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. 4, Issue. 12, December 2015, pg.1 – 7

Energy-Aware Dynamic Load Balancing of Virtual Machines (VMs) in Cloud Data Center with Adaptive Threshold (AT) based Migration

Khushbu Maurya
Computer Department,
A.I.T. Ahmedabad
khushbumaurya87@gmail.com

Richa Sinha
Information Technology Department
K.I.T. & R.C.
Kalol,
richa.872005@gmail.com

Nidhi Purohit
Computer Department,
Indus Institute of Technology & Engineering, Ahmedabad
purohit.nidhi@yahoo.com

Abstract—The in compliance between user requests and the specification of physical server in cloud computing led to poor load balancing, poor performance and large power consumption etc. Concentrating at this problem, this paper presents an energy-aware resource scheduling strategy based on dynamic migration of virtual machines. The strategy integrates the virtualization characteristics of cloud computing, presuming maximum and minimum thresholds, and selects those virtual machines on the physical servers that the load is over the upper values or load under the lower threshold values to be migrated, Boosting resource utilization of the cloud data center and reducing their energy consumption. To reduce migration here we integrate minimum migration time policy. Resulting experiments show that the strategy can reduce the migration times and the energy consumption of virtual machine migration, and further achieve load balancing and meet service level agreement (SLA) requirements.

Keywords: Cloud Computing; Dynamic Migration of Virtual Machines; Reducing Energy; Load Balancing; Service Level Agreement (SLA); Adaptive Threshold

1. INTRODUCTION

With the dawn of the Internet information age, user data increased greatly, traditional computing models being unable to meet the needs of large-scale data processing. Aiming at these circumstances this paper adopts a new computing model — cloud computing [1]. One important features of cloud computing is resource virtualization [2]. Increasing demand of computing resources raises energy consumption, this will lead to lower utilization and poor load balancing [3] of resources in resource pool,. As the traditional static migration strategy causes unnecessary overheads of migrations [4].Therefore, 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. Then, this paper proposes a resource scheduling strategy based on dynamic migration of VMs.

2. DYNAMIC MIGRATION OF VMS

Cloud computing architecture includes the application layer, the platform layer, and 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 [5]. 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 data center for processing user service requests to model. In order to allow different levels of performance isolation to simulate different resource scheduling strategy, using the two virtual machine scheduling model: One is the host level, another is virtual machine-level [6]. 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.

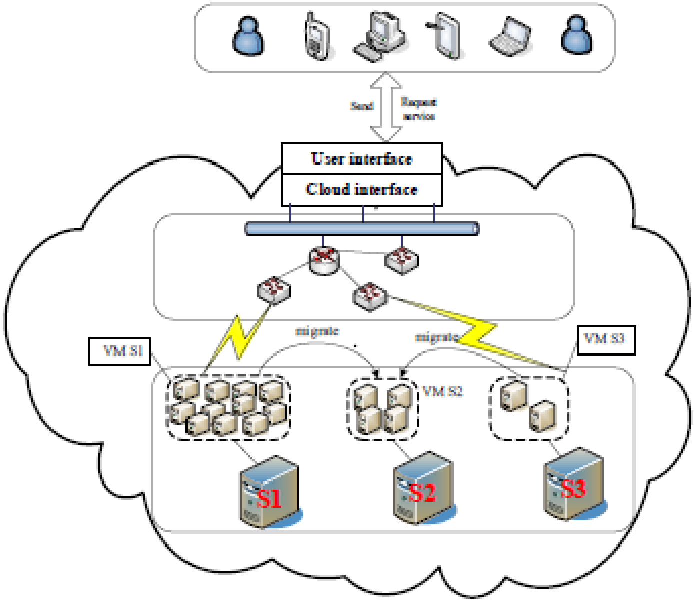

Figure 1: Dynamic Migration Architecture

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 [7], 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 1.

3. System Model

We consider the system model same as proposed in [8], the target systems are of IaaS Environment.

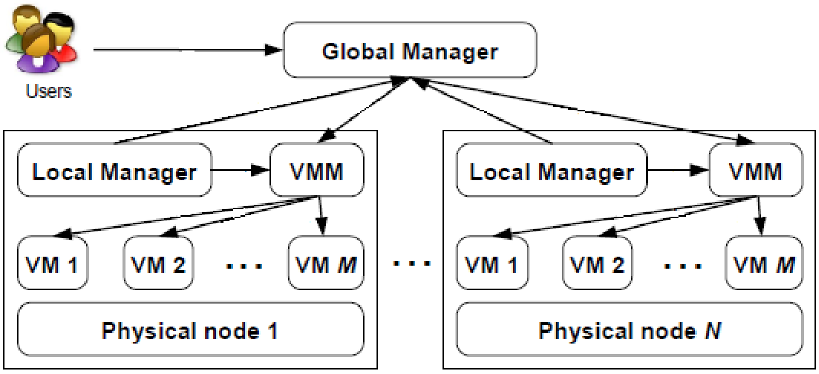

Figure 2: The System Model

As shown in the Figure 2, 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 sends 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.

3.1 Degree of Load Balancing:

degree of the load balancing by the variance of the server's CPU utilization, where *ui* is 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).

$$\bar{u} = \frac{\sum_{i=1}^{m} u_i}{m} \quad (1)$$

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

3.2 Energy Consumption:

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

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

3.2 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.

4. Proposed Structure

Here, we proposed Adaptive threshold based scheme with Minimum Migration Time (MMT) policy which minimizes migrations & improve Load balancing in data center & decrease SLA violations. We divide the algorithm in three parts: (1) VM Allocation using Adaptive Threshold policy (2)VM Selection using MMT policy & (3)VM Placement using Best Fit Decreasing Policy on proper host.

4.1 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):

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

Beside utilization we also have considered allocated ram and bandwidth for both virtual machine and host as shown in (6) :

Bw = Σcurrent allocated bandwidth for VMs for host
Ram= Σcurrent allocated Ram for VMs for host
And also Sum = ΣUvm (6)

We, hence show in our scheme the two threshold values.

4.1.1 Upper Threshold

The CPU will be considered overloaded when the utilization is above this value so we migrate some of the VMs. 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-(((0.95*temp)+Sum)-(0.90*Temp)+Sum))

Where, for each host we preserve amount of CPU capacity by upper 0.95 and lower as 0.90 probability limits.

4.1.2 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 [13], 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):

sum1=(Sum+Bw+Ram) / (n*Bw(host)+Ram(host)),

sqrt = 1-(0.3* temp)+ Sum),

Tlower =1 – ((0.3 * sqrt1)+sum) ,
if CPU utilization is < 30%
= 0.3 , if CPU utilization is >= 30% (8)

where, we considered maximum probability limit for this threshold as 0.3 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 then bfulower 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.

4.2 Selection of VM Using MMT Policy

Here We have used a Minimum Migration Time policy rather Simple Policy of migration for a better QoS.

Algorithm 2:Minimum Migration Time Policy

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

1. Get the list of VMs to migrate & hosts list.
2. If migratableVms list is empty then return null & Set vmToMigrate is equals to null.

3. Set variable minMetric is equals to MAX_VALUE.
4. Else if vm.isInMigration then get Ram value of VM & allocate it to variable Metric.
5. If value of Metric is less than minMetric then set minMeric is equals to value of Metric & vmToMigrate is equals to value of vm.
6. Return vmToMigrate.

4.3 VM Placement Using MBFD Policy

We have considered placing of VM as a bin packing[22] type of problem. So, for placing the VM we have used MBFD (Modified Best Fit Decreasing) algorithm. We describe the algorithm for placing VM as below.

Algorithm 3: Modified Best First 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.
5. Return allocation

5. Experimental Results

We tested our work on Cloudsim Toolkit [9]. In our experiment, we have worked with just one datacenter. We took up with 10 host on this datacenter which in turn is running 20 virtual machines on those host. Each node comprises of one CPU core with 10 GB ram/network bandwidth and storage space of 1TB. The host comprises of 1000, 2000 and 3000 MIPS accordingly. For each virtual machine on host ram size is 128MB and bandwidth size is 2500 MB with 250, 500, 750 and 1000 MIPS accordingly. 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 conceptual of dynamic migration and its implementation on Cloudsim Toolkit. Then we went on studying the power examples already implemented i.e. DVFS [10] , NPA [11] & Single Threshold. Then along with the understanding of Dynamic migration, we tried implementing Double threshold on it with static assignment of upper & lower limit threshold values. considering 0.7 as upper threshold and 0.3 as lower threshold. Finally, we moved on implementing our concept of Adaptive threshold using the threshold theories stated in previous section.

To calculate degree of load balancing, on every 10 seconds utilization of each physical machine is calculated with running virtual machines The results are as Figure 3:

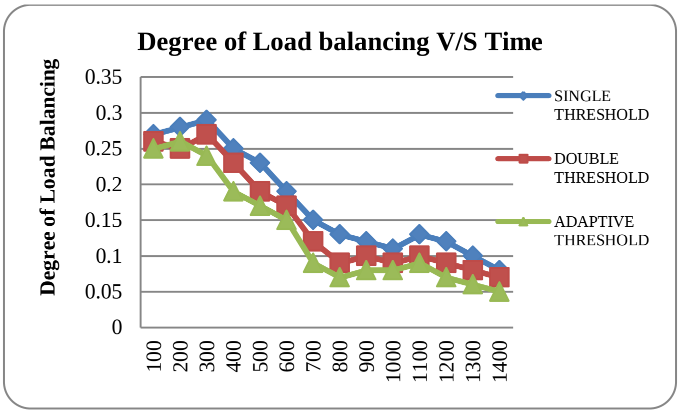

Figure 3: Degree of Load Balancing V/S Migration Time

It can be seen form Figure 4, with the increasing in migration time, the degree of load balance using Adaptive Threshold (AT) and Minimum Migration Time(MMT) is gradually reducing, AT policy has better load balancing effect, and thus more effective in improving resource utilization.
Throughout this experiment, the proposed policy is compared with NPA,DVFS & Single Threshold policy to reflect the goal of energy saving & load balancing with the Minimum number of Migrations and SLA violations. Results are shown in Table-1 :

POLICY	ENERGY CONSUPTION (Kwh)	NO. OF VM MIGRATION	NO. OF SLA VIOLATIONS	SLA VIOLATIONS (%)
DVFS	0.25	0	0	0
NPA	0.77	0	0	0
SINGLE THRESHOLD	0.22	37	2560	93.60
DOUBLE THRESOLD	0.20	26	2156	75.20
ADAPTIVE THRESHOLD	0.17	17	1645	62.83

Table-1

As shown in the table 1, we concluded that by using energy efficient policy for migration, energy usage can be minimized resulting into decreasing electricity bills for data centers. NPA is using maximum amount of power among all the theories taken into consideration resulting into more cost. DVFS may use less energy but for the real scenario it may change because it entire dependency is limited to voltage and frequency. The single threshold is violating the maximum number of SLA with nominal energy consumption. Next, in double threshold the SLA violation has dropped significantly from the single threshold. Finally comes the Adaptive threshold. Here also we found a more drop in SLA violation as compared to double threshold with a bit fluctuation in energy too. The results are shown in following figures.

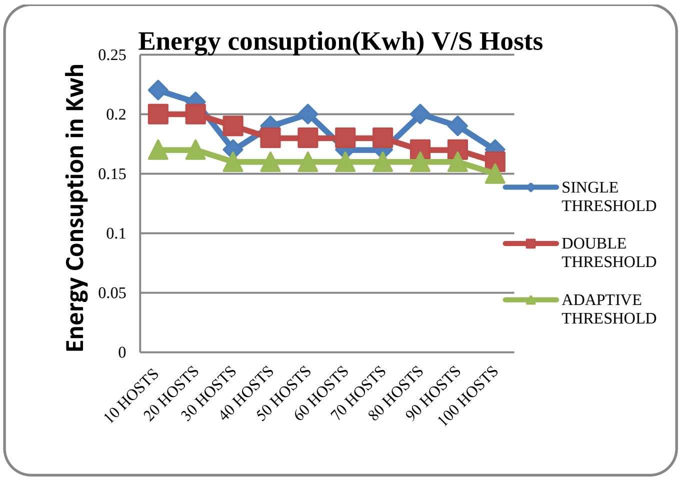

Figure 4: Energy consumption(Kwh) V/S Hosts

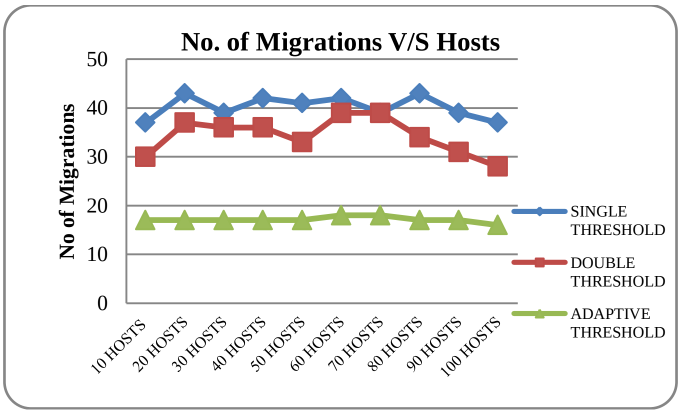

Figure 5: Energy consumption(Kwh) V/S Hosts

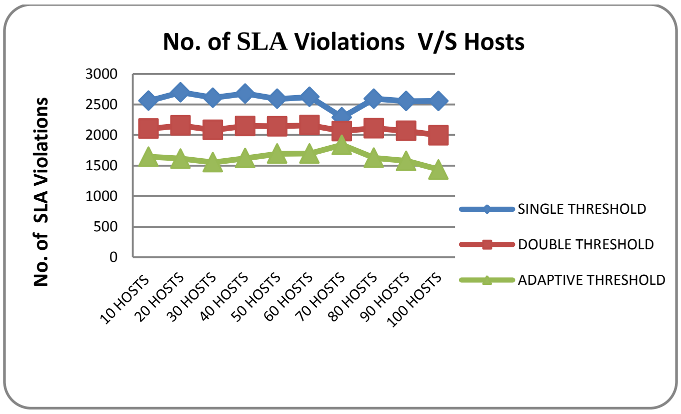

Figure 6: No. of SLA Violations V/S Hosts

6. Conclusion and Future Work

From our study we conclude that adaptive threshold based resource allocation with dynamic migration of VM and switching off idle servers maximizes the energy efficiency of the resource. The algorithm has tried to reduce the energy usage which can be a small step towards Green technology. Moreover, we have also considered the quality of service to the users by minimizing the SLA violation for the resource using MMT policy & also provides load balancing with migration time. This algorithm is been tested and simulated on with our results which clearly show that by increasing CPU utilization more work can be done.

For our future work, we would like to introduce an optimization policy to meet the cost requirement. Secondly, a test bed can be created to investigate the algorithm behavior with multiple numbers of resources. Thirdly, 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] R. Buyya, CS Yeo,S. Venugopal, J. Broberg, I. Brandic, “Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems, 2011

[2] Zhang bo, Gao ji, Ai Jieqing, Cloud Loading Balance Algorithm, Information Science and Engineering (ICISE), 2010 2nd International Conference on, 2010.

[3] Mauro Andreolini, Sara Casolari, Michele Colajanni, Dynamic load management of virtual machines in a cloud architecture, Department of Information Engineering, 2010.

[4] 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.

[5] 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.

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

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

[8] 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 .

[9] R. Calheiros, R Ranjan, César A. F. De Rose, R. Buyya, “ CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services” , 2011

[10] A. Beloglazov, R. Buyya, “Energy efficient allocation of virtual machines in cloud data centers”. 10th IEEE/ACM Intl. Symp. on Cluster, Cloud and Grid Computing ,2010.

[11] G Laszewskiy, L Wangz, A J. Youngez, X Hez,“Power-Aware Scheduling of Virtual Machines in DVFS-enabled Clusters, IEEE,2009

