A Survey on Load Balancing Techniques in Cloud Data Center using VM Migration

[1]Hitesh B Patel, [2]Prashant B Swadas, [3]Sudarshan N Patel
[1]PG Student, Computer Engineering Department,
BVM Engineering College, VV Nagar-388120, India.
[2]Head, Computer Engineering Department,
BVM Engineering College, VV Nagar-388120, India.
[3]Assistant Professor, Computer Engineering Department,
A D Patel Institute of Technology, New VV Nagar-388120, India.

hit19185@gmail.com, prashantswadas@yahoo.com, ssnpatel@gmail.com

Abstract: **-- In this paper we study and represent various techniques of load balancing in Data Center using Virtual Machine Migration. Virtual Machine Migration requires memory, storage, and network connectivity are transferred from overloaded host machine to under loaded destination machine. The main aim of VM migration is to balance the load of physical machines in Cloud Environment. This paper focuses on Virtual Machine allocation in Cloud Data Center. We have represented various existing load balancing techniques, recent work and recent methods of load balancing. Finally we have presented challenge issues and future research directions for load balancing in Cloud Environment.**

Keywords: **-- Cloud Computing, Virtualization, Virtual Machine Migration, Load Balancing.**

I. INTRODUCTION

Cloud Computing is Internet based computing where Virtual Shared Servers provide Software, Infrastructure, Platform, Devices and other resources and hosting to customers on a pay-as-you-use basis. Server resources [1] in a cloud data center are multiplexed across multiple applications-each server runs one or more applications, and application components may be distributed across multiple servers. Further, each application sees dynamic workload fluctuations caused by incremental growth, time-of-day effects, and flash crowds. Since applications need to operate above a certain performance level specified in terms of a service level agreement (SLA), effective management of data center resources while meeting SLAs is a complex task.

Through Virtualization an end user can get different services provided by the cloud. Server virtualization [2] makes it possible to execute concurrently several virtual machines (VM) on top of a single physical machine (PM), each VM hosting a complete software stack (operating system, middleware, applications) and being given a partition of the underlying resource capacity (mainly CPU power and RAM size). On top of that, the live migration capability of hypervisors allows a virtual machine to migrate from one physical host to another. Live migration makes it possible to dynamically adjust data center utilization and tune the resources allocated to the applications.

Consider a data center consisting [3] of "n" number of Physical Machines hosting "m" number of Virtual Machines implementing one customer application each. Resources (CPU, network, memory, I/O) are allocated to each Virtual Machine to handle the workload and operate at certain performance level or Service Level Agreement. Each Virtual Machine sees workload fluctuation from time to time as resource requirement changes e.g. the no. of user visit increases at particular VM.

An increase in workload of VM can be handled by allocating more resources to it, if idle resources are available. But what if Physical Machine does not have enough or no resource to fulfil VM's requirement, which leads to performance degradation of the application and SLA violation occurs. The one of the solution is Replicating VMs, but it causes memory overhead, applicable only for web hosting and also need for a load balancer between replicas. The other approach is Migrating VMs, but it is not applicable when VM resource requirements are higher than capacity of PM.

Consider a Homogeneous Cluster [3] in which as VMs resource requirements changes dynamically, the initial placement of "m" number of VMs on to "n" numbers of PMs may lead to SLA violation and performance degradation. In order to avoid the SLA violation, we have to decide when to trigger the migration, which VM to migrate, and where to migrate to reallocate VMs to PMs dynamically.

II. LOAD BALANCING TECHNIQUES

At present there are various techniques for load balancing in cloud environment. Some of them are discussed here.

A. Allocation of VM

In this technique each node in the data center runs a module of the VM monitor which observes the local resource usages of the node.

If the local observations reveal an anomaly that the resources are over-utilized or under-utilized, there are two decisions namely [1]:

1. Which VM to migrate from the problematic PM.
2. Which PM to migrate the chosen VM to.

VM Selection:

If the resources are over-utilized, there are one or more VMs need to be migrated. In order to reduce the number of migrations, the system sorts all VMs on the problematic PM in decreasing order of current utilization first, and then the system chooses the VM [1] which has the highest utilization in the decreasing order. If the resources are still over-utilized after the migration of the highest utilization of the VM, then the system chooses the next highest utilization of the VM in the decreasing order until the anomaly is resolved.

PM Selection:

When the choice of the VM is finished, the system begins to select the PM from the data center to migrate the chosen VM to. If there is no PM in the data center that can host the VM, then no migration is happened. Otherwise, all of the PMs in the data center that can host the VM without exceeding the resource threshold compose a set, and the system will choose the most suitable PM from the set using the TOPSIS [7] method.

TOPSIS:

TOPSIS (Technique for Order Preference by Similarity to Ideal Solution). The basic concept of this method is that the selected alternative should have the shortest distance from the ideal solution and the farthest distance from the negative-ideal solution in a geometrical sense.

TOPSIS assumes that each attribute has a tendency of monotonically increasing or decreasing utility. Therefore, it is easy to locate the ideal and negative-ideal solutions. The Euclidean distance approach is used to evaluate the relative closeness of alternatives to the ideal solution. Thus, the preference order of alternatives is yielded through comparing these relative distances.

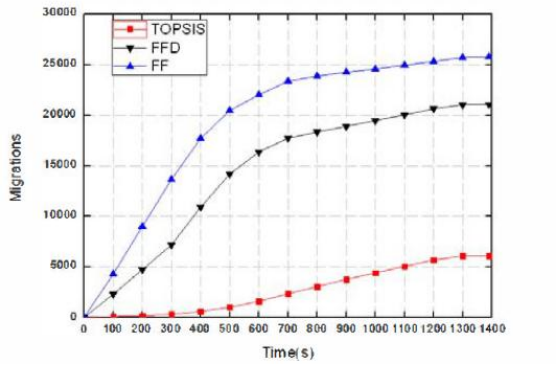

Fig. 2.1 VM migrations by three methods [1]

Experimental results [1] show that when the simulation is finished, the number of VM migrations by the TOPSIS method is only 6104, about one-third of number of VM migrations by FFD and one-fourth by FF, which are globally optimal solution makes a lot of unnecessary migrations, but the system by the TOPSIS method migrate VM just when an anomaly occurs. Less migration can make the system more stable.

Using this method we can achieve better load balancing in a large-scale cloud computing environment with less VM migration.

B. Sandpiper

This technique studies automated black-box and grey-box strategies for virtual machine migration in large data center. These techniques [4] automate the tasks of monitoring system resource usage, hotspot detection, determining a new mapping and initiating the necessary migrations. Black-box techniques can make these decisions by simply observing each virtual machine from the outside and without any knowledge of the application resident within each VM. Grey-box approach assumes access to a small amount of OS-level statistics in addition to external observations to better inform the migration algorithm.

Sandpiper [4] implements a hotspot detection algorithm that determines when to migrate virtual machines, and a hotspot mitigation algorithm that determines what and where to migrate and how much to allocate after the migration. The hotspot detection component employs a monitoring and profiling engine that gathers usage statistics on various virtual and physical servers and constructs profiles of resource usage. These profiles are used in conjunction with prediction techniques to detect hotspots in the system.

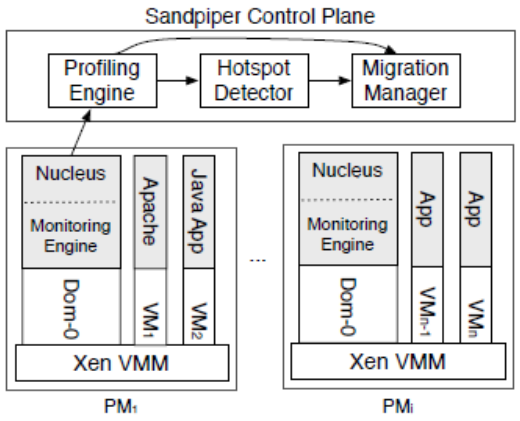

Fig. 2.2Sandpiper Architecture [4]

Upon detection, Sandpiper's migration manager is invoked for hotspot mitigation. The migration manager employs provisioning techniques to determine the resource needs of overloaded VMs and uses a greedy algorithm to determine a

sequence of moves or swaps to migrate overloaded VMs to under loaded servers.

Capturing Multi-dimensional Loads:

The migration manager [4] employs a greedy heuristic to determine which VMs need to be migrated. The basic idea is to move load from the most overloaded servers to the least-overloaded servers, while attempting to minimize data copying incurred during migration. Since a VM or a server can be overloaded along one or more of three dimensions– CPU, network and memory. The volume [4] of a physical or virtual server is defined as the product of its CPU, network and memory loads:

$$volume = \frac{1}{1-cpu} * \frac{1}{1-net} * \frac{1}{1-mem}$$

Here cpu, net and mem are the corresponding utilizations of that resource for the virtual or physical server.

The higher the utilization of a resource, the greater the volume; if multiple resources are heavily utilized, the above product results in a correspondingly higher volume.

Migration Phase:

To determine which VMs to migrate, the algorithm orders physical servers in decreasing order of their volumes. Within each server, VMs are considered in decreasing order of their volume-to-size ratio (VSR); where VSR is defined as Volume/Size; size is the memory footprint of the VM. By considering VMs in VSR order, the algorithm attempts to migrate the maximum volume (i.e., load) per unit byte moved, which has been shown to minimize migration overhead [8].

Swap Phase:

In cases where there aren't sufficient idle resources on less loaded servers to dissipate a hotspot, the migration algorithm considers VM swaps as an alternative. A swap involves exchanging a high VSR virtual machine from a loaded server with one or more low VSR VMs from an under loaded server. Such a swap reduces the overall utilization of the overloaded server.
This technique automates the task of monitoring and detecting hotspots, determining a new mapping of physical to virtual resources and initiating the necessary migrations in a virtualized data center.

C. Application Performance Management

This technique introduces the concept of server consolidation [5] using virtualization and associated issues that arise in the area of application performance. These problems can be solved by monitoring key performance metrics and using the data to trigger migration of Virtual Machines within physical servers. The algorithms using this technique attempt to minimize the cost of migration and maintain acceptable application performance levels.
Metrics representing CPU and memory utilization, disk usage, etc., are collected from both the VMs and the PMs hosting them using standard resource monitoring modules. Thus, from a resource usage viewpoint, each VM can be represented as a d-dimensional vector where each dimension represents one of the monitored resources. In this technique resource utilization of a virtual machine VM_i as a random process represented by a d-dimensional utilization vector ($U_i(t)$) at time t. For a physical machine PM_k the combined system utilization is represented by $L_k(t)$. So $L_k(t) = f(U_1(t), U_2(t)..)$ for all VM located on machine PM_k.

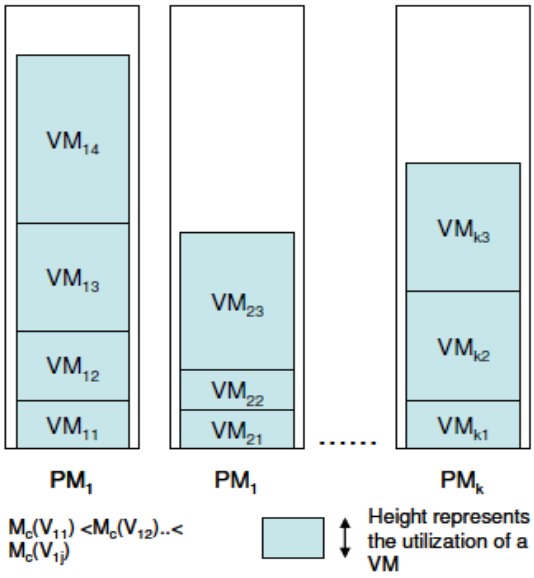

Fig. 2.3 The VMs on each PM are ordered with respect to their Migration Costs. [5]

Since migration cost is directly calculated based on the utilization, another way to look at the order is "Virtual Machines are ordered according to their migration costs within each Physical Machine". Here for each physical machine, calculation and storing its residual capacity $r_i(t)$ has been done. The list of residual capacities in non-decreasing order is shown in Figure 2.3. Residual capacity for a resource, such as CPU or memory, in a given machine denotes the unused portion of that resource that could be allocated to an incoming VM.

Important Features of the algorithm: [5]

- It can be used to perform online dynamic management.
- Also it tries to minimize the number of migrations.
- It minimizes the migration cost by choosing the VM with minimum utilization.
- Also it provides mechanism to add and remove physical machines thus providing dynamic resource management

Using this technique we can solve the problem of degrading application performance with changing workload in such virtualized environments. Specifically, changes in workload may increase CPU utilization or memory usage above acceptable thresholds, leading to SLA violations. It also used for how to detect such problems and, when they occur, how to resolve them by migrating virtual machines from one physical machine to another.

D. Load Imbalance

A common approach for quantifying physical server load [6] is to measure the utilization of its resources (e.g., CPU time, network, memory utilization and disk I/O traffic). Virtualization introduces another layer of abstraction on top of a physical server – virtual machines only know about virtualized hardware resources, while the hypervisor manages both virtual and physical hardware resources. As a result, it becomes more challenging to balance performance in terms of per-VM utilization of system resources given this added layer of abstraction.

Working at the hypervisor level [6] allows us to isolate each VM's virtual and physical resource consumption and enables us to isolate the resource consumption of a particular VM when making predictions on the overall system balance. Another desirable property of a virtualized server load metric is that it can be used on heterogeneous systems. Real world cluster systems are rarely homogeneous, so in order for this metric to apply to a variety of system configurations, we need to quantify the load of a server in a manner that does not depend on fixed resource units. A unit less metric allows the direct comparison of different servers in the system regardless of their internal components.

Suppose S be the set of physical servers and VM_{Host} be the set of virtual machines currently running on physical server Host, (Host ∈ S). Then, the Virtualized Server Load (VSL_{Host}) can be expressed as:

$$VSL_{Host} = \sum_{resource} Wresource * \frac{\sum_{v \in VM_{Host}} Vresource\ Usage}{Host\ resource\ Capacity}$$

Here resource ∈ {CPU, memory, disk} and $W_{resource}$ is a weight associated with each resource.

Load Imbalance Metric:

A typical imbalance metric based on the resource utilization of physical servers is the standard deviation of the CPU utilization [8]. The reasoning behind this metric is that if the server loads are evenly distributed, the standard deviation will be small. The smaller this metric, the greater the load balances in the system [6].

A virtualized server load metric defined earlier that is a function of VM resource usage. This metric considers information specific to each VM when quantifying the load of a particular physical server. This metric takes into consideration the usage of multiple resources by the VMs resident on the system. Based on this definition, we can generate a load set L that contains the VSL values corresponding to all physical servers.
The desired system imbalance metric can then be defined [6] in terms of the coefficient of variation of L:

$$C_L = \frac{\sigma L}{\mu L}$$

As shown, C_L is defined as the ratio of the standard deviation σ_L over the mean μ_L.
However, there are some problems that must be taken into consideration when using the CL as an imbalance metric. The most evident problem is in cases where μ_L is zero. The only time this could happen is when all servers are idle or when the virtual machine monitor (VMM) is not consuming any resources. Although these cases are extremely rare, in order to avoid this problem, so the imbalance metric defined [6] as, $I_{Metric} = 0$, if no VMs are active and C_L, otherwise.

Finally, a new virtualized server load metric that is based on the current resident VM resource usage named VSL. Also here an imbalance metric that is based on the variation in load present on the physical servers, which provide predictions of future system behaviour with high fidelity (i.e., with an error margin of less than 5%). This new imbalance metric was used to drive load balancing method (VIBM) on a virtualized enterprise server VIBM implements a greedy approach, selecting the VM migration that yields the most improvement of the imbalance metric at each time step.

III. COMPARISON OF VARIOUS LOAD BALANCING TECHNIQUES

Comparisons of load balancing techniques have been mainly concerned with what assumptions are made in the less VM migration with low cost. VM allocation using TOPSIS method shows less VM migration, while sandpiper automates the task of monitoring and detecting hotspots, and initiating the necessary migrations in a virtualized data center.

We can achieve less migration overhead by capturing multidimensional loads. By considering VMs in VSR order, we can migrate the maximum volume (i.e., load) per unit byte moved, which leads to minimum migration overhead. Migration cost is directly calculated based on the utilization; another way to look at the order of migration is "Virtual Machines are ordered according to their migration costs within each Physical Machine".

Following table shows the comparison of various load balancing techniques described earlier in terms of when to trigger for migration of the VM, which VM to migrate and where the VM should be migrated.

TABLE I
COMPARISON OF LOAD BALANCING TECHNIQUES

Techniques	When to Migrate?	Which VM to Migrate?	Where to Migrate?
Allocation of VM [1]	For a PM any of the resource usage > threshold. Threshold for each resources	Select overloaded VM from overloaded PM	to the under loaded PM
Sandpiper [4]	For a PM any of the resource usage > threshold. Threshold for each resources	From overloaded PM, choose VM having minimum Volume to Size Ratio.	to the under loaded PM
Application Performance [5]	Same as above	From overloaded PM, choose VM having minimum L (i.e. in terms of cost and utilization)	to the PM which has least enough residual capacity
Load-Imbalance [6]	For a cluster, coefficient of variance of PM's load > threshold	Select overloaded VM from overloaded PM	to the under loaded PM

The metric used in load imbalance technique is used to measure load imbalance and to construct a load-balancing VM migration framework.

IV. CHALLENGE ISSUES FOR LOAD BALANCING

Load balancing in cloud data center is really a challenge now. Always a distributed solution is required. Because it is not always practically feasible or cost efficient, to maintain one or more idle services just to fulfil the required demands. Jobs can't be assigned to appropriate servers and clients individually for efficient load balancing as cloud is a very complex structure and components are present throughout a wide spread area. Here some uncertainty is attached while jobs are assigned.

Migration cost and Migration overhead can be major issues while balancing load in large data center. They can be resolved using techniques described earlier. Also there are other issues while balancing the load in data center like power aware load balancing, Application performance based load balancing etc.

V. CONCLUSION AND FUTURE WORK

In cloud data center, the system should avoid wasting resources as a result of under-utilization and avoid lengthy response times as a result of over-utilization.

Energy efficiency is one of the most active topics in large scale of data center today [1]. So we can consider the power consumption of the data center when the system finds the most suitable PM for the migrated VMs.

Also we can consider a system which decides whether to migrate a VM or to spawn a replica in order to acquire more resources.

REFERENCES

[1] Fei. Ma, Feng Liu and Zhen Liu, "Distributed Load Balancing Allocation of Virtual Machine in Cloud Data Center "978-1-4673-2008-5/12 2012 IEEE.

[2] Susanta, Nanda and Tzi-cker, Chiueh. "A Survey on Virtualization Technologies," Experimental Computer Systems Lab, Feb 2005.

[3] Load Balancing in a Data Center using VM Migration by Senthil Nathan

[4] Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif "Black-box and Gray-box Strategies for Virtual Machine Migration " Univ. of Massachusetts Amherst Intel, Portland.

[5] Gunjan Khanna, Kirk Beaty, Gautam Kar, Andrzej Kochut" Application Performance Management in Virtualized Server Environments "1-4244-0143/06 2006 IEEE.

[6] Emmanuel Arzuaga and David R. Kaeli. "Quantifying load imbalance on virtualized enterprise servers", In WOSP/SIPEW '10: Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering, pages 235–242, New York, NY, USA, 2010. ACM.

[7] C.L. Hwang and K. Yoon, "Multiple attributes decision making methods and applications: a state of the art Survey," Springer-Verlag, New York, 1981.

[8] V. Sundaram, T. Wood, and P. Shenoy. "Efficient Data Migration in Self-managing Storage Systems", In Proc. ICAC '06.

[9] Jinhua Hu, Jianhua Gu, Guofei Sun and Tianhai Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", presented at 3rd International Symposium on Parallel Architectures, Algorithms and Programming.

[10] Qingyi Gao, Peng Tang, Ting Deng, Tianyu Wo, " VirtualRank : A Prediction Based Load Balancing

Technique In Virtual Computing Environment ", 2011 IEEE World Congress on Services.

[11] Junjie Ni, Yuanqiang Huang, Zhongzhi Luan, Juncheng Zhang, Depei Qian "Virtual Machine Mapping Policy Based on Load Balancing in Private Cloud Environment",2011 International Conference on Cloud and Service Computing.

[12] Xinyu Zhang, Yongli Zhao, Xin Su, Ruiying He, Weiwei Wang, Jie Zhang, " Load Balancing Algorithm based Virtual Machine Dynamic Migration Scheme for Datacenter Application with Optical Networks", 2012 7th International ICST Conference on Communications and Networking in China (CHINACOM).

[13] Jiann-Liang Chen, Yanuarius Teofilus Larosa and Pei-Jia Yang, " Optimal QoS Load Balancing Mechanism for Virtual Machines Scheduling in Eucalyptus Cloud Computing Platform ", 2012 2nd Baltic Congress on future Internet Communication.

[14] Manish Arora, Sajal K. Das and Rupak Biswas, "A De-centralized Scheduling and Load Balancing Algorithm for Heterogeneous Grid Environments", Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'02) 1530-2016/02 2002 IEEE.

[15] Chonggun Kim and Hisao Kameda, "An Algorithm for Optimal Static Load Balancing in Distributed Computer Systems ", IEEE TRANSACTIONSON COMPUTERS, VOL.41, NO.3, MARCH 1992.

[16] Che-Lun Hung, Hsiao-hsi Wang and Yu-Chen Hu, "Efficient Load Balancing Algorithm for Cloud Computing Network".

[17] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAW-HILL Edition 2010.

[18] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.