The Scientific Temper (2024) Vol. 15 (3): 2515-2522
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.3.17
https://scientifictemper.com/

RESEARCH ARTICLE

A PPR-based energy-efficient VM consolidation in cloud computing

Rahat Yezdani*, S. M. K. Quadri

Abstract

The tendency to do more jobs while consuming less energy is crucial to energy efficiency in the cloud environment. To use less energy while performing more tasks at the best throughput, this study provides an energy-efficient technique (PPR_DWMMT_1.1) for VM consolidation in a cloud domain. Our approach uses the PPR to determine the upper threshold for overload detection and the lower threshold for underload detection. Additionally, PPR_DWMMT_1.1 considers the overall workload utilization of the data center when selecting a lower threshold, which could reduce VM migrations. Our proposed method, PPR DWMMT 1.1, is compared to the simulation results of the four reference techniques, IQR_MMT_1.5, LR_MC_1.2, MAD_MU_2.5, and THR_RS_0.8. Our solution has been demonstrated to use less energy, trigger fewer host shutdowns and live migrations, and achieve the best performance when compared to the other four approaches.

Keywords: Cloud environment, Energy consumption, Energy-efficient approach, VM consolidation, VM migration.

Introduction

Cloud computing is a paradigm of computing that provides hosted services over the Internet *via* a browser or web-based application, such as social media sites and apps (Instagram, Facebook, YouTube, WhatsApp, and so on), email (Gmail, Proton Mail, Zoho Mail, Outlook, Yahoo! Mail, iCloud Mail, AOL Mail, GMX, and so on), banking and financial services, and many more (Buyya *et al*., 2008). Cloud computing provides several different services, referred to as service layers, such as infrastructure as a service, platform as a service, and software as a service. All of these services are available on a pay-as-you-go basis with no geographic limitations. Data centers are similar to farmsteads with multiple servers and provide services such as storage, data management, networking, application usage, various OS usage, and recovery and backup to consumers. At the infrastructure level, cooling systems consume a lot of energy to cool data centers, which generate a lot of heat and consume an immense amount of energy when the system is idle (el Kafhali & Salah, 2018).

This results in huge financial losses incurred by both users and service providers. A typical data center may use up to 25,000 kWh per day. Data centers consume roughly 3% of worldwide electricity and about 26 nuclear power plants, according to reports (Asad & Rehman Chaudhry, 2017). As a direct consequence, the exorbitant power consumption of virtualized data centers causes system instability, CO_2 emissions, energy waste, and a poor return on investment (Karuppasamy & Balakannan, 2018). So, in the current context, one of the primary challenges for data centers is the development of power management techniques to address the issue of energy consumption and CO_2 emissions in cloud data centers (Naidu & Chadha, 2020).

One approach to lowering energy consumption is virtual machine (VM) consolidation, in which VMs are replaced regularly to reduce the number of active servers. In VM consolidation, live migration to maximize resource utilization also maintains service-level agreements (SLAs) and application performance. The VM consolidation approach is divided into several steps (Moghaddam *et al*., 2019). The first step is the identification of over-utilized servers that may violate service-level agreements and under-utilized servers that must be shut down to lower the number

Department of Computer Science, Jamia Millia Islamia, New Delhi, India

***Corresponding Author:** Rahat Yezdani, Department of Computer Science, Jamia Millia Islamia, New Delhi, India, E-Mail: rahatyajdani123@gmail.com

How to cite this article: Yezdani, R, Quadri, S. M. K. (2024). A PPR-based energy-efficient VM consolidation in cloud computing. The Scientific Temper, **15**(3):2515-2522.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.3.17

Source of support: Nil

Conflict of interest: None.

Received: 10/07/2024 **Accepted:** 20/08/2024 **Published** : 10/09/2024

of active hosts. Then, a selection of VM aims to discover the best VMs to migrate from over-utilized servers, and finally, a VM placement method that selects hosting servers for the appropriately selected VMs. Once underutilized and overutilized servers have been identified, VM consolidation strategies must attempt to relocate some of their hosted VMs to meet SLAs and minimize overall data center energy usage (Yadav *et al*., 2018).

Best-known VM consolidation approaches that address energy usage use static or dynamic lower threshold levels based on existing server utilization to identify underutilized servers rather than taking into account the whole data center workload. Unfortunately, some of these solutions have drawbacks, such as deploying a VM on a target node that is about to be evacuated, which increases the number of active hosts per data center and leads to more VM migrations. The proposed strategy for detecting underutilized servers takes into account the entire data center workload. However, data center hosts are heterogeneous, containing both low- and high-performance computers. Low-performance servers emit more heat than high-performance servers despite carrying more workloads and using less electricity (Farahnakian, Hanini *et al*., 2019).

As a result, in this analysis, we examined both the overall data center workload and server characteristics. Our strategy also promotes the best possible balance of host utilization and energy saving. The goal is for servers to use the least amount of energy possible. For this purpose, we employed the upper and lower threshold values for overload and underload detections of servers.

Related Work

We evaluated and summarised the various existing research findings on VM consolidation to reduce energy consumption in data centers below.

Beloglazov *et al*. deployed dynamic server consolidation methodologies to achieve SLAs and reduce data center energy consumption. The linked framework employs higher and lower CPU utilization criteria, as well as a best-fit lowering technique, to manage the placement of the selected VMs, which are to be relocated from overutilized to underutilized hosts. Furthermore, to detect overutilized host methods, the median absolute deviation (MAD), interquartile range (IQR), and local regression (LR) algorithms were established. The offered solutions reduce SLA violations and optimize energy usage, but they increase the cost of live migrations (Beloglazov & Buyya, 2012).

Patel *et al*. developed a host utilization aware (HUA) method to find and place virtual machines (VMs) on under-utilized hosts. The approach considers the total usage of the entire data center to set the value of the lower limit. Experimental results showed that HUA is efficacious in identifying under-utilized hosts and reducing energy consumption (Patel & Patel, 2020).

Zahedi *et al*. suggested a unique VM consolidation approach for achieving load balancing in cloud data centers. The idea is to consolidate VMs on high-performance servers to save energy consumption while increasing workload (Zahedifard *et al*., 2017). The consolidation issue was considered a multi-issue by Li *et al*., with limitations, for instance, minimizing energy consumption, ensuring QoS, and maximizing resource utilization. An energy-aware DVM strategy was used to handle this challenge, which migrates VM while fulfilling limitations on the likelihood of several kinds of resources being overloaded. The suggested method achieves the best possible balance of increasing energy efficiency, maximizing resource use, and ensuring QoS (Li *et al*., 2018).

Zhang *et al*. suggested a two-strategy solution for work scheduling optimization in cloud computing. A classifier is employed in the first strategy to categorize tasks based on previous data. A matcher is used in the second step to dynamically match tasks to a concrete VM. The suggested technique surpasses current algorithms, including max-min and min-min in terms of task scheduling and execution, as per experimental evidence (Zhang & Zhou, 2018)

Wang *et al*. created a novel paradigm for sustainable cloud computing that is energy-aware dynamic VM consolidation. The principal goal of this scheme was to move VMs to hosts with the least amount of available MIPS after they have been assigned. The simulation results show that the suggested work saves energy and meets SLA better than other methods [Wang, H. & Tianfield, H., 2018]

Ruan *et al*. presented a method called "PPRGear", based on sampling utilization levels with different performance-to-power ratios (PPR) of every host. When compared to established techniques, IqrMc, MadMmt, and ThrRs, the proposed solution saves around 69.31% of energy while requiring fewer migrations and shutdowns and little performance degradation for cloud computing data centers (Ruan *et al*., 2019).

Unlike earlier studies in the field, we aim to improve cloud services by optimizing resource usage to lower energy usage whilst reducing the number of live migrations significantly in our proposed model.

Assessment Models and Metrics

In this study, we find that server power usage is proportional to CPU utilization, assuming that an idle server requires around 70% of the power that a fully occupied server requires. As a basis, we define power consumption P(cu) in terms of CPU utilization in Equation (1).

$$\mathrm{P(cu)} = \mathrm{k} \times P_{max} + (1 - \mathrm{k}) \times P_{max} \times \mathrm{cu} \tag{1}$$

Where k is the percentage of energy used by idle hosts (70%), Pmax is the maximum power of a running server at 100% CPU usage, and cu is the current CPU utilization.

Due to the workload uncertainty, CPU consumption may change over time and be defined as current utilization (t). As a result, Equation (2) expressed a host's energy consumption as an integral of power consumption over time.

$$Total\ Energy = \int P(cu(t))dt \tag{2}$$

Two metrics were utilized in the IaaS model to define service level agreement violation (SLAV). Equation (3) represents the function.

$$\mathrm{SLAV = SLAV\_TPAH \times PD\_LM} \tag{3}$$

Where SLAV_TPAH denotes service level agreement violation time per active host, which represents the proportion of time that servers have 100% CPU utilization. Equation. (4) represents the function:

$$\mathrm{SLAV\_TPAH} = \frac{1}{\mathrm{Pm}} \sum_{p=1}^{\mathrm{Pm}} \frac{\mathrm{T\_sp}}{\mathrm{T\_ap}} \tag{4}$$

Where Pm is the number of servers (i.e., physical machines); $T_{\_sp}$ denotes the total time that server p has experienced 100% CPU use, resulting in an SLA violation; and $T_{\_ap}$ denotes the total time that server p has been in active mode. Equation (5) presents the overall performance decrease by VMs owing to live migrations (PD_LM):

$$\mathrm{PD\_LM} = \frac{1}{\mathrm{Vm}} \sum_{v=1}^{\mathrm{Vm}} \frac{\mathrm{C\_dv}}{\mathrm{C\_rv}} \tag{5}$$

Where Vm is the number of VMs; $C_{\_dv}$ denotes the estimated performance loss of VM due to migrations, and $C_{\_rv}$ denotes the servers and performance loss owing to VM migrations.

To relate the efficiency of the algorithms with others, a new metric product of energy and SLA violation (ESV) was established in a study that combines energy usage and SLA violation, as shown below in Equation (6).

$$\mathrm{ESV = SLAV \times Energy} \tag{6}$$

The percentage enhancement (a)of the suggested approach over the existing one is calculated using Equation (7)

$$\alpha = \left(\frac{\mathrm{Existing_{Approach} - Proposed_{Approach}}}{\mathrm{Existing_{Approach}}}\right) \times 100 \tag{7}$$

Proposed Approach for VM Consolidation

The primary goal of this research is to reduce energy usage and keep the hosts' performance at its best. To achieve this goal, the performance-to-power ratio is used. Furthermore, the PPR specifies energy consumption efficacy, which is equal to the number of server-side Java operations (SSJ) completed within a certain period divided by the period's average active energy consumption. We used the PPR values from the Standard Performance and Evaluation Corporation (SPEC) website (https://www.spec.org/power ssj2008/results/) in this work. Furthermore, we proposed a novel VM consolidation technique for efficient power utilization.

Overutilized server detection

We check a server's PPR to identify over-utilized hosts, and an upper threshold is set based on the performance-to-power ratio (PPR). We are looking for the highest PPR between idle and over-utilized servers. The total CPU utilization of servers is then compared with the highest PPR. If the server's CPU utilization surpasses the highest PPR level, the server is considered over-utilized as over-utilized servers consume more power than medium-utilized. As a result, some of the VMs must be relocated in order to make them medium-utilized hosts. This design keeps all servers operating at peak performance over power, allowing the system to execute more tasks while consuming less power.

In this study, we have selected six distinct server configurations. Table 1 shows the PPR of these servers. The upper threshold is set at 90% CPU utilization for HP ProLiant DL325 and Dell Inc R7425 servers, 80% CPU utilization for HTC Fusion 2288H, 70% CPU utilization for HP ProLiant DL360 servers, and 60% CPU utilization for FUJITSU TX1320 M4 and Sugon I620-G30 servers. Table 2 shows the pseudocode for the server over-utilized detection algorithm.

Underutilized server detection

We examine the entire usage of the data center to identify underutilized servers. Next, we calculate the total expected

Table 1: PPR of selected servers

Model name	*100%*	*90%*	*80%*	*70%*	*60%*	*50%*
HP ProLiant DL325	11,726	11,299	10,570	9,735	8,812	7,797
Dell Inc R7425	15,238	14,597	13,888	13,150	12,312	11,153
HTC Fusion 2288H	13,351	15,413	16,675	16,386	15,864	14,702
HP ProLiant DL360	12,518	12,340	12,445	12,789	12,415	11,789
FUJITSU TX1320 M4	8,821	9,390	10,074	10,503	10,643	10,363
Sugon I620-G30	10,928	10,839	10,941	11,559	11,707	10,958

Table 2: Pseudocode of over-utilized server detection algorithm

```
1. Begin
2. Input: Hostlist, Vmlist and Initialize UpperThreshold = 0,
TotalRequestedMips = 0
3. for each Host Host1 to Hostn
3.1 Find the upper threshold of each host, that is the highest
performance-to-power ratio of the host
3.2 UpperThreshold = Host.GetPPR()
3.3 for each virtual machine get Vmlist Vmi to Vmj
3.3.1 Vmlist = Host.Getvmlist()
3.3.2 Update the TotalRequestedMips by adding the required Mips
of each Vms
3.4 End for
3.5 Obtain CPU Utilization of Host by
3.5.1 Utilization = (TotalRequestedMips) /(Host.Get Totalmips)
3.6. Comparison Of CPU Utilization and Upper Threshold by
3.6.1 If Utilization> UpperThreshold then
3.7 Host status is Over-Utilized
4. End for.
5. End
```

Table 3: Pseudocode of under-utilized server detection algorithm

1. Begin
2. Inputs: PowerHost HostList, PowerModel Type, ExcludedHostList, Initialize: LowerThreshold = 0, MinUtilization = 0, OverallUtilization = 0
3. Add all overutilized hosts and switched off hosts in the excluded hosts list
4. For each host of HostList Host1 to Hostn
4.1 if Host in Excluded HostList
4.1.1 continue
4.2 Utilization = host.getUtilizationofCPU()
5. End for
6. For each host of HostList Host1 to Hostn
6.1 Get HostList of each category by
6.2 if Host.getpowerModel().equals(type) then
6.2.1 hostListbyCategory.add(host)
6.3 Get overall utilization of host category-wise
6.4 OverallUtilization += host.getUtilizationofCPU()
7. End for
8. Get the Upper Threshold Value of the same category host by
8.1 UpperThreshold= hostListbyCategory.get(0).getPPR()
9. Find the Maximum No. of Under-Utilized hosts
9.1 MaxNoOf UnderUtilizedHosts = HostListbyCategorySize() – (OverallUtilization)/UpperThreshold
10. Sort of CPU Utilization of Categorised Host in Ascending order
11. Indexing the Underutilized host using Max No. of Under-Utilized hosts
11.1 index = MaxNoOf UnderUtilizedHosts - 1
12. Obtain a Lower Threshold value by
12.1 LowerThreshold = Category.get(index).getUtilizationofCPU()
13. Update the value of variable MinUtilization by MinUtilization = LowerThreshold
14. Comparison of Utilization and MinUtilization to get Under-Utilized host
14.1 If utilization > MinUtilization then
14.1.1 host status is Under-Utilized
14.2 End if
15. End

number of underutilized servers. The lower threshold value is set dynamically depending on the higher threshold value and predicted number of underutilized servers. Data centers may be a collection of clusters. Total CPU utilization is the sum of all CPU use across all hosts of the same cluster. As a result, we computed a lower threshold value based on cluster size, a higher threshold value, and total CPU utilization for each cluster. Table 3 shows the pseudocode for the server's underutilized detection algorithm.

VM selection

A physical machine (PM) may run many VMs. When a PM becomes overutilized, one or more VMs must be relocated to lower the PM's workload and restore its normal. As a consequence, the SLA will be upheld. This paper proposes a unique VM selection approach, deviation with minimal migration time (DWMMT), which calculates the difference between the upper threshold and CPU usage. The VM chosen to migrate from the overutilized host will have CPU utilization more than or equal to the deviance and the shortest migration time required to return the server to its normal state. Table 4 depicts the VM selection algorithm's pseudocode. The selection process employs the safety factor «sf» of 0.2 to avoid CPU overhead due to VM live migration, as live migration causes a 10% CPU usage overhead.

Table 4: Pseudocode of VM selection algorithm

1. Begin
2. Input: OverUtilizedHostList, Initialize: MinRam = MAX, OverallVmUtilization = 0, k=0.2, SelectedVm = null;
3. Obtain CPU Utilization of Host by hostUtilization = host.getUtilizationofCPU()
4. Obtain Deviation between Upper Threshold &Host Utilization Deviation = (hostUtilization – host.getPPR()) +sf
5. for Vm in VmList
5.1 VmUtilization= Vm.getUtilization
5.2 If VmUtilization >= Deviation //Comparison of Vm Utilization and Deviation.
5.2.1 If Vm.getRam < minRam//Obtain memory Utilization of Vm and compare with minRam
5.2.1.1 Update MinRam with current Vm Ram
5.2.2 End if
5.3 End if
6. End for
7. SelectedVm= Vm
8. If SelectedVm != NULL
8.1 VmListToMigrate.add(SelectedVm)
8.2 for each Vm in VmList
8.2.1 Update overall Utilization by OverallVmUtilization += Vm.getUtilization
8.2.2 Add vms to migrating vm list
8.2.3 If OverallVmUtilization >= Deviation
8.2.3.1 break //go to step 10
8.2.4 End if
8.3 End for
9. End if
10. Return VmListToMigrate
11. End

VM placement

Three classifications of data centers are under-utilized, middle-utilized, and over-utilized. The middle-utilized hosts will be the primary receptacles for VM deployment. Table 5 depicts the pseudocode of the VM placement algorithm.

Table 5: Pseudocode of VM placement algorithm

```
1. Begin
2. 1. Input: OverUtilizedHostList, UnderUtilizedHostList, ExcludedHostList; Initialize: indexKey = 0;
3. Get an UnderUtilized Host and a new placement for Vms // Add all the OverUtilized hosts and switched off hosts to ExcludedHostList for Finding UnderUtilizedHost and New Placement of Vms.
4. While IndexKey < UnderUtilizedHostList.size
4.1 If HostList.size == (ExcludedHostListFindingUnderUtilizedHost.size)
4.1.1 Break // go to the end of the while loop
4.2 End if
4.3 UnderUtilizedHost = getUnderUtilizedHost(excludedHostsForFindingUnderUtilizedHost)
4.4 if (UnderUtilizedHost == null)
4.4.1 Break // go to the end of the while loop
4.5 End if
4.6 ExcludedHostsFindingUnderUtilizedHost.add(UnderUtilizedHost)
4.7 ExcludedHostsFindingNewVmPlacement.add(UnderUtilizedHost)
4.8 For each vm of Hosts
4.8.1 minPowerConsumption = Double.MAX_VALUE
4.8.2 PowerHost allocatedHost = null
4.8.3 For each server in HostList
4.8.3.1 if ExcludedHostListFindingNewPlacecement.contains(server)
4.3.1.1 Continue// (go to for loop)
4.8.3.2 End if
4.8.3.3 Utilization=Server.getUtilizationOfCPUAfterAllocation(vm)
4.8.3.4 if Utilization >UpperThreshold for this server
4.8.3.4.1 Continue    // (go to for loop)
4.8.3.5 End if
4.8.3.6 PowerConsumption = server.getEstimatedPowerAfterAllocation (vm)
4.8.3.7 If PowerConsumption < minPowerConsumption
4.8.3.7.1 AllocatedHost = server
4.8.3.7.2 PowerConsumption = minPowerConsumption
4.8.3.8 End if
4.8.4 End for
4.8.5 If allocatedHost != NULL
4.8.5.1 If!ExcludedHostListFindingUnderUtilized.contains(allocatedHost)
4.8.5.1.1 ExcludedHostListFindingUnderUtilized.add (allocatedHost)
4.8.5.2 End if
4.8.6 allocation.add(vm, allocatedHost)
4.8.7 Else
4.8.7.1 Break
4.8.8 End if
4.9 End for
5. IndexKey = indexKey+1
6. End while
7. End
```

We add overutilized and switched-off to the excluded hosts list, first for finding under-utilized hosts and second for finding new locations for selected hosts. This operation helps to avoid future searches of such hosts, resulting in a computational improvement. Then the under-utilized server prevents considering a container for VM deployment. We then try to identify the best allocation map for the VM of each underutilized host to reduce total energy consumption. We also prevent overwhelming the hosting servers when deploying VMs.

Simulation setup

CloudSim is widely considered one of the most powerful cloud simulators. It supports virtualized resource management and modeling, as well as power consumption, workload dynamics, virtual machine migration, and SLA evaluations (Patel, 2016). To evaluate the power efficiency of our technique PPR_DWMMT1.1, CloudSim 3.0.3 simulator with the deployment of six real-world unique host models and various workloads used.

To simulate a cloud environment, a data center with 900 homogenous host PCs in six models: HP ProLiant DL325, Dell Inc R7425, HTC Fusion 2288H, HP ProLiant DL360, FUJITSU TX1320 M4 and Sugon I620-G30 is employed. The specifications of these servers are displayed in Table 6. Furthermore, we used the SPECpower ssj2008 benchmark suite (SPECpower ssj2008 Results) to deploy the power model in CloudSim, as shown in Table 7.

Heterogenous with single core 1000 VM were set up in response to customer requirements. Table 8 displays the characteristics of the VM classes based on Amazon EC2 (Amazon EC2 Instance Types - Amazon Web Services)(https://aws.amazon.com/ec2/instance-types/, s. d.).

For testing purposes, we used real-world system workload traces. This research makes use of Planet-Lab node data (Park & Pai, 2006). These data are based on CPU usage from 10 randomly chosen days in March and April 2011.

The proposed solutions leverage a safety parameter to manage energy consumption and SLA violations. We use this to change allocation when our method fails. In our situation, the safety parameter is 1.1, whereas the baseline techniques have 1.5, 1.2, 2.5, and 0.8 for IQR_MMT, LR_MC, MAD_MU and THR_RS, respectively.

Table 6: Configurations of servers

Model Name	*MIPS*	*Core*	*RAM (GB)*	*Bandwidth (Gbps)*	*Number of hosts*
HP ProLiant DL325	2000	32	128	1	150
Dell Inc R7425	2200	64	128	1	150
HTC Fusion 2288H	2000	56	112	1	150
HP ProLiant DL360	2500	28	48	1	150
FUJITSU TX1320 M4	3700	6	16	1	150
Sugon I620-G30	2100	44	192	1	150

Table 7: Power consumption (watt) at different loads

Model Name	*100%*	*90%*	*80%*	*70%*	*60%*	*50%*	*40%*	*30%*	*20%*	*10%*	*0%*
HP ProLiant DL325	181	170	161	153	145	136	127	116	105	92.3	61.7
Dell Inc R7425	287	271	253	234	214	197	182	168	154	138	84.9
HTC Fusion 2288H	437	341	280	249	220	198	180	161	142	123	51.3
HP ProLiant DL360	237	217	191	163	144	126	109	94.1	80.7	68.1	38.9
FUJITSU TX1320 M4	73.9	62.1	51.6	43.3	36.8	31.2	26.9	23.5	20.5	17.9	12.8
Sugon I620-G30	364	334	295	243	206	183	167	150	134	117	53.1

Table 8: Characteristics of VM categories

VM Category	*MIPS*	*Core*	*RAM (MB)*	*Bandwidth (Mbps)*
High-CPU medium instance	2500	1	870	100
Extra-large instance	2000	1	1740	100
Small instance	1000	1	1740	100
Micro instance	500	1	613	100

Results and Discussion

As demonstrated in Figure 1, PPR_DWMMT_1.1 consumes the least amount of energy among the other algorithms throughout the course of 10 days. This is because PPR_DWMMT1.1 determines upper thresholds based on the server's performance class. The adaptive upper thresholds reasonably split the cloud data center into performance clusters in which all PMs belonging to a specific class of performance work at their best throughput with the least amount of energy consumption.

In comparison to the other methods, PPR_DWMMT_1.1 significantly reduces the average energy consumption over 10 days. We found an average of 15.417 kWh for IQR_MMT_1.5, 14.643 kWh for LR_MC_1.2, 15.792 kWh for MAD_MU_2.5, 14.581 kWh for THR_RS_0.8 kWh and 1.98 kWh for PPR_DWMMT_1.1, as shown in Figure 2.

The number of live migrations continues to be a primary factor influencing energy consumption. Reducing such a number may decrease the total data center load and, as a result, may help to reduce SLA violations. As shown in Figure 3, our scheme has the lowest number of live migrations for the entire 10-day period. The cause for this is the dynamic lower barrier used to select underloaded hosts. By removing overloaded and switched-off hosts, the estimated value of the maximum number of hosts that can be evacuated remains a good measure for assessing the lower threshold, which is computed based on the associated higher threshold. Moreover, our VM selection strategy selects the VM with the best balance of workload and migration time. As a result, PPR_DWMMT_1.1 achieves fewer live migrations than the other remaining VM selection rules.

As shown in Figure 4, our scheme has the lowest number of host shutdowns for the entire 10-day period.

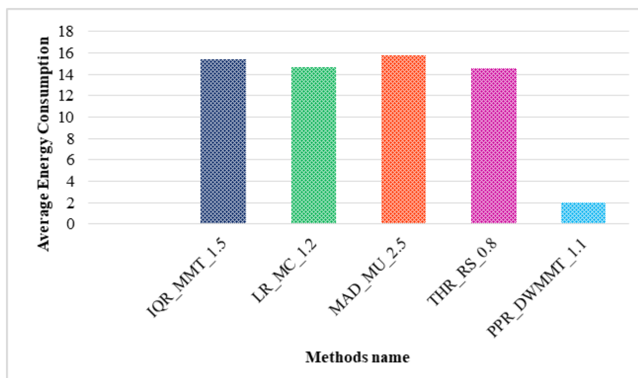

Figure 2: Average energy consumption

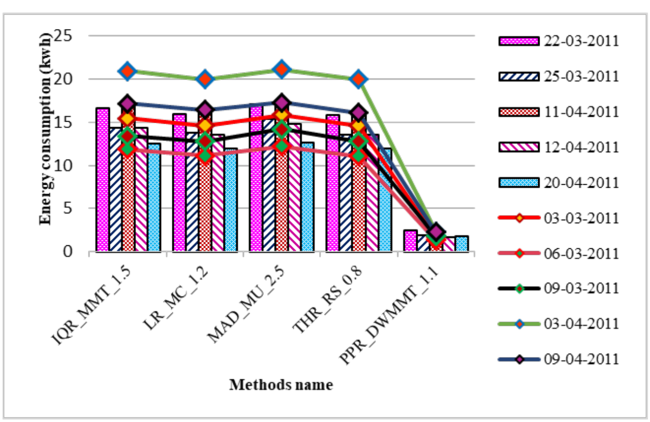

Figure 1: Total energy consumption (Kwh)

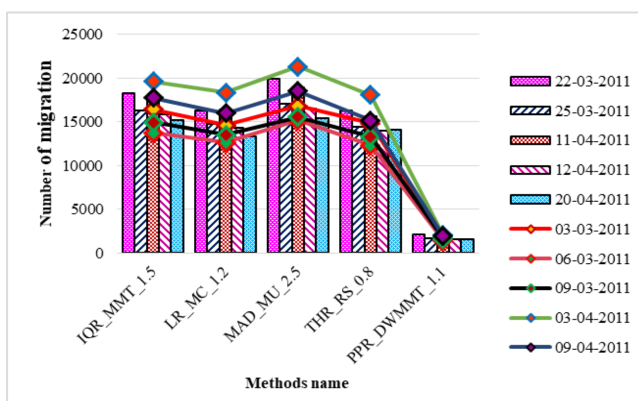

Figure 3: Number of VMs migration

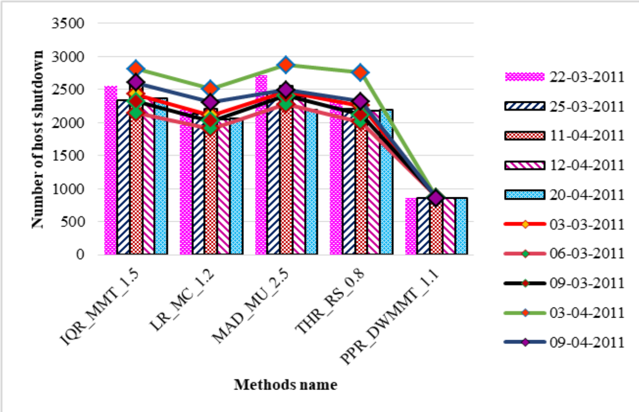

Figure 4: Number of hosts shutdown

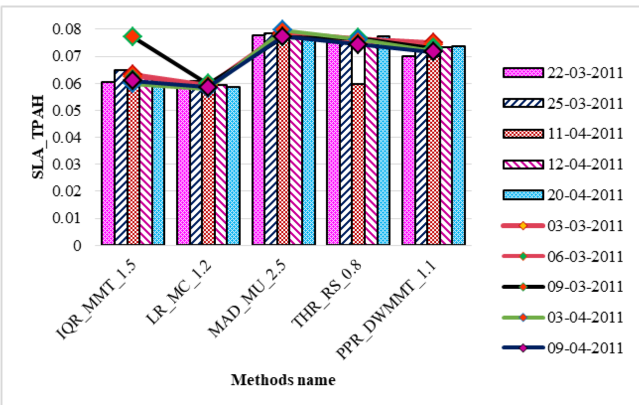

Figure 5: Service agreement violation time per active host

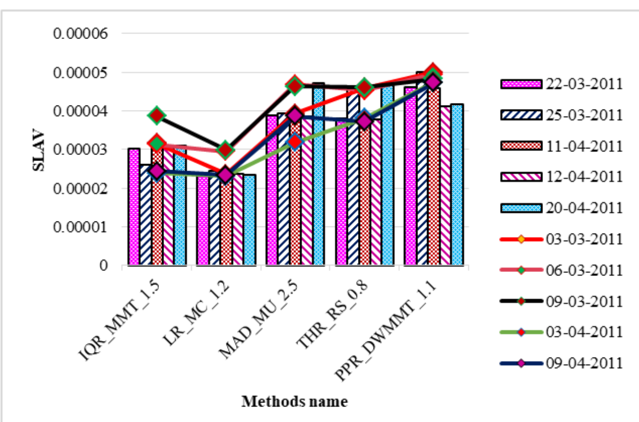

Figure 6: Service level agreement violation

Figure 5 depicts the service level agreement violation time per active host (SLAV_TPAH), the value of our scheme is comparable to all the baseline methods.

As shown in Figure 6, the proposed scheme's service level agreement violation (SLAV) is closer to both baseline methods. This is due to the proposed system's amazingly efficient energy usage.

The EESV values are depicted in Figure 7. Our approach has a lower value, giving a better balance between energy

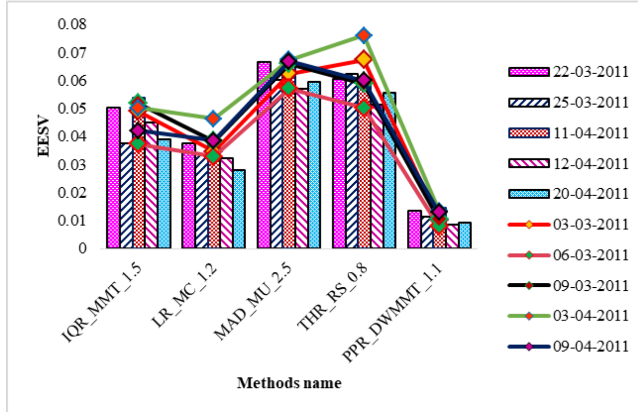

Figure 7: Energy and SLA violation

usage and SLA violation. SLAV is followed, and the energy consumption is reduced to better levels.

Conclusion

Reduced energy usage is a critical need for maximizing the benefits of cloud providers. A significant strategy for managing energy consumption is VM consolidation. This article presents an energy-efficient technique, namely PPR_DWMMT_1.1, for consolidating VMs in cloud data centers. The basic objective is to reduce energy usage while taking into account host utilization. We attempted to maintain the data center running at maximum throughput while consuming the least amount of energy.

The results of our proposed approach PPR_DWMMT_1.1 are compared with the four reference approaches, namely IQR_MMT_1.5, LR_MC_1.2, MAD_MU_2.5 and THR_RS_0.8. Our approach has been observed to reduce average energy consumption by 87.15, 86.47, 87.46 and 86.42% compared to existing approaches IQR_MMT_1.5, LR_MC_1.2, MAD_MU_2.5 and THR_RS_0.8, respectively. Furthermore, our approach reduced the number of live migrations, resulting in less performance deterioration by VMs owing to migrations. In future work, our proposed method can be extended to a real-time cloud environment with consideration of other variable factors.

Acknowledgment

The authors acknowledge that management and the principal responsibility for assisting in the conduct of research work.

References

Amazon EC2 Instance Types. https://aws.amazon.com/ec2/instance-types/. (s. d.).

Asad, Z. & Rehman Chaudhry, M. A. (2017). A Two-Way Street: Green Big Data Processing for a Greener Smart Grid. *IEEE Systems Journal*, *11*(2), 784-795. https://doi.org/10.1109/JSYST.2015.2498639

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. *Concurrency and Computation: Practice and Experience*, *24*(13), 1397-1420. https://doi.org/10.1002/cpe.1867

Buyya, R., Yeo, C. S. & Venugopal, S. (2008). Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. *10th IEEE International Conference on High Performance Computing and Communications*, 5-13.

el Kafhali, S. & Salah, K. (2018). Modeling and Analysis of Performance and Energy Consumption in Cloud Data Centers. *Arabian Journal for Science and Engineering*, *43*(12), 7789-7802. https://doi.org/10.1007/S13369-018-3196-0

Farahnakian, F., Pahikkala, T., Liljeberg, P., Plosila, J., Hieu, N. T. & Tenhunen, H. (2019). Energy-aware VM consolidation in cloud data centers using utilization prediction model. *IEEE Transactions on Cloud Computing*, *7*(2), 524-536. https://doi.org/10.1109/TCC.2016.2617374

Hanini, M., el Kafhali, S. & Salah, K. (2019). Dynamic VM allocation and traffic control to manage QoS and energy consumption in cloud computing environment. *International Journal of Computer Applications in Technology*, *60*(4), 307-316. https://doi.org/10.1504/IJCAT.2019.101168

Karuppasamy, M. & Balakannan, S. P. (2018). Energy saving from cloud resources for a sustainable green cloud computing environment. *Journal of Cyber Security and Mobility*, *7*(2), 95-108. https://doi.org/10.13052/jcsm2245-1439.718

Li, Z., Yan, C., Yu, L. & Yu, X. (2018). Energy-aware and multi-resource overload probability constraint-based virtual machine dynamic consolidation method. *Future Generation Computer Systems*, *80*, 139-156. https://doi.org/10.1016/J.FUTURE.2017.09.075

Moghaddam, S. M., Walker, C., Unsworth, C. P., Piraghaj, S. F. & O'Sullivan, M. (2019). Energy-efficient and SLA-aware virtual machine selection algorithm for dynamic resource allocation in cloud data centers. *Proceedings - 11th IEEE/ACM International Conference on Utility and Cloud Computing, UCC 2018*, 103-113. https://doi.org/10.1109/UCC.2018.00019

Naidu, P. A. & Chadha, P. (2020). Efficient Strategies for Green Cloud Computing. *Journal of Network Communications and Emerging Technologies (JNCET)*, *10*(6).

Park, K. S. & Pai, V. S. (2006). CoMon: a mostly-scalable monitoring system for PlanetLab. *ACM SIGOPS Oper. Syst. Rev.*, *40*(1), 65-74. https://doi.org/10.1145/1113361.1113374

Patel, N. (2016). A Comprehensive Assessment and Comparative Analysis of Simulations Tools for Cloud Computing. *International Journal Of Engineering And Computer Science*. https://doi.org/10.18535/ijecs/v5i11.43

Patel, N. & Patel, H. (2020). Energy efficient strategy for placement of virtual machines selected from underloaded servers in compute Cloud. *Journal of King Saud University - Computer and Information Sciences*, *32*(6), 700-708. https://doi.org/10.1016/J.JKSUCI.2017.11.003

Ruan, X., Chen, H., Tian, Y. & Yin, S. (2019). Virtual machine allocation and migration based on performance-to-power ratio in energy-efficient clouds. *Future Generation Computer Systems*, *100*, 380-394. https://doi.org/10.1016/j.future.2019.05.036

Wang, H. & Tianfield, H. (2018). Energy-Aware Dynamic Virtual Machine Consolidation for Cloud Datacenters. *IEEE Access*, *6*, 15259-15273. https://doi.org/10.1109/ACCESS.2018.2813541

Yadav, R., Zhang, W., Kaiwartya, O., Singh, P. R., Elgendy, I. A. & Tian, Y. C. (2018). Adaptive Energy-Aware Algorithms for Minimizing Energy Consumption and SLA Violation in Cloud Computing. *IEEE Access*, *6*, 55923-55936. https://doi.org/10.1109/ACCESS.2018.2872750

Zahedifard, S. Y., Ahmadi, M. R. & Adabi, S. (2017). A dynamic VM consolidation technique for QoS and energy consumption in cloud environment. *The Journal of Supercomputing*, *73*(10), 4347-4368. https://doi.org/10.1007/S11227-017-2016-8

Zhang, P. Y. & Zhou, M. C. (2018). Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy. *IEEE Transactions on Automation Science and Engineering*, *15*(2), 772-783. https://doi.org/10.1109/TASE.2017.2693688