

An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing

Swati Agarwal

Department of Computer Science and Technology, UPTU University, Lucknow, U.P. India
E-mail: swati.bmas@gmail.com

Shashank Yadav

Department of Computer Science and Technology, Hindustan Institute of Tech. & Mgmt., Agra, U.P., India
E-mail: shashankyadav48@gmail.com

Arun Kumar Yadav

Department of Computer Science and Technology, ITM University Gwalior, M.P., India
E-mail: arun26977@rediffmail.com

Abstract—Cloud computing is a model of sharing computing resources over any communication network by using virtualization. Virtualization allows a server to be sliced in virtual machines. Each virtual machine has its own operating system/applications that rapidly adjust resource allocation. Cloud computing offers many benefits, one of them is elastic resource allocation. To fulfill the requirements of clients, cloud environment should be flexible in nature and can be achieved by efficient resource allocation. Resource allocation is the process of assigning available resources to clients over the internet and plays a vital role in Infrastructure-as-a-Service (IaaS) model of cloud computing. Elastic resource allocation is required to optimize the allocation of resources, minimizing the response time and maximizing the throughput to improve the performance of cloud computing. Sufficient solutions have been proposed for cloud computing to improve the performance but for fog computing still efficient solutions have to be found. Fog computing is the virtualized intermediate layer between clients and cloud. It is a highly virtualized technology which is similar to cloud and provides data, computation, storage, and networking services between end users and cloud servers. This paper presents an efficient architecture and algorithm for resource provisioning in fog computing environment by using virtualization technique.

Index Terms—Cloud Computing, Fog Computing, Resource Allocation, virtual machine and virtualization.

I. INTRODUCTION

Today, cloud computing is an emerging technology that can be defined as a tool which provides enormous benefits to their end users. It is an on-demand service model which is remotely available to users, highly scalable and allocates resources to the users in pay-as-you-go manner. Cloud computing offers various services as on-demand self service, broad network access,

resource pooling, rapid elasticity and measured service. Cloud computing consists of four deployment models. 1). Public cloud 2). Private cloud 3). Hybrid cloud 4). Community cloud [1]. Public cloud provides services for general users over the internet. Private cloud is governed by some private organization for their private use. Hybrid cloud is union of public cloud and private clouds. Community cloud is a kind of private cloud which is shared by some organization having same requirements and objectives. Cloud computing offers service models are Software-as-a-service (SaaS), Platform-as-a-service (PaaS) and Infrastructure-as-a-service (IaaS) model. The combination of above three models is called XaaS that means anything-as-a-Service. Through SaaS service cloud offers readymade applications and required software to the end users over the web. PaaS provides platform to make coding or develop own software and application for the end users. IaaS offers virtual hardware and software like computing resource, storage, networks and operating systems to the client machine.

Cloud computing has number of benefits. Such as reduced cost, increased storage, flexibility, reduced time for implementation and shortened time life cycle. Fig.1. shows the characteristics of cloud applications.



Fig.1. Characteristic of Cloud Application

Beyond the cloud computing benefits and applications, it suffers from some issues and inefficiencies. To resolve the problems of cloud computing, highly virtualized fog computing technology plays a major role, which acts between end user and the cloud data centers hosted

within the internet. The main characteristics of fog computing is to support applications that require low latency, location awareness, distributed geographical distribution and support for mobility and real time interaction [2]. For optimal utilization of resources, server virtualization is one of the challenging issue in cloud computing. Server virtualization attempts to increase resource utilization by dividing individual physical server into multiple virtual servers each with its own operation environment and applications through the magic of virtualization [1]. Each virtual server looks and acts just like physical server, multiplying the capacity of any single machine. Fig. 2, shows the implementation of server virtualization.

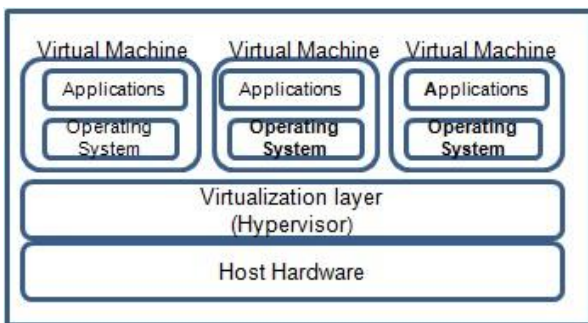


Fig.2. Use of Server Virtualization

We are proposing an efficient architecture and algorithm for resource allocation in fog computing environment. Proposed algorithm solves the issues on the requirement of service level elasticity and availability of resources.

A. Significance of Resource Allocation

Resource allocation is the systematic approach of allocating available resources to the needed cloud clients over the internet. These resources should be allocated optimally to the applications which are running in the virtualized cloud environment [3]. The order and time of allocation of resources is an important input for an optimal resource allocation. The benefit of resource allocation is that the user does not need to expand on hardware and software system. Server virtualization is used to achieve the user satisfaction and resource utilization ratio in high performance cloud computing. Proposed method provides the even distribution of resources across all the clients in the system. In the existing resource allocation techniques there is a problem of over-provisioning and under-provisioning; therefore we are going to propose a better architecture in fog environment to overcome this problem.

B. Issues in Resource Allocation

There are several issues considered in existing resource allocation techniques in cloud computing is discussed here.

Resource management: "how efficiently resources are utilized in the system?" is the primary issue in resource allocation.

Scalability: It is required that an algorithm should be scalable to perform resource allocation with any finite number of clients.

Energy efficiency: It is required to reduce the energy consumption while allocating the resources, so it will improve the performance.

Job scheduling: It is one of the main issues, used to perform to get maximum profit and get minimum execution time.

Load balancing: Overhead associated with the node should be minimized while implementing an effective load balancing methods.

In section II we focus on works related to the topic. In section III, proposed design model for resource allocation in fog environment is presented. We propose our algorithm in section IV and Simulation Setup and Expected Results in section V. Then there is result comparison and result analysis with other algorithms in section VI, VII. In the last, we give the conclusion and future work in section VIII and section IX respectively.

II. RELATED WORK

This section is about to study and review the work of other authors. In resource allocation, many authors already have been proposed their work in cloud computing environment. In our work, we are taking an intermediate layer of fog to make the architecture more efficient. Fog is implemented very close to the end users. Thus fog computing provides better quality of service in terms of network bandwidth, power consumption, throughput and response time as well as it reduces the traffic over the internet.

There are many resource allocation techniques in cloud computing. Network resource allocation strategies and how these strategies can be implemented in cloud computing environment discussed in [3]. There are number of scheduling algorithms for resource allocation, but there is a requirement of effective resource allocation strategy in order to fulfill the demand of users and minimize overall cost for the users as well as for cloud servers. The main objective of resource allocation algorithm is to schedule the VMs on the server that is resides in the data centers. There is a small survey of optimized resource scheduling algorithm, resource allocation strategy based on market, scheduling with multiple SLA parameters, resource allocation rule based model, scalable computing resource allocation, congestion control resource allocation model and service request prediction model.

Authors have focused on two scheduling and resource allocation problems in cloud computing environment [4]. First describe the Hadoop MapReduce and its schedulers and then the second scheduling problem is provisioning virtual machines to resources in cloud. MapReduce is a programming model for processing of very large scale data and originally developed at Google and Hadoop provided the implementation of MapReduce. There are three scheduler available; these are FIFO, fair scheduler

and capacity scheduler. The second scheduling problem is provisioning of virtual machines and assignment of virtual machines to physical machines. Resource allocation scheduling is the main issue in cloud computing. Cloud manager provides virtual resources for efficient system. Numbers of resource allocation methods have been to provide scalability and flexibility and are used in inter-operability between different clouds. Above two scheduling problems can be combined then MapReduce can be used to evaluate virtual machine provisioning.

There is an important relationship between infrastructure component and its power consumption for the implementation in cloud environment [5]. Energy consumption analysis of cloud clusters with the help of cloud cluster node has been proposed. The total energy consumption of the cloud system is a combination of nodes energy consumption, all switching equipment consumption, energy consumption of storage device and the energy consumption of other parts such as fans and current conversion loss. So, we have to set different frequency and modes to save the energy to the CPU, main memory, hard disk and other parts. CPU, memory, storage and network are the four selected parameters to calculate the energy consumption of the cloud environment. To implement the energy saving technique it create a framework to change the mode of cluster components to solve the problem of energy saving. This resource scheduling algorithm is only applicable for the nodes that have the same hardware configuration. The whole algorithm is implemented in three steps such as infrastructure Preparation, job preprocessing and job execution. The objective of the algorithm is to allocate the resources optimally based on energy saving technique. The results of this algorithm find that using only one node to execute the job will waste the energy. So, this algorithm supports parallel process. There is a turning point to achieve a balance between frequencies energy consumption. The limitation of this algorithm is there is no information regarding temperature or fan speed.

An optimized version of FCFS algorithm for resource allocation is implemented to address the main issues of resource allocation scheduling in cloud computing. The architecture is based on client-server format. This architecture has a resource allocator, which allocates and reallocates resource from the client [6]. In optimized version of FCFS, the algorithm gives the resources in parts or put the request in a wait queue and checks that next request can be serviced. This algorithm has deadline constraint or cost constraint, to allocate data to requests. This algorithm has the GUI based facility for the clients to log in and submit the request for resources in pay as you manner. The main advantage of this algorithm is to optimized use of memory and it improves overall performance of server and client.

Authors study different scheduling algorithm in different environment with their respective parameters. To get the maximum profit and to increase the efficiency of work load, scheduling is performed [7]. For this purpose we have different types of scheduling algorithms.

These are FCFS algorithm, round robin algorithm, Min-Min algorithm and Max-Min algorithm, but the most efficient technique is heuristic technique. There are three stages of scheduling in cloud computing i.e. discovering a resource, selecting a target resource and submission of task to a target resource.

Priority based resource allocation method is used for efficient resource allocation in cloud computing environment, without wastage of the resources [8]. The algorithm creates the batches of the user requirements such as amount of processor and memory required by the user. The throughput value is calculated according to the usage of the processor and RAM. Priority is assigned to the resources with the help of Analytical Hierarchy Process (AHP). This algorithm follows two decision making models to assign the priority. These are multi-criteria decision making model and multi-attribute decision making model.

In proposed algorithm two parameters preemptable task execution and SLA multiple parameters are considered. SLA parameters are network bandwidth, required time for CPU and memory usage. Two methodologies are combined in proposed algorithm; SLA based resource provisioning and online adaptive scheduling for preemptable task execution [9]. Here is a cloud provisioning model, to fulfill request from clients is hosted by three layers of resource allocation. The aim of scheduling heuristic is to schedule job on VMs based on agreed SLA objective and creating new VMs. The load balancer algorithm is presented in the paper; it ensures high and efficient resource utilization in cloud environment.

In cloud environment, number of issues has been considered; optimal allocation of resources among them was vital issue. Distributed clouds are same in characteristics to current cloud provider, but there is geo-diversity property in distributed cloud. Network Virtualization (NV) offers the resource allocation geo-distributed scenario [10]. Network Virtualization is important for distributed clouds, because it can easily model the geographical location of the allocated resources. There are four main challenges of distributed cloud, which are discussed in the paper. 1). Resource modeling, 2). Resource offering and treatment, 3). Resource discovery and monitoring and 4). Resource selection.

Dynamic bin packing (DBP) techniques is used for resource allocation in the cloud environment. The new version of this technique is MinTotalDBP problem [11], its main objective is to minimize the total cost. Total cost is due to the bins used over the time. In this algorithm an infinite number of bins are available for packing and all the bins have the capacity & the same cost. Authors defines some notations such as each item (r) to pack is associated with 3- tuple $(a(r), d(r), s(r))$, where $a(r)$ denotes the arrival time, $d(r)$ is departure time and $s(r)$ is the size of r . $I(r)$ denote the time interval in which item r stays in the system. There are number of packing algorithms such as first fit, best fit and any fit algorithm. In the next section, there is the analysis of the

competitive ratios of the packing algorithms for the MinTotal DBP Problem. Then, proposes a new modified first fit packing algorithm that can achieve improved competitive ratios. This algorithm uses the classical first fit algorithm to pack the large items and the small items separately. MFF algorithm achieve a competitive ratio of $8/7u + 55/7$ for the MinTotal DBP problem when u is not known and can achieve competitive ratio of $u+8$, when u is known.

In heuristic ant colony algorithm, there are some drawbacks. It has parameter selection problem and slow convergence speed. To solve the cloud computing resource allocation problem, the authors discussed in [12], proposes an optimized ant colony algorithm which is based on particle swarm. The new algorithm can solve many issues of heuristic ant colony algorithm with the help of cloud computing resource allocation framework. In ant colony optimization based particle swarm algorithm, we have to follow number of steps. First initialize the ant pheromones than update them as required. Then there is the prediction of execution speed of resource node in completion of the task, then setting of the next point of the ant and combinatorial optimization of initial amount of pheromones. Finally, we have a new ant colony optimization algorithm. This algorithm solves cloud computing resource allocation problems. Ants can find more efficient and appropriate virtual node very fast and the user tasks are completed as soon as possible.

Two vital issues in cloud computing for effective allocation of the processes to the associated cloud is discussed in the paper. These issues are overload and under load conditions [13]. Author defined a multi cloud environment. In this cloud environment, each cloud server has some limits in terms of CPU specification and in terms of memory. The objective of the proposed system is to create an Intermediate architecture, schedule the user request, perform the effective resource allocation and define the dynamic approach to perform the process migration from one cloud to another. The overload problem is solved by the concept of process migration. When the request is performed by the user, there are numbers of parameters include. 1). Arrival time, 2). Process time and deadline 3.) Input/output requirement of the processes. Each cloud has number of virtual machines and having its priority. The analysis of the work is done in terms of the wait time and process time.

To make the system efficient, virtualization technology is used to allocate the data center resources dynamically. This method is used in green computing to optimize the number of servers in use. The main focus on multiplexing of virtual resources onto physical hardware [14]. The algorithm aim to solve the two issues for resource allocation: Overload avoidance and green computing. It presents the design and implementation of resource allocation system, to achieve these two goals. There is a brief introduction of skewness which will measure the

performance of server utilization. This algorithm include load prediction algorithm to predict the resource usage. The architecture of the system has hot spot solver and cold spot solver to predict the future resource demand of virtual machines and future load of the physical machine (PM).

Number of researcher has proposed service broker policies for cloud computing system. The main policies are closest data center policy, optimal response time policy and dynamical reconfiguration with load balancing policy [15]. In the cloud computing environment the service broker is an interface between the client and the cloud server to provide the service. The role of these policies is to decide the data center to which the request of end users has to be routed. These policies are used in cloud Analyst simulation tool for executing and analyzing the new policies and algorithms. In the closest data center policy, there is a concept of region proximity to select the data center by the use of network latency parameter. The data center which is closed to the client is selected. In optimal response time policy, the service broker checks all data center availability and estimate the response time for each of them, than provides the best response time to the clients at the time of request is generated. In load balancing policy there is an additional task of dynamical scaling the application deployment. It increases and decreases the number of virtual machines according to the load currently facing.

III. PROPOSED ARCHITECTURE

In cloud computing, the efficient resource allocation is the main objective to get the economic benefits. Resource allocation plays an important role to enhance the performance of the entire system and increase the level of customer satisfaction. Server virtualization is an integral part of resource allocation. Server virtualization improves the resource utilization of the system, overall response time and total estimated cost. The work is based on cloud computing technology integrated with fog computing technology. The main features of fog computing is location awareness, mobility, low latency, and distributed geographically. Fog computing is not the replacement of cloud computing, but it reduces the drawbacks of cloud computing and make it efficient. Our paper is focuses on the efficient resource allocation algorithm and its applicability in fog environment [16]. We first analyses the various existing algorithms of resource allocation after that we have designed the framework to develop this algorithm. This proposed architecture is implemented for solving the problem related to fault tolerance, resource overflow and underflow as given in figure 3.

To handle the problem of resource allocation in fog environment, we have proposed a design model.

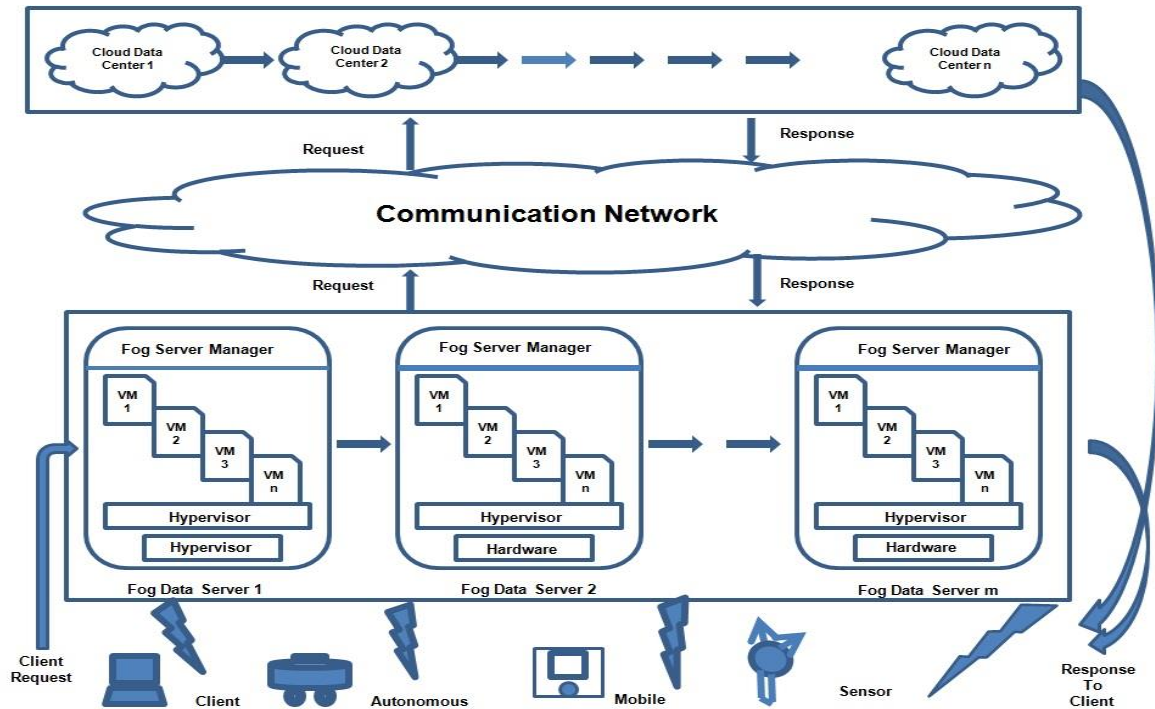


Fig.3. Three Layer Architecture for Resource Allocation in Cloud-Fog System

A. Design Model

This model is designed in a cloud-fog environment. So, the model has three layers named as client layer, fog layer and cloud layer. First, we implement the algorithm in client and fog layer to fulfill the requirement of resources for clients. If no resource is available in fog layer than move the request to cloud layer.

- Step 1. All the data centers will be arranged in fog layer and in cloud layer. Each fog layer has number of fog data server (FS) and cloud layer has number of cloud data center (CS).
- Step 2. Each fog data server (FS) will contains fog server manager (FSM) which will check the availability of the processor and have the responsibility to manage the VM's.
- Step 3. Initially, any client will submit their request to any fog data server (FS), and then fog data server loads the request to its fog server manager (FSM).
- Step 4. Fog server manager (FSM) will process the service request in following conditions.
 - I. If all the requesting processors are available to first fog data server, then it loads the result to the client and client sends an acknowledgement to its fog server manager to update the status of the list.
 - II. If only some requesting processors are available to fog data server, then total task is divided into number of subtasks as per availability.
 - III. If fog data server is in allocated but early release state, then client will wait for minimum time

- IV. If all processors receive at one fog data server but some are failing during processing, then it will again process the request as in II condition.
- V. If no processor is available in fog data servers within its fog cluster, then the request is propagated to cloud data server.

- Step 5. If the sender has not received the result of their request within maximum allotted time, then client will wait for processing.
- Step 6. For further processing client request is propagated to cloud data server (CS).
- Step 7. The cloud data server will provide the processor to client directly to increase the response time and sends an acknowledgement to respective fog server manager.

B. Functional Components

Role of FSM: To enlist all the processors available to the client.

Role of VMs: To handle the request for fog data server and processed the request then provide the result to fog server manager.

Role of FS: It contains one fog server manager and number of virtual machines to handle the request by using the server virtualization technique.

IV. PROPOSED ALGORITHM

This is a proposed algorithm for efficient resource allocation in fog computing environment. This algorithm

shows the pseudo code of Efficient Resource Allocation (ERA) algorithm. The objective of this algorithm is to proper utilization of the resources and to reduce the congestion by using the intermediate fog layer.

Request i_f : The request from users to fog data server.

Request i_c : The request from fog data server to cloud data server.

CS_i: Cloud server process the request.

FS_i: Fog server process the request.

FSM: Each fog server contains fog server manager.

VM_i: Virtual machines at fog server.

M_{st}: Minimum constraint time to release the processors.

Max_time: Maximum allotted time to release the processor.

T_i: Threshold value for each request i_f .

T: Total task

t_1, t_2, t_3, \dots : Sub task of total task.

t_1 : Number of processors available in FS₁.

t_2 : Number of processors available in FS₂.

1. For each request i_f .
2. Each request i_f is send to nearest location fog data server as user's location.
3. Each FS will process the service request.
4. FSM will process the service request in following conditions.
5. IF all requesting processors are available to first fog server.
6. THEN FS loads the result for client and sends an acknowledgement to its FSM.

ENDIF

7. IF only some requesting processors are available to FS.
8. THEN the total task T is divided into number of subtasks as per availability.
9. $T = t_1 * t_2 * t_3 * \dots * t_n$.

ENDIF

10. IF FS is in allocated but early release state.
11. THEN client will wait for M_{st}, and then loads its request to FS.

ENDIF

12. IF all processors receive at one FS but some are fail during processing.
13. THEN GOTO step 9.
14. IF no processor is available in FS within its fog cluster.
15. THEN request i_f is propagated to CS through proper communication network.

ENDIF

16. Calculate the threshold (T_i).
17. IF T_i <= Max_time
18. THEN client will get a message " Wait for processing "

ENDIF

19. For each request i_c .

20. Each request i_c is sending to nearest location CS as FS location.

21. Each CS will process the service request.

22. CS loads the result for client directly.

23. CS sends an acknowledgment to respective FSM.

This algorithm is based on middle layer (fog) which is existing between the clients and clouds. As the request will be performed by the client this request will be accepted by middle layer and not interrupt the clouds. If the request is not processed within its time limit than this request is forwarded to clouds by the middle layer. So this algorithm is efficient to allocate the resources, minimize the response time and maximize the throughput.

V. SIMULATION SETUP AND EXPECTED RESULTS

A. Simulation Tool (CloudAnalyst)

The proposed architecture and algorithm is simulated on CloudAnalyst software. To find out the performance of cloud model, modeling and simulation is the important tool to measure and evaluate the performance of proposed architecture [17]. The main application of CloudAnalyst is to evaluate the performance of the distributed cloud system and gives the result in the form of chart and table [18]. CloudAnalyst is an integral part of CloudSim framework and built on CloudSim tool kit [19]. CloudAnalyst developers can determine the best method for resource allocation of available data center and techniques to selecting data centers to serve requests and cost related to such operations [20]. CloudAnalyst is used for two policies: Load balancing policy and service broker policy. We have used this CloudAnalyst tool to evaluate the performance of the proposed architecture and algorithm.

B. Description for Simulation

To evaluate the performance of the proposed architecture and its algorithm, the parameters of the CloudAnalyst tool need to be set. The objective of this tool is to compare the existing algorithms. We have set the parameters in the configure simulation menu from the CloudAnalyst tool window according to our proposed architecture. We set the parameters for the proposed architecture in two phases and as well as we take the same parameters for other existing algorithms. We take the simulation duration as 60 minutes. Each data center having 5 VM's, Linux operating system, Xen VMM, 0.1 is cost per VM \$/Hr and 0.1 is data transfer cost \$/Gb. These parameters are set for all the configuration setup in first and second phase. In first phase we set the parameters for client-fog layer in closest data center service broker policy, in this we have taken three user bases (UB1, UB2, UB3) and data centers (DC1, DC2, DC3). The location of users has been defined in three different regions (4, 2 and 1) of the world. One data center is located in region 2, second in region 1 and third in region 4. This is the parameters set for client layer and

fog layer. Now, in second phase we set the parameters for fog-cloud layer in closest data center service broker policy. But now the fog layer becomes the user bases and cloud layer become data centers. In fog layer there is one user base UB1 and one data center (DC1) because only one nearest fog server is selected by client if no processor is available to fog layer. There is only one data center in second phase because one fog server sends its request to its nearest cloud server. Location of users has been defined in only one region 1 of the world and one data center is located in region 0. In the third phase, we set parameters for client-cloud layer and not considering the middle layer, in reconfigure dynamically with load balancing policy and in optimize response time. In client layer there are three user bases (UB1, UB2, UB3) and three data centers (DC1, DC2, DC3). Location of users has been defined in three different regions (4, 2 and 1) of the world. One data center is located in region 0, second in region 3 and third in region 5.

C. Output

After performing the simulation for two phases of our proposed architecture and for existing algorithms in cloud analyst the result computed for proposed algorithm is shown in the following figures [4, 5]. We have used the above defined configuration for closest data center policy in first and second phase but in third phase we have used reconfigure dynamically with load balancing policy and optimize response time policy. The detailed displayed result for response time, data center request processing time and estimated cost of first phase has shown in Figures [6, 7, 8], the detailed displayed result for response time, data center request processing time and estimated cost of second phase has shown in Figures [9, 10] and the detailed displayed result for response time, data center request processing time and estimated cost of third phase has shown in Figures [11, 12, 13, 14, 15, 16]. The final output of our proposed algorithm for an efficient resource allocation (ERA) is the summation of the results of first phase and second phase. The efficiency and performance of the proposed algorithm is shown under in given figures.

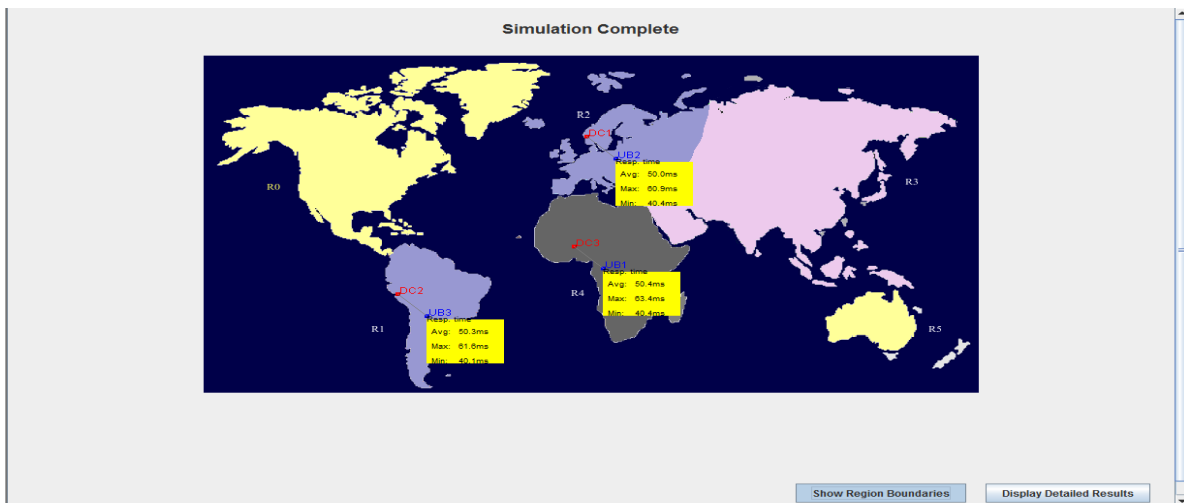


Fig.4. Output Screen of First Phase for Proposed Algorithm after Simulation in Cloud Analyst

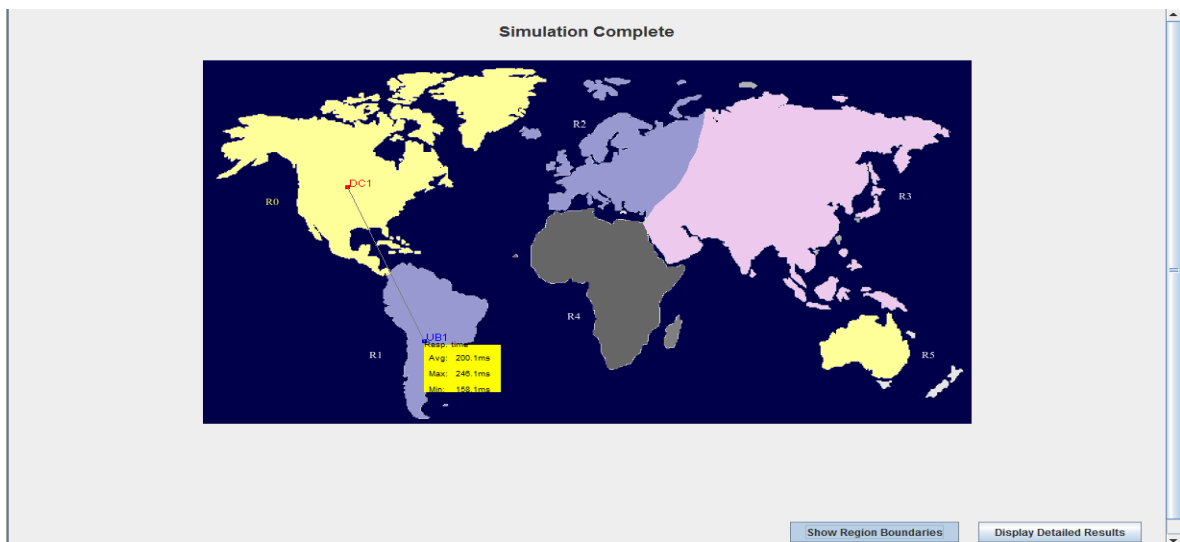


Fig.5. Output Screen of Second Phase for Proposed Algorithm after Simulation in Cloud Analyst

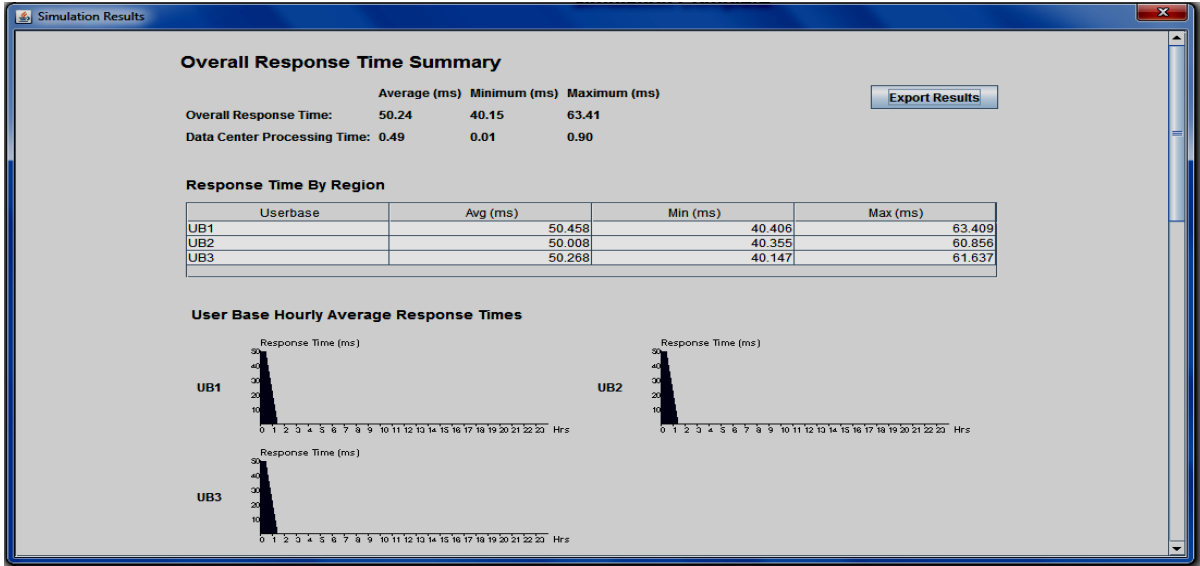


Fig.6. Overall Response Time and Average Response Times of First Phase

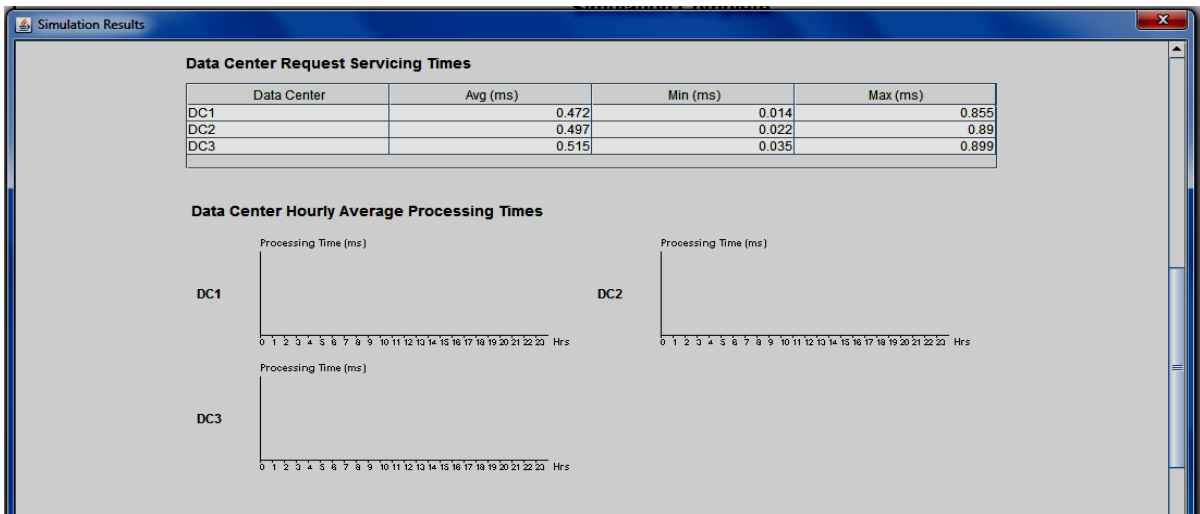


Fig.7. Data Center Request Times and Average Processing Times of First Phase

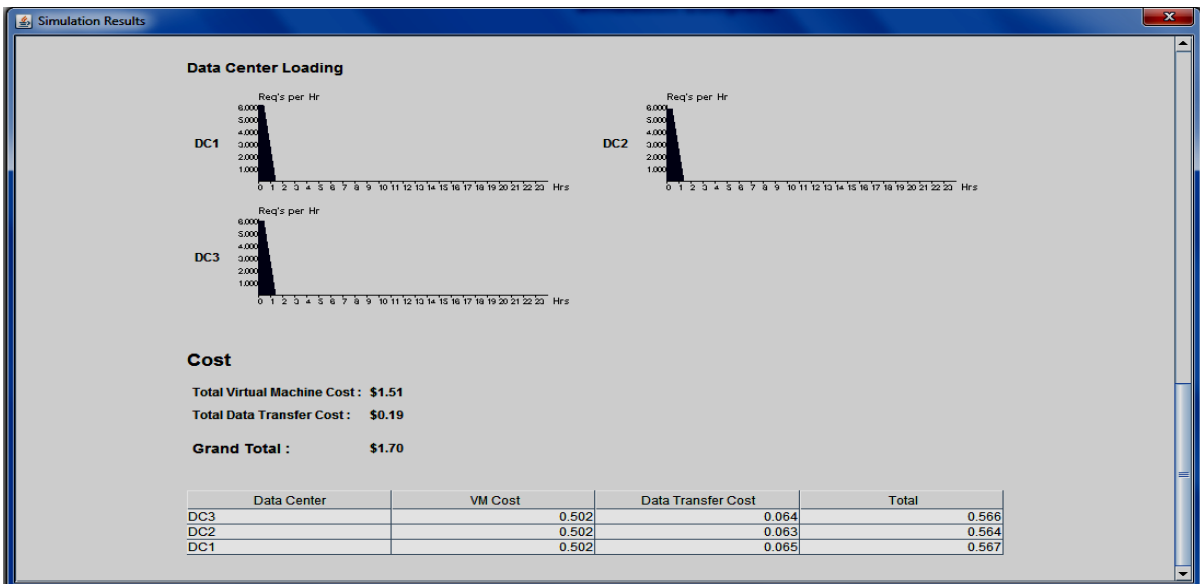


Fig.8. Data Center Loading and Total Cost of First Phase

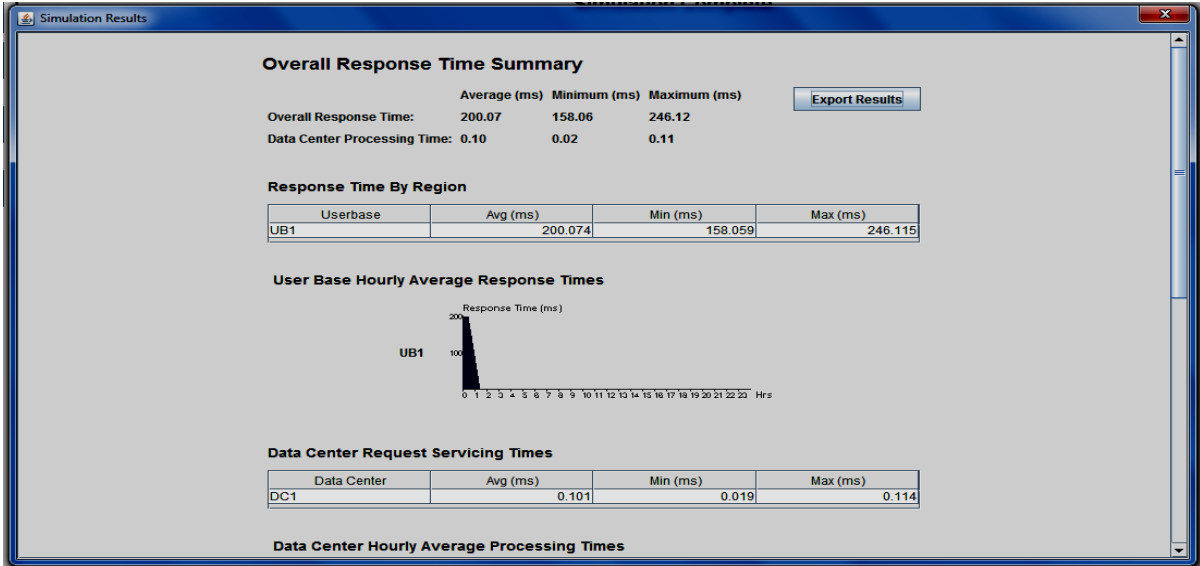


Fig.9. Overall Response Time and Average Response Times of Second Phase

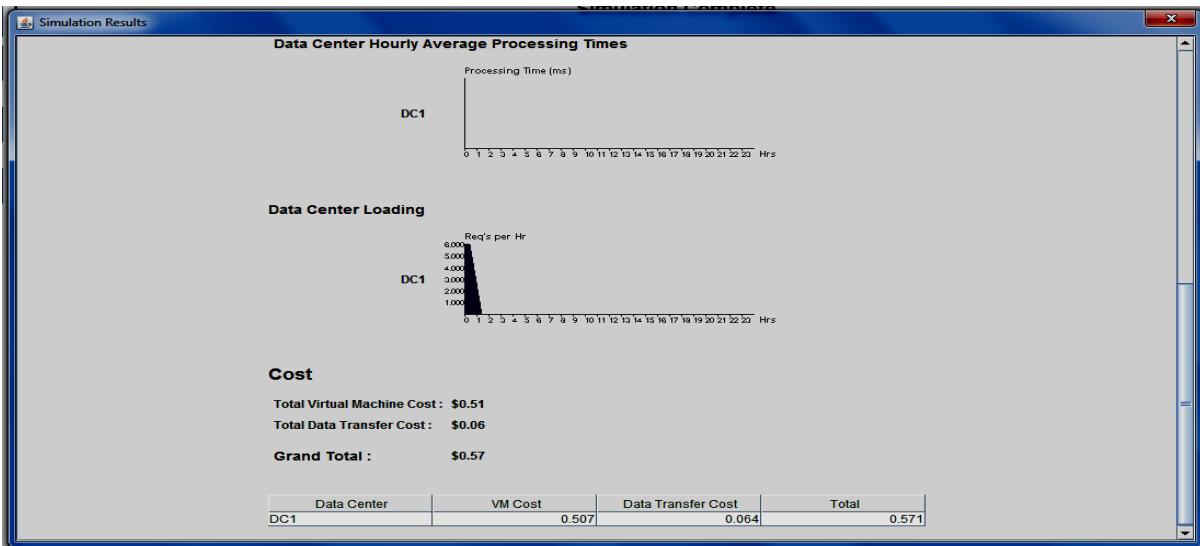


Fig.10. Data Center Request Times and Total Cost of Second Phase

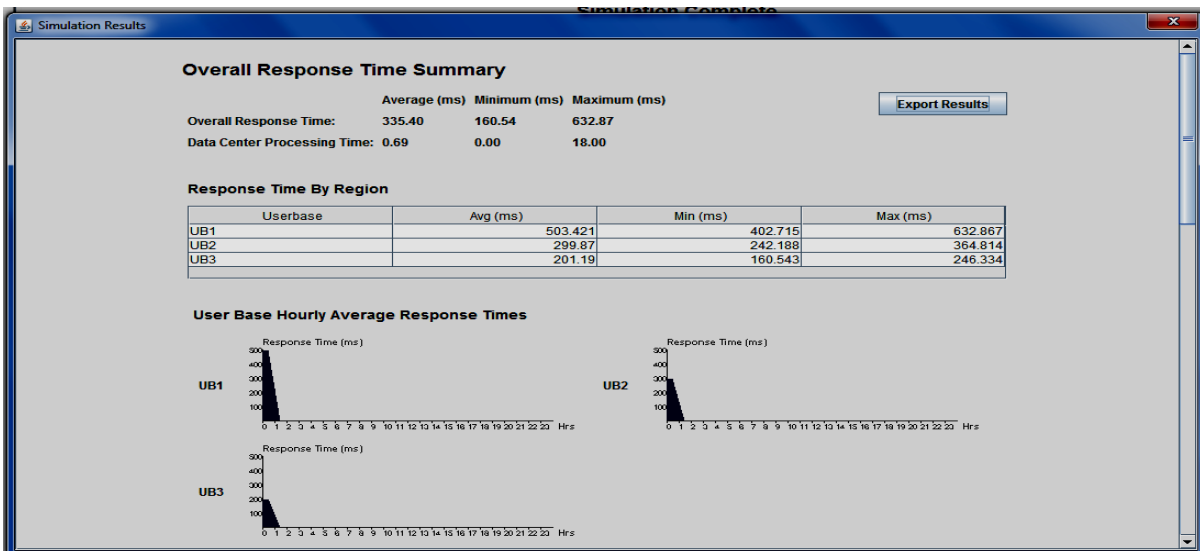


Fig.11. Overall Response Time and Average Response Times of Third Phase using Reconfigure Dynamically with Load Balancing Policy

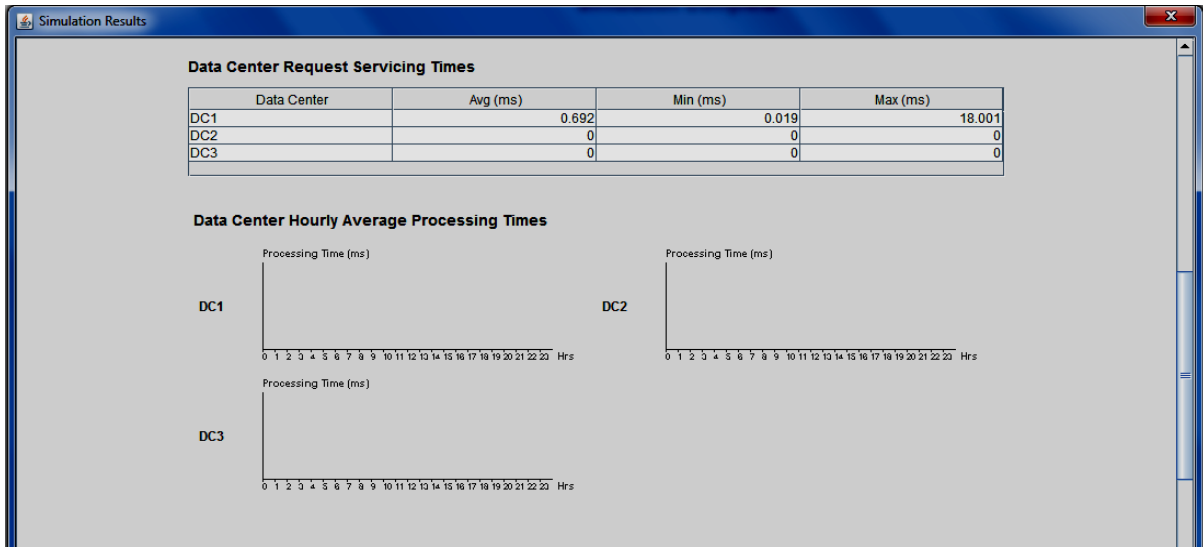


Fig.12. Data Center Request Times and Average Processing Times of Third Phase using Reconfigure Dynamically with Load Balancing Policy

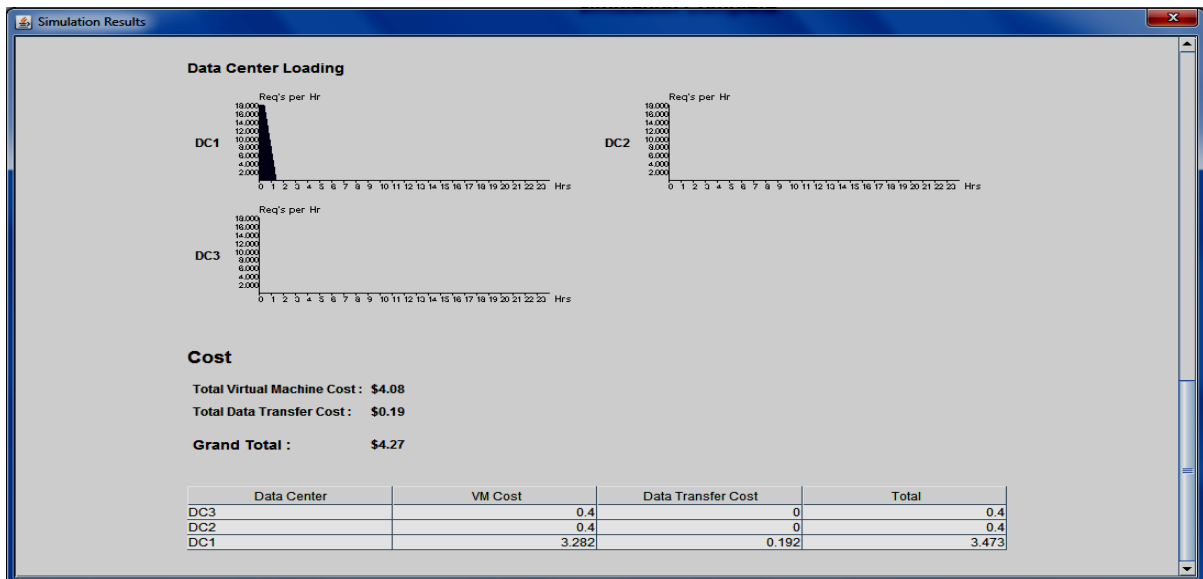


Fig.13. Data Center Loading and Total Cost of Third Phase using Reconfigure Dynamically with Load Balancing Policy

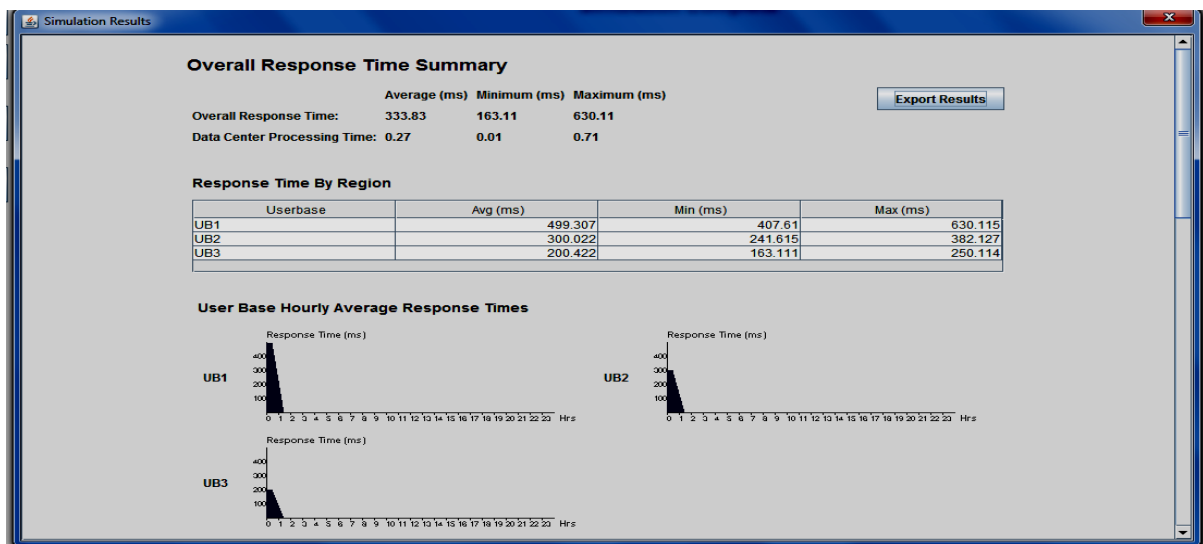


Fig.14. Overall Response Time and Average Response Times of Third Phase using Optimize Response Time Policy

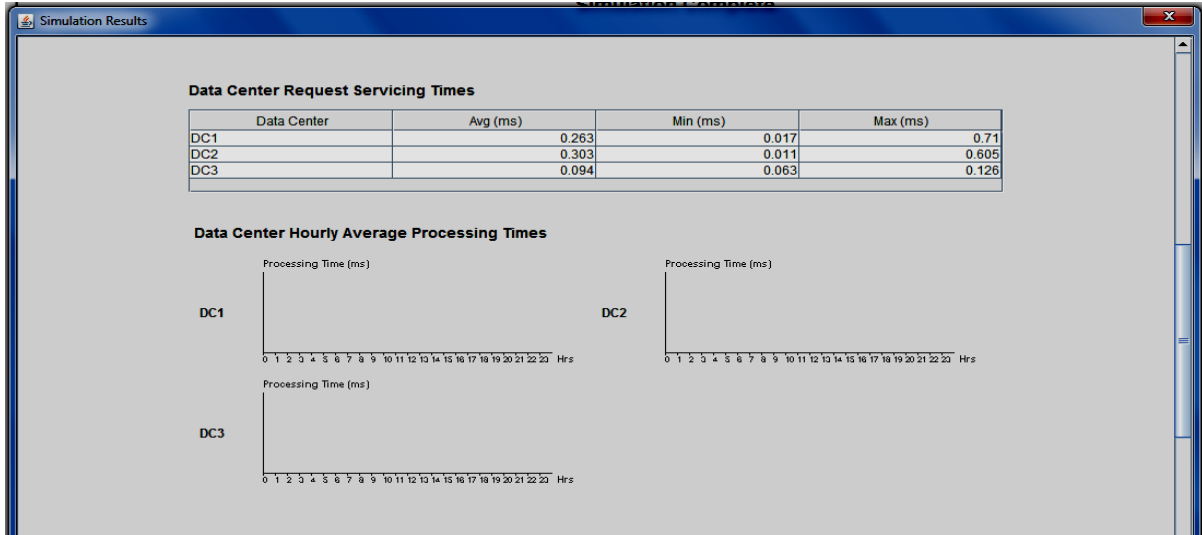


Fig.15. Data Center Request Times and Average Processing Times of Third Phase using Optimize Response Time Policy

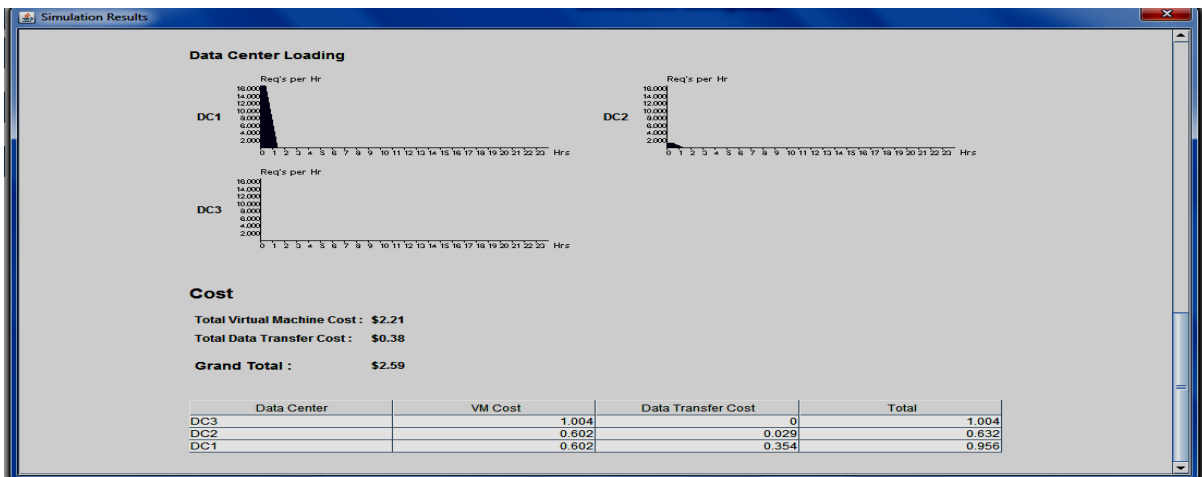


Fig.16. Data Center Loading and Total Cost of Third Phase using Optimize Response Time Policy

VI. RESULT COMPARISON

Finally, an analytical comparison is done between various existing algorithms and the proposed efficient resource allocation (ERA) algorithm is given in table 1. It is shown to be clear that our proposed technique is better

among existing techniques in terms of response time, bandwidth utilization, less power consumption, reduced data traffic over the internet and low latency. We can also remove the problem of server overflow and also found better strategy of resource allocation in fog computing high performance environment.

Table 1. A Comparative Analysis between Existing Algorithms and Proposed Algorithm

Algorithm\Matrices		Three layer processing	Overall Maximum Response Time (ms)	Data Center Processing Time Maximum(ms)	Total Cost (Virtual Machine + Data Transfer(\$))
Existing Algorithms	Reconfigure Dynamically with Load balancing	Client to cloud	632.87	18.0	4.27
	Optimize Response time	Client to cloud	630.11	.71	2.59
Proposed Algorithm	Efficient resource allocation(ERA)	Client to fog	63.41	.90	1.70
		Fog to cloud	246.12	.11	.57
		Overall result	309.53	1.01	2.27

VII. RESULT ANALYSIS

Results shown in table, we can compare overall maximum response time, data center processing time maximum and total cost of virtual machine and data transfer cost between existing algorithms and proposed algorithm. Figure 17(a), 17(b), 17(c) shows comparison of the results. Existing algorithms are reconfiguring dynamically with load balancing (RDLB) and optimize response time policy (ORT). The proposed algorithm is efficient resource allocation (ERA). For final result of the proposed algorithm, we have calculated the result of client to fog layer and then fog to cloud layer. Then the overall result will be the summation of both the results. In our algorithm we are taking the fog layer as the interface between the client and cloud, so mostly all the requests are fulfill by the fog layer and resources are allocated by the fog data server. But in any case if fog data server is not able to provide the resources, then the fog data server request is forwarded to cloud data center for resource allocation.

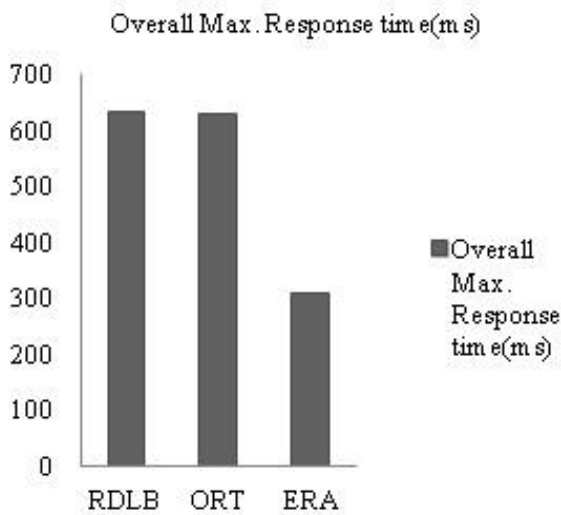


Fig.17(a). Comparison of Overall Maximum response time(ms)

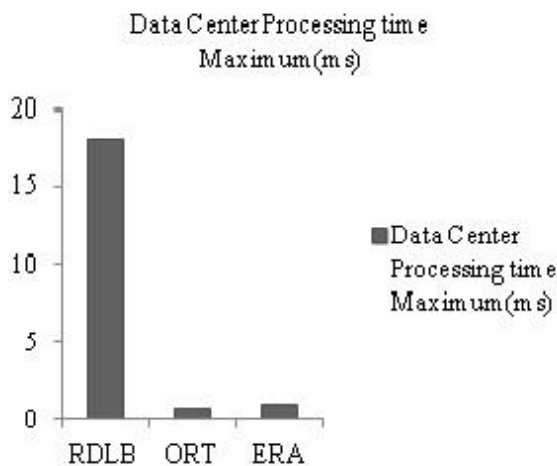


Fig.17(b). Comparison of Data Center Processing time Maximum (ms)

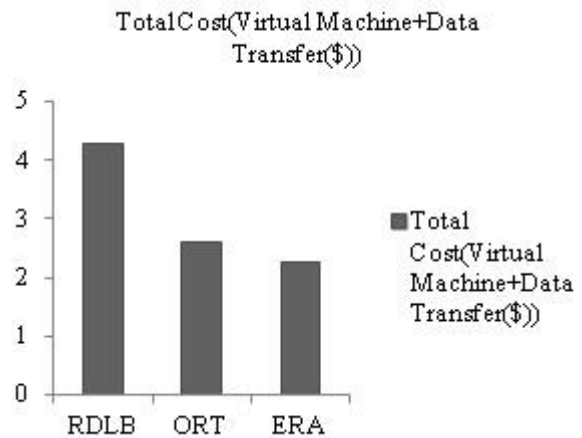


Fig.17(c). Comparison of Total Cost (\$)

Figure 17(a) shows the comparison of overall response time among the algorithms. We have calculated the response time of our proposed algorithm (ERA) is 309.53 ms, it is the summation of two results, two results are 63.41 ms and 246.12 ms. The result for client-fog layer is 63.41 ms and the result for fog-cloud layer is 246.12 ms. The response time of reconfigure dynamically with load balancing (RDLB) is 632.87 ms and the overall response time of optimize response time is 630.11 ms. We can easily determine that the response time of our proposed algorithm is very less, so if client request for processors to fog data server then fog data server allocate that resource to the client immediately.

Figure 17(b) shows the comparison of data center processing time of different existing algorithms and proposed algorithm. We can see from the table 1 that the data center processing time for client-fog layer is .90 ms and for fog-cloud layer is .11 ms. So total data center processing time for our proposed algorithm is 1.01 ms. The data center processing time for reconfigure dynamically with load balancing (RDLB) and optimize response time (ORT) algorithm is 18.0 ms and .71 ms respectively. From the graph we can easily analyses that in comparison to RDLB algorithm our proposed algorithm is far better and in comparison to ORT algorithm, our algorithm is just about equal.

In figure 17(c), our proposed algorithm (ERA) gives the minimum total cost in comparison to both existing algorithms. The total cost is obtained by the cost of virtual machines and data transfer cost used in the system. The total cost of client-fog layer is 1.70 \$ and the fog-cloud layer total cost is .57 \$. So the total of our proposed algorithm (ERA) is 2.27 \$. From the table 1, we can determine that the total cost of reconfigure dynamically with load balancing (RDLB) and optimize response time (ORT) algorithm is 4.27 \$ and 2.59 \$ respectively.

From this data analysis, we can easily find that our proposed algorithm can effectively minimize the overall response time, data center processing time and total cost. So, proposed strategy fully utilizes the bandwidth of the system, provide effective response time, data processing time and reduce data traffic over the internet in fog computing environment to allocate the resources. So we

can define that our proposed algorithm fulfills the requirement of resources to the clients immediately which is the objective of our algorithm or resources are provisioned optimally to the applications which are running in the virtualized cloud environment.

Now, we can easily find that our proposed algorithm can effectively minimize the overall response time, data center processing time and total cost. So, proposed strategy fully utilizes the bandwidth of the system, provide effective response time, data processing time and reduce data traffic over the internet in fog computing environment to allocate the resources.

VIII. CONCLUSION

Fog computing is used in our work because it improve efficiency of cloud computing and reduce the amount of data that needs to be transported to the cloud for data processing and storage. It is inefficient to transmit all the data of Internet of Thing (IoT) and sensors to the cloud, fog computing deal with this problem. So fog is meant to deliver the idea that the benefits of cloud computing can be brought closer to the client. This paper gives the idea about the significance of efficient resource allocation and its related concepts. Cloud computing is difficult to understand without resource allocation because it provides reduced infrastructure cost and elastic scalability. We survey various existing algorithms related to optimal resource allocation and different scheduling techniques. In this paper, an efficient resource allocation architecture and algorithm (ERA) has been proposed and implemented on cloud analyst tool to test the performance of the proposed technique in the fog environment. By implementing the proposed strategy, we find that the proposed strategy can be allocated resources in optimized way and better than existing algorithms in terms of overall response time, data transfer cost and bandwidth utilization in fog computing environment. This document shows the comparison of existing resource allocation strategy with the proposed algorithm in terms of overall estimated response time and cost.

IX. FUTURE WORK

Our future work can be extended towards the run time on demand resource allocation, because our algorithm is not satisfying the requirement of resources during the execution of request. We have only allocated those resources to client which are requesting before processing. This work may also be extended towards other issues and matrices in resource allocation such as job scheduling and load balancing by which the efficiency and performance of the system can be increased in fog computing environment.

REFERENCES

- [1] Kamyab Khajehei, "Role of virtualization in cloud computing", *International Journal of Advance Research in Computer Science and Management Studies* Volume 2, Issue 4, April 2014.
- [2] Ivan Stojmenovic, sheng Wen, "The Fog Computing Paradigm: Scenarios and security issues" *Proceedings of the IEEE International Federated Conference on Computer Science and Information Systems*, 2014, pp. 1-8.
- [3] N.R.RamMohan, E.Baburaj "Resource Allocation Techniques in Cloud computing-Research Challenges for Applications", *Proc. Of the IEEE Fourth International Conference on Computational Intelligence and Communication Networks*, 2012, pp. 556-560.
- [4] Eman Elghoneimy, Othmane Bouhali, Hussein Alnuweiri, "Resource Allocation and scheduling in Cloud Computing", *Proc. Of the IEEE International Workshop on Computing, Networking and Communications*, 2012, pp. 309 – 314.
- [5] Liang Luo, Wenjun Wu, Dichen Di, fei Zhang, Yizhou yan, Yaokuan Mao, " A Resource scheduling algorithm of cloud computing based on energy efficient optimization methods", *Proc. Of the IEEE International Green Computing Conference (IGCC)*, 2012, pp. 1 – 6.
- [6] Aditya Marphatia, Aditi Muhnot, Tanveer Sachdeva, Esha Shukla, Prof. Lakshmi Kurup, " Optimization of FCFS Based Resource Provisioning Algorithm for Cloud Computing" , *IOSR Journal of Computer Engineering (IOSR-JCE)* e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 10, Issue 5 (Mar. - Apr. 2013), PP 01-05.
- [7] Shimpy, Jagandeep Sidhu, " Different Scheduling Algorithms In Different Cloud Environment", *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 3, Issue 9, September 2014.
- [8] Savani Nirav M, Prof. Amar Buchade, "Priority Based Allocation in Cloud Computing", *International Journal of Engineering Research & Technology (IJERT)* ISSN: 2278-0181 IJERTV3IS051140 Vol. 3 Issue 5, May – 2014.
- [9] Chandrasekhar S. Pawar, Rajnikant B. Wagh, " Priority Based Dynamic resource allocation in Cloud Computing with modified Waiting Queue", *Proceeding of the IEEE 2013 International Conference on Intelligent System and Signal Processing(ISSP)* Pages 311-316.
- [10] Endo P.T., de Almeida Palhares, A.V., Pereira, N.N., Goncalves, G.E., "Resource allocation for distributed cloud: concepts and research challenges", *Proceeding of the IEEE Network* Vol. 24, Issue 4, July 2011, pp.42-46.
- [11] Yusen Li, Xueyan Tang, Wentong Cai, " Dynamic Bin packing for on demand cloud resource allocation ", *Proceedings of the IEEE Transactions on Parallel and Distributed Systems* ,2015, Paged 1-14.
- [12] Zhengqiu Yang, Meiling Liu, Jiapeng Xiu, Chen Liu, " Study on cloud resource allocation strategy based on particle swarm ant colony optimization algorithm" ,*IEEE 2nd International Conference on Cloud Computing and Intelligent Systems (CCIS)*, Nov. 2012 ,pp. 488 – 491.
- [13] Harpreet Kaur, Maninder Singh, " A Task Scheduling and Resource Allocation Algorithm for Cloud using Live Migration and Priorities", *International Journal of Computer Applications (0975 – 8887)* Volume 84 – No 13, December 2013.
- [14] Zhen Xiao, Senior Member, IEEE, Weijia Song, and Qi Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment", "*IEEE Transactions on Parallel and Distributed Systems*, Vol.24, No. 6, June 2013, pp. 1107-1117.
- [15] Kunal Kishor, Vivek Thapar, " An Efficient Service Broker Policy for cloud computing environment", *International Journal of Computer Science Trends and Technology (IJCTST) – Volume 2 Issue 4, July-Aug 2014.*

- [16] Swati Agarwal, Shashank Yadav, Arun Kumar Yadav, "An architecture for elastic resource allocation in Fog computing", *International Journal of Computer Science and Communication*, Vol. 6 Number 2 April - Sep 2015 pp. 201-207.
- [17] Rahul Malhotra, Prince Jain, "Study and Comparison of Various Cloud Simulators Available in the Cloud Computing", *International Journal of Advanced Research in Computer science and Software Engineering* ISSN: 2277 128X Vol. 3, Issue 9, September 2013.
- [18] Bhatiya Wickremasinghe "Cloud Analyst: A CloudSim Based Tool For Modeling And Analysis Of Large Scale Cloud Computing Environments. *MEDC Project*", Report 2010.
- [19] Hetal V. Patel, Ritesh Patel, "Cloud Analyst: An Insight of Service Broker Policy", *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 4, Issue 1, January 2015.
- [20] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling And Simulation Of Scalable Cloud Computing Environments And The CloudSim Toolkit: Challenges And Opportunities," *Proc. Of The 7th High Performance Computing and Simulation Conference (HPCS 09)*, IEEE Computer Society, June 2009.

Authors' Profiles



Swati Agarwal is pursuing M.Tech in Computer Science and Engineering from Uttarpradesh technical university, Lucknow, India. She received degree of B.Tech in Computer Science and Engineering from Sachdeva Institute of Technology and Management, Mathura, U.P., India in 2005.

Her research interest includes cloud computing and fog computing.



Shashank Yadav has done B.Tech. (Computer Science & Engineering) from Uttar Pradesh Technical University, Lucknow, Uttar Pradesh, India; M.Tech (Computer Science & Engineering) from ABV Indian Institute of Information Technology and Management, Gwalior, Madhya Pradesh, India. Presently he is working as Assistant Professor in the Department of Computer Science & Engineering at Hindustan Institute of Technology and Management, Agra, Uttar Pradesh, India.

His research interest includes Algorithms, Data Structure, Data Mining, Cloud Computing and Fog Computing.



Arun Kumar Yadav has done B.E. (Computer Science & Engineering) from G.B. Pant Engineering College, Pauri Garhwal, M.Tech (Information Technology) from Sam Higginbottom Institute of Agriculture, Technology & Sciences, Allahabad and pursuing Ph.D. in Computer Science and Engineering from Uttarakhand Technical University, Dehradun. Presently he is working as Associate Professor in the Department of Computer Science & Engineering at ITM University Gwalior, Madhya Pradesh, India.

His research interest includes Distributed Database Security, Cloud Computing and Fog Computing. He is a senior member of IACSIT and IAENG Technical Societies.

How to cite this paper: Swati Agarwal, Shashank Yadav, Arun Kumar Yadav, "An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing", *International Journal of Information Engineering and Electronic Business(IJIEEB)*, Vol.8, No.1, pp.48-61, 2016. DOI: 10.5815/ijieeb.2016.01.06