Understanding Storage Traffic Characteristics on Enterprise Virtual Desktop Infrastructure

Chunghan Lee Tatsuo Kumano Tatsuma Matsuki
Hiroshi Endo Naoto Fukumoto Mariko Sugawara
Fujitsu Laboratories Ltd., Kawasaki, Kanagawa 211-8588, Japan
{lee.chunghan, kumano_tatsuo, matsuki.tatsuma, endo-hiroshi, fukumoto.naoto, sugawara.mariko}@jp.fujitsu.com

ABSTRACT

Despite the growing popularity of enterprise virtual desktop infrastructure (VDI), little is known about its storage traffic characteristics. In addition, no prior work has considered the detailed characteristics of virtual machine (VM) behavior on VDI. In this paper, we analyze the enterprise storage traffic on commercial office VDI using designated VMs. For 28 consecutive days, we gathered various types of traces, including a usage questionnaire and active and passive measurements. To characterize the storage traffic, we focused on two perspectives: fibre channel (FC) traffic and VM behavior. From the FC traffic perspective, we found that read traffic is dominant, although the applications are similar to those in a previous small-scale VDI. In particular, the write response time of large transactions, e.g.,128 KiB, is strongly affected by a slight decrease in cache hits during an update storm. From the VM behavior, we found that all active user VMs generate only 25% of traffic. Although a few VMs generate massive traffic, their impact is small. These characteristics are unique in comparison with the small-scale VDI. Our results have significant implications for designing the next generation of VDI and improving its performance.

CCS Concepts

•**General and reference → Measurement; Performance;**

Keywords

Measurement; Performance analysis; Virtual Desktop Infrastructure (VDI)

SYSTOR 2017, May 22-24, 2017, Haifa, Israel
© 2017 ACM. ISBN 978-1-4503-5035-8/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3078468.3078479

1. INTRODUCTION

The past few years have seen dramatic growth in virtual desktop infrastructure (VDI) [12, 28]. VDI, whose benefits include improved operational efficiency and higher cost savings, has been used in real-world offices and universities. There are two types of VDI: pool-based VDI and VDI using designated virtual machines (VMs). The storage in the designated VMs is similar to remote storage in clouds. In both types, storage and the storage area network (SAN) greatly affect VDI performance. For instance, a large increase in the response time between VMs and storage can affect the quality of experience (QoE) of VM users and the overall VDI performance.

Many studies [4, 6, 9, 16, 17] have analyzed the common Internet file system (CIFS) and traffic in personal cloud storage. However, few studies have analyzed storage traffic characteristics on VDI by collecting traces from a specific component of a VDI system. Shamma et al. [25] analyzed a small VDI system (55 VMs) with actual VDI traces. Although they presented a workload analysis and the traffic characteristics, they only investigated specific monitoring components, e.g., a disk-level driver, on VMs.

The research goal of this paper is to provide a deep understanding of storage traffic characteristics in enterprise VDI using designated VMs from two perspectives: fibre channel (FC) traffic and VM behavior. We gathered actual data from a part of an enterprise VDI that provides office VDI to an IT company [1]. In this VDI, a designated Windows VM is provided to each user. Moreover, all VMs are always running regardless of logging events to avoid boot storms and increase VM connectivity. Between 2/22/2016 and 3/20/2016 (four weeks), we analyzed datasets corresponding to 79.8 TiB of storage traffic on approximately 300 VMs. The datasets were compiled from a VDI usage questionnaire, lists of applications on all VMs, FC traces

[1] We cannot disclose the VDI scale or the company because we have a contract with the company to operate their VDI system.

Table 1: Summary of major findings and their implications for the target VDI.

VDI Analysis	Findings	Implications
FC traffic	Read is dominant, although the applications are similar to the small-scale VDI [§3.2, §4.2] (different from [25]).	Cache and address management for read traffic should be enhanced in VDI storage.
	During an update storm, a large number of transactions access similar address blocks in bursts over a short period of time [§5.4] (similar to [14, 18, 22]).	A short time interval is required to detect the burst, and address prediction methods can help to reduce their response time.
	During the storm, the write response time of large transactions, e.g., 128 KiB, is sensitive to a slight decrease in write cache hit rate [§5.3, §5.4] (**N**).	The transaction size for write cache hits should be considered to improve overall VDI performance during the storm.
VM behavior	Although few VMs generate dominant traffic, their impact is small [§6.3] (different from [6, 9, 17]).	The VDI can perform sufficiently by using the uniform load balancing strategy.
	Anti-virus and Windows service are major causes of large traffic in idle VMs [§6.2] (similar to [25]).	These operations are essential, so a novel approach is required to control idle VMs.
	All active VMs generate only 25% of traffic, but heavy traffic from idle VMs affects their response time [§6.1] (**N**).	Traffic from active VMs should be more highly prioritized to shorten their response time.

N: new observation. § section index.

based on packet capture [27], and logs from storage, a VDI broker [12], and Windows Management Instrumentation (WMI) [21]. Hereafter, we refer to VMs connected to users as active VMs and VMs disconnected from them as idle VMs. Our unique datasets give us an opportunity to validate previous results, like those of Shamma et al. [25], and our key findings and implications (see Table 1 for a summary) provide a better understanding of the VDI system.

Our contributions are summarized below:
1) The traffic characteristics for large-scale VDI are compared with those for small-scale VDI [25].
2) VM behavior is comprehensively analyzed on the basis of storage traffic.
3) Our results are compared with those of past storage analysis studies.
4) A discussion of significant implications for future VDI and its performance improvement derives from the basis of our analysis.

The rest of this paper is organized as follows. Section 2 covers related work, and Sections 3 and 4, respectively, describe the measurements and the target VDI. Sections 5 and 6 present the results of the comprehensive analysis (FC traffic and VM behavior), and Section 7 discusses our findings and explains their implications. Section 8 concludes the paper with a brief summary.

2. RELATED WORK

This section gives an overview of previous related work, including that on the CIFS [16], a small-scale VDI [25], and personal cloud storages [6, 9, 17], from the perspectives of FC traffic and VM behavior. Although the applications running on our VDI are similar, storage characteristics differ from those in previous work [25]. The previous results were write-heavy in the input/output operations per second (IOPS) and read-heavy in throughput (measured in Mbps). The major cause of write-heavy was a large proportion of metadata. In the read traffic, *Windows* directory was frequently accessed and it generated a large proportion of read traffic. However, the read traffic is dominant at both IOPS and throughput in our results due to the idle VMs. Fundamentally, an update storm on our VDI differs from the traces [14, 22] of Microsoft Exchange e-mail server. But, our results show similar burstiness and spatial observation [18]. Additionally, we clarify how to affect the write transaction size during the storm. The periodical daily patterns were similar to those for an office CIFS [16] and mobile traffic [30] in a business area.

The previous studies focused on the unbalance usage of cloud storage. Drago et al. [6] presented that a heavy user group is responsible for 50% of Dropbox sessions in home networks. Liu et al. [17] showed that the small number of files accounts for as much as 70% of read requests. Gracia-Tinedo et al. [9] investigated user-level storage workloads on UbuntuOne. They found the inequality of the traffic across active users through Gini coefficient (0.89). The reason was that 1% of users account for the 65.6% of the total traffic. In our VDI, we find few VMs generate dominant traffic. However, Gini coefficient (0.709) on ours is smaller than UbuntuOne and 1% of active VMs generate only 16.6% of traffic. Compared with the studies, the workload is more homogeneously distributed across VMs on our VDI.

Extensive studies have investigated VM behavior with storage traffic and workload traces. Birke et al. [4] presented VM-level storage workloads on enterprise datacenters. Seo et al. [24] modeled I/O trace patterns with public traces [15]. Narayanan et al. [23] showed that workloads (such as web searches, content caching in edge nodes, and datacenter management software) have more read traffic than write traffic. Zhang et al. [31] investigated the workload of desktop applications (e.g. Microsoft Office, Skype, and Google Chrome) on 20 Windows desktops (Windows 7) and found that read traffic was mainly dominant in the top ones. The applications are similar to those in our target VDI, and the traffic pattern is also close to ours. Shamma et al. [25] introduced the storage workload of VDI, on which 55 VMs were running and that is mainly used for office work. They found that the anti-virus application generates many transactions at the peaks in workload. These findings are similar to ours, and we validate them on our large-scale VDI. In addition, we separate the storage traffic into active and idle VMs.

The main difference between our work and Shamma et al.’s is that we studied the VDI in detail using designated VMs with various datasets instead of simply measuring it from a specific viewpoint. Moreover, the VDI system (300 VMs) is larger than theirs (55 VMs). Thus, our results reveal the basic but essential storage traffic characteristics from which we can deeply understand the VDI.

3. METHODOLOGY & DATASETS

3.1 Measurement Methodology

There are three constraints on measuring commercial real-world VDI. One is that we cannot install monitoring components [25] on a VM or hypervisor. The VDI is in commercial production use, and it is unclear how the monitoring components affect the VMs and the entire VDI system. Another is that it is hard to ascertain which VMs are being used because they are always running. The other is that client contracts prohibit the management side from logging on to the VMs and inspecting them and their payload in packets. Thus, the management side can neither identify which applications are installed nor which ones are run frequently on VMs. To investigate the storage traffic characteristics of VDI, we used the following measurement methods:

VDI monitoring system: This system fetches performance metrics from the storage and VDI servers. First, we gathered the private storage MIBs for each LUN. Next, we collected metrics called Windows performance counters using WMI [21]. These metrics are associated with the virtual disks of individual VMs, such as *read/write bytes per second* and *read/write counts per second* [10]. This dataset was obtained every 30 seconds as mentioned above.

FC capture system [27]: This system stores the full payload as binary data and extracts a part of the FC headers. The packets are copied to a capture server using port mirroring. We calculated transaction size with *fc.relative_offset* at an FC header and the response time as the difference in the time stamps between the first and last packets in a transaction.

Connection broker [12]: This is used to identify the active VMs. It provides a login portal to users. Through the portal, the broker notes which users are currently active. We calculated the number of active VMs every 30 seconds.

Table 2: Summary of datasets.

Dataset collection term	28 days (2/22-3/20)
Active VMs (Total VMs)	262 (321)
Total connections	10,881
WMI and storage logs size	45.2 GiB
FC trace size	79.8 TiB
Total Read bytes	64.8 TiB (81%)
Total Write bytes	15.0 TiB (19%)
Total Read transactions	2,455,443,072 (73%)
Total Write transactions	898,281,224 (27%)

System profiler [11]: This is used to investigate the user applications. It provides a daily list of applications from all VMs. The original list includes Windows updates and application patches. Because the updates and patches are of no help in identifying a user’s applications, we empirically excluded them and categorized the types of applications. Process Monitor [20] is excluded in the profiler.

Usage questionnaire: The questionnaire was used to determine which applications were actually run. Even if many users install an application, they may not run it frequently. For instance, Python is installed by many users, but there is no evidence of its usage. As mentioned above, we never logged into the user VMs directly and could not gather the profiles of executed applications, so we used the questionnaires to gather information about their daily usage.

3.2 Datasets

We gathered the datasets from the various measurement methods for 24 hours each day from Monday 2/22/2016 00:00 to Monday 3/21/2016 00:00. Table 2 summarizes the various datasets. Read traffic is dominant in total bytes and transaction counts, although the applications are similar to those of the small-scale VDI [25]. Because this primary characteristic differs from that in previous results [25], we will clarify why the read traffic is dominant in this paper. Again, the previous work [25] only obtained datasets from a hypervisor. On the other hand, we obtained various types of logs and were able to use these datasets to precisely analyze our target VDI on the basis of not only FC but also VM behavior.

4. VIRTUAL DESKTOP INFRASTRUCTURE

4.1 Target VDI

Our target VDI is a part of a VDI system and has been in production use since April 2013. It consists of storage, six VDI servers (Windows 2008 R2), 10G Fibre Channel over Ethernet (FCoE) fabric, and a connection broker. An overview of the target VDI with the measurement systems is illustrated in Figure 1. Each VDI server has two physical CPUs and hosts about 50 VMs.

Table 3: Specifications of target VDI system.

VDI server	FUJITSU Server PRIMERGY RX200 S8	
	CPU	CPU Intel E5-2695v2 (2.4 GHz, 12 cores, 30 MiB) × 2
	Memory	256 GiB
Block storage	FUJITSU Storage ETERNUS DX90 S2	
	HDD	HDD 8 TB (900 GB SAS × 9, 10krpm) with RAID 6
Virtual Machine	OS	Microsoft Windows 7 Enterprise 32bit or 64bit
	Logical CPU	2 cores
	Memory	4 GB
	Disk	100 GB
Network	Commodity FCoE 10GbE switch fabric	
VDI software	Microsoft Windows Server 2008 R2(Hyper-V) + Citrix XenDesktop 5.6	

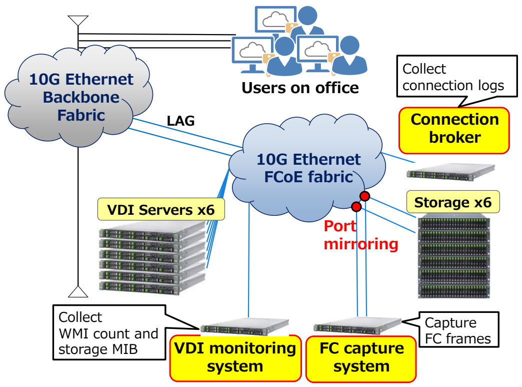


Figure 1: Target VDI with measurement systems.

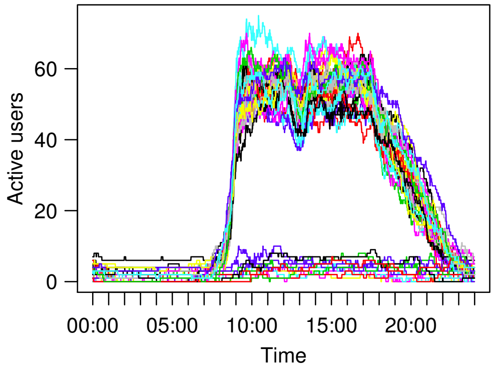


Figure 2: Active users for each day (each line indicates each day).

The tag VLAN (1002) with CoS (3) is used to separate VDI storage traffic on the FCoE fabric. The Xen Desktop [12] is only used as the connection broker. The host OS is Microsoft Windows Server 2008 R2, and the hypervisor is Hyper-V [19], which handles live migration of user VMs and their disks. In our VDI, the live migration is manually triggered by management side. Block storage is applied as a storage system to shorten the response time. In the storage, write-back for caching is used to improve the write performance. There is one LUN on one block storage, and it corresponds to one VDI server (50 VMs) only. This constraint makes it impossible to observe file sizes, types, and duplications at each VM, which is very different from a CIFS or cloud object storage service. Each component is summarized in Table 3.

As mentioned before, a designated Windows VM is provided to each user. It provides pre-installed applications, such as a mailer (Outlook), Microsoft Office (Word, PowerPoint, Excel), and anti-virus software (McAfee). If a user is disconnected from the VM on a pool-based VDI, a connection broker always allocates a new VM to the user, and then the user executes his/her programs again. However, such time-consuming operations are not required in a designated VM. Moreover, the user can install desired applications anytime and execute them freely, similar to ordinary cloud storage.

To clarify the active VMs, we used the connection logs [5] from the broker. These logs show which users are connected to their VMs and which are not. In Figure 2, each line represents the number of active users of each VM for each day. Unsurprisingly, there are two types of patterns: weekend and weekday. In the weekday pattern, there were approximately 50 to 70 active users from 08:00 to 20:00. On the other hand, there were few active users on the weekend. Therefore, we excluded the night and the weekend from our analysis lists. We also found a similar periodic pattern in the traffic (the periodicity is described in Section 5.2). Previous studies [4, 16] also showed a similar pattern. From the results, we chose the time range from 08:00 to 20:00 for our analysis.

4.2 User Applications

In this section, we mainly focus on what types of applications were frequently used on the target VDI. We manually and empirically classified them into nine categories and visualized them in a treemap. Figure 3 shows the treemap of applications installed on our target VDI. The thick-lined rectangles represent the nine categories. The largest rectangle is the pre-installed software, such as Microsoft Office and anti-virus. The rectangle for

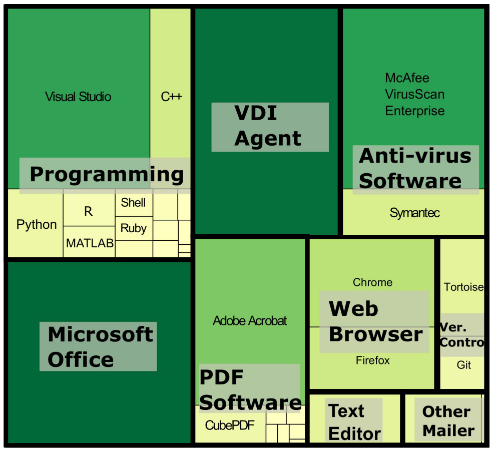


Figure 3: Treemap of applications installed on VDI (deeper colors and larger rectangles indicate larger proportions).

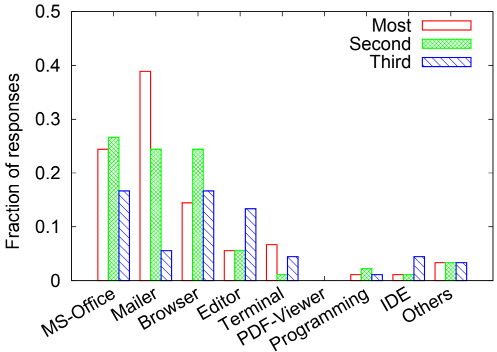


Figure 4: Questionnaire results (top three frequently used applications).

programming is also large and includes many types of applications, probably because several users installed different types of programming applications. Additionally, the web browser, text editor, and PDF software are also larger than other applications.

However, a question remains: were the installed applications actually executed by the users? Since we cannot access the user VMs directly, we distributed a set of questionnaires about the daily usage of the VMs and obtained 90 responses. Figure 4 shows the results for a question about the three applications the users most frequently used on their VMs. We learned from the responses that about 75.6% of users used their desktop for office work and 31.1% for backup storage only, in which the users back up their data, such as PowerPoint and Excel files, and restored them on their local PCs. The most frequently used applications were Microsoft Office, the mailer, and the web browser.

5. FC TRAFFIC-BASED ANALYSIS

5.1 Primary traffic characteristics

In this section, we present the primary characteristics of the FC traffic. To investigate changes in the transaction size, we calculated the traffic distribution on the basis of total counts and bytes. For write transaction size, the minimum and maximum values were 512 bytes and 512 KiB. Additionally, the write size distribution was wider than that in the previous study [26]. The read size was similar to the write size.

In the total counts, we found that approximately 80% of write transactions were smaller than 20 KiB. Unsurprisingly, we also found that 4 KiB had the largest ratio (approximately 40—50% of total transactions) in both read and write transactions, because 4 KiB is a common block size. In the total bytes, the transactions smaller than 4 KiB only constituted less than 20%. The ratio of 128 KiB was approximately 10%, but this was the largest ratio in the total bytes. Interestingly, these characteristics were similar to those [3, 13] between the mice and elephant flows in TCP/IP.

5.2 Periodicity of traffic pattern

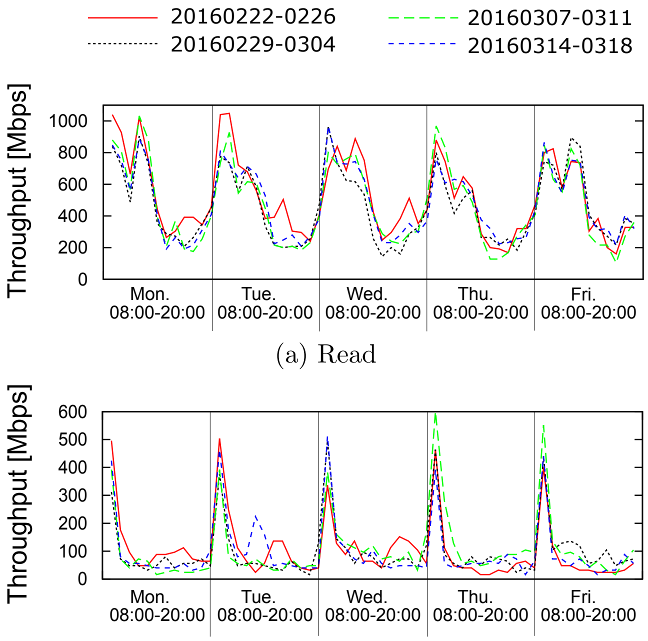
Here, we analyze the periodicity of traffic patterns. If there is periodicity, the traffic pattern will be easy to predict. The prediction results will then help to shorten the response time and improve cache memory usage.

We investigated the daily distribution of the throughput of both read and write traffic and found similar results over the four weeks. Figures 5(a) and 5(b) show the read and write throughput over the four weeks, respectively. These results are illustrated by storage logs and cover 08:00 to 20:00 on weekdays only. Thus, 12 hours equals one day. Each day, the throughput was higher in the morning than in the afternoon on the target VDI. Moreover, some spikes in read throughput are found around lunch time at the read. Next, we observed the throughput peaks at 09:00. The diurnal burstiness at write throughput is particularly pronounced. The major causes of the burstiness were update storms on the entire VDI system. The peak-to-mean ratio in write throughput is 6.29 and that in read throughput is 2.25. For each week, we found a strong periodic tendency of the throughput in both sets of results. At Lag=12 (1 day), the autocorrelation function (ACF) at both read and write is higher than 0.8. We also found a similar daily distribution for the IOPS, but these results are omitted for the sake of brevity.

In summary, there is an obvious periodicity pattern with three time bins: morning, lunch time, and afternoon. We will focus on these time bins to analyze the FC and VM behavior.

5.3 Response time

The response time is an important factor related to the overall VDI performance. It depends on many factors, such as cache, access patterns, and network con-



(a) Read

(b) Write

Figure 5: Daily distribution of read/write throughput between 08:00 and 20:00.

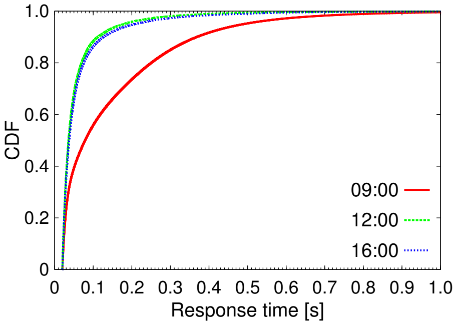
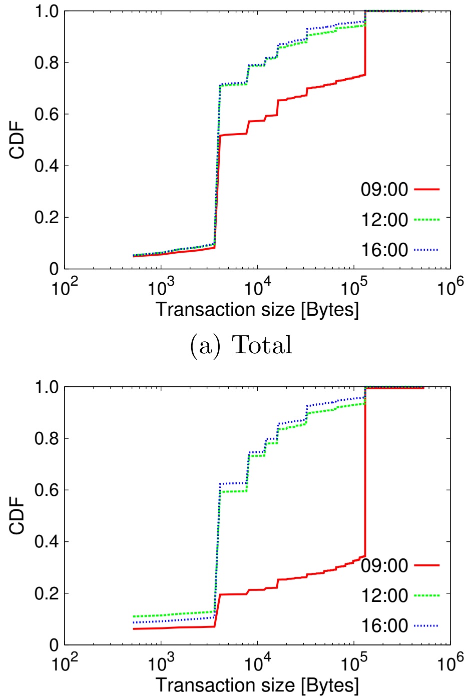


Figure 6: CDF of write response time over 20ms.

gestion. Typically, the recommended threshold of the response time is approximately 20 or 30 ms [29]. If the response time exceeds the threshold, the users experience a delayed response.

We can measure the response time of read/write transactions using the FC traces. Approximately 3.67% of the read was larger than 20 ms, while approximately 0.234% of the write was larger than the threshold. Through this analysis, we unfortunately were unable to find significant characteristics that contribute to the VDI storage traffic. Alternatively, we focused on the increased response time when the values are larger than the threshold in the morning (09:00), lunch time (12:00), and afternoon (16:00) time bins. As mentioned before, there was an update storm in the morning and the largest write throughput achieved in the morning. The CDF of the write response exceeding the threshold is shown in Figure 6. The CDF shape in the morning is different from the other bins. Moreover, 1.5% of the response time exceeded the threshold, and this was also higher than the entire ratio for the write (0.234%).



(a) Total

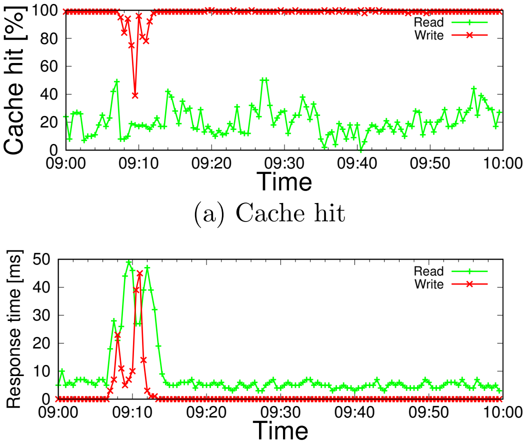
(b) Over 20 ms

Figure 7: CDF of write transaction sizes.

Since the write response time in the morning had different characteristics, we next ask: what are the write sizes distributed over the threshold? Figure 7 shows the CDF shape of write sizes for the time bins. In the total traces (Figure 7(a)), 4 KiB unsurprisingly has the largest proportion (approximately 50%). However, the CDF shape at 09:00 differs from those in the other time bins. In addition, 128 KiB has a larger proportion than the other sizes, such as 32 KiB. Next, the sizes over the threshold were extracted from the total traces. The CDF is shown in Figure 7(b). The CDF shapes in the other time bins are similar to those in the total traces. Meanwhile, 4 KiB greatly decreased from 50% to 12% in the morning. In particular, 128 KiB had the largest proportion (65%). Interestingly, this characteristic was only observed in the morning. Thus, the large writes were strongly affected in the morning. We will identify the major cause of the increased response time with the large writes in the morning in the next section.

5.4 Update storm (Burstiness)

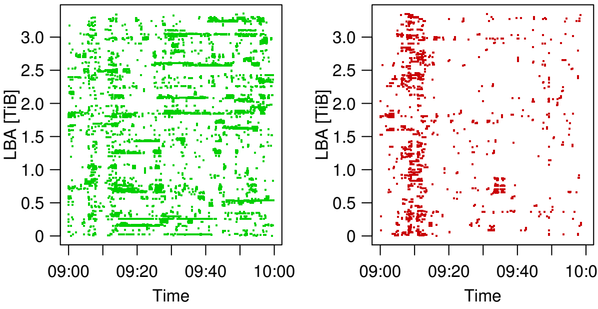
In this section, we investigate the cause of the increased response time and its impact. As mentioned in Section 4.1, McAfee is mainly used for anti-virus software, and the cause of the increased response time is its regular update. This update proceeds as follows. First, a McAfee agent is connected to an internal up-



(a) Cache hit

(b) Mean response time

Figure 8: Temporal distribution at LUN during update storm.

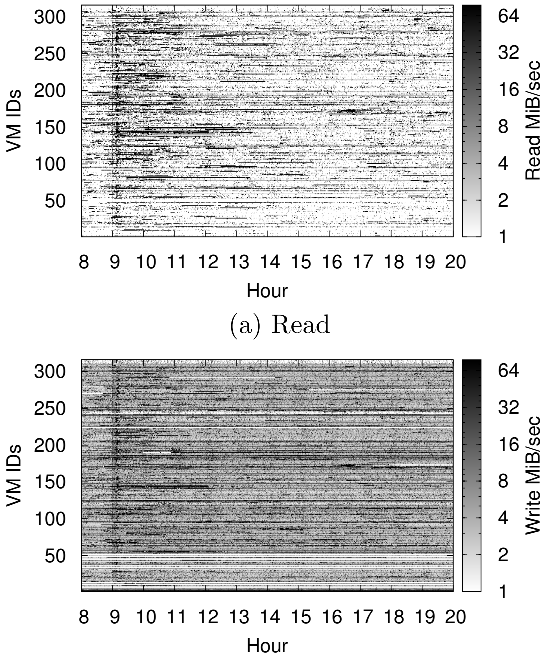


(a) Read (b) Write

Figure 9: Spatial distribution at LUN during update storm.

date server for a short period (09:05–09:20). The agent collects security events and reports them to the update server. Then it searches for the latest update patch (there are two types of patch, DAT and compressed DAT). Finally, the agent downloads the patch and installs it in the VM. The update is very simple but, because it is simultaneously executed by hundreds of VMs, it acts as an update storm.

During the update storms, the throughput of both read and write traffic increased dramatically to 6 or 7 Gbps. To clarify the detailed characteristics of a storm, we selected a representative LUN containing 47 VMs. All VMs in the LUN were related to this update storm. Again, there was only one LUN at one block storage. First, we calculated the read/write percentages for both the IOPS and throughput. In the storm, write traffic was dominant at both IOPS and throughput. In the IOPS, it was 57%, while the read was 43%. Similarly, write accounted for 68% of throughput, while read accounted for 32%. The characteristics will help in reproducing the storm using storage benchmarks [1, 2, 8].



(a) Read

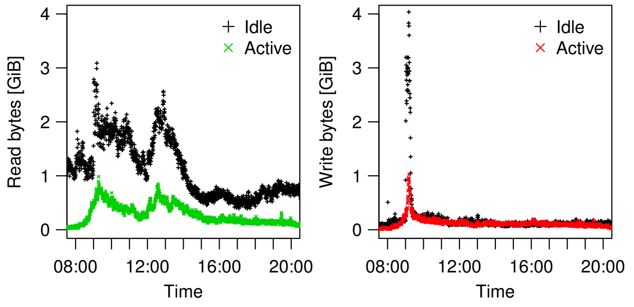
(b) Write

Figure 10: VM-level Throughput.

To clarify the other characteristics, such as storage cache hits and response time, we focused on the temporal distribution from the storage logs every 30 seconds. The cache hits and mean response time are shown in Figures 8(a) and 8(b). Although the write cache hit rate is normally close to approximately 100%, it dramatically decreases to 40% during the storm. When the cache hit rate is slightly decreased, the response time is largely increased. The maximum mean response time is about 45 ms, which exceeds the threshold of response time (20 ms). Thus, the cache hit rate and the response time are found to negatively correlate. Next, the read cache hit rate is always lower than the write cache hit rate regardless of the storm. After the storm, the read response time is lower than the threshold. Additionally, the read cache hit rate widely fluctuated from 18% to 90% in the different time bins.

Normally, write-back cache can help to reduce the response time if there is enough cache memory. Again, write-back for write caching was applied in the target VDI. However, the memory was obviously not sufficient to accommodate all write transactions when many write transactions were aggressively generated during the storm. When there was a lack of cache memory with respect to the volume of write traffic, the large writes were much more affected than the small ones. As a result, their response time was greatly increased.

Because we cannot accurately observe the spatial distribution of an access pattern from the storage logs, we used the FC traces. Through the traces, we can identify the access patterns of the transaction in the LUN. The



(a) Read (b) Write

Figure 11: Aggregate throughput of active and idle VMs.

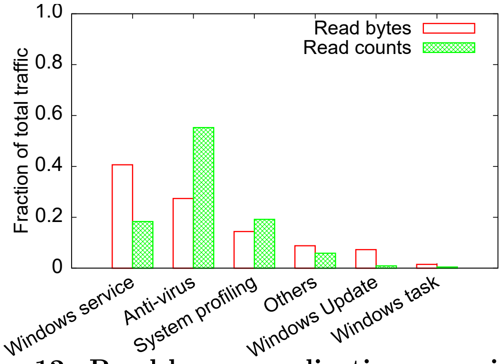


Figure 12: Read-heavy applications running on idle VM.

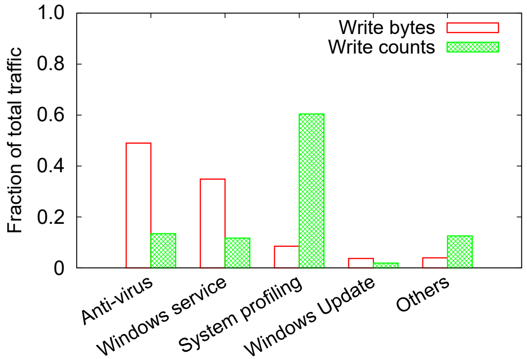


Figure 13: Write-heavy applications running on idle VM.

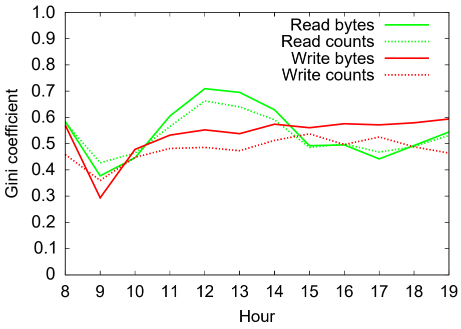


Figure 14: Gini coefficient for each hourly time bin

spatial distribution every 10 seconds during the storm is shown in Figure 9. If the read/write transactions are larger than threshold, we plotted a point in LBA unit (1 GiB). Many similar logical addresses were simultaneously accessed during a large number of write transactions, while the number of the read transactions was decreased because of the impact of the write transactions.

During the update storm, most VMs were forcedly frozen, so the user cannot execute any programs. The cause of decreased reads was that neither the active nor idle VMs can execute background processes or generate read operations. To mitigate the effects of the update storms, we analyzed the behavior of updates and dispersed the update schedule of VMs between 07:00 and 09:00. Since the updates were dispersed, the storm was mitigated on the target VDI.

6. CHARACTERIZING VM BEHAVIOR

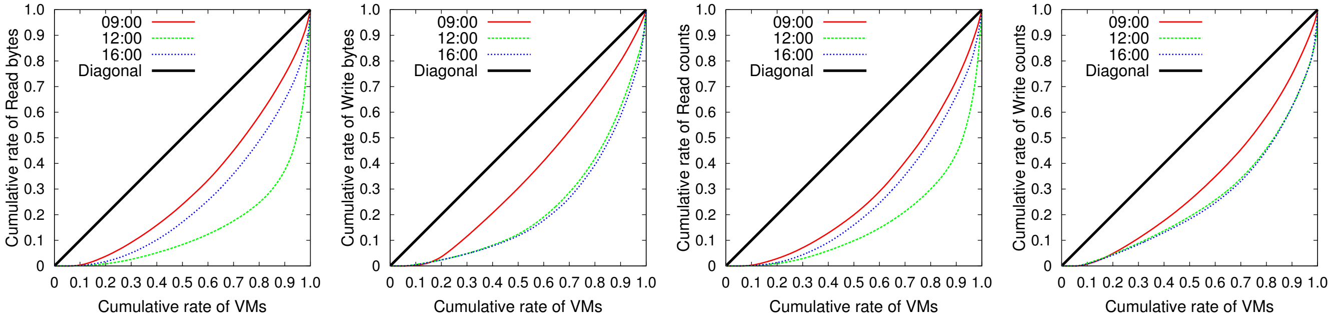
6.1 VM-level storage traffic

In this section, we quantitatively analyze the VM behavior in detail. For the analysis, we basically used the WMI logs in the VDI servers. VM-level traffic on one day (2/22/2016) is illustrated in Figures 10(a) and 10(b). Continuous small write traffic (light gray color) is observed, while burst read traffic (deep black color) occurs. We assume that such small traffic at the VMs would be generated for timestamping logs. Again, we cannot observe the packet payload due to the contracts. Next, the VM-level traffic widely fluctuates and depends both on time and the VMs. For example, massive traffic was intensively generated during the 09:00 time bin due to the update storm. Moreover, we can observe that some VMs have consecutive traffic generated over several hours. Such VMs can affect the performance of an entire VDI system. The write also showed similar characteristics.

We classified the VM-level traffic into that generated by active VMs (connected to users) and that generated by idle VMs (not connected to users) over the four weeks in Figure 11. The traffic of active VMs contains the foreground processes of users. It may affect their QoE. This figure indicates that the active VMs issue less traffic than the idle ones. The active VMs issued 24.5% (read 21.2%, write 39.1%) of total traffic. Thus, the traffic issued by the idle VMs accounted for 75.5% of total traffic. To improve the QoE of active users, it may be important to automatically distinguish and suppress the background traffic on a hypervisor, virtual switch, and switch fabric.

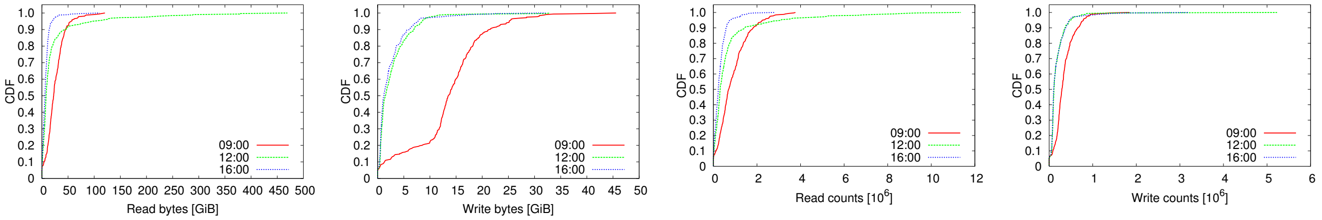
6.2 Idle VM behavior

To reveal why the idle VMs generate large traffic, we investigated the I/O of the individual processes on an idle VM by using Process Monitor [20]. Process Mon-



(a) Read bytes (b) Write bytes (c) Read counts (d) Write counts

Figure 15: Lorenz curves of total workload for each VM



(a) Read bytes (b) Write bytes (c) Read counts (d) Write counts

Figure 16: CDF for distribution of total workload on each VM

itor was only used in this measurement, and a remote CIFS disk was used to avoid overheads on the block storage. We deployed six idle VMs that have no post-installed programs or artificial overheads for about 24 hours. In the measurement, the storm was removed to disperse the update schedule and make the idle VMs behave similarly.

We selected a representative idle VM to introduce the result. In the idle VM, read traffic (93% [13.2 GiB]) was dominant in comparison with write traffic (7% [0.9 GiB]). This read ratio was higher than that in the dataset (Table 2). Figures 12 and 13 summarize the results for several read/write heavy applications. The traffic is mainly generated by the top three applications: anti-virus, system profiling [11], and Windows service. Windows service corresponds to a common Windows process called *svchost.exe*. This process manages multiple Windows daemon processes. The top three applications showed similar characteristics, and our results are also similar to those in previous work [25].

6.3 Traffic distribution across active VMs

In this section, we analyze the distribution of the VM-level traffic across the active VMs. We characterize the VMs that generated dominant traffic (we call them *heavy* VMs) through statistical analysis.

We calculated the Gini coefficients [7] (Figure 14) of traffic for each hourly time bin to reveal the existence of the *heavy* VMs. A low Gini coefficient means that the VM-level traffic is homogeneous, and a high one means a few *heavy* VMs generate the majority of overall traffic. We found the lowest Gini (0.377) from 09:00 through 10:00 because of the update on almost all VMs. On the other hand, the largest Gini (0.709) was in the traffic from 12:00 through 13:00. A major reason for this was the virus *scans* scheduled by some VMs during lunch time, which generates the many reads toward the virtual disks. Lastly, the afternoon has similar Gini coefficients. The results indicate the ordinary usage of VDI.

Through the Gini coefficient, we can also categorize three time-bin patterns: morning, lunch time, and afternoon. We plotted the corresponding Lorenz curve and CDF in Figures 15 and 16, where the three representative time bins are morning (09:00), lunch time (12:00), and afternoon (16:00). We reconfirmed from these results that 1) the traffic in the morning time bin has a narrow deviation across VMs because all VMs generated larger traffic (virus updates) and that 2) the read traffic during lunch has a wider deviation across VMs because about 10% of VMs generated large read traffic. In the Lorenz curve, the write in the morning is lower than in the other time bins. This means the morning has a more uniform distribution than the others. In contrast, the coefficient at lunch time is higher than the others. Thus, lunch time has a more unequal distribution than the others. Although some VMs continuously generated more traffic than the others, the Gini coefficients are smaller than in the previous results (0.89) [9]. Moreover, 1% of active VMs at lunch time generate only 16.6% of traffic. Although there are *heavy*

VMs at lunch time, their impact is not as large as in previous studies [6, 9, 17]. In conclusion, these results differ from those in previous studies [6, 9, 17], and the VM-level traffic is relatively homogeneous across VMs.

7. DISCUSSION

In this paper, we focus on deepening our understanding of the nature of storage traffic by presenting measurements and analyses of enterprise VDI. We believe that the results of our study will be useful for both academics and practitioners. Below, we discuss some implications and opportunities for improvement that can be obtained from our findings.

The implications derived from the FC-based analysis are summarized as follows. First, a short time interval is required to observe the burstiness accurately. During an update storm, many transactions were rapidly generated in a short time period. Ordinary measurement methods cannot catch the precise behavior of the storm. Second, a method for predicting block addresses can help to shorten the response time. During the storm, similar block addresses for each VM were accessed in a short time period. Third, cache and address block management at storage should be enhanced to handle the dominant read traffic. Lastly, the write response time was seriously affected by their cache misses during the storm. In particular, the large transactions, such as 128 KiB, significantly suffered. As a result, the entire VDI performance drastically decreased. The write transaction size for cache hits should be considered to improve the overall VDI performance.

The implications derived from the VM behavior are summarized as follows. First, the active VMs may be affected by the workload generated by the idle VMs. During an update storm, VM-level traffic was homogeneous and the entire response time was largely increased. Giving higher priority to the active VMs can improve their response time. In addition, the periodical traffic at the idle VMs can be reduced by the hypervisor. Second, the anti-virus was a dominant workload in idle VMs. It could be excluded from the active VM workload. It should also be carefully managed to improve disk performance. Finally, the storage traffic across VMs was relatively small. At the largest Gini coefficient, 1% of active VMs generated only 16.6% of traffic. We suggest that the uniform load balancing strategy for all VMs is sufficient and that no specific strategies for *heavy* VMs are required for office VDI.

8. CONCLUSION

In this paper, we focused on understanding the nature of storage traffic by extensively analyzing enterprise virtual desktop infrastructure (VDI) using designated virtual machines (VMs). The storage traffic characteristics were reported from two aspects: fibre channel (FC) analysis and VM behavior. Some of our findings differed from those in related studies, while others enabled us to validate those for small-scale VDI systems and from different perspectives. Our analysis and implications provide a deep and comprehensive understanding of storage traffic characteristics, which will be useful for designing future VDI and improving its performance.

In future work, we intend to analyze the characteristics of user application workloads on the active VMs. We will also investigate the storage traffic characteristics on pool-based VDI.

9. ACKNOWLEDGMENTS

We thank our shepherd Bianca Schroeder and the anonymous reviewers of SYSTOR 2017 for their detailed comments and helpful suggestions. We are grateful to Takeshi Horie, Toshihiro Ozawa, Tomohiro Ishihara, and Junji Ogawa for supporting this work. We also thank Mitsuhiko Ohta, Kenji Wakamiya, Manabu Adachi, Hideo Ichikawa, and Hiroyuki Koyama for gathering data and for their insightful feedback.

10. REFERENCES

[1] Fio. https://github.com/axboe/fio.

[2] IoMeter benchmark. http://www.iometer.org/.

[3] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Data center tcp (dctcp). In *Proceedings of the ACM SIGCOMM 2010*, SIGCOMM '10, pages 63–74, New York, NY, USA, 2010. ACM.

[4] R. Birke, M. Björkqvist, L. Y. Chen, E. Smirni, and T. Engbersen. (Big) data in a virtualized world: Volume, velocity, and variety in cloud datacenters. In *Proceedings of the 12th USENIX Conference on File and Storage Technologies*, FAST'14, pages 177–189, Berkeley, CA, USA, 2014. USENIX Association.

[5] Citrix. Get-BrokerConnectionLog. http://support.citrix.com/static/kc/CTX127254/help/Get-BrokerConnectionLog.html.

[6] I. Drago, M. Mellia, M. M. Munafo, A. Sperotto, R. Sadre, and A. Pras. Inside Dropbox: Understanding personal cloud storage services. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 481–494, New York, NY, USA, 2012. ACM.

[7] C. Gini. Measurement of Inequality of Incomes. *The Economic Journal*, 31(121):124–126, Mar. 1921.

[8] R. Gracia-Tinedo, D. Harnik, D. Naor, D. Sotnikov, S. Toledo, and A. Zuck. Sdgen: Mimicking datasets for content generation in storage benchmarks. In *Proceedings of the 13th USENIX Conference on File and Storage Technologies*, FAST'15, pages 317–330, Berkeley, CA, USA, 2015. USENIX Association.

[9] R. Gracia-Tinedo, Y. Tian, J. Sampé, H. Harkous, J. Lenton, P. García-López, M. Sánchez-Artigas,

and M. Vukolic. Dissecting UbuntuOne: Autopsy of a global-scale personal cloud back-end. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, IMC '15, pages 155–168, New York, NY, USA, 2015. ACM.

[10] WMI Performance. http://wutils.com/wmi/root/cimv2/win32_perfrawdata/#Childs.

[11] Systemwalker. http://www.fujitsu.com/global/products/software/middleware/operation-management/systemwalker/.

[12] Xendesktop. https://www.citrix.com/products/xendesktop/overview.html.

[13] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken. The nature of data center traffic: measurements and analysis. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 202–208, New York, NY, USA, 2009. ACM.

[14] S. Kavalanekar, B. Worthington, Q. Zhang, and V. Sharda. Characterization of storage workload traces from production Windows Servers. In *Workload Characterization, 2008. IISWC 2008. IEEE International Symposium on*, pages 119–128, Sept 2008.

[15] R. Koller and R. Rangaswami. I/O deduplication: Utilizing content similarity to improve I/O performance. *Trans. Storage*, 6(3):13:1–13:26, Sept. 2010.

[16] A. W. Leung, S. Pasupathy, G. Goodson, and E. L. Miller. Measurement and analysis of large-scale network file system workloads. In *USENIX 2008 Annual Technical Conference*, ATC'08, pages 213–226, Berkeley, CA, USA, 2008. USENIX Association.

[17] S. Liu, X. Huang, H. Fu, and G. Yang. Understanding data characteristics and access patterns in a cloud storage system. In *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pages 327–334, May 2013.

[18] L. Mei, G. Xu, W. Yanjun, Z. Chen, and L. Mingshu. Characterizing the spatio-temporal burstiness of storage workloads. In *Proceedings of the 5th International Workshop on Cloud Data and Platforms*, CloudDP '15, pages 2:1–2:6, New York, NY, USA, 2015. ACM.

[19] Microsoft. Hyper-V. https://technet.microsoft.com/library/hh831531.aspx/.

[20] Microsoft. Process Monitor. https://technet.microsoft.com/en-us/sysinternals/processmonitor.aspx.

[21] Microsoft. Windows Management Instrumentation (WMI). https://msdn.microsoft.com/en-us/library/.

[22] D. Narayanan, A. Donnelly, E. Thereska, S. Elnikety, and A. Rowstron. Everest: Scaling down peak loads through I/O off-loading. In *Proceedings of the 8th USENIX Conference on Operating Systems Design and Implementation*, OSDI'08, pages 15–28, Berkeley, CA, USA, 2008. USENIX Association.

[23] I. Narayanan, D. Wang, M. Jeon, B. Sharma, L. Caulfield, A. Sivasubramaniam, B. Cutler, J. Liu, B. Khessib, and K. Vaid. Ssd failures in datacenters: What? when? and why? In *Proceedings of the 9th ACM International on Systems and Storage Conference*, SYSTOR '16, pages 7:1–7:11, New York, NY, USA, 2016. ACM.

[24] B. Seo, S. Kang, J. Choi, J. Cha, Y. Won, and S. Yoon. IO workload characterization revisited: A data-mining approach. *IEEE Trans. Comput.*, 63(12):3026–3038, Dec. 2014.

[25] M. Shamma, D. T. Meyer, J. Wires, M. Ivanova, N. C. Hutchinson, and A. Warfield. Capo: Recapitulating storage for virtual desktops. In *Proceedings of the 9th USENIX Conference on File and Stroage Technologies*, FAST'11, pages 3–3, Berkeley, CA, USA, 2011. USENIX Association.

[26] H. Shim, P. Shilane, and W. Hsu. Characterization of incremental data changes for efficient data protection. In *Proceedings of the 2013 USENIX Conference on Annual Technical Conference*, USENIX ATC'13, pages 157–168, Berkeley, CA, USA, 2013. USENIX Association.

[27] M. Tamura, K. Iizawa, M. Maeda, J. Kato, T. Kumano, Y. Nomura, and T. Ozawa. Distributed object storage toward storage and usage of packet data in a high-speed network. In *The 16th Asia-Pacific Network Operations and Management Symposium, APNOMS 2014*, pages 1–6, New York, NY, USA, Sept 2014. IEEE Communications Society.

[28] VMware. Horizon View. http://www.vmware.com/products/horizon-view.

[29] VMware. Storage I/O control technical overview and considerations for deployment. http://www.vmware.com/files/pdf/techpaper/VMW-vSphere41-SIOC.pdf.

[30] H. Wang, F. Xu, Y. Li, P. Zhang, and D. Jin. Understanding mobile traffic patterns of large scale cellular towers in urban environment. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, IMC '15, pages 225–238, New York, NY, USA, 2015. ACM.

[31] Y. Zhang, P. Qu, Y. Li, H. Wang, and W. Zheng. Cache optimizations of distributed storage for software streaming services. In *Proceedings of the 13th International Conference on Algorithms and Architectures for Parallel Processing - Volume 8285*, ICA3PP 2013, pages 458–467, New York, NY, USA, 2013. Springer-Verlag New York, Inc.