, technologies, and controls that strengthen our data security.

**Load Balancing**

In cloud environment, Load balancing is a technique that circulates the overabundance dynamic neighborhood workload equitably across all the nodes. Load balancing is utilized for accomplishing a superior help provisioning, asset usage and improving the general execution of the system. For the appropriate load circulation a load balancer is utilized which got tasks from various area and afterward disseminated to the data center. A load balancer is a device that goes about as an opposite intermediary and appropriates network or application load across various workers. Figure 2 presents a system under which different load balancing calculations work in a cloud computing environment.

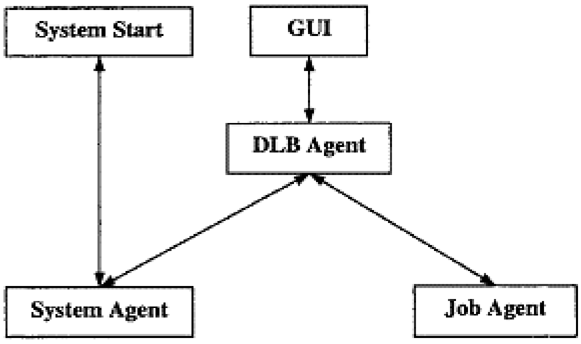


**Figure 2. Framework for working of Dynamic Load Balancing**

Load balancing is a technique of conveying the absolute load to the individual nodes of the aggregate system to the encourage networks and resources to improve the reaction season of the work with greatest throughput in the system. The significant things which said about load balancing are assessment of load, load correlation, distinctive system soundness, system execution, connection between the nodes, nature of work to be moved, choosing of nodes and numerous different ones to consider while growing such calculation. Nearby cloud computing, the fundamental target of load balancing techniques is to improve execution of computing in the cloud, reinforcement plan if there should arise an occurrence of system disappointment, keep up security and versatility for obliging an expansion in huge scope computing, diminishes related expenses and reaction time for working in the cloud and furthermore expands the accessibility of resources.

**Proposed Methodology**

To accomplish the objective of expanding and advancing the utilization of each asset, the adaptive load balancing calculation (ALBA) is introduced in this investigation. This calculation utilizes astute specialists for tracking load on virtual machines and for balancing load in and across different data centers. The ALBA expects to improve the effectiveness of the cloud environment. It involves numerous classes as illustrated in figure 1.

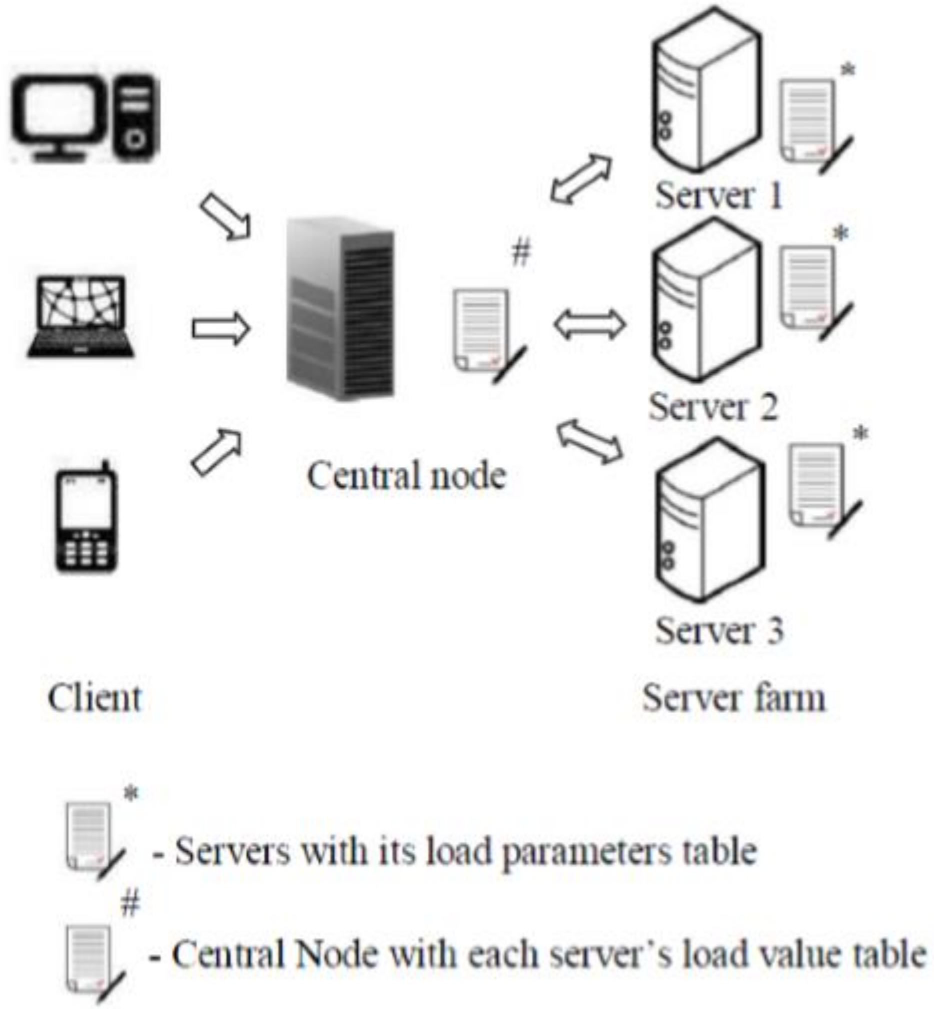

Server 1
Server 2
Server 3
Central node
Client
Server farm
#
*
- Servers with its load parameters table
- Central Node with each server's load value table

**Figure 3. Architecture of adaptive load balancing algorithm**

**Client Role**

Request the server for a page or file or process.

**Central Node Role**

It maintain the load table which has the data about the load estimation of workers and its load boundaries. Focal hub gets the customer demand searches up for the most un-loaded worker in load table and allots the separate worker to deal with the solicitation.

**Server Role**

Its primary role is to handle the solicitation. Aside from it likewise keeps a load boundary table which has the data of the nearby machine for example worker. It refreshes the data in certain cycle and trade to the focal hub. In the event that the load estimation of worker is high, it doesn't refresh since it's overloaded. So it holds up till it returns to normal.

**ALBA Architecture and Working**

Significant level perspective on ALBA is introduced in Figure 2 given underneath. Proposed structure presents canny specialists at two levels in cloud computing model. One at the data center level and other at the worldwide level. Each data center involves numerous virtual machines at various actual machines.

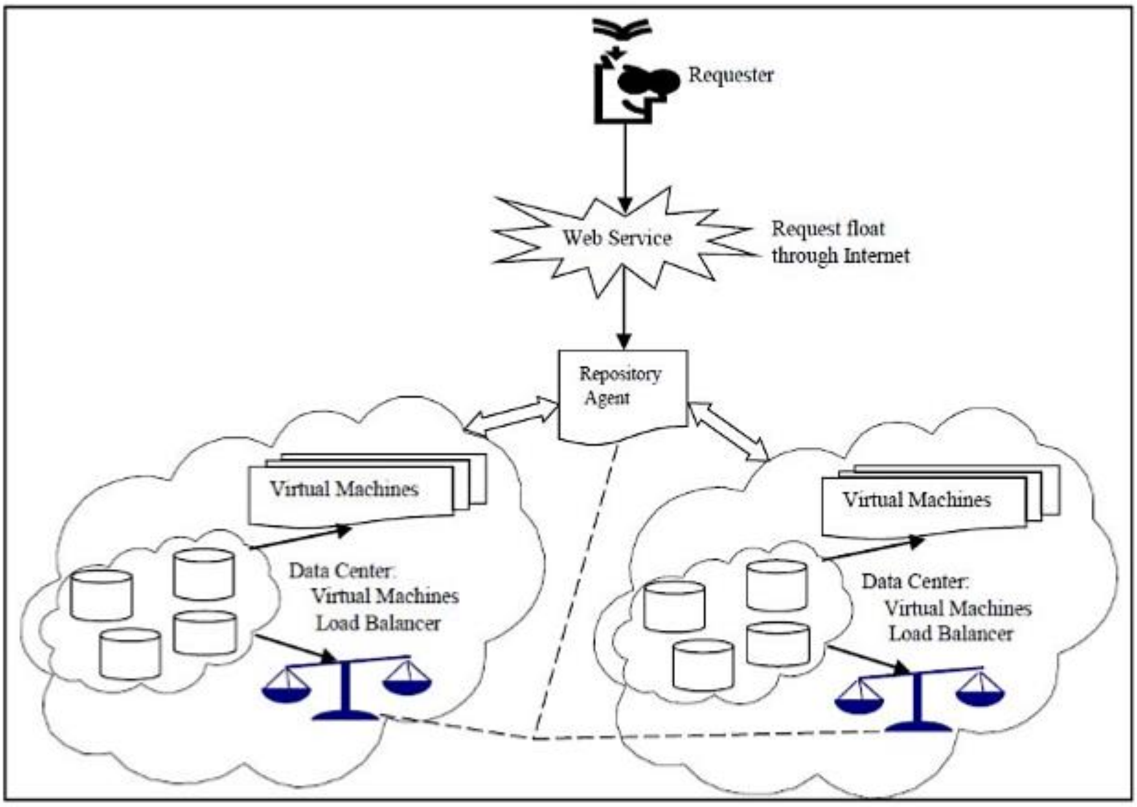


**Figure 4. ALBA Architecture**

Virtual machine load balancer agent is answerable for watching out for status of each virtual machine as far as its load. It keeps up data like resources accessible on virtual machines, reaction time and their line lengths.

Vault Agent is the one working at worldwide level, it keeps the record of all accessible virtual machines in a data center. It works together with virtual machine load balancer agent of different data centers.

VMLBA keeps a log document keeping the record of present place of employment executing on a VM and status of recently executed tasks. This log record causes it to compute normal holding up time and throughput of a VM. This data will additionally assist RA with knowing nature of all VMs. At whatever point, a client demand must be designated a few resources on a virtual machine, RA is been counseled which further checks with VMLBA to know present status of VM at a data center. VMLBA just shows nodes having load not exactly a pre-characterized limit esteem. Hence, in this path odds of a VM being overloaded will be limited in the system. In the event that the store agent finds mentioned resources from the mentioned data center, it apportions from that point just, in any case the storehouse agent will embrace the adaptability highlight of cloud and will found another appropriate data center with wanted resources. For this situation the embraced data center ought to have the base data transfer time.

**Table 1 Parameter Values**

| **Sr. No** | **Name of the parameter** | **Value** |
|---|---|---|
| 1 | Number of VM of a Data Center | 15 |
| 2 | VM Memory | 1 GB |
| 3 | Bandwidth | 1000000 Mb |

Step1: CPU scheduler set a time slice (t) and pick a process from ready queue for dispatching.
Step 2: If the burst time < t then
Step 3: CPU becomes free after execution and proceed further for next process in ready queue.
Step 4: Else
Step 5: after t time process is interrupted and taken out of CPU.
Step 6: Executed process will apply context switching and placed at end of the circular queue
Step 7: CPU scheduler will execute next process from the ready queue.

**Input Parameter**

Distance between Consumer and data centers (d ui): It is the distance between the location of the consumer and location of the data center. The delay time d ui is calculated using

$$d\_ui = |u- m\_i|$$

Where, u and m i are the location of consumer and i is the location of data center. ∂ is the network delay weight of the message requested which travels along the path between consumer and data center.

Workload on data centers (w i): This specifies workload on each datacenter. Workload on datacenter is given by total no. of virtual CPU present. The physical resources are divided among virtual CPU of VMs, which are considered as logical CPUs. Workload is allocated to corresponding threads of physical CPUs in the data center.

$$w\_i = \frac{\sum_{j=1}^{n} \alpha_{ij}}{p\_i} x\Gamma$$

$\alpha_{ij}$ is the total no. of virtual CPU present Γis the data load p_i is the total available no .of physical CPU threads i is the data center.

Power usage effectiveness (p): It defines computer efficiency and its data center usage power. It measures the capacity of each datacenter, by calculating the power used by the computing equipment. The power usage at each of the data center, is calculated by using the formula as follows

$$p= \frac{(Total\ Facility\ Powe}{(Total\ Equipment\ Pow}$$

Estimate the allocation delay time (E i): Delay time is time waiting for entering datacenter. It is given by

*E_i=w_i+d_ui*

w_i is the workload on data centeri and geographical distance is given by d_ui

**Output Parameter**

Throughput: This estimates the total number of tasks and complete execution of the given tasks successfully. System performance depends on high throughput, which is gained by execution of all tasks completely. Let T be the throughput, total task submitted be m and no of tasks executed be n. therefore throughput is given by T as

*T= n/m*

Response Time: It is the time interval between request sending and response received

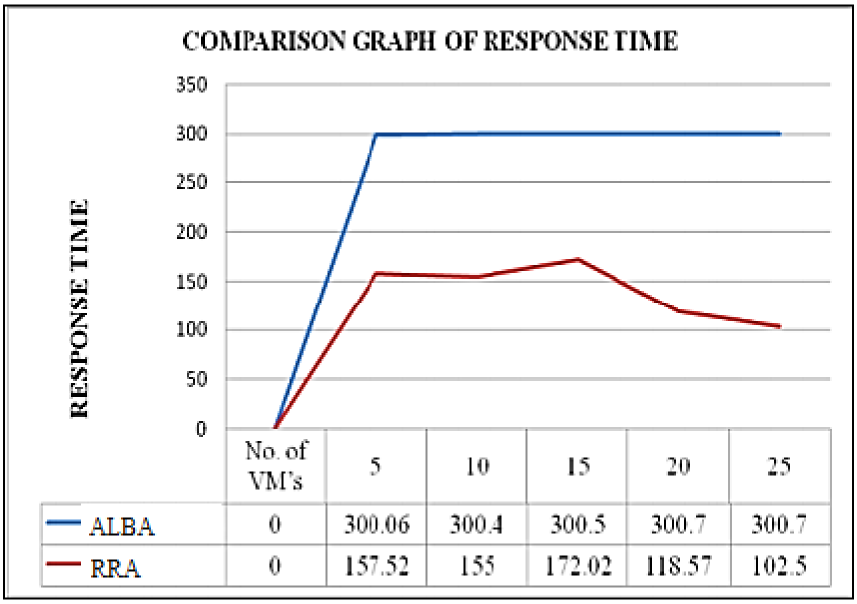
*Response Time = F_t– A_t + T_D*

**Results**

Proposed algorithm is implemented with the help of simulation package Cloudsim tool .Java language is used for implementing VM load balancing algorithm. We assume that the cloudsim toolkit has been deployed in one data centre having 5 virtual machines where the parameter values are as under.

Figure 5. Depicts that proposed approach with map reduce has a higher response time as compared to round robin, asthe no. VMs in the data centers increases there is a significant improvement in the response time of proposed approach.

Our proposed technique processes user requests in an efficient way as compared to the traditional approach. The results showed that the proposed approach is capable of obtaining near optimal solutions leading to significant improvement in throughput and response time.



| No. of VM's | 0 | 5 | 10 | 15 | 20 | 25 |
|---|---|---|---|---|---|---|
| ALBA | 0 | 300.06 | 300.4 | 300.5 | 300.7 | 300.7 |
| RRA | 0 | 157.52 | 155 | 172.02 | 118.57 | 102.5 |

**Figure 5 Comparison of Response Time**

## Conclusion

This paper proposed an agent-based adaptive load balancing calculation to dispense the cloud resources to different clients considering load balancing. The proposed model is effectively reenacted in cloud sim utilizing java language and contrasted the exhibition and RRA. In the trial, the adaptive load balancing calculation shows a better than the cooperative calculation as far as reaction time and throughput. Further, future work includes the versatility investigation of this calculation.

## REFERENCES

1. Aldinucci, Marco, Massimo Torquati, and MassimilianoMeneghin. ”FastFlow: Efficient parallel streaming applications on multi-core.” arXiv preprint arXiv:0909.1187 (2009).
2. Torquati, Massimo. ”Single-producer/single-consumer queues on shared cache multi-core systems.” arXiv preprint arXiv:1012.1824 (2010).
3. Vicente, Elder, and R. Matias. ”Exploratory study on the linuxos jitter.” In Computing System Engineering (SBESC), 2012 Brazilian Symposium on, pp. 19-24. IEEE, 2012.
4. Treibig, Jan, Georg Hager, and Gerhard Wellein. ”LIKWID: Lightweight Performance Tools.” In Competence in High Performance Computing 2010, pp. 165-175. Springer Berlin Heidelberg, 2012.
5. Goli, Mehdi, John McCall, Christopher Brown, Vladimir Janjic, and Kevin Hammond. ”Mapping parallel programs to heterogeneous CPU/GPU architectures using a montecarlo tree search.” In Evolutionary Computation (CEC), 2013 IEEE Congress on, pp. 2932-2939. IEEE, 2013.
6. Niethammer, Christoph, Colin W. Glass, and Jos´eGracia. ”Avoiding serialization effects in data/dependency aware task parallel algorithms for spatial decomposition.” In Parallel and Distributed Processing with Applications (ISPA), 2012 IEEE 10th International Symposium on, pp. 743-748. IEEE, 2012
7. Rossbory, Michael, and Werner Reisner. ”Parallelization of Algorithms for Linear Discrete Optimization Using ParaPhrase.” In DEXA Workshops, pp. 241-245. 2013.
8. Thomadakis, Michael E. ”The architecture of the Nehalem processor and Nehalem-EP SMP platforms.” Resource 3 (2011): 2.
9. Brown, Christopher, Vladimir Janjic, Kevin Hammond, HolgerSchoner, Kamran Idrees, and Colin Glass. ”Agricultural reform: more efficient farming using advanced parallel refactoring tools.” In Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on, pp. 36-43. IEEE, 2014.
10. T.MalarethinamKokilavani and Dr. D.I. George A, Load Balanced MinMin Algorithm for Static Meta-Task Scheduling in Grid Computing International Journal of Computer Applications ,Volume 20 No.2, April 2011. HarshithGupta,KalicharanSahu ,Honey Bee Behavior Based Load Balancing of Tasks in Cloud Computing ,IJSR,2012.
11. Vengatesan, K., Kumar, A., Subandh, T., Vincent, R., Sayyad, S., Singhal, A., & Wani, S. (2019). Secure Data Transmission Through Steganography with Blowfish Algorithm. In International Conference on Emerging Current Trends in Computing and Expert Technology (pp. 568–575)
12. N. S. Raghava and Deepti Singh, Comparative Study on Load Balancing Techniques in Cloud Computing Open Journal of Mobile computing and Cloud computing, volume 1, August 2014
13. Sayyad, S., Mohammed, A., Shaga, V., Kumar, A., & Vengatesan, K. (2018). Digital Marketing Framework Strategies Through Big Data. In International conference on Computer Networks, Big data and IoT (pp. 1065–1073).
14. Ajay Gulati and Ranjeev. K. Chopra, Dynamic Round Robin for Load Balancing in a Cloud Computing, IJCSMC, Vol. 2,, June 2013
15. Siva ThejaMaguluri and R. Srikant , Lei Ying ,Heavy Traffic Optimal Resource Allocation Algorithms for Cloud Computing Clusters, June 2012