ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

An Energy Efficient Virtual Machine Consolidation in Cloud Computing

Jayshri Damodar Pagare[1], Dr. Nitin A Koli [2]

Research Scholar, Sant Gadge Baba Amravati University, Amravati, India[1]

Head, Computer centre, Sant Gadge Baba Amravati University, Amravati, India[2]

ABSTRACT: Energy efficiency is becoming a very important issue in Cloud Computing environments and can be achieved using virtual machine consolidation. In this paper, for cloud environments, a VMs consolidation approach is proposed. Proposed algorithms have been implemented and evaluated using CloudSim simulator. Simulation results show improvement in energy consumption which leads to energy efficiency.

KEYWORDS: Energy efficiency; Data Center; Cloud Computing; virtual machine consolidation.

I. Introduction

The use of virtualization allows providers to run multiple workloads from different customers on the same computing infrastructure which makes Cloud Computing as a promising approach to improve resources utilization. With adoption of virtualization services, cloud computing platforms are becoming more popular. However, Cloud providers rely on large and power-consuming data centers. The wide adoption of virtualization and cloud computing has added another layer of complexity to enabling an energy-efficient use of computing power in large-scale settings[1].Energy consumption has always been a major concern in the design and cost of datacenters.

Energy-efficient Cloud environments need to deal not only with energy consumption but also with increasing demand, high QoS expectations and sustainable services[2]. However, a large number of cloud servers consume massive energy and produce huge pollution. The Smart2020 analysis shows that cloud-based computing data center and the telecommunication network will generate emission about 7% and 5% each year in 2002 and 2020,respectively.[3]
Power consumption is one of the most critical problems in data centers. One effective way to reduce power consumption is to consolidate the hosting workloads and shut down physical machines which become idle after consolidation.[4]

The solution to this issue is to migrate and consolidate VMs in data center. Although it can greatly reduce energy consumption, it will result in SLA violation. Obviously, we cannot eliminate the SLA violation by this way, but we can reduce SLA violation as much as possible while reducing energy consumption, which is called energy-consumption tradeoff and also the focus of this article.

The main focus of this work is to introduce an efficient SLA-aware algorithm(i.e., to avoid SLA violation as much as possible). The proposed algorithms consider the trade-off between energy consumption and performance. The most important contributions of this paper are as follows:
– Proposing an efficient algorithm which finds and decides overloaded host with SLA violation
– Proposing an efficient algorithm for finding underutilized hosts.
– Combine two proposed algorithms as virtual machine consolidation algorithm to get better results in both power consumption and SLA violation.

The overload detection finds overload host and get the status of Overload host whether it result in SLA Violation or not. If there is no SLA violation then no migration of virtual machines required which saves power required to migrate VMs. But if there is SLA violation then place all the VMs from this host to other hosts until the host becomes Underload.

DOI: 10.15680/ijircce.2015.0305091

The underload detection algorithm finds the host with the minimum utilization compared to the other hosts, and tries to place all the VMs from this host to other hosts keeping them not overloaded. If this can be done, the VMs are set for migration to the determined destination hosts, and the source host is switched to the sleep mode once all the migrations have been completed. If all the VMs from the source host cannot be allocated, the host is kept active. This algorithm is repeated for all hosts that have not been considered as being overloaded.

The paper is organized as follows. In Sect. II the related work is discussed. In Sect. III general overview of processing steps is given. In Sect. IV consolidation algorithm is introduced. In Sects. V mehodology and metrics are discussed. In Sect VI the experimental setup and their results are discussed and in Sect. VII the system model is introduced that has been used in our work. Finally Sect. VIII concludes the paper.

II. RELATED WORK

The first work in large-scale virtualized data centers to manage power efficiently ,in which power management has been applied in the context of virtualized data centers, has been done by Nathuji and Schwan [5]. The authors have proposed an architecture of a data center's resource management system where the resource management is split into local and global managers. At the local level the system leverages the guest OS's power management strategies. The global manager gets the information on the current resource allocation from the local managers and applies its policy to decide whether the VM placement needs to be adapted. Local manager coordinates power management methods of VMs in each host because the authors assumed that VM guests have a power aware OS. Global manager monitors the performance of multiple hosts and selects the appropriate host for requested VM migration.

Verma et al.[6] have formulated the problem of power-aware dynamic placement of applications in virtualized heterogeneous systems as bin packing problem with variable bin sizes and costs as continuous optimization: at each time frame the placement of VMs is optimized to minimize power consumption and maximize performance. Live migration is used to move VMs to new placement at each time frame. In their work SLA is not considered. In view of workload consolidation traditionally driven by performance objectives in virtualized environments, they have proposed an architecture of a power-aware application consolidation framework (pMapper), which can incorporate various scenarios including power and performance management using Virtualization technology.In their new work [7], VM consolidation strategies are divided into three categories:

static, semi-static and dynamic.
Two energy-aware task consolidation heuristics are proposed by Lee and Zomaya in [8] the objective is to maximize resource utilization. The work is based on a linear power model and the problem of task placement of a process in a Cloud infrastructure is taken as a Bin-Packing problem. The main focus is on the resource utilization and ignore other factors.
Srikantaiah et al. [9] model the consolidation problem as a modified multi-dimensional bin-packing problem.
In literature [10], Xu et al. treat VM consolidation as a multiobjective optimization problem while minimizing energy consumption and SLA violation simultaneously and propose an improved genetic algorithm with fuzzy multi-objective evaluation for VM consolidation.

The heuristic approach presented in [11] use technique derived, from the Best Fit Decreasing algorithm.The goal is to place each migrating VM on the server that minimizes the overall power consumption of the data center. Its notable drawbacks are first, they use deterministic and centralized algorithms whose efficiency deteriorates as the size of the data center grows. The second drawback is that mapping strategies may require the concurrent migration of many VMs, which can cause considerable performance degradation during the reassignment process
Anton et al. [12] have defined an architectural framework and principles for energy-aware Cloud computing and has developed algorithms for energy-aware mapping of VMs, they have used the concept of dynamic consolidation of VM resource partitions.

The fixed threshold in [12] for vm consolidation is not suitable for virtual environment with variable workloads. Therefore, they[13] illustrate that VM consolidation for variability of workloads. Then they propose novel adaptive heuristics for dynamic consolidation of VMs . Results show that the allocation and selection algorithms save energy consumption. However, the SLA violation and energy consumption produced by the framework has a scope of improvement. Therefore, we propose another VM consolidation algorithm in CloudSim, which performs better .

III. BACKGROUND

In data center, energy consumption is enormous. Therefore, the energy consumption has become an urgent and important issue to be solved. In general, in the data center, the host's statuses has three types: Overload, Underload and Idle.

Overload are the host those may generate SLA violation; Underload means the host is in use but generates no SLA violation; Idle indicates the host is available (not in use). And then VM consolidation needs three main steps in CloudSim [13]:

Step 1. Select the Overload hosts in data center, then choose VMs from those hosts to migrate (denote the VMs as VmsToMigrate) until the Overload hosts become Underload.

Step 2. Select suitable placements for VmsToMigrate on Underload or Idle hosts based on the Minimum Power policy [13]. As a result, some Underload hosts may turn off and become Idle, and some Idle hosts may turn on and become Underload.

Step 3. Assume the overall Underload hosts as OUH, then exclude those Underload hosts (denote as EUH) who receive any migration from VmsToMigrate, and denote the remaining Underload hosts as RUH. Finally, turn off all the RUH which can migrate all VMs (residing on them) to the remaining RUH or the EUH without resulting in Overload.

For above steps, the proposed framework [14] in CloudSim significantly achieves energy efficiency. Still the framework needs a great performance improvement in terms of energy consumption and SLA violation reduction.In step 1, the framework selects VMs from an Overload host until the host becomes Underload. If the Overload host does not generate SLA violation, then the migration will be in vain and result in high energy consumption. Therefore, we require a appropriate method to decide the VM selection in this step.By considering above problem, we generate the algorithm for VM consolidation .

IV. ENERGY EFFICIENT CONSOLIDATION ALGORITHM

The proposed virtual machine consolidation is shown in Algorithm 1 has following steps

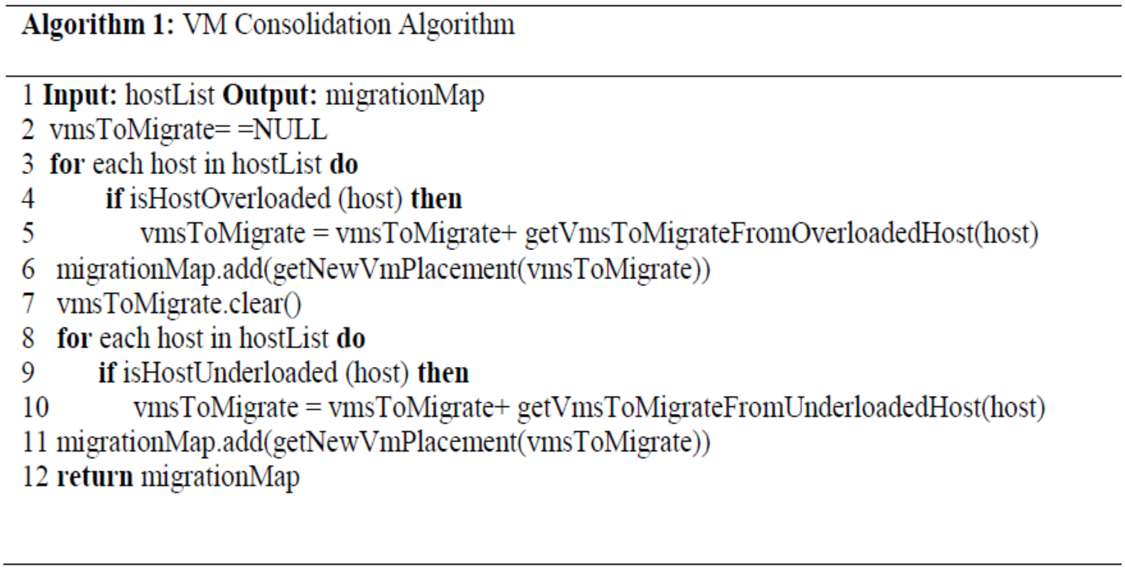
Step I:First, the algorithm see the list of hosts and by applying the proposed overload detection algorithm checks whether a host is overloaded. If the host is overloaded, the algorithm applies the VM selection policy to select VMs that need to be migrated from the host. If the Overload host does not generate SLA violation, then the migration will result in higher energy consumption. Therefore, we need a method to decide the status of Overload host whether it result in SLA violation or not.

Step II: Once the list of VMs to be migrated from the overloaded hosts is built, the VM placement algorithm in[13] is invoked to find a new placement for the VMs to be migrated.

Step III:Finding underloaded hosts and a placement of the VMs from these hosts.

The consolidation algorithm returns the combined migration map that contains the information on the new VM placement of the VM selected to be migrated from both overloaded and underloaded hosts. The complexity of the algorithm is 2N, where N is the number of hosts.

DOI: 10.15680/ijircce.2015.0305091



Algorithm 1: VM Consolidation Algorithm

```
1 Input: hostList Output: migrationMap
2  vmsToMigrate= =NULL
3  for each host in hostList do
4         if isHostOverloaded (host) then
5              vmsToMigrate = vmsToMigrate+ getVmsToMigrateFromOverloadedHost(host)
6  migrationMap.add(getNewVmPlacement(vmsToMigrate))
7  vmsToMigrate.clear()
8  for each host in hostList do
9        if isHostUnderloaded (host) then
10             vmsToMigrate = vmsToMigrate+ getVmsToMigrateFromUnderloadedHost(host)
11 migrationMap.add(getNewVmPlacement(vmsToMigrate))
12 return migrationMap
```

A. Overloaded host detection with status of SLA Violation

For step 4 in algorithm 1 above we have proposed an algorithm that finds and decides the status of overloaded host with SLA Violation Here we have used the parameters OverSLAV for the overloaded host with SLA violation , OverNSLAV for the overloaded with no SLA violation, Over for overloaded host, Under for underloaded host, Idle for the Idle host, Saturated for the host which does not send or receive any migration. According to [13], when the total request utilization of the VMs exceeds the allocated utilization of them on host Hi , Hi will generate SLA violation. If they are equal, it will be assumed to generate no SLA violation. The total allocated utilization of the VMs on host Hi can never exceed the maximum utilization of the host. It means that if the request utilization of the VMs on Hi exceeds the maximum utilization, the host will definitely generate SLA violation.

Once it has been decided that a host is overloaded and got its status , the next step is to select particular VMs to migrate from this host. Here we have used four policies for VM selection. The policies are applied iteratively. After a selection of a VM to migrate, the host is checked again for being overloaded. If it is still considered as being overloaded, the VM selection policy is applied again to select another VM to migrate from the host. This is repeated until the host is considered as being not overloaded. Here select VMs from the OverSLAV hosts until they become OverNSLAV or Saturated, and put the VMs into the VmsToMigrate. Finally, select VMs from the OverNS hosts using the proposed SLA violation decision algorithm until they become Saturated, and put the VMs into the VmsToMigrate.

B. Underloaded host detection

For step 9 in algorithm 1 above we have proposed an algorithm that find underloaded host .First find the CPU utilization of each host then sort in decreasing order so as to find the minimum utilization host as underloaded host to migrate all VMs from this host to other host by applying VM placement algorithm in[13] without overloading the other host.

The advantage of proposed algorithm is that it reduces no of migrations and hence reduces SLA violation and energy consumption, second a host with least number of VMs has a better chance to be switched to sleep mode in comparison with a host with more VMs.

V. METHODOLOGY AND METRICS

Energy efficiency metrics focus on advancing energy efficiency in data centers and computing ecosystems .

A. *The overload decision algorithm*

For ease, the algorithm can be abbreviated to ODA, which intent to decide a host Overloaded or not. Five ODAs have been implemented in CloudSim, i.e. Interquartile Range (IQR), Static Theshold (THR), Local Regression (LR), Robust Local Regression (LRR) and Median Absolute Deviation (MAD).
IQR As a measure of statistical dispersion in descriptive statistics, the IQR, also called the midspread or middle fifty, is equal to the difference between the upper (75%) and lower (25 %) quartiles.
THR The optimal online deterministic algorithm corresponds to the fixed threshold algorithm with threshold set to 100%,therefore,it is named THR-1.0.

LR In linear algebra, regression aims to find a trend line for a large set of data points. The LR aims to find a trend line by minimizing the sum of the absolute weighted distances between the line and the points.
LRR The LR is vulnerable to the outliers caused by heavy-tailed or other distributions. To make it robust, the LRR is proposed to assign an additional weight to each absolute distance in the LR so that it can weaken the ourliers.
MAD The process of MAD is as follows: firstly, calculate the median value of a set A. Then, take the absolute distances between the median and the points into set B. Finally, figure out the median value of the set B.

B. *The VM selection algorithm*

For ease, the algorithm can be abbreviated to VMSA, which aims to select VMs from Over hosts and prevent them from being Over. Four VMSAs have been implemented in CloudSim, i.e. Minimum Migration Time (MMT), Minimum Utilization (MU), Random Selection (RS) and Maximum Correlation (MC).
MMT The MMT aims to select a VMfrom Over host with the least migration time.
MU TheMU aims to select a VMfrom Over host with the least request utilization.
RS The RS aims to randomly select a VM from Over host.
MC The MC aims to select a VM with the maximum correlation with other VMs on Over host.

C. *SLA Violation Metrics*

Meeting QoS requirements is very important for Cloud computing. QoS requirements are commonly formalized as SLAs, which can be determined in terms of characteristics such as minimum throughput, maximum response time or minimum bandwidth and so on. These characteristics are workload or application dependent. However,the algorithm framework belongs to IaaS layer in Cloud computing and should be workload independent. Therefore, we use those SLA-related metrics defined in [13] to evaluate the proposed algorithm in our experiments. Simultaneously, we also use some other metrics including energy consumption, migrations and execution time.

(a) SLATAH (SLA violation Time per Active Host): The percentage of time, during which active hosts have experienced the CPU utilization of 100 %. When a host experiences 100 % utilization, it will not be able to allocate enough CPU to the VMs on it, so it will generate SLA violation. The SLATAH can be calculated using Eq. (1), where *Tsi* is the SLA violation time and *Tai* is the active time for *Hi* .

$$\text{SLATAH} = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{s_i}}{T_{a_i}} \qquad (1)$$

(b) Migrations: The total number migrations happened in the *N* hosts in data center.
(c) Energy: The total energy consumption in data center.
(d) PDM (Performance Degradation due to Migrations): The PDM can be calculated using Eq. (2), where *M* is the number of VMs in data center, *Cd j* is the estimate of the migration performance degradation of VM *j* , *Cr j* is the total

DOI: 10.15680/ijircce.2015.0305091

CPU utilization requested by *j* during its lifetime and *Cd j* is estimated as 10 % of the requested CPU utilization of *j* during its migration.

$$\text{PDM} = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{d_j}}{C_{r_j}} \qquad (2)$$

(e) SLAV (SLA Violation) and ESV (Energy and SLA Violation): The SLATAH is used to evaluate host-level SLA violation due to host overloading. And the PDM is used to evaluate VM-level SLA violation due to the VM migration. Since the two metrics are independent, a combined metric is needed to evaluate the two SLA violations. As a result, the SLAV is proposed to evaluate the two SLA violations, which can be calculated using Eq. (3). Since the Energy and the SLAV are two main metrics, the ESV is proposed to combine the two metrics, which can be calculated using Eq. (4).

$$\text{SLAV} = \text{SLATAH} \cdot \text{PDM} \qquad (3)$$

$$\text{ESV} = \text{SLAV} \cdot \text{Energy} \qquad (4)$$

(f) ETF (Execution Time of the Framework): The execution time of the frameworks.

VI. Performance evaluation

As the targeted system is an IaaS in cloud computing, it's better to evaluate the proposed algorithm on a large-scale virtualized data center infrastructure. However, it's very difficult to conduct repeatable large-scale experiments on a real infrastructure,which is required to evaluate and compare the proposed algorithm. Therefore, to ensure the repeatability of the experiments, simulations have been chosen as a suitable way to evaluate the performance of the proposed algorithm.

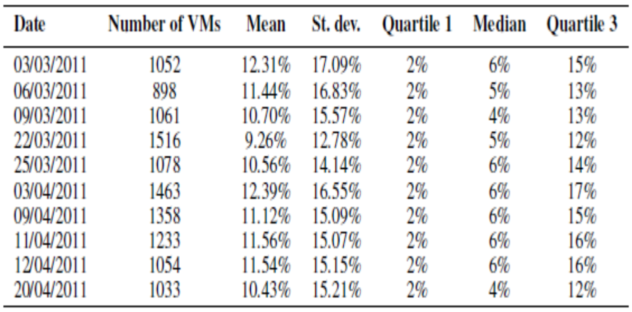
For our experiments, the CloudSim toolkit [14] has been chosen as a simulation framework. The toolkit has been developed by the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne. It supports both system and behavior modeling of cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. Currently, it supports modeling and simulation of Cloud computing environments consisting of both single and inter-networked clouds (federation of clouds),and also supports energy-efficient management of datacenter resources. Apart from the energy consumption modeling and accounting, the ability to simulate service applications with dynamic workloads has been incorporated

A. *Experimental Setup*

We have simulated a data center that comprises 800 heterogeneous physical nodes, half of which are HP ProLiant ML110 G4 servers, and the other half consists of HP ProLiant ML110 G5 servers.

The characteristics of the servers and data on their power consumption are given in VII a next.

Table I workload data characteristics(CPU utilization)



Date	Number of VMs	Mean	St. dev.	Quartile 1	Median	Quartile 3
03/03/2011	1052	12.31%	17.09%	2%	6%	15%
06/03/2011	898	11.44%	16.83%	2%	5%	13%
09/03/2011	1061	10.70%	15.57%	2%	4%	13%
22/03/2011	1516	9.26%	12.78%	2%	5%	12%
25/03/2011	1078	10.56%	14.14%	2%	6%	14%
03/04/2011	1463	12.39%	16.55%	2%	6%	17%
09/04/2011	1358	11.12%	15.09%	2%	6%	15%
11/04/2011	1233	11.56%	15.07%	2%	6%	16%
12/04/2011	1054	11.54%	15.15%	2%	6%	16%
20/04/2011	1033	10.43%	15.21%	2%	4%	12%

The frequency of the servers' CPUs are mapped onto MIPS ratings: 1860 MIPS each core of theHP 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. The characteristics of the VM types correspond to Amazon EC2 instance types with the only exception that all the VMs are single-core, which is explained by the fact that the workload data used for the simulations come from single-core VMs (Section VII a). For the same reason the amount of RAM is divided by the number of cores for each VM type: High-CPU Medium Instance (2500 MIPS, 0.85 GB); Extra Large Instance (2000MIPS, 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. However, during the lifetime, VMs utilize less resources according to the workload data, creating opportunities for dynamic consolidation.

B. *Workload Data*

In order to make the results more convincing, we need to use workload data obtained from real system. For our experiments, we have used data provided as a part of part of CoMon project, a monitoring infrastructure for PlanetLab [15], and it is about CPU utilization by more than a thousand VMs from servers located at more than 500places around the world. The interval of utilization measurements for the data is 5 minutes. There are 10-day CPU utilization records during March and April 2011 in the data. Table I given above from the literature [13], gives brief analysis about the workload.

C. *Result Analysis*

In the experiment, we use four-type ODAs (THR, LR, LRR and MAD) and four-type VMSAs (MC, MU, MMT, RS). Therefore, there are 16-type combinations policies for the ODA and the VMSA. According to the simulation results from reference [13], we set a suitable constant parameter for each ODA, and they are 0.8 for THR, 1.2 for LR, 1.2 for LRR and 2.5 for MAD respectively. The combination are THR_MC_0.8,THR_MU_0.8,THR_MMT_0.8, THR_RS_0.8, LR_MC_1.2, LR_MU_1.2, LR_MMT_1.2, LR_RS_1.2, LRR_MC_1.2,LRR_MU_1.2,LRR_MMT_1.2, LRR_RS_1.2, MAD_MC_2.5,MAD_MU_2.5,MAD_MMT_2.5, MAD_RS_2.5.

In figure 1, we use the 10-day workload to evaluate the proposed algorithm for the 16-type combination policies. In each subfigure of the figure, each candlestick represents results from the 10-day workload. For better comparison, we use the average of the results as an evaluation value.

1) The SLATAH evaluation: The minimum and the maximum evaluation values for the Origin are 5.5% and 7.87% respectively in figure 1.1(a). For the proposed algorithm, the values are 3.29% and 5.57% respectively as in figure 1.1(b).Compared with the Origin, the proposed algorithm has 40.18%~58.19% decrease for the 16-type combinations for the SLATAH respectively.

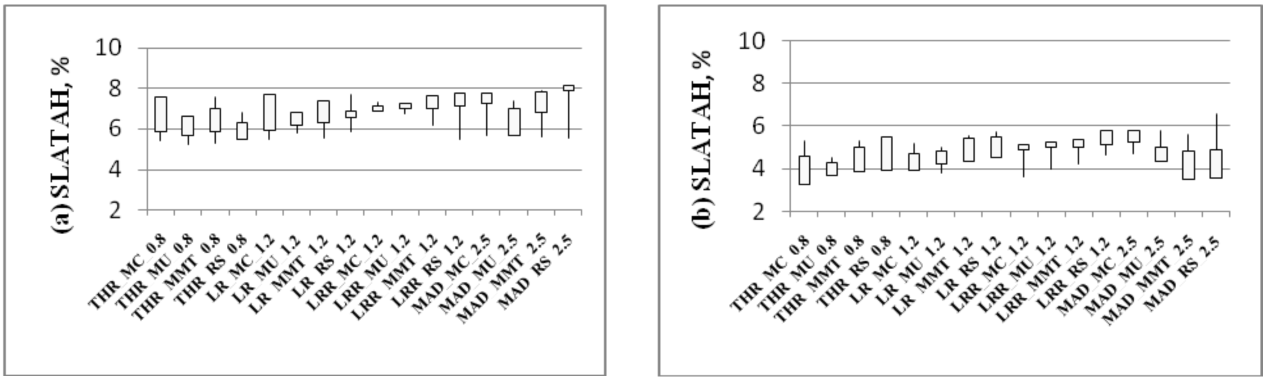


Figure 1.1 The SLATAH evaluation

2) The Migrations evaluation: In figure 1.2(a), for the Origin, the minimum and maximum evaluation values are 31701 migrations and 65762 migrations respectively. For proposed algorithm, the values are 17607 migrations and 45512 migrations respectively as in figure1.2(b). Compared with the Origin, the proposed algorithm has 44.45%~73.22% decrease for migrations for the 16-type combinations respectively.

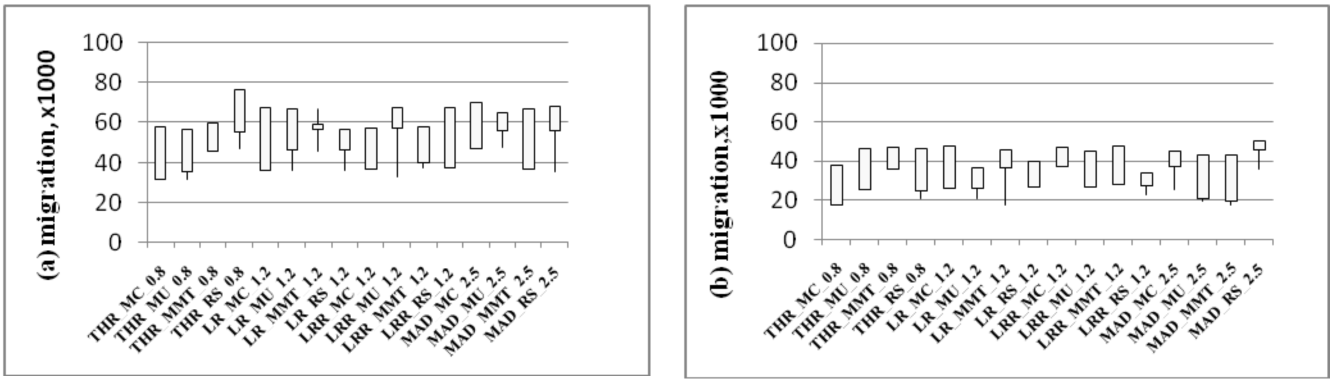


Figure 1.2 The Migrations evaluation

3) The Energy evaluation: In figure 1.3(a), for the Origin, the minimum and maximum evaluation values are 141.933 kWh and 171.444 kWh respectively. In figure 1.3(b) for the proposed algorithm, the values are 125.03 kWh and 128.29 kWh respectively. Compared with the Origin, the proposed algorithm has 11.9%~27.07%decrease for energy consumption for the 16-type combinations respectively.

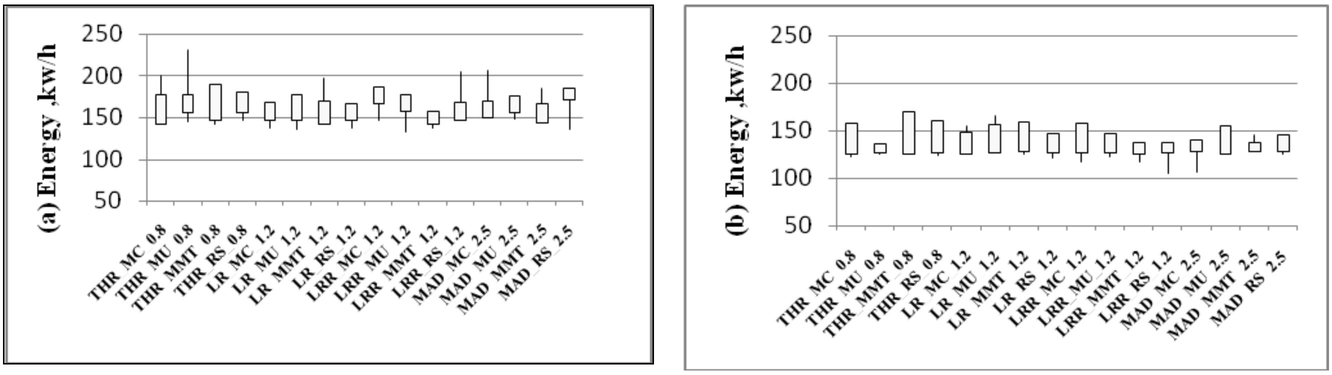


Figure 1.3 The Energy evaluation

4) The PDM evaluation: In figure 1.4(a), for the Origin, the minimum and maximum evaluation values are 0.126% and 0.194% respectively. For the proposed algorithm, the values are 0.09% and 0.15% respectively as in figure 1.4(b). Compared with the Origin, the proposed algorithm has 28.57%~53.6% decrease for the 16-type combinations for the PDM respectively.

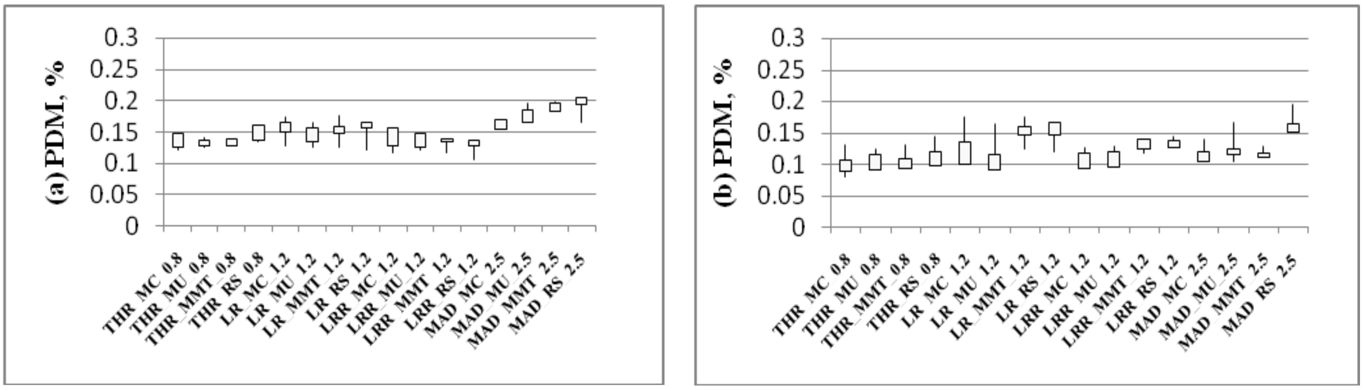


Figure 1.4 The PDM evaluation

5) The SLAV evaluation: In figure 1.5(a), for the Origin, the minimum and maximum evaluation values are 0.707% and 1.379%respectively. In figure 1.5(b) for the proposed algorithm, the values are 0.137% and 0.598% respectively. Compared with the Origin, the proposed algorithm has 80.62%~90% decrease for the 16-type combinations for the SLAV respectively.

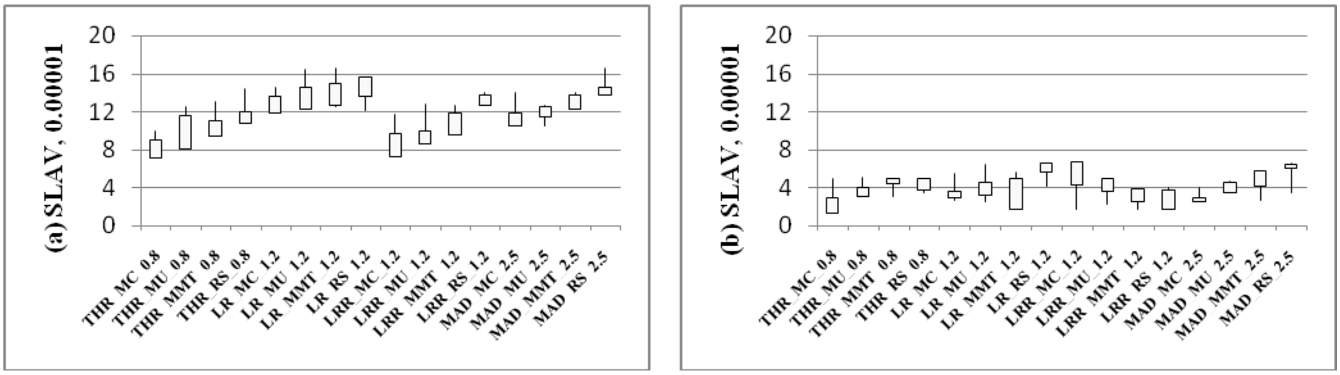


Figure 1.5 The SLAV evaluation

6) The ESV evaluation: In figure 1.6(a), for the Origin, the minimum and maximum evaluation values are 10.06% and 23.25% respectively .In figure1.6(b) for the proposed algorithm, the values are 2.87% and 6.98% respectively. Compared with the Origin, the proposed algorithm has 72.92%~87.65% decrease for the 16-type combinations for the ESV respectively.

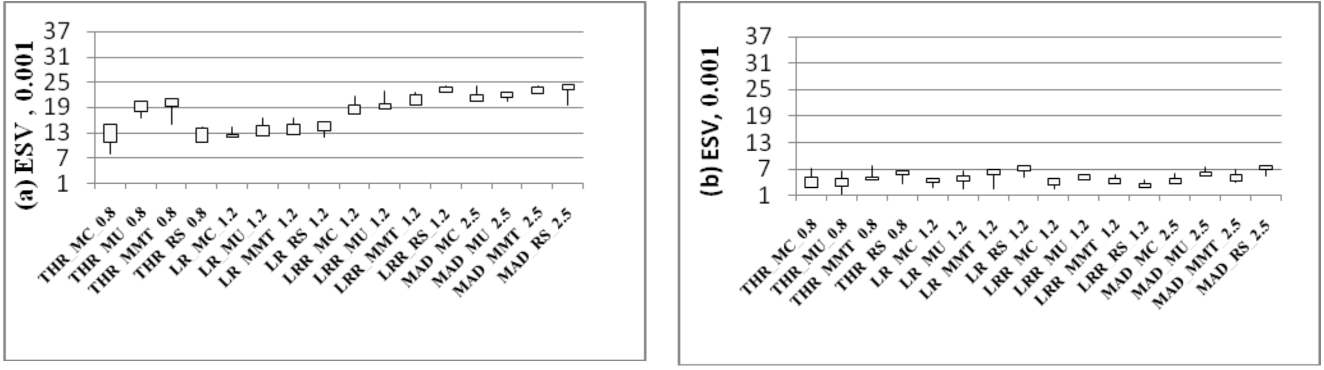


Figure 1.6 The ESV evaluation

7) The ETF evaluation: As we have 10-day workload and the 16-type combination policies. We record the wall-clock time of the scripts for each framework and repeat the experiment 10 times as shown in Table II. For the Origin, the ETF is 42,185 s on average. Compared with the Origin,the proposed algorithm's ETF is 45,403 s on average and obtains an increase of 7.62 %

Table II ETF evaluation

Experiment number	The ETF evaluation(s)	
	origin	Proposed
1	42218	45111
2	42161	45103
3	42208	45672
4	42174	45231
5	42229	45987
6	42305	45339
7	42149	45125
8	41984	45543
9	42253	45612
10	42172	45309
Average	42,185	45,403

In a nutshell, the proposed consolidation algorithm outperform the origin for the all above metrics. But worse for ETF.

D. *Simulators*

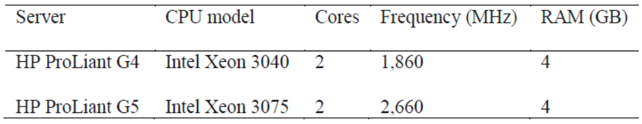
CloudSim offers the following novel features: (i)support for modeling and simulation of large scale Cloud computing infrastructure, including data centers on a single physical computing node; and (ii) a self-contained platform for modeling data centers, service brokers, scheduling, and allocations policies. Among the unique features of CloudSim, there are: (i) availability of virtualization engine, which aids in creation and management of multiple, independent, and co-hosted virtualized services on a datacenter node; and (ii) flexibility to switch between space shared and time-shared allocation of processing cores to virtualized services. These compelling features of CloudSim would speed up the development of new resource allocation policies and scheduling algorithms for Cloud computing.[16]

VII. SYSTEM MODEL

A.CPU architecture and power model

In this paper, it is assumed that each CPU has c cores and each core has m MIPS, so total MIPS of a CPU is $c \times m$. The servers used in this paper are: HP ProLiant ML110 G4, HP ProLiant ML110 G5. Tables III and IV show the configuration and power consumption, respectively.

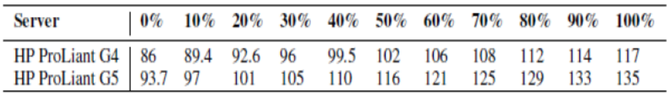
Table III configuration of servers



Server	CPU model	Cores	Frequency (MHz)	RAM (GB)
HP ProLiant G4	Intel Xeon 3040	2	1,860	4
HP ProLiant G5	Intel Xeon 3075	2	2,660	4

The reason why we have not chosen servers with more cores is that it is important to simulate a large number of servers to evaluate the effect of VM consolidation. Thus, simulating less powerful CPUs is advantageous, as less workload is required to overload a server. Nevertheless, dual-core CPUs are sufficient to evaluate resource management algorithms designed for multi-core CPU architectures.

Table IV Power consumption of server 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

VIII. Conclusion

In this paper, we have proposed an energy-efficient algorithm for VM consolidation which can reduce energy consumption and at the same time the SLAV. The simulation show that the proposed algorithms significantly reduce number of VM migration, SLAV in comparison with current algorithms[13].

We have evaluated proposed algorithm and the Origin framework through simulation on large-scale experiments driven by workload traces collected from more than a thousand PlanetLab VMs, and the results show that our proposed algorithm gets a better energy-performance. It guarantees 11.9–27.07 % decrease in energy consumption, 80.62–90 % decrease in SLA violation, 72–87%decrease in energy-performance metric, but 7% increase in execution time.

References

1. A. Carpen-Amarie, A.-C. Orgerie, and C. Morin, “Experimenta Study on the Energy Consumption in IaaS Cloud Environments,” *2013 IEEE/ACM 6th International Conference on Utility Cloud Computing*, pp. 42–49, Dec. 2013.
2. I. S. Moreno and J. Xu, “Customer-Aware Resource Overallocation to Improve Energy Efficiency in Real- Time Cloud Computing Data Centers.”
3. Z. Wang and Y.-Q. Zhang, “Energy-Efficient Task Scheduling Algorithms with Human Intelligence Based Task Shuffling and Task Relocation,” *2011 IEEE/ACM Int. Conf. Green Comput. Commun.*, pp. 38–43, Aug. 2011.
4. C.-C. Lin, P. Liu, and J.-J. Wu, “Energy-efficient Virtual Machine Provision Algorithms for Cloud Systems,” *2011 Fourth IEEE Int. Conf. Util. Cloud Comput.*, pp. 81–88, Dec. 2011.
5. Nathuji R, Schwan K. Virtual power: Coordinated power management in virtualized enterprise systems. ACMSIGOPS Operating Systems Review 2007; 41(6):265–278.
6. A. Verma, P. Ahuja, A. Neogi, pMapper: Power and migration cost aware application placement in virtualized systems,in: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, Springer, 2008, pp. 243-264.
7. Verma A, Dasgupta G, Nayak TK, De P, Kothari R (2009) Server workload analysis for power minimization using consolidation. In: Proceedings of the 2009 conference on USENIX annual technical conference. San Diego, USA, pp 28–28
8. Lee YC, Zomaya AY (2010) Energy efficient utilization of resources in cloud computing systems. J Supercomput 60(2):268– 280
9. S. Srikantaiah, A. Kansal , F. Zhao, Energy aware consolidation for cloud computing, Proceedings of the 2008 conference on Power aware computing and systems, San Diego, California, 2008, pp.10-10.
10. J. Xu, J.A.B. Fortes, Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments, 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing, Hangzhou, China, 2010, pp.179-188.
11. A. Beloglazov and R. Buyya, “Energy efficient allocation of virtual machines in cloud data centers,” in *10th IEEE/ACM Int. Symp. on Cluster Computing and the Grid, CCGrid 2010*, Melbourne, Australia,May 2010, pp. 577–578.
12. A. Beloglazov, J. Abawajy, R. Buyya, Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing, Future Generation Computer Systems (FGCS),Elsevier Science, Amsterdam, May 2012, 28(5): 755-768.
13. Beloglazov A, Buyya R (2011) 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
14. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. 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.
15. K.S. Park, V.S. Pai, CoMon: a mostly-scalable monitoring system for PlanetLab.ACM SIGOPS Operating SystemsReview2006,40(1):74.
16. Buyya R, Ranjan R, Calheiros RN (2009) Modeling andsimulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities. In: Proceedings of high performance computing and simulation. Kingston, USA, pp 1–11