International Journal of Computer Applications (0975 – 8887)
Volume 105 – No. 7, November 2014

RPOAWLB: Resource Provisioning Optimization Approach based on RPOA with Load Balance

Noha El.Attar
Ras-Elbar High Institute for Specific Studies and Computer
Ras El- Bar, Damietta, Egypt

Wael Awad
Dep. of Math. and Computer Science
Faculty of Science
Port Said University PortSaid, Egypt

Fatma Omara
Dep. of Computer Science
Faculty of Computers and Information
Cairo University, Cairo, Egypt

ABSTRACT
Consummation of the service providing with fulfilling the users QoS requirements and satisfying the service's provider, is considered a vital challenge for resource provisioning process. The QoS requirements usually have two main parameters that are needed to be optimized (i.e. minimizing service's response time, and total paid cost). On the other hand, the provider has the right to gain a profit for delivering the service. A modified algorithm based on an existed one called RPOA algorithm has been introduced to solve this problem. This modified algorithm is called The RPOA With Load Balance (RPOAWLB). According to the RPOAWLB algorithm, good utilization and load balance between resources capacities have been satisfied. This has been done by maximizing the resources utilization, and free up the low loaded resources. Therefore, this proposed algorithm provides the provider a chance to reallocate the free resources with new workloads.

Keywords
Cloud computing, Resource Provision, Load Balance

1. INTRODUCTION
Resource provisioning is the task of mapping the available resources in the data centers to different entities of cloud on demand basis [6]. The resource allocation process must be done in such a manner that satisfies both the service's customer and provider. This can be done by minimizing both response time and cost to fulfill the user's Quality of Service (QoS), and in the same time, making a good utilization of the available resources to maximize the provider profit.

Identifying the proper resource from the available pool of resources is considered the fundamental problem in the Cloud computing provisioning process. The main issue in the provisioning policy is how to spread the applications load on proper Cloud resources to achieve the optimization objective of the customer and the provider satisfaction [8].

Achieving the QoS requirements is the core of the customer satisfaction. It usually depends on two essential factors which are; the Response time, and the actual cost. On the other hand, the service provider needs to gain profit after submitting the demanded service. This can be done by determining the right amount of the needed resources to execute the service and guarantee resource utilization [3].

This paper is considered an extension of the Resource Provision Optimal Algorithm (RPOA) based on Particle Swarm Optimization (PSO) which has been introduced to find the near optimal resource utilization with considering the customer budget, the data transfer time, and suitable for deadline time constrain [7,8]. The RPOA algorithm has focused on optimizing both time and cost of the provided service to meet the customer's QoS requirements, but it didn't concern the service provider satisfactions. The main issue of the provider satisfaction is how to increase his profit by guaranteeing the load balance of the resources.

The rest of the paper is organized as follows; section 2 displays the related works about the resource provisioning problems in the cloud computing. The problem of resource provision with concerning the resources load balance in cloud computing environment is discussed in Section 3. Section 4 presents the resource provisioning optimization problem and the modification of the existed RPOA model by considering the resources load balance. Section 5 includes the modifications, implementation, and evaluation of the extended RPOA model with considering the resource load balance using CloudSim toolkit. The conclusions and future work are presented in section 6.

2. RELATED WORK
Efficient utilization of the resource and achieving QoS requirements are considered the core of resource provisioning techniques. There are many resource provisioning techniques that are based on static or dynamic provisioning and each one has its own pros and cons. The main common provisioning techniques are used to improve the QoS parameters for the cloud user (i.e. minimizing cost and time), and/or maximize the revenue for the Cloud Service Provider [4].

Chaisiri, et. al. [12] have proposed an Optimal Cloud Resource Provisioning (OCRP) algorithm to minimize the total cost for provisioning resources in a certain time period. The demand uncertainty from the cloud consumer side and price uncertainty from the cloud providers are taken into consideration to adjust the tradeoff between on-demand and oversubscribed costs to reach to an optimal decision. This optimal decision is obtained by formulating and solving a stochastic integer programming problem with multistage recourse.

Another resource allocation algorithm for optimizing the overall cloud operation cost is proposed by Woo; S., and Mirkovic; J. [14]. They formulate the resource allocation problem by finding a set of resources for a given application workflow which meets the set of SLA constraints. The best resource allocation path is found buy using specific steps. First, all possible resource allocations for each transaction are computed that meet their respective SLA constraints. Then, ordering these solutions by cost within each transaction, and building the final allocation by exhaustively combining allocations for each transaction. This exhaustive search is progressed from the cheapest to the most expensive solution and stops as soon as finding the best solution that meets all of the constraints.

Another issue for scheduling the Cloud resource is to satisfy a minimum response time. Chen; Y., and Tsai; S. [15] have developed a new version of PSO called Discrete Particle Swarm Optimization (DPSO) algorithm for tasks allocation. In this allocation method, the solutions are encoded in a t×p matrix, in which p is the number of available processors at the time of allocation and t is the number of required tasks. All the elements of the matrices have either the value of 0 or 1, each row has only one element with value ‘1’ and others are ‘0’, because each task should be allocated to exactly one processor. The experimental results showed that the DPSO algorithm has the ability to obtain faster and feasible allocation because of its speed of convergence.

Banerjee et. al [11] have proposed an initial heuristic algorithm based on Ant Colony Optimization (ACO) approach for the various service allocations and scheduling mechanism in the cloud paradigm. . The main target of the proposed optimization method is to minimize the makespan of the cloud computing based services.

The trend of reducing the resource computing power and load balancing is a different research direction for resource provisioning. As it is well known in the Cloud computing environment, multiple cloud users can request number of cloud services simultaneously, and the provisioning process must provide them with appropriate resources that execute user’s request efficiently to satisfy their needs. Resource provisioning is usually based on virtual machines to handle the user request, but sometimes a service interruption occurs when the virtual machine is overloaded. Suganya; S., et. al. [13] has tried to overcome this problem by migrating the tasks to the next low loaded virtual machine without service interruption. This live migration of the virtual machine allows more and more physical servers to be turned off. This made the total physical machine’s load is balanced, which led to better achievement of energy efficiency for data centers, and minimize the capital cost, as well as, the cost of power consumption.

Some researchers have tried to consider two criteria for resource provisioning optimization. Almeida et. al. [5], have presented a self-managing technique that jointly addresses the resource allocation and admission control optimization problems in virtualized servers. The proposed solution is considered the provider's revenues and resource utilization, and on the other hand, it concerns the customer QoS requirements which are specified in terms of the response time of individual requests. Results show that this solution can satisfy QoS constraints while still yielding a significant gain in terms of profits for the provider, especially under high workload conditions, if it is compared with the alternative methods.

El-Attar; N., et. al [7] have introduced a Resource Provision Optimal Algorithm (RPOA) based on Particle Swarm Optimization PSO. This algorithm concerns two main factors of the user’s QoS requirements which are (response time, and paid cost). The main issue of the RPOA algorithm is to minimize both cost and time of the provided service, and to find the best workload resource map that is commensurate with the customer defined budget and suitable for the service deadline time.

3. RESOURCE PROVISIONING AND RESOURCES LOAD BALANCING

Partitioning the available quantity of physical resources into multiple virtual resources is a significant process in resource provisioning algorithm. It is used to create independent virtual machines that are allocated on a varying fraction of capacity depending on the service's QoS requirements and current workload [5].

Distributing the workloads on the available resources must be in an efficient way to achieve the most possible profit to the service’s provider, in the same time, achieve the user’s QoS parameters. Good resources utilization is one of main criteria that must be considered in provisioning process, in service hosting, clustered servers run components of continuously running services on dedicated servers which achieve performance isolation and leads to low utilization. So, there is a strong incentive of sharing the cluster resources among services through the hosting platforms in the cloud environment [1]. The challenge is that how to allocate appropriate resource shares to required services and maximize the platform utilization without wasting any resources quantity as much as possible and without any overload in resource nodes.

4. RESOURCE OPTIMIZATION PROBLEM DEFINITION

The optimization function role is clearly evident in the resource provisioning in the cloud computing systems. The optimization problem depends on achieving the availability by allocating the appropriate resources that fulfill the QoS requirements (i.e. optimizing both response time and paid cost depending on the defined deadline time, and budget). Also, the optimization problem considers how to determine the right amount of resources that are required for the services execution.

This paper is considered an improvement of the RPOA algorithm that has been presented in [7]. The RPOA algorithm is based on optimizing both time and cost to achieve the QoS requirements without considering the load utilization which raises the service provider profit.

According to the existed RPOA algorithm, the available resources pools are distributed over different data centers on widespread different geographical areas. Each pool includes a specific resource type (e.g., computing, memory and storage resources). Each resource has its own configuration of a certain size of memory and storage and a certain power consumption of computing. Similarly, the amount of the computing power, memory or storage size which each workload needs certainly has to be known. The RPOA algorithm has been implemented by assuming the service requests memory resource, and all of the workloads have the same priority, and in the same time, they are independent. By considering independency of workloads, the measurement of the resources cost will be more accurate. Also, the RPOA algorithm provides the user the advantage of deciding how much he can pay according to the amount of usage he needs depending on his available budget [7].

Therefore, the RPOA algorithm has concerned only QoS requirements. But, the resource usage utilization is another important issue that has to be considered. The problem of load balancing and good utilization of the resources is not only an important issue for the service's provider, but it is also vital to service's requesters.

The proposed RPOAWLB algorithm tries to maximize the usage of the loaded resources and frees up the light loaded resources. This will solve the problem of resources wastage as much as possible, and will provide free resources with much full capacity which can be used to achieve other services requests.

4.1 The Proposed RPOAWLB Algorithm

According to the existed RPOA algorithm, it is assumed that there are 'm' numbers of the available resources; every resource 'j' has a defined fixed quantity of memory 'Q', and 'n' static workloads. Every workload 'i' needs a specific quantity 'q' which is not changed during the runtime. Each resource has a definite price 'cj' denoted by dollar and has default execution time 'tj' denoted by second. Every customer can decide how much he can pay for each workload 'Bi' [7]. Another constraint is to ensure that for all workloads, the available resources must not be less than the total amount of the required demands, and every workload quantity must not be more than the resource capacity to which it is assigned. Finally, each resource can execute more than one task at the same time, but every task can not be assigned to more than one resource.

Depending on RPOA implementation, the 'n' user's workloads which are denoted with a vector q=[q1,q2,q3,q4,……qn] are associated with the available 'm' resources with vector Q=[Q1,Q2,Q3,……Qm] is given by matrix qnm(n,m).

The modified RPOAWLB algorithm has been developed using three stages. The first stage is concerned with how to distribute the coming workloads on the available resource pool. The second one is responsible for checking if the used resource will fulfill the QoS requirements. The last stage is used to map the requests on the resources depending on the RPOA algorithm optimization. Particle Swarm Optimization (PSO) begins with initializing the load of user requests on the available resources, with two constraints; the capacity of each resource 'Qj' has to be more than the uploaded quantity 'qi', and the total quantity of workloads $\sum q_i$ has to be less than the total available resources $\sum Q_j$. For every user request 'qi', the distribution begins with the nearest resource 'Qj' to the user which achieves the QoS requirements depending on the RPOA utilization functions [7]. After loading the current request, the algorithm checks the unutilized quantity of resource Qj, and reallocates the free capacity to the next suitable requests. On the other words, the RPOAWLB algorithm is based on maximizing the load of the highly used resources and freeing up the low loaded resources to be able to serve other coming workloads. The illustration of the proposed RPOAWLB algorithm is presented in Figure 1.

For each particle *[T_i,R_j]*

  Initialize position and velocity

  Initialize *Pbest* and *Gbest*

End

Allocate every T_i to the nearest R_j

For each R_j

 For each T_i

  Calculate the unutilized quantity '*UZ*' by *Ti/Rj*

  Move to next task (T_{i+1})

  If T_{i+1} is more than *1- UZ*

   Allocate T_{i+1} to R_j

  End if

 Next *i*

 For each T_i

Calculate the actual spending time 'Tt_i' for every task by

$$Tt_i = \sum_{i=1} q_i / Q_j * t_j$$

Calculate the actual cost 'T_{ci}' for every task depending on spending time

$$Tc_i = \sum_{i=1} c_i * T_{tj}$$

Calculate the cost budget 'TB_i' depending on spending time

$$TB_i = \sum_{i=1} B_i * T_{tj}$$

Find the cost and time elasticity relations by

$$\rho_{ci} = \frac{T_{Bi} / T_{ci}}{T_{ti} / D_t} \quad \text{and} \quad \rho_{Ti} = \frac{T_{ti} / D_t}{T_{Bi} / T_{ci}}$$

Calculate the fitness function for T_i by

$$\ell_i = \frac{\rho_{ci} T_{ci} + \rho_{Ti} T_{ti}}{\rho_{ci} + \rho_{ti}}$$

find the actual paid cost by

$$CT_i = \ell_i * Tt_i$$

If CT_i is less than *Pbest*

 Set current value of CT_i as new *Pbest*

End if

Choose the particle with the best *Pbest* fitness of all particle as *Gbest*

Update particle velocity

Update particle position

  Next *i*

  End for

Next *j*

End for

Fig 1: The RPOA with Load Balance (RPOAWLB) Algorithm

5. THE IMPLEMENTATION AND EVALUATION OF THE RPOAWLB ALGORITHM

Spreading the data centers through geographical areas to build a cloud system and then distributing the workloads for the real applications on them are so expensive and time consuming processes. So, evaluating the performance of various usage and resource scenarios in a repeatable and controllable manner becomes almost impossible process. Therefore, most of the researchers have to use a simulation frame work like CloudSim [9].

CloudSim with CloudAnalyst interface is used as a simulated framework to evaluate the RPOAWLB algorithm performance. CloudSim is a framework that enables modeling, simulation the Cloud computing infrastructures [9]. CloudSim framework is built on top of GridSim framework which is developed by the GRIDS laboratory, and it is written in java 6.1[2] [10]. CloudAnalyst is a GUI simulator built on CloudSim framework which gives the ability of quickly and easily changing of the needed parameters to be assumed in the simulation that provides a high degree of configurability and flexibility [2].

The RPOAWLB algorithm is implemented using CloudAnalyst simulator, with some defined parameters which are; 1) the user requests for memory on the available physical resources in the data centers, 2) all of the physical resources have the same configurations (i.e. (x86) architecture, with Linux operating system, and have Xen virtual machines with time shared allocation policy).

The simulation is conducted with fifty of independent jobs and ten available data centers. Every job needs a specific amount of memory with predefined cost budget and deadline time. On the other hand, every data center has a pool of physical resources, with heterogeneous configurations of the amount of memory, but all of them have the same capacity of computing power, and storage space. Also, every resource has its predefined execution time and cost. Finally, the available bandwidth and its transmission cost and time are defined with the type of network communication and the available networks bit rate.

A sample of the jobs requests with details of the needed amount of memory, the available budget, and the deadline time is presented in Table 1. . Also, the configuration details of every data center are given in Table 2.

Table 1. Sample of twenty jobs request details

	Region	Request size per/G.Byte	User Budget per /$	Peak Hours Start (GMT)	Peak Hours End (GMT)	Deadline Time per/ sec
UB1	2	976.56	7.2	3	9	5.3
UB2	0	439.45	17.5	3	9	8.3
UB3	2	5859.38	42.7	3	9	34.8
UB4	5	3369.14	25.5	3	9	14.9
UB5	0	52734.38	384.2	3	9	394.7
UB6	2	2929.69	18.9	3	9	13.7
UB7	3	1464.84	18.9	3	9	8.9
UB8	1	3906.25	519	3	9	104.5
UB9	4	5468.75	195.8	3	9	75
UB10	2	8691.41	57.3	3	9	26.4
UB11	0	634.77	5.9	3	9	5.8
UB12	3	439.45	5.6	3	9	2.9
UB13	4	146.48	6.9	3	9	4.8
UB14	1	97.66	24.2	3	9	4.3
UB15	5	537.11	11.8	3	9	6.5
UB16	0	761.72	13.8	3	9	6.3
UB17	4	878.91	31.1	3	9	15
UB18	3	732.42	9.4	3	9	4.2
UB19	3	468.75	7.4	3	9	5
UB20	0	976.56	1.8	3	9	1.1

Table 2. Data canters configuration details

	Region	No. Physical Units	Total Capacity	Memory Cost $/sec	Trans. Cost $/sec	Proc. Cost $/sec
DC1	0	2	60000	0.7	1.1	0.4
DC2	1	3	8976.563	0.9	1.1	0.4
DC3	3	4	4843.75	0.5	1.1	0.4
DC4	5	2	18535.16	0.4	1.1	0.4
DC5	4	2	112988.3	0.7	1.1	0.4
DC6	1	3	15000	0.5	1.1	0.4
DC7	3	2	6000	1.2	1.1	0.4
DC8	2	2	28984.38	0.4	1.1	0.4
DC9	0	2	3953.125	0.7	1.1	0.4
DC10	5	2	48000	0.7	1.1	0.4

Figure 2 shows the simulation results interface from CloudAnalyst. It displays the allocation of the users' workloads on the selected proper datacenter.

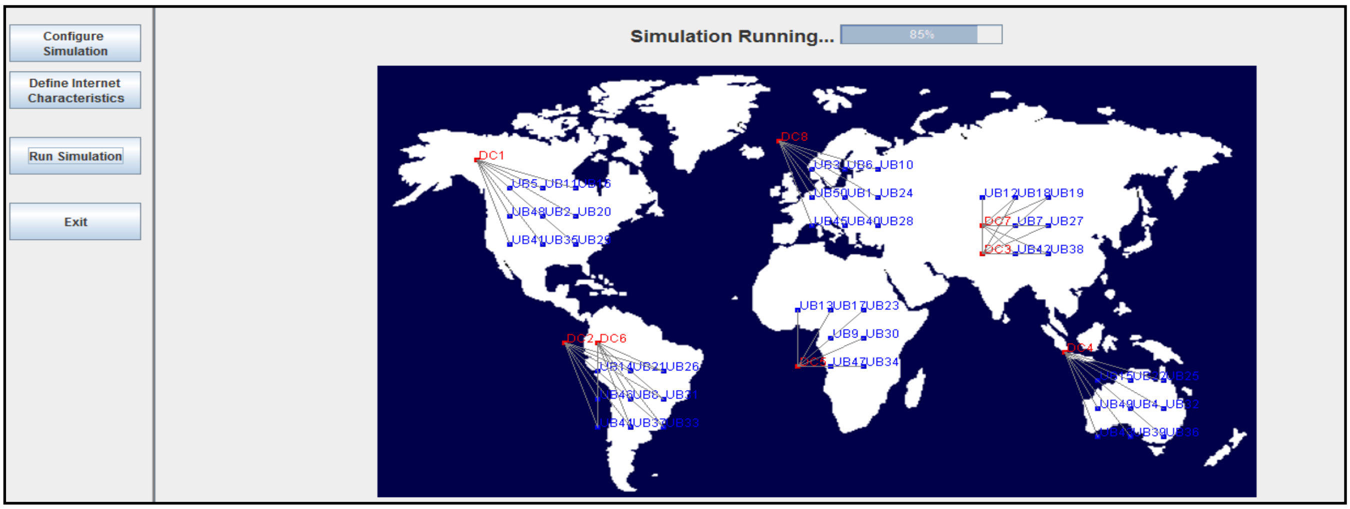

Fig 2: Simulation Result

After finishing the simulation, every user workload is allocated on one VM that also is allocated on one physical resource, but every physical resource can serve more than one workload (see Figure 2). Figures 3, 4 are the simulation result forms as appeared by CloudAnalyst.

Figure 3 represents the detailed workloads allocation (i.e. the average, minimum and maximum predicted responding time, and real paid cost). Figure 4, displays the graphs that clarify how much capacity of data centers is used.

Simulation Results

Overall Response Time Summary

	Average (Sec)	Minimum (Sec)	Maximum (Sec)
Overall Response Time:	4.43	0.05	268.20

Export Results

Response Time By Region

Userbase	Avg (sec)	Min (sec)	Max (sec)	DeadLine Time(...	Cost($)	Budget($)
UB10	6.342	2.866	11.619	19	37.109	57.3
UB11	1.919	0.296	4.083	5.8	12.576	5.9
UB12	1.255	0.277	3.548	3.8	10.753	5.5
UB13	1.447	0.345	3.399	4.3	0.005	6.9
UB14	1.253	0.973	1.559	3.8	9.401	19.5
UB15	1.911	0.598	3.93	5.7	12.143	12
UB16	2.645	0.66	5.227	7.9	1.54	13.2
UB17	5.281	1.508	15.838	15.8	41.584	30.2
UB18	1.311	0.47	3.873	3.9	13.103	9.4
UB19	1.287	0.372	2.802	3.9	8.576	7.4
UB1	1.302	0.361	4.696	3.9	7.084	7.2
UB20	0.299	0.087	0.814	0.9	1.087	1.7
UB21	1.186	0.338	3.889	3.6	10.653	6.8
UB22	1.095	0.277	3.741	3.3	4.145	5.5
UB23	18.848	7.228	38.739	56.5	55.716	144.6
UB24	5.26	1.575	9.122	15.8	41.701	31.5
UB25	11.132	1.217	28.128	33.4	105.757	24.3
UB26	31.745	24.892	41.62	95.2	266.88	497.8
UB27	2.068	0.617	5.692	6.2	20.618	12.3
UB28	5.321	4.699	6.257	16	6.451	94
UB29	0.915	0.2	2.536	2.7	4.862	4
UB2	2.484	2.007	3.085	7.5	24.251	40.1
UB30	2.338	0.552	7.989	7	8.973	11
UB31	0.612	0.131	1.242	1.8	3.85	2.6
UB32	2.678	0.813	6.548	8	3.322	16.3

Fig 3: Workloads Allocation Details

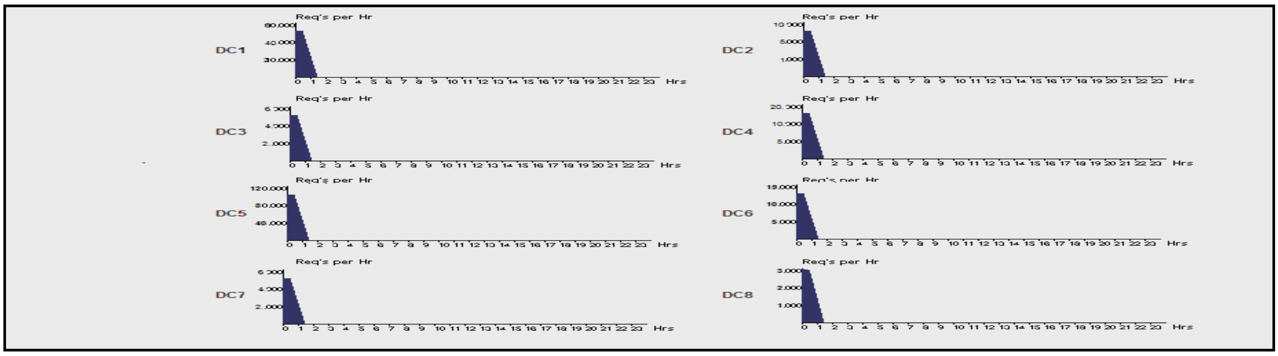

Fig 4: Utilization Capacity of Data Center

As shown in Figure 4, the physical resources which are used in the allocation process are only from eight data centers and they are used with good resource utilization, and still there are two unutilized data centers (DC9 and DC10) that can be used by another allocation process. Figure 5 is concerned with explaining the cost of the used data centers, and the total sum of paid cost based on the available budget.

Cost

Total Userbases Cost :	$2247.17
Total Userbases Budget :	$3155.55
Total Servers Cost :	$1470.04
Total Data Transfer Cost :	$363.00
Grand Total :	$1833.04

Data Center	Server Cost	Data Transfer Cost	Total Cost
DC6	120.003	15.046	135.049
DC5	210.006	51.661	261.667
DC4	120.003	62.798	182.801
DC3	180.005	8.548	188.553
DC2	240.006	14.767	254.774
DC1	120.003	126.489	246.493
DC8	180.005	75.49	255.494
DC7	300.008	8.203	308.211

Fig 5. Allocation Cost Details

Figures 6, 7 are the comparative graphs which display the relation between the requests size and the capacity that is used from the physical resources. Figure 6 shows that by using the RPOAWLB algorithm, the used data centers are utilized with good loading of every physical resource. The statistical evaluation shows that the total using of the reserved capacity is 93%, with releasing the unutilized resources to be able to receive other users' requests. Figure 7 displays the resources utilization by using the previous version of the RPOA algorithm. The graph indicates that the requests are allocated over all of the available resources without concerning the load balancing (i.e., the all ten data centers are used); with utilized capacity is 81% of the total resources capacity.

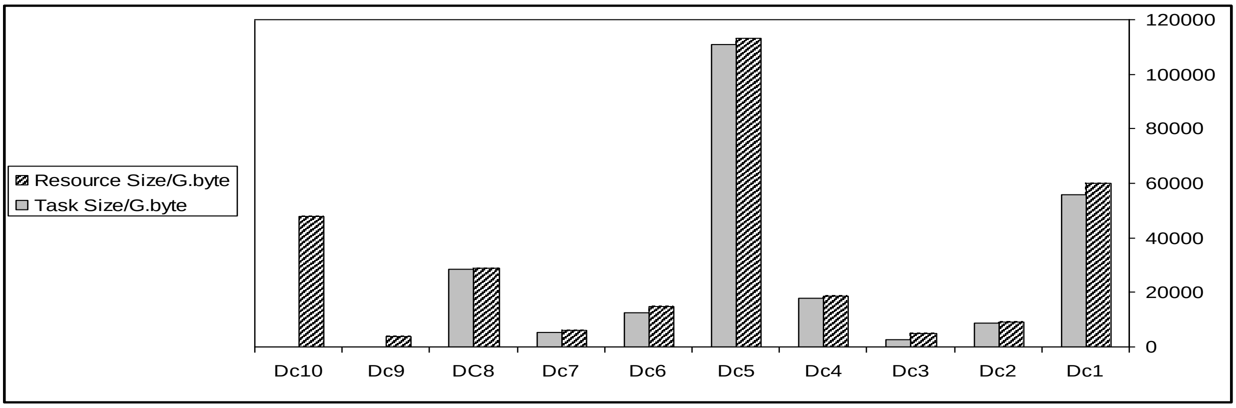

Fig 6: The Relation between Requests Size and Resource Size in RPOA with Load Balance Algorithm

Another comparative graph is displayed in Figures 8, and 9. The relation between the response time (i.e., the predicted average, and maximum responding time) and the predefined deadline time is stated in a statistical chart which is shown by Figure 8.

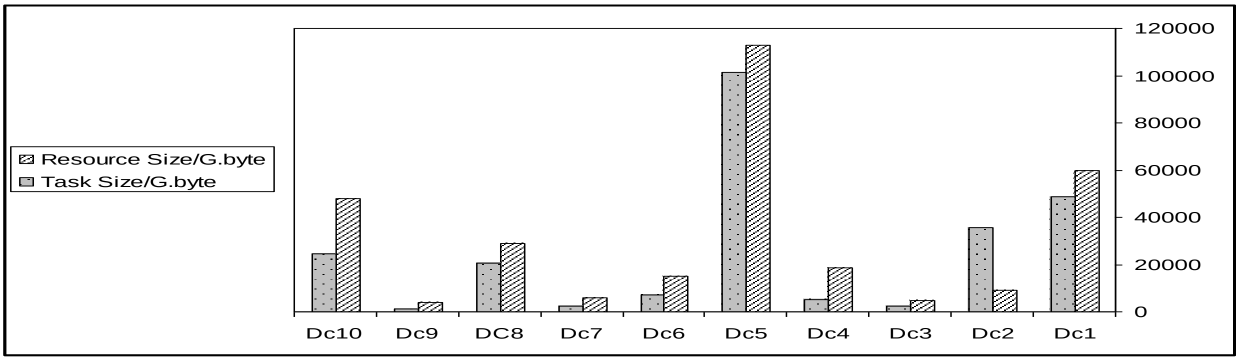

Fig 7: The Relation between Requests Size and Resource Size in the RPOA Algorithm

Also, Figure 9 represents the relation between the user budgets and the actual paid.

Figures 8, and 9 show the satisfaction rates for achieving the user's QoS parameters (i.e., response time, and actual cost) of the proposed sample of the fifty independent workloads. The RPOAWLB algorithm satisfies the user requirements with 88% for the service cost, and with 90% for service responding time.

The comparisons between RPOAWLB and existed RPOA algorithms are displayed in Figures 10, 11. The charts give the improvement rate in cost and time between the two algorithms.

Applying the RPOAWLB algorithm improves the real paid cost with 4% more than that RPOA algorithm (i.e. satisfaction rate with RPOA algorithm was 84% while with RPOAWB algorithm is 88%). Also, the RPOAWLB algorithm gives satisfaction rate 90% in the responding time while RPOA algorithm gives 88% (see Figure 11).

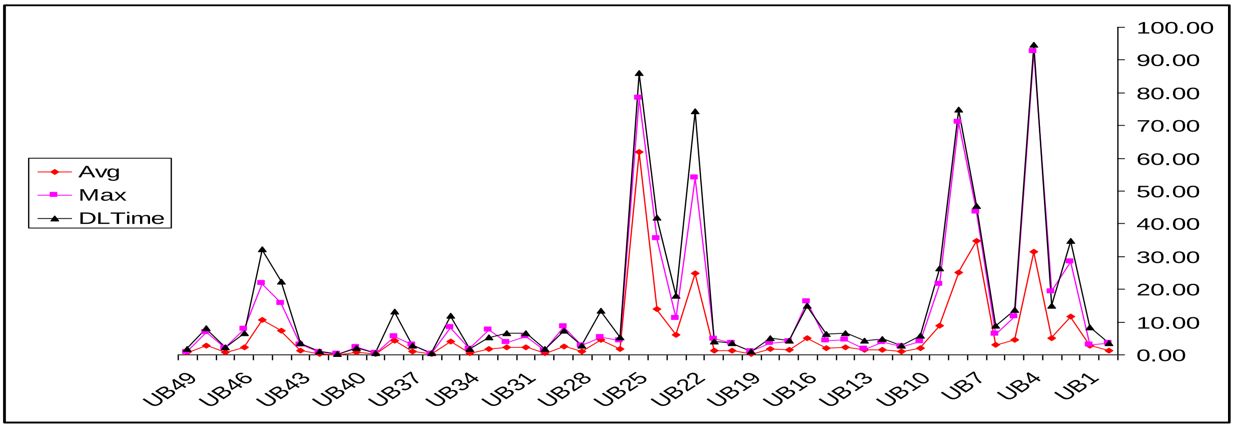

Fig 8: The Relation between Responding (Average and Maximum) Time and the Predefined Deadline Time

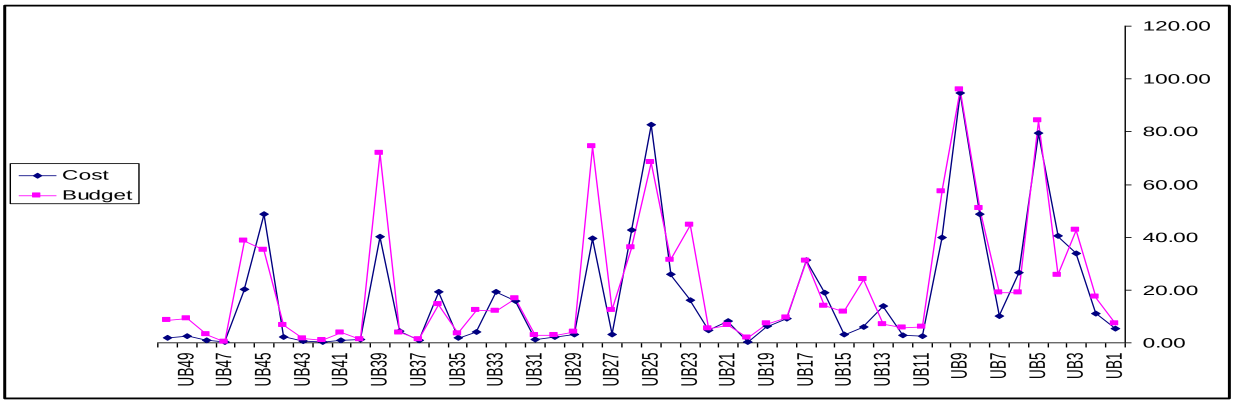

Fig 9.The Relation between Actual Paid Cost and the User Budget

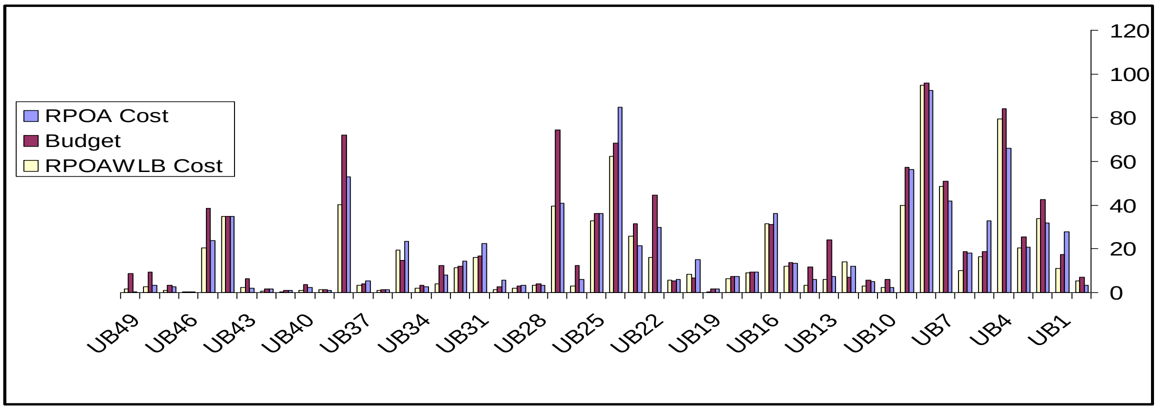

Fig 10: Comparison between paid cost in RPOA and RPOAWLB

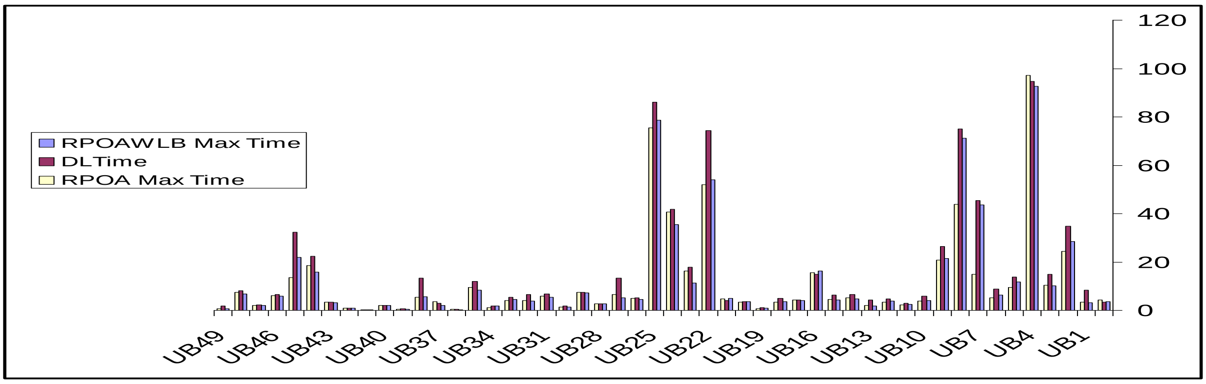

Fig 11: Comparison between responding time in RPOA and RPOAWLB algorithms

6. CONCLUSIONS AND FUTURE WORK

The proposed RPOAWLB algorithm tries to satisfy both the cloud's consumer QoS parameters, and the Cloud provider profits. Minimizing the response time, and the total paid cost of the delivered service are the main issues to achieve the user's QoS requirements. On the other hand, gaining a profit of providing the service is the main target of the provider. The RPOAWLB algorithm is based on satisfying good utilization and load balancing of resources capacity while distributing the workloads by maximizing the load on the utilized resources and freeing up the light loaded resources. According to the simulation results of the RPOAWLB algorithm using fifty workloads and ten available data centers, the utilized quantity is 93% from the reserved capacity, while there are two data centers that are still fully free and ready to receive new workloads.

The provisioning process is done by considering static workload which is considered the main drawback of the RPOAWLB algorithm which can be studied in the future work.

7. REFERENCES

[1] B. Urgaonkar, P. Shenoy, T. Roscoe, "Resource Overbooking and Application Profiling in Shared Hosting Platforms", SIGOPS Operating Systems Review Vol. 36, Issue SI, 2002, pp. 239-254.

[2] B. Wickremasinghe, "CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments", Distributed Computing Project, CSSE Dept., University Of Melbourne, 2009, pp. 433-659.

[3] E. Byun, Y. Kee, J. Kim, S. Maeng, “Cost Optimized Provisioning of Elastic Resources for Application Workflows”, Future Generation Computer System, Vol. 27, No. 8, 2011, pp. 1011-1026.

[4] H. Bhavani, H. Guruprasad, “Resource Provisioning Techniques in Cloud Computing Environment: A Survey”, International Journal of Research in Computer and Communication Technology, Vol. 3, Issue 3, pp. 395-401, 2014.

[5] J. Almeida, V. Almeida, D. Ardagna, I. Cunha, C. Francalanci, M. Trubian, “Joint Admission Control And Resource Allocation In Virtualized Servers”, J. Parallel Distrib. Comput., Vol. 70, Issue 4, 2010, pp. 344-362.

[6] M. Katyal, A. Mishra, “A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment”, International Journal of Distributed and Cloud Computing, Vol. 1, Issue 2, 2013, pp. 5-14.

[7] N. El.Attar, W. Awad, F. Omara, "Resource Provision for Services Workloads based on (RPOA)", Int. J. of Computer Science Issue (IJCSI), Vol. 9, Issue 3, 2012, pp. 553- 560.

[8] N. El.Attar, W. Awad, S. AbdElhafeez, F. Omara, “RPOA Model-Based Optimal Resource Provisioning”, International Journal of Advanced Computer Science and Applications, Vol. 5, No. 2, 2014, pp. 111-120.

[9] R. Buyya, R. Ranjan, R. Calheiros, “Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities”, in the 7th High Performance Computing and Simulation (HPCS 2009) Conf., June 21 - 24, 2009.

[10] R. Calheiros, R. Ranjan, A. Beloglazov, C. Rose, R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", J. of Software: Practice and Experience (SPE), Vol. 41, Issue 1, 2011, pp. 23-50.

[11] S. Banerjee, I. Mukherjee, K. Mahanti, “Cloud Computing Initiative using Modified Ant Colony Framework”, World Academy of Science, Engineering and Technology, Vol:3, Issue 23, 2009, pp. 200-203.

[12] S. Chaisiri, “Optimization of Resource Provisioning Cost in Cloud Computing”, IEEE Transactions On Services Computing, Vol. 5, No. 2, 2012, pp. 155-177.

[13] S. Suganya, P.RajiPriyadharshini, A.Masanam, “Resource Provisioning in Cloud Computing Based on Virtual Machines”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 3, 2014, pp. 5881- 5884.

[14] S. Woo, J. Mirkovic, “Optimal Application Allocation On Multiple Public Clouds”, Elsevier Computer Networks Journal, Vol. 68, 2014, pp. 138-148.

[15] Y. Chen, S. Tsai, “Optimal Provisioning of Resource in a Cloud Service”, (IJCSI) International Journal of Computer Science Issues, Vol. 7, Issue 6, 2010, pp. 95-99.