Energy Consumption Improvement and Cost Saving by Cloud Broker in Cloud Datacenters

Ahmad Karamolahy[1], Abdolah Chalechale[2], and Mahmoud Ahmadi[2]
[1]Radicloud Software Company Ilam, Iran
[2]Computer and Information Technology Department, Razi University, Iran

Abstract: *Using a single cloud datacenter in Cloud network can have several disadvantages for users, from excess energy consumption to increase dissatisfaction of users of service and price of provided services. The Cloud broker as an intermediary between users and datacenters can play a key role to enhance users' satisfaction and reducing energy consumption of datacenters that are located geographically in different areas. In this paper, we have attempted to provide an algorithm that assigns datacenter to users through rating various datacenters. This algorithm has been simulated by Cloudsim and will result in high levels of user satisfaction, cost-effectiveness and improving energy consumption. In this paper, we show that this algorithm can save 44% of energy consumption and 7% of cost saving to users are in sample simulation space.*

Keywords: *Cloud network, cloud broker, energy optimizing, cost saving.*

Received June 10, 2015; accepted December 9, 2015

1. Introduction

Despite the expanding world of Cloud computing, one of the most important part that consume a lot of energy hardware as an element in datacenter [1, 2]. On the other hand, however, some companies that use Cloud computing services do all their calculations on Cloud brokers outside of organization and use internal equipment (Hybrid Cloud Computing) that their energy consumption management will seem important for user [3]. Datacenters of Cloud computing become so large-scale and wide that have detrimental effects on environment [9] and effect on global warming. According to estimates [5] datacenters consume each year a large amount of electrical energy and import huge amounts of carbon dioxide into atmosphere and this amount is growing at a high rate annually, so that energy consumption of Cloud brokers of datacenters has increased 56% from 2005 to 2010. The energy consumption in 2010 will be 1 to 1.5% of Earth's total energy consumption[1].

The reason for our focus on processing resources of datacenters is it’s over consumption compared to other sectors of Cloud brokers so that, according to research [4, 17] 89% of total energy consumption of Facebook network datacenters is related to processing elements of datacenters. Energy consumption of datacenters related to Cloud brokers impose much costs to providers and government, so that consumed power cost of datacenters in United States in 2010 amounted to 23.6 billion dollars. Following the current status, a sum will be added to this amount annually. On the other hand, despite different Cloud brokers with different characteristics, users may not get all their needs from a single Cloud broker because a Cloud broker might not service based on budget and needs of users. Therefore, in this study we unveil a component on different Cloud brokers that will communicate with multiple Cloud brokers across geographic areas and guarantee user satisfaction based on user requirements and policies for improving energy consumption and improving service speed to users [6]. The main problems in here are how we can choose the suite resource in datacenter for each user’s request? And how we can improve satisfaction of users?In this paper, we will run an algorithm for rating datacenters on Cloud broker where energy consumption, speed of service to users and users’ cost will be improved. The proposed algorithm is simulated by Cloudsim. Cloudsimthat is a tool in Java language and is used in order to simulate various algorithms on Cloud networks. The result and comparison sectionis simulated using this tool.

This paper generally consists of four sections; in first section literature on optimization by Cloud brokers will be studied. In second section, we will examine the architecture of Cloud broker. Third section presents allocation algorithm of datacenter by Cloud broker based on ratings. Finally, in fourth section we will present the results and comparisons.

2. Literature Review

Many studies have been done in the field of reducing energy consumption in Cloud brokers, but the number of studies where the Cloud broker element is used as the main factor affecting energy consumption efficiency is very few. Beloglazov [2] deals with

[1] - However, this amount is currently estimated to be 2%

optimizing the energy consumption within Cloud brokers with methods that are planned on Grid. This research has also pointed to Cloud broker, but their role in management of energy consumption is very little. So further energy consumption optimization is done on providers' resources and energy consumption management has become an internal issue in Cloud brokers. Using methods such as maximum application of resources and avoiding workless internal processing elements within network are provide in this paper.

In another research, Zhang *et al*. [18] introduced an algorithm for load balancing on different Cloud brokers on Grids. In this paper, two types of load distribution have been done on different Cloud brokers

1. Load distribution on time components; where load distribution is done based on local times of different Cloud brokers.
2. Load distribution on load type; where Cloud brokers are classified and each of them provide a particular service. This study was conducted on Grids and several years have passed from writing time, however, its idea and thought has created a new way for Cloud networks. Here this idea will be used.

Several algorithms have been provided for promoting efficiency level of allocating Cloud resources to user requests. In an article conducted by Rodero *et al*. [11] allocating resources is done based on the nearest required hardware of users, so that each Cloud broker will have a special rating and user requests will be based on this rating. In rating algorithm, cost can be considered as a parameter.

In a study conducted by Koomey [7], the total electricity used by servers in the U.S. and the world has been estimated using combining measured data. Moreover, a comparison of synchronous and asynchronous data replication in cloud computing is investigated in [13]. Limbani and Oza [8] proposed a service broker policy for data centre selection in Cloud Environment. They also implement this policy and concluded that a cost effective routing of user requests has been achieved.

3. The Overall Architecture of Cloud Broker

There are three major components in Cloud networks that we consider here: Member users, Cloud broker anddatacenters. Firstly, user sends his request to Cloud broker that is executable code. Cloud broker will allocate a datacenter to user based on policies that are already defined. Figure 1 shows the overall architecture of network and how Cloud broker is placed there [10].

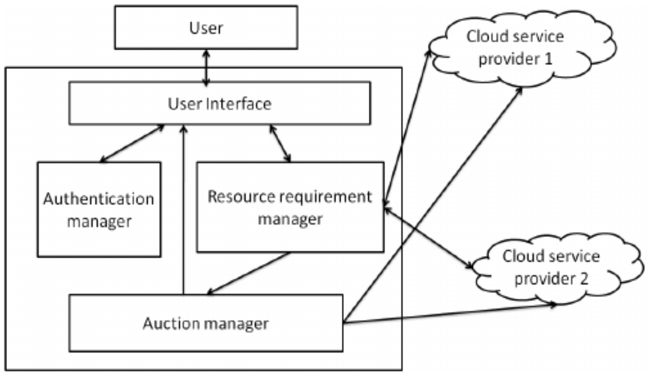

Figure 1. Overall architecture of cloud brokers in cloud.

As it can be seen in Figure 1; user asks his request in a sub-minor network that can be small company [11] cloud broker in cloud scheduler allocates resource (cloud broker appropriate cloudy server) to desired request. The main subject of our research is to study on resource allocation algorithms in order to reduce the costs of users. Requests are stored in a list as multiple virtual machines after scheduling and allocating resources. The virtual machines must constitute appropriate architecture and codes for each corresponding cloud servers. This list is compiled in resource allocation management and each request will be connected to its cloud server. Using this architecture, we can reduce costs related to lack of appropriate resource allocation to users, the users waiting time, the renting cost of different servers and finally ensure customer satisfaction due to time and cost optimization [12, 15]. Algorithm on cloud broker will be introduced in order to allocate appropriate datacenters to users.

4. The Proposed Algorithm

In this study, we introduce an algorithm which allocates resource for each request. This algorithm has two major phases:

1. Introduction phase that constant characteristics of any datacenters will be ranked by sophisticated algorithm. Ranks like energy, cost and performance will be calculated in this section.
2. Running phase that combines another characteristics of datacenter called low time. This rank points to peak work time of servers, if server in peak workload time we would use another servers. At the end of this phase we will calculate overall rank of datacenter and assign best server to proper users.

4.1. Introduction Phase

Features like hardware and cost characteristics of servers will not change through running phase. Therefore, we ranked this characteristics as a StaticRank for reduce time of processing. In following we describe how to calculate the ranks of energy, cost and performance that will construct the static rank. Figure 2 describes the flowchart of introduction phase.

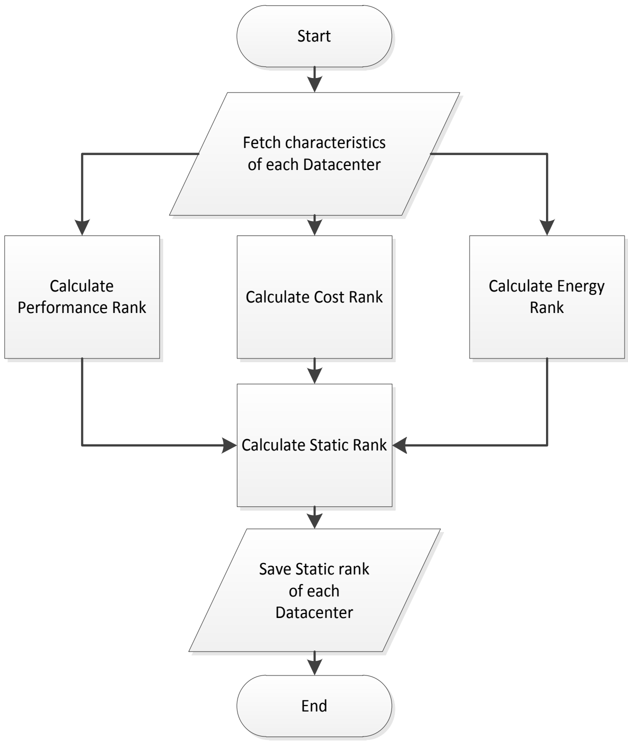

Figure 2. The flowchart of introduction phase.

4.1.1. Energy Rank

Energy consumption rating is based on processor, other hardware components and energy standards (each field will be a number between 1 and 10. Bigger=better). As depicted in Figure 3, the energy consumption is shown as 30% per CPU, 16% per RAM and other hardware components [1].

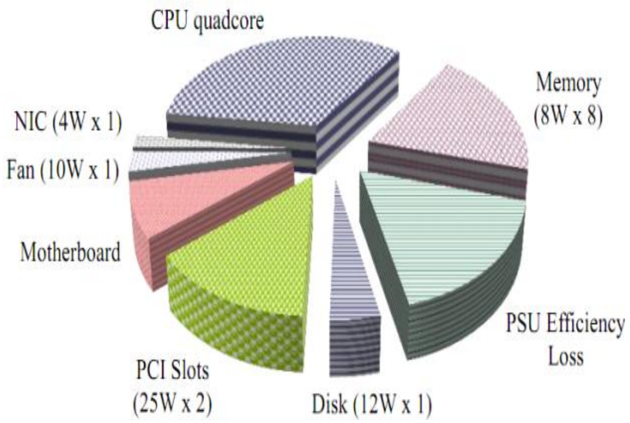

Figure 3. Contribution of each system components in energy consumption.

These values are obtained for Desktop devices and may be slightly different for server systems. It must be noted that acts such as virtualization of system can help to improve energy consumption. One of these algorithms is distribution of processor operating frequency that improves energy consumption using optimal and uniform application of processor. Processor has relatively uniform energy consumption in off peak or peak time and nature of this algorithm is originated from this point. Here, our aim is to provide a broader and physical algorithm that is done by Cloudy broker on different data centers not on one or more processors. Main memory has different situation that have less energy consumption based on architecture and technology. Here we consider 15% for it. Large amounts are spent for network equipment which are approximately the same on all architectures and generations and often are overlooked. For Power Suply Unit (PSU) that are responsible for Alternative Current (AC) to Direct Current (DC) conversion we cannot consider certain standard, because 60 and 80% of input energy is wasted in all units and this section will take up to 20 percent of total energy providers.

Equipment such as motherboards, fans and Peripheral Component Interconnect (PCI) slots can be argued that have same energy consumption on different architectures, so we ignore this component for energy rating of system. Disk power consumption is about 8 percent in system. If new architectures such as SSD and Flash are used, it can be very low [14]. Energy rating formula is as follows:

$$EnergyRank = (0.4)(CPU\ standards\ rank)+(0.15)(Ram\ rank)+(0.1)(PSU\ rank)+(0.15)(energy\ standard)+(0.2)(other) \quad (1)$$

Processor standards can be based on processor energy consumption, number of cores, processor performance per energy consumption and so forth. Here CPU standards will be achieved as a set of calculated properties, in a comparative table of all datacenters, then a rating from 1 to 10 will be given for each data.Main memory rating will be done on memory architectures according to researches and again ranking will be based on comparison between other components of system. PSU rating will be fairly constant rating that we can allocate a median number for each based on energy waste amount by various data centers. Ratings related to energy standards are weighted based on datacenter licenses and permits in a process. For example datacenters that have Green IT standard 5 of 10 will be awarded them or if they have an ISO standard, 3 scores will be awarded.

4.1.2. Cost Rank

Firstly, we insert cost of each of them on a table (as hour/dollar) with data center characteristic in order to get the cost rating of each datacenters. Then we arrange costs in descending form. The reason of sorting is finding mead element after that we calculate the difference of each cost to median element. In the following formula we discuss how it works. For each datacenter we need to calculate this cost Rank.

$$CostRank= (MID/Cost) + Constant \quad (2)$$

Constant in this formula using for normal the rank of cost because if we don't use this parameter the small difference of each cost may be decay that and assign low rate to it. In simulation section we discuss this ranking more precisely.

4.1.3. Performance Rank

In the present study, comparative algorithm of CPU 2006 is used to rate different processors. On the other hand, reliable Benchmark websites are used in order to rank RAM main memories and storage elements. We don't focus to performance rank in this research because we don't have any metrics to evaluate this feature (satisfying of users) however we present a surface view.

PerformanceRank=(0.4)(CPU rank)+(0.1)(ram rank)+(0.3)(bw rank)+(0.2)(other rating like distance, storage, num of vms, os, …) (3)

Hand shaking physical algorithm will be used for bwRank characteristic that indicates the bandwidth rank and network performance, so that a computational package will be sent to each data centers. After processing, delay of each resource will be saved and ranked. Ranking is done based on rating of other datacenters that is a number between 1 and 10.Other elements such as distance between data center and user, number of virtual machines, operating system, etc. will be given based on experimental algorithms of ranking and limited factor of algorithms (this section may be the subject of future studies).

4.1.4. Static Rank

Now we have rank of energy, cost and performance of each datacenter static rank can be calculated by sum of these ranks. But the important question is that what kind of ranks are more important for us? Well this depending to users or organization policy. If the organization concern about energy consumption or environmental they can choose the servers that have better energy rank. However in this research we use these weightings:

StaticRank=(0.4)EnergyRank+(0.2)CostRank+(0.2)PerformanceRank (4)

This weights can be varied depend on desires of designers. It should be noted that these weights has been tested for different values and for the proposed values the system performance has the most improvement. Subsequently, we discuss about running phase.

4.2. Running Phase

Users and datacenters are spread in vary geographic area. When users send their requests, broker that handle our algorithm assign proper datacenter to user. The geographic location and peak hour can be very important [16]. Another rank that we must calculate in running phase is low Time Rank. This rank is dynamic because changed at any received request. This rank calculated by Table 1.

Table 1. Fetch low time rank.

lowTimeRank	Local time of datacenter
10	22:00 to 6:00
9	6:00 to 8:00
4	8:00 to 10:00
0	10:00 to 14:00
3	14:00 to 16:00
4	16:00 to 18:00
8	18:00 to 20:00
9	20:00 to 22:00

This ranking not very accurate but we can do better by monitoring some real datacenters.

4.3. Overall Rank

So we have static and dynamic rank. We can calculate OverallRank.

OverallRank=StaticRank+(0.2)lowTimeRank (5)

Then best datacenters will assigning to users request.
In following section we describe how simulation works.

5. Simulation

Here we use Cloudsim tool that is a set of JavaScript codes in order to implement algorithm. The system used for simulation has the following characteristics: Processor: Intel CPU core i7 Q720 1.6Ghz, RAM: 4GB, operating system: Windows 7 and compiler: Netbeans 7 Software Development Kit (SDK).

5.1. Simulation Paradigm

Ratings of each data center has been calculated by a separate program and based on algorithm. We introduced 10 thousand requests to system as statistical population. These requests are run as separate cloudlets in simulator. Finally, we compared energy consumption of each request by three algorithms named local assign, random local assign and flat assign.The sample chooses of datacenters came from Jason Read's research that benchmarks real servers all around the world (see: https://www.scalescale.com/building-cloud-benchmarking-system). We extract 11 sample datacenters from this research(for approach our research to real world) and implement those characteristics in cloudsim. Each of these datacenters hasa virtual machine that equal to its characteristics. In the following table these servers are defined. All of extracted energy ranks were checked by energy star and greenIT standards [16, 17]. These datacenters will spread over our hypothetical geographic place. Figure 4 describe geographic locality of each datacenter.

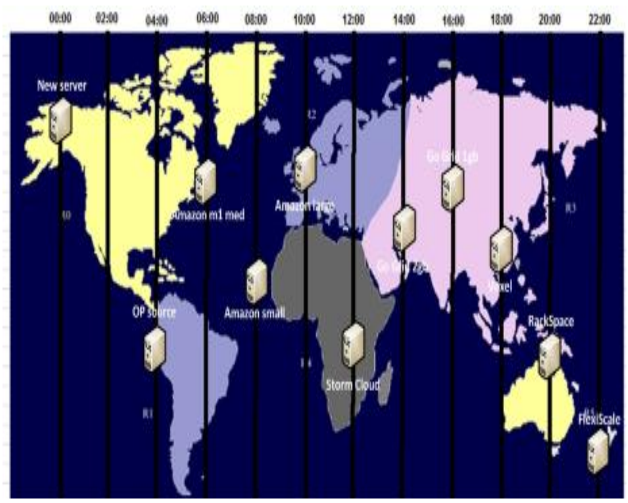

Figure 4. Geographic location of sample datacenters and their local time in our simulation.

In our simulation users are located in locality of each datacenter. Number of users of each datacenter describe in Table 2. In cloudsim we are coinciding users to their requests. The maximum number of requests in peak time can infer as the maximum number of users.

Table 2. Characteristics of sample datacenters.

DC name	CPU type	CPU power	#of cores	RAM	HDD	Cost $/hour	Geographic time	Maximum number of Cloudlet
Storm Cloud 16	Opteron 2350	2Ghz	4	15.2GB	3GB	0.34	12 GMT	35.000
Go grid 2gb	Xeon E5520	2.26Ghz	4	2GB	4GB	0.38	14 GMT	25.000
Go grid 1gb	Xeon E5520	2.26Ghz	4	1GB	1GB	0.19	16 GMT	20.000
Voxel 2gb	Xeon L5520	2.26Ghz	4	2GB	3GB	0.106	18 GMT	65.000
Amazon m1 small	Opteron 2218	2.6Ghz	2	1.7GB	0.5GB	0.095	8 GMT	22.000
Amazon c1 medium	Xeon E5410	2.33Ghz	4	1.7GB	3GB	0.17	6 GMT	40.000
Amazon m1 large	Xeon E5430	2.66Ghz	4	7.5GB	8GB	0.34	10 GMT	100.000
Rack space rs 4	Opteron 2374	2.2Ghz	4	2GB	2GB	0.24	20 GMT	55.000
Op source os 4-2	Xeon x7460	2.66Ghz	8	4GB	2GB	0.392	4 GMT	67.000
New server med	Xeon 3.2	3.2 Ghz	8	2GB	1GB	0.17	0 GMT	50.000
Flexi scale 2gb	Opteron 8218	2.6Ghz	2	2GB	1GB	0.13	22 GMT	22.000

5.2. Compared Algorithms

For showing the optimization of our algorithm we offer three others algorithms. In following we descript how these algorithms works.

5.2.1. Local Selection

Every user just use local server. In this algorithm we don't use broker and users directly use their local datacenter.

5.2.2. Random Local Selection

Like previous algorithm but in this algorithm we use broker and select randomly from utmost 3 near datacenter. Near datacenters indicate those datacenters that utmost 2 hours far from user's local datacenter.

5.2.3. Flat Distribution

At any time of simulation we dedicate same number of Cloudlet to datacenter. This algorithm can consider as fair algorithm.

5.3. Simulation Results

We implement each sample datacenter and their characteristics to cloudsim. Our simulation will run every 2 hour and result will be saved.

5.3.1. Energy Optimization

In Figure 5 we can see the energy consumption of any algorithm compared with our proposed algorithm.

The last algorithm that we use in our simulation is another proposed algorithm that changed weight of energy rank and cost rank. This can show the flexibility of our algorithm.

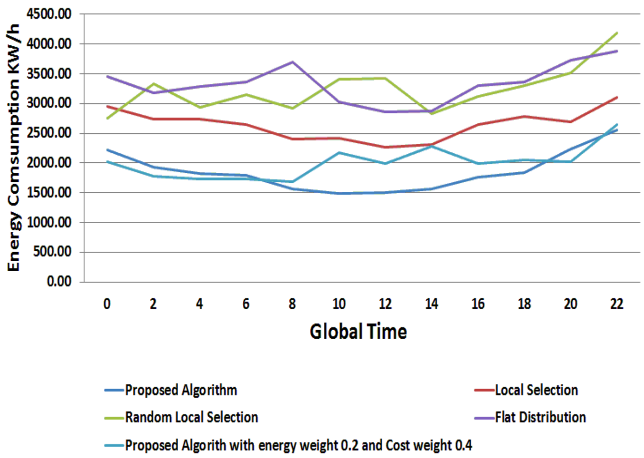

Figure 5. Result of nergy consumption.

5.3.2. Cost Optimization

In Figure 6 we can see users cost fees over our simulation.

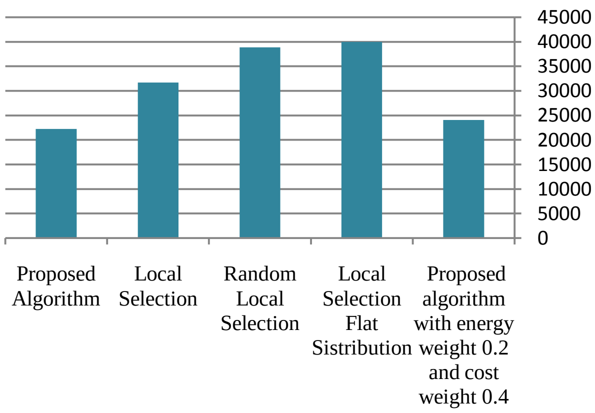

Figure 6. Compare energy consumption, the horizontal axis shows the policy and vertical axis shows consumed energy in term of kw/h.

6. Analyse of Results

As it can be observed in previous section, proposed algorithm can improve energy consumption and the cost. In Figure 7 and Tables 3 and 4 we present consumed energy and saved cost. The positive point of proposed algorithm is flexibility. That means designer can choose his/her policy and can reduce cost fee by choosing bigger weight over cost rank or same to energy rank.

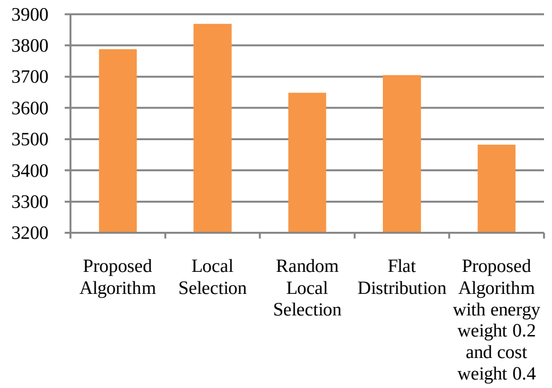

Figure 7. Compare Cost fees, the horizontal axis shows the policy and vertical axis shows the cost in term of dollars.

Table 3. Cear data for compare energy optimization.

Type of algorithm	**Energy improvement compared to local selection algorithm**	**Energy improvement compared to Random local selection algorithm**	**Energy improvement compared to Flat distribution**
Proposed algorithm	29%	42%	44%
Proposed algorithm with 0.2 weight energy rank and 0.4 cost rank	23%	38%	39%

Table 4. Clear data for compare cost fees.

Type of algorithm	**Cost saving compare to local selection algorithm**	**Cost saving compare to Random local selection algorithm**	**Cost saving compare to Flat distribution**
Proposed algorithm	+3%	-2%	-1%
Proposed algorithm with 0.2 weight energy rank and 0.4 cost rank	+10%	+5%	+7%

7. Conclusions

In this research we introduced an algorithm that worked with ranking datacenter. Algorithm mount over global broker Cloud and simulation implemented by Cloudsim. As we can see on simulation result, energy consumption can improve44% compare to other algorithms. And cost fees can reduce 7%.

There are several advantage of ranking datacenter algorithm:

1. Flexibility: broker can implement policies that may cost oriented policy or energy consumption policy.
2. Users satisfying: if you grant users cheaper service they will be happy this can be better if they know service has good performance and not charged environment.
3. Competition for better services: clearly, if broker can connect users to different kind of datacenter then datacenters will be going to race condition with each other and they must grant better services to user for survival.

Although we used IaaS datacenters in our algorithm that provide bare hardware for users but the architecture and type of machine will become an important issue.

References

[1] Beloglazov A. and Buyya R., "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers," *Concurrency and Computation Practice and Experience*, vol. 24, no. 13, pp. 1397-1420, 2012.

[2] Beloglazov A., Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing, Thesis, the University of Melbourne, 2013.

[3] http://adrianotto.com/2013/01/ssd-power-savings-pays-for-itself/, Last Visited, 2015.

[4] http://energystar.org/about.html, Last Visited, 2014.

[5] http://greenit.netwhygreenit.html, Last Visited, 2014.

[6] Kapgate D., "Efficient Service Broker Algorithm for Data Center Selection in Cloud Computing," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 1, pp. 355-365, 2014.

[7] Koomey J., "Estimation Total Power Consumption by Services in the U.S. and the World," Technical Report, 2007.

[8] Limbani D. and Oza B., "A Proposed Service Broker Policy for Data Center Selection in Cloud Environment with Implementation," *International Journal Computer Technology and Applications*, vol. 3, no. 3, pp. 1082-1087, 2012.

[9] Marinescu D., *Cloud Computing Theory and Practice*, Morgan Kaufmann, 2014.

[10] Raj G. and Setia S., "Effective Cost Mechanism for Cloudlet Retransmission and Prioritized VM Scheduling Mechanism over Broker Virtual Machine Communication Framework," *International Journal on Cloud Computing: Services and Architecture*, vol. 2, no. 3, 2012.

[11] Rodero I., Guim F., Corbalan J., Fong L., and Sadjadi S., "Grid Broker Selection Strategies Using Aggregated Resource Information," *Future Generation Computer Systems*, vol. 26, no. 1, pp. 72-86, 2010.

[12] Schlegel T., Kowalczyk R., and Vo Q., "Decentralized Co-Allocation of Interrelated Resources in Dynamic Environments," *in Proceedings IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Sydney, pp. 104-108, 2008.

[13] Subramani K., Shanmugasundaram V., and Chandran K., "Design and Implementation of a Synchronous and Asynchronous-Based Data Replication Technique in Cloud Computing," *The International Arab Journal of Information Technology*, vol. 13, no. 1, pp. 100-107, 2014.

[14] Teng F. and Magoul`es T., "A New Game Theoretical Resource Allocation Algorithm for Cloud Computing," *in Proceedings of International Conference on Green, Pervasive, and Cloud Computing*, Hualien, pp. 321-330, 2010.

[15] Tordsson J., Moreno R., Moreno-Vozmediano R., and Llorente I., "Cloud Brokering Mechanisms for Optimized Placement of Virtual Machines Across Multiple Providers," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 358-367, 2012.

[16] Tsai J., Fang J., and Chou J., "Optimized Task Scheduling and Resource Allocation on Cloud Computing Environment using Improved Differential Evolution Algorithm," *Computers and Operations Research*, vol. 40, no. 12, pp. 3045-3055, 2013.

[17] Yang C., Lin C., and Chen S., "A Workflow-Based Computational Resource Broker with Information Monitoring in Grids," *in Proceedings of 5th International Conference on Grid and Cooperative Computing*, Hunan, pp. 199-206, 2006.

[18] Zhang Z., Wang T., Xiao L., and Ruan L., "A Statistical Based Resource Allocation Scheme in Cloud," *in Proceedings of International Conference on Cloud and Service Computing*, Hong Kong, pp. 266-273, 2011.

Ahmad Karamolahy received the B.S. degree in Computer Engineering from Sham sipur College of Tehran, Iran. He obtained M.Sc. degree in Software Computer Engineering from Azad University of Kermanshah, Iran. His research interests are fields of Cloud Computing, Search Engine Crawler Algorithms and Analyzing connections in Social Networks.

Abdolah Chalechale born in Kermanshah, Iran, received his B.S. Electrical Engineering (Hardware) and Computer Engineering (Software) from Sharif University of Technology, Tehran, Iran. He received his Ph.D. degree from Wollongong University, NSW, Australia in 2005 and currently is with Razi University, Kermanshah, Iran. His research includes image processing, machine vision and human-machine interaction.

Mahmood Ahmadi received the B.S. degree in Computer engineering from Isfahan University, Isfahan, Iran. He received the M.Sc. degrees in Computer architecture and engineering from Tehran Poly technique University. He got his PhD in May 2010. His research interests include Computer architecture, network processing, signal processing and reconfigurable computing. He is currently working as an assistant professor in the Department of Computer Engineering at the Razi University, Kermanshah, Iran.