Load Balancing Strategy for Migration in Virtualized Data Center

Khin Swe Swe Myint
University of Computer Studies, Yangon
khinsweswemyint@gmail.com

Abstract

Virtualization techniques provide the carving individual physical servers into multiple virtual containers that can be run and managed separately. However, due to the uncertainty of workload variation, the workloads may be unbalanced in the virtualized system, leads to application's SLA violation. In order to deal with such problem, the mathematical expectation approach use to determine the hotspots in virtualized server. To relocate virtual machine (VMs) from overloaded servers to underloaded ones, the system uses the Balance migration to reduce the migration cost and it eliminates the hotspots.

1. Introduction

Virtualization is an effective approach to enhance resource utilization on demand in cluster server. How to manage the resource effectively to satisfy the Service Level Agreement (SLA) of applications in the virtualized cluster is a complex task. There are various virtualization system can be used cluster server, such as XEN, KVM, VMware, and Virtual PC etc [1]. The virtualization system can support consolidation of multiple application services on a same server and enables performance isolation among different applications.

With increasing scale and complexity of modern enterprise data centers, administrators are being forced to rethink the design of their data centers. In a traditional data center, application computation and application data are tied to specific servers and storage subsystems that are often over-provisioned to deal with workload surges and unexpected failures. Such configuration rigidity makes data centers expensive to maintain with wasted energy and floor space, low resource utilizations and significant management overheads.

Today, there is significant interest in developing more agile data centers, in which applications are loosely coupled to the underlying infrastructure and can easily share resources among themselves. Also desired is the ability to migrate an application from one set of resources to another in a non- disruptive manner. Such agility becomes key in modern cloud computing infrastructures that aim to efficiently share and manage extremely large data centers. One technology that is set to play an important role in this transformation is virtualization. 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. The maximum capacity is seldom reached and results in unused space and wasted resources.

An important characteristic for a well managed data center is its ability to avoid hotspots. Overloaded node often leads to performance degradation and is vulnerable to failures. To alleviate such hotspots, load must be migrated from the overloaded resource to an underutilized one. Migration is further complicated by the need to consider multiple resources—CPU, network, and memory—for each application and physical server.

The rest of this paper is organized as follows. It presents some extended motivation and address related work in Section 2. In Section 3, background and system overview is described in detail. It will be followed by conclusion in section 4.

2. Related Work

Currently, there are many researches about dynamic resource management in virtualization systems [7]. Nathuji [5] presents power efficient mechanisms to control various power management policies. The VMware Distributed Resource Scheduler (DRS) [9] performs such migrations based on only CPU and memory resources for all hosts and virtual machines. It cannot utilize application logs to respond directly to potential SLA violations or to improve placement decisions.

In C. Clark [2] dynamic network-bandwidth adaptation allows migration to proceed with minimal impact on running services, while reducing total downtime to below discernable thresholds. It introduces and analyzes the concept of writable working set, and presents the design, implementation and evaluation of high performance OS migration built on top of the Xen VMM. Menasce et al [4] consider CPU as single metric; it controls the server by dynamic CPU priority allocation, and assigns CPU shares to the various virtual machines by beam-search algorithm.

The Wood [7] proposes a Black-Gray box migration called as sandpiper. It migrates overload VMs to under-utilized nodes when hot spot was detected. It performs a comprehensive analysis, predict, and allocation with CPU, memory, and network. But it doesn't consider the final usage of the physical nodes and the migration effect, the effect of the migration could be improved.

A. Gambi, M. Pezze, M. Youong [3] show that SLA protection in a virtualized data center depends on structure and behavior at many abstraction levels, and argue that information required for defining autonomic control strategies can be captured by a set of interrelated models. It has identified SLA, workload, service composition, component architecture, virtual execution environment (VEE), virtual area network (VAN) and physical resource allocation models as key elements that impact autonomic control policies.

Then, a migration algorithm, called as Balance migration, is presented, to reduce the migration cost and enhance the stability of virtualized server after migration is done. It performs better in the migration times and the stability of the nodes after the migration than Black-Gray box (BGB) migration.

3. Background and System Overview

In the virtualized server, the load of the Web server is changed dynamically over time. According to the dynamical characteristics of the workload, the virtualization plays its important role through adjusting the resource allocation by dynamical migration. The system should be auto-control and self-learning to dynamically allocate resource to virtual machine.

A workload increase can be handled by increasing the resources allocated to a virtual server, if idle resources are available on the physical server, or by migrating the virtual server to a less loaded physical server. Two phases are present, the first phase is to mark out of the hotspot node, collect and handling with the system information and the quantity of the resource allocated to the migration nodes. The second phase is to determine which virtual machine to migrate, where to migrate and the migration policy to ensure the hotspot elimination after migration.

In Xen, it implements such architecture. Each virtual server is assumed to be allocated a certain slice of the physical server resources. In the case of CPU, this is achieved by assigning a weight to the virtual server and the underlying Xen CPU scheduler allocates CPU bandwidth in proportion to the weight. In case of the network interface, Xen is yet to implement a similar fair-share scheduler; a best-effort FIFO scheduler is currently used and Sandpiper is designed to work with this constraint. In case of memory, a slice is assigned by allocating a certain amount of RAM to each resident VM. All storage is assumed to be on a network file system or a storage area network, thereby eliminating the need to move disk state during VM migrations [2].

The hardware configuration of each server—its CPU, network interface and memory characteristics—is assumed to be known to the system. Each physical server (also referred to as a physical machine or PM) runs a virtual machine monitor and one or more virtual machines. Each virtual server runs an application or an application component (the terms virtual servers and virtual machine are used interchangeably). Figure 1 shows the overview of the virtual machine migration system. It is composed of three components, an estimate resource, load detection and migration manager. In the estimate resources, it calculates the resources (cpu, net and mem) using the G/G/1 queuing approach and load detection determines the threshold for physical resources. Then migration manager move the virtual machine to the lower resource utilization physical machine to eliminate the overload.

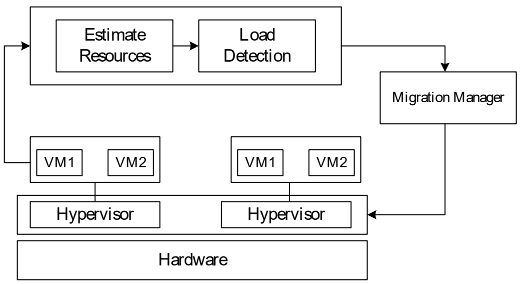

Figure 1. Overview of virtual machine migration system

3.1. Estimate Resources

To estimate peak needs, the peak request arrival rate is first estimated. Since the numbers of serviced request as well as the numbers of dropped request are typically logged, the incoming request rate is the summation of these two quantities. Let λ_{peak} denote the estimated peak arrival rate for the application. An application model is necessary to estimate the peak CPU needs. By using the G/G/1 queuing theory, the system can be captured the results where d is the mean response time of requests, s is the mean service time. λ_{cap} and λ_{mem} is the request arrival rate. σ^2_a and σ^2_b are the variance of inter-arrival time and the variance of service time, respectively.

$$\lambda \geq \left[s + \frac{\sigma_a^2 + \sigma_b^2}{2 \cdot (d - s)} \right]^{-1} \quad (1)$$

The desired response time d is specified by the SLA, the service time s of requests as well as the variance of inter-arrival and service times σ^2_a and σ^2_b can be determined from the server logs. λ is the $\lambda_{cap,}$ and λ_{mem} . They represent the current capacity of the VM. To service the estimated peak workload λ_{peak}, the current CPU capacity needs to be scaled by the factor $\frac{\lambda peak}{\lambda_{cap}}$ and the current memory capacity needs to be scaled by the factor $\frac{\lambda peak}{\lambda_{mem}}$. If the VM is currently assigned a CPU weight w1 and memory weight w2, its allocated share needs to be scaled up by the factor $\frac{\lambda peak}{\lambda_{cap}}$, $\frac{\lambda peak}{\lambda_{mem}}$ to service the peak workload.

The peak network bandwidth usage is simply estimated as the product of the estimated peak arrival rate λ_{peak} and the mean requested file size b. The mean request size can be computed from the server logs.

Table 2 summarizes the symbols used to describe the definition.

Table 2. Definition of notations

Symbols	Definition
λ_{peak}	Estimated peak arrival rate for the application
d	Mean response time of requests
s	Mean service time
λ_{cap}	The current CPU capacity
λ_{mem}	The current memory capacity
σ^2_a	The variance of inter-arrival time
σ^2_b	The variance of service time
b	Mean requested file size
f	Maximum requested file size

3.2. Load Detection

The load detection examines the mathematical expectation of the value of the physical node, for the observation sequence: R_1,

$R_{2...}R_k$. R represents any resources, and then the mathematical expectation can be expressed :

$$\mu = E(X) = \frac{\sum_{i=1}^{k} R_i}{k} \qquad (2)$$

When the mathematical expectation of observed value exceeds the threshold (75%), the system will be overloaded. In the first part, the system has established the threshold of the three kinds of resources and judge whether these physical nodes are hotspots through analyzing the usage of resource.

In the second part, it eliminates hotspots and is also taking into account the two aspects at the same time: first, how to detect the overload occupied and how to reduce the migration cost when eliminating hotspot. And then, eliminate the hotspot with least physical node.

3.3. Migration Manager

In this paper, it applies the two types of migration, BGB and balance migration. Once the hotspots have been marked, load detection will use the Migration Algorithm to decide the allocation scheme. The main objective of these migrations is to determine which virtual servers must be migrated and where to migrate to eliminate the hotspot. Assume the values of the VMs resources (cpu, net, and mem) for each PM already known by the system.

3.3.1Black-Gray box (BGB) Migration

In Black-Gray box system, it also defines two new metric that captures the combined CPU-network-memory load of a virtual and physical server. The volume of a physical or virtual server is defined as the product of its CPU, network and memory loads [7].

$$Vol = \frac{1}{1-cpu} * \frac{1}{1-net} * \frac{1}{1-mem} \qquad (3)$$

If there are multiple hotspots, the system can select the physical machine of the largest Volume, and move virtual machine of the largest Volume / RAM (VSR) in the physical machine. VSR is the volume by size of memory ratio. Where 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 and does not take into account energy conservation and migration cost. Figure 3 presents the resource utilization of virtual machines of physical machine (PM1.)

The paper aims at saving migration cost when designing the scheduling algorithm, it is necessary to minimize the times of migration mean while eliminating the hotspot.

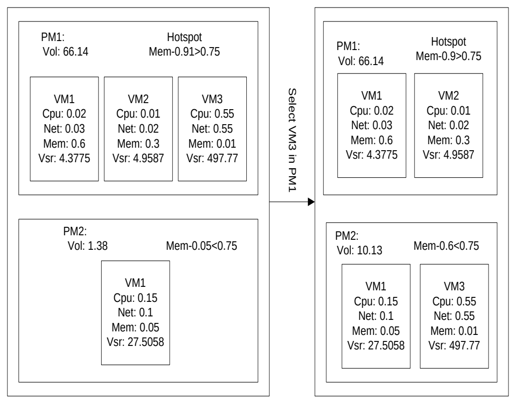

Figure 2. Relocation by black-gray box algorithm

Although the memory usage of the physical machine (PM1) exceeds the threshold, it cannot eliminate the hotspot after migration. In figure 2, the system chooses the virtual machine (VM3) depend on the VSR value to be migrate using the Black-Gray box (BGB) migration. Using this algorithm the system may generate new hotspots because of sudden change.

3.3.2. Balance Migration

This paper defines the balance function which denotes squared differences which every resource reach the optimal share [8].

The balance function is:

$$balance = \sum_{i=1}^{3} \left(\frac{r_i - r_{best}}{r_{best}} \right)^2 \qquad (4)$$

Where r_i is the i^{th} kind of resource and r_{best} is the parameter for the optimal share. For example, if the usage of resource exceeds 75%, the physical node is hotspot and 65% is the optimal share. Balance migration is to move out of the VM whose memory should be as small as possible and to make sure the usage of physical nodes can reach optimal share as possible after the migration.

In figure 3, virtual machines (VM1, VM2, and VM3) are running from physical machine (PM1). The mathematical expectation of PM1 memory usage exceed the threshold, PM1 will be occurred overloading problem. The system will select the VM2 which has the largest balance to the least loaded physical machine (PM2) and it will be eliminate the hotspot.

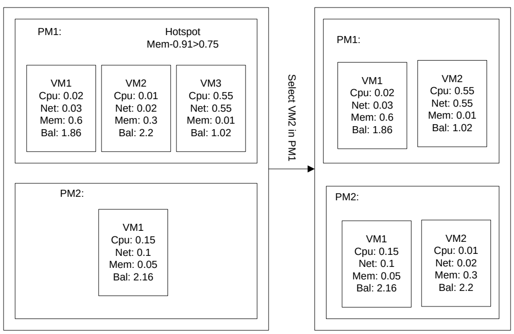

Figure 3. Relocation by balance migration algorithm

3.3.3. Performance Evaluation

In figure 4, there are two physical machines (PM1 and PM2). Three virtual machines (VM1, VM2 and VM3) are running in PM1 and next VM1 is running in PM2. When the memory usage of the PM1 exceeds the threshold, it will be marked the hotspot and move the VM to the under loaded PM2. By using BGB migration, it cannot eliminate the hotspot after migration. In balance migration, it can eliminate the hotspot after migration. Figure4 shows that the Balance migration is compared with Black-Gray box migration. A major problem of Black-Gray box migration is that it chooses the virtual machine with largest VSR value to migrate. This cannot guarantee the elimination of hotspot after migration. In balance migration, it can complete the migration and to eliminate the hotspots than Black-Gray box migration.

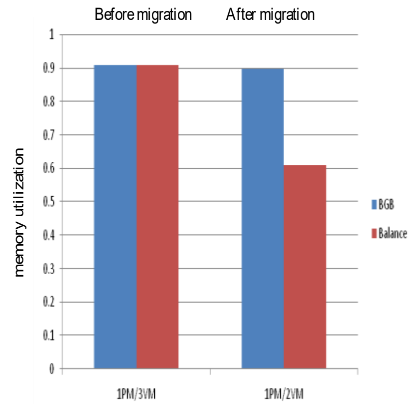

Figure 4. Hotspots comparison based on BGB and balance migration

4. Conclusion

This paper argued that virtualization provides significant benefits in data centers by enabling virtual machine migration to eliminate hotspots. A system that automates the task of monitoring and detecting hotspots, determining a new mapping of physical to virtual resources that necessary migration for the VMs. The Balance migration can eliminate the hotspots after migration than the Black-Gray box. It will be efficiently remove overloads without deviation of SLA on server and it also has advantages in migration cost.

References

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand,T.Harris, A. Ho, R. Neuge-bauer, I. Pratt, and A. War?eld, “Xen and the art of virtualization,” *In Proceedings of Symposium on Operating Systems Principles (SOSP)*, 2003.

[2] C. Clark, K. Fraser, S. Hand, J. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in NSDI, 2005.
[3] A. Gambi, M. Pezze, M. Youong, "SLA Protection Models for Virtualized Data Centers", *In Proceedings of the Workshop on Software Engineering for Adaptive and Self-Managing Systems (pp.10-19). IEEE,* 2009.
[4] C. Hyser, B. McKee, R. Gardner, Brian J, "Autonomic Virtual Machine Placement in the Data Center", HP Laboratories, HPL-2007-189, 2008.
[5] D.A Menasce, and M.N. Bennani, "Autonomic Virtualized Environments", *In proceedings of International Conference on Autonomic and Autonomous Systems, ICAS '0,* 2006.
[6] R. Nathuji and K. Schwan, "VirtualPower: coordinated power management in virtualized enterprise systems", *In proceedings of In 21st Symposium on Operating Systems Principles (SOSP),* 2007.
[7] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and Gray-box Strategies for Virtual Machine Migration," *In Proceedings of Symposium on Networked Systems Design and Implementation (NSDI),* 2007.
[8] X. Xu, Y. Jiang, and J. Wan, "A Balanced Migration Policy in Virtualized Cluster", *In Proceedings of the International Symposium on Intelligent Information Systems and Applications (IISA'09),* 2009.
[9] "VMware Infrastructure: Resource management with VMware DRS", VMware Whitepaper, 2006.