2016 45th International Conference on Parallel Processing

Guaranteed Bang for the Buck: Modeling VDI Applications with Guaranteed Quality of Service

Hao Wen, David H.C. Du
University of Minnesota, Twin Cities
{wenhao,du}@cs.umn.edu

Milan Shetti, Doug Voigt
HP Storage
{milan.shetti,doug.voigt}@hpe.com

Shanshan Li
National University of Defence Technology
shanshanli@nudt.edu.cn

***Abstract*—In cloud environment, most services are provided by virtual machines (VMs). Providing storage quality of service (QoS) for VMs is essential to user experiences while challenging. It first requires an accurate estimate and description of VM requirements, however, people usually describe this via rules of thumb. The problems are exacerbated by the diversity and special characteristics of VMs in a computing environment. This paper chooses Virtual Desktop Infrastructure (VDI), a prevalent and complicated VM application, to characterize QoS requirements of VMs and to guarantee QoS with minimal required resources. We create a model to describe QoS requirements of VDI. We have collected real VDI traces from HP to validate the correctness of the model. Then we generate QoS requirements of VDI and determine bottlenecks. Based on this, we can tell what minimum capability a storage appliance needs in order to satisfy a given VDI configuration and QoS requirements. By comparing with industry experience, we validate our model. And our model can describe more fine-grained VM requirements varying with time and virtual disk types, and provide more confidence on sizing storage for VDI as well.**

***Index Terms*—VDI; modeling; QoS;**

I. Introduction

With the rapid development of virtualization technology, traditional data centers are gradually superseding physical machines with virtual machines (VMs) to provide services. Apart from improving hardware utilization, virtualization enables migrating applications seamlessly to a different physical host for the purpose of load balancing, planned software/hardware upgrades, etc. To avoid migrating data along with the in-memory state of the virtual machines, virtual machine data is stored on shared storage. In the shared storage architecture, multiple VMs/applications will compete with each other for IO resources and capacity of the storage system, so how can we ensure quality of service (QoS) with minimum required resources?

In this paper, we investigate QoS guarantees for one popular type of VM application, Virtual Desktop Infrastructure (VDI) [12]. VDI runs VMs with different operating systems and applications on several physical servers in a data center. This type of VMs is also referred to as virtual desktops. Current VDI sizing work [8], [4], [5] is unable to give a description of accurate QoS requirements of virtual desktops. They either use rules of thumb to guide storage provisioning [7] or test the performance of their storage array under a given fixed number of VDI instances [5]. To ensure performance in practice, people always over provision storage resources, which may cause a huge amount of waste. Otherwise, VDI users will be at risk of degraded performance. In addition, how CPU, memory and storage resources of virtual desktops are configured may have a big influence on the IO behavior of VDI. For example, each virtual desktop may access multiple heterogeneous data disks at different time. Each data disk will see significantly different IO workload. Therefore, how physical storage is configured and where these data disks are placed will impact the QoS requirements. Unfortunately, current VDI sizing work fails to give a clear definition of VDI configuration.

When considering VDI QoS, CPU, memory and storage can all be potential bottlenecks for VDI. We assume enough CPU and memory are provided in a data center. Besides, VMs can be migrated to another host [10] if the current host utilization is high. In this paper, we focus on storage resources and storage QoS guarantees.

Our objective is guaranteeing QoS of VMs with minimal storage resources. It relies on many practical performance factors such as capacity, throughput, latency, etc. We first create a model to describe IO behaviors of a single virtual desktop, as well as a group of virtual desktops. With the model, we are able to tell when and where the bottlenecks occur. Based on this foundation, we can tell what capability a storage appliance needs in order to satisfy a given VDI configuration and QoS requirements.

To create such a model, we need to know the detailed implementation of VDI, virtual desktop types and access patterns of virtual desktops. The implementation includes the organization of underlying storage and the composition of each virtual desktop. Considering that there are multiple virtual desktop types, the model should adapt to both homogeneous and heterogeneous combinations of virtual desktops. The access pattern of a virtual desktop is affected by its current stage and the data disks it is accessing at different stages. Each desktop will undergo certain stages (boot, login or steady state) during its life cycle, and access multiple different data disks at different stages. Those data disks have different functions and see distinct IO access patterns. When large numbers of virtual desktops arrive at different time, the aggregation effect of IOs will lead to more variance of storage access.

Our contributions can be summarized as follows:

- We describe a representative VDI configuration and organize it in terms of QoS.

2332-5690/16 $31.00 © 2016 IEEE
DOI 10.1109/ICPP.2016.55

- We propose a system model to describe the storage QoS requirements of both homogeneous and heterogeneous VDI VMs. To the best of our knowledge, this is the first model to describe IO behaviors of a real life VDI system.
- We generate the QoS requirements of VDI and determine bottlenecks on specific target virtual disks at a specific time by plugging in the traces from HP into the model.

II. BACKGROUND

Virtual Desktop Infrastructure (VDI) is a virtualization solution to provide desktop environment to remote users. VDI runs desktop operating systems on virtual machines (VMs) in a data center and presents desktops as normal ones to users. Thus a virtual desktop (VD) is referred to as a desktop running in a VM. According to various types of data, a virtual desktop can be associated with multiple different virtual disks. These virtual disks can reside either on local or shared storage in a data center. Virtual desktops (VDs) are managed centrally in VDI. A user can use client devices such as personal desktop computers, tablets and mobile phones to connect to and operate his/her virtual desktops. In VDI, all virtual desktops can be referred to as clones. They are clones of a master image. A master image is a VM template from which other virtual desktops will originate.

In the remainder of this section, we will first introduce different clone types. Next we will describe how virtual desktops are assigned to users. Finally, we will introduce how we get different virtual desktop types by combining clone types and assignments. And for each virtual desktop type, the associated data disks are introduced.

A. Clone Type

There are mainly two types of virtual desktop clones. One is the full clone. A full clone will copy the master image into its own virtual disk (not shared). The other type is the linked clone. Different from full clones, linked clones will share same OS data as long as they are linking to the same replica (a clone of the master image). Each replica will serve as a common base for a group of linked clones.

B. Virtual Desktop Assignment

There are mainly two types of virtual desktop assignment: dedicated assignment and floating assignment. Dedicated assignment will assign a virtual desktop exclusively to certain users. After the assignment, when a user tries to log into his virtual desktop, it is always the same VM serving the user. Both full clones and linked clones can be dedicated. Floating assignment will assign a virtual desktop arbitrarily to users. Each time a user logs in, he/she may be assigned to a different VM. Only linked clones can be assigned as floating. Dedicated assignment has the advantages of good virtual desktop launching speed and user data access speed, but floating assignment is better for the whole system to flexibly allocate resources.

C. Virtual Desktop and Associated Disks

Combining clone type with assignment, we have floating linked clone, dedicated linked clone and dedicated full clone. For each type of virtual desktop, a different set of virtual disks will be associated. Overall, virtual disks can be master image, replica, primary disk, persistent disk, remote repository and full clone disk.

Floating linked clone. By the definition of linked clone, we know every linked clone will be linked to a shared replica. In the linked clone pool, we should provision spare space for multiple replicas with different operating systems. Besides the shared replica, a primary disk containing the essential system data that is needed for each linked clone to remain linked to the shared replica and to function as an individual desktop. Floating linked clones are usually configured to not save user profiles and user data in their local virtual disks. User profiles are preserved in a remote repository independent of the virtual desktop. Each user has his own repository. Typically, they are stored in a NAS (Network Attached Storage) device.

Dedicated linked clone. Dedicated linked clone includes those data disks essential for linked clone: replica and primary disk. But what is unique is that in dedicated linked clone, a separate persistent disk can be configured to store user profiles and user data. This disk is dedicated to a user. Attaching a persistent disk to a linked clone virtual desktop makes that virtual desktop dedicated to the user. A remote repository is also needed to permanently store the user data and profile.

Full clone. Full clone is always dedicated. Each full clone is an independent virtual desktop. So full clone uses its own full clone disks, its regular virtual disks to store operating system, user profiles and user data.

III. SYSTEM MODEL

In order to understand the QoS requirements of a VDI system, we propose a model to describe VDI system in a data center based on which we can infer when and where the bottlenecks are. Since virtual desktops run in VMs, we describe the IO behaviors of VMs that hosting virtual desktops in the model. We will first model a single VM and then integrate different types of VMs to model a large number of VMs in VDI.

A. VM Life Cycle

A VM in VDI has multiple stages during its life cycle. Each stage will show distinct IO behaviors. Overall, a VM life cycle has these stages: **boot, login, steady state and logoff**. In boot state, VMs are booting. If those desktops are powered on at the same time, or concentrating within a small time period, it becomes a storm. After desktops are powered on, users will log into the desktops. Since the boot stage can be a storm, the login stage can also be a storm just after boot. After users log in, they start their everyday work and virtual desktops transit to steady state. The IO accesses should be very random, except for periodic synchronization with NAS in linked clones. Logoff is the final stage during VM life cycle. A final synchronization with remote repository should happen for linked clone.

B. Data Access Sequence

As we talked in the previous section, the virtual disks accessed by various types of virtual desktops will be different. Even for the same virtual desktop type, virtual disks may see distinct IO access patterns at different stages. We will discuss how virtual disks are accessed at each stage for each virtual desktop type.

Floating Linked Clone Data Access In Figure 1, the green dashed line and red dashed line show the data access in boot and login stage respectively. When a virtual desktop is boot, shared OS data have to be read from the replica first. These data will be loaded into VM memory to initiate a system boot. Those essential binaries, libraries, etc. will be written to the linked clone primary disk as well for future access. When a user tries to log in, the virtual desktop must load user profiles from remote repository to memory first to authenticate the user and then configure the desktop settings. Those user profiles will also be written to linked clone primary disk for future access. After the desktop is set, the virtual desktop goes to the steady state. These operations may need to access user data, like user's own documents, videos, photos, music, etc. stored in the remote repository. These data are downloaded to the primary disk when first accessed. All subsequent accesses are will be directed to the copies on primary disk. Any changes to the user data will be synchronized to the remote repository at regular intervals. Once the user logs off, that virtual desktop will be refreshed, so no user profiles and user data will be saved on that primary disk. When that user logs in his/her desktop again, a different VM may be assigned.

Dedicated Linked Clone Data Access The data access is shown by the solid lines in Figure 1. During the boot process, there is no need to load OS data from replica anymore as long as it is not the first boot of a fresh desktop. Those OS data are stored in primary disk already. During the login process and steady state, user profiles and user data are read from the persistent disk rather than from the remote repository. The persistent disk performs as a cache of remote repository. During steady state, the synchronization of user profiles and user data happens between persistent disk and remote repository.

Full Clone Storage Data Access Full clone is like a regular desktop. All information including OS data, user profiles and user data are stored in full clone disk. So all IO accesses are on this type of virtual disk during all stages.

Table I shows when each virtual disk will be accessed for each type of virtual desktop.

C. VMs Model

We define a model to answer at time t, how much data will be read from each virtual disk and how much data will be written to each virtual disk. The basic idea is to integrate all read IOs or write IOs happening on the same virtual disk at time t. In the following of this section, we will discuss the model for a single virtual desktop and multiple virtual desktops respectively.

1) Single VM

Overall, the size of data accessed on *target* at VM life cycle *stage* for a single VM can be calculated by formula 1. The target is the virtual disk that IOs will reach listed in table I. The stage is the VM life cycle. $RWper_{\text{stage,target}}$ can be read ratio or write ratio during different stages on different targets when we calculate the size of read and write respectively. The IO sizes $S^i_{\text{stage,target}}$ are several discrete values. Since there are many different IO sizes, here we will only choose several significant IO sizes at each life cycle stage on each target. An IO size is significant when it accounts for most of the IOs. We decide it by two factors: 1) the frequency of the IO size is high. 2) The total size of data transferred under this IO size is large. The percentage of each significant IO size can be denoted by $Psize^i_{\text{stage,target}}$. We sum all significant IOs. $E_{\text{stage,target}}(t)$ describes the expected number of IOs at time *t*, which tells how many IOs arrive at *target* at *stage* at time *t*. In practice, when calculating how many IOs are expected to come at time *t*, we can multiply by a small time interval dt (e.g., 1 second).

$$\begin{aligned}\sum_i & E_{\text{stage,target}}(t) \times dt \times RWper_{\text{stage,target}} \\ & \times S^i_{\text{stage,target}} \times Psize^i_{\text{stage,target}},\end{aligned} \tag{1}$$

2) Multiple VMs

Different from a single VM, more factors need to be considered when integrating a number of virtual desktops. 1) VMs will start to boot or arrive at different time. 2) IO behaviors of different virtual desktop types are different. 3) IO behaviors of VMs running different operating systems or user applications are different. When we look at the size of data accessed on each target, we can determine the virtual desktop types and the stages they are in according to table I.

a) Multiple VMs of the same type

If multiple VMs are of the same virtual desktop and have the same operating system type and user applications, their parameters are all the same. Therefore, we only need to consider how to integrate the IO requests of VMs at different stages. Overall, the size of data accessed on *target* at VM life cycle *stage* for multiple VMs of the same type can be calculated by formula 2. $N(x)$ indicates the number of VMs arriving at time $x (x \leq t)$ (VM arrival rate). For each group of $N(x)$ VMs that arrive at time x, $E_{\text{stage,target}}(t)$ describes the expected number of IOs at time *t* for that particular group of VMs. It tells how many IOs arrive at *target* at *stage* at time *t*. For all those VMs that are now at *stage*, they arrive during time interval $[t_1, t_2]$. For every time point in $[t_1, t_2]$, we calculate how much data will be read or written at *stage* and add them together. The other parameters are the same as single VM model.

$$\begin{aligned}\sum_{x=t_1}^{t_2} & [N(x) \times \sum_i (E_{\text{stage,target}}(t) \times dt \times RWper_{\text{stage,target}} \\ & \times S^i_{\text{stage,target}} \times Psize^i_{\text{stage,target}})]\end{aligned} \tag{2}$$

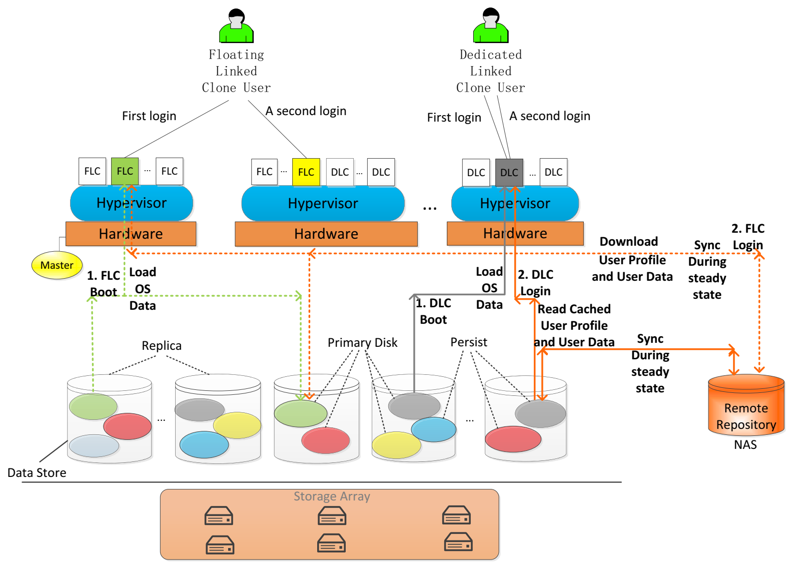


Fig. 1: VDI Storage Configuration.

TABLE I: Virtual Disks Accessed at Each Stage by Different Virtual Desktop Types
B=Boot L=Login S=Steady State

	Replica	Primary Disk	Persistent Disk	NAS	Full Clone Disk
Floating Linked Clone	B	B,L,S	-	L,S	-
Dedicated Linked Clone	-	B,L,S	L,S	S	-
Full Clone	-	-	-	-	B,L,S

b) Multiple VMs of Different Types

VMs with different virtual desktop types, operating systems and user applications will show different IO behaviors. We define a type of VM as VMs running the same type of virtual desktop, the same type of operating system and the same type of user application. For each VM type, we will apply formula (2) to calculate how much data are read from and written to each corresponding target at time t. The corresponding targets are chosen from Table I according to the virtual desktop type. Because in a data center each VM type accounts for a different proportion of IOs, when integrating them together, we need to plug in the weight of proportion of each type. We apply formula (2) for each VM type and then calculate the weighted average of all VM types to get the overall size of data accessed.

IV. Data Analysis

In order to get correct values of IO parameters in our model, we collected traces of different virtual desktop types in VDI during boot, login and steady state respectively running in HP storage. We then analyze IO behaviors of virtual desktops and derive those parameters that we need in model from the traces. The QoS demands are then generated. In this section, we will show a simulation of using our model by plugging in the traces to generate IO demands on each target.

Experiment Setup We assume a company uses VDI for work use and it has 5000 virtual desktop instances. Assume the arrival of employees follows Poisson distribution and the arrival rate is 10 per second. These 5000 virtual desktops can be all floating linked clone, all dedicated linked clone, all full clone or a mixture of all types. Next we will show under these occasions, how much data are accessed on each virtual disk since the first user arrives.

A. Floating Linked Clone

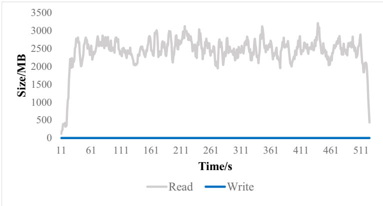
Figure 2 shows the size of data accessed on each of the targets from the first floating linked clone arrival (or start to boot) till all floating linked clones transiting to steady state.

On replica, as in figure 2(a) the IOs are read dominant and quite heavy. It rises sharply in the first 30s. In the next 500 seconds, the workload is relatively stable and stays high. Once all virtual desktops have arrived and tend to finish boot process, the IOs start to drop dramatically within the next 20 seconds. So overall, replica is read intensive for floating linked clone. Data read per second could be gigabytes for the above 5000 virtual desktops company. Unlike replica, the IOs on primary disk in figure 2(b) are more balanced as it is accessed during all stages. But it could still see a large volume of IOs. As shown in figure 2(c), size of data accessed on NAS is quite small. It will increase as more applications are installed.

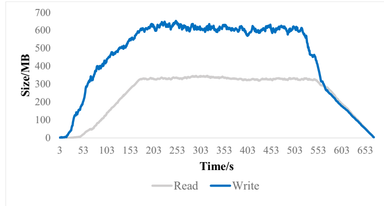
B. Dedicated Linked Clone

Figure 3 shows the size of data accessed on each of the targets from the first dedicated linked clone arrival (or start to boot) till all dedicated linked clones transiting to steady state.

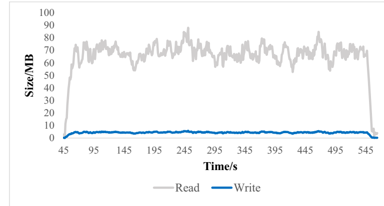
The IOs on primary disk of dedicated linked clone are much lighter than floating linked clone, as seen in figure 3(a). Once all VMs finish boot and login, the IOs on primary disk are



(a) Replica



(b) Primary Disk



(c) NAS

Fig. 2: Size of Data Accessed on Targets of Floating Linked Clone

minimal. On persistent disk, as we can see in figure 3(b), reads and writes mainly rise during the login stage and drop to minimum in steady state. Figure 3(c) shows IOs on NAS.

C. Full Clone

Figure 4 shows the amount of data read and written on full clone disk. We can see the total size of read is greater than the total size of write. And there is an obvious stage of high IOs where VMs are in boot and login stage. The IOs will drop down suddenly when all VMs finish booting and then drop to minimum slowly when all VMs transit to steady state.

V. Model Validation and Evaluation

In this section, we will first evaluate our model against the performance requirements provided by VMware. We calculate the IOPS demand on each type of virtual disk of different virtual desktops by plugging in the traces from HP into the model. By comparing with the IOPS requirements indicated by VMware, we not only demonstrate the correctness of our model but also show we can give a more accurate and fine-grained QoS requirements of VDI. Then we will discuss how we apply the model to configure storage system in order to meet QoS requirements of a typical VDI system in a more fine-grained way.

A. Compare with Industry Experience

When sizing VDI, VMware has given IOPS requirements as a rule of thumb in industry, as shown in table II. In our VDI traces, all virtual desktops run light load jobs, like text editing and pdf reading. They should be characterized as $Light$ or $Medium$. Based on the throughput of read and write of each virtual desktop and the significant IOs in the traces, we can easily calculate the average IOPS on each target of each virtual desktop type. Table III shows the average IOPS and the corresponding classification. We can see according to the rules of VMware, our virtual desktops are in $Light$ and $Medium$ classes. Thus our model is correct for describing QoS requirements of virtual desktops in VDI environment.

TABLE II: VDI IOPS Requirements from VMware

User Classification	IOPS Requirements Per User
Light	3-7
Medium	8-16
Standard	17-25
Heavy	25+

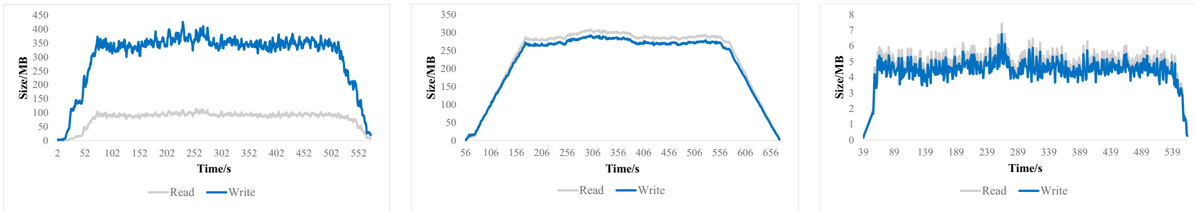
Our model is more accurate and fine-grained in describing QoS requirements of VDI. We know IOPS is widely used in industry to describe performance requirement and capabilities. VMware uses IOPS to guide the VDI sizing on performance. For example, they use the IOPS in Table II to calculate what the performance requirement is for each LUN when sizing storage for VDI [7]. However, only considering IOPS is less than enough. As we see from Section IV, the size of data read per second could be very large on replica during boot time. However, the IOPS on replica in Table III cannot show these information. If storage is allocated to replica based on that *Light* IOPS, user could experience long latency during boot time. Thus throughput and read write ratio should also be considered when sizing storage for VDI. Fortunately, our model can provide these information and guide on more fine-grained storage sizing. It can tell QoS requirements like storage capacity and throughput on each target directly and how they vary with time. And also it is not hard to get latency requirement based on Response Time/Throughput relationship, which can be easily found from SPC results [1], [6].

TABLE III: Average IOPS on Each Target of Each Virtual Desktop

Floating Linked Clone			Dedicated Linked Clone			Full Clone
Replica	Primary	NAS	Primary	Persistent	NAS	Full Clone Disk
5.45	12.25	0.61	5.27	8.26	0.22	9.52
Light	*Medium*	*Light*	*Light*	*Medium*	*Light*	*Medium*

B. Size Storage for VDI

When we size storage we can first base on the method of VMware [7], which only considers IOPS and storage capacity. Then we add more dimensions by considering the distinct IO access patterns in terms of read write ratio and throughput on different types of virtual disks. We suggest use tiered storage to guarantee storage QoS with minimum cost. For example, based on the 5000 virtual desktops example above, we know there is a heavy load on replicas and it is read dominant for floating linked clone. So it is perfect to deploy replica virtual disks on SSDs. We call this group of SSDs Tier-1 storage. Compared with replica, IOs on primary disk are more balanced. Considering it is more write intensive on primary disk, SSDs do not help much. We can place primary disks in HDDs. For better performance, we can use high performance HDDs, e.g. 15K HDD. We call it Tier-2 storage. For dedicated linked clone, IOs are minimal on replicas as long as it is not the first boot of a fresh desktop. Primary disks will see somewhat high volumes of IOs during boot process, but only less than 50% of those in floating linked clone, so we can still



(a) Primary Disk (b) Persistent Disk (c) NAS

Fig. 3: Size of Data Accessed on Targets of Dedicated Linked Clone

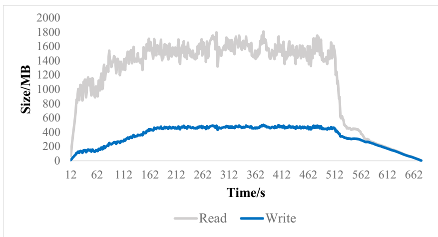


Fig. 4: Size of Data Accessed on Target of Full Clone

place primary disks on HDDs, Tier-2 storage. For full clones, since all IOs happen on one type of virtual disk and they are balanced, we can either place them on Tier-1 storage or Tier-2 storage according to the user requirements and overall system load. Other disks like persistent disks can be placed either in Tier-1 or Tier-2 storage based on the workload during steady state. If everything is placed in one tier HDD storage, a lot of capacity will be wasted in order to meet QoS requirement. Therefore, compared with a single HDD tier, tiered storage with several tiers can guarantee VDI QoS with minimal cost.

VI. Related Work

Currently, there are multiple VDI solutions like VMware Horizon View, Microsoft Virtual Desktop Infrastructure, RHEV VDI and Xen Citrix. No matter which solution is chosen, storage is a big hurdle on performance. VMware stated that over 70% of performance issues are related to storage. There are multiple storage solutions aiming to improve storage performance for VDI. VMware uses content-based read cache(CBRC)[9] to improve performance by caching common disk in ESX host server. Unlike VMware CBRC, which restricts cache access on the same host, Infinio builds a distributed version of host side cache [2]. Another solution from PernixData utilizes server flash to accelerate VDI performance [3].

Most of those researches trying to provide QoS guarantee solutions overlooked the characteristics of QoS requirements of VMs. Gulati et al. [11] did suggest how to improve performance of VMs, but it just uses Iometer to generate some workloads, thus cannot represent the QoS of VM. Commercial product manuals of VDI and underlying storage are based either on rules of thumb to guide storage provisioning [7] or test the performance of their storage array given a fixed number of VDI instances [5].

VII. Conclusions

In this paper, we create a model to describe the QoS requirements of one prevalent virtual machine type, VDI. We populate the parameters of the model with real traces that we collected from HP. Based on the model, we showed an example of how data access varied with time on different data disks for different types of virtual desktops. We demonstrated the correctness of our model and showed we could give a more accurate and fine-grained QoS requirements of VDI. Based on the QoS requirements generated and the bottlenecks determined, we are able to configure storage system to meet the QoS requirements in a more fine-grained way.

Acknowledgment: This paper is partially supported by NSF Award:1439622, 141913 and 1217569.

References

[1] Active spc-1 and spc-1/e results. http://www.storageperformance.org/results/benchmark_results_spc1_active/.
[2] Infinio. http://www.infinio.com/.
[3] Optimize vdi with server-side storage acceleration. http://pernixdata.com/sites/default/files/resources/PernixData_Optimize_VDI_WP_0.pdf.
[4] Server and storage sizing guide for windows 7 desktops in a virtual desktop infrastructure. http://www.vmware.com/files/pdf/view/Server-Storage-Sizing-Guide-Windows-7-TN.pdf.
[5] Sizing and best practices for deploying vmware view 5.1 on vmware vsphere 5.0 u1 with dell equallogic storage. http://en.community.dell.com/cfs-file/__key/telligent-evolution-components-attachments/13-3862-00-00-20-21-90-29/BP1033_2D00_Final_5F00_VMware_5F00_EqualLogic.pdf?forcedownload=true.
[6] Spc benchmark 1/energy extension official specification. http://www.storageperformance.org/specs/SPC-1_SPC-1E_v1.14.pdf.
[7] Storage considerations for vmware horizon view 5.2. http://www.vmware.com/files/pdf/view_storage_considerations.pdf.
[8] Vdi server sizing and scaling. https://www-lt.vmware.com/pdf/vdi_sizing_vi3.pdf.
[9] View storage accelerator in vmware view 5.1. https://www.vmware.com/files/pdf/techpaper/vmware-view-storage-accelerator-host-caching-content-based-read-cache.pdf.
[10] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pages 273–286. USENIX Association, 2005.
[11] A. Gulati, C. Kumar, and I. Ahmad. Modeling workloads and devices for io load balancing in virtualized environments. *ACM SIGMETRICS Performance Evaluation Review*, 37(3):61–66, 2010.
[12] A. Velte and T. Velte. *Microsoft virtualization with Hyper-V*. McGraw-Hill, Inc., 2009.