

Growth Potential Aware Virtual Machine Consolidation Framework

Swasthi Shetty ¹ and Annappa B ²

¹Independent Researcher

²Affiliation not available

October 30, 2023

Abstract

Virtual machine consolidation techniques provide ways to save energy and cost in cloud data centers. However, aggressive packing of virtual machines can cause performance degradation. Therefore, it is essential to strike a trade-off between energy and performance in data centers. Achieving this trade-off has been an active research area in recent years. In this paper, a host underload detection algorithm and a new VM selection and VM placement techniques are proposed to consolidate Virtual machines based on the growth potential of VMs. Growth potential is calculated based on the utilization history of VMs. The interdependence of VM selection and VM placement techniques are also studied in the proposed model. The proposed algorithms are evaluated on real- world PlanetLab workload on Cloudsim. The experimental evaluation shows that our proposed technique reduces Service Level Agreement Violation (SLAV) and energy consumption compared to the existing algorithms.

Growth Potential Aware Virtual Machine Consolidation Framework

Swasthi Shetty C S and B Annappa (Senior Member, IEEE)

Abstract—Virtual machine consolidation techniques provide ways to save energy and cost in cloud data centers. However, aggressive packing of virtual machines can cause performance degradation. Therefore, it is essential to strike a trade-off between energy and performance in data centers. Achieving this trade-off has been an active research area in recent years. In this paper, a host underload detection algorithm and a new VM selection and VM placement techniques are proposed to consolidate Virtual machines based on the growth potential of VMs. Growth potential is calculated based on the utilization history of VMs. The interdependence of VM selection and VM placement techniques are also studied in the proposed model. The proposed algorithms are evaluated on real-world PlanetLab workload on Cloudsim. The experimental evaluation shows that our proposed technique reduces Service Level Agreement Violation (SLAV) and energy consumption compared to the existing algorithms.

Index Terms— Virtual Machine Placement, VM consolidation, SLA Violation, Power Consumption, PlanetLab workload, Data Center

I. INTRODUCTION

CLOUD computing delivers infrastructure, software and platform as services to the users in pay as you go model in the form of infrastructure as a service (IaaS), software as a service (SaaS) and platform as a service (PaaS). As users pay for these services, they expect high quality of service from cloud providers. Cloud providers can improve the quality of services by providing users with the desired QoS. It is defined in terms of Service level agreements (SLA) and providers are expected to pay a penalty if they do not meet expected SLA. This is the reason why providers need to reduce SLA violation. At the same time, providers are benefitted if they can reduce power consumption in data center which in turn would reduce the operational cost. Data centers consist of a large number of computational nodes, which consume large amount of power. Data centers have high economic impact due to several reasons. Energy costs for powering up a typical data center double every 5 years [1]. In some cases, power costs are more than the cost of the purchased hardware. Hence it has become a necessity to reduce data center power consumption and costs [2].

Virtual Machine(VM) consolidation method is a powerful way to reduce power consumption and to achieve QoS

mentioned above. VM consolidation packs the VMs on Physical Machines(PM) based on certain characteristics and usage. This in turn allows some of the PMs experiencing lesser workload to be shut down in the data center, thereby reducing power consumption of the data center [3]. Servers in data centers utilize only 10-50% of resources [4]. So shutting down PMs that are not being utilized completely will save power. But this can also cause overloading in hosts, resulting in an increase in SLAV. Hence there is a need to solve this problem with an efficient way of handling both power consumption and SLAV.

This paper presents a model to optimize Virtual machine allocation on physical machines in order to reduce power consumption and minimize SLAV. Even when Dynamic Voltage and Frequency Scaling is considered, there is a relationship between power consumption and CPU utilization in servers [5,6]. So this paper focuses on CPU utilization to reduce power consumption. VM migration is one of the major reasons for SLAV to occur because it involves migration of memory pages and during this transition state, requested CPU cannot be delivered to the applications running on the VMs [7]. Hence the need to reduce the number of VM migrations along with power consumption and SLA violation to provide quality service to the users.

The main contributions of this paper are as follows:

- 1) Growth Potential Aware VM placement algorithm (GPAP) is proposed to perform VM placement to the appropriate physical machines.
- 2) Growth Potential Aware VM selection (GPS) algorithm is proposed to select Virtual machines for migration from highly loaded servers.
- 3) Under Utilized servers are detected considering utilization, number of VMs and Power consumption.
- 4) Evaluation of the proposed techniques using real world planetlab workload traces and comparison of results with the existing methods to prove the effectiveness of the proposed solution.

The rest of the paper is organized as follows: In the next section, related works of other researchers are discussed. In section 3, this paper discusses the problem statement and proposed solution. In the performance analysis section, performance metrics have been discussed and simulations are carried out using CloudSim for the proposed model and results are compared with the existing methods. In the final section,

this paper has discussed conclusions and future work and concluded the paper.

II. RELATED WORKS

Many VM consolidation algorithms have been proposed by researchers in cloud computing to reduce power consumption and minimize SLAV. Reduction of operational cost in data center while providing desired QoS to the users is an active research area and many techniques have been suggested through different VM allocation, VM selection and underload/overload detection mechanisms. In the following section, some of the works carried out in recent times have been discussed.

Ali Nadjar et al [8]. introduced combinatorial VM placement techniques considering load dispersion of all the hosts. VM placement strategy was dynamically arrived at, by calculating interquartile range, diving the set into four groups. The algorithm first tries to place VMs in the highly loaded region. In case of failure to accommodate all the VMs in the highly loaded servers, it falls back to the basic mechanism of placing VMs on the hosts. Since the host overloading was being detected dynamically, this proved to be more efficient compared to static threshold overloading detection techniques.

Abbas Horri et al [9]. proposed a VM placement technique that takes into account utilization of hosts and minimum correlation. Minimum correlation was based on the earlier work done by Verma et al [10] to select VM for migration with the highest correlation. This idea was used by Abbas Horri et al to modify VM placement technique to place VMs that has the least correlation with the PM.

Ruan and H. Chen [11] investigated the problem of VM consolidation and came up with a novel technique to consider performance to power ratio of the hosts. This was one of the first works to consider both performance and power to provide a balance between the two. They also evaluated on CloudSim to prove the effectiveness of their solution.

Kulkarni and B. Annappa [12] developed a context aware VM placement algorithm where load context in the data center is taken into consideration. Their work prevents redundant detection of underloaded hosts when data center load traffic is at its peak to avoid host shutdowns during the same. Also, their work considers performance to power ratio at different load levels and calculates the average P2P ratio for VM placement.

Zhibo Cao et al. [13] studied VM placement problem and introduced a modified placement strategy to the already existing Minimum Power VM placement technique. They proposed Minimum power and Maximum Utilization strategy where minimum and maximum utilization request VMs are selected alternatively for VM placement to not overload highly loaded servers. They also evaluated their technique using simulation and proved that their solution reduced energy consumption by 34% compared to the Minimum Power policy.

Monil and R. Rahman [14] proposed a technique by modifying overload detection and VM selection algorithm. They considered CPU utilization values of last n time frames and calculated the average of those values to select VM for migration. Also, for overload detection algorithm mean and standard deviation values were calculated to check if the host was overloaded.

There are many such techniques that were investigated in the past to reduce power consumption and SLAV. But the main drawback among all the techniques discussed above was they all considered SLAV and energy consumption, but at the same time it is important to consider number of VM migrations as the increase in migration would cause performance degradation as well as increase in operational cost.

III. PROPOSED SYSTEM MODEL

This paper proposes a new VM selection algorithm and a new VM placement algorithm by taking into consideration the potential growth in utilization of CPU. This research uses already existing Interquartile Range method for detection of overloaded servers. A modified underload detection algorithm has been implemented for reduction in number of VM migrations.

Currently, VM consolidation techniques aim to provide high performance while meeting SLAs and reducing power consumption. At the same time, it is also necessary to reduce the number of VM migrations while carrying out VM consolidation. VM migration effects performance of applications running on virtual machines. The proposed approach of VM selection and VM placement takes care of minimizing virtual machine migration along with reduction in power consumption and SLAV.

A. VM Selection Algorithm

VM selection algorithms choose one or more VMs from the set of VMs on the overloaded servers for migration. There are 3 built in VM selection methods on CloudSim proposed in research [15-17].

- 1) Minimum migration time (MMT) which selects VM that takes the least amount of time to migrate from source PM to destination PM.
- 2) Random selection (RS) which randomly selects VM for migration.
- 3) Maximum Correlation (MC) that selects VM which has the highest correlation with other VMs in the overloaded server.

This paper proposes a new VM selection method that takes into consideration - mean CPU utilization and current CPU utilization of VM. VM selection and VM placement algorithms are not considered in isolation. This method is dependent on the specific VM placement algorithm that is explained in the next section. VM selection is more efficient when it has knowledge on VM placement techniques [15-18]. The number of VM migrations can also be reduced if more details on VM placement are known to VM selection algorithms [16-19].

Mean utilization of VMs is calculated using the equation (1). If mean utilization is higher than the current utilization, then there is a high possibility that the VM will overload the server in future. So the VMs with greater mean value is selected for migration to a medium loaded server. This algorithm calculates the difference in mean utilization and current utilization and selects the VM that has the highest difference when the mean utilization is greater than the current utilization. The reason for selecting this VM is to migrate the VM to the lesser loaded servers from the highly loaded servers because these VMs have higher probability of overloading servers. In order to prevent these VMs from overloading the server again, these VMs are chosen for migration to lesser loaded servers. Along with this, algorithm needs to take care of not increasing the number of VM migrations. So the algorithm does not blindly consider the difference, but also tries to balance this out with the current CPU utilization as formalized in equation (4).

Let x be the VM and $U_z(x_t)$ be the CPU utilization of VM x at time t . The mean utilization (M_u) is calculated for the values of last n time frames using equation (1):

$$M_u = \frac{U_z(x_t) + U_z(x_{t-1}) + \dots + U_z(x_{t-n})}{(n + 1)} \quad (1)$$

Difference in mean utilization (M_u) and current utilization (C_u) is calculated using the equation (2) and a novel technique is proposed to compute utilization defined by utilization difference of VM using equation (3) and (4)

$$Vdiff = |M_u - C_u| \quad (2)$$

if $M_u > C_u$:

$$U_{t_m} = M_u + diff \quad (3)$$

if $C_u > M_u$:

$$U_{t_c} = C_u + diff \quad (4)$$

The VM with the maximum value $U_{t_{final}}$ calculated using equation (5) will be chosen for VM selection. This will ensure that VM with higher current utilization along with a higher probability to overload the server in future, is selected.

$$U_{t_{final}} = \begin{cases} \sum_{v \in VM} U_{t_m} \rightarrow \max M_u > C_u \\ \sum_{v \in VM} U_{t_c} \rightarrow \min M_u < C_u \end{cases} \quad (5)$$

The pseudo code for the Growth Potential Aware VM selection method is presented in Algorithm 1. The algorithm takes VM list as input and selected VM for migration is returned as output. For every VM, mean CPU utilization is calculated for the CPU utilization values of last n time frames and the difference between mean and current utilization is calculated. If mean value is greater than the current utilization then the calculation is carried out as per equation (3). If current value is

greater than mean the calculation is carried out as per formula (4). Finally, maximum of U_{t_m} and minimum of U_{t_c} among all the VMs is considered and maximum among those values will be returned as selected VM.

Algorithm 1: Growth Potential Aware VM selection

Input: VMList **Output:** selected_VM

```

selected_VM ← NULL
max_util ← MIN
min_util ← MAX
foreach VM in VMList do
  mean_util ← VM.get_mean_utilization
  current_util ← VM.get_TotalUtilization_Of_CPU
  if (mean_util > current_util)
    vdiff ← mean_util - current_util
    utilization ← mean_util + vdiff
    if utilization > max_util and min_util not MAX then
      max_util ← utilization;
      if max_util > min_util then
        VM.set_utilization_difference(vdiff)
        selected_VM ← VM
  else
    utilization ← current_util + (current_util - mean_util)
    if utilization < min_util then
      min_util ← utilization
      if (max_util < min_util) then
        selected_VM ← VM
return selected_VM

```

B. VM Placement Algorithm

VM placement algorithm is used to allocate a VM to a particular PM. Online VM placement problem can be considered as a variant of multi-dimensional bin packing problem [20].

The proposed VM placement algorithm presented in Algorithm 2 considers the difference in mean and current CPU utilization calculated in Algorithm 1. Every VM that has higher mean utilization than the current CPU utilization will be associated with a difference value (V_{diff}), calculated using equation (2). VMs are then sorted in the increasing order of the difference value (V_{diff}) that is set in algorithm 1. The reason to sort it in increasing order of the difference value is to ensure VMs with higher difference value are placed on the lightly loaded servers as they have higher probability of overloading the server in the near future. The VMs with current utilization equal to or higher than mean utilization will not have V_{diff} value. They will be sorted in decreasing order of current CPU utilization as they have lower probability of overloading the server. These VMs will be placed on highly loaded servers given they have enough resources to host the VM.

To avoid overloading of highly loaded servers, this paper considers a safety parameter for CPU utilization. Any Physical Machine with utilization that exceeds the safety parameter will not be considered for VM allocation. This is to avoid aggressive packing of VMs to prevent overloading of servers.

Algorithm 2 presents Growth Aware VM placement mechanism where PMs and VMs are passed as input and mapping of VMs to PMs is returned as output. Algorithm 1 sets a difference value V_{diff} for every VM with mean utilization greater than the current utilization. VMs with V_{diff} value are sorted in ascending order of V_{diff} and VMs without V_{diff} value are sorted in descending order of current CPU utilization. VMs with higher current utilization will be placed on hosts with maximum utilization but not more than the safety parameter, after checking for enough resource availability on the hosts. VMs with V_{diff} will be placed on the lesser loaded servers as they have higher probability of overloading the servers.

Algorithm 2: Growth Potential Aware VM placement

Input: hostList, VMList **Output:** allocation of VMs

```

safety_param ← 0.7
max_util ← MIN
foreach VM in vmList do
  if (VM.getDifference() > 0) then
    VMsToMigrate_diff.add(vm)
  else
    VMsToMigrate_non_diff.add(vm)
sortByCPUUtilization(VMsToMigrate_non_diff)
sortByUtilizationDifference(VMsToMigrate_diff)
VMList ← VMsToMigrate_non_diff + VMsToMigrate_diff
foreach VM in VMList do
  allocatedHost ← NULL
  foreach host in hostList do
    host_utilization ← host.getUtilizationOfCPU();
    if host has enough resource for VM then
      if host_utilization > max_util then
        if host_utilization < safety_param then
          max_util ← host_utilization
          allocatedHost ← host
      if allocatedHost ≠ NULL then
        allocate VM to allocatedHost
return allocation

```

Algorithm 3: Host Underload Detection

Input : hostList **Output:** underloadedHost

```

/* Initialize to static threshold value for under utilization check */
maxP2Nratio ← MIN
Utilization_min ← LOWER_THRESHOLD
/* Select power inefficient Host with lower than threshold CPU
utilization */
foreach host ∈ hostList do
  utilization ← host.getUtilizationOfCPU();
  if (utilization < Utilization_min) then
    power ← host.getPower()
    num_VMs ← host.getVMList().size();
    P2Nratio ← power/num_VMs
    if P2Nratio < maxP2Nratio then
      underUtilizedHost ← pm
      maxP2Nratio ← P2Nratio
return underloadedHost

```

C. Host underload detection

This paper proposes an underload detection algorithm to detect PMs taking into account the number of VMs, Power

consumption and CPU utilization. A static CPU utilization threshold value is set and among the PMs with CPU utilization lower than the static threshold value, the ratio (U_H) of power consumption to the number of VMs on the PMs is calculated as per the equation (6). The PM with the maximum U_H is chosen as underloaded host and is shut down after migrating all the VMs hosted on that particular PM. Most of the existing algorithms so far have considered power consumption and CPU utilization for host underload detection. The proposed algorithm also considers number of VMs located on the host to ensure the host selected has lesser number of VMs as all the VMs on the selected host has to be migrated.

$$U_H = P / N_{vms} \quad (6)$$

where P is the power consumption of the PM and N_{vms} is the number of VMs on the PM.

Algorithm 3 presents the proposed host underload detection algorithm.

IV. PERFORMANCE EVALUATION

A. Experimental Setup

Proposed algorithm has been implemented using CloudSim Toolkit [21] and results are compared and analyzed with the built-in methods. Real world workload traces of PlanetLab [22] are taken for this experiment. It consists of 800 Physical Machines and 898 Virtual machines. Out of 800 PMs, 400 are HP ProLiant G4 and the rest are HP ProLiant G5 servers. Energy consumption of these physical machines are calculated based on the values given in table 1.

Table 1. Power consumption based on CPU utilization

Utilization						
Power (Watt)	0%	20%	40%	60%	80%	100%
HP G4	86	92.6	99	106	112	117
HP G5	93.7	101	110	121	129	135

Table 2. Configuration of Virtual Machines on CloudSim

Serial No	VM Type	[CPU_MIPS,num_cores, RAM_in_MB,VM_Size_in_GB]
1	Type 1	[2500,1,870,2.5]
2	Type 2	[2000,1,1740,2.5]
3	Type 3	[1000,1,1740,2.5]
4	Type 4	[500,1,613,2.5]

Table 3. Performance evaluation result

Type	Energy (kWh)	VM migrations	SLATAH (%)	SLAV (%)	ESV	PDM
GPAP_GPS	119.08	3699	1.63	0.000489	0.05823012	0.03
IQR_MC	142.36	19004	6.79	0.00679	0.9666244	0.1
IQR_MMT	152.81	21953	4.98	0.003486	0.53269566	0.07
IQR_RS	143.18	19488	6.91	0.007601	1.08831118	0.11
LR_MMT	133.64	22359	5.86	0.004688	0.62650432	0.08
LR_RS	120.3	17370	6.93	0.00693	0.833679	0.1
LRR_MMT	133.64	22359	5.86	0.004688	0.62650432	0.08
LRR_RS	120.97	17790	6.91	0.00691	0.8359027	0.1
MAD_MC	140.04	19303	6.92	0.007612	1.06598448	0.11
MAD_MMT	149.08	21902	5.07	0.003549	0.52908492	0.07
MAD_RS	140.3	19762	7.03	0.007733	1.0849399	0.11
THR_MC	145.82	19257	6.78	0.007458	1.08752556	0.11
THR_MMT	155.27	22303	4.99	0.003493	0.54235811	0.07
THR_RS	148.14	19846	6.81	0.00681	1.0088334	0.1

These PMs are assigned with 1860MIPS and 2660 MIPS (Million Instructions Per Second) for each core of G4 and G5 servers. Network bandwidth is considered as 1GB/s. Created VMs were single core. VMs are assigned different types that are mentioned in table 2 [16] The real world PlanetLab data contains CPU utilization calculated for every 5 minute intervals for all the 898 VMs that are located on the PMs around the world [9]. The workload is representative of an IaaS cloud environment such as Amazon EC2, where users create and manage VMs independently [10].

B. Performance Metrics

This paper takes into consideration, the metrics presented in table 3, to evaluate and compare the proposed algorithm with the existing methods.

- **Energy Consumption:** This is the most important metric in VM consolidation as the one of the most important purposes of VM placement is to reduce energy consumption. Energy consumption is calculated in the simulation based on the utilization as given in table I. CPU utilization is measured at fixed intervals. Power

consumption is calculated for all the hosts based on the CPU utilization as shown in table 1.

- **SLAV:** Cloud providers and customers agree upon certain Service level agreement(SLA) in the beginning. When the providers fail to deliver expected SLA, SLA Violation will occur. This is calculated based on two independent metrics, SLA Time per Active Host(SLATAH) and Performance degradation due to migration (PDM). SLAV is calculated using the formula,

$$SLAV = SLATAH \times PDM \quad (7)$$

- **SLA Time per Active Host(SLATAH):** Time during which host is experiencing 100% CPU utilization is measure using SLATAH metric. This is the fraction of time when host cannot deliver SLA and hence SLA violation occurs.

$$SLATAH = \frac{1}{N} \sum_{k=1}^N \frac{T_{uk}}{T_{hk}} \quad (8)$$

Where N is the number of Hosts, T_{uk} is the total time during which host k experienced 100% CPU utilization, T_{hk} is the total active time of host k .

- **Energy and SLA Violation (ESV):** It is necessary to reduce Energy consumption as well as minimize SLA violation. So, the metric ESV is used to measure both energy consumption and SLAV. This metric is used to get holistic view of the proposed algorithm. ESV is calculated using the equation (9)

$$ESV = Energy \times SLAV \quad (9)$$

- **Number of VM migrations:** This metric is used to calculate total number of VM migrations occurred during simulation. VM migration is an important metric as unwanted migration causes SLA Violation and network traffic.

C. Experimental Result

Table 3 presents the simulation results of the proposed algorithm and compares it with the other existing algorithms. There are 5 different VM placement techniques – IQR, LR, LRR, MAD, THR and 3 VM selection methods – MC, MMT, RS [17]. The combination of these policies is considered for comparison with the proposed Growth Potential Aware Placement (GPAP) and Growth Potential Aware VM selection (GPS) algorithm.

The simulation results show that the proposed algorithms have outperformed the existing algorithms in all the metrics mentioned above. Bar graphs are plotted for all the metrics in Figs 1, 2, 3, 4 and 5. From figure 1, it can be seen that the proposed algorithm GPAP-GPS has lesser energy consumption compared to other algorithms. From figure 2, it can be seen that the proposed algorithm decreased SLA violation significantly. Minimum SLAV of all the other methods was around 0.003% whereas SLAV for GPAP-GPS is as low as 0.0004 % resulting in reduction of 83%. Reduction in SLAV is the significant contribution of the proposed research. In figure 3, graph is plotted based on ESV which is obvious to have reduced significantly as it is calculated based on Energy Consumption and SLAV. It proves that both the energy and SLA tradeoff has been achieved. Figure 3 shows that the minimum ESV among all the other algorithms is 0.52 and ESV of the proposed method is 0.05 resulting in 90 % reduction of ESV. Figure 4 presents a graph for number of VM migrations, and it can be inferred that the minimum number VM migrations among all the existing algorithms is 17,370 and for proposed method is 3699 resulting in 78% reduction.

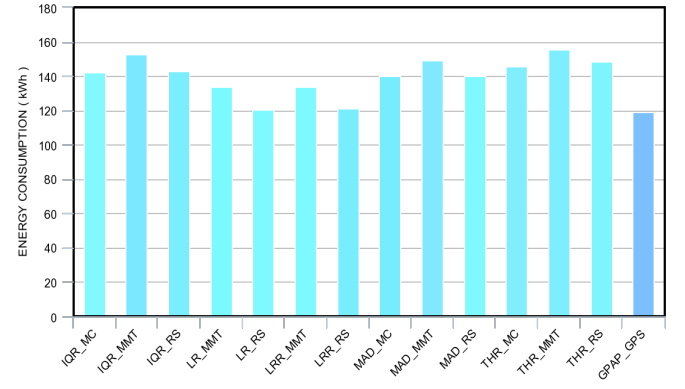


Figure 1. Energy consumption result

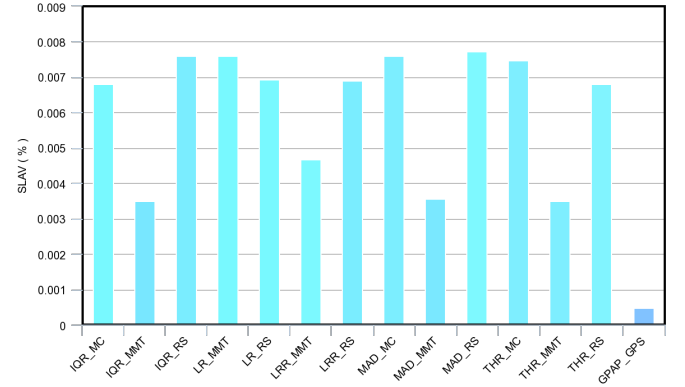


Figure 2. SLA violation result

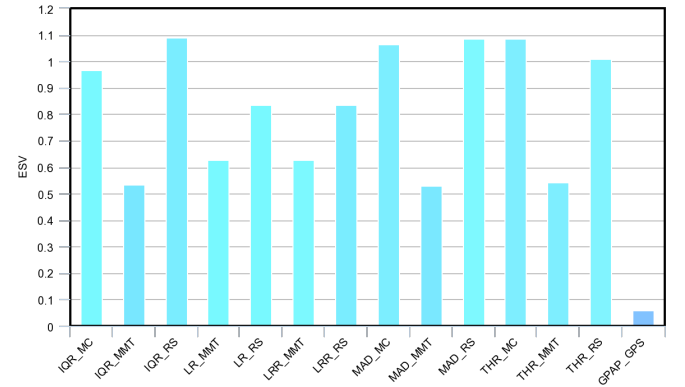


Figure 3. ESV

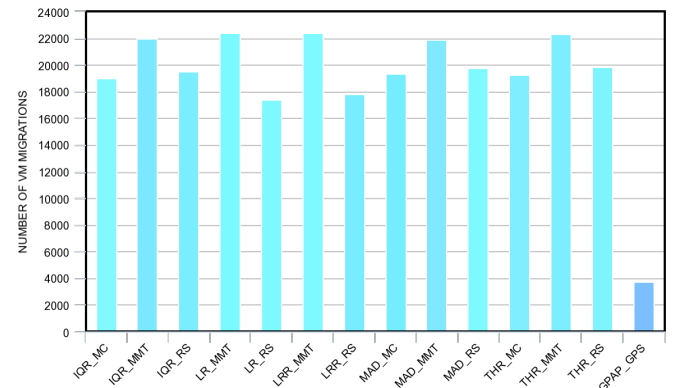


Figure 4. Number of VM migrations

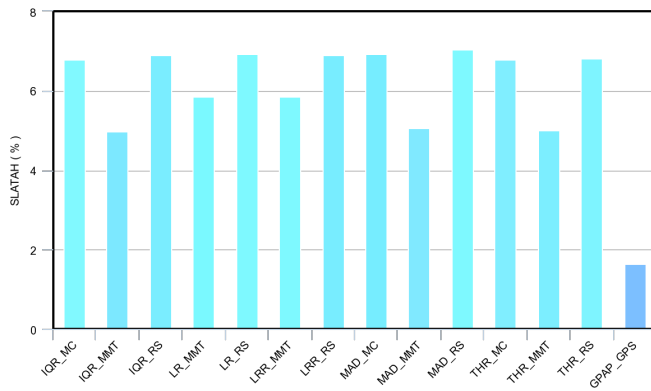


Figure 5. SLA Time per Active Host

From the comparison made above, it can be deduced that the proposed algorithm (GPAP-GPS) outperforms all the other existing algorithms significantly.

V. CONCLUSION AND FUTURE WORKS

This paper presents Growth potential aware VM placement and selection algorithms and a modified underload detection algorithm to reduce energy consumption and decrease SLA violation. Cloud providers have to reduce energy consumption to reduce data center cost and minimize SLA violation to avoid penalty from users. The main objective of this research is to take care of both energy consumption reduction and SLA violation while reducing the number of VM migrations which otherwise causes network traffic in data centers. Evaluations were conducted for the proposed technique against the existing methods in a real workload scenario and proposed scheme has been proved to outperform the current methods. The idea of the proposed technique was to migrate VMs that has a high probability of overloading the server, to avoid the overloading of servers again in the future. The plan is to enhance the algorithms further to take into consideration RAM and network as they are also critical resources to be considered along with CPU utilization.

VI. REFERENCES

- [1] R. Buyya, C. Vecchiola, and S. Selvi, "Mastering Cloud Computing" in Foundations and Applications Programming. Elsevier Inc., 2013.
- [2] M. Dayarathna, Y. Wen and R. Fan, "Data Center Energy Consumption Modeling: A Survey," in IEEE Communications Surveys & Tutorials, vol. 18, Firstquarter 2016. doi: 10.1109/COMST.2015.2481183
- [3] A. Varasteh and M. Goudarzi, "Server Consolidation Techniques in Virtualized Data Centers: A Survey," in IEEE Systems Journal, vol. 11, no. 2, June 2017. doi: 10.1109/JSYST.2015.2458273
- [4] Awada, Uchechukwu & Li, Keqiu & Shen, Yanming. "Energy Consumption in Cloud Computing Data Centers" International Journal of Cloud Computing and Services Science (IJ-CLOSER), 2014 doi: 10.11591/closer.v3i3.6346.
- [5] Yavari, M., Ghaffarpour Rahbar, A. & Fathi, M. J Cloud Comp , "Temperature and energy-aware consolidation algorithms in cloud computing", 2019 doi: 10.1186/s13677-019-0136-9
- [6] Arroba, P., Moya, J. M., Ayala, J. L., & Buyya, R. (2016). "Dynamic Voltage and Frequency Scaling-aware dynamic

consolidation of virtual machines for energy efficient cloud data centers", Concurrency and Computation: Practice and Experience doi:10.1002/cpe.4067

[7] Mohammad Alaul Haque Monil and Rashedur M. Rahman, "VM consolidation approach based on heuristics, fuzzy logic, and migration control", in Proceedings with Journal of Cloud Computing, 2016.

[8] Nadjar, A., Abrishami, S. & Deldari, H. J Supercomput "Load dispersion-aware VM placement in favor of energy-performance tradeoff", 2017. doi: 10.1007/s11227-016-1842-4

[9] Horri, A., Mozafari, M.S. & Dastghaibiyfard, G. J Supercomputer, "Novel resource allocation algorithms to performance and energy efficiency in cloud computing" 2014 . doi: 10.1007/s11227-014-1224-8

[10] Verma A, Dasgupta G, Nayak TK, De P, Kothari R "Server workload analysis for power minimization using consolidation". In: Proceedings of the 2009 conference on USENIX annual technical conference. San Diego, USA, 2009

[11] X. Ruan and H. Chen, "Performance-to-Power Ratio Aware Virtual Machine (VM) Allocation in Energy-Efficient Clouds", 2015 IEEE International Conference on Cluster Computing, Chicago, IL, 2015

[12] A. K. Kulkarni and B. Annappa, "Context Aware VM Placement Optimization Technique for Heterogeneous IaaS Cloud," in IEEE Access, vol. 7, 2019.

doi: 10.1109/ACCESS.2019.2926291

[13] Cao, Z. & Dong, S. J Supercomput, "An energy-aware heuristic framework for virtual machine consolidation in Cloud computing", 2014 . doi: 10.1007/s11227-014-1172-3

[14] M. A. H. Monil and R. M. Rahman, "Implementation of modified overload detection technique with VM selection strategies based on heuristics and migration control", 2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS), Las Vegas, NV, 2015. doi: 10.1109/ICIS.2015.7166597

[15]. Beloglazov A, Abawajy J, Buyya R (2011) "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing". Futur Gener Comput Syst (FGCS), 2011

[16]. 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 (CCPE), 2012

[17]. Beloglazov A, (2013) PhD Thesis: "Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing". 2016

[18] Z. Á. Mann, "Resource Optimization Across the Cloud Stack", in IEEE Transactions on Parallel and Distributed Systems, vol. 29, 2018. doi: 10.1109/TPDS.2017.2744627

[19] Mann Z.Á. "Interplay of Virtual Machine Selection and Virtual Machine Placement". In: Aiello M., Johnsen E., Dustdar S., Georgievski I. (eds) Service-Oriented and Cloud Computing. ESOC 2016. Lecture Notes in Computer Science, vol 9846. Springer. 2016

[20]. Li X, Qian Z, Lu S, Wu J (2013) "Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in data center". Mathematical Computer Modeling, 2013

[21] R.N.Calheiros, R.Ranjan, A.Beloglazov, C.A.F.DeRose, and R.Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Softw., Pract. Exper., 2011. doi: 10.1002/spe.995.

[22] K. Park and V. S. Pai, "CoMon: A mostly-scalable monitoring system for PlanetLab," ACM SIGOPS Operating Systems, 2006.

[23] Ehsan Arianayan, Hassan Taheri, Saeed Sharifian, "Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data centers", Computers and Electrical Engineering, v.47, 2015



SWASTHI SHETTY CS received the B.E degree in Information Science and Engineering from BMS College of Engineering, Bangalore, India. She is currently working as Member of Technical Staff at Nutanix, India. Her research interests include cloud computing, distributed systems and networking.



B. ANNAPPA received the B.E. degree from the Govt. B.D.T. College of Engineering, Davangere affiliated Mysore University, Karnataka, and the M.Tech. and Ph.D. degrees in computer science and engineering from the National Institute of Technology Karnataka, Surathkal. He is currently a Professor with the Department of Computer Science and Engineering,

National Institute of Technology Karnataka, Surathkal, India. He has more than 25 years of experience in teaching and research. He has published more than 100 research papers in international conferences and journals. His research interests include cloud computing, big data analytics, distributed computing, software engineering, and process mining. He is a Fellow of the Institution of Engineers (INDIA). He is also a Life Member of the Computer Society of India, the Indian Society of Technical Education, the Cloud Computing Innovation Council of India and the Advanced Computing and Communications Society. He was the Organizing Chair of the International Conference ADCONS-2013. He was the Chair of IEEE Mangalore Subsection, in 2018. He is also the General Chair of DISCOVER 2010. He serves on the TPC of many international conferences and reviewer for several journals. He is also the Chair of the IEEE Computer Society Chapter, India Council. He was the Secretary of the IEI Mangaluru Local Center.