Journal of King Saud University – Computer and Information Sciences 32 (2020) 700–708

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Energy efficient strategy for placement of virtual machines selected from underloaded servers in compute Cloud

Nimisha Patel [a,b,*], Hiren Patel [c]

[a] Rai University, Ahmedabad, India
[b] Sankalchand Patel College of Engineering, Visnagar, Gujarat, India
[c] LDRP Institute of Technology & Research, Gandhinagar, Gujarat, India

ARTICLE INFO

Article history:
Received 9 July 2017
Revised 28 October 2017
Accepted 19 November 2017
Available online 21 November 2017

Keywords:
Workload consolidation
VM placement
Utilization
Energy efficiency
Underloaded server

ABSTRACT

Workload consolidation is a phase in Cloud datacenter where tasks are allocated among the available hosts in such a way that a minimal number of hosts is used and users' need in terms of service level agreement (SLA) is fulfilled. To achieve workload consolidation, hosts are divided among three groups based on their utilization namely overloaded hosts, underloaded host and normal hosts. Detection of over or underloaded host is a challenging issue. Most of the existing researchers propose to use threshold values for such detection. We believe that there is a scope of improvement in existing methods of deciding underloaded hosts and subsequently taking off virtual machines (VMs) from them and placing them on other hosts. In this research, we propose Host Utilization Aware (HUA) Algorithm for underloaded host detection and placing its VMs on other hosts in a dynamic Cloud environment. We compare our proposed mechanism with existing one and with empirical analysis; it is shown that our proposal results into shutting off more number of hosts without compromising user's workload requirement which leads to an energy-efficient workload consolidation with minimal migration costs and efficient utilization of active hosts.

1. Introduction

Computing resources have become more powerful, cheaper, and ubiquitously available than ever before with the rapid development of computing and storage technologies and the extreme success of the Internet. This technological shift has become the reason for the realization of a new computing paradigm called Cloud Computing. The Cloud computing model (Buyya et al., 2010) has quickly exerted a pull on much of the user's focus in recent years. Mell and Grace (2011) define Cloud as a "model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." NIST further lists five essential characteristics of Cloud computing namely (i) on-demand self-service, (ii) broad network access, (iii) resource pooling, (iv) rapid elasticity or expansion, and (v) measured service. In addition to these five essential characteristics, the cloud community has extensively used the service models to categorize the cloud services namely (a) Software as a Service (SaaS), (b) Platform as a Service (PaaS) and (c) Infrastructure as a Service (IaaS), and four deployment models such as (1) private, (2) community, (3) public and (4) hybrid. These Cloud services and deployment models assist programmers having groundbreaking thoughts but lack huge capital investment in computing infrastructures to deploy their products in the real market. Cloud work on the top virtualization technology by (Fox et al., 2009). Virtualization creates virtual resources on the top of physical machines. These virtual resources may include computing resource, operating platforms, storage devices, main memory, internet bandwidth etc. Virtual machine (VM) is an emulated machine which provides the utility of offering resources in form of platform, storage, compute or network. When a task is submitted to Cloud for computation or for another purpose, the same is served through one or more virtual machines created at Cloud

* Corresponding author at: Sankalchand Patel College of Engineering, Visnagar, Gujarat, India.

E-mail addresses: nimishaa_25@yahoo.co.in (N. Patel), hbpatel1976@gmail.com (H. Patel).

Peer review under responsibility of King Saud University.

https://doi.org/10.1016/j.jksuci.2017.11.003
1319-1578/© 2017 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University.

Service Providers' (CSP) premise. Hence, any job submitted to Cloud is run under one or more VMs. Multiple logical VMs run under a common server generally known as the host. There are many hosts under a datacenter and one CSP may have several data centers.

To respond to the rapid growth of customer demands for processing power and storage, cloud providers like Amazon, Microsoft, and Google are deploying a large number of data centers across the world. The Cloud datacenters usually comprise of a great number of well-configured and interconnected computing resources (Luo et al., 2014) which consume a significant amount of electricity for their functioning. Increase usage of Cloud computing has lead to augmentation in electrical energy consumption by the huge amount of servers in a large number of data centers. The survey shows that the average energy consumption of a data center is comparable to that consumed by 25,000 domestic usages (Kaplan et al., 2008). This has attracted consideration of research community in recent years. Reduction of energy consumption can be achieved by switching idle physical servers to lower power states (suspended or turned off) while still preserving customers performance requirements. Out of many different mechanisms to address the issue, workload/server consolidation and task scheduling have been recognized as a few of the popular techniques. Server consolidation works on the principle of minimizing active servers in a data center without compromising the performance of tasks and user requirement. Sleep/Wakeup has been identified as one of the top classifications by (Brienza et al., 2016) in which some of the servers are switched off when not in use to save energy and are awakened whenever necessary. It has been seen that even idle servers consume about 70% of peak power (Fan et al., 2007). In a nutshell, proper distribution of existing tasks among available servers may result into minimizing the active servers without compromising SLA with Cloud users.

Hosts running in a data center are classified into three categories based on their usage namely (i) overloaded hosts (ii) underloaded hosts and (iii) normal hosts. This classification is based on host's utilization, for instance, hosts with utilization more than a certain value (commonly known as upper threshold) may be considered as overloaded hosts and similarly, hosts with utilization less than a certain value (commonly known as lower threshold) may be considered as underloaded hosts. All other hosts except these two categories are considered as normal hosts. According to (Barroso and Holzle, 2007), under the normal scenario, hosts in a datacenter operate only at 10%–50% of their peak capacity and these underloaded hosts become a reason for the waste of electricity. Hence, it is required to reduce the energy consumption by enhancing hosts' resource utilization in Cloud data centers through workload consolidation. Fig. 1 shows an example to understand workload consolidation.

Fig. 1 (Left) shows the status of workload before consolidation. It is seen that all eight hosts are utilized with utilization varying from 20% to 50%. Using the workload consolidation, we may shift few workloads from one host to another in such a way that the target host does not get overloaded. Fig. 1 (Right) shows the status of the workload after consolidation. It depicts that we could turn four hosts (3, 5, 6 and 7) into power saving mode leaving four hosts (1, 2, 4 and 8) active with utilization ranging from 55% to 70%.

Overall, the process of workload consolidation includes (i) selecting few VMs from overloaded hosts and trying to put them on other hosts such that this source host becomes normal and the target hosts do not get overloaded and (ii) selecting all VMs from underloaded hosts and trying to put them on other hosts such that target hosts do not get overloaded, and switching-off these underloaded hosts if all VMs are successfully migrated for the purpose of saving energy. In this research, we aim to address the second category of underloaded hosts' selection and vacating the VMs from them. We have carried out an exhaustive literature survey of the techniques involving detection of underloaded hosts and subsequently vacating them. We have identified that there is a scope of improvement in existing methods of deciding underloaded hosts and subsequent VMs placements on other hosts. In our work, we propose Host Utilization Aware (HUA) algorithm for underloaded host detection and placing its VMs on other hosts in a dynamic Cloud environment. Unlike most of the existing mechanism, HUA computes lower threshold which considers overall utilization of datacenter while taking into account, utilization of all the active hosts in the data center. HUA further predicts maximum number of hosts which can be vacated based on total data center workload. We compare HUA with the one proposed by (Beloglazov and Buyya, 2012) and through experimental analysis; we show that HUA results into shutting off more number of hosts without compromising user's workload requirement leading to an energy-efficient workload consolidation with minimal migration costs and efficient utilization of active hosts.

The overall organization of this article is as follows. Section 2 summarizes the outcome of our literature survey. In Section 3, we propose our method in form of algorithm. Section 4 depicts experimentation setup, data sets, results generated and subsequent discussions. In Section 5, we conclude our research followed by a list of references.

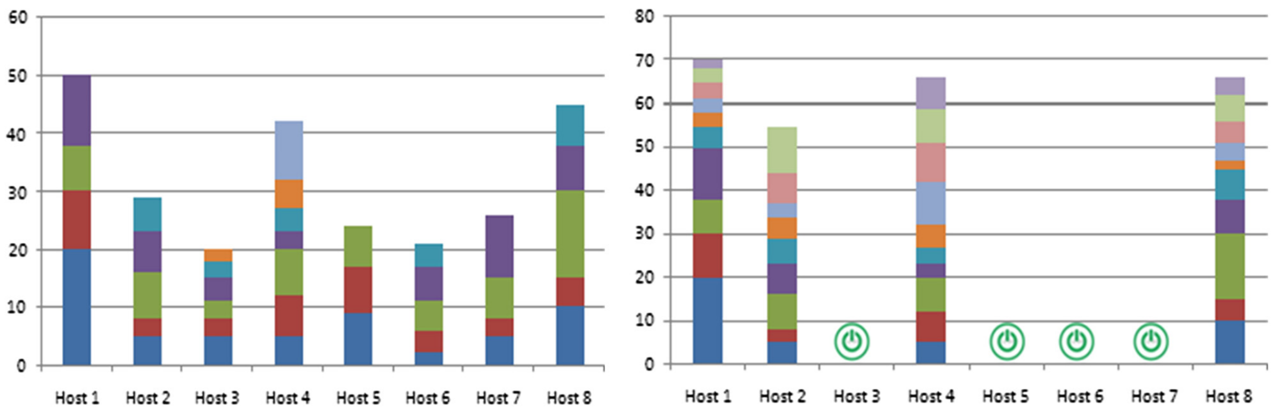

Fig. 1. Workload consolidation. Before consolidation (left) and after consolidation (right).

2. Related work

Robert et al. (2012) illustrated various power consumption estimation models in servers, storage devices and network equipment. Authors propose a three-step model that consisted of optimization, reconfiguration and monitoring to save power. The authors claimed that about 20% energy consumption could be saved if the energy optimization policy could be guided by power consumption prediction models. Nathuji and Schwan (2007) discussed the issue of power management in virtualized data centers during the early phase of Cloud emergence. Authors investigated a mechanism to unite power management mechanisms and policies with the virtualization technologies to actively deploy in data centers. In the mechanism, the resource manager is consisting of two components such as local and global managers. The local manager is considered to be the guest OS's power management strategies whereas the global manager gathers information from local managers for VM placement. But, the paper does not illustrate specific resource management policy for the global manager. Verma et al. (2008) observe the challenge of power-aware dynamic placement of applications as a bin packing problem. Bins are considered to be variable in sizes and costs. Live migration is used for VM migration from one host to another at regular scheduling interval. But, the authors do not talk about the SLA. Song et al. (2014) make use of the virtualization for dynamic resource allocation according to the workload's requirements and optimize the number of active hosts to achieve energy efficiency in the Cloud data center. A model has been proposed for resource allocation based on relaxed on-line bin packing problem and authors propose the variable item size bin packing (VISBP) algorithm. VISBP can be operable of reasonable size variation, as long as the classification (of VMs and PMs) rules are kept. Authors claim better performance in hot spots' migration and load balance when compared to the existing algorithm. However, all PMs are considered homogeneous with unit capacity may be the cardinal limitation of its application. Huang et al. (2013) devise a sub-optimal dynamic SLA-aware resource allocation strategy to achieve energy efficiency in Cloud computing. During the first phase, authors propose a prediction mechanism using support vector regressions (SVR) to estimate resource utilization considering SLA requirement. Later, using a genetic algorithm, resource reallocation mechanism is applied to determine user's VM requirement. Authors claim to maintain SLA by satisfying QoS and improve Cloud provider's profit. However, the GA algorithm does not converge to the local optimal solution with significant execution time. Beloglazov and Buyya (2010a) propose an idea of setting up upper and lower utilization thresholds to identify over and underloaded servers. If the host utilization exceeds the upper threshold, authors propose migrating some VMs from this host to reduce SLA violation (SLAV). In contrast, if the utilization falls below the lower threshold, all VMs on this underloaded host is to be migrated and the host is to be switched off to save energy consumption. However, no specific technique or method has been proposed for calculating upper and lower threshold. Beloglazov and Buyya (2010b) propose a technique based on statistical analysis of the VM utilization history for auto-adjustment of the utilization thresholds. Authors propose equation to calculate lower threshold while taking into consideration, factors such as host utilization, number of hosts, the standard deviation of utilization and probability intervals. However, the same authors, (Beloglazov and Buyya, 2012) propose a simple approach of comparing relative host utilization for detection of the underloaded host. This approach searches for the host with minimum utilization in a data center and tries to place all the VMs from this host to other host keeping them not overloaded. If this is successfully accomplished, the original host is switched to power saving mode and the process is iteratively repeated till there is no further possible VM placement. Extending the work of these researchers, (Horri et al., 2014) proposes a technique to compute host utilization while considering two factors namely (i) CPU utilization of the host and (ii) number of VMs on it for the detection of the underloaded host. Further, authors propose to assign dynamic weight to both of these factors. VM-based dynamic threshold (VDT) is applied periodically. Authors claim a significant reduction in a number of VM migration and hence, reduction in SLA violation and energy consumption. Lin et al. (2011) identify that peak-load-based static resource allocation schemes often result in the underutilization of computing resources. To address the issue, authors propose a technique to compute lower threshold by considering (i) maximum workload and normal workload of a virtual machine (ii) rate of maximum workload and normal workload (iii) current number of virtual machines and (iv) threshold rate ranging between 0 to 1. In this simple to implement scheme, authors claim to improve resource utilization and reduce the user usage cost. Yang et al. (2014) propose a scheme based on load ratios to determine the number of physical machines to be run or turned off. Authors calculate gross occupied resource weight ratio (that is the ratio of workload to physical capacity available). This ratio is compared with two parameters namely (i) maximum tolerant occupied resource weight ratio and (ii) minimum critical occupied resource weight ratio. If gross occupied resource weight ratio is greater than (i) and a number of running physical machine is less than the total number of the physical machine then wake up a standby physical machine to join the other running physical machines. If gross occupied resource weight ratio is less than (ii) and number of running physical machines is greater than 1 then select one of the running physical machines with least load as the machine to be migrated and move all VMs from this machine to other running physical machines and shut it off.

3. Host Utilization Aware (HUA) Algorithm – our proposal

Most of the existing researchers as mentioned in Section 2 carry out a common practice for identifying underloaded hosts by selecting the host with lowest utilization. Further, they try migrating all VMs from the host and continue the process for next host if all VMs can be migrated. This practice includes sorting all the hosts in descending order based on their utilization. Consequently, host with the least utilization is considered as the underloaded host and proceed for taking off VMs from that host to other hosts in such a way that the target hosts do not get overloaded. The process is repeated till we find placements for the all the VMs of underloaded hosts. The major drawback of this existing mechanism is that it does not take the total utilization (by all the hosts) of the data center into consideration while (i) detecting underloaded hosts and (ii) placing all the VMs from selected underloaded hosts on other hosts. We further believe that the knowledge of all host utilizations (i.e. overall workload in the data center) would help us in proper selection of the host(s) to be vacated and subsequent VM placement. Moreover, existing mechanisms incorporate a simple placement process where the selected VMs from least utilized host are placed on other hosts without considering the target host's utilization. It may result in placing a VM on a target host which was prospective to be vacated in near future which may lead to more number of active hosts per data center and more number of VM migrations in future.

Hence to overcome these challenges, we propose a novel mechanism to compute lower threshold which considers overall utilization of datacenter while taking into account, utilization of all the active hosts in the data center. We further predict a maximum number of hosts which can be vacated based on the total workload of data center. The value of the predicted maximum number of hosts which can be vacated is used to fetch the value of the lower threshold. Subsequently, we divide all the available hosts into two categories namely (i) hosts with utilization below the lower threshold and (ii) host with utilization equal or higher than the lower threshold. Consequently, we select VMs from host list of (i) and try to place them on the host list of (ii) with minimum change in the power consumption. Algorithm 1 depicts the steps for the same.

```
Algorithm 1: Host Utilization Aware (HUA) Algorithm for
  Underloaded Host Detection
1. Input: hostList, vmList, Output: vmAllocation
2. Add all overUtilizedHosts and switchedOffHost in exclude
  HostListforFindingUnderUtilized
3. Add all overUtilizedHosts and switchedOffHost in exclude
  HostListforFindingNewPlacement
4. hostList.sortIncreasingUtilization()
5. totalWorkLoadDatacenter $W_{TOTAL} \leftarrow \sum_{i=1}^{i=H_{TOTAL}} Ui$ where $U_i$ is
  Utilization of ith Host
6. Maximum number of hosts that can be evacuated $H_{MAX} \leftarrow$
  $H_{TOTAL} - \frac{W_{TOTAL}}{\text{Upper Threshold}}$
7. Lower Threshold $T_{LOWER} \leftarrow H_{SORT}[H_{MAX}]$
8. for each hosts with index in 1 to $H_{MAX}$ do
9.   excludeHostListforFindingNewPlacement.add($H_i$)
10. end for
11. hostNumberMinUtilization ← 1
12. while TRUE
13.   if $H_{TOTAL}$ = excludeHostListforFindingUnderUtilized.size()
14.     break
15.   endif
16.   if hostNumberMinUtilization does not belong into ex
  cludeHostListforFindingUnderUtilized then
17.     excludeHostListforFindingUnderUtilized.add(hostNum
  berMinUtilization)
18.   endif
19.   if hostNumberMinUtilization does not belong into ex
  cludeHostListforFindingNewPlacement then
20.     excludeHostListforFindingNewPlacement.add
  (hostNumberMinUtilization)
21.   endif
22.   for all vm in Host hostNumberMinUtilization do
23.     vmUtil ← getVmUtilization
  (vm, hostNumberMinUtilization)
24.     minPowerDiff ← MAX
25.     allocatedHost ← NULL
26.     for each host in hostList do
27.       if host belongs to excludeHostListforFindingNewPla
  cement then
28.         continue with next host
29.       endif
30.       if host has sufficient resources for vm then
31.         If $U_{host}$ (after placement) > UpperThreshold then
32.           continue with next host
33.         endif
34.         if $P_{host}$ (after placement) - $P_{host}$ (before placement)
   < minPowerDiff then
35.           minPowerDiff ← $P_{host}$ (after placement) - $P_{host}$
  (before placement)
36.           allocatedHost ← host
37.         endif
38.       endif
39.     end for
40.     if allocatedHost ≠ NULL then
41.       if allocatedHost does not belong to excludeHos
  tListforFindingUnderUtilized then
42.         excludeHostListforFindingUnderUtilized.add
  (allocatedHost)
43.       endif
44.       migrationMap.add(vm, allocatedHost)
45.     else
46.       break // Not all VMs can be reallocated from the
  host hence further reallocation stands cancelled
47.     endif
48.   end for
49.   hostNumberMinUtilization ←
  hostNumberMinUtilization + 1
50. end while
```

In, Host Utilization Aware (HUA) Algorithm, first we add all overutilized hosts and switched-off hosts in the lists namely *excludeHostListforFindingUnderUtilized* and *excludeHostListforFindingNewPlacement* to skip them from subsequent searching leading to efficient computation cost. Then, we calculate a maximum number of hosts to be vacated (H_{MAX}) based on the total workload of datacenter which can be calculated from the utilization of all active hosts. Subsequently, with the help of H_{MAX}, we come up with a lower threshold value (T_{LOWER}). We use this value to search an underutilized host and we can exclude the host having its utilization lower than T_{LOWER} for VM Placement too. Next, we select the lowest utilized host and try to migrate all its VM to another host. During placement of VM, we make sure that whether the target host has enough resources to accommodate the new VM or not and the target host does not get overloaded after placing the new VM(s). Upon satisfying above conditions, we select those hosts for new placement which results into minimum increase in power consumption after placement. Further, we verify whether all the VMs of underloaded host get a suitable host for placement or not. If yes, then we migrate all VMs and turn off the host. The process is repeated for next underutilized host till further placement possible.

4. Performance assessment

4.1. Experimentation testbed

To evaluate our proposed mechanism, it is required to be tested on large scale datacenter implementation of Cloud under the dynamic workload. But, as it is intricate to accomplish such requirement on an actual Cloud infrastructure, the CloudSim toolkit (Calheiros et al., 2014) has been selected as a simulation environment. CloudSim has been one of the efficient simulation environments for the factors such as virtualized resource management and modeling, energy consumption and account modeling, workload dynamism, VM migration and SLA computations (Patel and Patel, 2016).

For our experimentations, the testbed configurations are mentioned in Table 1 for 800 heterogeneous hosts half of which are HP ProLiant ML110 G4 servers (Type 1), and the other half consists of HP ProLiant ML110 G5 servers (Type 2). The frequency of the servers' CPUs is mapped onto MIPS ratings: 1860 MIPS each core of the HP ProLiant ML110 G5 server, and 2660 MIPS each core of the HP ProLiant ML110 G5 server. Each server is modeled to have 1 GB/s network bandwidth. Table 2 describes VM configurations.

Table 1
Specifications of Hosts.

	Name	MIPS	RAM (MB)	Bandwidth (Gb per sec)	Core/Processing Elements	Number of Hosts
Type 1	HpProLiantMl110G4Xeon3040	1860	4096	1	2	400
Type 2	HpProLiantMl110G5Xeon3075	2660	4096	1	2	400

Table 2
Specifications of Virtual Machines.

	MIPS	Core/Processing Elements	RAM (MB)	Bandwidth (Mb per sec)	Number of VM (Total:1052)
Type 1	2500	1	870	100	263
Type 2	2000	1	1740	100	263
Type 3	1000	1	1740	100	263
Type 4	500	1	613	100	263

Table 3
Power Consumption (Watts) at different load levels.

Host	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HpProLiantMl110G4Xeon3040	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HpProLiantMl110G5Xeon3075	93.7	97	101	105	110	116	121	125	129	133	135

All the VMs are single-core, each VM type: 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). Initially, the VMs are allocated according to the resource requirements defined by the VM types.

For power consumption by a host, we make use of real data of power consumption provided by SPECpower benchmark (SPECpower benchmark, 2008) which has been listed in Table 3. The power consumption by servers can be accurately described by a linear relationship between the power consumption and CPU utilization. As can be seen from the table that even at low utilization, the host consumes a significant amount of power. Hence it is required to turn off such kind of hosts, when not in use.

Workload traces are taken from real dataset provided as part of the CoMon project, a scalable monitoring infrastructure for PlanetLab (Park and Pai, 2006). The data collected are from slice-centric daemon showing resource consumption per slice. The type of data collected is context number (numeric userid for the slice), transmit and receive rates for the past 1 and 15 min, number of processes, physical and virtual memory consumption, overall CPU and memory utilization, and slice name. Most nodes have on the order of 50 slices running at a time. In our research, we mostly use the overall CPU utilization which taken at the interval of 5 min for more than thousand VMs from servers located across more than 500 places around the globe. For our experimentation, we have chosen the dataset taken on March 3rd 2011 comprising of 1052 VMs. These 1052 VMs are of type 1, type 2, type 3 and type 4 as mentioned in Table 2.

4.2. Simulation results

Table 4 depicts the simulation results of our proposal. Our proposal (HUA) is compared with existing approach mentioned in Beloglazov and Buyya (2012).

With the intention of comparing the efficiency of HUA algorithm with that of existing one, we use several metrics to assess the performance namely energy consumption, number of VM migration, Service Level Agreement Violation (SLAV), SLA performance degradation due to migration (PDM), SLA time per active host (SLATAH), overall SLA violation, average SLA violation and number of host shutdowns. Energy consumption by the hosts of a data center is one of the performance metrics. Energy consumption is calculated according to the power consumption of host as mentioned in Table 3. Another metric is the number of VM migrations initiated during VM placement after selection of VM from the underutilized host. An SLA violation caused during workload consolidation phase is described in terms of SLAV which is calculated based on SLATAH and PDM. SLATAH is defined as the percentage of time, during which active hosts have experienced the CPU utilization of 100%. The overall performance degradation due to VMs migrations is measured in terms of Performance Degradation due to Migrations (PDM).

We have used various combinations of VM selections policies and overloaded host detection methods. VM selections policies include maximum utilization (MU), maximum correlation (MC), minimum migration time (MMT) and random selection (RS). Overloaded host detection techniques comprise median absolute deviation (MAD), interquartile range (IQR), local regression (LR), robust local regression (LRR) and Static Threshold (ST). The value of safety parameter (s) is adjusted to control energy consumption and SLA violation. The system consolidates VMs based on the value of s. The low value of s results into less energy consumption but the higher level of SLA violations caused by the consolidation and vice versa. Hence, it is required to address the tradeoff between energy consumption and SLA violation.

We have simulated various combinations of the host overloading detection techniques (MAD, IQR, LR, LRR and ST) and VM selection techniques (MU, MC, MMT and RS) for existing and proposed an approach of underloaded host detection and subsequent placement. Following Figs. 2–7 depict the assessment of metrics mentioned as above. Fig. 2 depicts the status of the number of host shutdown after VM placement from underloaded hosts. It can be seen that the proposed method outperforms to increase the number of host shutdown (around 67%). This is due to the fact that unlike existing method which only considers the factor of power consumption, the proposed method also considers the exclusion of hosts which are estimated to be vacated in near future during VM placement. This results into better workload balancing which evacuates more number of hosts. Proposed method tries to make best possible utilization of a minimum number of the host while attempting to allocate a maximum number of jobs to the active hosts. As energy consumption is directly proportional to the number of active host in a datacenter, increase in the number of host shutdown would result into reduction in energy consumption. Fig. 3 illustrates the status of energy efficiency, which has been improved by average 12.59%. To achieve this improvement in

Table 4
Simulation Results.

Policy with Safety Parameter		Energy Consumption (kWh)	Number of VM Migration	SLAV (%)	SLA PDM (%)	SLATAH (%)	Overall SLA Violation (%)	Average SLA Violation (%)	Number of host shutdowns
MAD-MC-2.5	Existing	12.21	2062	0.0107	0.14	7.48	0.14	10	772
	HUA	10.67	2900	0.00612	0.12	5.05	0.13	10.42	1285
MAD-MU-2.5	Existing	12.41	2121	0.01084	0.14	7.58	0.14	10	787
	HUA	13.56	4122	0.01117	0.13	8.76	0.16	10.7	1541
MAD-MMT-2.5	Existing	12.28	2069	0.01038	0.14	7.41	0.14	10	769
	HUA	10.58	2868	0.00602	0.12	5.04	0.13	10.31	1270
MAD-RS-2.5	Existing	12.13	2012	0.01066	0.14	7.47	0.14	10	763
	HUA	9.56	2340	0.01012	0.15	6.78	0.16	10.5	1124
IQR-MC-1.5	Existing	12.23	2075	0.01074	0.14	7.51	0.14	10	770
	HUA	10.76	2961	0.00619	0.12	5.15	0.13	10.41	1288
IQR-MU-1.5	Existing	12.43	2133	0.01092	0.14	7.59	0.14	10	789
	HUA	13.66	4213	0.01159	0.13	9.04	0.16	10.7	1547
IQR-MMT-1.5	Existing	12.27	2064	0.01031	0.14	7.39	0.14	10	769
	HUA	10.65	2897	0.00626	0.12	5.1	0.13	10.42	1276
IQR-RS-1.5	Existing	12.13	2033	0.01091	0.14	7.58	0.14	10	763
	HUA	9.13	2322	0.01095	0.16	7.03	0.17	10.86	1081
LR-MC-1.2	Existing	12.2	2066	0.0107	0.14	7.49	0.14	10	771
	HUA	10.65	2855	0.00589	0.12	5.02	0.13	10.48	1282
LR-MU-1.2	Existing	12.37	2097	0.01071	0.14	7.53	0.14	10	786
	HUA	13.45	3984	0.01017	0.12	8.39	0.16	10.81	1532
LR-MMT-1.2	Existing	12.28	2063	0.01026	0.14	7.39	0.14	10	769
	HUA	10.53	2795	0.00567	0.12	4.93	0.12	10.31	1264
LR-RS-1.2	Existing	12.07	2028	0.01062	0.14	7.48	0.14	10	763
	HUA	9.16	2157	0.00907	0.14	6.54	0.15	10.35	1079
LRR-MC-1.2	Existing	12.2	2066	0.0107	0.14	7.49	0.14	10	771
	HUA	10.65	2855	0.00589	0.12	5.02	0.13	10.48	1282
LRR-MU-1.2	Existing	12.37	2097	0.01071	0.14	7.53	0.14	10	786
	HUA	13.45	3984	0.01017	0.12	8.39	0.16	10.81	1532
LRR-MMT-1.2	Existing	12.28	2063	0.01026	0.14	7.39	0.14	10	769
	HUA	10.53	2795	0.00567	0.12	4.93	0.12	10.31	1264
LRR-RS-1.2	Existing	12.11	2049	0.01089	0.15	7.49	0.15	10	767
	HUA	9.2	2217	0.0087	0.14	6.37	0.14	10.3	1094
ST-MC-0.8	Existing	12.24	2057	0.01051	0.14	7.5	0.14	10.02	769
	HUA	8.91	2811	0.00748	0.13	5.87	0.15	11.19	1212
ST-MU-0.8	Existing	12.42	2134	0.01066	0.14	7.66	0.14	10.05	786
	HUA	12.44	4314	0.01463	0.15	9.93	0.18	11.66	1591
ST-MMT-0.8	Existing	12.1	2056	0.01001	0.13	7.43	0.13	10	765
	HUA	8.86	2788	0.00733	0.13	5.84	0.14	11.16	1207
ST-RS-0.8	Existing	12.03	2045	0.01052	0.14	7.56	0.14	10	759
	HUA	7.83	2263	0.01358	0.16	8.5	0.21	11.03	1073

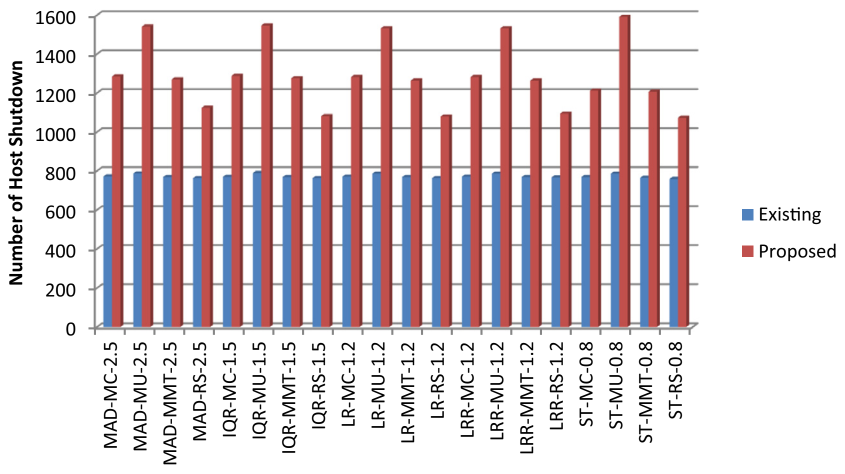

Fig. 2. Number of host shutdown.

energy consumption, it is obvious to shut down more hosts (which was improved by 67% as mentioned in Fig. 2). And to shut down more host, workload needs to be properly consolidated in a minimum number of hosts by the requirement of number of VM migrations. Hence, in our case, VM migration is increased to 45.58%. Fig. 4 depicts the number of VM migrations. However, as shown in Fig. 5, SLA Violation (SLAV) is reduced in our case which has been improved by 18.77%. The metric SLAV is computed from (i) SLA performance degradation due to migration (PDM) (Fig. 6 and (ii) SLA violation Time per Active Host (SLAT AH) (Fig. 7). As shown in Fig. 6, PDM has been improved by 6.40% whereas SLAT AH has been improved by 12.35%. Our experimental results show that

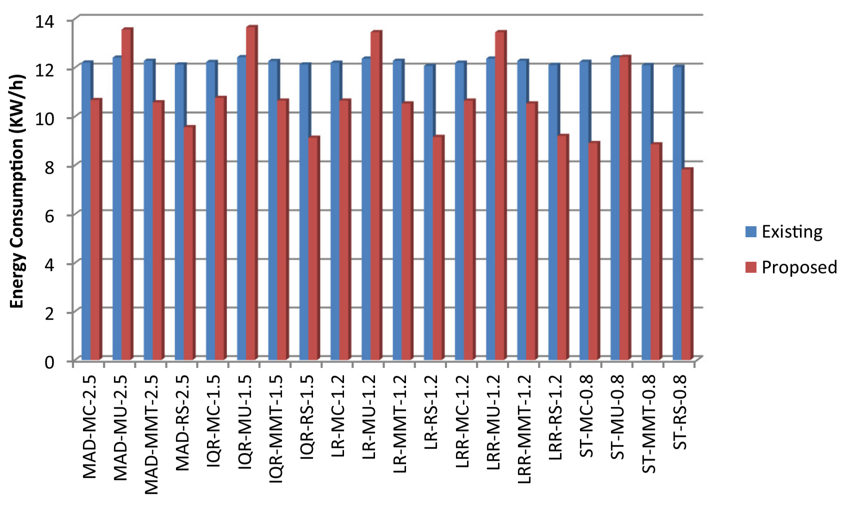

Fig. 3. Energy consumption (kW/h).

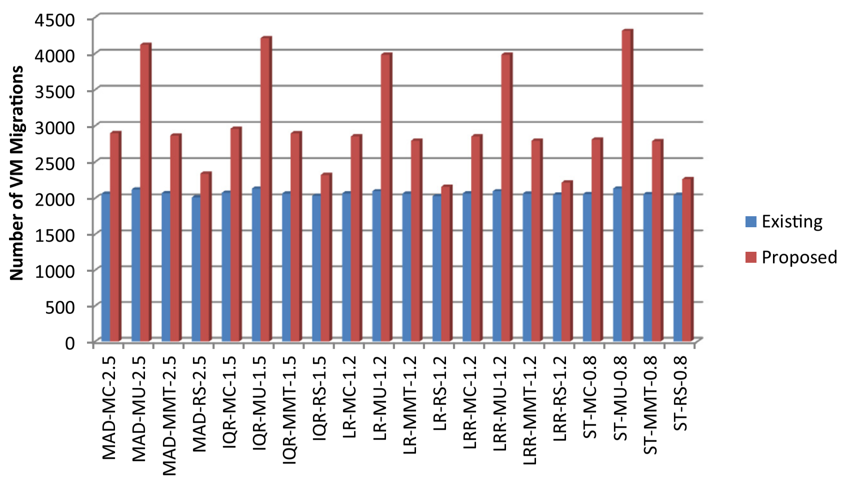

Fig. 4. Number of VM migrations.

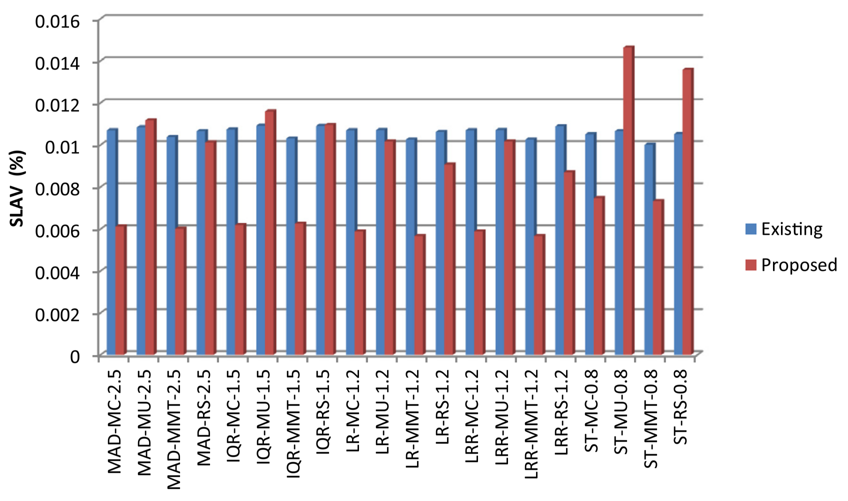

Fig. 5. SLAV.

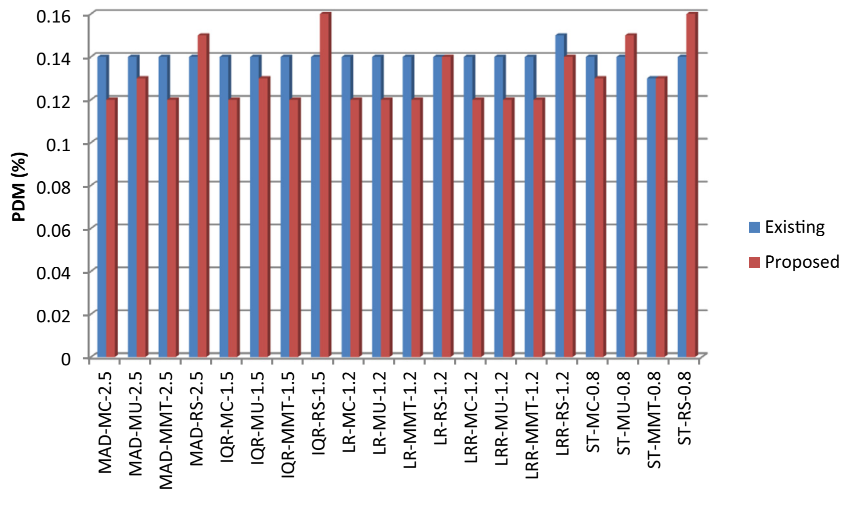

Fig. 6. SLA performance degradation due to migration (PDM).

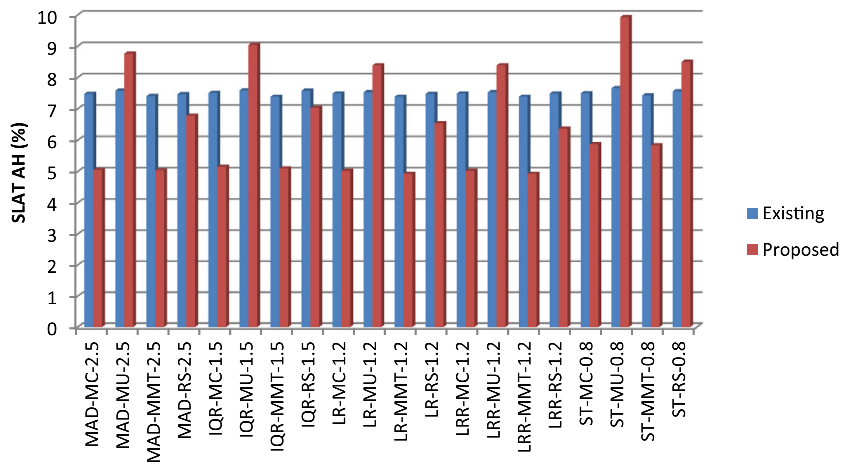

Fig. 7. SLA violation Time per Active Host (SLAT AH).

HUA is efficient in deciding underloaded hosts and subsequently vacating more number of hosts resulting in saving in energy consumption while maintaining SLA violation.

5. Conclusion

Underloaded host detection is one of the important phases in workload consolidation process. In this research, we have surveyed existing methods to calculate lower threshold value which is used for detection of the underloaded host. Existing methods calculate lower threshold value based on sorted utilization of available hosts, however, they do not consider the entire scenario of total workload across the datacenters. To overcome the issue, we have proposed a novel technique, Host Utilization Aware (HUA) Algorithm for underloaded host detection, for predicting a maximum number of hosts which can be vacated by computing lower threshold that considers overall utilization of datacenter. Our experimental results have proved that HUA is efficient in deciding underloaded hosts and subsequently vacating more number of hosts resulting in saving in energy consumption while maintaining SLA violation. In future, the proposed technique can be extended from simulation setup to real-time data center environment with other variable factors.

References

Barroso, L.A., Holzle, U., 2007. The case for energy-proportional computing. Computer 40, 268–280. https://doi.org/10.1109/MC.2007.443.

Beloglazov, A., Buyya, R., 2010a. Energy efficient resource management in virtualized cloud data centers. In: Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. IEEE Computer Society, pp. 826–831.

Beloglazov, A., Buyya, R., 2010. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. In ACM Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science. Vol. 4.

Beloglazov, A., Buyya, R., 2012. 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. 24 (13), 1397–1420.

Brienza, S., Cebeci, S.E., Masoumzadeh, S.S., Hlavacs, H., Özkasap, Ö., Anastasi, G., 2016. A survey on energy efficiency in P2P systems: file distribution, content streaming, and epidemics. ACM Comput. Surveys (CSUR) 48 (3), 36.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I., 2010. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. 25, 599–616. https://doi.org/10.1016/j.future.2008.12.001.

Calheiros, R.N., Ranjan, R., Beloglazov, A., Rose, C.A.F.D., Buyya, R., 2014. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Pract. Exp. 41 (1), 23–50.

Fan, X., Weber, W.D., Barroso, L.A., 2007. Power provisioning for a warehouse-sized computer. In: Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA 2007). ACM New York, NY, USA, pp. 13–23.

Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Stoica, I., 2009. Above the Clouds: A Berkeley View of Cloud Computing. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA.

Horri, A., Mozafari, M.S., Dastghaibyfard, G., 2014. Novel resource allocation algorithms to performance and energy efficiency in cloud computing. J. Supercomput. 69 (3), 1445–1461.

Huang, C.J., Guan, C.T., Chen, H.M., Wang, Y.W., Chang, S.C., Li, C.Y., Weng, C.H., 2013. An adaptive resource management scheme in cloud computing. Eng. Appl. Artif. Intell. 26, 382–389. https://doi.org/10.1016/j.engappai.2012.10.004.

Kaplan, J.M., Forrest, W., Kindler, N., 2008. Revolutionizing Data Center Energy Efficiency. McKinsey & Company, New York, NY, USA. Technical Report.

Lin, W., Wang, J.Z., Liang, C., Qi, D., 2011. A threshold-based dynamic resource allocation scheme for cloud computing. Procedia Eng. 23, 695–703.

Luo, J., Li, X., Chen, M., 2014. Hybrid shuffled frog leaping algorithm for energy-efficient dynamic consolidation of virtual machines in cloud data centers. Expert Syst. Appl. 41, 5804–5816. https://doi.org/10.1016/j.eswa.2014.03.039.

Mell, P., Grace, T., 2011. The NIST definition of cloud computing (draft), NIST Special publication, 800, 145

Nathuji, R., Schwan, K., 2007. Virtual power: coordinated power management in virtualized enterprise systems. ACM SIGOPS Oper. Syst. Rev. ACM. 2007 (41), 265–278. https://doi.org/10.1145/1323293.1294287.

Park, K.S., Pai, V.S., 2006. CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Oper. Syst. Rev. 40, 65–74. https://doi.org/10.1145/1113361.1113374.

Patel, N., Patel, H., 2016. A comprehensive assessment and comparative analysis of simulations tools for cloud computing. Int. J. Eng. Computer Sci. 5 (11).

Robert, B., Hermann De, M., Ricardo, L., Giovanni, G., 2012. Cloud computing and its interest in saving energy: the use case of a private Cloud. J. Cloud Comput. Adv. Syst. Appl. 1, 5.

Song, W., Xiao, Z., Chen, Q., Luo, H., 2014. Adaptive resource provisioning for the Cloud using online bin packing. IEEE Trans. Comput. 63, 2647–2660. https://doi.org/10.1109/TC.2013.148.

The SPECpower benchmark, 2008. http://www.spec.org/power_ssj2008/.

Verma, A., Ahuja, P., Neogi, A., 2008. pMapper: power and migration cost aware application placement in virtualized systems, Proceedings of ACM/IFIP/USENIX 9th International Middleware Conference, Leuven, Belgium. pp. 243–264.

Yang, C.T., Liu, J.C., Huang, K.L., Jiang, F.C., 2014. A method for managing green power of a virtual machine cluster in cloud. Future Gener. Comput. Syst. 37, 26–36.