Dyanamic Load Balancing With Cost And Energy Optimization In Cloud Computing

Jasnil Bodele
MTech, CSE
IIT Roorkee
Roorkee, India

Anil K. Sarje
Professor, ECD
IIT Roorkee
Roorkee, India

***Abstract*— Cloud Computing is becoming a backbone of nearly all enterprises. Companies are using cloud for their extensive computational work as well as data storage. However the prime objective is to outsource the cost of infrastructure and maintenance, as companies want to focus on their core line of business. Hence they require not only an efficient cloud infrastructure but also a cloud which is optimized in terms of cost. This requirement of companies provides us the idea of creation of an algorithm, which will be both able to balance Load as well as optimize total cost of maintenance.**

Keywords— Cloud computing; Dynamic Load Balancing, Cost/Energy optimization, SLA (service level agreement) violation, VM (Virtual machine) migration

I. INTRODUCTION

Cloud provides on-demand and pay-as-per-requirement business model. There are three levels of services in cloud computing, Infrastructure as a service (Iaas), Platform as a service (Paas) and Software as a service (Saas). Iaas is very similar to a private cloud, except for the fact that clients do not own the server. Instead, a third party allows client to install client's own virtual server on their IT infrastructure in exchange for a rental fee. Paas provides developers with a framework that they can build upon their own applications or customize existing Saas applications. In Saas, client uses the graphical interface of the application and this application is managed and hosted by a third party. The focus of this paper will be on Iaas.

Generally a cloud is over provisioned so that it can handle higher workloads. These high workloads may occur occasionally [10]. This over provisioning makes cloud expensive and space consuming. Space consuming in the sense space/room required to keep a server. Over provisioning leads to low resource utilization, wastage in energy and management overhead. American society of heating, refrigerating and air-conditioning engineers (ASHRAE)[1] 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]. Thus reduction in this energy cost is essential for overall cost reduction. The prime focus of this paper is energy optimization.

Data collected from more than 5000 production servers over a six-month period have shown that although servers usually are not idle, the utilization rarely approaches 100% [3]. Most of the time the server is only 10-50% utilized leading to wastage of energy. Even a completely idle server consumes 70% of its peak power [4]. So keeping the server underutilized is inefficient from energy saving point of view. Moreover for each watt of power consumed by computing resources, additional 0.5-1 W is required for the cooling system [5]. The high energy consumption also leads to release of green house gas CO_2 [6]. So, this wastage of energy is not only cost ineffective but also harmful to the environment.

Such wastage of energy can be avoided by putting the idle servers to sleep mode. Servers whose utilization goes below a certain threshold can be put to sleep, if it is possible to move all of their VMs to some other servers without making them overloaded. However if VMs are migrated extensively this may lead to performance degradation. The response time of the applications running on VM under migration will increase which can lead to Service Level Agreement (SLA) violation established between cloud provider and the client. A SLA is contract/agreement between Client and the service provider stating the terms and conditions of services, payments and penalties etc. As an example, if a certain agreed service is not provided within some specified time interval some penalty needs to be paid by the service provider to the client for delay in service. Thus too much VM migration for saving energy, may also lead to increased downtime of service provided by that VM under migration, thus attracting SLA violation [10]. Therefore there is a tradeoff between reduction of energy consumption and reduction of SLA violation due to VM migration.

Even if a server is 100% utilized then also SLA violation may occur [10]. If the demand of the

application is satisfied then there is no SLA violation, but if the computational demand of the application is not met and the cloud's VM is restricting its demand by getting 100% utilized, then it can be an SLA violation (if stated in agreement). These kinds of SLA's are defined in agreement because cloud may be unknowingly restricting the computing demand of the application. Hence if the utilization goes above an upper threshold there are chances that it may get 100% utilized, which may lead to SLA violation and thus some of its VMs must be migrated to less loaded servers.

In this paper we propose an Optimized Load Balancing algorithm (OLB) which not only balances the load among the servers but also reduces energy consumption and SLA violation.

The organization of paper is as follows Section 2 discusses method for host overloading detection. Section 3 focuses on VM selection strategy for migration. Section 4 discusses the profit of finding target host for the selected VMs. Host underloading detection is discussed in Section 5. Calculation of imbalance factor is given in Section 6. Simulation results are presented in Section 7 and conclusion is given in Section 8.

II. HOST OVERLOADING DETECTION

We need to migrate VMs from over loaded host so that it will not cause SLA violation. Whenever the CPU utilization of a host crosses a threshold it is regarded as overloaded. We can have a static threshold defined for this purpose. But as discussed in [10], under dynamic workload environment static threshold will not give good performance. Therefore need some dynamic/adaptive threshold based on history of utilization.

Three methods are suggested in the literature for adjustment of upper threshold based on historical data of CPU utilization: Median Absolute Deviation (MAD) [10], Interquartile Range (IQR) [14] and Local regression (LR) [11].

Performance evaluation has shown that LR outperforms MAD and IQR [10]. Hence we use LR for deciding whether host is overloaded or not.

LR [11] is proposed by Cleveland. The main idea of this method is fitting simple models such as straight line or some well known curve to localized subsets of data to build up a curve that approximates the original data.

III. VM SELECTION POLICIES FOR MIGRATION

After the host is found to be overloaded, the next step is to select the VMs to be migrated away from that host. There are various polices for this, namely Random Selection (RS) policy, Maximum correlation policy (MC) and Minimum Migration Time (MMT).

Random selection (RS) [10] as the name suggests selects VM at random from the host selected for VM migration. This policy is fairly simple to implement and also running time is quite low.

Maximum correlation (MC) policy is based on the idea proposed by Verma et al. [17]. The higher is correlation between the resource usage by applications running on an oversubscribed server, the higher is the probability of the server overloading. According to this idea, we select those VMs to be migrated that have the highest correlation of the CPU utilization with other VMs. Details of calculation of correlation can be found in Verma et al. [17]. It is shown in [10] that MC is not as good as MMT.

Minimum Migration Time (MMT) [10] policy migrates a VM 'v' that requires the minimum time to complete a migration relatively to the other VMs allocated to the host. The migration time is estimated as the amount of RAM utilized by the VM 'v' divided by the spare network bandwidth available for the host j currently hosting VM 'v'. Let Vj be the set of VMs currently allocated to the host j. The MMT policy finds a VM 'v' that satisfies condition 3.1.

$$v \in Vj \mid \forall a \in Vj, \frac{RAMu(v)}{NETj} \leq \frac{RAMu(a)}{NETj} \qquad (3.1)$$

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

If the host j is still overloaded then VM selection policy is applied again till we can call it 'not overloaded'.

We need a policy which will help us select a VM requiring minimum migration time. This choice of VM selection will help us in reducing the downtime of application on the migrating VM. Thus it will help us reducing SLA violation. Performance evaluation proves that MMT outperforms MC and RS in terms of migration time [10]. Thus we use MMT as our VM selection policy.

IV. TARGET HOST FOR VM UNDER MIGRATION

Once a VM is selected for migration we need to find a host for migrating the VM. We should take care that the target host should not get overloaded after we place our selected VM. Traditional algorithms focus mostly only on CPU utilization for deciding whether host is overloaded or not while allocating VMs [7]. Algorithms proposed by Wood et al. [8], Zheng et al. [9] and DAIRS (dynamic and integrated resource scheduling algorithms) [7] consider CPU, network

and memory capability of server for calculation of its integrated utilization.

Wood et al. [8] proposed a method for calculation of integrated utilization of host combining its CPU, memory and network bandwidth given by equation 4.1.

$$V = \frac{1}{(1-CPUu)(1-MEMu)(1-NETu)} \quad (4.1)$$

Where V is the integrated load and CPU_u, MEM_u and NET_u are the CPU, memory and network utilization of the host. The larger the value of V is, the higher is the integrated utilization. This actually is a strategy of minimizing integrated resource utilization. This algorithm suggests that server with lowest value of V be chosen as target for VM v.

Zheng et al. [9] proposed another integrated load-balancing measurement B:

$$B = \frac{a.N1.Ci}{N1m.Cm} + \frac{b.N2i.Mi}{N2m.Mm} + \frac{c.N3i.Di}{N3m.Dm} + \frac{d.N1i.NETi}{NETm} \quad (4.2)$$

The referred physical server m is selected first. Then each of the other physical servers i is compared to server m. $N1_i$ is the CPU capability, $N2_i$ is for memory capability, $N3_i$ is for hard disk. C_i , M_i is for average utilization of CPU and memory respectively, D_i is for data transferring rate of hard disk, NETi is for network bandwidth. Constants a,b,c and d are weighting factors of CPU, memory, hard disk and network bandwidth respectively. The major idea of this algorithm is to choose the smallest value B among all physical servers to allocate VMs.

Now we discuss DAIRS [7] algorithm for finding target host for VM 'v'. DAIRS outperforms algorithms proposed by Wood et al. and Zheng at al. in terms of Load Balancing [7]. Thus, we will be using it as our VM allocation policy (i.e. finding target host for VM 'v').

1. Average CPU, memory and network utilization of all servers in datacenter is calculated (cpuA, memA, netA).
2. For the host under consideration/candidate find its cpu, memory and network utilization (cpuUtil, netUtil and memUtil).
3. Calculate integrated average of cpu, memory and network utilizations :

$$iA = (cputUtil + netUtil + memUtil)/3 \quad (4.3)$$

4. Integrated load imbalance of host is given by,

$$ILB = \frac{(iA-cpuA)+(iA-memA)+(iA-netA)}{3} \quad (4.4)$$

5. Now we select host with minimum ILB value given by equation 4.4

   Note: Host will not be considered as candidate if
   a. After assignment it is getting overloaded.
   b. Host is not suitable for VM in terms of resource specifications.

V. CALCULATING IMBALANCE LEVEL

Average imbalance is defined as the arithmetic mean of ILB of all servers [7].

$$iblAvg = \frac{\sum_{i=1}^{N} ILBi}{N} \quad (5.1)$$

Where 'N' is the number of servers, ILB_i is the integrated load imbalance (4.4) of server 'i'.

Lower the imbalance value balanced is the load on that server.

Our goal will be to reduce the average imbalance iblAvg (equation 5.1).

VI. UNDERLOADED HOST DETECTION

The reason for finding an underloaded host is to move its VMs to other hosts so that this underloaded host can be put to sleep thus saving energy/power.

Following is an approach for finding an underloaded host and its VMs migration [10].

1. Find all overloaded hosts (overHosts) using overloaded host detection algorithm described in Section 2. We regard all hosts which are not 'overHosts' as 'candidateHosts'.
2. From 'candidateHosts' find a host i which has lowest CPU utilization among all candidateHosts.
3. Let targetList = cadidateHosts – host i.
4. Try to migrate a VM from host i to a host from 'targetList'. Similarly migrate all VMs from host i to hosts from 'targetList'. Migration is possible if the target host has sufficient resource requirements (CPU, memory etc.) for the VM under migration.
5. If all the VMs can be migrated from host i to hosts from 'targetList' then the host i is put to sleep else host i is kept active.

VII. SIMULATIONS

It is very difficult to conduct repeatable experiments on real infrastructure that is why simulation on Cloudsim [12] environment was chosen to show the effectiveness of the proposed scheme. Simulation time is 30 minutes.

A. *Experimental Setup*

We selected two server configurations for testing:

a) HP ProLiant ML110 **G4** (Intel Xeon 3040, 2 cores * 1860 MHz, 4GB) hereafter referred as G4.
b) HP Proliant ML110 **G5** (Intel Xeon 3075, 2 cores * 2660 MHz, 4 GB) hereafter referred as G5.

Host B/w= 1GBps and storage=1GB.

Four types of VMs were used.

a) Type1: cpu=25MHz mem=87MB
b) Type2: cpu=20MHz mem=174MB
c) Type3: cpu=10MHz mem=174MB
d) Type4: cpu=5MHz mem=61MB

VM b/w=10Mbps and size=0.25GB

B. *Power Model*

We utilize real data on power consumption provided by the results of the SPECpower benchmark [13]. The selected servers G4 and G5 are with low computing capacity so that a lot of VM migrations should occur, and we will be able to capture results effectively.

Power consumption by servers G4 and G5 at different load levels in Watts against utilization (first column) in percentage is given in Table I.

Table I

Utilization in %	Power consumption by servers in watts	
	G4	G5
0%	86	93.7
10%	89.4	97
20%	92.6	101
30%	96	105
40%	99.5	110
50%	102	116
60%	106	121
70%	108	125
80%	112	129
90%	114	133
100%	117	135

Power consumption in watts at different load levels

C. *Power Model*

Simulation results were obtained for following three algorithms.

a. Optimized Load Balancing algorithm (OLB): This is our optimized algorithm which optimizes between energy consumption while balancing the load.
b. Dynamic and Integrated Load-Balancing scheduling algorithm (DAIRS) [7] which is designed for balancing load among servers.
c. Local Regression with minimum migration time (LR_MMT) [10] which focuses only on reduction in energy consumption and SLA violations.

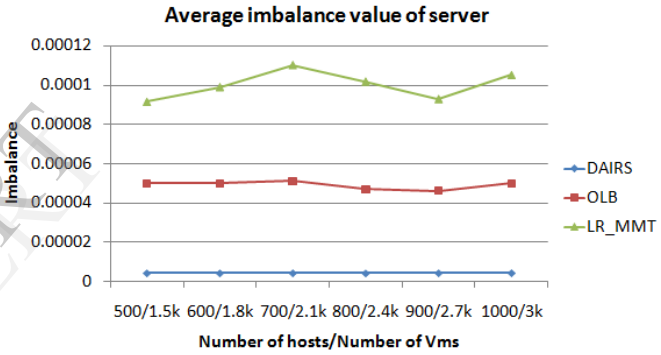

Fig. 1
Average imbalance level of servers under various algorithms with various numbers of hosts and VMs.
(see section 5 from calculations)

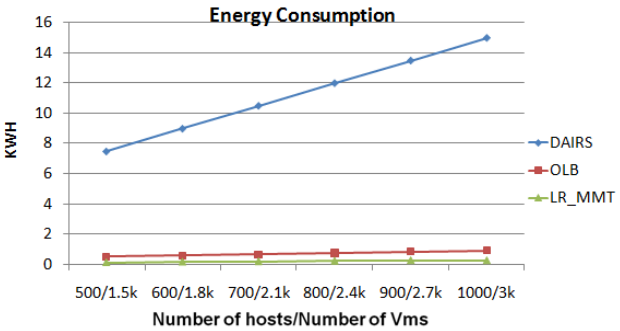

Fig. 2
Overall Energy Consumption of all servers under various algorithms with various numbers of hosts and VMs.

Figure 1 provides average imbalance level of all servers calculated using Equation 5.1. It can be clearly seen from the figure that our optimal load balancing algorithm (OLB) has optimal load balancing capability when compared to DAIRS (which is highly load efficient) and LR_MMT (which

is not at all designed to balance load). The lower the value of load imbalance on Y-axis, the more balanced the load is. Figure 2 gives the energy consumption in KWh. It can be seen from the figure that DAIRS algorithm is highly energy inefficient; on the contrary LR_MMT is highly energy efficient. Our OLB has achieved energy efficiency very close to LR_MMT, is a significant improvement in energy efficiency by a load balancing algorithm.

VIII. Conclusion and Future Work

In this paper we have proposed an optimized load balancing (OLB) algorithm. This proposed algorithm optimizes load balancing and energy efficiency. Results demonstrate the correctness of our OLB algorithm. The energy efficiency is highly optimized. However we expect to improve the load imbalance level further in future research. We also expect to optimize the number of VM migration.

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, vol. 25(6), pp. 599-616.

[2] ASHRAE Technical committee 99: "Datacom equipment power trends and cooling applications 2005".

[3] Barroso LA, Holzle U. "The case for energy-proportional computing". 2007, vol. 40(12), pp. 33-37.

[4] Fan X, Weber WD, Barroso La. "Power provisioning for a warehouse-sized computer". Proceedings of the 34ths Annual international symposium on computer architecture (ISCA 2007), ACM New York, NY, USA, 2007, pp. 13-23.

[5] Ranganathan P, Leech P, Irwin D, Chase J. "Ensemble-level power management for dense blade servers". Proceedings of the 33rd International symposium on computer architecture (2006), Boston, MA, USA, 2006, pp. 66-77.

[6] The green grid consortium 2011. URL http://www.thegreengrid.org. accessed 7th March, 2013.

[7] Wenhong tian, Yong Zhao, Yuanliang Zhong, Minxian Xu, chen Jing. "A dynamic and integrated Load-Balancing scheduling algorithm for cloud datacenters". Proceedings of IEEE CCIS2011, pp. 311-315.

[8] T. Wood, et. al., "Black-Box and Gray-box strategies for virtual machine migration" in the proceedings of Symp. On networked systems design and implementation (NSDI), 2007.

[9] H. Zheng, L. Zhou, J. WU, "Design and implementation of Load Balancing in Web Server cluster System", Journal of Nanjing University of Aeronotics & Astronautics Vol. 38 No. 3 Jun. 2006

[10] 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 Interscience (www.interscience.wiley.com). Concurrency and computation: Practice and experience.2012, vol. 24, pp. 1397-1420.

[11] Cleveland WS. "Robust locally wighted regression and smoothing scatterplots". Journal of the American statistical association 1979, vol. 74(368), pp. 829-836.

[12] 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, vol. 41(1), pp. 23-50.

[13] http://www.spec.org/power_ssj2008. accessed 23rd Feb 2013.

[14] http://www.mathsisfun.com/data/quartiles.html accessed 5th March 2013.

[15] Kendall MG, Ord JK. "Time-series". Oxford University Press, Oxford, 1973.

[16] Cleveland WS, Loader C. "Smoothing by local regression: Principles and methods. Statistical theory and computational aspects of smoothing 1996", vol. 1049, pp. 10-49.

[17] 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, pp. 28-28.