Heuristic based Energy-aware Resource Allocation by Dynamic Consolidation of Virtual Machines in Cloud Data Center

Md. Sabbir Hasan, Eui-Nam Huh*
Department of Computer Engineering, Kyung Hee University
Republic of Korea
[e-mail: sabbir@khu.ac.kr, johnhuh@khu.ac.kr]
*Corresponding author: Eui-Nam Huh

Received April 8, 2013; revised June 5, 2013; accepted July 8, 2013; published August 30, 2013

Abstract

Rapid growth of the IT industry has led to significant energy consumption in the last decade. Data centers swallow an enormous amount of electrical energy and have high operating costs and carbon dioxide excretions. In response to this, the dynamic consolidation of virtual machines (VMs) allows for efficient resource management and reduces power consumption through the live migration of VMs in the hosts. Moreover, each client typically has a service level agreement (SLA), this leads to stipulations in dealing with energy-performance trade-offs, as aggressive consolidation may lead to performance degradation beyond the negotiation. In this paper we propose a heuristic based resource allocation of VM selection and a VM allocation approach that aims to minimize the total energy consumption and operating costs while meeting the client-level SLA. Our experiment results demonstrate significant enhancements in cloud providers' profit and energy savings while improving the SLA at a certain level.

Keywords: Cloud Computing, Green IT, Resource Management, dynamic consolidation, Virtualization, Service Level Agreement

http://dx.doi.org/10.3837/tiis.2013.08.005

1. Introduction

Cloud Computing has received significant attention in recent times. The hype has largely been created and responded by companies such as Amazon, IBM, Google, Yahoo!, Microsoft, Sun, NASA and RackSpace by making their own cloud platforms for consumers and enterprises to access cloud resources through these services. The rapid development of virtualization technology, including the advantages of isolation, consolidation and multiplexing of resources, became a key role to deploy in modern data centers [1]. Due to virtualization, numerous tasks are seen as a single entity in a virtual machine. However, virtualization fetches another abstraction layer that impedes conventional energy and resource management techniques from performing proficiently [2]. Virtualization opens new capabilities such as live migration [3], which has attracted substantial attention in recent years to respond to the challenges of cloud computing. It does this by providing load balancing, power efficiency, and transparent infrastructure maintenance to the virtual machines, which requires new management logic.

The problem with data centers is high energy consumption, which has risen by 56% from 2005 to 2010, and in 2010 accounted for 1.15% to 1.5% of global electricity use [4]. In addition, high energy ingestion by the infrastructure leads to substantial carbon dioxide (CO_2) emissions causative to the greenhouse effect [5]. The energy consumption can be firmed by the resource management system deployed in the infrastructure and the efficiency of applications running in the system. To improve the utilization of data center resources, virtual machine consolidation has been shown to be efficient [7][8][9][10][11][12]. This process leverages live migrations of VMs with changing workloads in the physical host to reduce the number of physical hosts or servers. To eliminate the static power, idle servers are switched to sleep mode to reduce the energy consumption; idle hosts can be reactivated when the resource demand increases. However, infrastructure providers often end up over-provisioning resources to maximize the quality of service; this results in high energy consumption, poor resource management and large operational costs. QoS requirements can be formalized in a service level agreement that serves as the foundation for the expected level of service between the cloud consumer and the service or cloud provider. The purpose of optimal resource provisioning through VM consolidation is to make an energy-performance trade-off rather than only minimizing the power consumption. Optimal resource allocation and management is challenging due to the diversity present in the clients' applications and the dynamic workloads that have to be processed by hosts in the data center.

In our work, we articulate the VM resource allocation problem into a multidimensional bin-packing problem, which is NP-complete [13]. Similar to this problem, we considered VMs as items, dimensions as their capacity and hosts as bins. Our estimation on overload detection provides more dynamicity to resource allocation techniques and adopts a different environment that is independent of the workload. Our proposed technique is more robust in regards to the problem of overload detection; this reduces the number of unnecessary migrations of virtual machines and decreases power consumption and performance degradation. Specifically our work aims to:

- Define an architectural framework for optimal resource allocation and management.

- Examine heuristic based energy-aware resource provisioning without violating the negotiated SLA.
- Provide dynamicity and self-managing changes for allocation of resources for dynamic and unpredictable workloads.
- Develop efficient algorithms for VM consolidation and mapping to hosts to reduce energy consumption and operational cost.

Several experiments have been carried out with real life workloads to validate the efficiency of our proposed VM resource allocation policy and algorithm. We have compared our results with several researchers ([14][15]) and found outstanding improvements including reduced power consumption up to 36.37%, improvement of the SLA up to 15% and a 46.25% reduction of operational costs. The remainder of this paper is organized as follows: Section 2 describes some work related to our research, Section 3 defines an architectural framework for optimal resource allocation and management, our heuristic based approach is proposed in Section 4, Section 5 has the performance evaluation and analysis, and Section 6 concludes the paper with our future research issues and conclusion.

2. Related Work

Our work is based on the dynamic consolidation of virtual machines and retaining a strict service level agreement to consumers while considering the minimization of energy consumption. There are several research groups in both academia and industry working on energy aware resource allocation and management by performing static and dynamic consolidation of VMs and servers.

Bobroff et al. [16] proposed and evaluated a dynamic server consolidation to reduce the amount of physical capacity while maintaining low SLA violation. The algorithm uses historical data for demand forecasting to periodically minimize the number of physical servers. In contrast, we consider the reduction of energy consumption and the cloud provider's SLA penalties while confirming a lesser number of VM migrations and physical servers used during the whole execution period. Cardosa et al. [17] proposed running heterogeneous applications as a solution for VM placement and power-efficient consolidation of VMs in modern data centers. They adopted min, max, and share parameters of XEN and VMware, these represent the utilization limit of upper and lower CPU allocation and sharing of the same resources by different VMs. They also considered a priority based approach for the peak load of the enterprise environment. As a result, it does not support strict SLA and the VM allocation is static.

Carrera et al. [18] investigated the performance degradation in multi-tier web applications hosted on a cloud. They designed and implemented algorithms for automatic bottleneck detection and resolution to make to make resource management dynamic and SLA-driven. However their investigation is not suitable for scenarios where power consumption should be reduced and cost optimized. Verma et al. [19] described a power aware application placement framework in which at each time frame the placement of VMs is optimized to minimize the power consumption and maximize the performance at certain levels. The main difference between his work and ours is that our proposed algorithm does not violate strict SLA requirements when the workload is varied and unpredictable.

Stilwell et al. [20] proposed a formulation of the resource allocation problem in a shared hosting platform for static workloads with servers that provide multiple types of resources. Their algorithm runs faster in large systems and fulfills QoS requirements but it lacks dynamicity when the workload is unpredictable and dynamic. Like Stilwell other researchers ([21] [22]) also studied VM resource management techniques to maintain QoS requirements when the workload is static in cloud computing. Wood et al. [23] developed the Sandpiper System that monitors and detects hotspots and reconfigures VMs when necessary. In order to choose which VMs to migrate, their system sorts them using volume-size-ratio, which is a metric based on CPU, network and memory loads; whereas we consider both the size of the VM and the migration time required to maintain a strict SLA.

Ferreto et al. [15] proposed server consolidation with migration control to reduce the number of migrations of VMs with minimal penalty in the number of physical servers. They constructed an LP formulation and heuristics to control VM migration, which prioritizes virtual machines with a steady capacity. However, our experimental results show greater improvement in power reduction and cost optimization than their approach. Anton et al. [14] designed and implemented an online algorithm and adaptive heuristic based approach for an energy efficient data center. They constructed several approaches to find the trade off point where power consumption is decreased but a strong SLA is maintained. In contrast to their approach, our algorithms include host utilization, power consumption, and migration time of VMs. Further modification of their approach shows better results in terms of cost optimization, and reduced power consumption and SLA violations. Our estimator for host overload detection provides better consolidation of VMs as it reduces the number of unnecessary VM migrations.

3. Proposed Framework

Fig. 1 depicts our proposed system architecture. Our only consideration is infrastructure as a service (IaaS).To maintain the scalability, have efficient use of resources and provide on demand service to meet the service level agreement we consider several modules that can provide better monitoring and on demand provisioning of resources.

Over-provisioning of resources can fulfill the service level agreement in greater extent but from the cloud provider’s view it is considered non-profitable as it can increase the resource overhead of the system. A demand estimator module estimates the amount of resources needed to serve one application’s demand based on the feedback from an available capacity estimator and the overload of the system infrastructure. The demand estimator plays an important role as the decision maker to serve applications with proper SLAs. If overload is detected, the demand estimator can also report the recovery or migration of VMs from the serving host to another host. On the other hand, a workload profiler allocates resources depending on the variability and the characteristics of different types of application workloads via an executor. The resource allocator allocates the resources for the application or for the whole session of the task; this is based on the estimation from the demand estimator and the characteristics of the workload from the workload profiler. The resource allocator is loosely coupled with the system so that addition and removal of resources can be done in run time. Usually data centers receives various types of workloads from various geographic locations, so synchronization between VMs, physical hosts and network connection is more important these days. For this reason we illustrate registration and synchronization, session manager and decommissioning of the session for a user or period of time in the virtual machine manager (VMM). The jobs scheduler/queue manager can be designed as either the cloud provider's or user's perspective for different kinds of workloads or jobs (e.g., arrival rate, polynomial distribution, and round robin). Therefore, in order to correctly allocate resources and serve the service request with a strict SLA, an efficient, cost effective and optimal VM resource allocation algorithm is necessary.

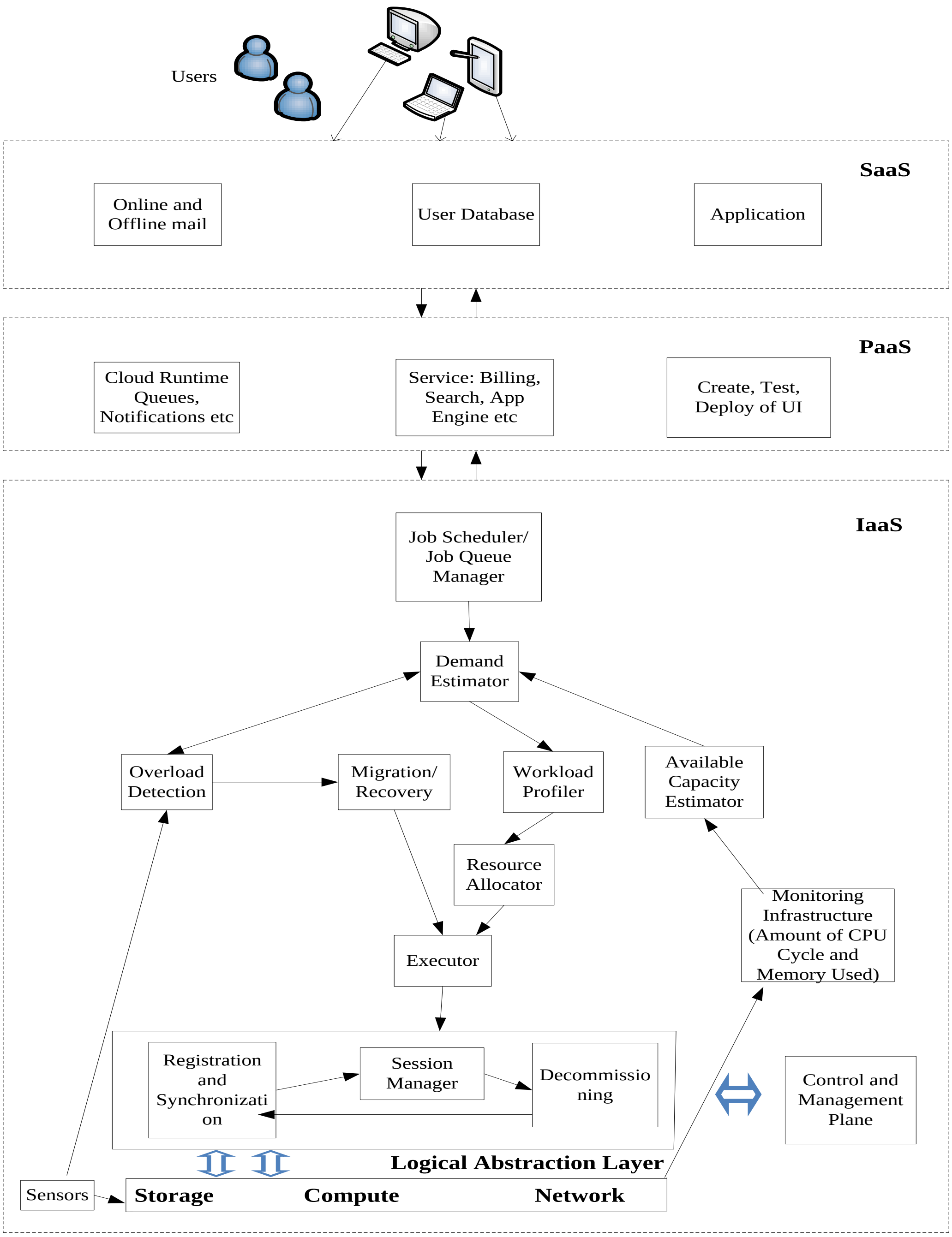

Fig. 1. Resource allocation framework for infrastructure-as-a-service

4. Heuristics for Dynamic VM Consolidation

Based on an analysis of historical data of the resource usage by VMs, we propose several heuristics for dynamic consolidation of VMs. Our heuristics include: (1) Determining the dynamic upper utilization threshold for overload detection, (2) Selecting suitable VMs for migration from an overloaded host, (3) Finding new placements for the VM selected for migration from the overloaded host, and (4) Determining a host considered as under loaded and migrating all of the VMs from the under loaded host.

4.1 Detection of Host Overloading

Some of the researcher's proposed heuristic for determining the time to migrate a VM from a host is based on a utilization threshold [21] [24]. Normally their idea is based on setting upper and lower utilization thresholds for hosts and keeping the total utilization of the CPU by all the VMs between these thresholds. In [21], they set 25% and 75% of the total utilization as the lower and upper utilization thresholds, respectively. In contrast to this, Anton et al. [24] experimented with different values for the lower and upper thresholds and found different levels of power consumption based on increasing and decreasing the values. Moreover, if the host utilization falls under the lower utilization thresholds, all VMs have to be migrated from the host; the host has to be in the sleep mode in order to obliterate the idle energy consumption. To maintain a strong SLA, the opposite occurs when the host gets overloaded.

In the case of dynamic, unpredictable workloads and different types of applications that share the same physical resources, fixed values for the utilization threshold are not suitable. Statistical analysis of historical data techniques plays an important role in adjusting the threshold values based on the behavior of the workload pattern. For classic parametric tests to produce accurate results, the assumptions underlying them (e.g., normality and homoscedasticity) must be satisfied. These assumptions are rarely met while analyzing real data. To get an accurate computation of *P* values, effect sizes and confidence intervals, robust methods are more effective than classical methods [25]. Anton et al. [14] provided four robust statistical methods that can be used as an estimator to design the threshold value. Among them, MAD is a more robust estimator of scale than the sample variance or standard deviation. We propose as an alternative to MAD, a more robust estimator that can be used as an initial or ancillary scale. It estimates the same way but is more efficient and not imbalanced towards symmetric distribution [26]. The estimator can be expressed for univariate data sets of m- $X_1, X_2, ..., X_n$:

$$E_n = cmed_i\{med \mid X_i - X_j \mid\} \tag{1}$$

$$T_u = 1 - s.E_n, \qquad \text{where, } s \in R, s > 0 \tag{2}$$

For each i we compute the median of $\{\mid X_i - X_j \mid; j = 1, ..., n\}$. This yields n number,

the median of which gives our final estimation E_n from equation (1) (The factor C is for consistency). We define the upper threshold and introduce a safety parameter in (2). We can adjust the safety of the method by controlling the parameter. The higher the value of s, the lower the level of the SLA violation, but higher the energy consumption caused by consolidation.

4.2 VM Selection

Our policy selects and migrates a VM to lower the CPU utilization if the CPU utilization of a host goes beyond the upper utilization threshold. We consider both the VM's CPU utilization and migration time when determining which VM has to be migrated. The migration time can be estimated as the amount of memory used by the VM divided by the bandwidth availability of that particular host [14].

Algorithm 1: VM Selection Algorithm

```
Input: hostList  Output: migrationList
foreach h in hostList do
      vmList ← h.getVmList ( )
      vmList.sortDecreasingUtilization ( )
      hUtil ← h.getUtil ( )
      bestFitUtil ← MAX
   while hUtil > THRESH_UP do
      foreach vm in vmList do
         if vm.getUtil ( ) > hUtil – THRESH_UP then
            t ← vm.getUtil ( ) – hUtil + THRESH_UP
            r ← vm.getRam ( )
            c ← sqrt (sqr (t) + sqr (r) )
            if c < bestFitUtil then
               bestFitUtil ← c
               bestFitVm ← vm
            end
         else
            if  bestFitUtil = MAX then
              bestFitVm ← vm
            break
         end
      end
   end
      hUtill ← hUtil-bestFitVm.getUtil ( )
      migrationList.add(bestFitVm)
      vmList.remove(bestFitVm)
return migrationList
```

Let M_j be a set of VMs currently allocated to the host j. We calculate the value of c by taking the square of both the utilization and the amount of RAM used by the VMs in Host j and then taking the root of the value. From the result we consider the lowest value and, if needed, round the value to get the desired VM to be migrated.

$$c = \sqrt{\{u_k(M_j)\}^2 + \{RAM_u(M_j)\}^2} \tag{3}$$

$$0 < c_s < c_a, \qquad c_s \in c, \forall c_a \in c, s \neq a \tag{4}$$

$$u_k(s) > | u_j - T_u |, \qquad s \in M_j \tag{5}$$

$$\frac{RAM_u(s)}{SBw_j} \leq \frac{RAM_u(a)}{SBw_j}, \quad s \in M_j, \forall a \in M_{j,} s \neq a \tag{6}$$

Where $u_k(s)$ is the fraction of CPU utilization by the VM; u_j is the current host CPU utilization; T_u is the upper utilization threshold; $RAM_u(s)$ and $RAM_u(a)$ are the amount of RAM currently utilized by the VMs s and a.

The pseudo-code of the VM selection for the over-utilization case is presented in Algorithm 1. The algorithm sorts the VMs in decreasing order then finds a VM in the list based on the decision and returns the migration list. The algorithm selects a VM if it satisfies the condition. It first sorts those VMs that have higher CPU utilization than the difference between the current host utilization and the upper utilization threshold value. Then it calculates the migration time for all VMs and gets the desired VM by taking the minimum value from equation (3). If there is no such VM, it will select the best utilized VM for migration from the host.

4.3 VM Placement

The problems with VM allocation can be divided in two ways: the admission of new requirements for VM provisioning and enlisting the VMs in the host, and the optimization of current VM allocation. The first can be seen as a bin packing problem with variable bin size and prices, where bins represent the physical hosts, bin sizes are the available CPU capacity and bin processes are the energy consumption. We solve the first problem by using the best fit decreasing algorithm that is shown to use no more than bins $\frac{11}{9}.OPT+1$ [27].

Algorithm 2: VM Allocation Algorithm

```
Input: bestFitVm, vmList, hostList  Output: Allocation of VM's
foreach bestFitVm in vmList do
|     minPower ← Max
|     allocatedHost ← Null
|     maxHost ← MIN
|     mindiagonal ← MAX
|   foreach host in hostList  do
|   |  if bestVmUtil() < THRESH_UP – hUtil() then
|   |   |  powerdiff ← powerAfterAllocation – getPower (host)
|   |   |  hUtil ← getUtilizationOfCpu (host)
|   |   |  A← sqrt (sqr (powerdiff) + sqr (hUtil))
|   |   |  if   A < mindiagonal then
|   |   |   |   allocatedHost ← A
|   |   |  end
|   |  else
|   |   |  if hUtil > maxHost
|   |   |  | allocatedHost ← host
|   |   |  end
|   |  end
|  end
|
|   if  allocatedHost ≠ NULL then
|   |  allocate vm to allocatedHost
```

```
    end
end
return allocation
```

However, to solve the second problem we consider power consumption and CPU utilization of hosts. The pseudo-code for the VM placement is presented in Algorithm 2. We calculate the other available host's current CPU utilization and increase of power consumption if the selected VM had been migrated. By using $\sqrt{x^2+y^2}$, we indicate the tradeoff point of the CPU utilization and increased power consumption of the hosts and find a suitable host for the VM. Otherwise it will find a host in which the utilization is calculated greater so that the host doesn't get overloaded if the VM is migrated to it.

4.4 Detection of Host Under Loading

We propose a very simple approach for host under load detection to reduce the power consumption and the number of running physical hosts in a data center. Our approach is to find the hosts with less CPU utilized compared to other hosts and try to move the VMs to other hosts until these hosts get overloaded. If this can be achieved then we set VMs for migration from the host and put the host in shutdown mode to reduce energy consumption, as, on average, an idle server consumes approximately 70% of the power consumed by a server running at the full CPU speed [24]. This is done iteratively for all hosts that are not considered overloaded. This approach certainly reduces the number of hosts required to accomplish the tasks shown in the experimental results.

4.5 SLA Metrics and Cost Function for Cloud Providers

Maintaining a strong SLA and meeting QoS requirements is enormously important for a cloud computing environment. Overprovisioning of resources to meet the target SLA can increase the power consumption, as energy-aware resource management is highly important these days. QoS requirements can be governed by service response time or throughput, and these can be varied for different applications. However for IaaS, workload independent metrics should be considered to calculate the SLA percentage or violation. For measuring the SLA violation in an IaaS environment we consider two metrics: the percentage of time an active host has experienced 100% CPU utilization and performance degradation due to VM migration from one host to another. These same metrics have been considered by past researchers [14], but our result shows better improvement in maintaining the SLA when considering these two metrics. We consider SLAs as delivered when 100% of an application's requirements have been served by a VM. However, when a host experiences 100% CPU utilization, the performance of applications degrades due to a lack of capacity.

$$\text{SLA time per active host} = \frac{1}{N}\sum_{k=1}^{N}\frac{X_{sk}}{X_{ak}} \tag{7}$$

$$\text{Performance degradation due to migration} = \frac{1}{M}\sum_{l=1}^{M}\frac{P_{dl}}{P_{al}} \tag{8}$$

Where N and M are the number of hosts and VMs respectively; X_{sk} is the total time when host k has experienced 100% CPU utilization and caused a SLA violation; X_{ak} is the total time the host was active; P_{dl} is the approximation of performance degradation due to migration of VM l (we consider it 10%) [14]; P_{al} is the total CPU capacity requested by VM l during its epoch.

Based on the above two metrics we formulate a cost function to calculate the total cost that one cloud provider has to pay for our scenario. We consider the power consumption, number of VM migrations and number of active hosts during the execution of tasks and the SLA violations to formulate the cost function.

$$C_p.E + C_k(\sum_{i=1}^{M} P_i.P_{mj}) + C_l(\sum_{i=1}^{N} R_i.R_{nj}) + (\frac{1}{N}\sum_{k=1}^{N}\frac{X_{sk}}{X_{ak}}.\frac{1}{M}\sum_{l=1}^{M}\frac{P_{dl}}{P_{al}}) \times C_u.v + \alpha \qquad (9)$$

Where C_p, C_k, C_l are the costs for energy consumption, for the active host during task execution time and for VM migration, respectively; C_u is the pay per use of the user or customer and v is the factor for an SLA penalty; P_i is the number of hosts shutdown and P_{mj} is the mean time before the host shutdown; R_i and R_{nj} corresponds to the number of VM migrations and mean time before a Vm migrates to the other host; α is the co-efficient that includes the costs of memory, RAM, cooling, space, server racks, etc. In our experiments we haven't considered memory and RAM when calculating power consumption as the CPU holds the major portion of this.

5. Performance Evaluation

5.1 Test Bed Setup

We evaluate our proposed algorithm in the CloudSim toolkit. CloudSim is an extensible simulation toolkit that enables modeling and simulation of cloud computing systems and an application provisioning system created by CLOUDS Lab, University of Melbourne [28]. We chose this as it is enormously difficult to conduct large-scale experiments in real infrastructure. Eight hundred heterogeneous physical nodes are used to do the experiment. The server model is HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores $\times$1860 MHz, 4 GB) and HP ProLiant ML110 G5 (Intel Xeon 3075, 2 cores $\times$2660 MHz, 4 GB). For our simulation we considered MIPS ratings (instead of MHz) for the two servers as 1860 MIPS and 2660 MIPS, respectively. Four types of VMs were used as the High-CPU medium instance (2500 MIPS, 0.85 GB), Extra Large instance (2000 MIPS, 3.75 GB) Small instance (1000 MIPS, 1.7 GB) and Micro Instance (500 MIPS, 613 MB). In our experiments, we extended CloudSim for simulating the function to dynamic consolidation of VMs. The PowerVmAllocationPolicyMigrationAbstract and PowerVmSelectionMinimumMigrationTime classes are modified according to our proposed algorithm. We have conducted experiments with real life data provided by the CoMon project, a monitoring infrastructure for PlanetLab [29]. We have used CPU utilization data from more than a thousand VMs from different geographic places. We have traced random workload from four days in March 2011; the average CPU load was below 60% and the workload was assigned to VMs randomly. We compared local regression- minimizing migration time (LR-MMT), best fit decreasing (BFD), and first fit decreasing (FFD) to show the improvement of our results. The LR-MMT method was so far the best approach in this area.

Table 1. Summary of test bed characteristics

Simulator	CloudSim
Number of Host	800
Host features	Intel Xeon 3040, 2 cores ×1860 MHz, 4 GB, Intel Xeon 3075, 2 cores ×2660 MHz, 4 GB
Number of VMs	1052, 898, 1061, 1516
VMs Feature	Amazon EC2 instance type [30]
Workload Data	CPU Utilization From PlanetLab for four days
Control Time	5 minutes
Power Consumption Cost for Provider	$1.5
Active Host Cost for Provider	$.25
VM Migration Cost for Provider	$.15

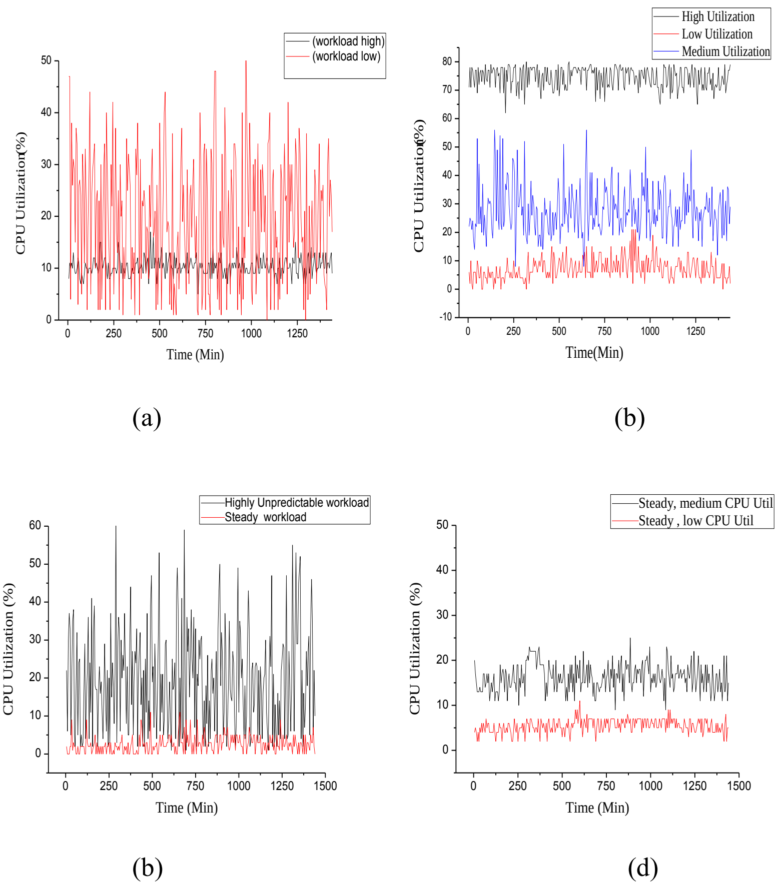

Fig. 2. Workload variation from day 1 to 4.

Table 2. Summary of test bed characteristics

Algorithm	Power Consumption (Kwh)	Number of VM Migration	SLA violation ($\times 10^{-5}$)	Operational Cost ($/Hour)
LR-MMT	143.395	23027	4.00	552.571
BFD	100.375	18417	3.124	350.41
FFD	103.445	20887	3.245	381.89
Proposed Algorithm	96.56	10445	3.37	297.02

5.1 Power Consumption

Our proposed algorithms show significant reduction of power consumption when compared with Anton et al. [14] algorithm and solution proposed by [15]. When considering the VM placement from the overloaded host to the other host, our algorithm had better results than others for both the increase of power consumption and CPU utilization. There was approximately a 36.37% power consumption reduction compared to LR-MMT[14], and reductions of 9.21% and 11.76% compared to traditional BFD and FFD algorithms, respectively.

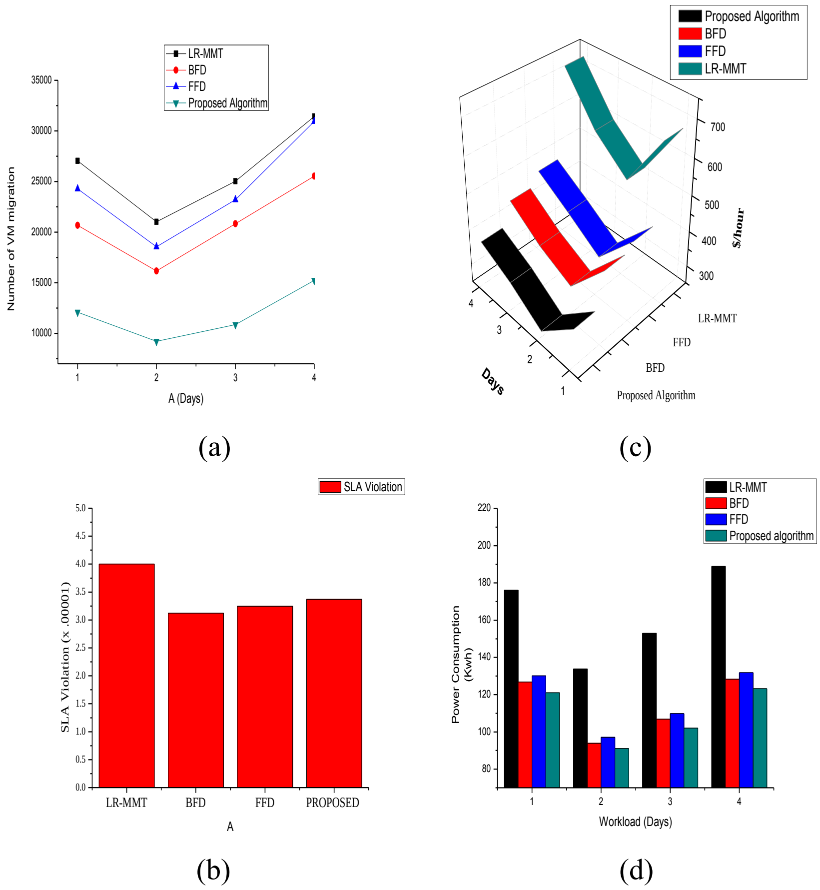

Fig. 3. (a) Number of VM migrations, (b) SLA violation comparison, (c) Operational cost per hour, and

(d) Power consumption.

Recent [31][32] studies have shown that the application of DVFS on the CPU results in almost linear power-to-frequency relationship for a server. Moreover, these studies show that on average an idle server consumes approximately 70% of the power consumed by the server running at the full CPU speed. We mentioned two types of server model which were used in our simulation named HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores 1860 MHz, 4 GB) and HP ProLiant ML110 G5 (Intel Xeon 3075, 2 cores 2660 MHz, 4 GB). Power consumption in watts by those servers is given below [14].

Table 3. Power consumption by the selected servers at different load levels in Watts

Server	*0%*	*10%*	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*	*90%*	*100%*
HP ProLiant G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	93.7	97	101	105	110	116	121	125	129	133	135

So from the above table, it can be shown that power consumption increases only 31-40 watts for “No CPU load” to “Full CPU load server”. If we could shut down these least load servers, enormous amount of energy can be saved and that was our main motivation in this paper. Comparing to the above scenario, VM migration consumes negligible energy (provided that an idle server consumes 70% of the power consumed by full CPU load server), but we have not neglected this factor rather we considered 10% of CPU utilization during all migrations of the VM occurs in the servers that results more energy consumption. Moreover, CPU utilization needed for VM migration was considered while calculating the SLA violation.

5.3 SLA Violation

In order to compare the efficiency we have also evaluated performance metrics such as SLA violations with other researchers’ proposed algorithms. Our result not only shows a better maintaining SLA, rather it demonstrates rigorous reduction in power consumption while maintaining SLA violations at a manageable level. From the experiment we calculated the amount of time an active host experienced 100% and caused performance degradation for the application in this epoch, and the number of VM migrations that occurred due to VM consolidation. We did not do aggressive consolidation; otherwise this would have resulted in an increase in the SLA violations. By dynamically selecting the upper threshold and efficiently placing the VMs in the hosts, a decreased number of migrations and higher rate of maintaining the SLA occurs. Fig. 3 (b) shows a 15% improvement of maintaining the SLA compared to the LR-MMT algorithm and a degradation of 6.1% and 3% relating to BFD and FFD, respectively, while keeping energy consumption at a very decent level. Our algorithm makes a tradeoff between user or customer satisfaction and power consumption; this enormously increases the chance of the cloud provider having profit.

5.4 Cost Optimization

Fig. 3 (c) illustrates the operational cost per hour by our proposed approach compared to other algorithms. The analysis shows that the operational cost reduced to 46.25%, 15.24% and 22.3%

compared to LR-MMT, BFD and FFD, respectively. This happens due to decreased power consumption, a low number of VM migrations and an active host during task execution. The SLA plays an important role when constructing the cost function as an increase in violations of the SLA causes SLA penalty that results in more operational cost.

Table 4. Comparisons of the estimation function of overload detection.

	Workload 1		Workload 2		Workload 3		Workload 4	
Metrics	MAD-MMT	Proposed	MAD-MMT	Proposed	MAD-MMT	Proposed	MAD-MMT	Proposed
Energy Consumption (Kwh)	184.88	120.97	141.28	91.03	162.72	102.09	197.33	123.24
Number of VM Migrations	26292	12094	21111	9221	23314	10868	28628	15241
SLA ($\times 10^{-5}$)	3.31	2.86	3.78	2.67	3.86	3.88	3.02	3.97
Number of Host Shutdowns	5759	989	4663	955	5123	1052	5927	1040
SLA Time per active Host	5.03%	4.69%	5.23%	4.51%	5.23%	5.14%	5.04%	5.88%

5.5 Estimation Function

Table 3 illustrates the impact on host overload detection of our estimation function E_n. We compared our proposed scheme with the MAD-MMT (median absolute deviation-minimization of migration) approach [14] where they used the median absolute deviation from the data sets of 5 minutes to estimate the next 5 minute upper utilization threshold for particular host. We used a modified median absolute deviation rather their approach. The results show significant reduction of power consumption and unnecessary VM migration; negligible performance degradation occurs for workload types 3 and 4. Our approach adapts well with different workload scenarios and can be implemented in large scale data centers for cloud computing environments.

6. Conclusion

Energy consumption is a critical issue for large-scale data centers which crowd thousands of hosts and cooling equipment. Additionally, cloud providers' profit and providing a negotiated SLA is becoming more concerning these days. Virtualization enables all of these possibilities for the modern world by providing consolidation and dynamism. However, it requires new policies for managing the new virtualization proficiencies.

Our approach in this study reduces a significant amount of energy consumption and operational costs under certain QoS requirements. We use the CloudSim simulator to justify our approach compared with other researchers using a real-life workload. Analysis shows a reduction of power consumption up to 36.37%, an improvement of the SLA up to 15% and an increase in the cloud providers' profit to 46.25%. However, in this proposed approach we have not accounted for network considerations such as bandwidth, parallelization, or co-location. As data centers from different geographic places serve user requests, network condition and latency can be a major concern for research. Our future work will focus on extending our current approach with different network topologies in data centers.

7. Acknowledgement

This work was supported by a post doctoral fellowship from the Kyung Hee University, Global Campus, Korea in 2011 (KHU20110217).

References

[1] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Computing*, pp. 14–22, 2009. Article (CrossRef Link)

[2] I. Goiri, J.Ll. Berral, J. Fito, F. Julia, R. Nou, J. Guitart, R. Gavalda, and J. Torres. "Energy-efficient and multifaceted resource management for profit-driven virtualized data centers," *Future Generation Computer System*, 28:718–731, 2012. Article (CrossRef Link)

[3] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proc. of the Second Symposium on Networked Systems Design and Implementation* (NSDI'05), 2005. pp. 273–286. Article (CrossRef Link)

[4] The Green grid Consortium 2011. URL http://www.thegreengrid.org/

[5] J.Koomey, Growth in data center electricity use 2005 to 2010. *Oakland CA: Analytics Press*, 2011.

[6] H. Goudarzi, M. Ghasemazar and M. Pedram, "SLA-based Optimization of Power and Migration Cost in Cloud Computing," in *Proc. of the 12th IEEE/ACM International Symposium on Cluster Computing and the Grid*, CCGRID 2012, May 2012. Article (CrossRef Link)

[7] R. Nathuji and K. Schwan, "Virtualpower: Coordinated power management in virtualized enterprise systems", *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, pp. 265–278, 2007.Article (CrossRef Link)

[8] X. Zhu, D. Young, B. J. Watson, Z. Wang, J. Rolia, S. Singhal, B. McKee, C. Hyser et al., "1000 Islands: Integrated capacity and workload management for the next generation data center," in *Proc. of the 5th Intl. Conf. on Autonomic Computing (ICAC'08)*, pp. 172–181, 2008. Article (CrossRef Link)

[9] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Resource pool management: Reactive versus proactive or lets be friends," *Computer Networks*, vol. 53, no. 17, pp. 2905–2922, 2009. Article (CrossRef Link)

[10] VMware Inc., "VMware distributed power management concepts and use," *Information Guide*, 2010.

[11] W. Zheng, R. Bianchini, G. Janakiraman, J. Santos, and Y. Turner, "JustRunIt: Experiment-based management of virtualized datacenters," in *Proc. of the 2009 USENIX Annual Technical Conf*, pp. 18–33,2009. Article (CrossRef Link)

[12] B. Guenter, N. Jain, and C. Williams, "Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning," in *Proc. of the 30st Annual IEEE Intl. Conf. on Computer*

Communications (INFOCOM), pp. 1332–1340, 2011. Article (CrossRef Link)

[13] L.T Kou, G Markowsky, "Multi dimensional Bin Packing algorithms," *IBM J. Research and Development* , vol. 21, pp. 443-448, September, 1977. Article (CrossRef Link)

[14] A. Beloglazov, R. Buyya, "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 (CCPE),* 2012, DOI: 10.1002/cpe.1867, (in press). Article (CrossRef Link)

[15] Tiago C. Ferreto, Marco A.S. Netto, Rodrigo N. Calheiros, Cesar A.F De Rose, "Server Consolidation with migration control for virtualized data centers," *Future Generation Computer System*, pp. 1027-1034, 2011. Article (CrossRef Link)

[16] N. Bobroff, A. Kochut, K.A Beaty, "Dynamic Placement of virtual machines for managing SLA violations," in *Proc of 10^{th} IFIP/IEEE International Symposium on Integrated Network Management IM'07*, pp. 119-128, May 2007. Article (CrossRef Link)

[17] M. Cardosa, M. Korupolu, and A. Singh, "Share and utilities based power consolidation in virtualized server environments," in *Proc of IFIP/IEEE International Symposium on Integrated Network Management IM'09*, 2009. Article (CrossRef Link)

[18] W. Iqbal, M. N. Dailey and D. Carrera, "SLA-Driven Dynamic Resource Management for Multi-tier Web Applications in a Cloud," in *Proc. Of the CCGrid, Melbourne, Australia,* pp.832-837 2010. Article (CrossRef Link)

[19] A. Verma, P. Ahuja, A. Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems," in *Proc of the 9^{th} ACM/IFIP/USENIX International Conference on Middleware, Springer*, pp. 243-264, 2008. Article (CrossRef Link)

[20] M. Stilwell, D. Schanzenbach, F. Vivien, H. Casanova, "Resource allocation algorithms for virtualized service hostings platform," *Journal of Parallel and distributed Computing*, vol.70, no. 9 pp. 962-974, 2010. Article (CrossRef Link)

[21] Jeffrey M. Galloway, Karl L. Smith, Susan S. Vrbsky, "Power aware load balancing for Cloud Computing," in *Proc of the World Congress on Engineering and Computer Science 2011 Vol I* October 19-21, WCECS 2011.

[22] Kuo-Qin Yan ; Wen-Pin Liao ; Shun-Sheng Wang , "Towards a Load Balancing in a three-level cloud computing network," in *Proc of 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), 2010*, Vol-1, pp.-108-113, 2010. Article (CrossRef Link)

[23] T. Wood, P. Shenoy, A. Venkataramani, M. Yousif: "Sandpiper: Black-box and gray-box resource management for virtual machines," *Computer Networks* 53(17), 2923-2938 (2009). Article (CrossRef Link)

[24] A. Beloglazov, J. Abwajy, R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", *Future Generation Computer Systems* 28 (5) (2012) 755–768. Article (CrossRef Link)

[25] D.M. Erceg-Hurn,VM Mirosevich, "Modern Robust Statistical Method: An easy way to maximize the accuracy and power of your research," *American Psychologist*, 63, 591-601, 2008. Article (CrossRef Link)

[26] Peter J. Rousseeuw, Christophe Croux, "Alternative to the median absolute deviation," *Journal of American statistical Association*, vol: 88, pp.1273-1283, 1993. Article (CrossRef Link)

[27] Yue M. "A simple proof of the inequality FFD (L) < 11/9 OPT (L) + 1, for all 1 for the FFD bin packing algorithm," *Acta Mathematicae Applicatae Sinica (English Series)* 1991; 7(4): 321-331. Article (CrossRef Link)

[28] RN Calheiros, R. Ranjan, A. Beloglazov, CAFD Rose, R. Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, 2011; 41(1): 23-50. Article (CrossRef Link)

[29] KS Park, VS Pai, "CoMon: a mostly scalable monitoring system for PlanetLab," in *Proc. of ACM SIGOPS Operating Systems Review* 2006; 40(1):74. Article (CrossRef Link)

[30] Amazon EC2 Instance, http://aws.amazon.com/ec2/instance-types/

[31] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, X. Zhu, No ''power struggles: coordinated multi-level power management for the data center," *SIGARCH Computer Architecture News*, 36 (1) (2008) 48–59. Article (CrossRef Link)

[32] D. Kusic, J.O. Kephart, J.E. Hanson, N. Kandasamy, G. Jiang, "Power and performancee management of virtualized computing environments via lookahead control," *Cluster Computing*, 12 (1) (2009) 1–15. Article (CrossRef Link)

Md. Sabbir Hasan has earned his B.Sc degree in the field of Electrical and Electronic Engineering from Islamic University of Technology in Bangladesh in 2009. He was working as OMCB Junior Engineer at Orascom Telecom Bangladesh Limited, Banglalink, Bangladesh from 2010 to 2011. Currently he has finished his Master's in Computer Engineering from Kyung Hee University. He is a member of Korea Information Process Society. His interested research areas are: Cloud Computing, Service Oriented Computing, Distributed System and Energy aware Computing.

Eui-Nam Huh has earned BS degree from Busan National University in Korea, Master's degree in Computer Science from University of Texas, USA in 1995 and Ph.D degree from the Ohio University, USA in 2002. He was a director of Computer Information Center and Assistant Professor in Sahmyook University, South Korea during the academic year 2001 and 2002. He has also served for the WPDRTS/IPDPS community as program chair in 2003. He has been an editor of Journal of Korean Society for Internet Information and Korea Grid Standard group chair since 2002. He was also an Assistant Professor in Seoul Women's University, South Korea. Now he is with Kyung Hee University, South Korea as Professor in Dept. of Computer Engineering. His interested research areas are: High Performance Network, Sensor Network, Distributed Real Time System, Grid, Cloud Computing, and Network security.