An Improved Energy-Efficient Policy for Overload Host Detection in Cloud Environment

Satveer
Department of Computer Science
Gurukul Kangri Vishwavidhyalaya, Haridwar, India

Mahendra Singh
Department of Computer Science
Gurukul Kangri Vishwavidhyalaya, Haridwar, India

ABSTRACT

As the cloud users and their data are growing very rapidly, the cloud service providers are also establishing the power hungry datacenters across the world to grant all types of cloud services and to store the data. Cloud providers are facing challenging problems of energy and SLA tradeoff, minimization of operating cost and CO_2 emission in environment. The VM consolidation is extremely efficient and proactive approach for saving the energy online with dynamic workloads in cloud datacenters. In this paper, we have proposed a new host overload detection policy for reducing the energy consumption with low SLA violation. The simulation results with cloudsim guarantees for minimizing the energy consumption and maximizing the SLA by preserving the VM migration, average SLAV, frequency of host shutdown in comparison with state of arts.

Keywords

Energy, SLAV, Host, Virtual Machines (VM), VM consolidation (VMC), Datacenters (DC).

1. INTRODUCTION

Today, in twenty-first century the peoples and enterprises are continuously shifting their business on cloud to make the business fast and platform independent. The search engines, web-mail, facebook, twitter and YouTube are general services of cloud [1]. United States Data Center Energy Usage Report shows that energy consumption of the cloud datacenters will be approximately 73 billion KWh by 2020 [3]. On other side, Global Sustainability Initiative (GeSI) stated that the cloud DCs will be amenable for 18% CO2 ejection and CO2 emission will be 1430 million metric tons in coming next two years. This massive expenditure of energy is unacceptable sign for environmental and operating cost of the DCs. In cloud, virtualization is the key technology to minimize the energy misuse by sharing the same resource among the multiple guest OS called Virtual machines (VMs) [4]. Virtualization helps to create multiple heterogeneous VMs and run them in parallel manner on a host. The Virtual Machine Consolidation (VMC) is most encouraging approach in order to downplay the energy misuse by optimizing the resource usage enabled by virtualization. VMC makes possible to keep the host shutdown or in energy saving mode to the idle or under loaded host, towards eliminating the idle power usage by reallocating the VMs employing according to their current resource usage and running workloads [5]. Whenever the workload increases more number of hosts awake up from energy saving mode to activated mode. Wherever, VMC tries to lower the energy misuse, it also affects the Service Level Agreement (SLA) negatively due to high VM migrations. Exploring the solutions for achieving healthy tradeoff between energy and SLA is very crucial and important challenge for both the industry academia researchers.

In this paper, we proposed Cm (cubic mean), a new host overload detection policy to acquire the best trade-off between the energy consumption and SLAV. The proposed policy is designed based on analyzing the historical data of CPU utilizations by the VMs of the host.

Rest of the paper is formed as follows. In Section 2 we review significant related state of art works. Section 3 presents the proposed overload host detection algorithm. Portion 4 discuss the evaluation methodology and performance metrics. Section 5 explains the simulation results of the proposed solution and finally we conclude the paper in section 6.

2. RELATED WORK

The VMC is the most prominent and widely used approach in literature to reduce the power usage. This section gives a brief review about the various existing VMCs, which primarily consider the energy and SLA trade-off. A VM consolidation model was proposed by Bruno et al. using AI based on Pseudo-Boolean (PB) Constraints in [6]. A PB Constraints is used to optimize costs. The proposed work reduces power by optimizing the CPU and RAM. Anton has developed the fixed utilization threshold (THR) scheme by setting up upper threshold and lower threshold for finding the overloaded and under loaded host [7]. Whenever the current utilization of the host goes below the lower threshold the host considered as the under loaded, and they migrate all the running VMs onto another normal loaded hosts, so that host can stay in power saving mode. While on other side, if the current utilization of the CPU goes high from the upper threshold then the host considered as the overloaded host, and make the migration of some required VMs to another normal loaded host till the host becomes normal and do not leads to SLAV.

A static threshold based strategy has been proposed by Zhu et.al., to spot out the overload host in DC. If the current utilization of CPU is beyond the 85% of its total CPU capacity, then host is declared as the overloaded host [8]. Moreover, as the current decade research is going on further, the researchers are using the historical data of the VMs stored while VMs are alive. Authors have studied problem with static threshold and recommended dynamic system based on the workload figure presented by the applications. Anton proposed a novel dynamic threshold [9] strategy and it used random variable of accounting the total CPU utilizations of the host. In order to find the overloaded host in the DCs, Anton also proposed some more dynamic threshold solutions based on predication of CPU utilization. The proposed overload host detection algorithms are:

RLL, LR, IQR and MAD. To select the required VMs from overloaded host author also proposed three algorithms for VMs selection: MC, RS, and MMT. The results of simulation show that the MMT is more efficient then MC and RS strategies and LR outplay the previous static threshold and adaptive threshold based policies [10].

In order to gain the more effective tradeoff of energy and SLA, Phinheiro proposed the eco-friendly DC management by dynamically switching the servers into ON-OF for saving the power in [11]. However, the system is not flexible to adjust the current dynamic resources intensive workload situation at reasonable rate power dissipation and SLA violation. The proposed solutions are able to save energy in static environment by setting up static lower and upper thresholds; however these are not convenient according to the modern exhaustive forthcoming workload they need be more dynamic in order to adopt the system perfectly.

3. PROPOSED POLICY FOR HOST OVERLOADED DETECTION

We proposed the Cubic mean (Cm) policy to make system more energy efficient under the low SLA violation constraint. This policy takes the host list as input and decides dynamically whether a host is overloaded or not. We have used two existing VM selection policies MMT and Mu to select the required VMs from the overloaded host. The objective of the proposed Cm policy is to set adaptive CPU utilization threshold by accounting VMs' average utilization. The cubic mean is calculated by dividing the total requested MIPS by VMs with the total number of VMs running on that host. Suppose that we have set U of VMs CPU utilizations, {U1, U2, U3, …., Un}. We calculated the total sum of CPU utilizations of VMs by sorting values in increasing order. Let Si is the sum of VM's CPU utilization on host Hi in time t as:

$$\mathrm{Si} = \sum_{j=1}^{Ni} U\mathrm{ij},_{\mathrm{t}} \qquad (1)$$

Where, N is the number of VMs and t is time. Next we have calculated Cubic mean as:

$$\mathrm{Cm} = \sqrt[3]{\frac{1}{N}\sum_{t=1}^{N}(S_i)^3} \qquad (2)$$

After calculating the cubic mean, we predict the upper threshold for the current host. According to [10] the forecasting CPU utilization threshold T can be estimated as:

$$\mathrm{T} = 1 - \mathrm{s.Cm} \qquad (3)$$

Where, s is the constant value and known as safety parameter. If, T crossover the current CPU utilization of the host Hi, then it is considered as the overloaded host. The pseudo code of the policy is given in algorithm 1,

Algorithm. 1: Cubic Mean Host Overload Detection Scheme

```
1: Input: Host List
2: Output: Whether a host is overloaded or not
3: $S_i \leftarrow 1$,
4: for each VM of host $H_i$ in VM List do
5: $S_i \leftarrow$ get current demanded MIPS by VMs on Host $H_i$
6:Cm$\leftarrow \sqrt[3]{\frac{1}{N}\sum_{t=1}^{N}(S_i)^3}$
7: T$\leftarrow$ 1-s.Cm
8: if T > Current utilization of the host Hi then
9: return true
10: else false
```

4. EVOLUTION METHODOLOGY AND EFFICIENCY METRICS

4.1 Simulation Testbed

To evaluate proposed policy our essential need is to implement the Infrastructure- as- service of the cloud. Although to examine the policy on real infrastructure in repetitive way is not possible and highly risky. Since, we prefer the cloudsim simulator tool to test the performance efficiency of the proposed policy. It is modern and extensively used to optimize the power usage in DCs and it also enables for creating cloud entities i.e. hosts, VMs, DCs. It also support VM's allocations, selection, migrations, power models, and with providing real workloads. We used the power models as in the [12]. To evaluate the performance of the algorithm we have deployed 800 heterogeneous servers, which have real configuration. The configuration features and energy consumption of the server at different workloads are also presented Table 2 and Table 1 respectively. The traced the energy consumption of each host according its CPU utilization have gathered from SPEC power [12].

Table 1: The power consumption of underlying hosts at various level of workload in (KWh)

Server Type	**Idle**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Fujitsu M1	13.3	18.3	21.1	23.4	26.5	29.6	34.7	40.7	46.8	57.4	60
Fujitsu M3	12.4	16.7	19.4	21.4	23.4	26.1	29.7	34.8	41	47.1	51.2
Hitachi TS10	37	39.9	43.2	45.5	48.8	52.8	57.8	65.1	73.8	80.8	85.2
Hitachi SS10	36	38.8	41.2	43.7	46.3	49.4	53.1	58.8	64.2	67	69.7

Table 2: The Characteristics of the underlying hosts

Server Type	CPU (Xeon)	Cores	MIPS	Ram (MB)	BW (MB)
Fujitsu M1	1230	4	2700	8192	1024
Fujitsu M3	1230	4	3500	8192	1024
Hitachi TS10	1230	4	3500	8192	1024
Hitachi SS10	1230	4	3600	8192	1024

Table 3: VM Specifications (Amazon EC2 VM types)

VM Types instance	# of Cores	MIPS	RAM (MB)
High-CPU	1	850	2500
Extra-Large	1	3750	2000
Small	1	1700	1000
Micro	1	613	500

4.2 Real Workload

The important key point is that, we have conducted the experiment with using the real workload. This workload is captured in April 2011 as the part of CoMon project, monitoring infrastructure for PlanetLab [13]. The preferred real workload consist thousands heterogeneous VMs CPU utilizations data from more than 500 different servers. The detail features of the data are discussed in [10]. The VMs types and their specifications are shown in Table 3.

4.3 Peformance Metrics

Reducing the power wastage under the QoS constraints is very tedious and critical in cloud DCs and QoS generally modeled in the form of SLA. It is prerequisites to explore the basic metrics that can be helpful to figure out the energy and SLA obsessed. Therefore, we used such metrics that can demonstrate the efficiency of any policy in terms of energy and SLA.

4.3.1 Energy Consumption

The energy consumption is the total amount of electrical power obsesses by the deployed DCs.

4.3.2 SLA Violation (SLAV)

The QoS management is the key need to maintain the SLA in cloud and SLA is an agreement between user and CSP. To evaluate the SLA impairment, we have considered the SLA Violation metric and can be calculated as:

$$SLAV = SLATAH*PDM \qquad (4)$$

Where,

$$SLATAH = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{si}}{T_{ai}} \qquad (5)$$

$$PDM = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{dj}}{C_{rj}} \qquad (6)$$

Where Tsi is the time unit of SLAV at host Hi, Tai is the active time of the host Hi, M is the total number of VMs, Cdj is performance degradation by VMj due to migration, Crj is the total requested MIPS by VMj Urj

4.3.3 Average SLA Violation Percentage

This metric is the percentage of CPU MIPS that has not been assigned to the task, when requested and arising in SLA violation. The average SLAV can be calculated as

$$SLA = \frac{\sum_{j=1}^{M}\int Urj(t)-Uaj(t)dt}{\sum_{j=1}^{M}\int Urj()} \qquad (7)$$

Where, Urj (t), Uaj(t) is the total requested and allocated MIPS by all VMs, M is the no. of VMs.

4.3.4 Number of VM Migrations

This metric shows the total number of total migrated VMs from one host to another host.

4.3.5 Number of Host Shut Down

The number of host shut down is frequency of host's reactivation occurring during the simulation.

4.3.6 Performance Metric (Pertric)

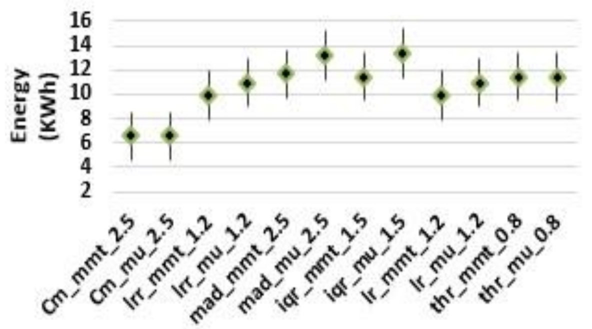
To measure the overall performance of the deployed system we imported new performance metric [14]:

$$Pertric = SLAV * HS * E \qquad (8)$$

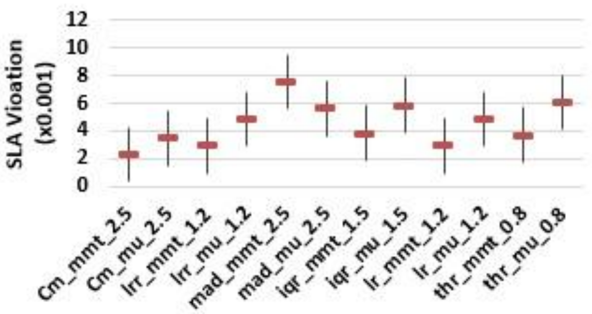
Where, SLAV is the SLA violation, HS show the total number of host shutdown, and E is the total amount of energy consumed in DC.

5. SIMULATION RESULTS

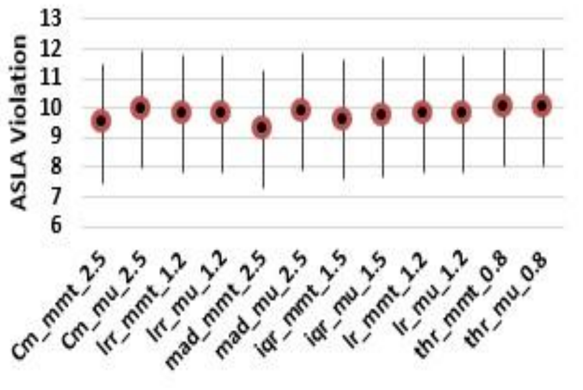
In order to evaluate the performance, the proposed algorithm Cm was simulated together with the VM selection variants MMT and Mu using the cloudsim. The simulation results of the Cm_MMT and Cm_Mu were compared with the existing host overload detection policies such MAD, IQR, LR, LRR and THR with MMT and Mu as VM selection variant. The used performance metrics are energy consumption, SLAV, ASLV, number of VM migrations, pertric and no. of host shutdown. The real workloads have been used to evaluate the performance of the algorithm. The experimental results show that the proposed policy Cm performs better by maintaining the best tradeoff between energy consumption and SLAV in comparison with other proposed algorithms.



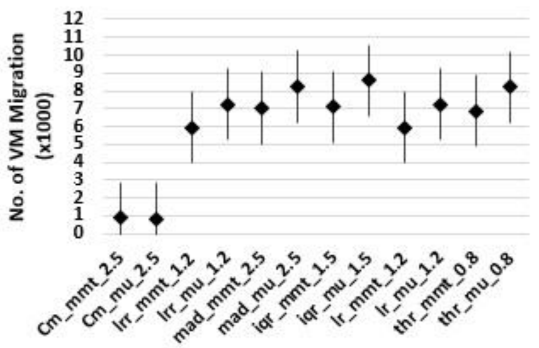
(a)Energy Consumption



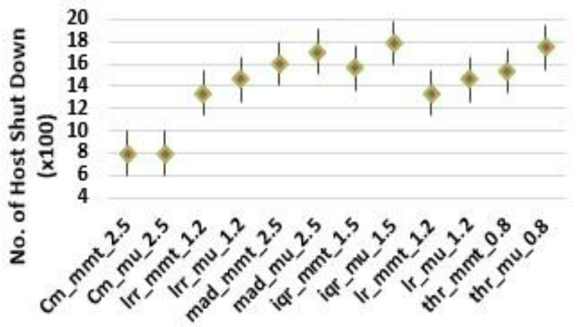
(b) SLA Violation



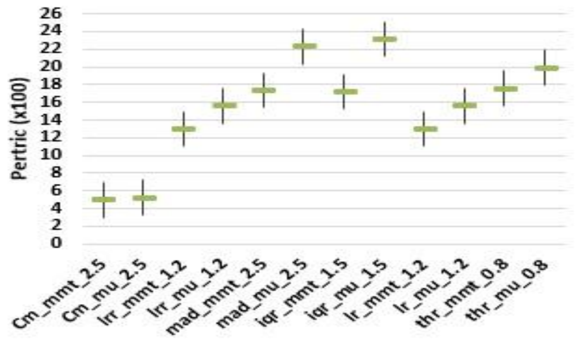
(c) Average SLAV



(d) VM Migrations



(e) No. of Host Shutdown



(f) Performance Metric

6. CONCLUSION

Energy and SLA management are the serious concern for the cloud providers as well as for academia and industry researchers to maintain the environment. A new host overload detection policy named as Cubic mean (Cm) has been proposed for energy and SLA management for virtualized datacenters. The proposed policy is based on the idea of calculating the cubic mean to predict the upper threshold utilization of the current host. The proposed policy reduces the power consumption at reasonable price in terms of SLA. The simulation results show that developed algorithm outperform many of the existing policies with regard to energy consumption, SLAV, average SLAV, VM migrations and number of host shutdown. Thus, the designed policy also enhances the overall performance of the cloud data centers. The work to further improve the VM selection policy is going on.

7. REFERENCES

[1] Jain Li, Kai Shuang, Sen Su, and Qingjia (2012) “Reducing Opertional Cost through Consolidation with Resource prediction in the Cloud” 12th IEEE/ACM International Symposium on cluster, Cloud and Grid Computing 987-0-7695—4691-9/12 $26.00 2012 IEEE, DOI 10.1109/CCGrid.2012.50

[2] (2015) Cisco global cloud index: Forecast and methodology, 20142019. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/

[3] Arman Shehabi, Sarah Smith and Dale Sartor, (2016) “United States Data Center Energy Usage Report” Ernest Orlando Lawrence Berkeley National Laboratory, LBNL-1005775 https://www.connaissancedesenergies.org/

[4] Fahimeh Farahnakian, Tapio Pahikkala, Pasi Liljeberg, and Juha Plosila (2013), Energy Aware Consolidation Algorithm based on K-nearest Neighbor Regression for Cloud Data Centers. IEEE/ACM 6th International Conference on Utility and Cloud Computing. CFP13UCC-USB/13$26.00 2103 IEEE, DOI 10.1109/UCC.2013.51

[5] RNathuji and K. Schwan (2007) Virtual Power: Coordinated Power Management in Virtualized Enterprises System, Proceedings of the 22st ACM Symposium on Operating System Principles (SOSP' 07),pp.265-278

[6] Ribas B.C., Suguimoto R.M., Montaño R.A.N.R., Silva F., de Bona L., Castilho M.A. (2012) On Modelling Virtual Machine Consolidation to Pseudo-Boolean Constraints. In: Pavón J., Duque-Méndez N.D., Fuentes-Fernández R. (eds) Advances in Artificial Intelligence – IBERAMIA 2012. IBERAMIA 2012. Lecture Notes in Computer Science, vol 7637. Springer, Berlin, Heidelberg

[7] A.Beloglazov, Jemal Abawajy, and R. Buyya (2012) Energy-aware resource allocation heuristics for efficient management of datacenters for cloud computing, Journal of Future Generation Computer System, pp.755-768

[8] X. Zhu, D. Young, B. J. Watson, Z. Wang, J. Rolia, S. Singhal, B. McKee, C. Hyser, D. Gmach, and R. Gardner (2008). 1000 islands: Integrated capacity and workload management for the next generation data center. in Autonomic Computing, 2008. ICAC'08. International Conference on. IEEE, pp. 172–181.

[9] Anton Beloglazov and Rajkumar Buyya (2010) Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers MGC '2010, ACM 978-1-4503-0453-5/10/11

[10] Anton Beloglazov and Rajkumar Buyya (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Published in Journal Concurrency and Computation: Practice and Experience, vol. 24, no. 13, pp. 1397– 1420.

[11] E.Pinheriro, R. Bianchini, E.V. Carrera, and T. Health (2001). Load balancing and unbalancing for power and performance in cluster-based systems. Workshops on compilers and operating systems for low power, Barcelona, pp.182, Spain.

[12] Dianne Rice, Diana Cercy, Jason Glick, Cathy Sandifer and Bob.Spec.org/power_ssj2008/results/

[13] K.S. Par and V.S. Pai. (2006). CoMon: a mostly-scalable monitoring system for PlanetLab. Published in Newsletter ACM SIGOPS Operating System

[14] S. Esfandiarpoor, A. Pahlavan, and M. Goudarzi, (2105). Structure-aware online virtual machine Review, pp. 65-47, 2006 consolidation for datacenter energy improvement in cloud computing. Journal of Computers & Electrical Engineering, vol. 42, pp. 74–89.