A Co-operative Fog-based Load Balance (CFBLB) Strategy

Mary M. Fouad
Faculty of Engineering
Mansoura University

Ahmed I. Saleh
Faculty of Engineering
Mansoura University

Mohamed F. EL-Rahamawy
Faculty of Computer Science
Mansoura University

ABSTRACT

The most factor affects the performance of fog computing is load balancing which means resource management, it is significant to get a satisfying implementation of fog computing. The existing algorithms of LB in a fog computing environment are not extremely active. today, to predict the user requests arrivals on the fog manager is not possible so Load balancing is a complex mission. Each machine has different characteristics, So the job scheduling process among nodes turns into a very hard process Lately, load balancing is the main goal to many researchers in fog computing which means resource management. that produce many algorithms to achieve this goal such as dynamic Algorithm, the proposed algorithm used to improve load balance model for the fog. This algorithm is being introduced to improve resource utilization and response time in the fog environment, which applies fuzzy inference with load scheduling that takes advantage of Fuzzy Logic. by using java The output produced refinement on resource utilization and processing time.

Keywords

Load balancing, fog computing, Fuzzy inference, Virtualization.

1. INTRODUCTION

Cloud computing is defined as the internet service is available to the user at any time and from any place that means resources are scalable and dynamic. The client's requirement at a specific time which may be software, information, shared resources, and other devices are provided by Cloud computing, so it is an on-demand service. Cloud computing According to (NIST)[1], in it the network can be accessed on demand and conveniently. That will reduce the effort managed and achieve a rapid provision to computing resource Two models of cloud computing :(i) Deployment model: which manages the cloud infrastructure. (ii) Service model: accessed on platform. In recent times, cloud computing is the most broadcast technology that takes efforts from researchers. cloud computing has been considered as a type of platform as well as an application for evolution. In cloud computing users can access computer resources on the internet easily .it is pay as you go .it is a data center handles the big data virtually. By using virtualization the platforms are dynamically built to allocate the resources to several users which may lead to imbalance and congestion in the network [3].thus the system need a mechanism for distributing the load among the resources to solve this problem which called a load balancing strategy, by managing or handling the load to get the goal which is minimum response time. The load balancing strategy achieves optimal resource utilization. the resources needed by user request may be processors, memory buffers or drivers .these resources must be used efficiently and effectively, that is the main goal. Due to the increasing use of cloud resources, It is required to promise resources effectively to provide good Service Level Agreement (SLA) with zero downtime claims. Recently, many companies offer many cloud services such as IBM, Microsoft, Google, Amazon, HP, Citrix, Salesforce, EMC, Oracle, etc. and also there are many numbers of clients. It is clear that the growing numbers of clients will represent a big challenge due to the dynamic requirements. In 2012, the need to extend cloud computing with fog computing emerged, to cope with a huge number of IoT devices and big data volumes for real-time low-latency applications[3]. Fog computing anew expression generated by

Cisco points to widening cloud computing to the edge of an enterprise's network. Also known as Edge Computing or fogging, fog computing facilitates the operation of computing, storage, and networking services between end devices and cloud computing data centers. While edge computing is typically referred to the location where services are instantiated, fog computing implies distribution of the communication, computation, and storage resources and services on or close to devices and systems in the control of end-users[4] .fog computing main characteristics are low latency, low energy consumption, real-time interactions, geographical distribution and data security and privacy protection. Virtualization: Same as virtualization in cloud computing, virtualization in Fog makes it possible to share the hardware resources (CPU, memory, storage, network) safely and fairly between users of the platform. Imagine dynamically launch and delete certain numbers and certain types of virtual machines for users. Virtualization technology is fundamental for Cloud and Fog, in terms of resource control, economic model, and the realization of the system. The examples of VMware and Virtual Box talk about heavy VM, there exists also light VM (or container) which makes virtualization more flexible. it means to create an imaginary form of a resource, like a storage device, server, an operating system or network in a computing machine. Virtualization gives all the services of as actual but it means something imaginary. The user may utilize many services of fog computing when using virtualization consequently the fog computing is connected with Virtualizations. Virtualizations enables dividing the resource into one or more carry out environments. it provides a way to separate the physical hardware and virtual Software OS. and application There are two types: Full Virtualization in which VM will have the software that exists in the real server, in which a full installation Done on the machine. The second type called Para virtualization in which multiple operating systems runs on only one machine by permission of the hardware. example VMware software. Rather than the full services, here all the services are partially provided. Load balancing:[5] can be defined as a way to distribute the amount of work or the jobs reached to the network on the multiple devices to obtain maximum throughput and minimizes overall response time and to get optimal resource utilization. Load balancing achieves that the traffic is divided among nodes that lead to avoid the overload on resources, it will minimize the total waiting time of the resources. As a result, data can be sent and received without maximum delay. it is used to make sure that none existing resources are idle while others are being utilized.LB takes great attention in fog computing because of the scale-up in demands. The ultimate goal of load balancing is as follows: Even distribution of load to each node, Minimization of processing time for each job and Maximum utilization of each resource.

2. PREVIOUS WORK

All Researchers cannot predict the user traffic in Virtualized fog manager so seriously need a good load balancing algorithms. The response time is the main goal in the load balancing process so the equal distribution of load among the nodes will provide good response time. It's obvious if the incoming load of fog manager is routed to a few servers while others seating idle, that will increases the response time. The ideas behind fog computing are not new. Many years ago, before IoT became a term and people who now work on IoT called their work “sensor networks”, researchers were thinking about how a sensor network would operate with a powerful gateway (which called now a fog node) and without it. the presence of additional local logic in a fog node can help IoT devices. The world now in an exciting stage of technology development where deploying such local logic is on the verge of becoming a reality. Fog computing is meant to work with cloud computing, not replace it. Take a look at Amazon Green grass architecture, for example, which uses fog computing/edge computing for improving local response times and reliability for IoT applications, while still maintaining a connection to the cloud. Such architectures make a lot of sense; will be seen many more of them over the next couple of years. Fog computing helps mobile devices do more — it supports wide mobile experiences, not just mobility. If all intelligence is placed in the cloud, then cloud response time becomes the bottleneck which stops many mobile applications from being developed or even conceived. For example, many interactive experiences require response times of under 20ms, while the average cloud response time is more than 70ms, with large variations between different geographical locations and between different conditions. By reducing this time to get to intelligence, fog allows mobile devices to think faster, so to speak. Many types of research provide many numbers of load balancing algorithms. These algorithms are described briefly below. Classification of algorithms depending on the implementation of the method fog computing dynamic load balancing mechanism [6]. (i)the first one is FCFS if two tasks become equal credit the first come first service will be applied. (ii)the second is the priority which takes into consideration task priority and its length calculates a combination of both to maximize the resource utilization. The task executed first if it has high credit. (iii)the third is HDLB which uses the static segmentation algorithm, in which the information of the system state should be recomputed of nodes needs to move. (iv)the fourth uses the dynamic segmentation algorithm based on graph repartition. the current task allocation table of each VM will be scanned by the active load balancer to return with the most suitable VM which has the minimum task allocation status. After that, the task allocation counter will be increased by 1. The active monitor load balancer decreases after the task is executed in its suitable VM. In[8] modified active monitoring load balancer: it allocates the incoming request to the least loaded virtual machine without monitoring the memory utilization. fog can be offline but in the cloud, it depends completely on internet connection. for example, it may need wireless connection to process data from sensors, while a fog in a smart house or a factory is wired used for monitoring data and processes.

3. THE PROPOSED CO-OPERATIVE FOG –BASED LOAD BALANCER(CFBLB).

proposed algorithm in this paper the fog-based load balancing algorithm related to fuzzy in the Virtual Machine (VM) environment reaches the best latency, resource utilization, and processing time. This new algorithm is applied before the job reaches the servers .it is scheduled depends on three parameters the first one is the processor speed, the second is the assigned load of Virtual Machine (VM) the last is the bandwidth of the network. It preserves the numbers of requests currently allocated to VM and information in each VM of the system. The proposed algorithm characterized the machine which has the least load when a request comes to schedule and it characterized the first one if there is more than one machine have the least load applying the proposed load balancing technique based on Fuzzy logic. This research uses fuzzy logic for many reasons one of them is, it is easy and natural .it's advantages are easy to deal with imprecise data, easy to understand, flexible and can model nonlinear functions [7][8]. Fuzzy inference defined as the process of eliciting the mapping from a given input to an output using fuzzy logic. It gives a basis from which decisions can be made, or patterns recognized by the process of mapping. In our proposed work, the fuzzifier converts the inputs of each VM and give one output. illustrate the process workflow. In our proposed work, the research uses fuzzy logic considering three inputs which are the processor speed, assigned load in VM and the bandwidth as three inputs to achieve maximal value to balance the load in the system. It takes the input to the fuzzifier as the three parameters, the balanced load has been measured by these parameters to get the output. The effective VM which processes the client request has been identified by this algorithm. when the job arrives at the job manager to be scheduled to the perfect machine according to the classification of the data it can be handled .if it is a big data should be handled in cloud computing and sent it to the data center otherwise it sent to the fog. according to the stability and the statues of the fog if it is busy, or middle. it will be decided by applied the following algorithm :

```
Algorithm 1:OPTIMAL FOG LOAD
Input:FogNode(Fn);
FogCapacity(Fc);CurrentLoad(Cl);OverLoad(Hl).

Parameters:OffloadValue(Fv);LoadToShare(Ls);
TasksQueue(q).

Output: ListOfNodes=(Fn,Ls);Having (Ls≤Hl).

Initialization :Fn= $\varphi$; Fc= $\varphi$; Cl= $\varphi$;Hl= $\varphi$;Fv= $\varphi$; Ls= $\varphi$.

Result :Distribute/Share the Overload requests on fogs

 1 Procedure 1.Determine the overload by
2 if Fn =$\varphi$ then
3 Cl=Fnq$\cup$ Tn          Tn new requests
4 Fc =getCapacity(Fn)
5 Hl=Cl-Fc
6 else7 Fn $\leftarrow$ getNode(out Fn=Cl$\geq$Fc)
8 goto 3
```

```
9 eng
10 if Hl $>$0 then
11 return Fn $\leftarrow$Hl
12 else
13 return Fn
14 end
15 end
16 Procedure 2. Determine Best Neighboring by
17 Fl=list{ $\varphi$ } Fl initiate fog list
18 Fl $\leftarrow$getFogNodes()
19 Fl =sort (Fl,by Fc DESC)
20 for each Fn $\in$ Fl do
21 if Fn $\leftarrow$(Cl $\geq$Fcmax) then
22 pop(Fl,Fn) remove busy nodes
23 else
24 goto 20
25 end
26 end
27 costfunction(Fl, DESC) get best nodes
28 Fl $\leftarrow$ Fl{Fn,Ls}
29 return F
30End:
```

The conversion of the physical node of fog computing into a virtual machine accomplished by the process of cloud atomization treatment[6]. The need for fog concerning the following properties of a system:

(1) The dynamics of your IoT system,

(2) The volume of data produced by it.

The more dynamic the IoT system is and the more data the system produces, the more likely it is to benefit from having a fog component. E.g., if you want your system to the cloud for processing may be the best architecture for you There are three demands required from user request. this algorithm uses these demands to assess the status of the load. the load status has been deduced by A Type-I Fuzzy System. To find the suitable VM at which the request can be scheduled the fog manager sends the resource requirements of that request to the load balancer. the next step, load balancer scans the Processor speed, assigned _load of each VM and bandwidth of network (step 4.a). Then resource demands are added with actual resource usage to obtain the predictable status of each VM (step4.b). After that the crisp value of Processor speed, assigned _load and bandwidth are converted to fuzzy value. The proposed algorithm using a fuzzy inference system (FIS) with 36rules. mention here some rules not all, of course, like If (Processor_ Speed is low) and (Assigned_ load is low) and (bandwidth is low) then (balanced_ load is low). After that came the rule of the load balancer to esteem the load status of each VM in step 4.c. In step 6-7 the load balancer picks out the VM that has a minimal load and gets its id to Fog Manager (FM). There is a counter in each VM counting the number of incoming serviced tasks so that according to the previous step FM should increase that counter. In step 9.a, After the task has been serviced, the counter should be reduced. The process will be repeated to service another task as in step -10.

Current Allocation Counter of the request, Vc(i)

Task of the user, T; virtual machines number, n;

Fuzzy Variables: Processor_ speed,
PS{low , medium, high, very high}

Assigned_ load, AL {low, medium, high}

bandwidth, BW {low, medium, high}

Virtual Machine Status , {low, medium, high}

Required: convenient Virtual Machine ,VM out.

LB and Request Adapting:

1-Task reached to Fog Manager ,choose the best fog

2-task reached to specified fog , sf from user-base via internet.

3- calling the Fuzzy Load Balancer.

4- VM requirements.

a. inputs, PS ,AL,BW

b. computes the expected inputs for task T

c. VM status (i)= FIS(PS,AL ,BW);

5- End

6- minimal loaded VM selected,

VMOUT=min(VM status).

7- fog manager receives VMOUT from the Load balancer.

8- fog manager update counter

V c(VMOUT)= V c(VMOUT)+1;

De-allocation after the task processed:

9-after task serviced,

a. fog manager update tables, V c(VMOUT)= V c(VMOUT)-1;

10- repeat From Step 2

11-End

Getting the output in a fuzzy engine by using a defined rule on defuzzification process using the method of SOM, Smallest of minimum. A membership function is calculated :

VMout_SOM=MIN(□VM €VM1, VM2,..VMn) where VMout is the minimum loaded virtual machine among VM1, VM2,..VM. According to bases of fuzzy control, this research can simulate human decision making by using the Fuzzy Inference System (FIS). For the output response, a collection of IF-THEN rules are defined Using the rule-based structure of fuzzy logic. There is 36 possibility output in The proposed algorithm, product output response. In the fuzzy system the process which converts the output set into a single value is known as defuzzification there are many methods used in defuzzification but here using the method known as the smallest of minimum. The fuzzy output is a range of values transferred to a single value by defuzzification of this range of values. Defuzzifier produces a non-fuzzy output after it takes over the range of values from the derived fuzzy control action. This output represents the balanced load convenient to load conditions. the membership function has been computed by The defuzzification method for the aggregated output [11][12]. In the first step in our algorithm, the request needs to connect to a resource according to the load serviced by this resource. then select the connection. After that, this connection is used to access the resource by using fuzzy logic according to processor speed, assigned load and bandwidth in a virtual machine. Table -1 show 5virtual machines with different processor speed and assigned load and bandwidth To be the input of the fuzzy system in our experimental simulation. Table-2 shows the characteristics of the data center in our system.

Table 1.input to fuzzy system

Virtual Machine no	Processor speed GHZ	Assigned load	Bandwidth HZ
VM1	3	2	300
VM2	9	5	1000
VM3	4	6	700
VM4	8	2	800
VM5	5	0	900

Table 2.characteristics of fog manager

Parameters	Value used
Number of machines of specified fog	5
specified Fog - Architecture	X86
OS- specified fog	Windows
Memory/Machine – specified fog	2048Mb
VMM- specified fog	Xen
Processor Speed- specified fog	10000 MIPS
VM Policy- specified fog	Time Share

4. EXPEREMENTAL RESULTS

The theoretical results of the proposed policy(CFBLB)will be verified in this section. The main goal of our research is to get the total average response time and resource utilization which should be minimized comparing to other algorithms. Analyzing the performance of five VM load balancing algorithm by our Experimental results .by using mat lab R2014a and connecting this with iFogSim to find out the best resource to the request.

4.1 Evaluation Parameters

The first parameter should be evaluated is Resource Utilization. This simulation mainly validates the advantage of the resource utilization between fuzzy load balancing and HDBL algorithm and Dynamic segmentation repartition algorithm . the expression of resource utilization is defined as In equation (1)

$$RUsed = \frac{1}{m}\sum_{j=0}^{m-1}\frac{Rt_j}{Rc_j}(1)$$

Table 3 illustrate the meaning of symbols of equation (1), [11]:

Table 3. symbols and meaning of equation 1

Symbol	meaning
Rt_j	is the total execution time in the host h_j
Rc_j	is the actual time in host h_j
m	is the number of host
$RUsed$	resource utilization

The second parameter is Response Time Calculation: [12] calculation of the expected response time is the purpose of this algorithm . the equation used to calculate the response time is as follow:

Rt=Ft-At+Td

where, Rt :is the Response time, Ft is the finish time of the job, At is the arrival time of user job and Td is the delay of transmission. Where Td calculated as follow:

Td = Tl + Tr

Tl is the latency of the network and Tr: defined as transferring data (D)from a job from source to destination take Tr time. Applying Poisson distribution on latency matrix it for distributing it.

Tr = D / Bw_{user}

where Bw_{user} = Bwt / N

Bwt: defined as the total available bandwidth .

N : defined as the Current number of transmitted requests. for the value of N take into consideration by the internet characteristics

4.2 Comparison Results

the validity of the algorithm has been demonstrated by recording response time and resource utilization. This research reaches the conclusion that users can get the target of increasing resource utilization and reduce the response time if the algorithm chose a suitable virtual machine. the overall performance will be affected if the number of requests in FM fog manager increased. After the simulation, the results were satisfied .so that better output for the five machines comparing with the other algorithms was obtained. The tables 4,5,6 obviously show that a comparison between the previous algorithms like HDLB and dynamic repartition That show us some facts like resource utilization, response time and data processing time give the same values in the previous algorithms. figure 1 shows a comparison of the previous algorithms and our algorithm. Thus, can be decided which one is better than the other. HDLB Larger tasks take a long time. Before making a decision taking into consideration the length of instruction per request. As shown in table-1 the simulator has simulated the result by utilizing 5machines, waiting for a hundred number of requests to achieve load balancing. the result of the simulation is represented in fig 1,2and 3. Figure1 shows that in our proposed algorithm the fog manager processing time is least value with respect to the other algorithms as the number of task increase the fog manager processing time will be increased but still in CFBLB is the least one comparing with FCFS, priority, HDLB and dynamic repartition .in the previous algorithms sometimes there is a long wait, so they don't work very well, because these methods do not take into account processing time factor, therefore, their performance is not the best. our algorithm CFBLB considers affirmation on time consumption to do not exist and another factor. So its results are ideal. Figure 2 shows that the overall response time also is the least one in our proposed algorithm with respect to the other algorithms. Because in FCFS the task with short response time may wait longer until be serviced, this problem may be solved by priority algorithm .in the proposed algorithm all these problems are solved and achieves a good result and gives the shortest overall response time. Figure 3shows that resource utilization gives the highest value in the proposed algorithm with respect to the other algorithms. concluding that our proposed algorithm increases the performance of the system rather than the HDLB and the Dynamic repartition and also this algorithm achieves our main goals which are increasing the resource utilization, decreasing the fog computing processing time and also decreasing the overall response time

Table 4 . No of tasks and Fog manager processing time

No of tasks	Fog Manager processing time(ms)				
	FCFS	Priority	HDLB	Dynamic repartition	CFBLB
100	0.83	0.61	0.52	0.35	0.25
200	0.87	0.65	0.53	0.45	0.31
300	1.1	0.79	0.62	0.50	0.35
400	1.22	0.98	0.79	0.65	0.42
500	1.32	1.19	0.95	0.75	0.49
600	1.45	1.26	1.10	0.90	0.54
700	1.56	1.4	1.20	0.98	0.72
800	1.68	1.51	1.30	1.22	0.79
900	1.78	1.62	1.40	1.30	0.94
1000	1.9	1.78	1.50	1.40	0.98

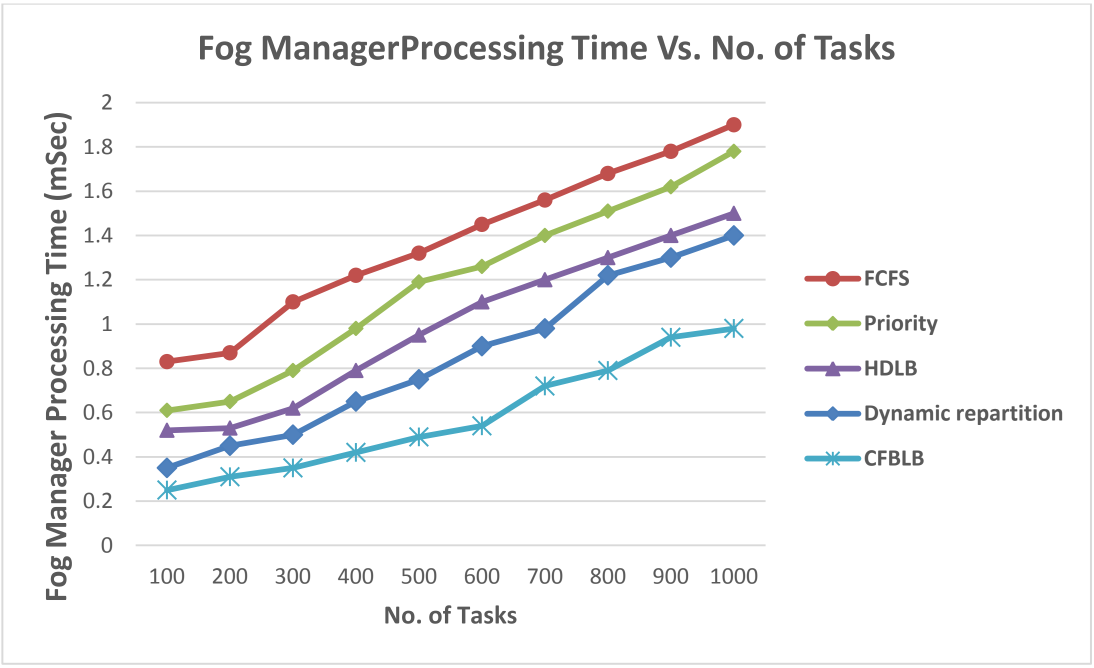

Fig 1: fog manager processing time vs. Number of tasks

Table 5 . No of tasks and Overall response time

No of tasks	Overall response time(ms)				
	FCFS	Priority	HDLB	Dynamic repartition	CFBLB
100	326.1	324.4	321.50	317.50	309.20
200	326.7	323.7	321.70	317.70	309.50
300	327.2	323.8	321.80	317.90	310.60
400	327.9	324.1	322.00	318.20	311.40
500	328.2	324.6	322.10	318.90	311.90
600	328.8	324.9	322.30	319.50	312.20
700	329.3	325.3	322.40	319.90	312.80
800	329.7	325.8	322.50	320.10	313.10
900	329.9	326.1	322.60	320.50	313.30
1000	330.9	326.5	322.70	320.70	313.50

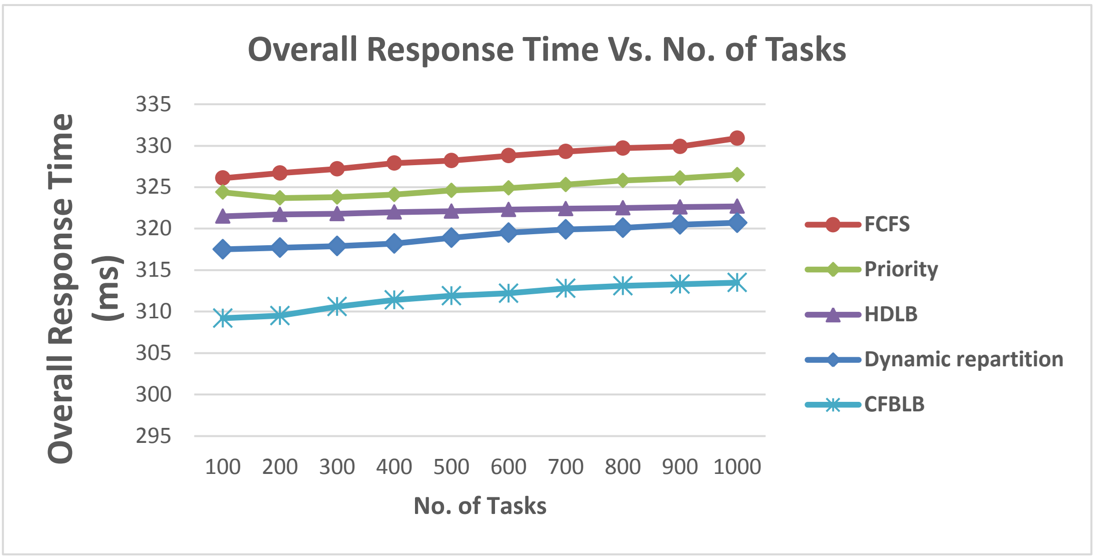

Fig 2: Overall Response Time vs. Number of tasks.

Table 6. No of tasks and Resource utilization

No of tasks	Resource utilization				
	FCFS	Priority	HDLB	Dynamic repartition	CFBLB
100	0.395	0.49	0.59	0.65	0.89
200	0.382	0.48	0.6	0.63	0.84
300	0.37	0.46	0.55	0.67	0.80
400	0.35	0.44	0.53	0.62	0.77
500	0.32	0.43	0.49	0.61	0.72
600	0.31	0.42	0.47	0.6	0.71
700	0.29	0.41	0.45	0.58	0.66
800	0.28	0.39	0.43	0.61	0.69
900	0.26	0.37	0.42	0.62	0.70
1000	0.25	0.35	0.41	0.6	0.68

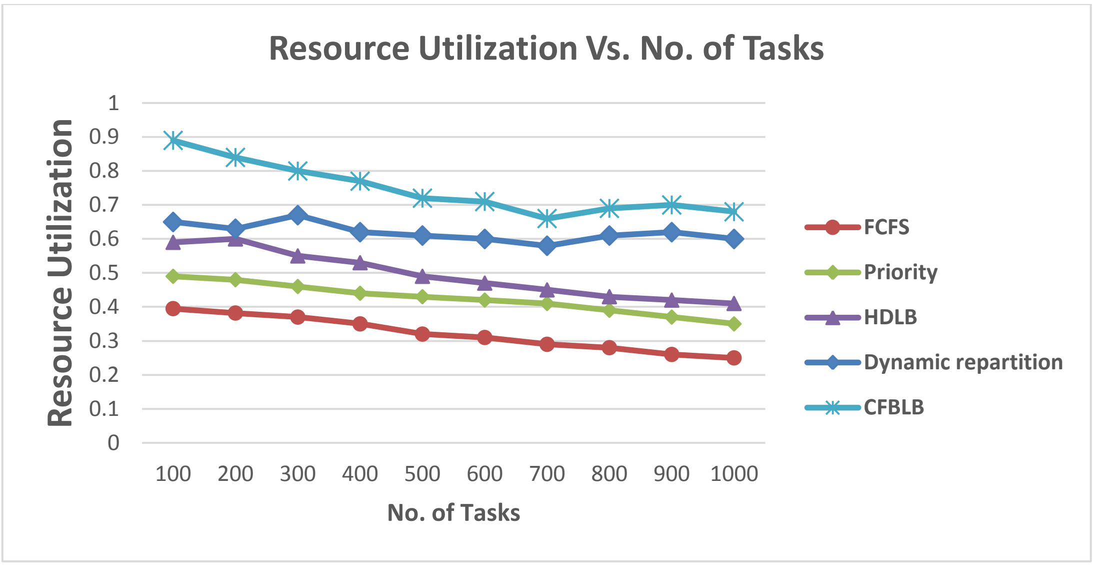

Fig 3: Resource utilization vs. Number of tasks

5. CONCLUSION

Load balancing is a basic challenge in cloud computing and also in fog computing. To achieve high user satisfaction minimize the overall response time and increase resource utilization also decrease the latency. our requirement is to divide the load equally among the nodes to accomplish our goals mentioned before and also to achieve high user satisfaction. In this research, the load balancing algorithm in fog has been introduced by using fuzzy logic. So that, this research reach to a system with a good performance better than the previous algorithms .also getting a good quality of service There is a large area of application in today's industry for fog computing as It is used in various fields such as industrial development, entertainment, medical, traffic, smart houses, etc. so the challenges to increase aside. As future work, the proposed algorithm can be improved by considering a more dynamic situation of the incoming request.

6. REFERENCES

[1] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems.

[2] Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. The University of Maryland at College Park.

[3] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems

[4] Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[5] Sannella, M. J. 1994 Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398.,The University of Washington.

[6] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[7] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[8] Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2005, in press.

[9] W. Pedrycz and F. Gomide, "An introduction to fuzzy sets: analysis and design", complex adaptive systems. MIT Press, (1998).

[10] J. J. Buckley, E. Eslami, and E. Esfandiar." An introduction to fuzzy logic and fuzzy sets" advances in soft computing, Physica Verlag,(2002).

[11] K. Dasgupta, B. Mandal, P. Dutta, "Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach", Elsevier (C3IT) 2012.

[12] International Journal of Computer Applications (0975 – 8887) Volume 69– No.17, May 2013