Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 89 (2016) 27 – 33

Procedia Computer Science

Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016)

VM Consolidation for Cloud Data Center using Median based Threshold Approach

Oshin Sharma and Hemraj Saini*

Jaypee University of Information Technology, Waknaghat 173 234, H. P., India

Abstract

Cloud computing is an on demand computing model which requires large amount of physical devices and provide services to users on the basis of pay per usage model, therefore excessive demand of cloud computing have also led to the growth of computational power inside datacenters. These datacenters consumes huge amount of energy which results high carbon emission. For the optimization of resources and reduction of energy consumption, virtual machine consolidation can be used by switching the idle nodes to sleep mode or by turning them off and by using live migration of virtual machines. Here, we propose a novel method for consolidation of virtual machines such that it meets Service Level Agreements (SLA) and deals with energy-performance trade-off. Therefore, reduction of SLA violation and minimize the performance degradation during migration are two main objectives in this paper. For the allocation and reallocation of virtual resources depending upon their load, this threshold based approach can be used, in which Median method is used to find lower and upper threshold values. Proposed Median based threshold approach is implemented by using CloudSim and validation of this approach is performed across different workload traces of PlanetLab servers and using some random configuration of Datacenters. Experimental results show that this scheme can provide better SLA performance.

Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016

Keywords: Cloud Computing; Median Method; Threshold Approach; Service Level Agreement; VM Consolidation.

1. Introduction

Cloud computing is a distributed computing with number of virtualized and interconnected computers that provision the computing resources on the basis of SLA (Service Level Agreement) between cloud users and providers. Cloud providers deliver these services and resources depending upon the services offered by cloud architecture with three different layers. SaaS (Software as a service) provides application software as a service. PaaS (Platform as a service) provides platform to deploy the services and application on it and third is IaaS (Infrastructure as a service) which provides basic infrastructure to cloud users. With increasing demand of cloud environment, energy consumption inside data centres is also continuously increasing and results high carbon emission which should be taken care of. Virtualization is the key feature of cloud computing that allows multiple virtual machines inside one physical machine and perform live migration of VMs as well[1]. Different applications with different resource requirements are running simultaneously on same physical machine which led to variable workload on machine. Therefore, consolidation of VM

*Corresponding author. Tel: +918894969853; Fax: +91 1792 245362.
E-mail address: hemraj1977@yahoo.co.in

Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016
doi:10.1016/j.procs.2016.06.005

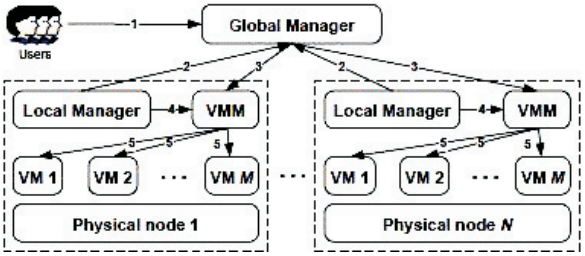

Fig. 1. System Model[13].

on minimum number of active hosts and switching off the ideal host is a novel method to save the energy consumption of environment. But, excessive consolidation may also degrade the performance; therefore to provide high quality of services to users it should be necessary to deal with energy and performance trade-offs[1].

To provide high level of QOS to cloud users, live migration approach is used to consolidate the VMs using four different steps[2]. In which all the virtual machines will be migrated to some another host and switching the ideal node into sleep mode, if the PM is underutilized (with CPU utilization less than some threshold value) and similarly some of the virtual machines will be migrated to different host machine if host machine is overloaded (with CPU utilization greater than some threshold value). Four different steps are: to select which PM is over utilized, to select which PM is underutilized?, select the VM from these underutilized and over utilized host machines, find new placement techniques of VM over the host machine. In this paper, our main focus is on first two parts i.e. to find out the over utilized machine and underutilized machines. For which we proposed a median based threshold approach, whose main objective is to reduce energy consumption as well as SLA violation, performance degradation.

In addition to this paper is organized in following sections. Section 2 presents brief review of related work in VM consolidation. In section 3 we propose median based threshold approach and overview of experimental details. Then, in section 4 we conduct the performance analysis using simulation and analyze the results of our proposed method with previous methods and finally paper is concluded in section 5 along with future scope.

2. Related Work

There is an excessive amount of research on VM consolidation that addresses that it is the best solution for performance and energy management in cloud data centers. According to which workload is consolidated among the lesser number of physical machines. First work in this field, for power management of data centers have proposed by Nathuji and sachwan[3]. An architectural model has been proposed in this paper, in which resource management can be done by global and local managers both. Global manager monitors utilization of host machine for the selection of most appropriate host machine to migration and local manager monitors the power management of guest VM. Process of VM consolidation has been divided into three categories: dynamic VM consolidation, static and semi static[4] by Verma A. and they have used static and semi static approach for live migration and showed that it is more advantageous than dynamic consolidation.

Anton and Rajkumar Buyya[5] presented the VM consolidation into four steps[6]: detection of over utilized and underutilized host machines, selection of VM from these host machines and ten placements of VM over some new host machine. They have also proposed policies such as Random choice policy, minimum migration time, maximum correlation, for the selection of virtual machines from the selected host for migration. Later on, minimization of energy consumption by considering the structural components[7] such as cooling equipments, network topology, rack utilization has been presented by Sina Esfandiarpoor. He also presented structure aware virtual machine placement methods like: NUR, RBR, and HSRC. Abbas Horri also presented their work by considering the consolidation process same as[6]. They have used linear regression (LR) method for over utilized host machines and proposed VM placement algorithm.

Novel heuristics have been introduced for the determination of underutilized host machine by Ehsan Arianyan[1] such as: Migration delay (MDL), Available capacity (AC), TOP-SIS available capacity, number of VMs and migration delay (TACND). They have improved the number of migrations and SLA violation in comparison to previous policies.

Sandpiper[10] introduced the algorithm to reduce virtual machine migration and also presented some heuristics to select virtual machines to migrate from over utilized host and the target host for this selected VM. For the placements of virtual machines over host machine, the bin packing algorithm has been introduced[11,12]. This algorithm has been implemented in order to minimize the number of bins used during the packing of objects to achieve minimum power consumption.

The linear regression based approach has been implemented by Fahimeh Farahnakian[9]. The CPU usage of the host machine is predicted on the basis of linear regression technique and then live migration process is used to detect underutilized and over utilized machine whereas, In static threshold approach[13], a threshold value is used to decide which host machine is over utilized. If the CPU utilization of host is greater than the static threshold value (suppose 85%) of total capacity, then host machine is considered as over utilized. The static threshold method or fixed values of thresholds are not suitable for dynamic workload, as they do not adapt the changes in workload. Therefore, recent work has more focused on the heuristics that can determine the upper and lower threshold for dynamic workloads depending upon the deviation of CPU utilization of the host. In current work, we propose a dynamic method to set the lower and upper threshold values for determining over utilized and underutilized host machines depending upon their history of resource usage.

3. Proposed Approach

3.1 Median based threshold approach for finding over utilized and underutilized host machines (MEDTH)

The proposed approach of Anton Beloglazov[2] for determining underutilized and over utilized machines for dynamic workload is extended by considering CPU utilization of all host machines. We have proposed a Median based threshold method for detection of over utilized and underutilized host machine. If a host machine is determined as a over utilized machine then some of the VMs from this host machine are migrated to another host, and if the host machine is determined as underutilized machine then all VMs of this host machine are migrated to another host machine such that not to making them overloaded and switching these idle machines to sleep mode or turn off.

First we have calculated CPU utilization of all host machines present inside datacenter and then upper and lower values of threshold are to be calculated. These threshold values can be used for detection of over utilized and underutilized host machine. Here we have generated the CPU utilization of host machine using random generator.

C_i represents the CPU utilization of A_i numbers of host machines, where $i \in R^+$ and A_i number of host machines can be arranged as: $\{A_i = A_1, A_2, A_3 \ldots A_i\}$. Host machines can be even and odd in numbers, therefore median of even numbers of machines i.e. A_{2i} gives two new sets: first is (A_1 to A_i represented as X_i), and second is ($A_i + 1$ to A_{2i} represented as X_j). Similarly, two different sets can also be generated for odd numbers of host machines i.e. $A_{2i} + 1$. First set is (A_1 to A_i represented as Y_i) and second set is ($A_i + 2$ to A_{2i} +1 represented as Y_j).

Similarly, upper and lower threshold limits T_u and T_l for both even and odd number of host machines present inside data centers can be detected using median method formularized in equation (1) and equation (2).

$$\text{If } \frac{A_i}{2} = 2x(x \in 1, 2, 3 \ldots \infty) \begin{cases} T_l = \text{median} & (X_i) \\ T_u = \text{median} & (X_j) \end{cases} \tag{1}$$

$$\text{If } \frac{A_i}{2} = 2x + 1(x \in 1, 2, 3 \ldots \infty) \begin{cases} T_l = \text{median} & (Y_i) \\ T_u = \text{median} & (Y_j) \end{cases} \tag{2}$$

With the help of above two equations, finally the over utilized and underutilized machines can be detected as following:

$$\begin{cases} \text{if} & CA_i > T_u(A_i = O_h) \\ \text{if} & CA_i < T_l(A_i = U_h) \end{cases} \tag{3}$$

Here in above eqn. (3), O_h and U_h represents over utilized and underutilized host machines. If CPU utilization is less than lower threshold value then host machine is considered to be underutilized and if the utilization is greater than

upper threshold then it is considered as over utilized machine. The Pseudo-code for finding upper and lower threshold using our proposed Median method is following:

Input: list of hosts
Output: lower threshold value

1. Begin
2. Sort the list in increasing order according to utilization history
3. M ← median of (hosts)
4. For each host in the list less than M
5. List2 ← hosts
6. End for
7. LT ← median of (List2)
8. End

Algorithm 1. To find Lower Threshold

Input: list of hosts
Output: upper threshold value

1. Begin
2. Sort the list in increasing order according to utilization history
3. M ← median of (hosts)
4. For each host in the list greater than M
5. List2 ← hosts
6. End for
7. UT ← median of (List2)
8. End

Algorithm 2. To find Upper Threshold

For both of these algorithms, initially we sort the list of host in increasing order according to utilization history. Then we calculate the median of both these sets and these median values will become lower and upper thresholds. This can be varying according to workloads. In addition to this, LT (lower threshold) and UT (upper threshold) are used for determining underutilized and over utilized machines and algorithms for their detection are following:

Input: list of hosts
Output: lower threshold value

1. Begin
2. Sort the list in increasing order according to utilization history
3. M ← median of (hosts)
4. For each host in the list less than M
5. List2 ← hosts
6. End for
7. LT ← median of (List2)
8. End

Algorithm 3. To Check Over Utilized Host

Input: list of hosts
Output: upper threshold value

1. Begin
2. Sort the list in increasing order according to utilization history
3. M ← median of (hosts)
4. For each host in the list greater than M
5. List2 ← hosts
6. End for
7. UT ← median of (List2)
8. End

Algorithm 4. To Check Underutilized Host

Initially host utilization is compared with UT and LT values and then accordingly Boolean results are returned. Here we have taken 10 as a safety parameter for over utilized and underutilized detection of machines. In order to improve SLA and performance or to minimize energy consumption, all the hosts from underutilized machines should be migrated to another host and switches these hosts to sleep mode. We compared our method with three others in CloudSim[2] with same allocation policies they have.

3.2 Performance metrics

To evaluate the efficiency of our proposed median based threshold approach, we simulate different resource allocation scenarios on Cloudsim toolkit[2] and then compare our proposed method with them such as static Threshold (THR), Median absolute deviation (MAD), Interquartile range (IQR). It is very difficult to perform the evaluations on real infrastructure therefore; CloudSim provides support for efficient and repeatable experiments. In our current work, we used two metrics to evaluate the performance. First is SLATAH (SLA violation time per active host), second is PDM (performance degradation due to migration) and third metric is SLAV (SLA violation).

3.2.1 SLATAH

SLA violation time per active host represents the percentage of time when CPU utilization reaches to 100%. In the following equation (4), N represents number of host machines, T_{si} represents the time when CPU utilization reaches

to 100% and T_{ai} represents total time for which host remains active.

$$SLATAH = \sum_{i=1}^{N} \frac{T_s i}{T_a i} \tag{4}$$

3.3 PDM

PDM represents the total performance degradation during virtual machine migration. Performance degradation depends upon the CPU utilization of host[14]. In equation (5), M represents number of virtual machines. C_{dj} represents performance degradation during migration and C_{rj} represents total CPU capacity requested by VM.

$$PDM = \frac{1}{M} \sum_{j=1}^{M} \frac{C_{dj}}{C_{rj}} \tag{5}$$

3.3.1 SLAV

SLAV provides the combined results from both of above metrics SLATAH and PDM. As, these two metrics are equally important for SLA violation

$$SLAV = SLATAH * PDM \tag{6}$$

4. Performance Analysis and Simulation Results

4.1 Experimental details

For the performance evaluation of current work, we obtained the results for both random and real workloads. For random workloads, data center comprises with 800 heterogeneous hosts and request for the provisioning of 800 VMs are submitted by users. For real workload we have selected the data from CoMon project, a monitoring infrastructure of PlanetLab[15]. Here in this project, data for the CPU utilization of thousands of VM is obtained from servers situated more than 500 places around the world and data is collected after every five minutes. We selected data of four days from workload traces of the project during March 2011. There are different numbers of VMs for each day, from which each VM is randomly assigned a workload trace for every corresponding day. Table 1 shows the number of virtual machine for each day.

4.2 Result analysis

The simulation of these two different scenarios in Cloudsim toolkit provide following results. Table 2 illustrates the SLATAH, PDM and SLA violation caused by MEDTH, THR, MAD and IQR for random workloads and Table 3 illustrates the results for real workload. Table 2 shows that MEDTH minimizes the percentage of SLATAH, PDM and SLAV than THR, IQR and MAD.

Table 3 shows the results of performance metrics for MEDTH, THR, IQR and MAD on four different dates from 3rd March to 22nd March. Results shows that our MEDTH method provides minimum level of SLA violation with

Table 1. Number of VMs for Real Workload.

Date	Number of VMs
3 March 2011	1,052
6 March 2011	898
9 March 2011	1,061
22 March 2011	1,516

Table 2. Percentage of performance metrics for random workload.

Policies for detection of over utilized and underutilized host	SLATAH	PDM	SLAV
MEDTH	0.91%	0.02%	0.0182%
THR	1.78%	0.05%	0.089%
IQR	3.2%	0.11%	0.03872%
MAD	2.11%	0.09%	0.1899%

Table 3. Percentage of performance metrics for real workload.

Policies for detection of over utilized and underutilized host	Date	SLATAH	PDM	SLAV
MEDTH	3 March, 2011	2.46%	0.05%	0.123%
	6 March, 2011	2.26%	0.06%	0.1356%
	9 March, 2011	2.42%	0.06%	0.1452%
	22 March, 2011	2.39%	0.05%	0.1195%
THR	3 March, 2011	4.95%	0.07%	0.3465%
	6 March, 2011	5.08%	0.07%	0.3556%
	9 March, 2011	5.21%	0.08%	0.4168%
	22 March, 2011	5.11%	0.06%	0.3066%
IQR	3 March, 2011	5.01%	0.07%	0.3507%
	6 March, 2011	5.02%	0.07%	0.3154%
	9 March, 2011	5.27%	0.08%	0.033728%
	22 March, 2011	4.93%	0.06%	0.2958%
MAD	3 March, 2011	5.23%	0.07%	0.3661%
	6 March, 2011	5.26%	0.07%	0.3682%
	9 March, 2011	5.47%	0.08%	0.4376%
	22 March, 2011	5.13%	0.06%	0.318%

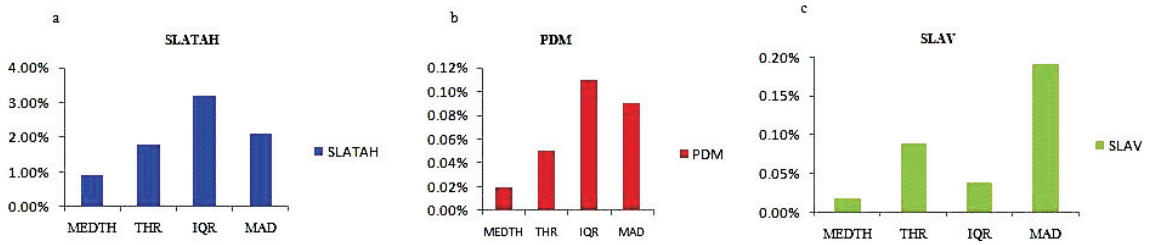

Fig. 2. (a) % age of SLATAH; (b) %age of PDM; (c) % age of SLAV.

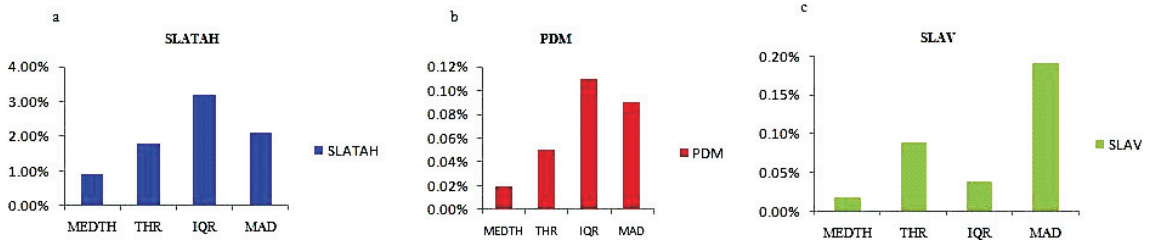

Fig. 3. (a) % age for SLATAH; (b) % age for PDM; (c) % age for SLAV.

lesser performance degradation and lesser SLA time per active host on comparison to other three policies, in which we have used median method to find upper and lower threshold values for detecting over utilized and underutilized host machine. As the workload is varying after every time frame therefore, for dynamic workload this auto adjustment method provide significant results which shows the efficiency of our proposed approach in terms of lesser SLA violation, lesser performance degradation during migration and lesser SLA time per active host as well. These two metrics are equally important for minimizing the SLA violation. Figure 2 and Fig. 3 also illustrate the results of our proposed method which shows that we have achieved our main objective in this work by minimizing the SLA violation.

5. Conclusions

For the optimization of resources and reduction of energy consumption inside cloud data centers, we have used VM consolidation process. But the concept of the virtual machine consolidation for energy efficient performance of cloud is not trivial, as it results the high level of SLA violation that negotiated between users and service providers. In this paper we propose a new method for auto-adjustment of lower and upper threshold values for dynamic consolidation of VMs. Experimental results shows that our proposed method provides minimum level of SLA violation with minimum performance degradation during migration and less SLA time per active host with same level of energy consumption. We have mentioned the results for both random and real workloads. For the future work we will try to investigate the method which provides lesser SLA violation with low energy consumption as well.

References

[1] Ehsan Arianyan, Hassan Taheri and Saeed Sharifian, Novel Heuristics for Consolidation of Virtual Machines in Cloud Data Centers using Multicriteria Resource Management Solutions, *Computer & Electrical Engineering*, (2015).

[2] A. Beloglazov and R. Buyya, Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers, *Concurrenc and Computation: Practice and Experience (CCPE)*, vol. 24, pp. 1397–1420, (2012).

[3] Ripal Nathuji and Karsten Schwan, Virtualpower: Coordinated Power Management in Virtualized Enterprise Systems, In *ACM SIGOPS Operating Systems Review, ACM*, vol. 41, pp. 265–278, (2007).

[4] Akshat Verma, Gargi Dasgupta, Tapan Kumar Nayak, Pradipta De and Ravi Kothari, Server Workload Analysis for Power Minimization using Consolidation, In *Proceedings of the 2009 Conference on USENIX Annual Technical Conference, USENIX Association*, pp. 28–28, (2009).

[5] Anton Beloglazov, Jemal Abawajy and Rajkumar Buyya, Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing, *Future Generation Computer Systems*, vol. 28(5), pp. 755–768, (2012).

[6] 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, *Concurrency and Computation: Practice and Experience*, vol. 24(12), pp. 1397–1420, (2012).

[7] Sina Esfandiarpoor, Ali Pahlavan and Maziar Goudarzi, Structure-Aware Online Virtual Machine Consolidation for Datacenter Energy Improvement in Cloud Computing, *Computers & Electrical Engineering*, vol. 42, pp. 74–89, (2015).

[8] Abbas Horri, Mohammad Sadegh Mozafari and Gholamhossein Dastghaibyfard, Novel Resource Allocation Algorithms to Performance and Energy Efficiency in Cloud Computing, *The Journal of Super Computing*, vol. 69(3), pp. 1445–1461, (2014).

[9] Fahimeh Farahnakian, Pasi Liljeberg and Juha Plosila, LiRCUP: Linear Regression based CPU Usage Prediction Algorithm for Live Migration of Virtual Machine, *Proceedings of the 39th Euromicro Conference Series on Software Engineering and Advanced Applications (SEAA)*, pp. 357–364, (2013).

[10] T. Wood, P. J. Shenoy, A. Venkataramani and M. S. Yousif, Sandpiper: Black-Box and Gray-Box Resource Management for Virtual Machines, *Journal of Computer Networks*, vol. 53, pp. 2923–2938, (2009).

[11] Chaima Ghribi, Makhlouf Hadji and Djamal Zeghlache, Energy Efficient VM Scheduling for Cloud Data Centers: Exact Allocation and Migration Algorithms, In *13*th *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), IEEE*, pp. 671–678, (2013).

[12] Y. Ajiro and A. Tanaka, Improving Packing Algorithms for Server Consolidation, In *Proceedings of the International Conference for the Computer Measurement Group (CMG)*, pp. 399–407, (2007).

[13] Anoton Beloglazov and Rajkumar Buyya, Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data centers, In *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science, ACM*, pp. 4, (2010).

[14] William Voorsluys, James Broberg, Srikumar Venugopal and Rajkumar Buyya, Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation, In *Cloud Computing, Springer*, pp. 254–265, (2009).

[15] K. S. Park and V. S. Pai, CoMon: A Mostly-Scalable Monitoring System for PlanetLab, *ACM SIGOPS Operating Systems Review*, pp. 65–47, (2006).