Latency Optimization in Logistics Information Systems Using a Distributed Load Balancing Algorithm

Rajesh Kumar Sharma
Department of Computer Science and Engineering
Faculty of Engineering and Technology, University of Delhi
New Delhi, India

Abstract
Cloud computing has become an important platform for storing, processing, and distributing information across geographically distributed data centers. In logistics information systems, cloud-based services enable logistics providers, partners, and users to access resources such as execution time, bandwidth, storage, memory, and computing capacity without managing the physical infrastructure directly. However, as the number of logistics partners and users increases, real-time information sharing becomes more difficult because of network congestion, uneven workload distribution, higher response time, and increased service cost. This paper presents a Distributed Service Broker Policy (DSBP) algorithm for improving load balancing and reducing latency in cloud-based logistics systems. The proposed approach considers job size, expected processing time, network latency, available bandwidth, and data center capacity when selecting the most suitable data center for a user request. The algorithm is evaluated using the Cloud Analyst simulator and compared with existing load balancing approaches, including Round Robin, Active Monitoring, and Throttled algorithms. The simulation results show that the proposed DSBP algorithm improves overall response time, reduces data center processing time, and lowers total cost compared with the existing methods. The findings indicate that a distributed load balancing approach can significantly enhance the performance, availability, and cost efficiency of logistics information systems operating in cloud environments.

Keywords: Cloud computing, Logistics information system, Distributed load balancing, Latency optimization, DSBP

1. Introduction

A data center is a centralized repository used for the storage, processing, management, and distribution of information. Cloud computing extends this concept by allowing users to access computing resources and data center services from virtually any location in the world. In a cloud environment, logistics providers and users can obtain resources such as execution time, response time, storage, memory, and cost-efficient computing capacity on demand. They are not required to know the physical location of the data center or to operate and maintain the underlying infrastructure. Instead, they only need to connect to the cloud platform and use the required applications to support their logistics operations.

In logistics information systems, the exchange of information between suppliers, service providers, and users is a critical requirement. As the number of logistics users and internet-based requests continues to grow, data centers may experience traffic overload, increased latency, and reduced availability. Although several load balancing algorithms have been proposed for cloud computing environments, many of them still face limitations in reducing response time, processing delay, and overall cost simultaneously. Therefore, cloud-based distributed load balancing algorithms can provide an effective solution for handling multiple requests, improving service availability, and supporting real-time information sharing among logistics partners. This paper evaluates existing load balancing algorithms and proposes a distributed load balancing approach using the Cloud Analyst simulator. The performance of the proposed algorithm is then compared with existing algorithms in terms of response time, data center processing time, and cost.

2. Related Works

Load balancing is a technique used to distribute workloads across multiple computing resources, servers, virtual machines, or network links in order to improve resource utilization, reduce response time, and prevent system overload. In cloud computing, effective load balancing ensures that user requests are processed by suitable data centers or virtual machines with minimum delay and acceptable cost. This is especially important in logistics information systems, where real-time information exchange is required among distributed users and service providers.

As logistics information systems increasingly rely on cloud-based services, distributed data centers must be able to handle a growing number of user requests efficiently. Several studies have proposed load balancing strategies to improve performance in cloud environments. Sharma et al. (2012) proposed a Central Load Balancing Policy for Virtual Machines (CLBVM) to balance workloads in distributed environments, although fault tolerance was not fully considered. James and Verma (2012) examined virtual machine load balancing for Infrastructure as a Service (IaaS) and implemented a weighted active monitoring algorithm to reduce response time and processing time using CloudSim. Limbani and Oza (2012) proposed an extended service proximity-based routing policy for cost-effective data center selection using Cloud Analyst.

Bhargava et al. (2013) analyzed the Round Robin algorithm across different data centers and evaluated its impact on overall response time. Priya and Subramani (2013) proposed a load balancing method that combined active monitoring with resource-aware scheduling to improve cloud performance by selecting efficient virtual machines and reducing task execution time. Singh and Gangwar (2014) presented a comparative study of load balancing algorithms based on parameters such as response time and cost. Mishra and Bhukya (2014) proposed priority-based and extended priority-based service broker algorithms that distribute requests according to data center ratings.

Kapgate (2014) introduced a new service broker algorithm for data center selection and showed that it could reduce service response time. Nayak and Patel (2015) compared an improved throttled algorithm with the existing throttled load balancing algorithm in terms of response time, data center service request time, and cost using Cloud Analyst. Panwar and Mallick (2015) reviewed various load balancing algorithms and compared Round Robin and Throttled scheduling algorithms using response time and data processing time as performance measures.

Bodke (2016) compared several load balancing policies, including Throttled, Round Robin, First Come First Serve (FCFS), and load equalization algorithms. The study showed that response time could be reduced, although cost was not fully considered. Khanchi and Tyagi (2016) proposed a hybrid approach for virtual machine-level load balancing to distribute workloads among available virtual machines and reduce overall response time and processing time.

Building on these studies, this paper proposes a Distributed Service Broker Policy (DSBP) algorithm designed to select the most suitable data center by considering latency, available bandwidth, job size, and processing requirements. The proposed DSBP algorithm is developed as an improvement over the throttled load balancing approach and aims to achieve lower response time and reduced cost for cloud-based logistics information systems.

3. Problems in Existing Algorithms

In distributed data center environments, service brokers are responsible for directing user requests to the most appropriate data center. The selection process should consider several performance factors, including response time, processing time, network latency, available bandwidth, and cost. Existing service broker and load balancing policies, such as network latency-based routing, service proximity-based routing, Round Robin, Active Monitoring, and Throttled algorithms, have improved certain aspects of cloud performance. However, many of these methods do not optimize response time and cost at the same time.

Previous studies show that some algorithms are effective in reducing response time but may increase overall cost, while others reduce cost but do not perform well when the number of data centers or user requests increases. For example, modified Throttled and Round Robin-based algorithms have been shown to improve response time, but they may not sufficiently address data transfer cost or processing delay. Similarly, priority-based service broker algorithms can improve data center selection, but cost and latency optimization remain important challenges.

Another limitation of several existing algorithms is that they do not fully account for the dynamic nature of cloud-based logistics environments. Logistics systems often involve geographically distributed users, real-time tracking requirements, variable request sizes, and time-sensitive information sharing. In such settings, selecting a data center only on the basis of proximity or availability may not be sufficient. A more effective algorithm should consider both the computational capacity of the data center and the network-related parameters that influence latency and cost.

Round Robin algorithms allocate requests sequentially, which makes them simple to implement but less responsive to differences in virtual machine capacity. Active Monitoring algorithms improve allocation by monitoring the current load, but they may still be limited when network

conditions vary. Throttled algorithms assign requests only to available virtual machines, but performance may decline when request volume increases or when data centers have different processing capacities.

The main problem addressed in this paper is therefore the need for a load balancing approach that can reduce latency, avoid overload, and minimize total cost in a distributed logistics cloud environment. The proposed DSBP algorithm attempts to address this gap by extending the throttled approach and incorporating job size, bandwidth, and latency into the data center selection process.

Table 1 summarizes the main algorithms reviewed in previous studies and highlights their reported performance in terms of response time and cost. The comparison shows that while many algorithms improve one performance dimension, there is still a need for an approach that balances response time, data center processing time, and cost together.

Table 1: Existing algorithm simulation results

Authors (Year)	**Algorithms**			**Results**	
	Round Robin Algorithm	Active Monitoring Algorithm	Throttled Load balancing Algorithm	Average Response Time (ms)	Total Cost ($)
Limabani & Oza (2012)	*	*	*	*	967.27
Jasmin James (2012)	✓	✓	✓	✓	694.82
Harvinder Singh (2013)	*	✓	✓	112.18	*
S. Mohana Priya (2013)	*	*	*	171.43	242.05
Rakesh kr. (2014)	*	*	*	98.18	144.52
Kunal Kishore (2014)	*	*	*	481.54	*

Ali Naser (2014)	*	*	✓	139.98	59.50
Sunny Nandwani (2015)	✓	✓	✓	92.10	*
Slesha (2015)	*	*	✓	50.1004	1.324
Reena Panwar (2015)	✓	*	✓	101.61	50.27
Sarika Vasantrao Bodke (2016)	✓	*	✓	*	10.10
Mamta Khanchi (2016)	✓	*	✓	92.45	*
Simar Preet Singh (2016)	✓	*	✓	57.68	*
Er. Pooja (2016)	*	*	*	57.08	12.85
Pradeep Singh (2016)	✓	*	*	484.25	*
Saurabh Shukla (2016)	✓	*	✓	150.87	*

Based on the limitations identified in the reviewed algorithms, the DSBP algorithm is proposed as an enhancement of the throttled load balancing algorithm in a virtual machine cloud environment. The objective is to achieve better response time, lower processing time, and reduced cost for logistics information systems.

4. Implementation of Existing and Proposed Algorithms

The proposed service broker policy selects the data center by considering job size, expected processing time, network latency, and available bandwidth. These factors are used to minimize the overall response time and processing time. The routing policy also considers request size, number of users, number of virtual machines, number of data centers, available bandwidth, response time, and data transfer cost. Processing time depends on data center specifications such as RAM, CPU capacity, and virtual machine configuration. The DSBP algorithm is implemented as a modification of the throttled load balancing algorithm.

The proposed policy uses real-time values to estimate processing time and to reduce the time required by the broker to make forwarding decisions. Since the service broker does not analyze the internal complexity of each job, the size of the job is used as an indicator of the required processing time. Larger jobs are assumed to require more processing time, while smaller jobs are expected to be processed more quickly. This approach allows the broker to make practical and efficient decisions in distributed cloud environments. Table 2 presents the implemented methods of the existing algorithms and the proposed DSBP algorithm.

Table 2: Methods of implemented load balancing algorithms

Algorithm	**Round Robin**	**Active Monitoring**	**Throttled Load Balancing**	**DSBP Proposed Algorithm**
Method	Allocates requests sequentially to available virtual machines in a cyclic order. It is simple and easy to implement but does not consider current workload or different VM capacities.	Monitors current allocation counts and assigns new requests to the virtual machine with the lowest current load. It improves load distribution but may not fully consider network latency or cost.	Assigns requests only to available virtual machines. If no VM is available, the request waits until a VM becomes free. It prevents over-allocation but may increase delay under high traffic.	Extends the throttled approach by considering job size, expected processing time, network latency, and available bandwidth before selecting a data center or VM. It aims to reduce response time, processing time, and total cost.

5. Experimental Results and Performance Analysis

The performance of the proposed DSBP algorithm was evaluated through simulation using Cloud Analyst. The simulator classes were extended to incorporate the proposed DSBP algorithm. The evaluation compared DSBP with Round Robin, Active Monitoring, and Throttled load balancing algorithms. The main performance measures were overall response time, data center processing time, and total cost in a distributed data center environment.

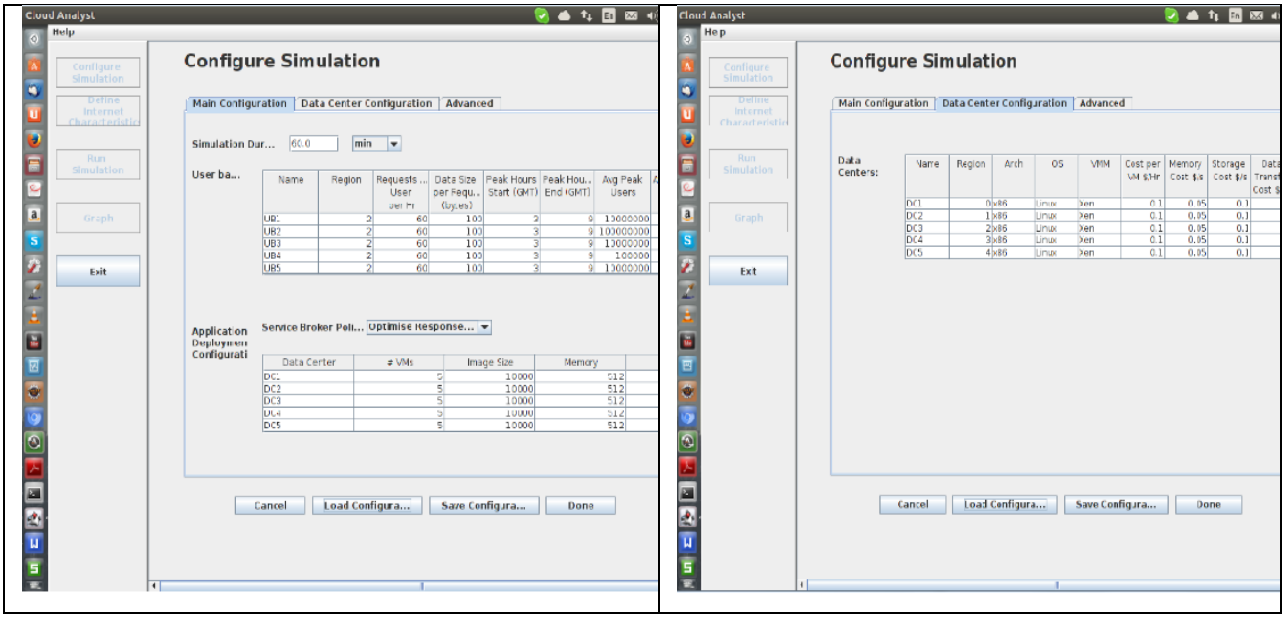

Figure (a) Figure (b)

Figure 1(a): Main configuration screen of User Base and application deployment. Figure 1(b): Main configuration screen of Data Center

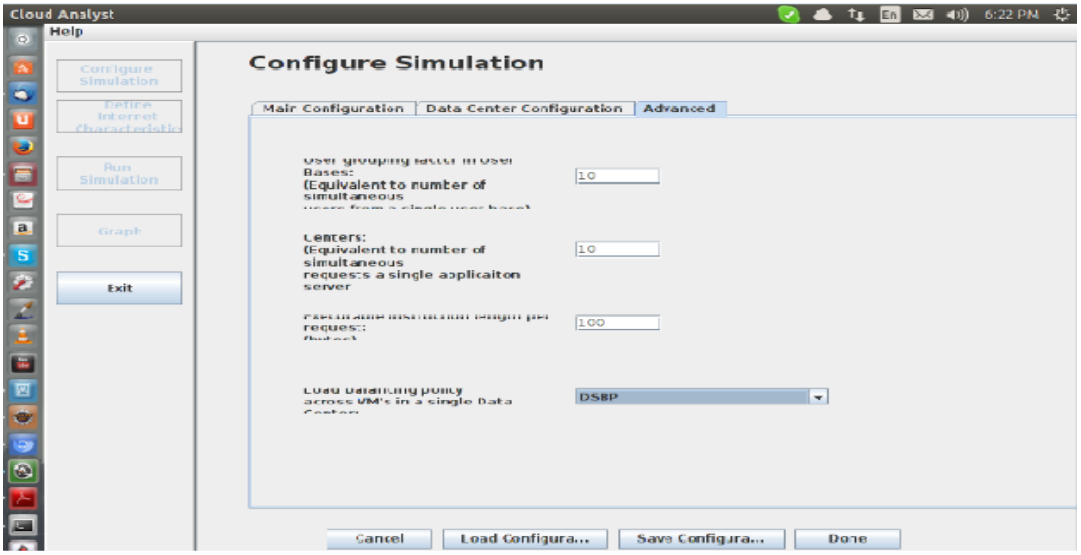

Figure 2: Configuration screen for the proposed Distributed Service Broker Policy algorithm

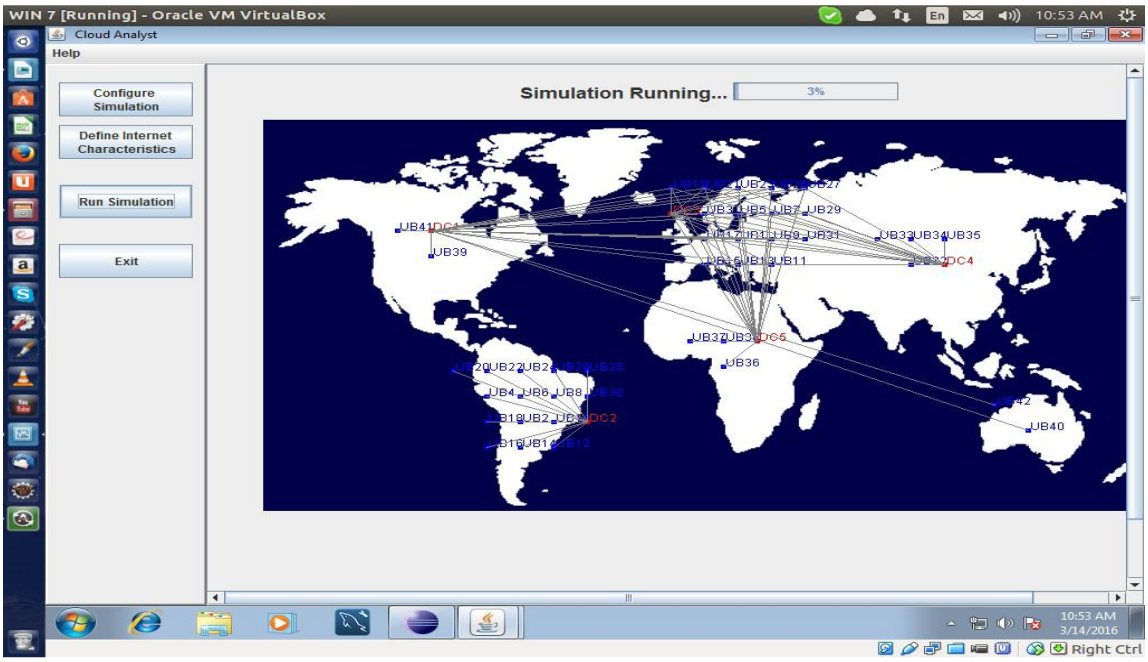

Figure 3: Simulation running screen of distributed data center

The simulation was carried out for a period of 60 hours using different numbers of users and six data centers, labeled DC1 to DC6. Each data center contained five virtual machines. Other simulation parameters were kept constant, as shown in Table 3.

Table 3: Parameters and values

Parameters	Values
VM image size	10000
VM Memory	1024 MB
VM Bandwidth	1000
Data Center (DC)	6
Virtual Machine (VM) in each data center	5
DC Memory per machine	1000 MBPS
DC storage per machine	1GB

DC available bandwidth per machine	1000000

After running the experiments successfully in Cloud Analyst, the overall response time and data center processing time of the different load balancing algorithms were obtained. Table 4 presents the comparison of Round Robin, Active Monitoring, Throttled, and DSBP algorithms.

Table 4: Comparison of overall response time and data center processing time summary

Distributed Data Center Over All Response Time (RT)												
Algorithm	Round Robin			Active Monitoring			Throttled			DSBP		
Metric	Avg(ms)	Min(ms)	Max(ms)	Avg(ms)	Min(ms)	Max(ms)	Avg(ms)	Min(ms)	Max(ms)	Avg(ms)	Min(ms)	Max(ms)
Overall RT	1188.11	14.58	5780.21	117.17	10.50	4933.89	185.50	14.62	5780.05	58.52	0.00	65.00
DC Processing Time	960.43	0.00	36061.25	520.79	0.00	30765.56	946.23	0.01	36059.00	6.68	0.00	100.56

Table 5: Comparison of total cost summary

Distributed Data Center Overall Cost Summary				
Cost	**Round Robin Algorithm**	**Active Monitoring Algorithm**	**Throttled Algorithm**	**DSBP Algorithm**
Total VM cost ($)	0.04	0.42	0.42	0.40
Total data transfer cost ($)	0.85	0.85	0.85	0.08
Grand total ($)	1.25	1.25	1.27	0.48

The simulation results indicate that the proposed DSBP algorithm performs better than the existing algorithms in terms of overall response time and total cost. The average response time of DSBP is lower than Round Robin, Active Monitoring, and Throttled algorithms. In addition, the total cost is reduced from more than one dollar in the existing algorithms to 0.48 dollars in the proposed DSBP algorithm. These results show that considering latency, bandwidth, and job size during data center selection can significantly improve the performance of cloud-based logistics information systems.

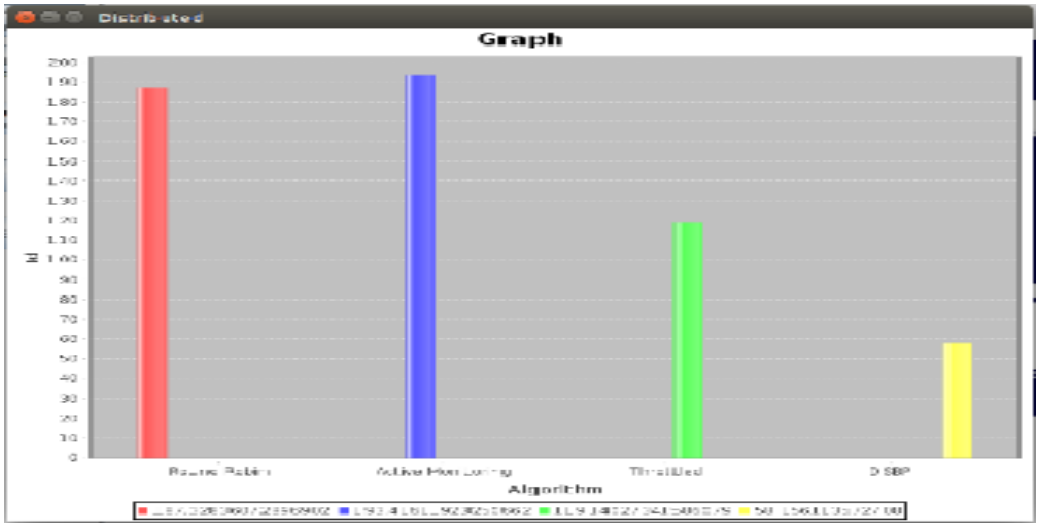

Figure 4: Simulation graph of distributed data center load balancing

6. Conclusion

This paper proposed and evaluated a Distributed Service Broker Policy (DSBP) algorithm for reducing latency and improving load balancing in cloud-based logistics information systems. The algorithm was designed to select the most suitable data center by considering job size, processing requirements, network latency, and available bandwidth. The simulation results obtained through Cloud Analyst show that DSBP outperforms Round Robin, Active Monitoring, and Throttled algorithms in terms of overall response time, data center processing time, and total cost. The proposed algorithm therefore provides an effective solution for logistics environments that require real-time information sharing, high availability, and cost-efficient cloud service management. Future research can extend the proposed approach by considering fault tolerance, energy efficiency, security, and larger-scale real-world logistics datasets.

References

[1] Meenakshi Sharma, Pankaj Sharma, and Sandeep Sharma, Efficient Load Balancing Algorithm in VM Cloud Environment, International Journal of Computer Science, vol. 3, issue 1, March 2012, ISSN: 0976-8491.

[2] Jasmin James and Bhupendra Verma, Efficient VM Load Balancing Algorithm for a Cloud Computing Environment, International Journal on Computer Science and Engineering, vol. 9, issue 9, September 2012, ISSN: 0975-3397.

[3] Dhaval Limbani and Bhavesh Oza, A Proposed Service Broker Strategy in Cloud Analyst for Cost-Effective Data Center Selection, International Journal of Engineering Research and Applications, vol. 2, issue 1, pp. 793-797, February 2012.

[4] Neeraj Bhargava, Rita Bhargava et al., Performance Analysis of Cloud Computing for Distributed Clients, International Journal of Computer Science and Mobile Computing, vol. 2, issue 6, June 2013, ISSN: 2320-088X.

[5] S. Mohana Priya and B. Subramani, A New Approach for Load Balancing in Cloud Computing, International Journal of Engineering and Computer Science, vol. 2, issue 5, May 2013, ISSN: 2319-7242.

[6] Harvinder Singh and Rakesh Chandra Gangwar, Efficient Approach for Load Balancing in Virtual Cloud Computing Environment, An International Journal of Advanced Computer Technology, vol. 3, issue 10, October 2014.

[7] Rakesh Kr. Mishra and Sreenu Naik Bhukya, Service Broker Algorithm for Cloud Analyst, International Journal of Computer Science and Information Technologies, vol. 5, issue 3, 2014.

[8] Deepak Kapgate, Efficient Service Broker Algorithm for Data Center Selection in Cloud Computing, International Journal of Computer Science and Mobile Computing, vol. 3, issue 1, January 2014, ISSN: 2320-088X.

[9] Slesha Nayak and Prangesh Patel, Analytical Study of Throttled Algorithm of Load Balancing in Cloud Computing using Cloud Analyst, International Journal of Science, Technology and Engineering, vol. 1, issue 12, June 2015.

[10] Reena Panwar and Bhawna Mallick, Load Balancing in Cloud Computing using Dynamic Load Management Algorithm, Green Computing and Internet of Things (ICGCIoT), 8-10 October 2015, published by IEEE Xplore, 14 January 2016.

[11] Sarika Vasantrao Bodke, Cloud Computing Load Balancing Algorithm, International Journal of Advanced Engineering, Management and Science, vol. 2, issue 5, May 2016, ISSN: 2454-1311.

[12] Mamta Khanchi and Sanja Tyagi, An Efficient Algorithm for Load Balancing in Cloud Computing, International Journal of Engineering Science & Research Technology, vol. 5, issue 6, June 2016, ISSN: 2277-9655.