International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

Energy Efficient Host Overloading Detection Algorithm in Cloud Computing

N. Kumar[1*], R. Kumar[2]

[1]Dept. of Computer Science & Engineering, Gurukula Kangri University, Haridwar, Uttarakhand, India
[2] Dept. of Computer Science, Faculty of Technology, Gurukula Kangri University, Haridwar, Uttarakhand, India

**Corresponding Author: nishant@gkv.ac.in, Tel.: +91-8979798008*

Available online at: www.ijcseonline.org

Accepted: 19/Jul/2018, Published: 31/Jul/2018

Abstract— Cloud computing is now a most popular technology of the present generation. Energy efficiency is big aspect to think as the big data center is consuming a lot of energy to run and to serve their customers. Energy efficient algorithm and techniques are required to reduce the carbon emissions. In this paper we have worked for consolidation of Virtual Machine(VM) by detecting over-utilized hosts by using Pattern matching and reduced number of migrations by taking a new approach of Mode Absolute Deviation. It analyzes the historical data of CPU usages to search the usage pattern of CPU and finds the dynamic thresholds values for migration of virtual machine. The work has been carried out in CloudSim and the results in our work has been better than previous work[1] and we are able to save energy and reduce the number of migrations by using our proposed method.

Keywords— Energy Efficient, host overloading, VM Consolidation, VM Migration, Mode, Cloud Computing.

I. INTRODUCTION

Cloud computing has reached a high level and now an emerging technology which changes the way we live with the computers. Most of the big giants have already shifted to this technology and now offering cloud services to other companies and customers. This transition of computing inspires the 100% utilization of resources. It is required to recheck all the points where energy and Quality of Service(QoS) is affecting and redesign all to save power, one such point is Host Overloading Detection. Virtual Machine(VM) is the technology running behind the cloud, so for running applications or providing any service to the customers, VM will initiate to work and it takes a good amount of energy to run VM's on the host machines. There can be many VM on one host machine, when the VM increases load on the host machine increases and reached to its limits, then hosts started taking huge amount of energy, to reduce the load on the host, VM is required to migrate to the other host. As our aim is to reduce the energy consumption and less carbon footprints to the environment, it is require to balance the tasks and predict to the future load for the VM placement on a host. It is must that host runs on optimal frequency *i.e.* 80% usage of CPU which is a trade-off between the power and performance. Optimal use of processors will leave less carbon footprint and consumes less power and release less heat.

In proposed work the detection of host overloading algorithm is based on calculating the threshold values by analyzing the usage of host in previous cases. By analyzing the previous allocation of work and host throughput we can predict the future workload and decide about overloading of host and prevent it from overloading.

II. RELATED WORK

Efficiency in terms of power is key area, because it incurs environment pollution and costs. A technique to minimize the power consumption is varying the voltage frequency [2]. Another method to save power is turn-off host machines which are not in used, which works on threshold values and if the host is over-utilized then another host will wake-up to take the load. Migration of virtual machines also needs energy[3][4]. Our work has also implemented the minimum migrations and checking the over utilization by pattern matching which is better way.

Pinheiro and Bianchiniet al. [5], technique to save power is by putting hosts to 100% usages and turn-off idle machines, this procedure lacks in power performance trade-off. The proposed algorithm keeps checking the usages of resources to minimize the power usages by switching them off if not in used. This technique has a disadvantage of single point failure(SPF) because load balancing is dependent on it.

Chase et al. [6], it first finds the requirement of resources by applications and can negotiate on SLA as per budget to keep up quality of services. In this paper author has suggested a bidding model, bidding of resources, so that resource utilization can be managed to be 100%. Thus another host can be turned off for power saving. In our work we tried not to put machine on 100% load but operate in on optimal level.

Kusic et al. [7] has used LLC (Limited Look ahead Control) to save power in heterogeneous virtual environment. It has also used the Kalman Filter to predict the future load and managing resources accordingly, this method results in minimizing power usage and also keeping SLA violation under control.

Luo et al. [8] shows a concern about the exponentially increasing power requirement in cloud. In data center, the requirement of power is high and as the load is increasing it is increasing thus increasing the need of cooling machines which again take up a lot of energy and not environmental friendly. This all has become a major concern now. This research directed to research for energy efficiently. Proposed method provides the optimum energy efficiency.

Forsman et al. [9] proposed two techniques push and pull, push strategy is to transfer the load to another less utilized physical machine. In pull strategy the underutilized host itself requests for the workload. This is focused on balancing workload efficiently and re-distribute it if required. Redistribution process involves three steps physical machine state after migration, migration cost and workload prediction. This work states the workload prediction after and before migration. This lacks in when to migrate to make system running on optimal frequency.

Song et al. [10] has proposed a Variable item Size Bin Packaging (VISBP) algorithm to use cloud data-centre resources on the run-time to at the time of requirement for efficient green computing.

Han et al. [11] given the concept of shifting load on few host machines and turning off rest machines to save power. But large VM consolidation results in high SLA violations.

III. Dynamic VM Consolidation

The problem of dynamic VM consolidation identified by A. Beloglazov [14] distributed into following parts:

1. Finding out when a host should be considered as overloaded which requires VM migrations from this host to balance the load.
2. Finding out when a host is being considered as under-loaded and helps to save power by migrating all the VM to other host and switching off the machine.
3. After finding the machine is overloaded, it is require to find out which VM should migrate in-order to balance the load.
4. After selection of virtual machine which needs to migrate, now requires to search a host which can take the load.

The algorithm 1[14] checks the hosts list to find out the overloaded hosts by applying host overloading detection algorithm. After detection the VM selection policy selects the virtual machine which needs to migrate from the host. As the lists of VM which needs to migrate is ready, the requirement of VMs can be found out to analyse and to assign the VM to the new host, this can be done by VM placement policies. The output of the algorithm is migration map which contains the information of the new placement of the VM and selected VM which will be migrate.

Algorithm 1: VM placement Optimization

1. INPUT: hosts
2. Check if host is Overload then
3. Add host in migration list
4. Add/Create Migration Map with all host in migration list
5. Check if host in underload then
6. Add host in migration list
7. Add/Create Migration Map with all host in migration list
8. Return Migration Map
9. Output: Migration Map

A. Proposed Host Overloading Detection Approach

Pattern based threshold approach for finding over-utilized host machines.

As per [13], a heuristic has been proposed to find the right time to migrate the virtual machine (VM), it is based on setting lower and upper threshold and keeping all the VM on the host machine between these two thresholds for the optimal performance. If CPU utilization exceeds the upper threshold than there is a provision to migrate some of the virtual machine to other host. If CPU utilization lowers down from the lower threshold then in that case, migrating all the virtual machines to other hosts and putting the system in the sleep mode to save the power.

Fixed values of thresholds are not preferred for constantly changing environment where workload is varying frequently. So, setting up thresholds dynamically requires to learn and analyse the historical data for pattern of usages by virtual machines. Dynamic threshold provides an optimal performance and it requires to get statistical analysis of historical or logged data. This procedure learns the pattern of frequencies of which the host is being used mostly and based on that it sets the upper threshold and lower threshold dynamically. These thresholds marks either host is overload or not. The Mode Absolute Deviation (MOAD) is a measure of statistical distribution. For a univariate data set X1,X2, ...,Xn, the MAD is defined as the median of the absolute.

MOAD= modei(|Xj- mode(Xj)|)

That is, starting with the data mode, the MOAD is the mode of absolute values. The upper threshold (TU) is

TU= 1- s . MOAD ,

where ‘s’ is use to adjust the safety of method.

Public class PatternModMmt {


```
public static void main(String[] args) throws IOException {
        String workload = "20110303";
        String AllocationPolicy = "pattern";
        String SelectionPolicy = "modmmt";
        String parameter = "";
        new PlanetLabRunner(
                enableOutput,
                outputToFile,
                inputFolder,
                outputFolder,
                workload,
                AllocationPolicy,
                SelectionPolicy,
                parameter);
    }
}
```

Further is calls to classes that is

```
public class PowerVmAllocationPolicyMigrationPattern extendsPowerVmAllocationPolicyMigrationAbstract{
publicclass publicVmSelectionPolicyModifiedMinimumMigrationTime extends PowerVmSelectionPolicy
```

B. VM Selection Policy: The Minimum of Migration Policy(MMT)

As per [9], MMT policy is the most efficient policy to migrate the virtual machine, it decides which VM needs to migrate in-order to get the high efficiency. To decide which one to migrate it sorts all the present VM in decreasing order. Then it keeps checking repeatedly, for the virtual machine which will be migrate for the based on fulfillment of following two conditions-

1. VM utilization needed to be higher than the host's overall utilization and the upper utilization threshold difference.
2. If VM is migrated, then the difference of upper threshold and the new utilization should be the least across all the VMs.

If any VM do not fits into the criteria, its simply takes the VM of higher utilization and removes it from the VMs list, and proceed with allocation in the another host. It keeps working on same process until the utilization comes down to upper utilization threshold.

1. **Input- hostlist**
2. ***foreach*** host in hostlist ***do***
3. get_vmlist
4. sort_vmlist_decreasingorderutilization()
5. get_hostutilization()
6. bestfituilization=max
7. **while** hostutilization > threshup **do**
8. **foreach** virtmachine in vmlist **do**
9. **if** virtmachine.getutil() > hostutilization+threshup **then**
10. k= virtmachine.getutilization() – hostutilization + threshup
11. ***if*** k < bestfitutilization ***then***
12. bestfitutiliztion = k
13. bestfitvirtmachine= virtmachine
14. ***else***
15. ***if*** bestfitutilization=maximum ***then***
16. bestfitvirtmachine= virtmachine
17. ***break***
18. hostutilization=hostutilization – bestfitvirtmachine.getutilization()
19. Add bestfit_virtual machine to the Migration List
20. Remove virtual_machine from vmlist
21. ***if*** hostutilization < threshlow ***then***
22. Add low utilization virtual_machine into Migration List
23. Remove virtual_machine from vmlist
24. ***return*** virtual_machine_migration_list
25. **Output: Migration List**

Algorithm 1 Minimum Migration Time

C. Host Under-Loading Detection

To find out the under-loaded hosts, after the new placement of VM, selected by overloading detection algorithm to another host. The system re-analyzes the utilization and minimum utilized host tries to place its VM to other hosts, making sure that the other host will not be over-utilized with this placement. As the new placement is found then the VM will be migrated to other host and previous host will put to sleep to save energy. This process is iteratively running for all the hosts which are not found overloaded.

D. VM Selection Policy: The Minimum of Migration Policy(MMT)

To solve the VM placement problem we are using Modified Best Fit Decreasing(MBFD) algorithm, firstly it is required to sort all the virtual machines in the decreasing order as per their current utilization and allocate VM to host that provides the least impact on the performance and power consumption caused by the allocation.

IV. EXPERIMENTAL SETUP

Setting up the complete environment of cloud computing to check the algorithms is a very costly affair, cloudsim, cloud simulator provides the environment of cloud computing. The CloudSim[15] is used for simulation platform as this is being used worldwide for research purposes. CloudSim has a great advantage over other available cloud simulation platform(SimGrid, GangSim), it provides the facility of modelling of virtualized environment and supports on-demand resource provisioning. It also, enables the energy-aware simulation capability.

Simulation has been done with 800 physical nodes, with two server configuration, 50% each HP ProLiant ML110 G4 and

HP ProLiant ML110 G5. The CPU frequency of the server is mapped on MIPS, 1860 each core to G5 and 2660 to G4. Each having the network bandwidth of 1GB/s.
Experiments are done on the data provided of CoMon project, a monitoring infrastructure for PlanetLab [16]. Ram is divided as per the number of cores and requirements of VM as follows:
Instance 1: 2500MIPS, 0.85GB
Instance 2: 2000 MIPS, 3.75 GB
Instance 3: 1000 MIPS, 1.7 GB
Instance 4: 500 MIPS, 613 MB
Initially VM allocation is done as per requirement defined in above VM types. Later VMs automatically adjust to use less resources according the workload and creat options for dynamic consolidation.

V. Simulation Results and Analysis

The workload used for this setup is 20110303 in which there are 1052 virtual machines deployed on 800 host machines. Simulation is compared with all the existing overloading detection algorithm and proposed algorithm to show the comparative analysis.

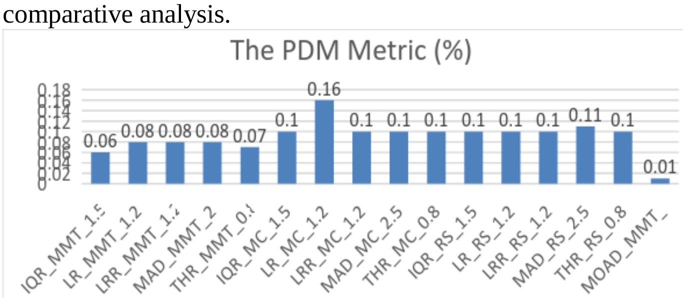

Fig 1 The PDM Metric

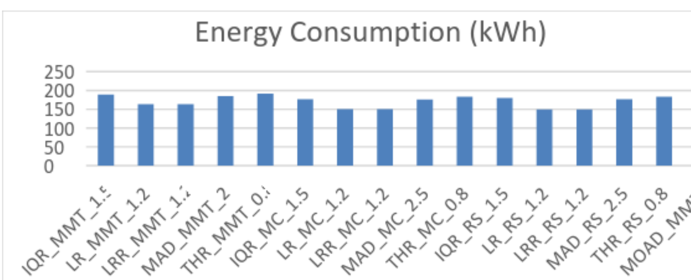

Fig 2 Energy Consumption

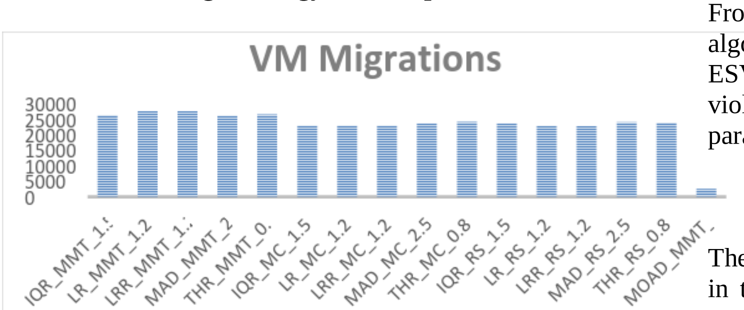

Fig 3 Virtual Machine Migrations

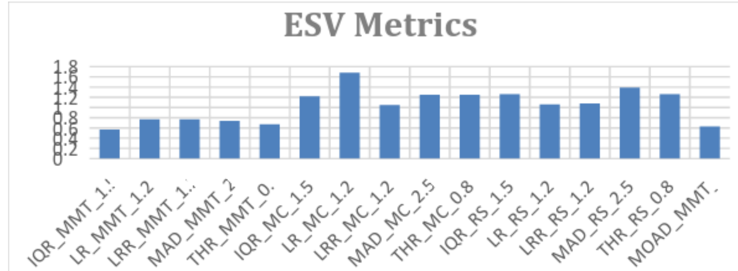

Fig 4 ESV Metrics

Proposed algorithm results shows a huge difference in power consumption which is below 100 while the lowest is LS_MMT 149kwh and highest is IQR_MMT 188kwh, it has also managed to reduce the migration of virtual machines and SLA degradation due to migration (PDM Metrics) while ESV is slightly up comparatively to IQR_MMT.

Table 1 Comparative Analysis

Policy	Energy Consumption (Kwh)	ESV ($x10^{-3}$)	VM Migrations
MOAD_MMT_2.5	92.86	0.63	2546
LRR_RS_1.2	149.41	1.08	22858
LR_RS_1.2	149.70	1.06	22915
LR_MC_1.2	150.33	1.68	23004
LRR_MC_1.2	150.33	1.05	23004
LR_MMT_1.2	163.15	0.77	27632
LRR_MMT_1.2	163.15	0.77	27632
MAD_MC_2.5	176.13	1.25	23691
MAD_RS_2.5	176.50	1.39	24086
IQR_MC_1.5	177.10	1.22	23035
IQR_RS_1.5	179.99	1.26	23726
THR_RS_0.8	183.30	1.26	24010
THR_MC_0.8	183.61	1.25	24235
MAD_MMT_2.5	184.88	0.74	26292
IQR_MMT_1.5	188.86	0.57	26476
THR_MMT_0.8	191.73	0.67	26634

From the above comparison it is clear that the proposed algorithm is efficient in terms of power but slight increase in ESV. The trade-off between power efficiency and SLA violation can be controlled by the varying the safety parameter of MOAD_MMT algorithm.

VI. Conclusions

The results clearly shows the proposed model is much better in terms of power saving and making a huge difference in comparison to the existing algorithm. Using the approach we are able to achieve a good energy efficiency. The model though results in more SLA violation which can be adjusted by safety parameter.

References

[1] O. Sharma and H. Saini, "VM Consolidation for Cloud Data Center Using Median Based Threshold Approach," *Procedia Comput. Sci.*, vol. 89, pp. 27–33, 2016.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," ACM SIGOPS Oper. Syst. Rev., vol. 37, no. 5, p. 164, 2003.

[3] T. Guerout, T. Monteil, G. Da Costa, R. Neves Calheiros, R. Buyya, and M. Alexandru, "Energy-aware simulation with DVFS," Simul. Model. Pract. Theory, vol. 39, pp. 76–91, 2013.

[4] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," NSDI'05 Proc. 2nd Conf. Symp. Networked Syst. Des. Implement., no. Vmm, pp. 273–286, 2005.

[5] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5931 LNCS, pp. 254–265, 2009.

[6] E. Pinheiro and R. Bianchini, "Load balancing and unbalancing for power and performance in cluster-based systems," … Syst. Low Power, pp. 1–8, 2001.

[7] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, "Managing energy and server resources in hosting centers," Acm Sigops, vol. 35, no. 5, p. 103, 2001.

[8] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and Performance Management of Virtualized Computing Environments Via Lookahead Control," 2008 Int. Conf. Auton. Comput., pp. 3–12, 2008.

[9] J. Luo, X. Li, and M. Chen, "Hybrid shuffled frog leaping algorithm for energy-efficient dynamic consolidation of virtual machines in cloud data centers," Expert Syst. Appl., vol. 41, no. 13, pp. 5804–5816, 2014.

[10] M. Forsman, A. Glad, L. Lundberg, and D. Ilie, "Algorithms for automated live migration of virtual machines," J. Syst. Softw., vol. 101, pp. 110–126, 2015.

[11] W. Song, Z. Xiao, Q. Chen, and H. Luo, "Adaptive resource provisioning for the cloud using online bin packing - Wagner," Comput. IEEE Trans., vol. X, no. X, pp. 1–14, 2013.

[12] G. Han, W. Que, G. Jia, and L. Shu, "An efficient virtual machine consolidation scheme for multimedia cloud computing," Sensors (Switzerland), vol. 16, no. 2, pp. 1–17, 2016.

[13] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-Efficient Management of Data Center Resources for Cloud Computing : A Vision , Architectural Elements , and Open Challenges Clou d Computing and D istributed S ystems (CLOUDS) Laboratory Department of Computer Science and Software Engineering The," Univ. Melbourne, Aust., no. Vm, pp. 1–12, 2010.

[14] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Futur. Gener. Comput. Syst., 2012.

[15] 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," Concurr. Comput. Pract. Exp., vol. 24, no. 13, pp. 1397–1420, 2012.

[16] A. Khajeh-Hosseini, D. Greenwood, J. Smith, and I. Sommerville, "The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise," Softw. - Pract. Exp., vol. 43, no. 4, pp. 447–465, 2012.

[17] K. Park and V. S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," ACM SIGOPS Oper. Syst. Rev., vol. 40, no. 1, pp. 65–74, 2006

Authors Profile

Dr. RajKumar pursued Bachelor of Science and Master of Science from Gurukula Kangri Vishavidyalaya, Haridwar. MCA from IGNOU. He completed his PhD in Wireless Sensor Network from Gurukula Kangri Vishwavidyalaya, Haridwar. He has published more than 15 research in different areas in reputed journals(SCI, Scopus Indexed) and conferences including IEEE. He has more than 20 years of teaching experience and more than 10 years experience in research. His area of research in the wireless sensor netowork, software product line, cloud computing, software testing.

Mr. Nishant Kumar pursued Bachelor of Engineering & Technology from Gurukula Kangri Vishavidyalaya, Haridwar. M.Tech Computer Science from SRM University, Chennai. He is pursuing his PhD in Cloud Computing from Gurukula Kangri Vishwavidyalaya, Haridwar. He has published more than 12 research papers in reputed journals(including Scopus Indexed) and conferences including Springer. He has more than 7 years of teaching experience and more than 3 years experience in research.

