Reducing Latency Using a Distributed Load Balancing Algorithm in Logistics Systems

Rahul Sharma[1] and Anita Verma[2]
[1]Academy of Technical Education, Noida, India
[2]Ansal University, Gurgaun, India

Abstract
Cloud computing enables access to data centers from virtually any location around the world. It provides a wide range of resources, such as processing time, bandwidth, storage, memory, and cost-efficient services, that logistics providers and users can utilize on demand. With cloud computing, users are not required to know the physical location of data centers or manage the underlying infrastructure; instead, they only need to connect to the platform and use the applications necessary for their operations. Despite these advantages, many organizations still rely on their own local data centers for service delivery. In logistics information systems, efficient information sharing and data transfer among multiple partners through cloud service providers remains a significant challenge. Logistics users increasingly.

Keywords: Cloud Computing, Logistics information system, Load balancing algorithms, DSBP

1. Introduction

A data center is a centralized repository designed for the storage, management, and distribution of information. Cloud computing extends this concept by enabling access to data centers from virtually any location worldwide. Within such environments, numerous resources—such as execution time, response time, storage, memory, and cost-efficient services—are available for logistics providers and users to utilize on demand. Users are not required to know the physical location of the data center or manage its infrastructure; instead, they only need to connect to the system and use the applications necessary for their operations. However, some organizations still prefer to rely on their own local data centers for service delivery.

In logistics information systems, efficient information sharing and data transfer among multiple partners via cloud service providers remain significant challenges. Real-time information exchange is particularly critical for logistics users. Although centralized data centers offer efficient data transfer and service management, the rapid increase in the number of users has led to issues such as network congestion and traffic overload, which can negatively affect performance and availability.

To address these challenges, load balancing plays a crucial role in distributing network traffic and optimizing resource utilization. While various load balancing algorithms exist, many still face limitations in reducing response time, managing overload, and minimizing cost. Distributed load balancing algorithms in cloud-based logistics systems offer a promising solution to these issues by improving performance and reducing delays caused by simultaneous requests.

This paper evaluates existing load balancing algorithms and proposes a new distributed load balancing approach using the Cloud Analyst simulator. Furthermore, it compares the performance of the proposed algorithm with existing methods to demonstrate its effectiveness in enhancing system efficiency.

2. Related Work

Load balancing is a method or technique, which is used for distributing workload on the multiple computers cluster across the network links to achieve recourse utilization and overall time. Load balancing avoids lot of overloading on the resources and divides the traffic between servers and data, so the data can be sent and received without minimum delay.

For new transmission to survive in the competitive market logistic information system is using cloud based service for developing distributed data center. Load balancing helps to balance the increased user's requests on the distributed data center so it is important to use load balancing on web servers in logistics information system. Meenakshi Sharma (2012) proposed a central load balancing policy for virtual machine (CLBVM) to balance the load in distributed environment but it does not consider fault tolerant system. Bhupendra Verma (2012) analyzed virtual machine load balancing and proposed new virtual machine load balancing algorithm for IaaS framework, implemented of weighted Active Monitoring load balancing algorithm to achieve better performance in response time and data processing time by using Cloud Sim. Dhaval Limbani and Bhavesh Oza (2012) proposed the extended service proximity based routing policy algorithm for routing of user requests such that cost effective data center selection by using Cloud Analyst. Neeraj Bhargava, Rita Bhargava et.al (2013) analyzed and discussed round robin algorithm used by different data center and calculated the overall response time in better performance. S. Mohana Priya and B. Subramani (2013) proposed algorithm by using active monitoring load balancing algorithm and resource aware scheduling algorithm for high performance in cloud systems. The experiment result of proposed algorithm is the efficient virtual machine is selected for process and minimum execution time of task.

It increases the performance and reduces the response time and cost. S. Mohana Priya and B. Subramani (2013) proposed algorithm by using active monitoring load balancing algorithm and resource aware scheduling algorithm for improving resource utilization and scheduled load balancing for high performance in cloud systems. The experiment result of proposed algorithm is the efficient virtual machine is selected for process and minimum execution time of task, it increases the performance and reduces the response time and cost. Harvinder Singh and Rakesh Chandra Gangwar (2014) presented comparative study of load balancing algorithms based on the parameters like; response time and cost which are reduced by round robin and active monitoring load balancing algorithm. Rakesh kr. Mishra and Sreenu Naik Bhukya (2014) proposed priority and extended priority based round robin service broker algorithms which distribute the requests based on the rating of data centers and gives better performance.

Deepak Kapgate (2014) proposed and compared the new service broker (DC selection) algorithm with existing service broker algorithm. The proposed algorithm reduced service response time

and improved performance of data center. Slesha Nayak and Prangesh Patel (2015) presented comparative study for existing throttled algorithms and proposed throttled algorithm of load balancing in cloud computing. Both are tested and compared in terms of response time, data center service request time and cost by using Cloud Analyst.

Reena Panwar and Bhawna Mallick (2015) discussed the various load balancing algorithms and compared them based on parameters like; data processing time and response time etc. by using round robin and throttled scheduling algorithms in Cloud Analyst. Sarika Vasantrao Bodke (2016) presented a comparison of assorted policies which is used for load reconciliation by using throttled, round robin and first come first serve (FCFS) and load aqualization algorithms. This comparison shows that response time was effectively reduced but they did not consider the cost. Mamta Khanchi and Sanja tyagi (2016) proposed and implemented a hybrid approach for virtual machine level load balancing. This algorithm distribute workload among the virtual machines that are available in data center at the same time to minimize the overall response time and data center processing time.

In this paper we propose a distributed service broker policy algorithm (DSBP) with the best possible response time, delay and minimum cost in selecting the most suitable data center. DSBP is the mainly implementation of throttled algorithm for taking bandwidth, latency and size of job which achieves minimum response time and minimum cost.

3. Problems in Existing Algorithms

In distributed data center, the main propose of service brokers is to direct the user request to the best data center with better performance because service broker policy has to efficiently select the best data center for the job considering many issues like response time, service time and cost. For directing the user request to the best data center there are so many load balancing policy algorithms like network latency based, service proximity based routing, etc. (Sharma, 2012) modified the throttled load balancing algorithm and proposed virtual machine load balancing algorithm which reduce response time effectively but not reduce cost. (Bhargava, 2013) analyzed round robin algorithm by using different data center and user base (UB).

This result shows that response time is to be minimize but cost should need to be reduce also. (Priya, 2013) proposed a new load balancing algorithm for virtual machine, in which virtual

machine is selected for process and minimum execution time of task. It reduces the response time and total cost but there is some issues to improve the response time efficiency parallel to the cost performance. (Harvinder, 2014) analyzed the comparison of three existing load balancing algorithm; round robin, active monitoring, throttled algorithm. The experimental result of theses algorithms shows the reduction in response but increment in cost.

(Rakesh, 2014) . proposed priority and extended priority based round robin service broker algorithms which distribute the requests based on the rating of data centers and gives better performance in response time but cost is not reduces similar to response time. (Deepak Kapgate, 2014) proposed and compared the new service broker (DC selection) algorithm with existing service broker algorithm. The proposed algorithm reduced service response time but cost is not considered.

(Slesha, 2015) proposed throttled algorithm and compared it with existing throttled algorithm. The proposed algorithm reduces response time and cost in better performance but if data center will be increased then response time cost will be increased. (Reena ,2015) discussed the various load balancing algorithms and compared them based on parameters like; data processing time and response time etc. by using round robin and throttled scheduling algorithms in Cloud Analyst. The result of both algorithms shows reduction in response time and cost but if data centers will be increased then response time and cost will increase by using separately round robin and throttled algorithm.

(Bodke, 2016) also presented round robin algorithm. In this algorithm the time slits into multiple slices and every node is given a specific time slice. There are multiple requests in multiple processes. Each process is given time slot. If user request completes among time then user must not wait otherwise user need to watch for its next time slot. This may create slow process and overloading.

(Mamta, 2016) proposed virtual machine load balancing algorithm which is a combination of round robin, throttled, (ESCE) equally spread current execution and hybrid algorithm. The result shows minimization of overall response time but overall performance is not much improved by using ESCE algorithm. In the below table (Tabel.1) we can see so many highlights of different algorithms and drawbacks. we analyze the results of various algorithms in terms of response time and cost to identify the improvement of round robin, active monitoring and throttled algorithms.

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	*

A new DSBP algorithm has been proposed from modifying the throttled load balancing algorithm in Virtual Machine environment of cloud computing in order to achieve better response time, processing time and cost.

4. Implementation of Existing and Proposed Algorithms

The proposed service broker policy selects the data center based on the job size, the expected processing time, the network latency and the available bandwidth to minimize the overall response and processing time. The routing policy considers different factors like; request size, user size, number of virtual machines, number of data centers, available bandwidth, response time, data transfer cost etc. The processing time is based on the data center specification such as Ram, CPU and VM configuration. In the below table we have implementation the method of round robin, active monitoring and throttled load balancing algorithm. DSBP is the modification of throttled load balancing algorithm.

As a result, the proposed policy accommodates the current needs by taking real-time values to calculate the processing time to minimize the time needed to make the forwarding decision by the broker. Note that the job processing time can vary depending on the computational task to be performed. For instance, a smaller job requires less processing time if there was no I/O operation involved. However, since it is not the service broker functionality to analyze the jobs and

determine their complexity, we considered the size of the job as an indication to the needed processing time with a positive relation between them. Here we have all the implemented methods of some existing algorithms. In the below table we can see all the methods of existing and proposed DSBP algorithms. The proposed algorithm is used to improve throttled algorithm. This improved throttled algorithm works well even though underlying capacity of each virtual machines are different because the hardware configuration of virtual machines are different.

Tabel.2: Methods of Implemented Load Balancing Algorithms

Algorithms	**Round Robin**	**Active Monitoring**	**Throttled load balancing**	**DSBP Proposed Algorithm**
Methods	**public int** getNextAvailableVm() { currVm++; **if** (currVm >= vmStatesList.size()) { currVm = 0; } allocatedVm(currV	**if** (currentAllocationCounts.size() < vmStatesList.size()){ **for** (**int** availableVmId : vmStatesList.keySet()){ **if** (!currentAllocationCounts.containsKey(availableVmId)){ vmId = availableVmId; **break**; } } } **else** { **int** currCount; **int** minCount = Integer.**MAX_VALUE**; **for** (**int** thisVmId :	**public int** getNextAvailableVm() { **int** vmId = -1; **if** (vmStatesList.size() > 0){ **int** temp; **for** (Iterator<Integer> itr = vmStatesList.keySet().iterator(); itr.hasNext();){ temp = itr.next(); VirtualMachineState state = vmStatesList.get(temp); //System.out.println(temp + " state is " + state	**public int** getNextAvailableVm(){ **int** vmId = -1; **if** (currentAllocationCounts.size() < vmStatesList.size()+1){ **for** (**int** availableVmId : vmStatesList.keySet()){ **if** (!currentAllocationCounts.containsKey(availableVmId)){ vmId = availableVmId; **break**; } } } **else** { **int** currCount;

	m);
	return currVm;
	}
		currentAllocationCounts.keySet()){
		currCount = currentAllocationCounts.get(thisVmId);
		if (currCount < minCount){
		minCount = currCount;
		vmId = thisVmId;
		}
		}
		}
		allocatedVm(vmId);
		return vmId;
			+ " total vms " + vmStatesList.size());
			if (state.equals(VirtualMachineState.**AVAILABLE**)){
			vmId = temp;
			break;
			}
			}
			}
			allocatedVm(vmId);
			return vmId;
			}
				int minCount = Integer.**MAX_VALUE**;
				for (**int** thisVmId : currentAllocationCounts.keySet()){
				currCount = currentAllocationCounts.get(thisVmId);
				if (currCount < minCount){
				minCount = currCount;
				vmId = thisVmId;
				}
				}
				}
				allocatedVm(vmId);
				return vmId;
				}

5. Experimental Results and Performance Analysis

The performance of the proposed DSBP algorithm has been analyzed based on the result of simulation done in Cloud Analyst. The classes of Cloud Analyst simulator have been extended to utilize newly proposed DSBP algorithm. In the below illustrations, the response time and load are analyzed by using methods of round robin, active monitoring, throttled and DSBP algorithm for distributed data center.

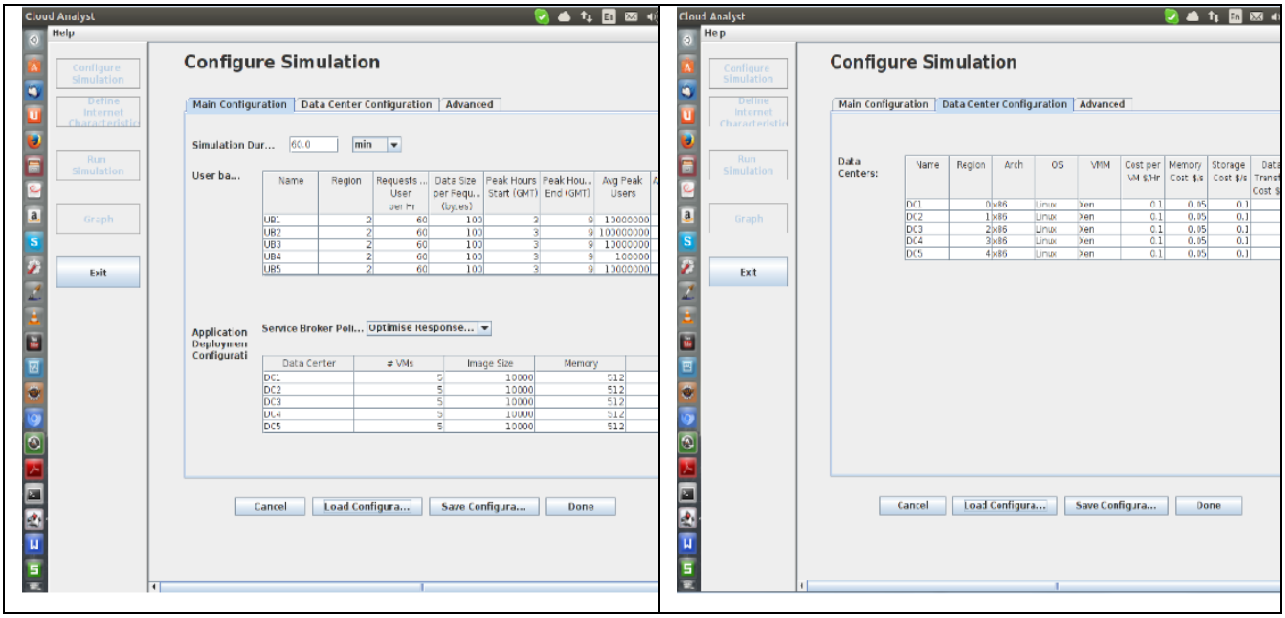

Figure (a) Figure (b)

Figure.1(a): Main Configuration Screen of UserBase and Application Deployment. (b) Main Configuration Screen of Data Center

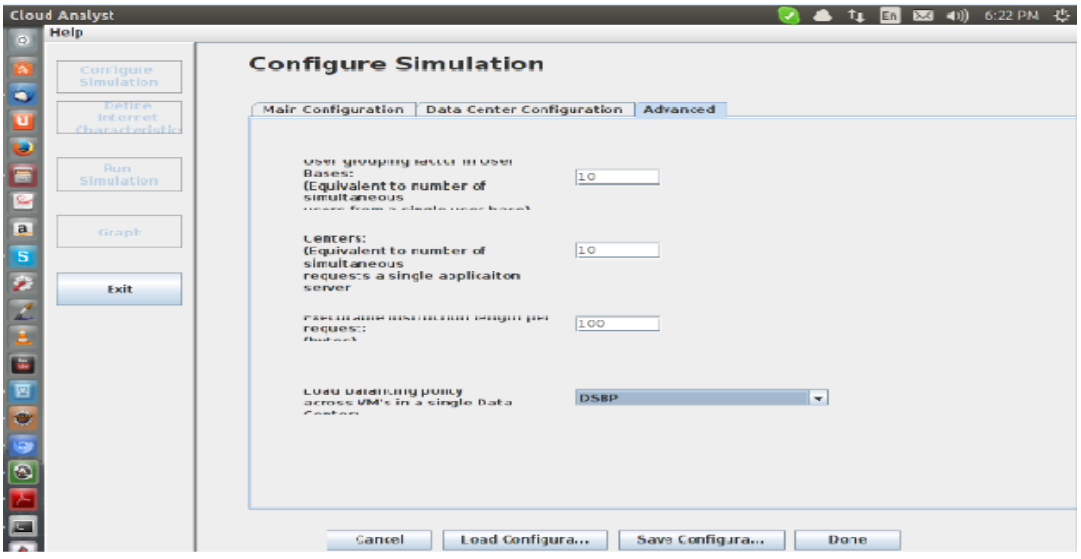

Figure.2: Configure Simulation Screen for proposed Distributed Service Broker Policy Algorithm

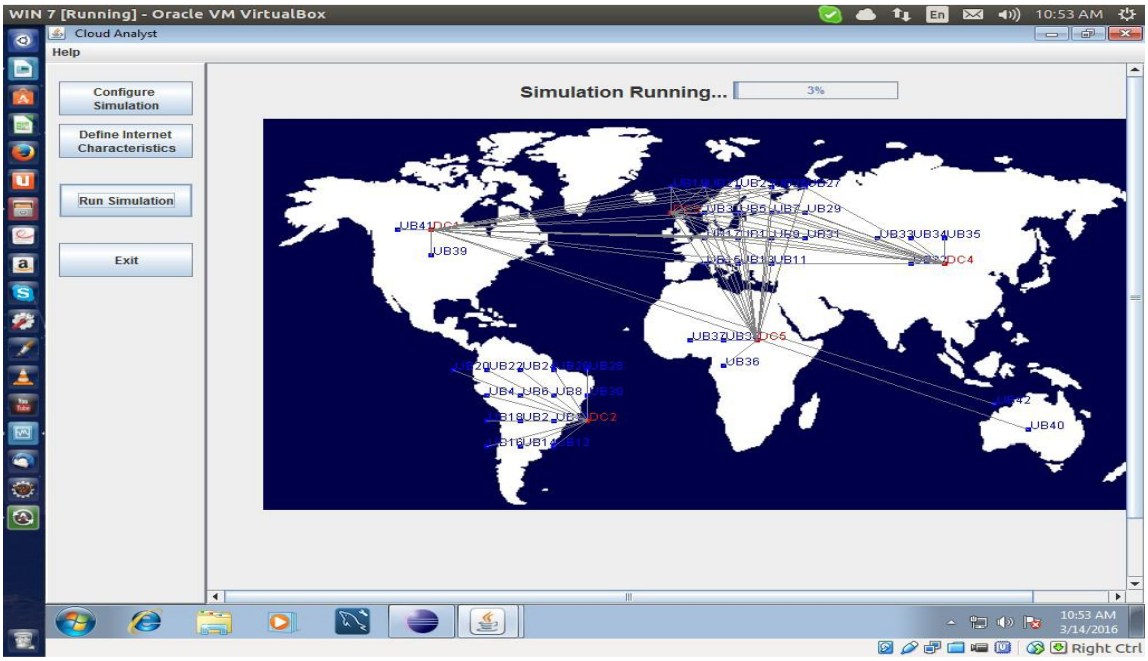

Figure.3: Simulation (Running) Screen of Distributed Data Center

Simulation Setup and analysis of result s are carried out for a period of 60 hrs by taking different number of users and 6 data centers from DC1, DC2, ……DC6. Each data center has 5 virtual machines separately. The other parameters are fixed according to table no.

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 performance different experiments by cloud analyst successfully, we get overall response time of different load balancing algorithms as given in the table. Analytical result of overall response time and load based on various algorithms in cloud computing environment is shown:

Table.4: Comparison of Overall Response Time and Data Center (DC) processing Time Summary

Distributed Data Center Over All Response Time (RT)												
Algorithm	**Round Robin Algorithm**			**Active Monitoring Algorithm**			**Throttled Algorithm**			**DSBP Algorithm**		
Response Time	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 Over All Response Time (RT)				
Cost	**Round Robin Algorithm**	**Active Monitoring Algorithm**	**Throttled Algorithm**	**DSBP Algorithm**
Total VM cost ($)	0.04	0.42	0.42	0.40
Total Data Transfe	0.85	0.85	0.85	0.08

r cost ($)				
Grand Total ($)	1.25	1.25	1.27	0.48

The above table represents the values of all existing algorithms and proposed DSBP algorithm for overall response time from different data centers which shows DSBP has better performance showing in below graph (figure. no. 2).

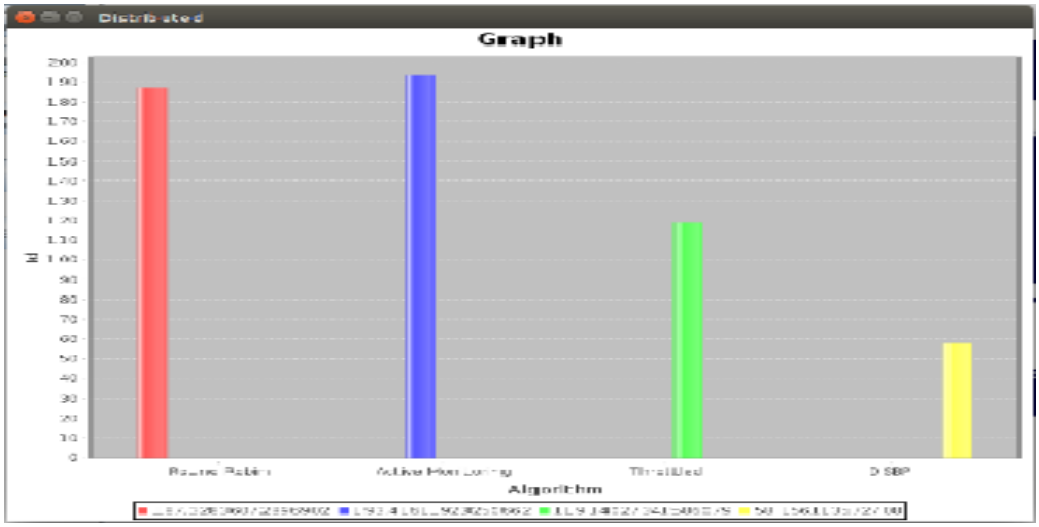

Figure.2: Simulation Graph of Distributed Data Center for load balancing

6. Conclusion

This paper presents the improvement of load balancing algorithm as servers used to get overloaded. Hence we conclude that DSBP algorithm is an effective and efficient for distributed data center in logistics operations. After analyzing the performance results, its seen that DSBP performs better than the round robin, active monitoring and throttled load balancing algorithms in the same time. The proposed algorithm DSBP not only processes more transactions, it also reduces overall response time and load on the distributed data center. Performance of proposed

algorithm confirms that it will be beneficial for logistics information system to track and manage the operation for sharing information from different location at same time.

References

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

[2] Jasmin James, Bhupendra Verma, Effcient 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, pages:793-797, Feb (2012)

[4] Neeraj Bhargava, Rita Bhargava et.al, Performance Analysis of Cloud Computing for Distributed Client, 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, jan-2014, ISSN 2320-088x (2014)

[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