IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 9, April 2014)

Green Cloud Computing: Greedy Algorithms for Virtual Machines Migration and Consolidation to Optimize Energy Consumption in a Data Center

Rasoul Beik

Islamic Azad University – Khomeinishahr Branch, Isfahan, Iran

beik@iaukhsh.ac.ir

Abstract— **Cloud Computing provides services on the Internet or intranet. There are three types of services in cloud computing: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (Saas). In cloud computing, customers do not need to have their own dedicated resources; cloud providers provide services they need on the Internet or intranet. Therefore customers only pay for services and they do not spend for hardware, software, and communication resources. In cloud computing, customers and providers negotiate and agree on an acceptable level of quality of service; and after that, utilizing resources, scaling-up or scaling-down resources, and generally managing resources is one of the most important tasks of cloud computing providers to satisfy customers. When there are more requests for services, adding extra resources is the simplest way to keep the quality of service. But this solution is in contrast to energy usage optimization in data centers. Energy is consumed in a data center by servers, storage devices, communication and network devices, and overhead equipments (like cooling and lighting). Studies show that servers consume about 30 percent of total energy consumed in a data center; so reducing number of servers in data centers not only affects on energy consumption directly, but also affects on overall energy consumption because of reducing overhead equipments utilized in data center. There are a lot of improvements in hardware technologies caused less energy consumption by hardware, but it is not possible to ignore the role of software in improving resource utilization and energy consumption. There are different software architectures used in cloud computing. Some of them provide a layer which is responsible for managing resources to optimize energy consumption. Different algorithms are designed and implemented in this layer and each one has some advantages and some disadvantages. This paper, first summarizes various ways of resource usage optimization, then focuses on algorithms used in energy-aware layer of architecture, and finally introduces some greedy algorithms based on different criteria to study allocating resources to virtual machines and applications.**

Keywords — **data center, energy consumption, greedy algorithm, green cloud computing, resource management**

I. Introduction

In the traditional approach to software, an enterprise has to own everything – network infrastructure, hardware resources, and software resources – to deploy software and run it efficiently, but in this approach there are some drawbacks especially wasting energy. Emerging cloud computing has changed this approach. Cloud computing means service delivery over the Internet. Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) are three types of services delivered by cloud computing; providers provide these services elastically and based on user demands [3]. In cloud service, clients are not the owner of hardware and software infrastructure; they access software, storage, and computing resources but they do not pay for infrastructure costs [4]. There is a shared pool of resources that can be accessed based on on-demand user request. Rapid provisioning and releasing of resources with the least management effort or service provider interaction are some appealing features of cloud computing. It provides better access to software, computing facilities, and data through the network rather than using premises software and hosting [4]. Resource pooling in cloud computing points that users do not know about the location of resources requested by them; there is a pool of shared resources and each one can be assigned to user's request by cloud [4]. Based on the users' perspective, elasticity is one of the most attractive features of cloud computing [4]; cloud resources can be rapidly scaled-out and scaled-in, so users would not worry about the availability of resources. It helps enterprises to scale their IT services as their businesses grow in a cost effective manner. Enterprises do not need to worry about the cost of hardware, software, or other IT resources when their businesses are growing fast; they just pay for the services on the cloud by their demands. On the other hand, because providers can provide services for more enterprises simultaneously, they can manage resources more efficiently and therefore overall energy

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 9, April 2014)

consumption will be optimized. They can share hardware resources, like servers, between more applications, they can design software components that can be shared between applications, and they can scale-up and scale-down resources based on user demands [3]. So, some advantages of cloud computing are: lower hardware and software costs, fast and continuous software upgrade, large amount of storage, more reliable data, global access to documents, hardware independent, convenience, and on-demand access to shared resources. But on the other hand there are some disadvantages too: permanent Internet access, inefficient with low speed internet, and lower security.

Cloud computing is growing; many computing services moving to the cloud, and cloud computing has to scale-up dynamically to support new demands without losing quality of service. Scale-up means more power is consumed by newly deployed resources and consequently there will be more environmental impacts and CO_2 emissions.

Therefore cloud providers need to use resources on the cloud more efficiently to reduce energy consumption. They deploy resource usage optimization policies; they use hardware devices consuming less energy; they use virtualization to improve resource utilization; and they design self-optimized software applications to consume energy efficiently.

Resource usage optimization is one of the ways being used to reduce energy consumption. Resource usage optimization not only reduces energy consumption but also helps managers to control data center costs; especially they can control server usage because servers are the most important resource in all data centers [1]. It should be considered that business customers need to access efficient applications, and most times the best way to increase efficiency is adding more resources for faster processing, but it is in contrast to energy consumption optimization. The best solution to use resources more efficiently without losing application performance is designing architecturally efficient applications. Software architects must consider some issues like composition, factorization, functionality, dependencies, and scalability in designing new software to use resources more efficiently. Applications must be designed as a set of components that are able to work independently; they must be platform independent and can be configured dynamically [1]. There should be a control layer in software architecture which is responsible for managing resources, allocating resources to components, and sharing resources between components to improve efficiency. Reference [2] proposes a dynamic power management layer in software architecture which is responsible for scaling-up or scaling-down resources based on resource demands and resource availability.

Hardware solutions try to design and manufacture devices consuming less energy without losing their performance. Reference [5] mentions some mechanisms to power down processors and reducing energy consumption in memory. Running servers at a level lower than their utilization in data centers is one of the main causes of consuming energy inefficiently. Energy efficient hardware and power minimization in servers and networks are suggested solutions to cure that problem [6], [17].

In software solutions, virtualization is one of the best technologies for consuming energy efficiently in data centers. Virtualization and delivery of computing as services are the most important features of cloud computing [6]. Virtualization is an unreal computing environment created through the hypervisor. The hypervisor is a software which can manage several varied computing platforms running concurrently. Each client has its own unique copy of the selected platform and hypervisor passes control to a specific client instance based on demands [4]. Virtualization is an attractive tool for applications to use fewer resources and to utilize servers more effectively [1], [5]. Cloud virtualization can improve energy consumption by reducing infrastructure resources; it can be more efficient if new features, like resource sharing, are considered in software architecture.

Some important benefits of virtualization are: improving hardware utilization and reducing hardware costs, efficient and easy servers and services management and reducing management costs, efficient use of data center space, and reducing energy consumption in data centers.

In addition to virtualization, designers and architects try to deploy applications which use hardware resources more efficiently. Developing energy-aware scheduling and smart traffic routing algorithm are some proposed solutions [5], [6]. There are different types of devices that are used in a network infrastructure of a cloud, and how much energy each type of device consumes depends on user behavior and the type of applications which are run on a cloud and communicate with each other. Keeping quality of service at an acceptable level while computing services are growing on the cloud is an issue that leads to deploy new software architectures which can optimize themselves dynamically [7].

There are different energy-aware algorithms to optimize energy consumption in data centers while keeping the quality of service at an acceptable level. Reference [8] introduced a framework to simulate and evaluate different algorithms in ClousSim[9] and based on it evaluated some energy-aware algorithms. Mentioned framework is expandable and can be used to evaluate other algorithms too. Greedy algorithms are famous algorithms in software and are used in different softwares. In this research, greedy

IJDACR International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 9, April 2014)

algorithms were added to energy-aware framework of CloudSim and used to evaluate energy consumption and quality of service based on different metrics of data centers such as server power usage and server current utilization level.

II. Related Works

This section reviews some of the studies in the areas of energy consumption, quality of service and service level agreement in data centers and cloud computing environments.

This article [8] analyzes the competitive dynamic migration of virtual machines and studies related issues for dynamic consolidation of virtual machines. It is necessary to produce random or adaptive algorithms to improve the efficiency of deterministic algorithms. Based on the analysis, some novel adaptive heuristics proposed which are based on data history analysis using dynamic resource consolidation for virtual machines while maintaining the performance and energy efficiency. Results of simulation indicate that the combination of local regression algorithm and the least time migration is better than the other algorithms; this combination not only reduces energy consumption but also optimizes SLA violation effectively and reduces the number of VMs migrations.

This paper [10] defines an architectural framework and the principles of energy-aware cloud computing. Based on this architecture, resource procurement and allocation algorithms are offered for energy-efficient management of cloud computing environments. The proposed energy-aware algorithms allocate resources to applications in a way that either improves energy consumption or guarantees level of quality of service. The main idea is based on dynamic consolidation of virtual machines. CloudSim is used to evaluate the proposed algorithms and the results indicate that this approach reduces energy consumption in data centers in comparison to static resource allocation approach.

Today the energy efficiency of the Internet equipment has become an important research topic. In [11] several experiments have been done to measure energy consumption of NetFPGA card per packet and per byte. Based on this study, it is possible to estimate the energy consumption in a network based on the network traffic and different applications; and also it provides a way to compare and improve different architectures and protocols.

This paper [12] proposes a green policy reducing carbon emission in data centers. In mentioned policy, workloads of applications are scheduled on data centers which are more energy-efficient and have less carbon emission. Results of simulation show that the proposed architecture saves energy up to 23% and reduces carbon emissions by about 25%.

In [13], an adaptive power management policy is proposed that uses virtual machine preemption to regulate energy consumption based on users' efficiency requirements. The proposed energy management policy uses a fuzzy reasoning engine to determine that a request should be allocated to a resource by replacing, preempting, or consolidating other resources. Results show that energy consumption reduces about 18% without any considerable problem for requests with lower priority.

Reference [14] deals with the issue of resource allocation in a data center and proposes a new technique that maximizes the efficiency of the data center when executes heterogeneous applications with different SLA requirements. This approach allocates virtual machines in a dynamic way that will satisfy any customer's SLA requirement. The evaluation results show that this mechanism worked with 60% of the total number of servers that use strategies such as consolidation or migration, while SLA violations in this technique are negligible.

In [15] and [16], greedy algorithms are used to improve task scheduling in cloud-based applications. Improvement of task completion time and improvement of task execution cost are factors considered and simulation shows that greedy algorithms improve those factors in respect to sequential scheduling algorithms. But none of the cases considers resource usage optimization and especially energy consumption optimization in scheduling algorithms recommended.

III. Green Cloud Architecture and Virtualization

The purpose of the clouds is designing data center networks containing virtual services (such as hardware, database, and user interface, application) for users who can use applications with competitive prices along with the required quality of service. A typical cloud architecture contains an infrastructure layer (virtual machines and physical machines) and an upper layer which consist of consumers (users and brokers) who send their requests for cloud services. In the green cloud architecture, a layer is added between the top layer and the infrastructure layer which acts as the interface between infrastructure and consumers. This layer contains components that interact with each other to handle resource allocation in an energy-efficient manner.

Virtualization in data centers is growing. Support of moving virtual machines between physical nodes allows dynamic movement of VMs based on performance requirements. When VMs, allocated to physical nodes, do not use total capacity of provided resources, they can be consolidated in minimum number of physical nodes; so in this situation, idle nodes can be switched to sleep mode to reduce total energy consumed by data center.

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 9, April 2014)

Three important issues should be considered to achieve server performance and energy efficiency simultaneously. First, high switching of server reduces its reliability. Second, from the point of view of quality of service, turning off the resources in a dynamic environment has many risks. Dynamic nature of workload and high consolidation result in some VMs cannot access required resources in pick workload and therefore they fail to keep the quality of service. Third, guarantee of the service level of agreement in order to manage application performance is difficult in virtualized environments. All these require efficient consolidation policies that minimize the power consumption while keep the quality of service specified by the user.

IV. Greedy Algorithms for Improving CPU Efficiency

In this research three greedy algorithms in allocating virtual machines to the physical nodes are evaluated and energy consumption of each algorithm has been calculated in different performance thresholds. In the following, we will describe these algorithms.

A. Minimum Host Utilization Selection, Maximum Host Utilization Allocation (MinMax_HU)

As mentioned before, also more CPU utilization means more energy consumption, but this relation is not linear; a CPU in idle mode consumes 70% energy when it works on 100% utilization. Therefore it is possible to improve energy consumption by switching low utilization hosts to sleep mode.

In this algorithm hosts are periodically sorted based on their utilization; VMs in the less efficient hosts are moved to the more efficient hosts, provided that utilization of efficient host does not exceed utilization threshold defined in the algorithm. Figure 1 shows MaxMin_HU algorithm.

```
input: hostList, utilThreshold
hostList.sortDecreasingUtilization()
lastIndex = hostList.indexOfLastNonZeroHostUtilization()
for i := lastIndex downto 1
  for j := 1 to i - 1
    if hostList[j].currentUtil() + hostList[i].currentUtil() <= utilThreshold
          vmList = hostList[i].getAllVms()
          host[i].deleteAllVms()
          host[j].addVms(vmList)
          break
```

Figure 1. MaxMin_HU algorithm

B. Maximum Host Power Usage Selection, Minimum Host Power Usage Allocation (MaxMin_HPU)

In a heterogeneous data center, servers may be added to the data center in different times and they may have different power consumption. One way to improve the energy consumption is moving virtual machines from servers with higher power usage to the servers with lower power consumption. In this situation, the utilization of server with lower power usage must not exceed utilization threshold defined in the algorithm. Figure 2 shows this algorithm.

```
input: hostList, utilThreshold
hostList.sortIncreasingPowerUsage()
for i := hostList.size() downto 1
  if host[i] is off break
  for j := 1 to i - 1
    if hostList[j].currentUtil() + hostList[i].currentUtil() <= utilThreshold
          vmList = hostList[i].getAllVms()
          host[i].deleteAllVms()
          host[j].addVms(vmList)
          break
```

Figure 2. MaxMin_HPU algorithm

C. Minimum Host MIPS Selection, Maximum Host MIPS Allocation (MinMax_HM)

In this algorithm, computing power (MIPS) is the major criteria. Servers are sorted based on their computing power in descending order, then VMs from servers with lower MIPS are moved to the servers with higher MIPS. It should be noted that the utilization of server with higher MIPS must be less than the utilization threshold defined in the algorithm. Figure 3 shows MinMAX_HM.

```
input: hostList, utilThreshold
hostList.sortDecreasingTotalMips()
for i := hostList.size() downto 1
  for j := 1 to i - 1
    if hostList[j].currentUtil() + hostList[i].currentUtil() <= utilThreshold
          vmList = hostList[i].getAllVms()
          host[i].deleteAllVms()
          host[j].addVms(vmList)
          break
```

Figure 3. MinMax_HM algorithm

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 9, April 2014)

V. Designing Energy-Aware Layer Based on Greedy Algorithms

In [8], the energy-aware allocation layer has been added to CloudSim. This layer has been designed and implemented in two parts: the VM selection algorithms and the VM allocation algorithms to migrate virtual machines to the appropriate hosts. So based on this design, different selection and allocation policies can be used and compared together. However, this method does not use to implement the greedy algorithms and the selection and allocation algorithms should be based on a single criteria, but to maintain the integrity with the rest of the CloudSim, this study follows the same manner too.

A. *Selection Algorithms*

Figure 4 shows the general class diagram of the design of the VM selection algorithms from different servers. In this structure, each algorithm has been implemented in a class. Classes specified in gray are sections of CloudSim and blue classes have been included in this study.

- **PowerVmSelectionPolicyGreedyAbstract**: This class contains common methods of all greedy algorithms.
- **PowerVmSelectionPolicyMinHostUtilization**: This class selects VMs from low utilization hosts to migrate to hosts with high utilization.
- **PowerVmSelectionPolicyMaxHostPowerUsage**: This class selects VMs from hosts with high power usage to migrate to hosts with low power usage.
- **PowerSelectionPolicyMinHostMips**: This class selects VMs from hosts with lower computing power to migrate to hosts with higher computing power.

B. *Allocation Algorithms*

The main task of allocation algorithms is to find suitable hosts for VMs selected for migration. Figure 5 shows the general class diagram of this section. Gray classes are pre-existing classes in CloudSim and blue classes have been added in this study.

- **PowerVmAllocationGreedyAbstract**: This class contains basic methods for allocating VMs to suitable servers based on greedy algorithms.
- **PowerVmAllocationMaxHostUtilization**: This class allocates VMs selected from the servers with low utilization to the servers with high utilization.
- **PowerVmAllocationMinHostPowerUsage**: This class allocates VMs selected from the servers with high power usage to the servers with low power usage
- **PowerVmAllocationMaxHostMips**: This class allocated VMs selected from the servers with lower computing power to the servers with higher computing power.

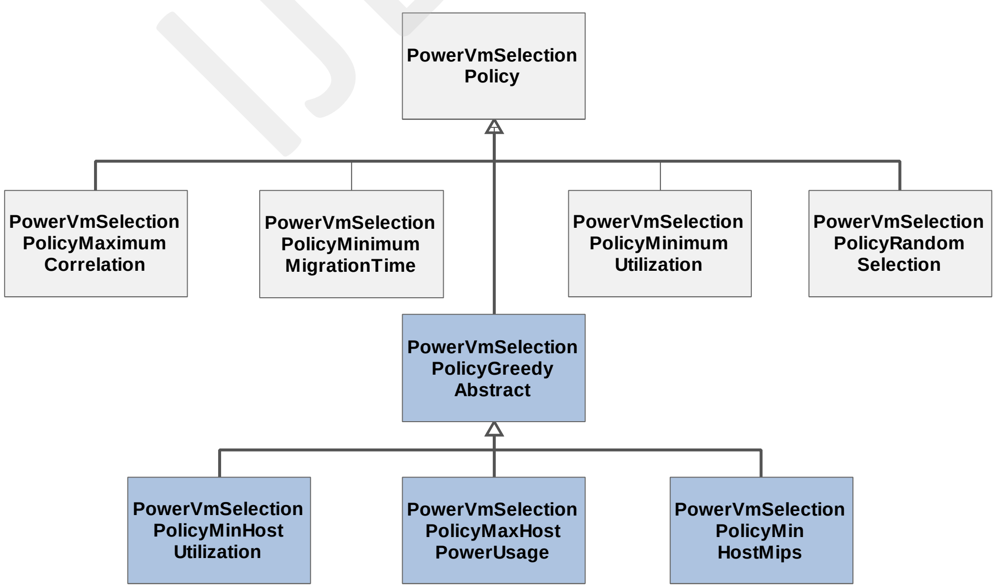

Figure 4 Class diagram of VM selection

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 9, April 2014)

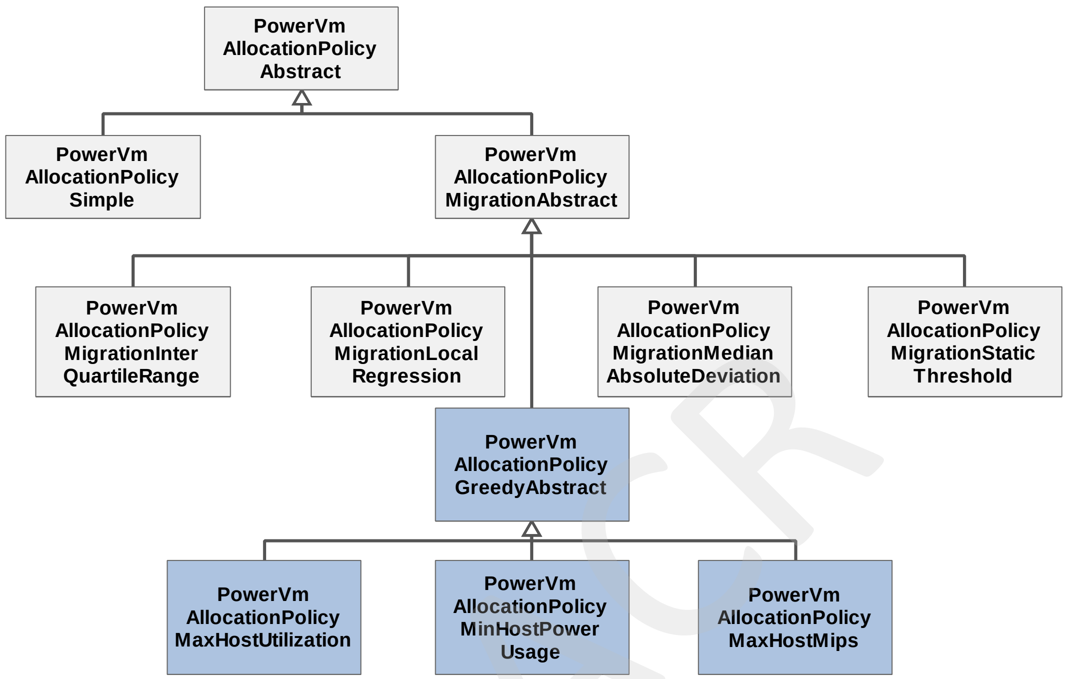

Figure 5 Class diagram of VM allocation

VI. Performance Criteria

In order to compare the performance of algorithms, some criteria are defined to evaluate their performance. The first measure is "Total Energy Consumption" by the physical resources of the data center; this amount of energy is consumed due to the workload of applications. The second performance measure is "Percent of SLA Violation"; by definition it is the ratio of number of SLA violation events to the total number of time frames have been processed. SLA violation occurs when a virtual machine fails to take the amount of MIPS requested from a host. This occurs when the consolidated virtual machines on a shared host cannot reach the needed performance. The third criterion is "the Number of VM Migrations" which is conducted by Virtual Machine Manager during VM movements. The last measure is "Average SLA Violation" that shows the average performance of the CPU when an application has requested it and the CPU has not been allocated to the application; therefore there was loss of productivity.

VII. Environment and Conditions of Simulation

It is obvious that the algorithms should be evaluated in a large and real data center. But these large-scale tests in a repeatable manner and in real environments are very difficult and sometimes impossible to perform. Therefore, in order to ensure the repeatability of the experiments, simulation is selected to evaluate the performance of the algorithms.

CloudSim[9] is a modern tool developed in order to simulate cloud environments and is used to evaluate greedy algorithms. It provides a new, general and extensible framework that enables the modeling and simulation of infrastructures and applications in an integrated simulated cloud computing environment. Using CloudSim, researchers and cloud providers can test a new application service in a controlled environment that is easy to set up and re-configure. It is possible to adjust performance of services appropriately based on the evaluation results which are reported by CloudSim. The main advantages of using CloudSim for initial performance testing include: time efficient, flexibility, and applicability.

Extended version of this tool has been developed for energy-aware simulations that facilitate evaluation of different algorithms based on various evaluation metrics [8]. To support modeling and simulation of different models of power consumption and power management techniques, such as DVFS [17], an implementation of an abstract class called PowerModel is provided. This abstract class should be extended for a customized simulation of the power consumption of a processing unit. This capability enables the creation of energy-aware allocation policies

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 9, April 2014)

that need to know about the real time power consumption of the cloud components. In addition, it can calculate the total amount of energy consumed by the system during the simulation period.

Simulated data center is an assumed data center that is made up of three types of servers in a total of 50 servers. General specifications of the servers[1] are in Table I.

TABLE I
GENERAL SPECIFICATION OF SERVERS

Server Name	# of CPU	# of Cores	MIPS	Memory (GB)
HP Proliant Ml110G Xeon 3040	1	2	1860	4
HP Proliant Ml110G Xeon 30 75	1	2	2660	4
IBM 3250 Xeon 3470	1	4	2933	8

These 50 servers are considered to run 100 virtual machines; VMs have four different types based on the recommendations of Amazon EC2[2]. Table II shows the properties of virtual machines.

TABLE II
VIRTUAL MACHINES PROPERTIES

VM Type	# of Required Cores	Required MIPS	Required Memory (MB)
1	1	2500	870
2	1	2000	1740
3	1	1000	1740
4	1	500	613

It is assumed that each virtual machine runs a web application or any other application with different working load.

VIII. SIMULATION RESULTS

Designed greedy algorithms are single threshold algorithms and the results of their implementation are compared to some single-threshold algorithms discussed in [8]. In addition, energy-unawrae algorithm and DVFS algorithm are also simulated; they are used as a basic measure for comparing performance of different algorithms. It should be pointed out that there is no migration of virtual machines between servers in the two mentioned algorithms.

In the simulation environment, energy consumption of energy-unaware algorithm and DVFS was estimated 72.6 kwh and 29.46 kwh respectively. For other algorithms, the utilization threshold was modified between 20% and 100% and the results were compared. The following charts (figure 6) summarize the results and the utilization based on various criteria.

Generally, the results of the greedy algorithms are better than the results of the algorithms presented in [8]. But it should be noted that in cases such as SLA violation, the results are weaker; because cloud providers are obligated to pay fines due to loss of service quality, these cases should be considered too.

Results of greedy algorithms show that the "Minimum Host MIPS Selection, Maximum Host MIPS Allocation (MinMax_HM)" method has better performance than the performance of the other algorithms. Although the SLA violation ratio and number of VMs migrations of the other two methods are less than VMs migrations of MinMax_HM method, but reduction of energy consumption in MinMax_HM is considerably further than energy consumption of the two other methods, as in the case of 100% server utilization, power consumption of MinMax_HM is about 58% of "Maximum Host Power Usage Selection, Minimum Host Power Usage Allocation (MaxMin_HPU)" method.

---

1 http://www.spec.org/power_ssj2008/results/

2 http://aws.amazon.com/ec2/instance-types/

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 9, April 2014)

Energy Consumption

	20	30	40	50	60	70	80	90	100
Random_Thr_Rs	45.25	33.85	24.96	23.8	17.12	14.66	12.81	12.21	11.51
Random_Thr_Mu	45.26	38.81	31.1	28.23	22.01	17.17	15.59	14.42	13.48
Random_Thr_Mmt	44.82	34.11	26.42	23.86	18.88	15.77	14.59	13.29	12.59
MinMax_HU	49.74	49.74	49.74	49.74	32.23	32.23	23.69	16.26	13.55
MaxMin_HPU	49.74	49.74	49.74	49.74	31.54	15.64	15.64	12.51	11.65
MinMax_HM	14.95	11.79	11.3	10.21	9.78	9.37	8.17	7.78	7.02

VM Migration

	20	30	40	50	60	70	80	90	100
Random_Thr_Rs	3090	3552	3422	4043	3240	3144	3003	3031	3021
Random_Thr_Mu	3856	5172	5215	5530	4550	3746	3557	3426	3381
Random_Thr_Mmt	3497	4437	4209	4445	3898	3515	3395	3373	3133
MinMax_HU	0	0	0	0	31	31	52	67	72
MaxMin_HPU	0	0	0	0	23	62	62	68	72
MinMax_HM	106	116	99	144	110	123	139	118	120

Overall SLA Violation

	20	30	40	50	60	70	80	90	100
Random_Thr_Rs	0.74	0.37	0.17	0.27	0.27	0.49	1.21	2.25	4.79
Random_Thr_Mu	0.76	0.46	0.37	0.42	0.39	0.34	0.86	1.63	2.78
Random_Thr_Mmt	1	0.45	0.13	0.17	0.19	0.53	0.97	2.58	3.56
MinMax_HU	0	0	0	0	0	0	0	0	0
MaxMin_HPU	0	0	0	0	0	0.01	0.01	0	0
MinMax_HM	0	0.1	0.1	0.82	2.1	2.81	9.66	12.83	20.29

Average SLA Violation

	20	30	40	50	60	70	80	90	100
Random_Thr_Rs	10.38	10.21	9.99	10.19	10.1	9.93	9.95	10.89	13.19
Random_Thr_Mu	10.35	10.22	10.21	10.3	10.02	10.15	10.44	11.51	12.53
Random_Thr_Mmt	10.71	10.38	9.98	9.98	10	9.84	10.34	11.51	12.82
MinMax_HU	0	0	0	0	10	10	10	10	10
MaxMin_HPU	0	0	0	0	10	9.62	9.62	10	10
MinMax_HM	10	8.27	9.27	8.76	9.78	11.59	16.24	17.23	22.25

Host Shutdowns

	20	30	40	50	60	70	80	90	100
Random_Thr_Rs	657	884	695	894	775	676	575	574	546
Random_Thr_Mu	666	1024	1008	1140	1051	850	800	684	671
Random_Thr_Mmt	703	1182	799	969	890	754	677	648	613
MinMax_HU	0	0	0	0	16	16	25	32	35
MaxMin_HPU	0	0	0	0	16	32	32	34	36
MinMax_HM	45	41	40	48	44	45	48	46	46

Figure 6 Simulation results

IV. Conclusions

Cloud computing is one of the ways used in data centers to provide better and more qualified services. In the architecture of cloud computing, services are provided based on three general types: Infrastructure as a Service (SaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). In the cloud computing, service is separated from the location of service and consumers do not need to know about the physical location of hardware and software and thus the quality of service in cloud computing has improved. But rapid growth of Internet and Web services increases the demand for cloud computing services and causes the need for more hardware in their data centers which leads to more energy consumption. The energy consumption in data centers has increased and so the cloud service providers have to maintain quality of service, while avoiding energy consumption exceedingly in data centers.

Virtualization is one of the technologies used in the data center for better utilization of hardware resources. One of the main characteristics of virtualization is migration of virtual machines between physical nodes. VMs can migrate between hosts dynamically. This feature helps to consolidate VMs in servers and thus it increases efficient use of servers. Various algorithms have been designed and implemented to select virtual machines on a host and allocate them to other hosts that mainly aimed at improving the energy efficiency of data centers.

In this research, three approaches based on greedy algorithms in terms of energy consumption have been examined. Three algorithms are as follows: MinMax_HU, MaxMin_HPU, MinMax_HM. Simulation in CloudSim results in that the three algorithms improved energy consumption considerably compared to the energy-unaware and DVFS algorithms; and also they have better consequences than algorithms that separate selection methods from allocation methods. But in some cases such as SLA violation results were not appropriate and should be considered further.

In continuation of this research, the following issues will be discussed: Designing and implementing double-threshold greedy algorithms, improving MinMax_MH

IJDACR International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 9, April 2014)

algorithm to reduce SLA violation, and designing and implementation of greedy algorithms for other data center resources such as memory, disk and network interface.

REFERENCES

[1] D. Rogers and U. Homann, “Application patterns for green IT,” Archit. J., vol. 18, pp. 16–21, 2008.

[2] G. Rangaraj and R. Bahsoon, “Green Software Architectures: A Market-Based Approach,” in The Second International Workshop on Software Research and Climate Change (WSRCC), in affiliation with the ACM/IEEE 32nd International Conference on Software Engineering (ICSE), Cape Town, South Africa, 2010.

[3] A. Spellmann, R. Gimarc, M. Preston, and O. I. Hyperformix, “Leveraging the cloud for green IT: predicting the energy, cost and performance of cloud computing,” in Computer Measurement Group Conference (CMG’09), Dallas, Texas, 2009.

[4] H. Katzan Jr, “Cloud software service: concepts, technology, economics,” Serv. Sci., vol. 1, no. 4, pp. 256–269, 2009.

[5] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis, “Energy-efficient cloud computing,” Comput. J., vol. 53, no. 7, pp. 1045–1051, 2010.

[6] F.-S. Chu, K.-C. Chen, and C.-M. Cheng, “Toward green cloud computing,” in Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication, 2011, p. 31.

[7] R. Bahsoon, “A framework for dynamic self-optimization of power and dependability requirements in Green Cloud architectures,” in Software Architecture, Springer, 2010, pp. 510–514.

[8] A. Beloglazov and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers,” Concurr. Comput. Pract. Exp., vol. 24, no. 13, pp. 1397–1420, 2012.

[9] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” Softw. Pract. Exp., vol. 41, no. 1, pp. 23–50, 2011.

[10] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,” Futur. Gener. Comput. Syst., vol. 28, no. 5, pp. 755–768, 2012.

[11] V. Sivaraman, A. Vishwanath, Z. Zhao, and C. Russell, “Profiling per-packet and per-byte energy consumption in the NetFPGA Gigabit router,” in Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on, 2011, pp. 331–336.

[12] S. K. Garg, C. S. Yeo, and R. Buyya, “Green cloud framework for improving carbon efficiency of clouds,” in Euro-Par 2011 Parallel Processing, Springer, 2011, pp. 491–502.

[13] M. A. Salehi, P. Radha Krishna, K. S. Deepak, and R. Buyya, “Preemption-Aware Energy Management in Virtualized Data Centers,” in Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, 2012, pp. 844–851.

[14] S. K. Garg, S. K. Gopalaiyengar, and R. Buyya, “SLA-based resource provisioning for heterogeneous workloads in a virtualized cloud datacenter,” in Algorithms and Architectures for Parallel Processing, Springer, 2011, pp. 371–384.

[15] M. Choudhary and S. K. Peddoju, “A dynamic optimization algorithm for task scheduling in cloud environment,” Int. J. Eng. Res. Appl., vol. 2, no. 3, pp. 2564–2568, 2012.

[16] N. S. Ram Kumar Sharma, “A Dynamic Optimization Algorithm for Task Scheduling in Cloud Computing With Resource Utilization,” Int. J. Sci. Eng. Technol., vol. 2, no. 10, pp. 1062–1068, Oct. 2013.

[17] A. Beloglazov and R. Buyya, “A taxonomy and survey of energy-efficient data centers and cloud computing systems,” Adv. Comput., vol. 82, no. 2, pp. 47–111, 2011.