Journal of Engineering and Applied Sciences 14 (2): 507-515, 2019
ISSN: 1816-949X

Implementation of Load Balancing Algorithm with Cloud Collaboration for Logistics

[1]Shivani Dubey, [2]Mamta Dahiya and [3]Sunayana Jain
[1]Department of Computer Applications, JSS Academy of Technical Education, Noida, India
[2]School of Engineering and Technology, Ansal University, Gurugaon, India
[3]Institute of Management Studies, Department of Operation and Information Technology, Ghaziabad, India

Abstract: Cloud computing provides a facility of any data center at any location in the world. There are various applications used in a centralized and distributed data center that suppliers and users can purchase, sell and rent for their information and product accessing. The suppliers and user have no knowledge about where these data centers are located and how they will be operated or maintained by using cloud collaboration. They only know how to connect or use the applications of cloud to perform their jobs. Today, there is lot of demand of cloud in various companies to establish their own data center. In logistics management, sharing of information at different level by different logistics partners, users and suppliers is a big challenge. They all want that every type of information should be shared in real time without any delay at minimum cost. For this, they need cloud based load balancing approaches to control network traffic and overloading on the data center. Load balancing is a distributed technique of workload for load balancing between two or more cloud servers. Load balancing considers service providers management, high traffic and always be ready to reduce and balance peak load on the server. Load balancing have always task to optimize resource use, virtual machine maintains data center cost, maximize, throughput, minimize response time and minimize overload. There are so many load balancing algorithms utilized to balance the load on the servers. In this study, we present technical review on different load balancing algorithms.

Key words: Cloud computing, logistics information system, load balancing algorithms, proposed DSBP algorithm, utilized, applications

INTRODUCTION

Cloud computing is exploring dynamic and scalable computing and storage platform which creates needs of users because of its easily adoption. Cloud plays an important role in public sector organization for reducing cost for using information technologies and services (Alsanea *et al*., 2014). Centralized data center is repository for information distribution and information storage. The centralized data center provides highly efficient for transferring of information and its services for managing all levels of logistics information over the cloud. Logistics management faces different types of issues such as accessing of information at different levels, logistics operations processing, distribution of information inventory and changing in users demand (Feng, 2013). Distribution of any type of information among the suppliers, partners and users is very big issue because number of users is increasing to use various websites over the internet which creates overloading and traffic for providing efficient information for availability of different types of products from different data centers. Load balancing is a distributed technique of workload for balancing between two or more cloud servers. Load balancing considers managing service provided high traffic and always be ready to reduce and balance peak load on the server. Load balancing have always task to optimize resource use, virtual machine maintenance, data center cost, maximum throughput, minimum response time and overload. Load balancing applications redundant mirrored databases in clusters techniques across multiple availability reduces the chance of outages have simultaneously affect the cloud services. If an outage affects one system, the load balancer switches to another available resource. Load balancing technique reduces cost with document management systems and increases availability of resources to reduce amount of downtime that affects business during outages. Load balancing is a new technique which provides facilities of network and resources with maximum throughput and minmum response time. Load balancing provides different solutions applied redundant servers which helps a better distribution of the communication traffic, so that, website availability is conclusively maintained in our daily life.

Corresponding Author: Shivani Dubey, Department of Computer Applications, JSS Academy of Technical Education, Noida, India

Today, load balancing technique is used to enhance the software embedded on compatible hardware in distributed network. Load balancing ensures the implementation of distributed system with restructuring the system workload at highest users and maximum throughput. Load balancing is applied to avoid large difference in workload on servers. There are so many load balancing algorithms utilized to balance the load on servers. In this study, we have presented different load balancing algorithm and evaluated them on the basis of overall response time, data center processing time and total cost of virtual machine and data transfer cost. There are various load balancing algorithms but all existing algorithms have issues to reduce response time, overloading and data transfer cost at same time. We develop a cloud based application by using our proposed distributed service broker policy algorithm for logistics to avoid delay in response time and cost due to multiple users request at same time from different data centers. We present evaluation of some existing load balancing algorithms, round robin algorithm, active monitoring algorithm and throttled load balancing algorithm by using cloud analyst tool and we will also compare the result performance of our propose solution DSBP with these existing algorithms.

Literature review: Load balancing ensures the better performance of distributed system by using redesigning the system overload with highest customer's satisfaction at lower cost. Load balancing techniques are applied for avoiding the situation of different types of over loading on the server. Safiriyu *et al*. defined Distributed Web Servers (DWS) which provides an effective solution to improve the quality of web services. The experimental result has shown the better performance of network traffic, system throughput, mean response time and system utilization of Cooperative Adaptive Symmetrical Initiated Dynamic/diffusion (CASID) and compared it with PLB (Platform for Load Balancing). Randles *et al*. (2010) presented a comparison of static and dynamic load balancing algorithms for cloud computing. Li *et al*. (2010) proposed resource allocation mechanism with preemptable task execution which enhance the utilization of clouds. Singh *et al*. (2010) proposed Modified Round Robin (MRR) which is better than RR and have less response time and reduced the overhead and saving of memory space (Yaashuwanth and Ramesh, 2010). Yaashuwanth and Ramesh (2010) proposed RR (MRR) algorithm which removes the limitations of simple RR (Singh *et al*., 2010). Padhy and Rao (2011) defined some existing load balancing algorithms which can be applied over the cloud. Mohanty *et al*. (2011) also presented dynamic time quantum which modifies with every round of exe cution and result shows that PBDRR performs better than MRR algorithm. Sharma *et al*. (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. James and Verma (2012) also 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 cloudSim. Limbani and Oza (2012) proposed the extended service proximity based routing policy algorithm forrouting of user requests such that cost effective data center selection by using cloud analyst. Mohapatra *et al*. (2013) analyzed various policies utilization of different algorithms for load balancing by using cloud analyst tool. Elina *et al*. evaluated particle swarm optimization based cloud scheduler. The study discussed the number of cloud users are able to serve successfully and total number of created Virtual Machines (VMs) in scheduling scenarios, although, the number of intra cloud network messages sent is also evaluated. The simulated results shows some problems related to scheduler succeed to balance the random assignment and genetic algorithms. Bhargava *et al*. (2013) also analyzed and discussed round robin algorithm used by different data center and calculated the overall response time in better performance. Priya and 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 (Priya and Subramani, 2013). Singh and 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. Mishra and 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. Kapgate (2014) also compared the new service broker (DC selection) algorithm with existing service broker algorithm. The proposed algorithm reduced service response time and improved performance of daa center. Nayak and Patel (2015) presented comparison of some 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. Panwar and 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. Bodake (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. Khanchi and Tyagi (2016) also 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 (Khanchi and Tyagi, 2016). Norwal and Dhingra (2017) proposed a multi objective task scheduling algorithm which consider a wide variety of attributes in cloud environment and uses non dominate sorting for prioritizing the task. The proposed algorithm improved the processing time, cost and average waiting time in comparison with First Come First Serve (FCFS) algorithm (Norwal and Dhingra, 2017). Kanmani and Jayabalan (2017) presented a survey and comparison of existing innovative algorithms based on different parameters like, throughput maitainces, reliability, efficiency, scalability and performance. In this study, 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 bandwdth, latency and size of job which achieves minimum response time and minimum cost.

MATERIALS AND METHODS

Issues 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 *et al*. (2012) implemented the throttled load balancing algorithm and proposed virtual machine load balancing algorithm which reduce response time effectively but not reduce cost. Bhargava *et al*. (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 and Subramani (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. Singh and Gangwar (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. Mishra and Bhukya (2014) also 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. 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. Nayak and Patel (2015) proposed throttled algorithm and compared it with existing throttled algorithm. The proposed algorithm reduces response time and cost in better performance butif data center will be increased then response time cost will be increased. Panwar and Mallick (2015) also 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. Bodake (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. Khanchi and Tyagi (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. Mehar and Acharya (2017) presented a comprehensive summary of load balancing algorithm in cloud computing. In this study, existing approaches are analyzed to provide load balancing in cloud environment. The comparison of different algorithms presented minimum response time, throughput, fault tolerance, overhead and speed (Mehar and Acharya, 2017). Reema and Sehgal evaluated optimize response time and reconfigure dynamically policies for evaluating task mapping algorithm to minimize resource cost, VM cost, data center cost and processing time but results are not showed successfully. In Table 1, we can see, so many results of different algorithms in perspective of response time and cost to identify the improvement of round robin, active monitoring and throttled algorithms.

Table 1: Existing algorithm simulation results (Kishore and Thapper, 2014; Behal and Kumar, 2014; Nandwani *et al*., 2015; Patel and Rajawat, 2013; Singh *et al*., 2016a, b; Shukla and Arora, 2016; Sharma and Gupta, 2016

	Algorithms			Results	
Researcher's (Years)	Round Robin algorithm	Active monitoring algorithm	Throttled load balancing algorithm	Average response time (msec)	Total cost ($)
Kishore and Thappar (2014)	No	No	No	481.54	Nil
Behal and Kumar (2014)	Yes	No	No	322.55	2505.48
Ali Naser	No	No	Yes	139.98	59.50
Nandwani *et al*. (2015)	Yes	Yes	Yes	92.10	Nil
Patel and Rajawat (2015)	Yes	No	Yes	177.59	2.28
Singh *et al*. (2016a, b)	Yes	No	Yes	57.68	Nil
Er. Pooja	No	No	No	57.08	12.85
Singh *et al*. (2016a, b)	Yes	No	No	484.25	Nil
Shukla and Arora (2016)	Yes	No	Yes	150.87	Nil
Sharma *et al*. (2012)	No	No	No	333.09	Nil
Shipra Gayal	Yes	No	Yes	338.27	Nil
Manasrah *et al*. (2017)	Yes	Yes	Yes	498.93	973.53

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

In the above table, we can see the performance of existing algorithms for implementing the performance of response time and data transfer cost but we can analyze that performance of response time and data transfer cost not be in improving way at same time. Our proposed DSBP algorithm is basically implementation of throttled load balancing algorithm within cloud environment to achieve better response time and data transfer cost.

Proposed solution: Our proposed DSBP algorithm selects the data center according to job size, the expected processing time, network latency and availability of bandwidth to reduce the overall response time and processing time. The routing policy includes different parameters like, request size, user size, number of data centers, number of virtual machines, bandwidth, response time, data transfer cost, etc. The processing time depends upon the data center specification such as RAM, CPU and VM configuration. In Table 2, we have implemented the methods of round robin, active mentoring and throttled load balancing algorithm. Our proposed DSBP algorithm is the implementation of throttled load balancing algorithm. As a result, our proposed algorithm accommodates the current needs by taking real time values to evaluate the processing time to reduce the time needed to make the forwarding decision by the broker. The job processing time can depend on the computational task to be performed. For instance, a smaller job requires minimum processing time if there was no any I/O operation involved. However, since, it is not the service broker functionality to analyze the jobs and examine their complexity, we have considered the job size as an indication to the needed processing time with a positive relation between them. In Table 2, we present our implemented methods of existing and proposed DSBP algorithms. The proposed DSBP algorithm is the implementation of throttled load balancing algorithm, which works efficiently under the capacity of each virtual machine is different because the hardware configuration of virtual machines is different.

RESULTS AND DISCUSSION

Experimental setup and result analysis: In our experiment, multiple users are capable to use distributed data base center by using cloud based scheduling algorithms. For compute overloading, it is more useful to distribute load across the data centers over the parameters, bandwidth, VM image size, VM storage, VM memory, etc. for data center workload, a large number of users would be served by a cloud analyst where result can be easily persisted on defined parameters. There are three principles presented in cloud analyst, User Base (UB), data center controller and internet. In cloud analyst, world is divided into 6 regions and the primary substances are user bases and data centers have a place to one of regions. This topographical collection is utilized to keep up a level of reasonable for the extensive scaled reproduction being endeavored in the cloud analyst. The cloud analyst internet presents reality of internet which highlights transmission dormancy and information exchange delay. Our simulation setup are carried out for 60 min by taking different number of users in form of 50 UB and 6 data centers from DC1-DC6. Each data center has 5 virtual machines separately with different parameters (Fig. 1-3).

The above configuration and simulation screens presents the performance of our proposed DSBP algorithm based on defined parameters in Table 3. This simulation result provides processing time, over all response time and data transfer cost in distributed data center environment experimented by cloud analyst. We have also taken this type of simulation results for existing round robin, active monitoring and throttled load balancing algorithms. After the performance all existing algorithms and our proposed DSBP, we get different type of values for processing time, response time and data transfer cost as given in Table 4, 5 and Fig. 4, 5.

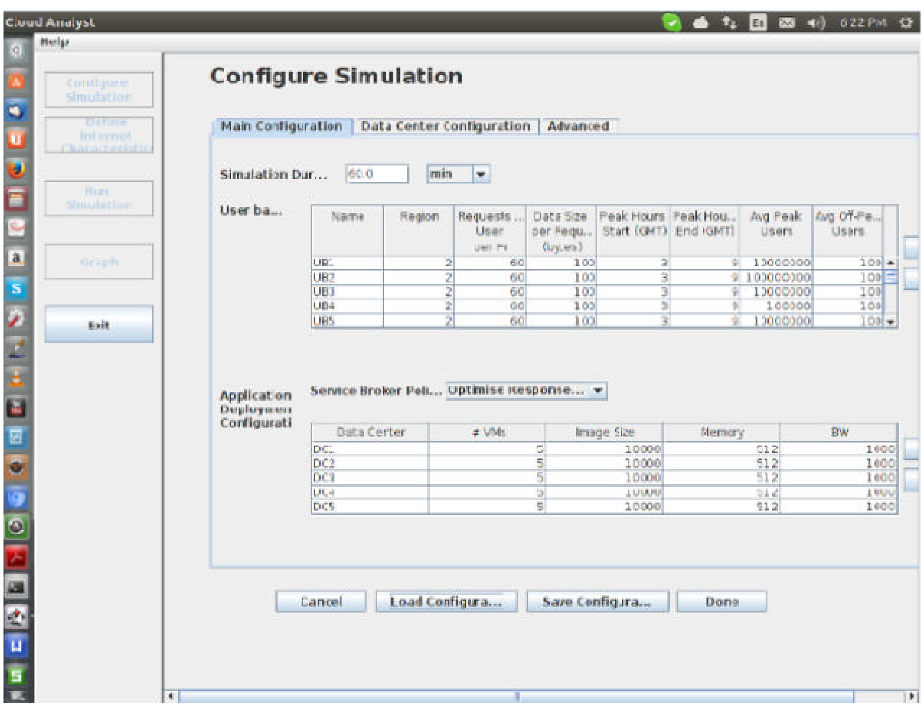

Fig. 1: Main configuration screen of userbase and application deployment

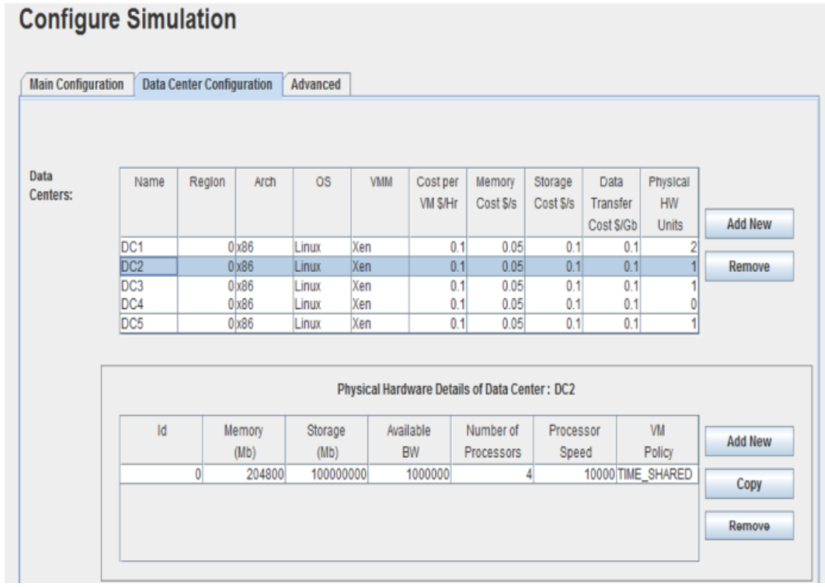

Fig. 2: Main configuration screen of data center

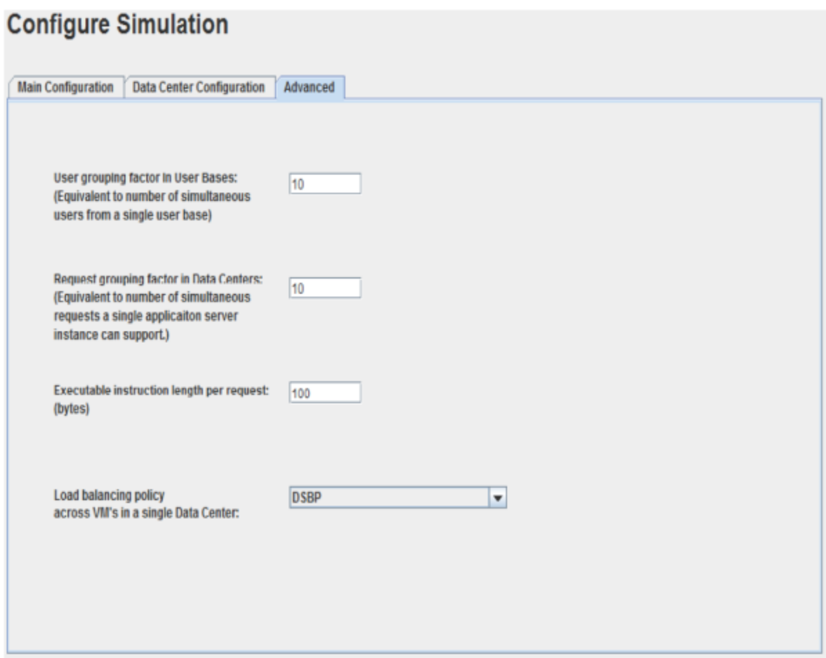

Fig. 3: Configure simulation screen for proposed DSBP algorithm

Table 3: Parameters and values

Parameters	Values
V-M 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
MBPS DC storage per machine	1GB
DC available bandwidth per machine	1000000

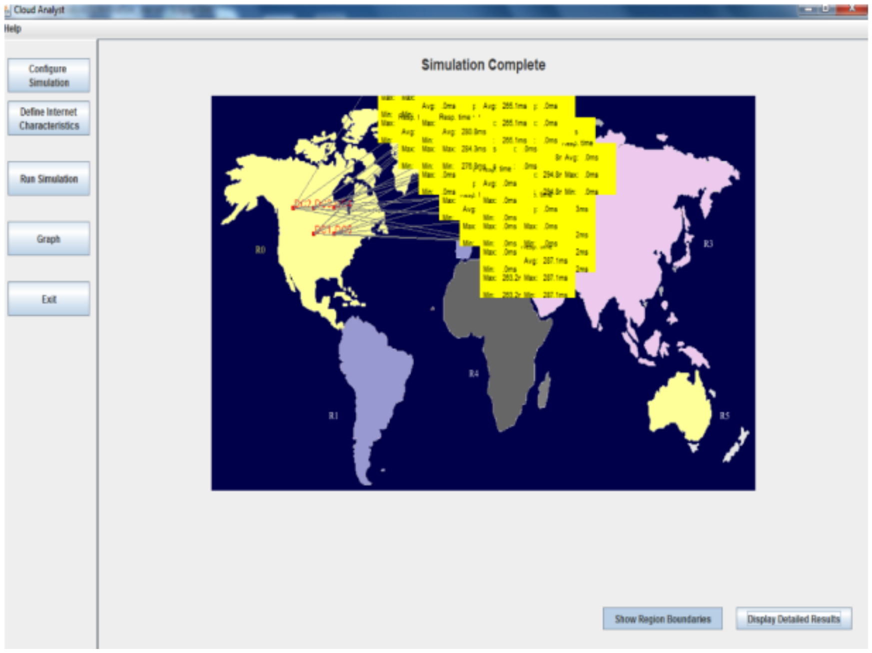

Fig. 4: Simulation (running) screen of distributed data center

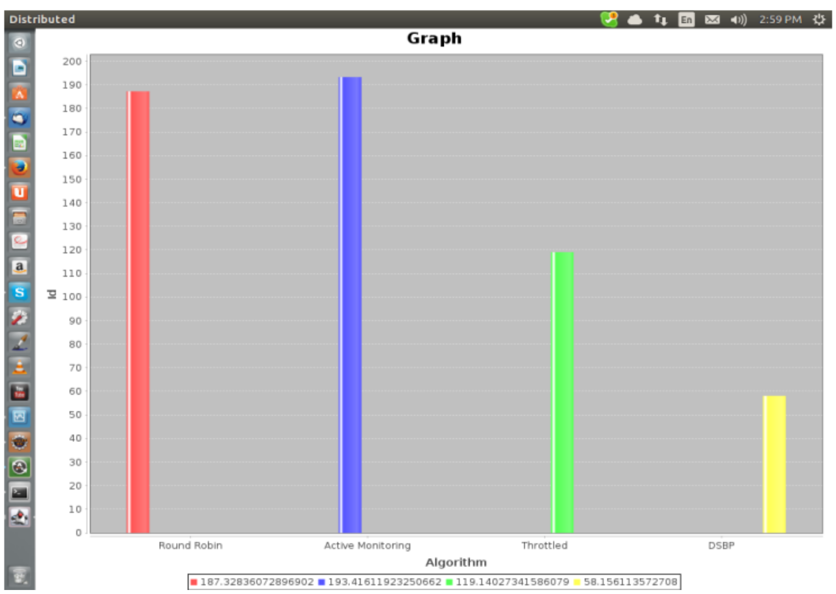

Fig. 5: Simulation graph of distributed data center for load balancing

Table 4: Comparison of overall response time and Data Center (DC) processing time summary

Distributed data center over all Response Time (RT)

Algorithm/ Response time	Round robin active (msec)			Active monitoring algorithm (msec)			Throttled algorithm (msec)			DSBP algorithm (msec)		
	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.
Over all RI	108.09	12.38	68.15	111.17	9.25	70.77	85.50	8.75	69.11	54.72	0.01	60.38
DC processing time	98.3	0.00	0.22	72.19	0.00	0.22	60.01	43.23	0.22	0.12	0.03	0.20

Table 5: Comparison of total cost summary

Algorithm/ Response time	Round robin active (msec) Avg.	Min.	Max.	Active monitoring algorithm (msec) Avg.	Min.	Max.	Throttled algorithm (msec) Avg.	Min.	Max.	DSBP algorithm (msec) Avg.	Min.	Max.
Distributed data center over all Response Time (RT)												
Over all RI	108.09	12.38	68.15	111.17	9.25	70.77	85.50	8.75	69.11	54.72	0.01	60.38
DC processing time	98.3	0.00	0.22	72.19	0.00	0.22	60.01	43.23	0.22	0.12	0.03	0.20

CONCLUSION

For compute overloading, it is more useful to distribute across load of users request rather than within each data center over the different parameters (data center storage, memory, bandwidth, etc.) for data center workload. A large number of users would be served by a cloud analyst where result can be easily persisted on defined parameters. Our proposed solution is expected to have significant impact on the performance of distributed data center by virtue of its capabilities to adopt cloud based policies and load balancing algorithms. Load balancing and cloud collaboration drive the best solution in performance of distributed data center environment. Our proposed DSBP algorithm is able to play an important role to reflect the load balancing of different data centers for information processing at different level of logistics in real world computing environment. Hence, we conclude DSBP algorithm has effective result in logistics operations. We can also see an analytical performance of DSBP which has better results in comparison to round robin, active monitoring and throttled load balancing algorithms at a same time for reducing overall response time, data transfer cost and load in the distributed data center. The significance of this research is the demonstration of the DSBP algorithm using a cloud collaboration of advanced technologies which will surely help to present efficient performance of logistics management for tracking the information and managing the operations to distribute information from different location in distributed data center at same time. Future research is related to design a new load balancing algorithm for better resource utilization, minimum response time and minimum cost for fast throughput of cloud computing environment for logistics management to also maintain inventory distribution, vehicle locations etc.

ACKNOWLEDGEMENTS

The researchers are highly thankful to Dr. Monika Khurana associate Professor, Ansal University, Gurgaun and also to Dr. Abhaya Nand assistant Professor, JSS Academy of Technical Education, Noida for their continues support and guidance in preparing this study, so, meticulously.

REFERENCES

Alsanea, M., J. Barth and R. Griffith, 2014. Factors affecting the adoption of cloud computing in the government sector: A case study of Saudi Arabia. Intl. J. Cloud Comput. Serv. Sci., 1: 1-16.

Behal, V. and A. Kumar, 2014. Comparative study of load balancing algorithms in cloud environment using cloud analyst. Intl. J. Comput. Appl., 97: 36-40.

Bhargava, N., R. Bhargava, M. Mathuria and R. Daima, 2013. Performance analysis of cloud computing for distributed client. Intl. J. Comput. Sci. Mobile Comput., 2: 97-104.

Bodake, S.V., 2016. Cloud computing load balancing algorithms comparison based survey. Intl. J. Adv. Eng. Manage. Sci., 2: 282-287.

Feng, Z., 2013. A logistics distribution plan based on cloud computing. Intl. J. Cloud Comput. Serv. Sci., 2: 331-335.

James, J. and B. Verma, 2012. Efficient VM load balancing algorithm for a cloud computing environment. Intl. J. Comput. Sci. Eng., 4: 1658-1663.

Kanmani, G. and E. Jayabalan, 2017. A survey on recent improved LB algorithms in cloud environment. SSRG. Intl. J. Comput. Sci. Eng. NCSACT., 1: 12-15.

Kapgate, D., 2014. Efficient service broker algorithm for data center selection in cloud computing. Intl. J. Comput. Sci. Mobile Comput., 3: 355-365.

Khanchi, M. and S. Tyagi, 2016. An efficient algorithm for load balancing in cloud computing. Intl. J. Eng. Sci. Res. Technol., 5: 468-475.

Kishore, K. and V. Thapper, 2014. An efficient service broker policy for cloud computing environment. Intl. J. Comput. Sci. Trends Technol., 2: 104-109.

Li, J., M. Qiu, J.W. Niu, Y. Chen and Z. Ming, 2010. Adaptive resource allocation for preemptable jobs in cloud systems. Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications (ISDA), November 29-December 1, 2010, Cairo, Egypt, pp: 31-36.

Limbani, D. and B. Oza, 2012. A proposed service broker strategy in cloud analyst for cost effective data center selection. Intl. J. Eng. Res. Appl., 2: 793-797.

Manasrah, A.M., T. Smadi and A. ALmomani, 2017. A variable service broker routing policy for data center selection in cloud analyst. J. King Saud Univ. Comput. Inf. Sci., 29: 365-377.

Mehar, S. and H. Acharya, 2017. A survey on modern load balancing algorithms in cloud computing. Intl. J. Eng. Res., 6: 228-232.

Mishra, R.K. and S.N. Bhukya, 2014. Service broker algorithm for cloud-analyst. Intl. J. Comput. Sci. Inf. Technol., 5: 3957-3962.

Mohanty, R., H.S. Behera, K. Patwari, M. Dash and M.L. Prasanna, 2011. Priority Based Dynamic Round Robin (PBDRR) algorithm with intelligent time slice for soft real time systems. Intl. J. Adv. Comput. Sci. Appl., 2: 46-50.

Mohapatra, S., S. Mohanty and K.S. Rekha, 2013. Analysis of different variants in round robin algorithms for load balancing in cloud computing. Intl. J. Comput. Appl., 69: 17-21.

Nandwani, S., M. Achhra, R. Shah, A. Tamrakar and K.K. Joshi *et al*., 2015. Analysis of service broker and load balancing in cloud computing. Intl. J. Curr. Eng. Sci. Res., 2: 92-98.

Nayak, S. and P. Patel, 2015. Analytical study of throttled algorithm of load balancing in cloud computing using cloud analyst. Intl. J. Sci. Technol. Eng., 1: 90-100.

Norwal, A. and S. Dhingra, 2017. Task scheduling algorithm using multi objective functions for cloud computing environment. Intl. J. Control Theory Appl., 10: 277-278.

Padhy, R.P. and P. Rao, 2011. Load balancing in cloud computing systems. BA Thesis, National Institutes of Technology, India.

Panwar, R. and B. Mallick, 2015. Load balancing in cloud computing using dynamic load management algorithm. Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT'15), October 8-10, 2015, IEEE, Delhi, India, ISBN:978-1-4673-7909-0, pp: 773-778.

Patel, D. and A.S. Rajawat, 2015. Efficient throttled load balancing algorithm in cloud environment. Intl. J. Mordern Trends Eng. Res., 2: 463-480.

Priya, S.M. and B. Subramani, 2013. A new approach for load balancing in cloud computing. Int. J. Eng. Comput. Sci., 2: 1636-1640.

Randles, M., D. Lamb and B.A. Taleb, 2010. A comparative study into distributed load balancing algorithms for cloud computing. Proceedings of the 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, April 20-23, 2010, IEEE., Perth, Australia, pp: 551-556.

Sharma, M., P. Sharma and S. Sharma, 2012. Efficient load balancing algorithm in VM cloud environment. Intl. J. Comput. Sci., 3: 439-441.

Shukla, S. and D. Arora, 2016. A comparative performance analysis of load balancing policies in cloud computing using cloud analyst. Intl. J. Innovative Res. Comput. Commun. Eng., 4: 8381-8386.

Singh, A., P. Goyal and S. Batra, 2010. An optimized round robin scheduling algorithm for CPU scheduling. Intl. J. Comput. Sci. Eng., 2: 2383-2385.

Singh, H. and R.C. Gangwar, 2014. Efficient approach for load balancing in virtual cloud computing environment. Compusoft, 3: 1204-1210.

Singh, P., P. Dimri, G.P. Saroha and V. Barthwal, 2016a. A load balancing analysis of cloud base application with different service broker policies. Intl. J. Comput. Appl., 135: 11-15.

Singh, S.P., A. Sharma and R. Kumar, 2016b. Analysis of load balancing algorithms using cloud analyst. Intl. J. Grid Distrib. Comput., 9: 11-24.

Yaashuwanth, C. and R. Ramesh, 2010. Web-enabled framework for real-time scheduler simulator: a teaching tool. J. Comput. Sci., 6: 374-380.