Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 3, March 2013, pg.74 – 82

RESEARCH ARTICLE

Energy Conscious Dynamic Provisioning of Virtual Machines using Adaptive Migration Thresholds in Cloud Data Center

Khushbu Maurya[1], Richa Sinha[2]
[1]Computer Department, Gujarat Technological University, India
[2]IT Department, Gujarat Technological University, India

[1]Khushbumaurya87@gmail.com, [2]richa.872005@gmail.com

Abstract— The fast growing demand for computational power utilized by modern applications with rapidly changing Cloud computing technology have directed to the foundation of large-scale virtualized data centers. Such data centers consume massive amounts of electrical energy resulting in high operating costs and carbon dioxide (CO2) emissions. Dynamic consolidation of virtual machines (VMs) using Dynamic migration and switching off idle nodes to the sleep mode provide better optimized resource usage, lower energy consumption, which provides high performance & better quality of service. However incompatibility between specification of physical machine and user requests in cloud, leads towards problems like poor load balancing, energy-performance trade-off and large power consumption etc. Also the VM placement should be optimized continuously in an online manner because of fast varying workloads in current application. To understand the inferences of the online behaviour of the problem, we conduct competitive analysis of optimal online deterministic & Adaptive Migration Thresholds based algorithms for the single VM migration and dynamic VM consolidation problem.

Concentrating at this issue, this paper presents an energy conscious, power aware load balancing strategy based on adaptive migration of virtual machines (VMs). This strategy will be applied to virtual machines on cloud, considering higher and lower thresholds for migration of virtual machines on the servers also here we consider RAM & Bandwidth for better performance & load balancing. If the load is greater or lower then defined upper & lower thresholds, VMs will be migrated respectively, boosting resource utilization of the cloud data center and reducing their energy consumption. To reduce number of migration we integrate minimum migration time policy which is capable of reducing the number of migration and the energy consumption of virtual machine migration also achieves load balancing and meet service level agreement (SLA) requirements. This document gives formatting instructions for authors preparing papers for publication in the Proceedings of an IEEE conference. The authors must follow the instructions given in the document for the papers to be published. You can use this document as both an instruction set and as a template into which you can type your own text.

Key Terms: - Cloud Computing, Virtual Machines, Reducing Energy, Dynamic Provisioning, Service Level Agreement (SLA), Adaptive Migration Thresholds, Resource Allocation.

I. INTRODUCTION

With the dawn of the Internet information age, user data increased greatly. The Cloud computing model influences virtualization of computing resources allowing customers to deliver on-demand resources on a pay-as-you-go basis [1]. Rather than obtaining high costs in buying IT infrastructure and dealing with the maintenance and upgrades of both software and hardware, organizations can outsource their computational requirements to the Cloud. The propagation of Cloud computing has given rise to the foundation of large-scale data centers containing thousands of computing nodes and consuming massive amounts of electrical energy. it has been estimated that by 2014 infrastructure and energy costs would contribute about 75%,whereas IT would contribute just 25% to the overall cost of operating a data center [2].

The reason behind this enormously high energy consumption is not just the quantity of computing resources and the power inefficiency of hardware, but it is because of inefficient usage of these resources. Data collected from more than 5000 servers over a six-month period have shown that although servers typically are not idle, the utilization hardly ever approaches 100% [3]. Moreover, maintaining over-provisioned resources outcomes in the higher Total Cost of Ownership (TCO).Second problem is the constricted dynamic power range of servers: even completely idle servers still consume about 70% of their peak power [4]. So, keeping servers underutilized is extremely ineffective from the energy consumption view. Moreover, high energy consumption by the organization moves towards significant carbon dioxide (CO2) emissions contributing to the greenhouse effect [5].

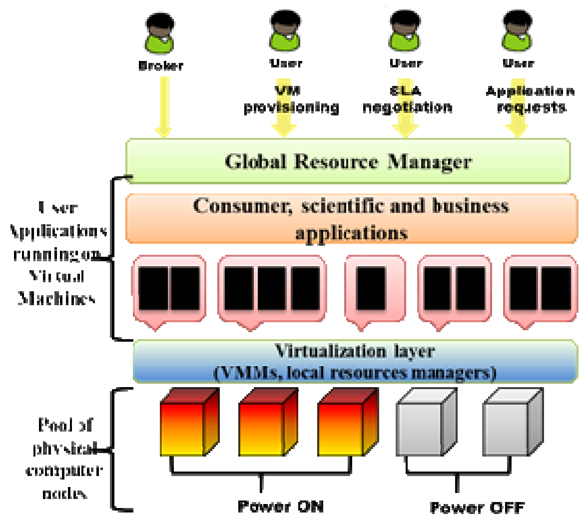


Figure 1: The System view

One of the techniques to report the energy inefficiency is to influence the competencies of the virtualization technology [6]. The virtualization technology permits Cloud providers to create multiple Virtual Machine (VMs) requests on a single physical server, hence increasing resources utilization & Return On Investment (ROI). Energy consumption can be reduced by switching idle nodes to low-power modes (i.e. sleep, hibernation), therefore eliminating the idle power consumption (Figure 1). Additionally, by using Dynamic migration [7] the VMs can be dynamically consolidated to the minimal number of physical nodes respected to their current resource needs. Yet, efficient resource management in Clouds is not trivial, as modern service applications often involve rapidly changing workloads resulting dynamic resource usage patterns. Hence, antagonistic consolidation of VMs can result into degraded performance, when applications have growing demand of resources, which results in an unpredicted rise of the resource usage & if they are not fulfilled, the application can suffer increased response times, time-outs or failures. Reliable Quality of Service (QoS) defined via Service Level Agreements (SLAs) should be established between Cloud providers and their customers is must for Cloud environments; thus, Cloud providers have to more concentrate on the energy-performance trade-off – the minimization of energy consumption, at the same time meet the SLAs. As the traditional static migration strategy causes unnecessary overheads of migrations [8]. Hence, to ensure user's tasks continue to run during migration process, to reduce the SLA violations of virtual machines, and to reduce the costs of power consumption that are caused by low workload resources, and such as lack of consideration of these aspects, paper proposes a resource scheduling strategy based on Adaptive migration Thresholds of VMs.

A. Adaptive Migration of VMs

Cloud computing architecture includes the application layer, the platform layer, virtualization and the infrastructure layer. A very important aspect of cloud computing different from grid computing is large-scale deployment of virtualized devices and virtualized environments [9]. In cloud computing, the cloud architecture adds a new layer - virtualization layer, as execution environment and hosting environment of cloud-based application. Cloud core hardware infrastructure consists of a datacentre for processing user service requests to model. In order to allow different levels of performance isolation to simulate different load balancing strategy, using the two virtual machine scheduling model: One is the host level, another is virtual machine-level [10]. The host level can specify in each core in a host processing power (MIPS) allocated to VMs, at virtual machine level it is in the amounts of processing power for a single task unit which hosts on waiting queue of the level.

The process of VM migration is that users first submit jobs to cloud computing environment on any one server, the system will automatically deploy user requests to the appropriate server [11] and based on the server's load conditions which server to complete user's tasks is decided. To achieve proper load balancing cloud data center dynamically migrate and deploy virtual machine to meet user's needs. The migration architecture is shown in Figure 2.

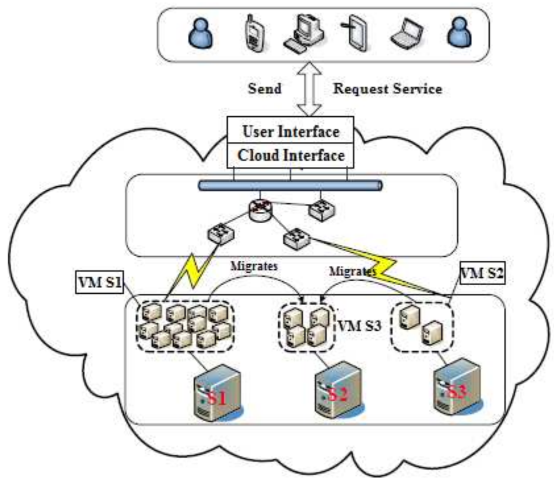


Figure 2: Dynamic Migration Architecture

II. RELATED STUDY

Lots of work has been proposed & done for energy efficiency & management on data centers for cloud. One of the first works, in which power management has been applied in the context of virtualized data centers, has been done by Nathuji and Schwan [12]. The authors have projected architecture of a data centre's resource management system where resource management is distributed into local and global policies. At the local level the system influences the guest OS's power management policies. The global manager catches the information on the current resource allocation from the local managers and applies its procedure to choose the VM placement requirements to be adapted. However, the authors have not proposed a specific strategy for automatic resource management at the global level. Kusic et al. [13] have well-defined the problem of power management in virtualized heterogeneous environments as a sequential optimization and addressed it using Limited Look ahead Control (LLC). The objective is to take full advantage of the resource provider's revenue by reducing both power consumption and SLA violation. Kalman filter is applied to evaluate the number of future demands to forecast the future state of the system and accomplish necessary reallocations. However, in contrast to heuristic-based approaches, the proposed model requires simulation-based learning for the application-specific modifications, which cannot be performed by Infrastructure as a Service (IaaS) Cloud providers. Verma et al. [14] have expressed the problem of power-aware dynamic placement of applications in virtualized heterogeneous systems as continuous optimization: at each time frame, the placement of VMs is optimized to reduce power consumption and improve performance. Equally to [15], live migration of VMs is used to attain a new placement at each time frame. The proposed algorithms, on the contrast to our approach, do not support SLAs: the performance of applications can be degraded because of to the workload inconsistency. In their more recent work [16], Verma et al. have proposed isolating VM consolidation approaches into static (monthly, yearly), semi static (days, weeks) and dynamic (minutes, hours) consolidation. Jung et al. [17], [18] have examined the problem of dynamic consolidation of VMs running, a multi-tier web-application using live migration, while meeting SLA needs. Kumar et al. [19] have suggested an approach for dynamic VM consolidation established on an approximation of "stability" – the possibility that a proposed VM reallocation will stay active for some time in the future. In contrast to the discussed studies, we propose efficient adaptive Migration thresholds for dynamic adaption of VM allocation at run-time according to the current utilization of resources applying dynamic migration, switching idle nodes to the sleep mode, and thus reducing energy consumption. The proposed method can effectively handle strict QoS requirements, multi-core CPU architectures, heterogeneous infrastructure and heterogeneous VMs. The algorithms adjust the actions according to the experimental performance characteristics of VMs. Hence, here we propose a strategy based on Adaptive Migration threshold which determines the CPU utilization dynamically with consideration of RAM & Bandwidth to satisfy unpredictable workloads & provide better performance & higher Quality of Service.

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

III. SYSTEM MODEL

We consider the system model same as described in [20], the target systems are of IaaS Environment.

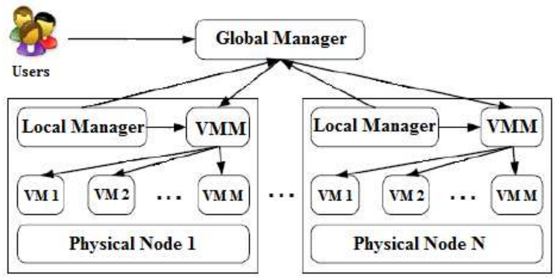


Figure 3: The System Model

As shown in the Figure 3, the system models consist of global and local manager. The local managers, which are part of VM monitor, resides on each node and are responsible for keeping continuous observation on when to migrate a VM and utilization of the node. The end-user refers its service request along with some CPU performance parameters like MIPS (Million Instruction per second), RAM, memory and network bandwidth to a global manager which in turns intimates the VM monitor for VM Allocation. The local manager reports the global manager about the utilization check of its node. And thus, global manager keeps the check of overall utilization of the resource. Our system model considers three main theories.

A. Degree of Load Balancing

Please Degree of the load balancing by the variance of the server's CPU utilization, where uiis the current CPU utilization of server resource i, m is the number of hosts, at a time all the hosts' CPU average utilization for the formula (1), then we obtain the degree of the load Balancing with the formula (2).

$$\bar{u} = \frac{\sum_{i=1}^{m} u_i}{m} \quad (1) \qquad B = \sqrt{\frac{1}{m} * \sum_{i=1}^{m} (u_i - \bar{u})^2} \quad (2)$$

B. Energy Consumption

For the calculation of energy consumption, assuming the server is idle, the percentage of its energy consumption is k. where Pfullis power consumption of the server at full load. Uiis the CPU utilization of the server. So as time, the total energy consumption growth E as

$$E = \int_t \left(k * P_{full} + (1-k) * P_{full} * u_i\right) \quad (3)$$

C. SLA Violations

QoS needed to be met for Cloud computing environments. QoS is determined in the form of SLA (Service Level Agreement), which is determined either by minimum throughput or maximizes response time. This can differ from system to system. For our studies, we consider SLA violation as shown in (4):

$$SLA = \frac{\Sigma\,(\text{requested MIPS}) - \Sigma\,(\text{allocated MIPS})}{\Sigma\,(\text{requested MIPS})} \quad (4)$$

The percentage of this value will show CPU is not allocated even if it is demanded. Therefore, in order to increase the QoS for the end-users, our prior goal is to minimize this SLA from getting violated.

IV. PROPOSED STRUCTURE

Here, we proposed Adaptive Migration threshold based approach with Minimum Migration Time(MMT) policy which minimizes migrations & improve Load balancing in data centers & reduce SLA violations. Here we divide the problem of dynamic VM consolidation into four parts:

(1) Determining when a host is considered as being overloaded requiring migration of one or more VMs from this host;

(2) Determining when a host is considered as being under loaded leading to a decision to migrate all VMs from this host and switch the host to the sleep mode;

(3) Selection of VMs that should be migrated from an overloaded host;

(4) Finding a new placement of the VMs selected for migration from the overloaded and under loaded hosts. We discuss the defined sub problems in the following sections.

A. *VM Allocation using Adaptive Threshold policy*

The selection of VM for migration is done to optimize the allocation. Here, we first calculated the CPU utilization of all VMs as shown below in (5):

$$Uvm = \frac{totalRequestedMips}{totalMips\ for\ that\ VM} \qquad (5)$$

$$Bw = \Sigma current\ allocated\ bandwidth\ for\ VMs\ for\ host$$
$$Ram = \Sigma current\ allocated\ Ram\ for\ VMs\ for\ host$$
$$And\ also\ \ Sum = \Sigma Uvm \qquad (6)$$

1) Upper Threshold

The CPU will be considered overloaded when the utilization is above this value so we migrate some of the VMs. Her

Here, so went on calculating this value i.e. Tupper for each host separately by following equations in (7):

$$temp = Sum + (Bw / Bw(host)) + (Ram/Ram(host))$$
$$Tupper = 1-(((Puu * temp) + Sum) - ((Pul * temp) + Sum)) \qquad (7)$$

Where, the variable temp gives the summation of parameter that we have taken into consideration. For each host amount of spare CPU capacity is preserved by upper probability (Puu) and lower probability (Pul).

2) Lower Threshold

The node is considered to be underutilized when the CPU utilization is below this value so all VMs are migrated to other node. From our study in [21], we considered that if the CPU utilization is above 30%, lower threshold (Tlower) is always 0.3. So, we define equations for calculating lower threshold for each node as follows in (8):

$$Temp = Sum + (Bw / Bw(host)) + (Ram/Ram(host))$$

$$Tlower = 1 - ((Pl * temp) + sum),\quad if\ CPU\ utilization\ is < 30\%$$
$$= 0.3, if\ CPUutilization\ is >= 30\% \qquad (8)$$

Where lower probability limit (Pl) is as lower limit for spare CPU capacity and n is number VMs on the host.

After defining the dynamicity of lower and upper threshold from the equation (7) and (8) respectively, we describe our theory for Adaptive Threshold based Dynamic Migration as shown in the Algorithm 1.

Algorithm 1: Dynamic Migration using Adaptive threshold

Input: host list, VM list **Output:** migration list

1. Sort the VM list in the decreasing order of its VM utilization.
2. For each host in host list compare the current host utilization value to the upper threshold value of that host. If the value is greater goto 3 else goto 7. Fix two best fit utilization: bfuupper and bfulower with max value assignment.

3. Get the each VM for the current host. If VM utilization is greater than the difference of current host utilization and upper threshold value, then goto 4 else goto 5.
4. If VM utilization – host utilization + upper threshold of host is greater than bfuupper then, bfuupper = VMuitilization–(host utilization – upper threshold) and best fit VM is current VM.
5. If bfuupper= max then, if (hostutilization – upperthreshold) – VMUtilization is less thanbfulower then, bfulower = (hostutilization – upper threshold) – VMutilization. And best fit VM is current VM.
6. Adjust the value of host utilization as difference of current host utilization and best fit VM utilization and add the best fit VM to the migration list and remove the VM from the current host.
7. If host utilization value is less than lower threshold value than add all the VM of the host to the migration list and remove all the VM from the host.
8. Return the migration list.

B. *VM Selection with MMT Policy*

Once it has been decided that a host is over utilized, the next step is to select specific VMs to migrate from this host. In this section Minimum Migration Time Policy for VM selection is used & applied iteratively. After a selection of a VM to migrate, the host is tested again for being over utilized. If it is still found as being over utilized, the VM selection policy is applied again to select another VM to migrate from the host. This is reiterated until the host is considered as being not overloaded.

The Minimum Migration Time (MMT) policy migrates a VM v that needs the minimum time to complete a migration comparatively to the other VMs allocated to the host. The migration time is calculated as the amount of RAM used by the VM shared by the spare network bandwidth offered for the host j. Let Vj be a set of VMs presently allocated to the host j. The MMT policy finds a VM v that fulfils following conditions defined in (9).

$$v \in Vj \mid \forall a \in Vj ,\ \frac{RAMu(V)}{NETj} \le \frac{RAMu(a)}{NETj} \qquad (9)$$

Where, RAMu(a) is the amount of RAM currently utilized by the VM a; and NETj is the spare network bandwidth available for the host j.

Algorithm 2:Minimum Migration Time Policy

Input: host list, VM migration list **Output:** VMs to be migrated

1. Get the list of VMs to migrate & hosts list.
2. If migratableVms list is empty then return nullandSetvmToMigrate is equals to null.
3. Set variable minMetric is equals to MAX_VALUE.
4. Else if vm.isInMigration then getRam value of VM and allocate it to variable Metric.
5. If value of Metric is less thanminMetric then setminMeric is equals to value of Metric &vmToMigrate is equals to value of vm.
6. Return vmToMigrate.

C. *VM Placement Using Power Aware BestFitDecresing Policy*

The general algorithm of VM placement optimization is shown in Algorithm 1. First, the algorithm looks through the list of hosts and by applying the overloading 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. Once the list of VMs to be migrated from the overloaded hosts is built, the VM placement algorithm is invoked to find a new placement for the VMs to be migrated. The second phase of the algorithm is finding under loaded hosts and a placement of the VMs from these hosts. The 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 under loaded hosts. The complexity of the algorithm is 2N, where N is the number of hosts.

Algorithm 3: Power Aware Best Fit Decreasing Policy

Input: host list, VM list **Output:** allocation of VMs

1. Sort the VM list in the decreasing order of its VM utilization.
2. For each VM in VM list, allocate minpower as maximum power and allocatedHost as null.

3. For each host in host list, if host has enough resource for VM then estimate power of VM and host. If power is less than minpower then allocated host is current host and minpower is power difference of VM and host.
4. If allocatedHost is not null then allocate VM to the allocatedHost.

Return allocation

V. EXPERIMENTAL RESULTS

CloudSim toolkit [22] has been preferred as a simulation platform, as it is a modern simulation framework targeted at Cloud computing environments. In contrast to another simulation toolkits (e.g. SimGrid, GangSim), it permits the modelling of virtualized environments, supporting on demand resource provisioning, and their management. It has been extended to enable energy aware simulations, as the core framework does not provide this ability. Apart from the energy consumption modelling and accounting to the capability to simulate service applications with dynamic workloads has been derived. The implemented extensions have been included in the 3.0 version of the CloudSim toolkit. In our experiment, we have worked with just one datacenter. We took up with 100 host on this datacenter which in turn is running 100 virtual machines on those hosts. Each node comprises of one CPU core with 10 GB ram/network bandwidth and storage space of 1TB. The host comprises of 2500, 2000, 1000 and 500 MIPS accordingly. For each virtual machine on host ram size is 1024, 512, 256 & 128 MB respectively and bandwidth size is 100 Mbit/s with VM size 2.5 GB. For our experiment we have just worked with one resource. Initially the VMs are considered to be utilized by 100% of time.

Firstly, we tried to work on analysis of concept of Adaptive Migration Threshold and its implementation on Cloudsim Toolkit. Then we went on studying online deterministic energy efficient algorithms[23] already implemented in Cloud Sim 3.0 in power package ,such as Median Absolute Deviation(MAD,)Static Threshold (ST),inter Quartile Range, Local Regression(LR), Local Regression Robust(LRR). Along with the understanding of Adaptive migration Threshold base approach here implemented the algorithm of Adaptive Migration thresholds with consideration of RAM & Bandwidth for proper utilization of resources, high performance & better quality of service. Instead of static threshold values for Upper & Lower thresholds, here Dynamic Upper & Lower Threshold values calculated according to RAM & BANDWIDTH parameter at each iteration accordingly. The results are taken as shown below:

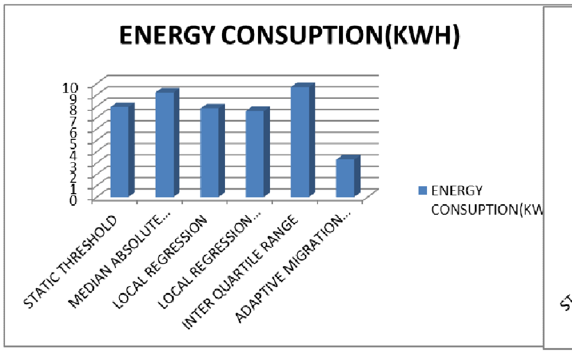
COMPARATIVE RESULTS OF DIFFERENT POLICIES UNDER 100 HOSTS

	ENERGY CONSUPTION (KWH)	% OF SLA VIOLATIONS	NO. OF VM MIGRATIONS	NO OF SLA VIOLATIONS
STATIC THRESHOLD	**7.92**	**14.18**	**747**	**233**
MEDIAN ABSOLUTE DEVIATION	**9.21**	**11.8**	**754**	**133**
LOCAL REGRESSION	**7.81**	**12.73**	**482**	**220**
LOCAL REGRESSION ROBUST	**7.57**	**12.73**	**486**	**220**
INTER QUARTILE RANGE	**9.7**	**11.29**	**762**	**610**
ADAPTIVE MIGRATION THRESHOLD	**3.29**	**10.21**	**382**	**100**

Throughout this experiment, the proposed policy(Adaptive Migration Thresholds) is compared with other dynamic policies to reflect the goal of energy saving and reducing number of Migrations and SLA violations. Results are shown in Table 1:

From the Table 1, we concluded that by using Adaptive Migration Thresholds for migration, energy usage can be minimized resulting into decreasing electricity bills for data centers as compare to other algorithms such as Median Absolute Deviation(MAD), Static Threshold (ST),inter Quartile Range, Local Regression(LR), Local Regression Robust(LRR).

Following figures shows the comparative results of all algorithms represented as graphs.



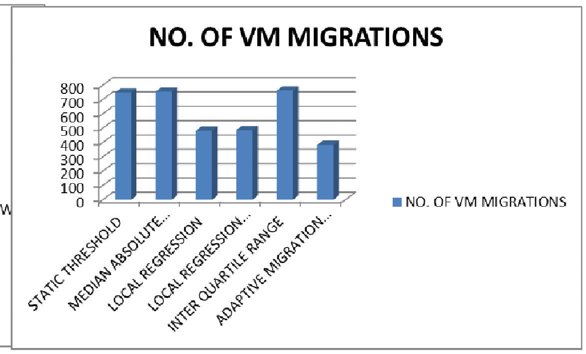


Figure 4: Energy consumption (Kwh) V/S Hosts

Figure 5: No. of Migrations V/S Hosts

From the figure 4, we can see that the energy consumption for Adaptive Migration Thresholds (ADT) based Policy is significantly less than the other dynamic policies. Results show ADT has upto 50% less energy consumption than other policies.

Figure 5, shows graph of VM migrations v/s number of hosts. Here also proposed ADT policy works better than others in reducing No. of Migrations. Median Absolute Deviation (MAD) Static Threshold (ST) & inter Quartile Range have highest number of VM migrations. Local Regression (LR), Local Regression Robust (LRR). Less VM migrations improves energy efficiency of system, hence improve QOS & SLA.

From the Figure 6 & Figure 7, we can show that Adaptive Migration Thresholds (ADT) significantly decreases SLA Violations rather than other dynamic policies. Inter Quartile Range (IQR) has highest SLA violations. Static Threshold (ST) ,Local Regression(LR) & Local Regression Robust (LRR) have same SLA violations in average. Median Absolute Deviation (MAD) & Adaptive Migration Thresholds (ADT) have closure results to each other. Minimized SLA violations provide better load balancing to the system & improve efficiency of system.

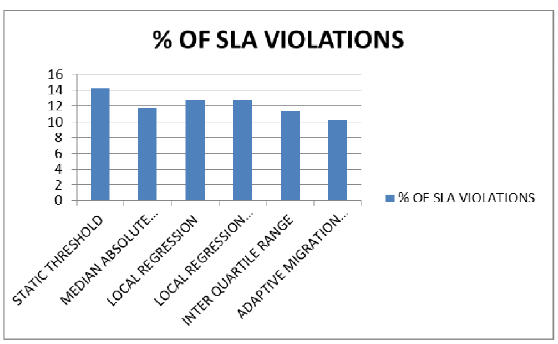


Figure 6: No. of SLA Violations V/S Hosts

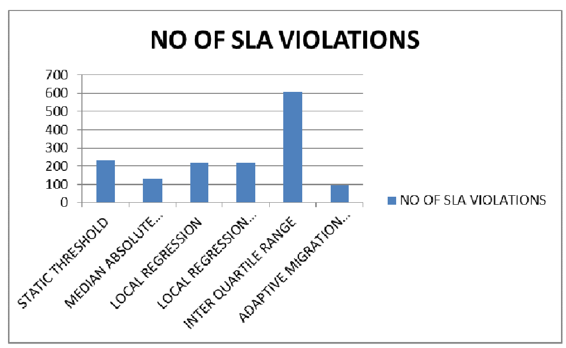


Figure 7: % SLA Violations V/S Hosts

VI. Conclusion and Future Work

From our study we conclude that Adaptive Migration Thresholds based dynamic provisioning of VM and switching off idle servers reduces the energy consumption dramatically, hence improve Energy Efficiency. The main goal of algorithm is to reduce the energy usage which can be a small step towards Green technology. Moreover, we have also considered the Quality of Service (QoS) to the users by minimizing the SLA violations for the resource using MMT policy and also provides balanced load in datacentre. Here also VM migrations are significantly reduced as compare to other dynamic algorithms. The algorithm has been tested and simulated on with our results which clearly show that by Adaptive Migration Thresholds we can have better results & more work can be done.

For our future work, a test bed can be created to investigate the algorithm behaviour with multiple numbers of resources & we would also investigate this technique on real cloud setup and check what will be its exact reaction of on environment. This can be a small social step for significant decrease in emission of carbon dioxide along with reduction in infrastructure and operating cost.

REFERENCES

[1] Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems 2009; 25(6):599–616.

[2] Belady C. In the data center, power and cooling costs more than the it equipment it supports 2007. URLhttp://www.electronics-cooling.com/articles/2007/feb/a3/.

[3] Barroso LA, Holzle U. The case for energy-proportional computing. Computer 2007; 40(12):33–37.

[4] Fan X, Weber WD, Barroso LA. Power provisioning for a warehouse-sized computer. Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA 2007), ACM New York, NY, USA, 2007; 13–23

[5] The green grid consortium 2011.URL : Http://www.thegreengrid.org.

[6] Xen and the art of virtualization .Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP2003), Bolton Landing, NY, USA.

[7] Clark C, Fraser K, Hand S, Hansen JG, Jul E, Limpach C, Pratt I, Warfield A. Live migration of virtual machines. Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI 2005), USENIX, Boston, MA, USA, 2005.

[8] Daniel Versick and Djamshid Tavangarian, Reducing Energy consumption by Load Aggregation with an Optimized Dynamic Live Migration of Virtual Machines, International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010, pp. 164 { 170.

[9] Po-Cheng Chen, Cheng-I Lin, Sheng-Wei Huang, etc., A performance Study of Virtual Machine Migration vs. Thread Migration for Grid Systems, 22th International Conference on Advanced information Networking and Applications (IEEE), 2008, pp. 86 { 91.

[10] Timothy Wood, K. K. Ramarkrishnan, CloudNet: Dynamic Pooling of Cloud Resources by LiveWAN Migration of Virtual Machines, VEE '11 Proceedings of the 7th ACM SIGPLAN/SIGOPS international Conference on Virtual execution environments (ISBN), 2011, pp. 121 { 132.

[11] Chongguang REN, An Improved Adaptive Dynamic Programming Algorithm for Cloud Storage Resource Allocation, Journal of Computational Information Systems, 2011, pp. 5041 { 5048.

[12] Nathuji R, Schwan K. Virtual power: Coordinated power management in virtualized enterprise systems. ACM SIGOPS Operating Systems Review 2007; 41(6):265–278.

[13] Kusic D, Kephart JO, Hanson JE, Kandasamy N, Jiang G. Power and performance management of virtualized computing environments via look ahead control. Cluster Computing 2009; 12(1):1–15.

[14] Verma A, Ahuja P, Neogi A. p Mapper: Power and migration cost aware application placement in virtualized systems. Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware (Middleware 2008), Springer, Leuven, Belgium, 2008; 243–264.

[15] Verma A, Dasgupta G, Nayak TK, De P, Kothari R. Server workload analysis for power minimization using consolidation. Proceedings of the 2009 USENIX Annual Technical Conference, San Diego, CA, USA, 2009; 28–28.

[16] Jung G, Joshi KR, Hiltunen MA, Schlichting RD, Pu C. Generating adaptation policies for multi-tier applications in consolidated server environments. Proceedings of the 5th IEEE International Conference on Autonomic Computing (ICAC 2008), Chicago, IL, USA, 2008; 23–32.

[17] Jung G, Joshi KR, Hiltunen MA, Schlichting RD, Pu C. Generating adaptation policies for multi-tier applications in consolidated server environments. Proceedings of the 5th IEEE International Conference on Autonomic Computing (ICAC 2008), Chicago, IL, USA, 2008; 23–32.

[18] Jung G, Joshi KR, Hiltunen MA, Schlichting RD, Pu C. A cost-sensitive adaptation engine for server consolidation of multitier applications.Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware (Middleware 2009), Urbana Champaign, IL, USA, 2009;1–20.

[19] Kumar S, Talwar V, Kumar V, Ranganathan P, Schwan K. vManage: loosely coupled platform and virtualization management in data centers. Proceedings of the 6th international conference on Autonomic computing (ICAC 2009),Barcelona,Spain, 2009;127–136.

[20] Y. Song, “Multi-Tiered On-Demand resource scheduling for VM-Based data center” In Proc. of the 2009 9th IEEE/ACM Intl. Symp. On Cluster Computing, 155, 2009.

[21] A Beloglazov, R Buyya, “Energy Efficient Resource Management in Virtualized Cloud Data Centers”10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010.

[22] Calheiros RN, Ranjan R, Beloglazov A, Rose CAFD, Buyya R. 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.

[23] Anton Beloglazov and Rajkumar Buyya, Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers, Published online in Wiley Inter Science (www.interscience.wiley.com). DOI: 10.1002/cpe, Concurrency Computat.: Pract. Exper. 2011; 00:1–24

