Live Migration Decision based on Integrated Server-Storage Virtualization

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

Abstract

Virtualization can provide significant benefits in data centers by eliminating overload nodes. An increasing workload can be handled by allocating the resources to a virtual server and storage, if idle resources are available on the physical servers, or by simply migration of the virtual machine (VM) to a less loaded physical server. In case of migration, which VM migrate to which server is important for resource utilization. There are many considerations; application-level statistics (such as request rate, service time and response time) and physical resources in host (such as memory, CPU, I/O bandwidth, network bandwidth) and storage node (such as space capacity and I/O capacity). This paper proposes a VM migration decision applying G/G/1 queuing theory. It reduces the overhead in changing data center across multiple resource layers-server and storage nodes. Finally, we will measure how much efficient on open source Xen virtualized infrastructure as an ongoing work.

1. Introduction

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. Application users have always requested storage administrators to provision more capacity than needed in order to guarantee support for future growth. The maximum capacity is seldom reached and therefore results in unused space and wasted resources. It also encroaches into the space that could have been used by other applications.

An important characteristic for a well managed data center is its ability to avoid hotspots. Overloaded nodes (servers, storage) often lead to performance degradation and are vulnerable to failures. To alleviate such hotspots, load must be migrated from the overloaded resource to an underutilized one. Integrated server and storage virtualization can play a key role by migrating virtual machines or virtual disks without causing disruption to the application workload. Migration is further complicated by the need to consider multiple resources—CPU, network, and memory—for each application and physical server and physical storage. However, intelligently deciding which virtual items (VM or

Vdisk) from all that are running on the overloaded resource is to be migrated and to where can be a challenging task.

The rest of this paper is organized as follows. It presents some extended motivation and address related work in Section 2. In Section 3, the resource provisioning is described in detail. Section 4 discusses the implementation of system architecture and it will be followed by conclusion in section 5.

2. Related Work

The VMware Distributed Resource Scheduler (DRS) [8] 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. Sandpiper [7] describe an enhanced load balancing system for Xen that accounts for network usage as well and show that Sandpiper is able to resolve single server hotspots within 20 seconds and scales well to larger data center environments. N. Bobroff, A. Kochut, and K. Beaty [2] describe other resource management and determine the best placement of virtual machines on physical machines.

However such work is restricted only to the server level in data centers and does not take into account the hierarchical data center topology spanning servers, and storage nodes. Techniques for storage migration have also received considerable attention, ranging from efficient data migration methods to reducing application impact [1], such techniques can be used for performing load balancing at the storage level. SoftUDC [5] describes a vision for a virtualization based data center combining server, storage and network virtualization. It controls system spans all the virtual machine monitor (VMMs), providing a unified console for data center resources and functions. From this console, an administrator can deploy services and modify virtual farms without reconfiguring the physical infrastructure. It also automates many common administrative tasks such as performing routine maintenance, deploying new applications, and dynamic load balancing. Parallax [6] also describes an integrated server storage virtualization technique to scale to large amounts of storage but only for direct attached non-SAN environments.

In C. Clark [3] 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. A. Gambi, M. Pezze, M. Youong [4] 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.

In this paper, the system will be described that integrates server and storage virtualization in a real data center. It tracks application computation (in the form of VMs) and application data (in the form of Vdisks) and continuously monitors the resource usages of servers and storage nodes in the data center. It can also orchestrate live migrations of virtual machines and virtual disks in changing data center conditions.

3. Resource Provisioning

Existing approaches to dynamic provisioning have either focused on dynamic replication, where the number of servers allocated to an application is varied, or dynamic slicing, where the fraction of a server allocated to an application is varied; none have considered application migration as an option for dynamic provisioning, primarily since migration is not a feasible option in the absence of virtualization. Since migration is transparent to applications executing within virtual machines, the proposed system considers the third approach—resource provisioning via dynamic migrations in virtualized data centers.

In this system, the provisioning component needs to estimate the peak CPU, network, I/O and memory requirements of each overloaded VM and peak space, I/O requirements of each Vdisk and ensures that the SLAs are not violated even in the presence of peak workloads.

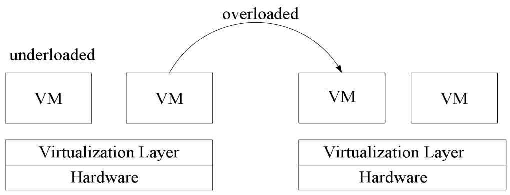

Figure 1. Load balancing for virtual machine

Figure 1 shows the four VMs running from the two physical servers and if the application requests are heavily utilized to the VMs from the server, it will be overloaded nodes (servers) and often lead to performance degradation and are vulnerable to failures. To alleviate such hotspots, load must be migrated from the overloaded resource to an underutilized one.

Managing storage allocations to support dynamic environments can be a time-consuming process that requires extensive coordination between application owners, virtual machine owners and storage administrators, often resulting in downtime for critical applications. Thin provisioning allows virtual disks to use only the amount of storage space they currently need.

Allocation Type of Disks: A virtual machine’s disk (VMDK) can be allocated as one of three types shown in table 1.

Zeroing: It is the process of disk blocks that are overwritten with zeroes to ensure that no prior data is leaked into the VMDK that is allocated with these blocks.

Thick Disks: There are two types of thick disk allocation types: zeroed thick and eager zeroed thick. Zeroed thick is the default allocation type for virtual disks on hosts. Eager zeroed thick pre-allocates and dedicates a user-defined amount of space for a virtual machine’s disk operations.

Thin Disks: TP involves the creation of thin virtual disks, which are VMDK files. Thin virtual disks are not any larger than they need to be (that is, they are not pre-allocated), and they are not zeroed out until run-time. Blocks in a thin VMDK file are not written during non-write operations like read and backup.

Table 1. Three types of VMDK disk allocation

Allocation Type	Pre-allocated	Zeroing
Zeroed thick (default)	Yes	Run-time
Eager zeroed thick	Yes	Create-time
Thin	No	Run-time

Thin provisioning (TP) is a method of optimizing the efficiency with which the available space is utilized in storage area networks (SAN) [9]. TP operates by allocating disk storage space in a flexible manner among multiple users, based on the minimum space required by each user at any given time.

Figure 2 presents the disk allocation for resource provisioning in virtual disks (VMDK). When a VMDK file is allocated, it can be allocated as either thick or thin. In thick allocation, VM1 will be fixed in storage space and the rest of VMs will be provision in storage space because of thin allocation.

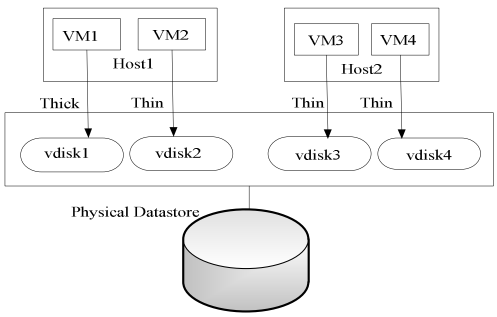

Figure 2. Disk allocation for virtual machine

Furthermore, delay during the process of storage allocation at any layer, storage to application can result in prolonged application downtime. By eliminating the need to periodically provision more capacity, our system resource provisioning applying G/G/1 queuing theory will eliminate the application downtime. We model a server at system as a G/G/1 system, since it is sufficiently general to capture arbitrary arrival distributions and service time distributions.

4. System Architecture

Firstly, virtual machine monitor(VMM) will access to application (AP) level logs, information contained in the logs can be utilized to estimate the peak resources needs of the application and will estimate the resources for CPU, network, I/O and memory usages of the physical server using the G/G/1 queuing theory. To estimate peak resource needs, the peak request arrival rate is first estimated. Since the number of serviced requests and the number of dropped requests 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. Figure 3 shows the system architecture running from the two physical machines (PM1 and PM2) and it comprises of estimate resources and load detection.

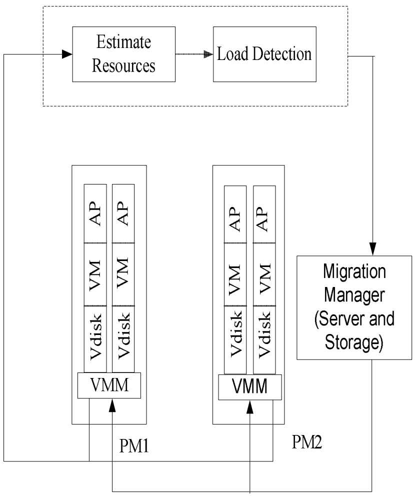

Figure 3. System architecture for live migration decision

Estimate Resources: 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}, λ_{mem} and λ_{sp} 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.

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
λ_{sp}	The current space 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

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

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 λ_{cap}, λ_{mem} and λ_{sp}. They represent the current capacity of the VM and Vdisk. To service the estimated peak workload λ_{peak}, the current CPU capacity needs to be scaled by the factor $\frac{\lambda_{peak}}{\lambda_{cap}}$, the current memory capacity needs to be scaled by the factor $\frac{\lambda_{peak}}{\lambda_{mem}}$ and the current space capacity needs to be scaled by the factor $\frac{\lambda_{peak}}{\lambda_{sp}}$. If the VM is currently assigned a CPU weight w1, memory weight w2 and disk space weight w3, its allocated share needs to be scaled up by the factor $\frac{\lambda_{peak}}{\lambda_{cap}}$, $\frac{\lambda_{peak}}{\lambda_{mem}}$ and $\frac{\lambda_{peak}}{\lambda_{sp}}$ 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 and the I/O bandwidth is estimated as the product of estimated peak arrival rate λ_{peak} and the maximum requested file size f; b is the amount of data transferred over the network to service the peak workload. The mean request size can be computed from the server logs.

Since a VM or a server can be overloaded along one or more of four dimensions–CPU, network, I/O and memory and a Vdisk or storage can also be overloaded along one or more of two dimensional-space and I/O.

Load Detection: Our system monitor defines two new metric that capture the combined CPU-network-memory load of a virtual and physical server and also capture the space-I/O load of a virtual and physical storage. The volume of a physical or virtual server is defined as the product of its CPU, network, I/O and memory loads.

$$Vol1 = \frac{1}{1-cpu} * \frac{1}{1-net} * \frac{1}{1-io} * \frac{1}{1-mem} \quad (2)$$

The volume of a physical of virtual storage is defined as the product of its space and I/O loads

$$Vol2 = \frac{1}{1-sp} * \frac{1}{1-io} \quad (3)$$

Where cpu, net, io and mem are the corresponding utilizations of that resource for the virtual or physical server and space and I/O are the resource utilizations of virtual or physical storage. 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 Manager: In a migration manager, VMs or Vdisks to migrate, the algorithm orders physical servers and physical storage in decreasing order of their volumes.

5. Conclusion

In this paper, we presented our design of a virtualized data center with integrated server and storage virtualization along with the resource provisioning. 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 and Vdisks respectively. An evaluation of Xen based prototype will be showed that VM and Vdisk migration is a viable technique for rapid hotspot elimination in data center environments and it will be efficiently remove overloads without deviation of SLA on server and storage nodes.

References

[1] E. Anderson, J. Hall, J. Hartline, M. Hobbs, A. Karlin, J. Saia, R. Swaminathan, and J. Wilkes, "An experimental study of data migration algorithms," in Proceedings of International Workshop on Algorithm Engineering, 2001.

[2] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," in Proceedings of the 10th IEEE Symposium on Integrated Management (IM), 2007.

[3] 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.

[4] A. Gambi, M. Pezze, M. Youong, "SLA Protection Models for Virtualized Data Centers", 2009.

[5] M. Kallahalla, M. Uysal, R. Swaminathan, D. E. Lowell, M. Wray, T. Christian, N. Edwards, C. I. Dalton, and F. Gittler, "SoftUDC: A software based data center for utility computing," IEEE Computer, 2004.

[6] A. Warfield, R. Ross, K. Fraser, C. Limpach, and S. Hand, "Parallax: managing storage for a million machines," in HOTOS'05: Proceedings of the 10th Conference on Hot Topics in Operating Systems, 2005.

[7] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and Gray-box Strategies for Virtual Machine Migration," in Proceedings of Symp. on Networked Systems Design and Implementation (NSDI), 2007.

[8] VMware Infrastructure: Resource management with VMware DRS, VMware Whitepaper, 2006

[9]http://searchstorage.techtarget.com/sDefinition/0,sid5_gci1214867,00.html