An In-Depth Comparative Analysis of Cloud Block Storage Workloads: Findings and Implications *

Jinhong Li†, Qiuping Wang†, Patrick P. C. Lee†, Chao Shi‡
†The Chinese University of Hong Kong
‡Alibaba Group

Abstract

Cloud block storage systems support diverse types of applications in modern cloud services. Characterizing their I/O activities is critical for guiding better system designs and optimizations. In this paper, we present an in-depth comparative analysis of production cloud block storage workloads through the block-level I/O traces of billions of I/O requests collected from two production systems, Alibaba Cloud and Tencent Cloud Block Storage. We study their characteristics of load intensities, spatial patterns, and temporal patterns. We also compare the cloud block storage workloads with the notable public block-level I/O workloads from the enterprise data centers at Microsoft Research Cambridge, and identify the commonalities and differences of the three sources of traces. To this end, we provide 6 findings through the high-level analysis and 16 findings through the detailed analysis on load intensity, spatial patterns, and temporal patterns. We discuss the implications of our findings on load balancing, cache efficiency, and storage cluster management in cloud block storage systems.

1 Introduction

Traditional desktop and server applications, such as virtual desktops, operating systems, web services, relational databases, and key-value stores, are now moving to the cloud. *Cloud block storage* systems [3, 4, 21, 28, 29, 52, 53] form an infrastructure that allows cloud service providers to manage large-scale physical storage clusters. They also provide virtual disks, referred to as *volumes*, for clients to host various types of applications. To allow performance optimizations and efficient resource provisioning of cloud block storage systems, it is critical to characterize and understand the I/O behaviors of the applications in production environments.

Several field studies have analyzed the I/O behaviors of various architectures via the collection and characterization of block-level I/O traces [1, 16, 19, 20, 33, 54]. In particular, the public block-level I/O traces released by Microsoft Research Cambridge [33] have received wide attention from researchers and practitioners. The traces, which we refer to as *MSRC*, have been extensively analyzed to motivate storage system designs and optimizations, such as I/O scheduling [10, 23, 24, 33], caching [36, 37], erasure-coded storage [12, 53], as well as cloud block storage [21].

However, the MSRC traces, which were collected from enterprise data centers more than a decade ago, may not necessarily reflect the actual I/O behaviors of today's cloud block storage systems. Modern cloud environments often host much more diverse types of applications, some of which feature unique characteristics (e.g., short-lived tasks [32]) that are not commonly found in traditional data center environments. Also, the workloads in MSRC are generally read-dominant [33], while the workloads in cloud environments are often write-dominant due to the heavy use of read caches in cloud applications [25, 46]. Such mismatches

*An earlier version of this article appeared in [22]. In this extended version, we further include the workload traces from Tencent Cloud Block Storage [52] in our analysis. We extend our findings to show the commonalities and differences between the cloud block storage workloads from Alibaba Cloud and Tencent Cloud Block Storage.

motivate the need of collecting and analyzing comprehensive block-level I/O traces from real-world cloud block storage systems in large-scale production.

In this paper, we present an in-depth comparative study on the block-level I/O traces from two production cloud block storage systems. The first set of traces, which we refer to as *AliCloud*, is collected by ourselves from a production cloud block storage system deployed at Alibaba Cloud [22] and covers one-month I/O activities of 1,000 volumes. The second set of traces, which we refer to as *TencentCloud*, is collected from the Tencent Cloud Block Storage by Zhang et al. [52] and covers I/O activities of 4,995 volumes over around nine days. Both sets of traces feature a large data scale, totaling billions of I/O requests and hundreds of terabytes of I/O traffic. We compare the cloud block storage workload characteristics of both AliCloud and TencentCloud traces with those of the MSRC traces, and identify the commonalities and differences of the three sources of traces. To this end, we provide 6 findings through the high-level analysis on the basic I/O characteristics of the traces, and further provide 16 findings through the detailed analysis on the I/O behaviors in terms of the load intensities, spatial patterns, and temporal patterns. We provide insights into load balancing, cache efficiency, and storage cluster management in cloud block storage systems. Note that Zhang et al. [52] mainly use the TencentCloud traces for designing efficient cache allocation schemes in cloud block storage systems, but do not provide an in-depth analysis on the TencentCloud traces. To the best of our knowledge, compared with prior measurement studies on block-level I/O traces [1, 16, 19, 20, 33, 49, 54] (see details in §5), our trace analysis is one of the largest measurement studies on block-level I/O traces reported in the literature. We make the source code of all our analysis scripts available at: **http://adslab.cse.cuhk.edu.hk/software/blockanalysis**.

We highlight some major findings of our trace analysis. From the high-level analysis, small I/O requests dominate in all traces. Both AliCloud and TencentCloud are write-dominant, while MSRC is read-dominant. For load intensities, all traces show similar amounts of I/O traffic, while AliCloud and TencentCloud show more diverse burstiness across volumes and have higher activeness than MSRC. For spatial patterns, all traces show aggregations of reads and writes in small working sets. In particular, TencentCloud shows the highest level of aggregations of reads, implying a more skewed access pattern in reads. All traces also show high fractions of random I/Os and varying patterns in the update coverage across volumes. For temporal patterns, all traces have varying temporal update patterns across volumes and different access tendencies for the written blocks. For example, each written block in AliCloud and TencentCloud is likely to be followed by a write, while that in MSRC is about equally likely to be followed by either a read or a write.

The rest of the paper proceeds as follows. In §2, we present our cloud block storage architecture and its design considerations. In §3, we introduce the traces for our analysis, and present 6 findings via our high-level analysis. In §4, we conduct an in-depth analysis and provide 16 findings on load intensities, spatial patterns, and temporal patterns. We emphasize the commonalities and differences of the findings between AliCloud and TencentCloud. We further discuss the implications of our findings in terms of load balancing, cache efficiency, and storage cluster management in cloud block storage systems. In §5, we review related work. In §6, we conclude the paper.

2 Background and Methodology

We introduce the cloud block storage architecture considered in the paper (§2.1). We further elaborate on how our trace analysis should characterize the I/O activities in response to the design considerations for cloud block storage (§2.2).

2.1 Cloud Block Storage

Figure 1 depicts the architecture of a cloud block storage system considered in the paper. The cloud block storage system serves as a middleware layer that bridges: (i) the virtual disks (referred to as *volumes*) that are perceived by upper-layer applications, and (ii) the storage clusters that provide physical storage space owned by cloud service providers. Each application is allocated with a dedicated volume. It issues read or write

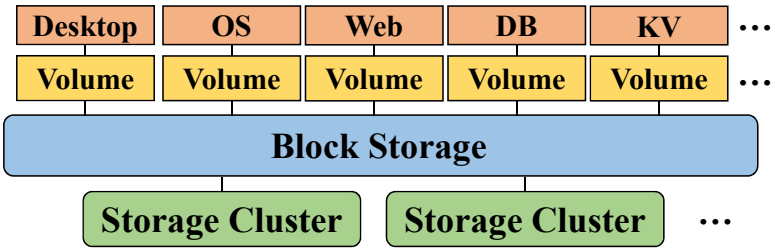


Figure 1: Architecture of a cloud block storage system. It comprises multiple volumes that host a mix of cloud applications (e.g., virtual desktops, operating systems, web services, relational databases, and key-value stores).

requests through the dedicated volume to the storage clusters. Each volume is typically replicated across multiple storage clusters for fault tolerance. For performance and reliability, today's storage cluster are often backed by flash-based solid-state drives (SSDs) instead of hard disk drives (HDDs) [15, 27, 48, 49].

In production, a cloud block storage system may manage diverse types of upper-layer cloud applications (Figure 1). The I/O characteristics of such applications are often largely different, as we show in this paper.

2.2 Analysis Methodology

Cloud block storage systems should maintain quality-of-services guarantees (e.g., low-latency requests and fairness) and efficient resource utilizations (e.g., long device lifetime). We highlight three design considerations for cloud block storage systems, namely *load balancing*, *cache efficiency*, and *storage cluster management*. In the following, we explain how each of the design considerations can be addressed in our trace analysis of I/O activities in cloud block storage.

Load balancing. Maintaining load balancing across storage devices is important for availability and performance. If load imbalance exists, some storage devices may be overloaded by a large number of I/O requests and cannot serve incoming requests in a timely manner, thereby increasing the overall I/O latencies. In addition, the overloading of I/O requests may aggravate flash wearing [48], leading to reduced endurance. Since load balancing addresses the performance differences due to the uneven distribution of I/O traffic, our trace analysis should examine the load intensities of I/O traffic.

Cache efficiency. To speed up I/O performance, storage systems typically cache frequently accessed data based on efficient resource allocation schemes and admission policies [5, 44]. However, the high variations of I/O characteristics may introduce improper cache space allocation and cache management policies, which degrade hit ratios and increase the overall I/O latencies. To investigate how the caching design can leverage workload characteristics, our trace analysis should address the spatial and temporal aggregations of I/O traffic.

Storage cluster management. Enterprise storage clusters increasingly move to flash-based storage, which is sensitive to varying workload patterns in both performance and endurance. In particular, the update patterns can determine the effectiveness of garbage collection and wear-leveling in flash [17]. Storage cluster management should address the variations of workload patterns, so as to maintain high performance and endurance of the underlying flash devices. Thus, our trace analysis should focus on the spatial and temporal patterns for update requests. Also, as small and random I/Os can degrade the performance and endurance of flash storage [31], our trace analysis should also examine the randomness of I/Os.

3 Traces

We describe the three sets of traces used in our analysis and state the limitations of our trace analysis (§3.1). We then present a high-level analysis on the basic statistics as well as the commonalities and differences on all three traces (§3.2).

3.1 Trace Overview

Our trace analysis is based on three sets of block-level I/O traces collected from different production environments. For brevity, we refer to the traces as *AliCloud*, *TencentCloud*, and *MSRC* in short in the following discussion.

AliCloud. The traces were collected by ourselves from a cloud block storage system deployed at Alibaba Cloud over a one-month period in January 2020. The traces are now released at [2]. They comprise block-level I/O requests collected from 1,000 volumes, each of which has a raw capacity from 40 GiB to 5,000 GiB. The workloads span diverse types of cloud applications (§2.1). Each collected I/O request specifies the volume number, request type, request offset, request size, and timestamp (in units of microseconds).

TencentCloud. The traces were collected from the cloud block storage system at Tencent Cloud Block Storage [52] from 12:00 AM on October 1, 2018 to 1:00 AM on October 10, 2018 in the GMT+8 time zone (i.e., 9.04 days in total); note that the requests are missing between 1:00 AM and 2:00 AM on October 8, 2018. The traces can be downloaded from the SNIA IOTTA repository [40]. They comprise block-level I/O requests collected from 4,995 volumes. The workloads are based on a mixture of cloud applications, including applications dominated by random accesses and applications with large amount of I/O activity [52]. Each collected I/O request contains the volume number, request type, request offset, request size, and timestamp as in the AliCloud traces, except that the timestamp in the TencentCloud traces is in units of seconds. However, the TencentCloud traces do not provide the raw capacities of individual volumes.

MSRC [33]. The traces were collected by Microsoft Research Cambridge from a data center of Microsoft Windows servers over a one-week period in February 2007 and can be downloaded from the SNIA IOTTA repository [30]. They comprise block-level I/O requests from 36 volumes over 179 disks in 13 servers. The workloads span a variety of applications, including home directories, project directories, web services, source control, media services, etc. Each collected I/O request includes the volume number, request type, request offset, request size, and timestamp as in the AliCloud and TencentCloud traces; it also includes the response time of the request. Both the timestamp and the response time are specified in units of 100 ns, based on the Windows Filetime timestamp format used by Microsoft Windows servers.

Limitations of our trace analysis. Our trace analysis has several limitations. First, both the AliCloud and TencentCloud traces do not record the response times of the I/O requests as in MSRC, so we cannot conduct latency analysis on I/O requests in actual deployment. In particular, the timestamp field in the TencentCloud traces is in units of seconds, so it is difficult to conduct fine-grained analysis on the inter-arrival times of requests in the TencentCloud traces. Also, both traces do not indicate the specific applications running atop individual volumes, so we cannot investigate the relationship between specific application workloads and their I/O patterns. Furthermore, all three traces do not include the caching policies, cache hit/miss ratios (i.e., we cannot study the impact of caching in actual deployment) and the age or usage of each volume (i.e., we cannot study the relationships between I/O patterns and volume ages). Finally, all three traces do not capture the information of physical storage devices (e.g., data placement and failure statistics), so we cannot study the performance and reliability correlations in physical storage.

3.2 High-level Comparative Analysis

We now present a high-level comparative analysis on AliCloud, TencentCloud, and MSRC by collectively analyzing the I/O requests of all volumes in each set of traces and presenting the overall basic statistics. Our goal is to examine the basic properties of the cloud block storage workloads in AliCloud and TencentCloud as well as the classical enterprise data center workloads in MSRC. To this end, we identify the commonalities and differences of all three traces.

Table 1 summarizes different categories of basic statistics in AliCloud, TencentCloud, and MSRC, including: (i) the numbers of reads and writes, (ii) the total amounts of data read, written, and updated, as well as (iii) the *working set sizes (WSSs)* of reads, writes, and updates (an *update* request refers to a write

	AliCloud	TencentCloud	MSRC
#Volumes	1,000	4,995	36
Duration (days)	31	9.04	7
#Reads (millions)	5,058.6	10,030.2	304.9
#Writes (millions)	15,174.4	23,592.0	128.9
Read Traffic (TiB)	161.6	282.3	9.04
Write Traffic (TiB)	455.5	837.2	2.39
Update Traffic (TiB)	429.2	804.2	2.01
Total WSS (TiB)	29.5	38.7	2.87
Read WSS (TiB)	10.1	14.6	2.82
Write WSS (TiB)	26.3	33.0	0.38
Update WSS (TiB)	18.6	21.2	0.17

Table 1: Basic statistics of AliCloud, TencentCloud, and MSRC.

	AliCloud	TencentCloud	MSRC
Read WSS over total WSS (A.1)	34.3%	37.6%	98.4%
75th percentiles of read/write sizes (A.2)	12 KiB/16 KiB	32 KiB/12 KiB	64 KiB/20 KiB
Volumes with short active periods (A.3)	15.7%	1.7%	0.0%
Write-dominant volumes (A.4)	91.5%	92.3%	52.8%
Higher fractions of WSS in larger volumes (A.5)	Yes	-	-
Larger requests in larger volumes (A.6)	Yes	-	-

Table 2: Summary of the key properties of AliCloud, TencentCloud, and MSRC in Findings A.1-A.6.

request to a block that has been written at least once). We define the total read, write, and update WSSs as the numbers of unique logical addresses being accessed (i.e., read, written, and updated, respectively) by all I/O requests of the traces multiplied by the block size 4 KiB. Table 2 further summarizes the key properties of all three traces observed in our high-level analysis.

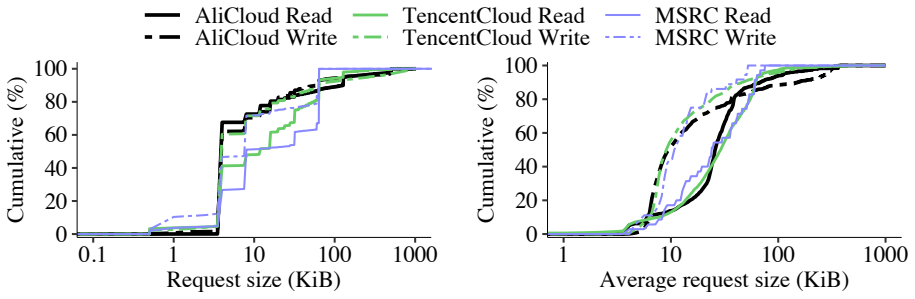
In terms of scale, both AliCloud and TencentCloud have much larger scale than MSRC in various aspects, including the number of volumes, the trace duration, the total number of I/O requests, and the total I/O traffic size. For example, AliCloud contains 20.2 billion I/O requests, 46.6× the total number of I/O requests in MSRC. It also has more volumes (27.8×), a larger total I/O traffic size (54.1×), and a larger WSS (10.3×), compared with those in MSRC. TencentCloud has an even larger scale than AliCloud in terms of the number of volumes (5.0×), the total number of I/O requests (1.7×), and the total I/O traffic size (1.8×), except that its duration only lasts for 9.04 days. In the following, we elaborate the I/O characteristics of the three traces.

Finding A.1: *Reads span a small proportion of working sets in both AliCloud and TencentCloud.*

Referring to Table 1, reads in AliCloud and TencentCloud only occupy 34.3% and 37.6% of the total WSS, respectively, while reads in MSRC occupy a much larger proportion (98.4%) of the total WSS. On the other hand, writes in AliCloud and TencentCloud occupy 89.4% and 85.2% of the total WSS, respectively. The results indicate that a substantial amount of written data is never read again in both AliCloud and TencentCloud. One possible reason is that some applications tend to only write data but rarely read data (e.g., backups or journaling), although we cannot identify the specific applications running on individual volumes (§3.1).

Finding A.2: *Small-size I/Os dominate in all AliCloud, TencentCloud, and MSRC.*

Figure 2(a) shows the cumulative distributions of request sizes across all I/O requests in all three traces. We see that all traces feature small-size I/O requests (less than 100 KiB). Specifically, in AliCloud, 75% of reads and writes are no larger than 12 KiB and 16 KiB, respectively, while in TencentCloud, 75% of reads



(a) Cumulative distribution of request sizes across all requests

(b) Cumulative distribution of average request sizes across all volumes

(c) Cumulative distribution of median request sizes across all volumes

(d) Cumulative distribution of 75th percentiles of request sizes across all volumes

Figure 2: Finding A.2: Cumulative distributions of I/O request sizes.

and writes are no larger than 32 KiB and 12 KiB, respectively. In MSRC, 75% of reads and writes are no larger than 64 KiB and 20 KiB, respectively.

The dominance of small-size I/O requests also holds in individual volumes. We compute the average request size for each volume. Figure 2(b) shows the cumulative distributions of the average request sizes of all volumes in all three traces. We see that 75% of the average read and write sizes in AliCloud are less than 37.0 KiB and 26.8 KiB, respectively, while 75% of the average read and write sizes in TencentCloud are less than 49.8 KiB and 19.0 KiB, respectively. For MSRC, 75% of the average read and write sizes are less than 48.4 KiB and 16.4 KiB, respectively. Small I/Os are also commonly found in enterprise and desktop file system workloads [1, 34].

To ensure that our observations are not biased by outliers, we further measure the cumulative distributions of the medians and 75th percentiles of request sizes across all volumes, as shown in Figures 2(c) and 2(d), respectively. In Figure 2(c), we see that the median read and write sizes in 75% of volumes in all three traces are no more than 64 KiB and 4 KiB, respectively. Similarly, in Figure 2(d), we see that the 75th percentiles of read and write sizes in 75% of volumes in all three traces are no more than 64 KiB and 12 KiB, respectively. The results indicate that small-size I/O requests dominate even if we consider medians and 75th percentiles of (instead of average) request sizes.

Finding A.3: *A non-negligible fraction of volumes in AliCloud are active in short time periods, but it is not the case in TencentCloud and MSRC.*

We study the activeness of individual volumes. Here, we measure the number of active days for each volume, in which a volume is said to be active in a day if it receives at least one I/O request (i.e., up to 31, 9, and 7 active days in AliCloud, TencentCloud, and MSRC, respectively). Figure 3 depicts the cumulative distributions of numbers of active days across all volumes in all three traces. In AliCloud, 15.7% of volumes (i.e., 157 volumes) are active for only one day. We find that 147 out of the 157 volumes are active in only four hours, and the total WSS and I/O traffic of the 157 volumes account for only 1.4% and 0.07% of all 1,000 volumes, respectively. One possible reason for the short active periods in such volumes in AliCloud is

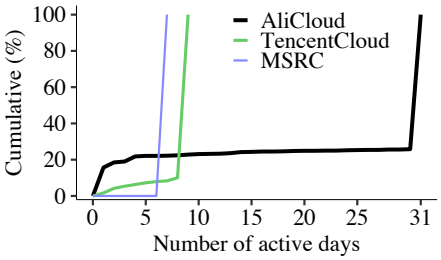


Figure 3: Finding A.3: Cumulative distributions of numbers of active days across all volumes.

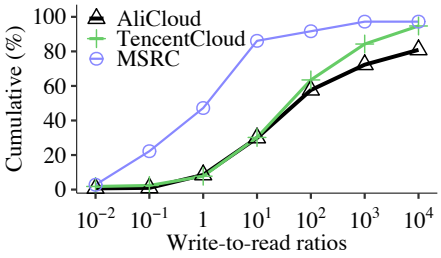
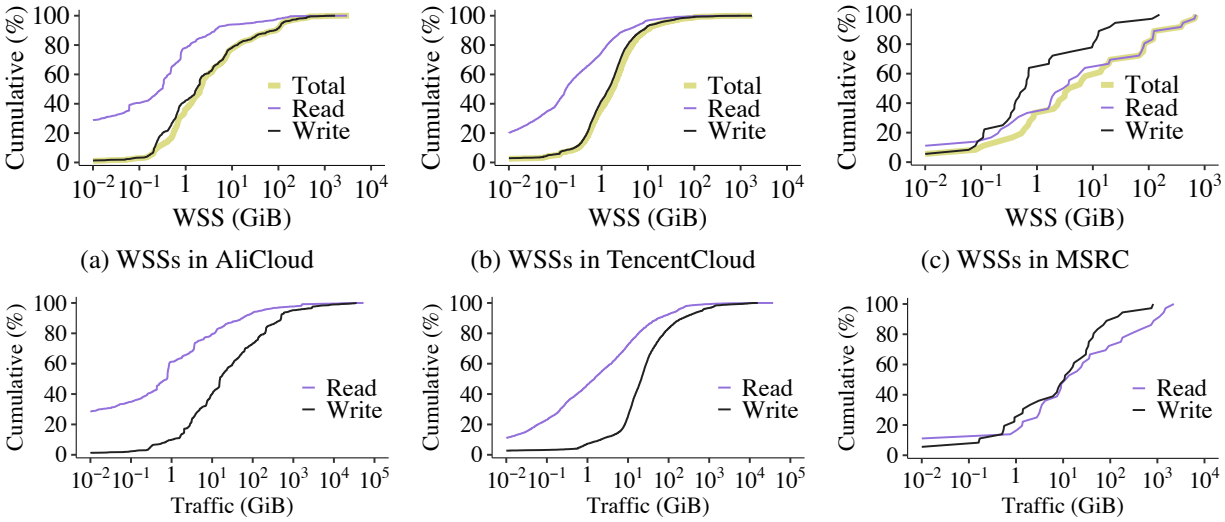


Figure 4: Finding A.4: Cumulative distributions of write-to-read ratios across all volumes.



(a) WSSs in AliCloud (b) WSSs in TencentCloud (c) WSSs in MSRC

(d) I/O traffic sizes in AliCloud (e) I/O traffic sizes in TencentCloud (f) I/O traffic sizes in MSRC

Figure 5: Finding A.4: Cumulative distributions of WSSs (figures (a)-(c)) and I/O traffic sizes (figures (d)-(f)) across all volumes.

the presence of short-lived tasks in cloud applications [32]. On the other hand, in TencentCloud, only 1.7% of volumes are active for only one day, while 90.1% of volumes are active for all nine days in the entire trace duration. Also, all volumes in MSRC are active for all seven days in the entire trace duration.

Finding A.4: *Most volumes in AliCloud and TencentCloud are write-dominant.*

Referring to Table 1, the overall write-to-read ratio (i.e., the ratio between the number of writes and the number of reads) in AliCloud is 3:1, and that in TencentCloud is 2.35:1. However, the write-to-read ratio in MSRC is 0.42:1 only. We further analyze the write-to-read ratios on a per-volume basis. Figure 4 shows the cumulative distributions of write-to-read ratios across all volumes in all three traces. In AliCloud and TencentCloud, 91.5% (i.e., 915 out of 1,000) and 92.3% (i.e., 4,608 out of 4,995) of the volumes are write-dominant (i.e., the write-to-read ratios are larger than 1). Also, 42.4% and 36.5% of the volumes in AliCloud and TencentCloud even have very high write-to-read ratios that are larger than 100, respectively. On the other hand, MSRC has an opposite pattern, in which only 52.8% (19 out of 36) of volumes are write-dominant. Note that prior work [37] also shows the existence of write-dominant workloads in MSRC, especially in files such as mail boxes, search indexes, registry files, and file system metadata files.

Figure 5 further analyzes the cumulative distributions of WSSs and I/O traffic sizes across all volumes in all three traces. Figures 5(a)-5(c) show the cumulative distributions of the total WSSs, read WSSs, and write WSSs across all volumes. The write WSSs of both AliCloud and TencentCloud are significantly larger than

the read WSSs and are close to the total WSSs (Figures 5(a) and 5(b)). This implies that the total WSSs of both traces are mainly determined by writes. In contrast, the read WSSs of MSRC are close to the total WSSs (Figure 5(c)). We also make similar observations in the I/O traffic sizes, in which AliCloud and TencentCloud are write-dominant (Figures 5(d)-5(f)). One possible reason of the write dominance in both AliCloud and TencentCloud is the wide use of application-level read caches in cloud storage, in which reads are mostly absorbed in the application layer without being issued to the storage layer [46].

Finding A.5: *In AliCloud, larger volumes tend to have larger percentages of the total WSS over the raw capacity.*

We examine the relationship between the total WSS and the raw capacity of a volume. Our analysis here focuses on the AliCloud traces, which provide the capacity information of individual volumes. Specifically, we divide the 1,000 volumes in AliCloud into four groups by volume capacities, including 40-49 GiB, 50-99 GiB, 100-199 GiB, and 200-5,000 GiB (note that each volume capacity is represented as an integer GiB). They account for 444, 179, 170, and 207 volumes, respectively. For each volume, we calculate the WSS-to-capacity percentage (i.e., the percentage of the total WSS over the raw capacity of the volume). We then plot the cumulative distributions of the WSS-to-capacity percentages for the four groups.

Figure 6 shows the results. For small volumes, the WSS-to-capacity percentages tend to be low (i.e., the usage of disk space is low). Specifically, 80% of the volumes in the 40-49 GiB, 50-99 GiB, and 100-199 GiB groups have WSS-to-capacity percentages of less than 7.3%, 16.1%, and 9.7%, respectively. On the other hand, in the 200-5,000 GiB group, half of the volumes have a WSS-to-capacity percentage of more than 16.7%, and 35.7% of the volumes have WSS-to-capacity percentages of more than 50%. In particular, the largest volume (i.e., with a raw capacity of 5,000 GiB) has a WSS-to-capacity percentage of 66.9%.

Finding A.6: *In AliCloud, larger volumes tend to have larger write request sizes, but have similar read request sizes compared to smaller volumes.*

We examine the average request sizes of individual volumes with respect to their raw capacities. We again divide the volumes by raw capacities into four groups (i.e., 40-49 GiB, 50-99 GiB, 100-199 GiB, and 200-5,000 GiB) as above.

Figure 7 shows the results. Overall, the average request sizes in the 200-5,000 GiB group are larger than in the other three groups with smaller raw capacities. Specifically, the median of average request sizes in the 200-5,000 GiB group is 36.2 KiB, while the medians of average request sizes in the 40-49 GiB, 50-99 GiB, and 100-199 GiB groups are 11.9 KiB, 8.1 KiB, and 9.5 KiB, respectively. A possible reason is that larger volumes are often related to the workloads with the larger-size sequential data accesses.

We further examine the average read and write request sizes of individual volumes to understand where the larger requests in larger volumes come from. Figure 8 shows the results. We find that the average read sizes are close in four groups; the medians of average read request sizes in the 40-49 GiB, 50-99 GiB, 100-199 GiB, and 200-5,000 GiB groups are 29.3 KiB, 23.0 KiB, 24.9 KiB, and 24.4 KiB, respectively (Figure 8(a)). In contrast, the median of average write request sizes in the 200-5000 GiB group is 36.5 KiB, while the medians of average write request sizes in the 40-49 GiB, 50-99 GiB, and 100-199 GiB groups are 9.1 KiB, 7.0 KiB, and 9.2 KiB, respectively (Figure 8(b)).

Summary. AliCloud, TencentCloud, and MSRC have some common aspects, such as the dominance of small-size I/O requests, yet they also have many differences. In particular, both AliCloud and TencentCloud are write-dominant, while MSRC is read-dominant. A unique aspect for AliCloud, as opposed to TencentCloud and MSRC, is that it has a non-negligible fraction of volumes with short active periods. Also, large volumes in AliCloud tend to have larger WSS-to-capacity percentages and larger request sizes.

4 Detailed Analysis

In this section, we conduct an in-depth comparative analysis on AliCloud, TencentCloud, and MSRC in three aspects: load intensities, spatial patterns, and temporal patterns. We report 16 findings from our analysis.

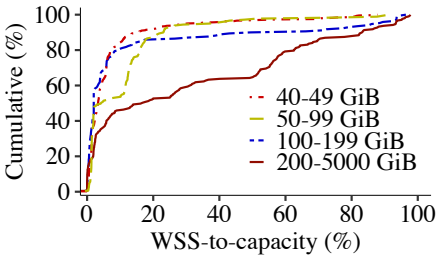


Figure 6: Finding A.5: Cumulative distributions of WSS-to-capacity percentages across all volumes of different groups of raw capacities in AliCloud.

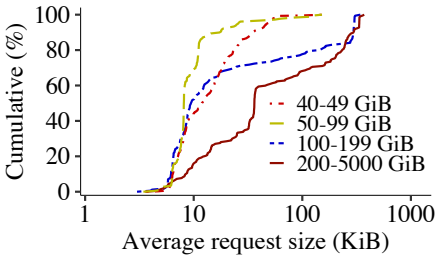
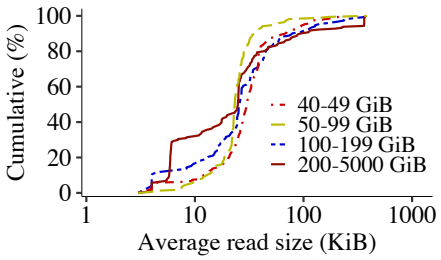
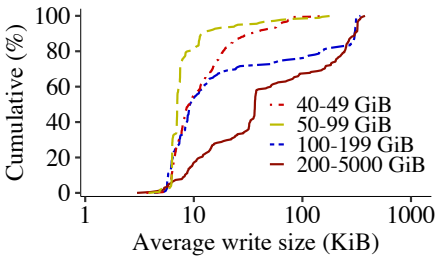


Figure 7: Finding A.6: Cumulative distributions of average request sizes across all volumes of different groups of raw capacities in AliCloud.



(a) Average read request sizes



(b) Average write request sizes

Figure 8: Finding A.6: Cumulative distributions of average read and write request sizes across all volumes of different groups of raw capacities in AliCloud.

Table 3 summarizes and compares the key properties of all three traces observed in our detailed analysis.

4.1 Load Intensities

We study the characteristics of load intensities in the volumes of AliCloud, TencentCloud, and MSRC through a number of metrics. Specifically, we examine the average and peak load intensities [33] and the distribution of inter-arrival times of requests [42]. We also examine the activeness of volumes through the number of active volumes [33] and the active period of each volume.

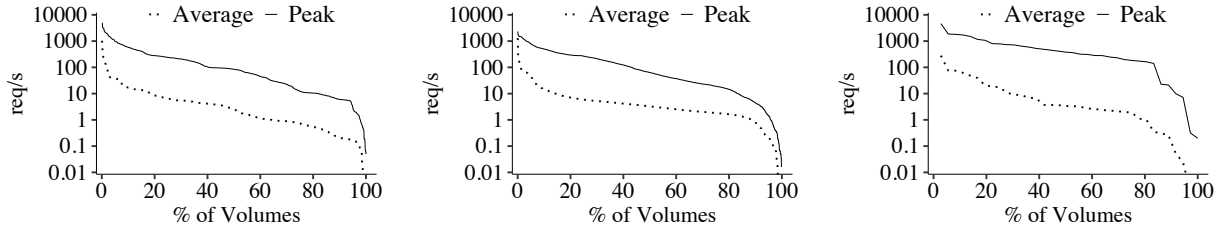
Finding B.1: *AliCloud, TencentCloud, and MSRC have similar average load intensities of volumes, while the peak load intensities of AliCloud and TencentCloud are generally lower than that of MSRC.*

We measure the load intensities of individual volumes, in terms of the number of requests per second (req/s), in two aspects. We first measure the *average intensity* of a volume, defined as the total number of requests divided by the time elapsed between the first and last requests of the volume. Note that for TencentCloud, if the missing hour of requests lies between the first and last requests (§3.1), we subtract the elapsed time by one hour. We also measure the *peak intensity* of a volume, in which we divide the whole duration of requests of the volume into one-minute intervals and find the peak intensity as the maximum number of requests (per second) across all intervals; we use one-minute intervals instead of one-second intervals since one-minute intervals are long enough to accumulate sufficient bursty requests.

Figure 9 shows the average and peak intensities of volumes in AliCloud, TencentCloud, and MSRC, sorted by the average intensities of volumes in descending order. We observe similar trends of average intensities in all three traces, but different patterns of peak intensities in TencentCloud. In AliCloud, TencentCloud, and MSRC, only 1.90%, 1.16%, and 2.78% of volumes have average intensities above 100 req/s, and the percentages of volumes with average intensities lower than 10 req/s are 81.6%, 85.7%, and 72.2%, respectively. Furthermore, their medians of average intensities are 2.55 req/s, 3.27 req/s, and 3.36 req/s, respectively. A possible reason of having similar average intensities in all three traces is that more applications are moving to

	Property	AliCloud	TencentCloud	MSRC
Load intensity	**Average intensities (B.1)**	Similar		
	Peak intensities (B.1)	Low	Low	High
	Burstiness in volumes (B.2)	Yes	Yes	Yes
	Diversity of burstiness (B.3)	High	High	Low
	Inter-arrival times of requests (B.4)	Short	-	Short
	Activeness (B.5)	High	Highest	Low
	Activeness dominated by writes (B.6)	Yes	Yes	Yes
	Activeness of reads (B.7)	Low	Highest	High
	I/O traffic in daytime (B.8)	52.0%	49.6%	23.3%
Spatial patterns	**Fractions of random I/Os (B.9)**	High	Highest	Low
	Spatial aggregations of reads (B.10 and B.11)	High	Highest	High
	Spatial aggregations of writes (B.10 and B.11)	High	High	Low
	Update coverages (B.12)	High	High	Low
Temporal patterns	**Read-after-write (RAW) times (B.13)**	Large	Small	Large
	Write-after-write (WAW) times (B.13)	Large	Small	Small
	More WAW requests than RAW requests (B.13)	Yes	Yes	No
	Read-after-read (RAR) times (B.14)	Large	Small	Large
	Write-after-read (WAR) times (B.14)	Large	Small	Large
	More RAR requests than WAR requests (B.14)	Yes	Yes	Yes
	Varying update intervals (B.15)	Yes	Yes	Yes
	Miss ratios (B.16)	High	Low	High

Table 3: Summary of the key properties of AliCloud, TencentCloud, and MSRC in Findings B.1-B.16.



(a) AliCloud (b) TencentCloud (c) MSRC

Figure 9: Finding B.1: Average and peak intensities of volumes. Note that we sort the average and peak intensities of volumes in descending order in their respective curves to make the distribution of each type of intensities clearly shown, so the curves are different from the ones in our conference version [22].

the cloud [21], so the average intensities are similar in both cloud and traditional data center environments. However, as for the peak intensities, their 90th percentiles of peak intensities are 578.7 req/s, 498.9 req/s, and 1,612.1 req/s, respectively, in which the peak intensity of MSRC is much higher than those of other two traces.

Finding B.2: *AliCloud, TencentCloud, and MSRC have high burstiness in a non-negligible fraction of volumes, but their overall burstiness is mild.*

We examine the burstiness of all three traces by measuring the *burstiness ratio* of a volume, defined as the ratio between the peak intensity and the average intensity of the volume. Figure 10 shows the cumulative distributions of burstiness ratios across all volumes in AliCloud, TencentCloud, and MSRC. We see that a non-negligible fraction of volumes (20.7%, 10.9%, and 38.9% in AliCloud, TencentCloud, and MSRC,

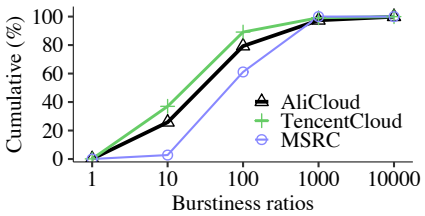


Figure 10: Findings B.2-B.3: Cumulative distribution of burstiness ratios of volumes.

Traces	AliCloud	TencentCloud	MSRC
Peak intensity (req/s)	15,965.8	70,910.1	5,296.8
Average intensity (req/s)	7,554.1	43,036.0	717.2
Burstiness ratio	2.11	1.65	7.39

Table 4: Finding B.2: Overall peak and average intensities as well as burstiness ratios.

respectively) have burstiness ratios higher than 100. Also, 74.2%, 63.0%, and 97.2% of the volumes in AliCloud, TencentCloud, and MSRC have burstiness ratios higher than 10, respectively. This implies that burstiness is common, and such bursty volumes can observe load imbalance at some time. Note that prior work [34] also shows that burstiness exists across various types of applications, such as enterprise systems, desktops, consumer electronics, web servers, and file systems. On the other hand, if we examine overall burstiness level by aggregating all volumes of the whole traces, the burstiness ratios are mild, with 2.11 in AliCloud, 1.65 in TencentCloud, and 7.39 in MSRC (see Table 4). Compared with both AliCloud and MSRC, TencentCloud has a lower percentage of volumes with high burstiness ratios, as well as a lower overall burstiness. This shows that the burstiness level is mild from the whole-system's perspective, but is significant for some of the volumes.

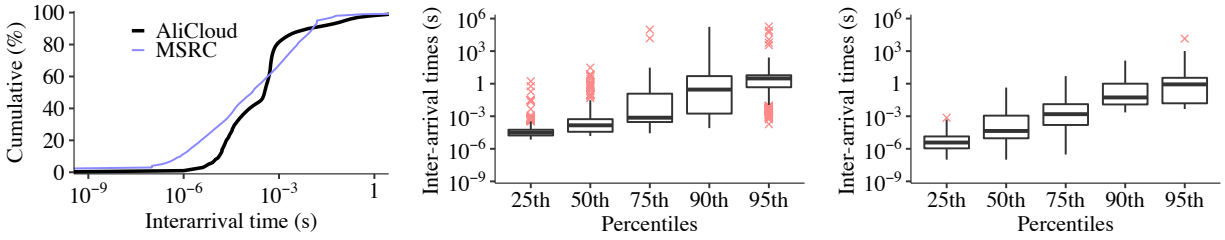
Finding B.3: *AliCloud and TencentCloud have more diverse burstiness across volumes than MSRC.*

The volumes in both AliCloud and TencentCloud span a wider range of burstiness than those in MSRC. Referring to Figure 10, for the volumes with low burstiness, 25.8% and 37.0% of volumes in AliCloud and TencentCloud have burstiness ratios less than 10, while the corresponding percentage is only 2.78% in MSRC. On the other hand, for the volumes with high burstiness, 2.60% and 0.80% of volumes in AliCloud and TencentCloud have burstiness ratios larger than 1,000, respectively, while there are no such volumes in MSRC. The higher diversities of burstiness in AliCloud and TencentCloud suggest larger variations in workload characteristics among different volumes in cloud block storage.

Finding B.4: *Both AliCloud and MSRC have high short-term burstiness from the perspective of inter-arrival times of requests.*

We measure the inter-arrival times of I/O requests (i.e., the elapsed time between two adjacent requests) for each volume. We first examine the cumulative distributions of inter-arrival times across all volumes. Figure 11(a) shows the results. Note that we do not consider TencentCloud, as its timestamps are in units of seconds and we cannot accurately measure the inter-arrival times at finer-grained granularities (e.g., at the microsecond level). We find that most of the inter-arrival times are smaller than one second. For example, in AliCloud and MSRC, 50% of the inter-arrival times are smaller than 351 μs and 142 μs, respectively, and 99% of the inter-arrival times are smaller than 3,140 ms and 484 ms, respectively. The results indicate that short inter-arrival times are common in both traces.

We also consider five groups of percentiles of inter-arrival times for each volume, including the 25th, 50th, 75th, 90th, and 95th percentiles. We represent each group of percentile values of all volumes by boxplots. Figures 11(b) and 11(c) show the results of both AliCloud and MSRC, respectively. Both AliCloud and MSRC traces have a high number of bursty requests, as indicated by large fractions of short inter-arrival times in the volumes. In particular, the medians of the groups of 25th, 50th, and 75th percentiles are lower than 1.3 ms, or equivalently over 700 req/s (i.e., 31 μs, 145 μs, and 735 μs in AliCloud, and 3.5 μs, 30.5 μs, and 1.3 ms in MSRC, respectively). Also, the volumes in AliCloud have much higher inter-arrival times of requests than those in MSRC. For example, half of the volumes in AliCloud have 25th percentiles higher than 31 μs (Figure 11(a)), while half of the volumes in MSRC have 25th percentiles higher than 3.5 μs (Figure 11(b)). Note that prior work [19] also identifies the existence of short inter-arrival times (e.g., a few



(a) Cumulative distributions (b) AliCloud (c) MSRC

Figure 11: Finding B.4: Inter-arrival times of requests. Figure (a) shows the cumulative distributions of inter-arrival times across all volumes. In figures (b) and (c), each boxplot represents the distribution of all the values collected in each volume according to the corresponding percentile.

milliseconds) in server workloads.

Finding B.5: *Most of the volumes in AliCloud, TencentCloud, and MSRC are active throughout the trace periods, while AliCloud and TencentCloud are more active than MSRC. TencentCloud has the highest activeness among all three traces, from both the perspectives of active volumes and active time periods.*

Recall from Section 3.2 that we examine the activeness of volumes of all three traces on a per-day basis. We now revisit the activeness of volumes of all three traces in a more fine-grained manner. Specifically, we divide the traces into 10-minute intervals. We say that a volume is *active* in an interval if it has at least one request in the interval. We also say that a volume is *read-active* and *write-active* in an interval if it has at least one read request and one write request in the interval, respectively.

Figure 12 depicts the numbers of active, read-active, and write-active volumes throughout the trace periods in AliCloud, TencentCloud, and MSRC (recall that they have 1,000, 4,995, and 36 volumes, respectively). We find that the percentages of active volumes throughout the trace duration are always larger than 73.1%, 88.2%, and 59.4% in AliCloud, TencentCloud, and MSRC, respectively. In particular, TencentCloud has the highest fraction of active volumes throughout the trace periods. Also, the numbers of active volumes in AliCloud and TencentCloud have more stable trends compared with that in MSRC. Furthermore, the number of read-active volumes in AliCloud shows diurnal patterns by often having less than 200 read-active volumes at night and more than 300 read-active volumes at daytime, which is similar to the patterns found in object storage systems [8], enterprise virtual desktops [20], and key-value stores [7, 11, 51].

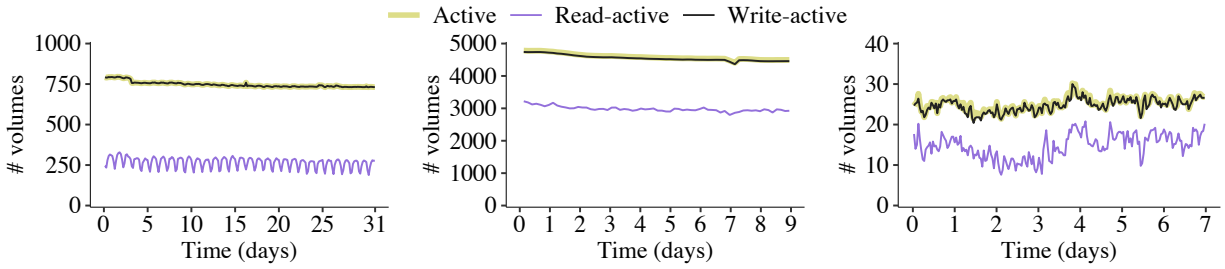
We also measure the active time period of each volume, based on the number of 10-minute intervals in which the volume is active. Figure 13 depicts the cumulative percentages of active time periods across all volumes in all three traces. More than 72.2%, 88.2%, and 55.6% of the volumes are active during 95% of the whole trace periods in AliCloud, TencentCloud, and MSRC, respectively. This indicates that most of the volumes in all three traces have high activeness throughout the whole trace periods, and AliCloud and TencentCloud have higher activeness in general than MSRC. Also, in terms of the active time in each volume, the activeness is the highest in the TencentCloud volumes.

Finding B.6: *Writes are the dominant factor in determining the activeness in AliCloud, TencentCloud, and MSRC.*

Referring to both Figures 12 and 13, the curves of "Active" and "Write-active" nearly overlap with each other in all three traces. It suggests that the activeness of all three traces (in terms of the number of active volumes and the active time period of a volume) is mainly determined by the presence of writes. Thus, load balancing on writes is important for the volumes in cloud block storage.

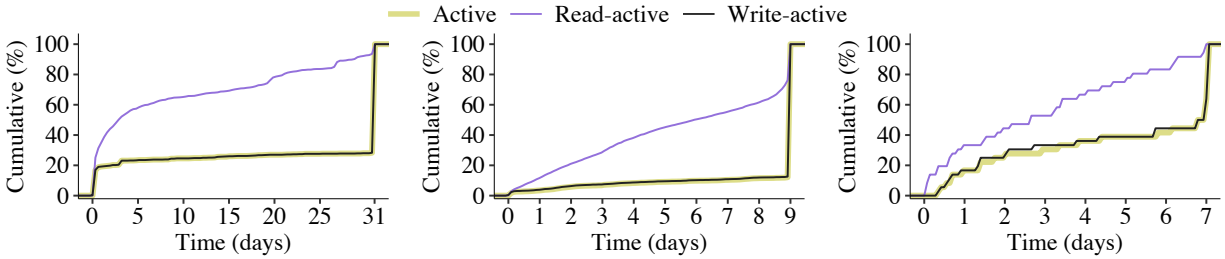
Finding B.7: *Removing write requests shows drastic decreases in activeness in AliCloud, TencentCloud, and MSRC. AliCloud is less read-active than MSRC, and TencentCloud is the most read-active among all three traces.*

If we remove write requests and consider only the read-active volumes, Figure 12 shows that the



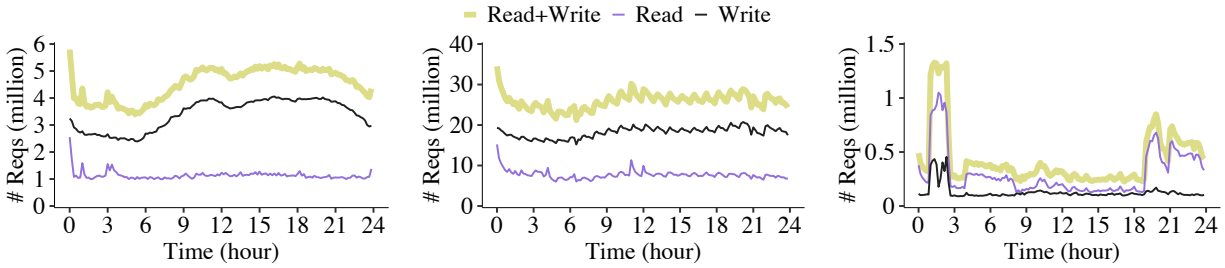
(a) AliCloud (b) TencentCloud (c) MSRC

Figure 12: Findings B.5-B.7: Numbers of active, read-active, and write-active volumes. Note that the "Active" and "Write-active" curves almost overlap with each other.



(a) AliCloud (b) TencentCloud (c) MSRC

Figure 13: Findings B.5-B.7: Cumulative distributions of active time periods measured in 10-minute intervals across all volumes. Note that the "Active" and "Write-active" curves almost overlap with each other.



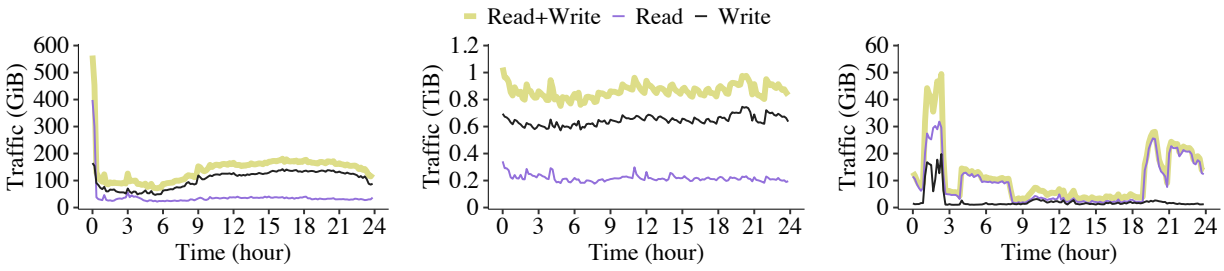
(a) AliCloud (b) TencentCloud (c) MSRC

Figure 14: Finding B.8: Average number of requests in 10-minute intervals each day from 12:00 AM to 11:59 PM.

number of active volumes decreases drastically. In particular, the number of active volumes reduces by 58.6-74.2% in AliCloud (Figure 12(a)), 32.7-37.5% in TencentCloud (Figure 12(b)), and 24.6-65.8% in MSRC (Figure 12(c)).

Figure 13 further shows that the volumes in all three traces have low read-active time, which is consistent with results of prior work [33]. In AliCloud, TencentCloud, and MSRC, half of the volumes have less than 2.83 days, 5.92 days, and 2.61 days of read-active time after removing writes, corresponding to 9.1%, 65.4%, and 37.3% of their whole trace durations, respectively (Figures 13(a)-13(c)). In terms of the percentage of volumes that have long read-active time, we find that 7.6%, 38.4%, 16.7% of the volumes can reach more than 30 days, 8 days, and 6 days of read-active time in AliCloud, TencentCloud, and MSRC, respectively (recall that their trace durations are 31, 9.04, and 7 days, respectively). This suggests that AliCloud is less read-active than MSRC, while TencentCloud is the most read-active among all three traces.

Finding B.8: *I/O traffic in both AliCloud and TencentCloud is almost evenly spread across daytime and nighttime. AliCloud and MSRC have large amounts of read traffic near midnight.*



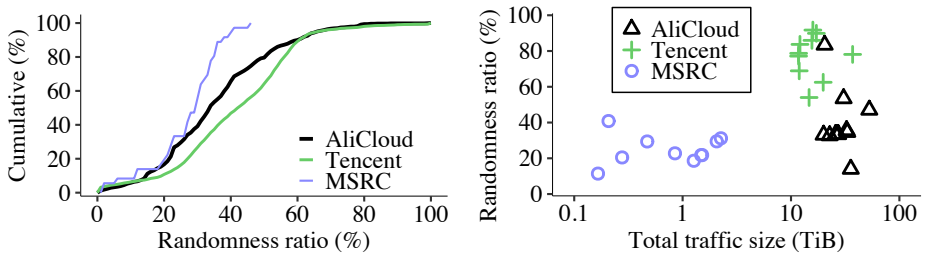
(a) AliCloud (b) TencentCloud (c) MSRC

Figure 15: Finding B.8: Average amount of traffic in 10-minute intervals each day from 12:00 AM to 11:59 PM.

We calculate the average number of requests and the average amount of traffic in 10-minute intervals each day from 12:00 AM to 11:59 PM. We divide one day into 144 10-minute timeslots. We collect the total number of requests and the total traffic size in each timeslot, and divide them by the number of days to obtain the averages for each timeslot. We align the timestamps of all three traces to their respective time zones: for AliCloud and TencentCloud, we align the timestamps of the requests to GMT+8, while for MSRC, we align them to GMT+0. Recall that the TencentCloud traces end at 1:00 AM on the tenth day and have a missing hour of requests at 1:00 AM-2:00 AM (§3.1). The average numbers of requests per day in the timeslots of 12:00 AM-1:00 AM and 1:00 AM-2:00 AM are obtained by the total number of requests in the timeslots divided by ten and eight, respectively; the average amounts of traffic are handled similarly.

Figures 14 and 15 show the distributions of the average numbers of requests and the average amounts of traffic across different timeslots for all three traces, respectively. We see that in both AliCloud and TencentCloud, their average numbers of requests and sizes of traffic are almost evenly spread across daytime (6:00 AM to 6:00 PM) and nighttime (6:00 PM to 6:00 AM). In AliCloud, the daytime has 52.6% of the total average number of I/O requests (Figure 14(a)) and 52.0% of total average traffic in all timeslots (Figure 15(a)), while in TencentCloud, the daytime has 50.3% of average I/O requests (Figure 14(b)) and 49.6% of average traffic (Figure 15(b)). However, in MSRC, the daytime only has 33.7% of average I/O requests and 23.3% of average traffic, and the I/O requests and traffic mainly dominate in nighttime (Figures 14(c) and 15(c)).

We also find that from Figures 15(a) and 15(c), AliCloud and MSRC volumes have spikes for reads at 0:00 AM-0:20 AM and 1:00 AM-2:30 AM, respectively, accounting for 13.1% and 18.8% of all read traffic, respectively; in contrast, TencentCloud does not have such spikes. For AliCloud, the reason of the spikes is that 12 out of the 1,000 volumes only have read requests during 0:00 AM-0:20 AM, and they contain significant numbers of large read requests. Each of these 12 volumes has a total of more than 50.2 GiB of average read traffic at 0:00 AM-0:20 AM per day. Their average read request sizes are larger than 360 KiB (note that most of the read requests are smaller than 100 KiB; see Finding A.2 in §3.2). We suspect that there exist scheduled scan activities in these 12 volumes at midnight, although we cannot identify their specific applications (§3.1). If we exclude these 12 volumes, the overall intensities of read traffic for AliCloud at midnight will be comparable to other time intervals. For MSRC, the reason of the spikes near midnight is that a volume called *src1_1* has an extremely large amount of read traffic at 1:00 AM-2:30 AM, accounting for 56.8-82.9% of the read traffic of all volumes in MSRC in the corresponding 10-minute time intervals. Note that the average read request sizes of *src1_1* are smaller than 43.1 KiB during the spike period, as opposed to the large requests in AliCloud. Read spikes near midnight are common in production; for example, in prior work [19], massive read spikes are observed at about 3:30 AM in a production server due to the scheduled replication tasks in early mornings.



(a) Randomness ratio distribution (b) Randomness ratio vs. traffic

Figure 16: Finding B.9: Cumulative distributions of randomness ratios of volumes (figure (a)) and the relationship between the randomness ratios and total traffic sizes in top-10 traffic-intensive volumes.

4.2 Spatial Patterns

We study the spatial characteristics of volumes in AliCloud, TencentCloud, and MSRC through the following metrics. First, we study the randomness of I/O requests by examining the offset differences of recent requests [1, 39], as random I/Os can compromise the performance and endurance of flash-based storage [31]. Second, we examine the aggregations of reads and writes in working sets, so as to provide guidelines for resource allocation in caching [21, 23, 37]. Finally, we examine the patterns of update coverage (i.e., the percentage of WSS for updates), which is important for optimizing update performance in storage cluster management [12].

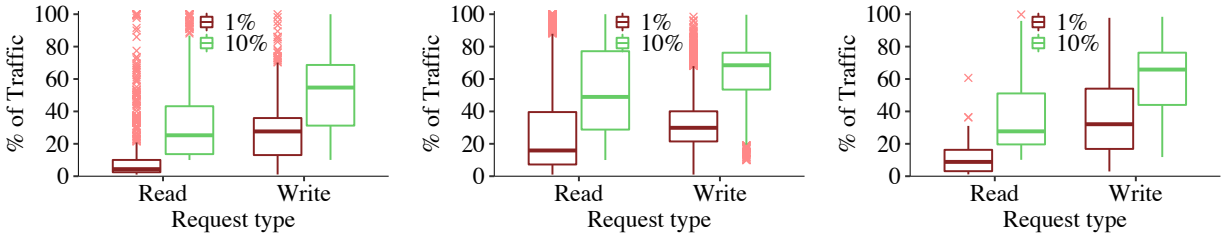
Finding B.9: *Random I/Os are common in AliCloud, TencentCloud, and MSRC. The volumes in AliCloud and TencentCloud see higher percentages of random I/Os than those in MSRC.*

We study the randomness of I/O requests by examining the spatial relationships among adjacent requests. To quantify the randomness of a request, we measure the minimum distance between the current offset of the request and the offsets of the previous 32 requests [1, 39]. If the minimum distance exceeds a threshold (e.g., 128 KiB, which is the read-ahead length of the surveyed drives in [39]), we regard the request as a random request. We measure the *randomness ratio* of a volume, defined as the percentage of random requests over all requests.

Figure 16(a) shows the cumulative distributions of randomness ratios of volumes in AliCloud, TencentCloud, and MSRC. We find that random I/Os are common in all three traces. Half of the volumes have at least 33.5%, 42.1%, and 29.4% of random I/Os in AliCloud, TencentCloud, and MSRC, respectively. Also, AliCloud and TencentCloud in general show higher randomness ratios than MSRC. In particular, all volumes in MSRC have less than 46% of random requests, while 20.4% and 35.8% of volumes in AliCloud and TencentCloud have more than 50% of random requests, respectively. The existence of randomness may come from file system workloads as witnessed in [17].

We further examine the randomness ratios of the top-10 volumes that have the most I/O traffic in each trace. Figure 16(b) shows the relationships between the randomness ratios and the total I/O traffic sizes of the top-10 volumes. We see that the volumes with large amounts of I/O traffic have high randomness ratios in general. The randomness ratios of the top-10 volumes in AliCloud, TencentCloud, and MSRC are 14.0-83.4%, 54.0-91.7%, and 11.4-40.8%, respectively, and their I/O traffic sizes are 20.0-52.8 TiB, 11.7-36.8 TiB, and 0.17-2.26 TiB, respectively. We also examine the Spearman correlation coefficients [38] between the I/O traffic sizes and the randomness ratios in the top-10 volumes of all three traces. We find that the coefficients are 0.079, 0.164, and 0.333 in AliCloud, TencentCloud, and MSRC, respectively, implying that there exist positive correlations between the two metrics in the top-10 volumes. The results indicate that random I/Os are common in traffic-intensive volumes.

Combining with the observation that small-size I/O requests dominate in all three traces (§3.2), we see that random and small I/Os are common in all three traces, especially in AliCloud and TencentCloud. Such



(a) AliCloud (b) TencentCloud (c) MSRC

Figure 17: Finding B.10: Boxplots of percentages of traffic sizes for the top-1% and top-10% read and write blocks across all volumes.

access patterns can compromise the performance and endurance of flash-based storage [17, 31].

Finding B.10: *Reads and writes are aggregated in small working sets in non-negligible fractions of volumes in AliCloud, TencentCloud, and MSRC, while TencentCloud has the highest aggregation of reads among all three traces. Writes are more aggregated than reads.*

We study how reads and writes are aggregated in the working sets of each volume. Specifically, in the read (or write) working sets, we focus on the top-1% and top-10% of unique blocks that receive the most read (or write) traffic. We examine the percentage of the read (or write) traffic size of each such block over the total read (or write) traffic size; a higher percentage implies that the I/O traffic is more aggregated in a block.

Figure 17 shows the boxplots of percentages of traffic sizes for the top-1% and top-10% blocks across all volumes in AliCloud, TencentCloud, and MSRC. We first focus on read traffic. We see that read traffic is aggregated in the top-1% and top-10% blocks in non-negligible fractions of volumes. In AliCloud, 75% of volumes have at least 2.5% and 13.6% of read traffic in the top-1% and top-10% read blocks, respectively (Figure 17(a)). In TencentCloud, the corresponding percentages are 7.2% and 28.8%, respectively (Figure 17(b)), and in MSRC, the corresponding percentages are 3.1% and 19.6%, respectively (Figure 17(c)). In particular, the aggregation for reads is the highest in TencentCloud among all three traces.

In AliCloud, the boxplots indicate 147 volumes as outliers in the top-1% read blocks (Figure 17(a)). Such outlier volumes have more than 21.3% of read traffic in their top-1% read blocks. It implies that a small read cache can absorb a substantial amount of read traffic for such volumes.

Compared with reads, writes are more aggregated. In AliCloud, the 25th percentiles of read traffic in the top-1% and top-10% read blocks are 2.5% and 13.6%, respectively, while the 25th percentiles of write traffic in the top-1% and top-10% written blocks increase to 13.0% and 31.2%, respectively (Figure 17(a)). Similar observations hold in TencentCloud and MSRC. For example, in TencentCloud, the 25th percentiles of read and write traffic in the top-10% read and written blocks are 28.8% and 53.5%, respectively (Figure 17(b)), and in MSRC, the corresponding 25th percentiles are 19.6% and 44.0%, respectively (Figure 17(c)). Note that the spatial clustering of writes is common in desktop applications and is related to files such as mail boxes, search indexes, and file system metadata [37].

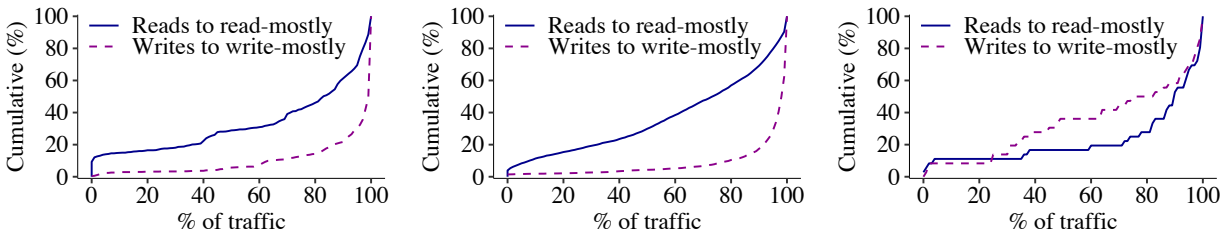
Finding B.11: *Reads and writes tend to be aggregated in read-mostly and write-mostly blocks, respectively, in AliCloud and TencentCloud.*

We further classify the blocks into different types as in [23] and examine the aggregation of reads and writes. Specifically, we classify a block as *read-mostly* (or *write-mostly*) if its read (or write) traffic occupies more than 95% of its total I/O traffic. We examine the percentage of all read (or write) traffic in the whole trace duration that goes to read-mostly (or write-mostly) blocks.

Table 5 shows the overall percentages of all read and write traffic that goes to read-mostly and write-mostly blocks in AliCloud, TencentCloud, and MSRC. In AliCloud and TencentCloud, the majority of read traffic (59.1% and 78.5%, respectively) and write traffic (80.7% and 90.8%, respectively) goes to read-mostly blocks and write-mostly blocks, respectively. In MSRC, 72.1% of read traffic goes to read-mostly blocks;

Traces	AliCloud	TencentCloud	MSRC
Reads to read-mostly blocks (%)	59.2	78.5	75.9
Writes to write-mostly blocks (%)	80.7	90.8	33.5

Table 5: Finding B.11: Percentages of all read and write traffic going to read-mostly and write-mostly blocks, respectively.



(a) AliCloud (b) TencentCloud (c) MSRC

Figure 18: Finding B.11: Cumulative distributions of all percentages of read and write traffic going to read-mostly and write-mostly blocks, respectively, across all volumes.

however, only 32.4% of write traffic goes to write-mostly blocks. Overall, both AliCloud and TencentCloud show prominent aggregations of reads and writes in read-mostly and write-mostly blocks, respectively, but it is not the case in MSRC. Note that the limited aggregation of writes in write-mostly blocks in MSRC is inconsistent with the prior finding in [23]. The reason is that the study in [23] considers only 12 out of 36 volumes in MSRC, while we consider all 36 volumes.

Figure 18 shows the cumulative distributions of percentages of read and write traffic that goes to read-mostly and write-mostly blocks, respectively, across all volumes in AliCloud, TencentCloud, and MSRC. Most of the volumes in all three traces have high percentages of all read and write traffic aggregated in read-mostly and write-mostly blocks, respectively. In AliCloud, half of the volumes have more than 82.6% of reads going to read-mostly blocks and more than 99.2% writes going to write-mostly blocks (Figure 18(a)); in TencentCloud, the corresponding percentages are 73.4% and 98.0%, respectively (Figure 18(b)); in MSRC, the corresponding percentages are 89.4% and 78.3%, respectively (Figure 18(c)).

Finding B.12: *AliCloud and TencentCloud generally have higher update coverages and higher percentages of update traffic than MSRC. The update coverage also varies across volumes.*

Recall that Table 1 (Section 3.2) shows the overall WSSs (working set sizes) for reads, writes, and updates. We now examine the spatial characteristics of updates. We focus on the update working set, which covers the blocks that are written more than once. We measure the *update coverage* of a volume, defined as the ratio between the update WSS and the total WSS of the volume [12]. In addition, we measure the percentages of update traffic over the total amount of traffic across all volumes.

Table 6 shows the averages, medians, and 90th percentiles of update coverages of all volumes in all three traces. In general, AliCloud and TencentCloud have higher update coverages than MSRC. In AliCloud and TencentCloud, half of the volumes have update coverages of more than 61.2% and 56.7%, respectively, while in MSRC, the corresponding percentage is 9.4% only. In addition, Table 7 shows that AliCloud and TencentCloud have higher percentages of update traffic than MSRC in terms of the averages, medians, and 90th percentiles. This suggests that AliCloud and TencentCloud are more update-intensive than MSRC.

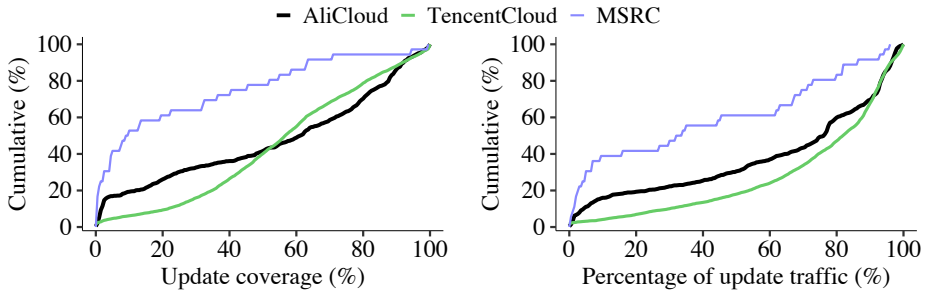
Figure 19(a) shows the cumulative distributions of update coverages across all volumes in AliCloud, TencentCloud, and MSRC. In AliCloud and TencentCloud, 45.2% and 37.2% of volumes have update coverages larger than 65%, respectively, and their update coverages are more diverse than MSRC. On the other hand, in MSRC, 33 out of 36 volumes have update coverages below 65%. Figure 19(b) shows the cumulative distributions of percentages of update traffic. AliCloud and TencentCloud show higher percentages

Traces	AliCloud	TencentCloud	MSRC
Mean (%)	52.7	56.2	24.1
Median (%)	61.2	56.7	9.4
90th PCTL (%)	92.1	91.9	63.0

Table 6: Finding B.12: Means, medians and 90th percentiles of update coverages of all volumes.

Traces	AliCloud	TencentCloud	MSRC
Mean (%)	62.5	72.5	38.7
Median (%)	76.0	81.9	32.7
90th PCTL (%)	96.4	96.1	84.2

Table 7: Finding B.12: Means, medians, and 90th percentiles of the percentages of update traffic of all volumes.



(a) Update coverages (b) Percentages of update traffic

Figure 19: Finding B.12: Cumulative distributions of update coverages and percentages of update traffic across all volumes.

of update traffic than MSRC, and TencentCloud generally has a higher percentage than AliCloud.

4.3 Temporal Patterns

We study the temporal characteristics of volumes in AliCloud, TencentCloud, and MSRC by examining the temporal relationships of adjacent I/O requests. We first examine the time elapsed between adjacent requests to the same block with respect to different combinations of read and write requests for workload-aware caching designs [37]. We also study the update interval (i.e., the time interval between two consecutive writes to the same block), which facilitates flash-based storage management [10, 24]. Finally, we study the miss ratios under least recently used (LRU) caching, which reflects the temporal aggregation of traffic for caching efficiency [43, 47].

Recall that the TencentCloud traces have a missing hour of requests at 1:00 AM-2:00 AM on the eighth day (§3.1). Thus, in our following analysis for TencentCloud, we discard the adjacent requests that span across the missing hour.

Finding B.13: *The read-after-write (RAW) times in AliCloud, TencentCloud, and MSRC are generally larger than the write-after-write (WAW) times. Also, TencentCloud generally has smaller RAW times than AliCloud and MSRC, while AliCloud generally has larger WAW times than TencentCloud and MSRC. Furthermore, AliCloud and TencentCloud have significantly more WAW requests than RAW requests.*

We first examine two types of adjacent requests [37]: (i) a *read-after-write (RAW)* request, which refers to the read following immediately the write to the same block; and (ii) a *write-after-write (WAW)* request, which refers to the write following immediately the write to the same block. We measure the time of a RAW (resp. WAW) request as the elapsed time between the adjacent read and write (resp. the two adjacent writes) to the same block.

Figures 20(a)-20(c) show the cumulative distributions of RAW and WAW times across all RAW and WAW requests, respectively, in all three traces. All three traces generally have larger RAW times than WAW times. Specifically, the 50th percentiles of the RAW time in AliCloud, TencentCloud, and MSRC are 3.0 hours, 4.9 minutes, and 16.1 hours, respectively, while the 50th percentiles of the WAW time are only 1.3 hours,

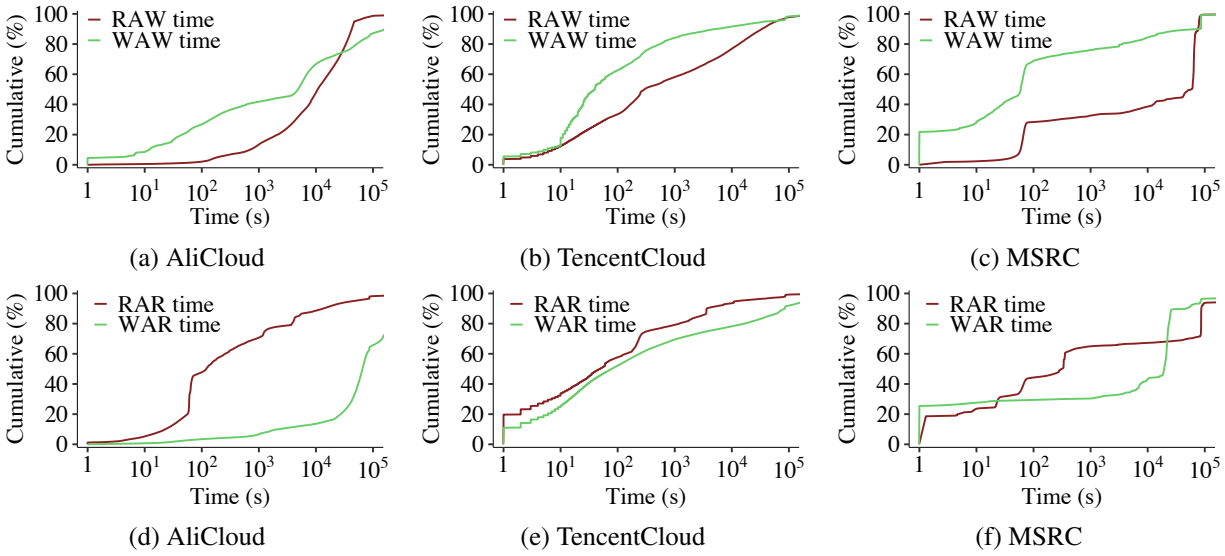


Figure 20: Findings B.13-B.14: Cumulative distributions of RAW, WAW, RAR, and WAR times across all RAW, WAW, RAR, and WAR requests, respectively.

0.7 minutes, and 1.0 minute, respectively. Such findings are consistent with those in prior studies [18, 37]. As for the possible reasons, the smaller WAW times are likely to appear in desktop workloads [37], while the larger RAW times are possibly related to the large OS-level buffer caches [37].

Also, all three traces have different percentages of large and small RAW and WAW times. To aid our analysis, we treat the times smaller than 1.0 minute and larger than 15.0 minutes as small and large, respectively, as also used in [37]. For RAW times, TencentCloud has the largest percentage of RAW times smaller than 1.0 minute among all three traces, while most of the RAW times in AliCloud and MSRC are larger than 15.0 minutes. Specifically, 1.4%, 29.8%, and 10.9% of the RAW times are smaller than 1.0 minute in AliCloud, TencentCloud, and MSRC, respectively, while 87.8%, 42.5%, and 67.8% of the RAW times are larger than 15.0 minutes in the three traces, respectively. On the other hand, most of the WAW times are smaller than 1.0 minute in TencentCloud and MSRC, while most of the WAW times are larger than 15.0 minutes in AliCloud. Specifically, 22.7%, 57.7%, and 51.5% of WAW times are smaller than 1.0 minute in AliCloud, TencentCloud, and MSRC, respectively, while 58.5%, 16.4%, and 24.3% of WAW times are larger than 15.0 minutes in the three traces, respectively.

Table 8 shows the numbers of RAW and WAW requests in AliCloud, TencentCloud, and MSRC. We observe a large difference in the numbers of RAW and WAW requests in both AliCloud and TencentCloud, but a small difference in MSRC. Specifically, in AliCloud, the numbers of RAW and WAW requests are 12.4 billion and 103.7 billion, respectively; the number of WAW requests is 8.3× that of RAW requests. In TencentCloud, the numbers of RAW and WAW requests are 8.8 billion and 204.9 billion, respectively, and the number of WAW requests is even 23.3× that of RAW requests. In MSRC, those numbers are 297.2 million and 289.2 million, respectively, and are close to each other. We suspect that the larger number of WAW requests in AliCloud and TencentCloud may be related to the metadata and log files in the workloads; for example, the fraction of WAW requests drops significantly after the metadata and log files are excluded [18].

Finding B.14: *TencentCloud has the highest fractions of small RAR and WAR times and the smallest fractions of large RAR and WAR times in all three traces. There exist extremely small RAR and WAR times in MSRC. In all three traces, the WAR time is much larger than the RAR time, and there are much more RAR requests than WAR requests.*

We further examine two types of adjacent requests: (i) a *read-after-read (RAR)* request, which refers to the read following immediately the read to the same block; and (ii) a *write-after-read (WAR)* request, which

Traces	RAW (M)	WAW (M)	RAR (M)	WAR (M)
AliCloud	12,432.7	103,708.4	29,845.0	11,760.6
TencentCloud	8,796.0	204,856.2	63,990.4	7,930.3
MSRC	297.2	289.2	1,382.4	330.0

Table 8: Findings B.13-B.14: Numbers of RAW, WAW, RAR, and WAR requests (in millions).

refers to the write following immediately the read to the same block.

Figures 20(d)-20(f) show the cumulative distributions of RAR and WAR times across all RAR and WAR requests, respectively, in all three traces. We again treat the times smaller than 1.0 minute and larger than 15.0 minutes as small and large, respectively, as above. TencentCloud has the highest fractions of RAR and WAR times smaller than 1.0 minute, and the lowest fractions of RAR and WAR times larger than 15.0 minutes. Specifically, 28.5%, 53.4%, and 35.6% of the RAR times are smaller than 1.0 minute in AliCloud, TencentCloud, and MSRC, respectively, while 30.0%, 21.2%, and 35.2% of the RAR times are larger than 15.0 minutes, respectively. On the other hand, 2.8%, 47.6%, and 29.2% of the WAR times are smaller than 1 minute in AliCloud, TencentCloud, and MSRC, respectively, while 93.8%, 31.1%, and 69.7% of the WAR times are larger than 15 minutes, respectively. In particular, in MSRC, there exist non-negligible fractions of extremely small RAR and WAR times (18.5% and 25.4%, respectively) that are smaller than 1 second, which is not the case for AliCloud and TencentCloud.

Overall, in all three traces, the WAR time is much larger than the RAR time. In AliCloud, the 50th percentiles of RAR and WAR times are 2.0 minutes and 18.2 hours, respectively, and 21.0% and 88.8% of RAR and WAR times are larger than 1 hour, respectively (Figure 20(d)). In TencentCloud, the 50th percentiles of RAR and WAR time are 49 seconds and 78 seconds, respectively, and 11.1% and 25.2% of RAR and WAR times are larger than 1 hour, respectively (Figure 20(e)). In MSRC, the 50th percentiles of RAR and WAR times are 5.2 minutes and 5.4 hours, respectively, and 33.6% and 66.7% of RAR and WAR times are larger than 1 hour, respectively (Figure 20(f)). The results indicate that a block being read is likely read again soon.

We also examine the numbers of RAR and WAR requests in AliCloud, TencentCloud, and MSRC, as shown in Table 8. In AliCloud, TencentCloud, and MSRC, the numbers of RAR requests are 2.54×, 8.07×, and 4.19× those of WAR requests, respectively.

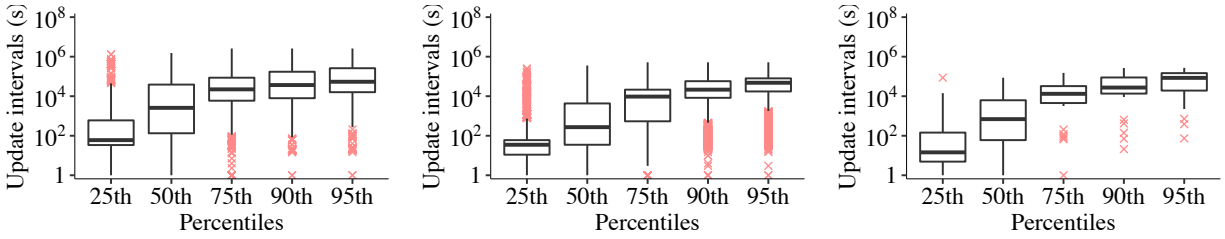
Finding B.15: *Written blocks have varying update intervals.*

We measure the *update interval* of a block, defined as the elapsed time between two consecutive writes to the same block. Note that the update interval differs from the WAW time, as the former allows reads between two writes. Each block may be written more than once, so it may be associated with multiple update intervals (e.g., a block that is written M times has $M - 1$ update intervals). The update interval of a block describes the lifetime of the block data.

Table 9 shows different percentiles of update intervals across all volumes in all three traces. In AliCloud, the update intervals generally have long durations, while in TencentCloud and MSRC, the update intervals are generally small, especially in TencentCloud. In AliCloud, 50% of update intervals are larger than 95.2 minutes (1.6 hours), and the 90th percentile is 3,017.4 minutes (50.3 hours). In TencentCloud, the 25th, 50th, and 75th percentiles are only several seconds or minutes (0.23 minutes, 0.67 minutes, and 5.4 minutes, respectively), implying that the majority of updated blocks have extremely high update frequencies. However, some updated blocks still have high update intervals, as the 90th and 95th percentiles are 120.0 minutes (2.0 hours) and 973.1 minutes (16.2 hours), respectively. In MSRC, the update intervals have a bimodal pattern, in which 50% of update intervals are smaller than 1.25 minutes, while 25% of update intervals are larger than 1,438.9 minutes (24.0 hours). The reason of such a bimodal pattern in MSRC is that a volume is responsible for source control (i.e., *src1_0*) and updates data blocks daily. If we exclude the daily updates,

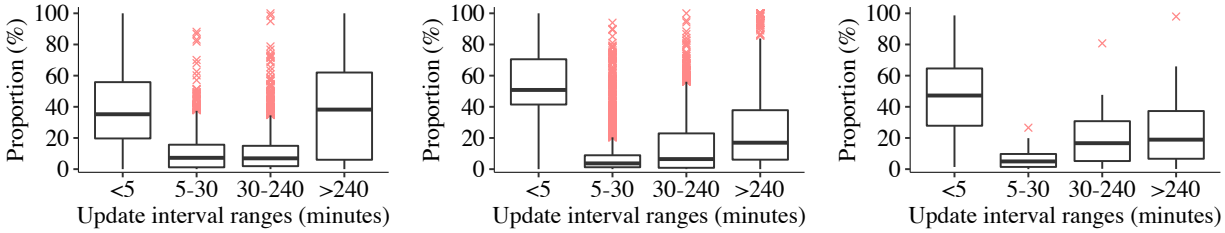
Percentiles (minutes)	25th	50th	75th	90th	95th
AliCloud	1.86	95.2	926.3	3,017.4	7,200.5
TencentCloud	0.23	0.67	5.4	120.0	973.1
MSRC	0.73	1.25	1,438.9	1,440.5	1,444.1

Table 9: Finding B.15: Overall percentiles of update intervals across all volumes.



(a) AliCloud (b) TencentCloud (c) MSRC

Figure 21: Finding B.15: Boxplots of percentiles of update intervals across all volumes.



(a) AliCloud (b) TencentCloud (c) MSRC

Figure 22: Finding B.15: Boxplots of proportions for the four groups of update intervals across all volumes.

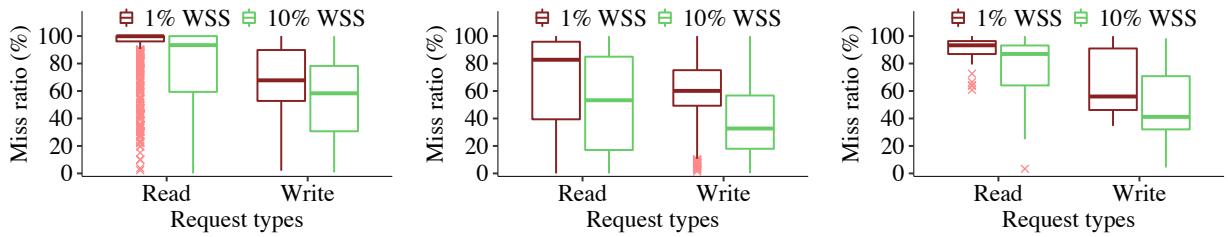
most of the written blocks in MSRC have very short update intervals.

Figure 21 shows the boxplots of update intervals of different groups of percentiles across all volumes in AliCloud, TencentCloud, and MSRC. We see that the distributions of update intervals have high variations across volumes in all three traces. For example, in AliCloud, the 50th percentiles of update intervals of all volumes range from 1 second to 17.8 days (Figure 21(a)); in TencentCloud, the 50th percentiles vary between 1 second and 4.14 days (Figure 21(b)); in MSRC, the 50th percentiles of update intervals of all volumes range from 1 second to 24 hours.

Many volumes have non-negligible proportions of short update intervals in their update requests. To further examine the distributions of update intervals in individual volumes, we divide the update intervals into four groups: (i) less than 5 minutes, (ii) 5-30 minutes, (iii) 30-240 minutes, and (iv) more than 240 minutes. We calculate the proportions for the four groups of update intervals for each volume, and represent the proportions across all volumes by boxplots.

Figure 22 shows the boxplots of proportions for the four groups of update intervals across all volumes in all three traces. All three traces have large proportions of either very small or very large update intervals. In AliCloud, half of the volumes have more than 35.2% and 38.2% of update intervals in less than 5 minutes and in more than 240 minutes, respectively (Figure 22(a)). In TencentCloud, the corresponding percentages are 50.8% and 17.0%, respectively (Figure 22(b)), while in MSRC, the corresponding percentages are 47.2% and 18.9%, respectively (Figure 22(c)). Thus, a substantial amount of data is either updated frequently or not updated for long.

Finding B.16. *Many volumes in TencentCloud have low miss ratios even under a small cache size, while there are fewer such volumes with low miss ratios in AliCloud and MSRC. Also, when the cache size increases, AliCloud and TencentCloud show the highest absolute reductions in read and write miss ratios, respectively,*



(a) AliCloud (b) TencentCloud (c) MSRC

Figure 23: Finding B.16: Boxplots of miss ratios for reads and writes across all volumes, under the cache sizes of 1% and 10% of the WSS of a volume.

in all three traces.

Finally, we study the impact of caching with respect to the temporal patterns of the volumes. For each volume, we simulate a fixed-size cache for both reads and writes using the LRU policy, and evaluate the corresponding cache miss ratios for reads and writes. Here, we select 1% and 10% of the WSS of a volume as the cache size.

Figure 23 shows the boxplots of miss ratios across all volumes in all three traces. Some volumes show low miss ratios (i.e., LRU-based caching is effective). For the cache size of 10% of WSS, the 25th percentiles of the miss ratios for reads and writes are 59.4% and 30.7% in AliCloud, respectively (Figure 23(a)), while the corresponding miss ratios are 17.0% and 17.9% in TencentCloud, respectively (Figure 23(b)), and 64.1% and 32.0% in MSRC, respectively (Figure 23(c)). Also, some volumes in TencentCloud can have very low miss ratios when the cache size is only 1% of WSS, implying that the access patterns of such volumes have high temporal locality, while AliCloud and MSRC have fewer such volumes. The low miss ratios in TencentCloud suggest that we can potentially improve the read and write performance using a small-size cache. Such findings are also consistent with those in [52], which shows low miss ratios of some selected volumes with small-size cache.

AliCloud has the highest absolute reductions in read miss ratios when the cache size increases from 1% to 10% of WSS among all three traces; for write miss ratios, TencentCloud shows the highest absolute reductions for increased cache sizes. In AliCloud, the 25th percentiles of the miss ratios for reads and writes reduce from 96.1% to 59.4% and from 52.8% to 30.7% (i.e., 36.7% and 22.1% of absolute reduction), respectively (Figure 23(a)). In TencentCloud, the 25th percentiles of the miss ratios for reads and writes reduce from 39.4% to 17.0% and from 49.3% to 17.9% (i.e., 22.4% and 31.4% of absolute reduction, Figure 23(b)), while in MSRC, the 25th percentiles of the miss ratios for reads and writes reduce from 86.9% to 64.1% and from 46.2% to 32.1% (i.e., 22.8% and 14.1% of absolute reduction), respectively (Figure 23(c)). In short, in all three traces, AliCloud and TencentCloud have the highest reductions in read and write miss ratios, respectively, implying the significance of enlarging the cache size in cloud block storage workloads.

4.4 Similarities and Differences Between Two Cloud Block Storage Traces

We highlight the major similarities and differences between the two cloud block storage traces, AliCloud and TencentCloud.

Load intensities. AliCloud and TencentCloud show similar intensities of volumes (Finding B.1), but different observations in burstiness and activeness.

- In terms of burstiness, TencentCloud has lower overall burstiness than AliCloud, and also has a lower fraction of volumes with high burstiness ratios (Finding B.2). Nevertheless, both AliCloud and TencentCloud have more diverse burstiness across volumes than MSRC (Finding B.3).
- In terms of activeness, TencentCloud has higher activeness than AliCloud, in both the number of active volumes and the active time in each volume (Finding B.5). While writes are the dominant factor in

the activeness of both cloud block storage traces (Finding B.6), TencentCloud is more read-active than AliCloud (Finding B.7).
- In terms of the distribution of traffic per day, unlike MSRC, the traffic of both AliCloud and TencentCloud is almost evenly spread across daytime and nighttime. While AliCloud shows substantial read traffic at midnight, TencentCloud does not (Finding B.8).

Spatial patterns. While AliCloud and TencentCloud show higher randomness than MSRC, TencentCloud generally has higher randomness in I/O requests and higher levels of traffic aggregation in blocks than AliCloud.
- In terms of the randomness in I/Os, the volumes in both AliCloud and TencentCloud have higher percentages of random I/Os than those in MSRC. Compared with AliCloud, TencentCloud generally shows a higher percentage of random I/Os. For example, TencentCloud has a higher fraction of random requests in half of the volumes than AliCloud (42.1% versus 33.5%) (Finding B.9).
- In terms of the aggregation of traffic, TencentCloud shows more traffic in top-1% and top-10% of the most access-intensive blocks than AliCloud, indicating that TencentCloud has a higher degree of traffic aggregation in a small set of blocks (Finding B.10). Also, while both AliCloud and TencentCloud have higher percentages of writes in write-mostly blocks than MSRC, TencentCloud has higher read and write traffic aggregations than AliCloud in read-mostly and write-mostly blocks, respectively (Finding B.11). Furthermore, while both AliCloud and TencentCloud have higher fractions of update-intensive volumes (i.e., the volumes with high percentages of update traffic) than MSRC, TencentCloud has a higher percentage of update-intensive volumes than AliCloud (Finding B.12).

Temporal patterns. TencentCloud generally has lower access intervals on blocks in reads, writes, and updates. It also has lower miss ratios than AliCloud under the same cache configurations.
- In terms of the time intervals in accessing the same blocks, both AliCloud and TencentCloud generally have smaller WAW times than RAW times and smaller RAR times than WAR times. However, TencentCloud generally has lower RAW, WAW, RAR, and WAR times than AliCloud (Findings B.13 and B.14). It indicates that TencentCloud has more access-intensive workloads than AliCloud.
- In terms of the update intervals, TencentCloud has 90% of its update intervals smaller than 2.0 hours, and its update intervals are generally smaller than AliCloud (Finding B.15).
- In terms of miss ratios, TencentCloud has lower miss ratios under the same cache configurations. It has lower absolute reductions in read miss ratios than AliCloud when the cache size increases from 1% to 10% of WSS. On the contrary, it has higher absolute reductions in write miss ratios than AliCloud when the cache size increases from 1% to 10% of WSS (Finding B.16).

4.5 Summary of Findings

Finally, we discuss the implications of our findings of the trace analysis in AliCloud, TencentCloud, and MSRC. We show how the findings address the design considerations for cloud block storage, including load balancing, cache efficiency, and storage cluster management (§2.2).

Load balancing. We focus on the average and peak intensities as well as the activeness of volumes. From Finding B.1, we observe that while many applications are hosted in the cloud, the volumes in cloud block storage (i.e., AliCloud and TencentCloud) have similar average load intensities to those in traditional data centers (i.e., MSRC) which were monitored more than a decade ago; however, the peak intensities are generally smaller.

From Findings B.2-B.4, we observe the existence of burstiness in a non-negligible fraction of volumes. While the overall burstiness remains low, the burstiness can be severe in individual volumes across many types of applications [34], indicating that these volumes may be provisioned for high peak intensities but most of the bandwidth resources remain unused [33]. The high burstiness may hence lead to performance degradations if load balancing is not properly maintained. Furthermore, the higher diversity of workload

burstiness makes load balancing in cloud block storage more challenging than in traditional data centers. Failing to deal with load imbalance and the diversity of workloads may cause problems to the physical devices in cloud block storage, such as higher flash failure rates [48]. Applying shared logs or distributed caches can ease the load imbalance among volumes [26, 48]. Also, the burstiness, as shown by the short inter-arrival times, suggests that I/O requests tend to arrive in groups and can be further exploited to improve I/O performance [18].

From Findings B.5-B.7, we observe that writes are the dominant factor of activeness in all three traces. In particular, most volumes in cloud block storage (i.e., AliCloud and TencentCloud) are write-dominant (§3.2). The differences of activeness in reads and writes indicate that removing writes can produce a high level of idle periods, so it is possible to offload writes (e.g., by redirecting writes to other storage locations) to create idle periods in cloud block storage workloads for power savings [33].

From Finding B.8, I/O traffic is evenly spread across both daytime and nighttime in AliCloud and TencentCloud. This challenges background task scheduling (e.g., garbage collection, defragmentation, and flushing caches [18, 34]) in cloud block storage. For example, performing background tasks only at nighttime may be ineffective to reduce the interference with foreground I/Os. A careful design of I/O scheduling for background and foreground activities becomes necessary for cloud block storage. In particular, we observe large read spikes near midnight on a daily basis in some of the volumes in AliCloud. How to prevent such I/O spikes from interfering with cloud block storage system as a whole needs careful attention.

Concerning the design of load balancing, the data placement strategies should be aware of the diversity of workloads, the burstiness of individual volumes, and the traffic distribution over time. The log-structured design [35] is proven useful for balancing the write traffic in cloud-scale flash-based storage [48].

Cache efficiency. We study the spatial and temporal characteristics of volumes, which provide guidelines for motivating new caching designs for cloud block storage.

From Findings B.10 and B.16, we observe the patterns of both spatial and temporal traffic aggregations in a small fraction of blocks, especially for writes. TencentCloud shows a stronger traffic aggregation compared with AliCloud and MSRC. Many volumes in cloud block storage show high aggregations of reads and writes, implying that it is viable to allocate limited cache resources for absorbing substantial amounts of reads and writes.

From Finding B.11, we observe that many volumes in cloud block storage have reads and writes aggregated in read-mostly and write-mostly blocks, respectively. Thus, one possible caching admission policy is to identify the read-mostly and write-mostly blocks in workloads, as such blocks can absorb a substantial amount of I/O traffic. Also, the read-mostly and write-mostly blocks can be put into different devices to separately reduce the read and write latencies [23].

From Findings B.13 and B.14, the blocks that have been written tend to be rewritten again. In contrast, the blocks that have been read tend to receive another write after a long period of time. Thus, if our goal is to absorb writes with caching, a possible strategy is to favor the caching of the blocks that have been written rather than those that have been read, as the latter may unlikely generate write hits. Also, cloud block storage can benefit from disk-based write caching [37], due to the limited reads from the disk-based cache.

Storage cluster management. Characterizing the spatial and temporal characteristics of volumes is also critical for storage cluster management. Here, we focus on flash-based storage (§2.1).

From Finding B.9, we observe that upper-layer applications in cloud block storage issue a high fraction of small and random I/Os, which are known to hurt both the performance and endurance of flash-based storage [31]. The log-structured storage design [35] and I/O clustering [31] can help mitigate the overhead of small and random I/Os.

From Findings B.12 and B.15, updates are common and have high variations across volumes, both spatially and temporally. The varying update coverage across different volumes requires the underlying caches to use adaptive caching methods to absorb update traffic. Also, the varying update patterns can harm

the effectiveness of garbage collection and wear leveling in flash [17]. Thus, cloud block storage systems should take into account the varying patterns when optimizing update workloads for flash-based storage. A possible direction is to maintain the flash-translation layer (FTL) at the system level [13] to flexibly coordinate the I/Os issued to flash.

From Finding B.13, a larger number of WAW requests than RAW requests in AliCloud and TencentCloud indicates that the next issued requests to newly written blocks tend to be writes instead of reads. If these written blocks are replicated across different nodes, we may choose to update only one copy and invalidate other copies, instead of updating all copies, in order to save the update overhead since the written data is likely to be rewritten again [18].

Unexpected results. Finally, we highlight some unexpected results reported in our findings.

From Finding B.6, we observe that the activeness of MSRC is dominated by writes, yet the MSRC is read-dominant (§3.2). The results indicate that MSRC is write-dominant from the perspective of activeness, but is read-dominant from the perspective of the amount of I/O traffic.

From Finding B.8, we observe read spikes near midnight in both AliCloud and MSRC traces. In the corresponding read requests of the spike period, the average read request size in AliCloud is larger than 360 KiB, while that in MSRC is smaller than 43.1 KiB. We suspect that there exist scheduled scan activities in the corresponding volumes of AliCloud.

From Finding B.11, the limited aggregation of writes in write-mostly blocks in MSRC is inconsistent with prior work [23], which emphasizes that most of the write requests access write-mostly blocks. The reason is that the previous study [23] considers 12 volumes, while we consider all 36 volumes instead.

5 Related Work

We review related work on the field studies on storage workloads and how they inspire storage system designs.

Characterization of storage workloads. Several field studies characterize storage workloads using block-level I/O traces in various architectures, such as consumer electronics [34], virtual machines [1], Windows servers [19, 33], smartphone applications [54], containerized applications [16], and virtual desktop infrastructures [20]. Yadgar et al. [49] perform I/O workload analysis and study the performance implications (e.g., read/write amplifications and flash read costs) for SSD-based storage. In contrast, our field study focuses on cloud block storage that supports a diverse set of cloud applications in large-scale production. In particular, we provide findings and insights on performance optimizations for load balancing, caching efficiency, and storage cluster management.

Table 10 summarizes the traces used in the existing block-level trace studies [1, 16, 19, 20, 33, 49, 52, 54] in the literature, in terms of the number of traces (or volumes in our case), the trace durations, the number of read and write requests, and the total data traffic of reads and writes. Our trace analysis has the largest scale compared with the existing block-level trace studies at the time of the writing. Ahmad et al. [1] do not show the overall statistics but mention that the total I/O size is at most 10 GiB. Harter et al. [16] only focus on reads, and their analysis comprises of 57 docker images, each of which has the average read traffic of 27 MiB. Zhang et al. [52] collected the TencentCloud traces, but they mainly focus on the cache allocation design based on trace-driven evaluation instead of providing detailed trace analysis.

Inspirations from load intensity. Some designs are inspired by the characteristics of load intensity in storage workloads. Narayanan et al. [33] offload writes to reduce power consumptions with the observation that some volumes are idle in reads, thereby removing writes in those volumes can increase the idle periods for power saving. SRCMap [41] reduces power consumptions using sampling and replication, based on the observation on the I/O size and intensity of active data sets. Ursa [21] adopts the log-structured design, based on the observation that small writes dominate in real-world workloads.

Inspirations from spatial patterns. Some designs exploit the spatial characteristics of storage workloads. BORG [9] organizes frequently written data in a small dedicated disk partition to reduce the I/O seek time.

	MSRC [33]	MS-Prod [19]	Zhou et al. [54]	Lee et al. [20]	Tencent -Cloud [52]	SSDTrace [49]	AliCloud
#Volumes	36	43	25	321	4,995	1	1,000
Duration (days)	7	0.003-1	< 0.34	28	9.04	0.40	31
#Reads (millions)	304.9	126.2	0.04	2455.4	10,030.2	342.9	5,058.6
#Writes (millions)	128.9	87.3	0.13	898.3	23,592.0	9.69	15,174.4
Read Traffic (TiB)	9.04	2.98	0.002	64.8	282.3	2.94	161.6
Write Traffic (TiB)	2.39	1.70	0.006	15.0	837.2	6.38	455.5

Table 10: Statistics of existing block-level trace studies [19, 20, 33, 49, 52, 54].

FlashTier [36] manages sparse address mappings in flash caching, as storage I/Os are often aggregated in a small number of blocks. Desnoyers [14] proposes an analytical model for cleaning algorithms in flash devices and analyzes the aggregation of written blocks in specific working sets. ACGR [23] regulates I/O accesses for flash storage, based on the observation of read and write aggregations in read-only and write-only blocks, respectively. To improve the update performance in erasure-coded storage, CodFS [12] proposes dynamic reserved space management for parity updates to address the varying working sets of updates across storage workloads, while PBS [53] exploits the large fractions of overwrites to mitigate parity update overhead.

Inspirations from temporal patterns. Some designs exploit the temporal characteristics of storage workloads. Griffin [37] leverages the large time intervals between writes and the subsequent reads to the same block to build an HDD-based write cache for improving the SSD lifetime. Arteaga et al. [6] propose a cache-optimized RAID technique to minimize the RAID overhead in cloud storage, based on the comparisons between write-back caching and write-through caching on a set of block I/O traces from production servers in the cloud. CloudCache [5] chooses the window size of the model based on the hit ratio analysis on two-week traces in the cloud. Some studies leverage the characteristics of update intervals in storage workloads for improving write performance [24], lifetime [10], garbage collection modeling, and data reduction [50] in SSDs. Counter Stacks [47], SHARDS [43], and OSCA [52] consider the reuse distance (i.e., the number of distinct items accessed between two accesses to the same item) to improve caching efficiency.

Cloud block storage systems. Several cloud block storage designs are proposed in the literature. Parallax [28] provides storage virtualization for virtual machines atop shared block storage. Blizzard [29] manages POSIX applications atop cloud block storage. Ursa [21] is a hybrid block storage system that combines HDDs and SSDs for cloud-scale virtual disks. PBS [53] supports erasure-coded cloud block storage with efficient updates. Our recent work SepBIT [45] is a data placement scheme that mitigates the write amplification of garbage collection in log-structured cloud block storage, and its design and evaluation are based on the AliCloud and TencentCloud traces. In this work, we conduct an in-depth trace analysis that provides suggestions for improving the cloud block storage design.

6 Conclusion

We present an in-depth comparative trace analysis on the production block-level I/O traces at Alibaba Cloud (AliCloud), Tencent Cloud Block Storage (TencentCloud), and Microsoft Research Cambridge (MSRC); the AliCloud and TencentCloud traces are from cloud block storage systems, while the MSRC trace is collected from enterprise data centers. We reveal the commonalities and differences of the three sources of traces. We first identify 6 findings through the high-level analysis on the basic I/O statistics. We also identify 16 findings through the detailed analysis, based on which we further discuss the implications on three practical design considerations for cloud block storage, including load balancing, cache efficiency, and storage cluster management.

References

[1] I. Ahmad. Easy and efficient disk I/O workload characterization in VMware ESX server. In *Proceedings of the 2007 IEEE International Symposium on Workload Characterization (IISWC'07)*, pages 149–158, 2007.

[2] Alibaba. Alibaba Block Traces. https://github.com/alibaba/block-traces, 2022.

[3] Alibaba. Alibaba Cloud Block Storage. https://www.alibabacloud.com/help/doc-detail/63136.htm, 2022.

[4] Amazon. Amazon EBS. https://aws.amazon.com/ebs/, 2022.

[5] D. Arteaga, J. Cabrera, J. Xu, S. Sundararaman, and M. Zhao. CloudCache: On-demand flash cache management for cloud computing. In *Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST'16)*, pages 355–369, 2016.

[6] D. Arteaga and M. Zhao. Client-side flash caching for cloud systems. In *Proceedings of the 7th ACM International Systems and Storage Conference (SYSTOR'14)*, pages 1–11, 2014.

[7] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, and M. Paleczny. Workload analysis of a large-scale key-value store. In *Proceedings of ACM SIGMETRICS*, pages 53–64, 2012.

[8] D. Beaver, S. Kumar, H. C. Li, J. Sobel, and P. Vajgel. Finding a needle in Haystack: Facebook's photo storage. In *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI'10)*, pages 47–60, 2010.

[9] M. Bhadkamkar, J. Guerra, L. Useche, S. Burnett, J. Liptak, R. Rangaswami, and V. Hristidis. BORG: Block-reORGanization for self-optimizing storage systems. In *Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST'09)*, pages 183–196, 2009.

[10] Y. Cai, Y. Luo, E. F. Haratsch, K. Mai, and O. Mutlu. Data retention in MLC NAND flash memory: Characterization, optimization, and recovery. In *Proceedings of the 21st IEEE International Symposium on High Performance Computer Architecture (HPCA'15)*, pages 551–563. IEEE, 2015.

[11] Z. Cao, S. Dong, S. Vemuri, and D. H. C. Du. Characterizing, modeling, and benchmarking RocksDB key-value workloads at Facebook. In *Proceedings of the 18th USENIX Conference on File and Storage Technologies (FAST'20)*, pages 209–223, 2020.

[12] J. C. W. Chan, Q. Ding, P. P. C. Lee, and H. H. W. Chan. Parity logging with reserved space: Towards efficient updates and recovery in erasure-coded clustered storage. In *Proceedings of the 12th USENIX Conference on File and Storage Technologies (FAST'14)*, pages 163–176, 2014.

[13] T.-c. Chiueh, W. Tsao, H.-C. Sun, T.-F. Chien, A.-N. Chang, and C.-D. Chen. Software orchestrated flash array. In *Proceedings of the 7th ACM International Systems and Storage Conference (SYSTOR'14)*, pages 1–11, 2014.

[14] P. Desnoyers. Analytic modeling of SSD write performance. In *Proceedings of the 5th ACM International Systems and Storage Conference (SYSTOR'12)*, pages 1–10, 2012.

[15] S. Han, P. P. C. Lee, F. Xu, Y. Liu, C. He, and J. Liu. An in-depth study of correlated failures in production SSD-based data centers. In *Proceedings of the 19th USENIX Conference on File and Storage Technologies (FAST'21)*, pages 417–429, 2021.

[16] T. Harter, B. Salmon, R. Liu, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau. Slacker: Fast distribution with lazy docker containers. In *Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST'16)*, pages 181–195, 2016.

[17] J. He, S. Kannan, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau. The unwritten contract of solid state drives. In *Proceedings of the 12th ACM European Conference on Computer Systems (EuroSys'17)*, pages 127–144, 2017.

[18] W. W. Hsu and A. J. Smith. Characteristics of I/O traffic in personal computer and server workloads. *IBM System Journal*, 42(2):347–372, 2003.

[19] S. Kavalanekar, B. Worthington, Q. Zhang, and V. Sharda. Characterization of storage workload traces from production windows servers. In *Proceedings of the 2008 IEEE International Symposium on Workload Characterization (IISWC'08)*, pages 119–128, 2008.

[20] C. Lee, T. Kumano, T. Matsuki, H. Endo, N. Fukumoto, and M. Sugawara. Understanding storage traffic characteristics on enterprise virtual desktop infrastructure. In *Proceedings of the 10th ACM International Systems and Storage Conference (SYSTOR'17)*, pages 1–11, 2017.

[21] H. Li, Y. Zhang, D. Li, Z. Zhang, S. Liu, P. Huang, Z. Qin, K. Chen, and Y. Xiong. URSA: Hybrid block storage for cloud-scale virtual disks. In *Proceedings of the 14th ACM European Conference on Computer Systems (EuroSys'19)*, pages 1–17, 2019.

[22] J. Li, Q. Wang, P. P. C. Lee, and C. Shi. An in-depth analysis of cloud block storage workloads in large scale production. In *Proceedings of the 2020 IEEE International Symposium on Workload Characterization (IISWC'20)*, pages 37–47, 2020.

[23] Q. Li, L. Shi, C. J. Xue, K. Wu, C. Ji, Q. Zhuge, and E. H.-M. Sha. Access characteristic guided read and write cost regulation for performance improvement on flash memory. In *Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST'16)*, pages 125–132, 2016.

[24] R. S. Liu, C. L. Yang, and W. Wu. Optimizing NAND flash-based SSDs via retention relaxation. In *Proceedings of the 10th USENIX Conference on File and Storage Technologies (FAST'12)*, pages 1–11, 2012.

[25] S. Liu, S. Wang, Q. Cao, Z. Lu, H. Jiang, J. Yao, Y. Dong, and P. Yang. Analysis of and optimization for write-dominated hybrid storage nodes in cloud. In *Proceedings of ACM Symposium on Cloud Computing 2019 (SoCC'19)*, pages 403–415, 2019.

[26] Z. Liu, Z. Bai, Z. Liu, X. Li, C. Kim, V. Braverman, X. Jin, and I. Stoica. DistCache: provable load balancing for large-scale storage systems with distributed caching. In *Proceedings of the 17th USENIX Conference on File and Storage Technologies (FAST'19)*, pages 143–157, 2019.

[27] S. Maneas, K. Mahdaviani, T. Emami, and B. Schroeder. A study of SSD reliability in large scale enterprise storage deployments. In *Proceedings of the 18th USENIX Conference on File and Storage Technologies (FAST'20)*, pages 137–149, 2020.

[28] D. T. Meyer, G. Aggarwal, B. Cully, G. Lefebvre, M. J. Feeley, N. C. Hutchinson, and A. Warfield. Parallax: Virtual disks for virtual machines. In *Proceedings of the 3rd ACM European Conference on Computer Systems (EuroSys'08)*, pages 41–54, 2008.

[29] J. Mickens, E. B. Nightingale, J. Elson, K. Nareddy, D. Gehring, B. Fan, A. Kadav, V. Chidambaram, and O. Khan. Blizzard: Fast, cloud-scale block storage for cloud-oblivious applications. In *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI'14)*, pages 257–273, 2014.

[30] Microsoft. MSR Cambridge Traces. http://iotta.snia.org/traces/388, 2022.

[31] C. Min, K. Kim, H. Cho, S.-W. Lee, and Y. I. Eom. SFS: Random write considered harmful in solid state drives. In *Proceedings of the 10th USENIX Conference on File and Storage Technologies (FAST'12)*, pages 1–16, 2012.

[32] A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das. Towards characterizing cloud backend workloads: Insights from google compute clusters. In *Proceedings of ACM SIGMETRICS*, pages 34–41, 2010.

[33] D. Narayanan, A. Donnelly, and A. Rowstron. Write Off-Loading: Practical power management for enterprise storage. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST'08)*, pages 253–267, 2008.

[34] A. Riska and E. Riedel. Disk drive level workload characterization. In *Proceedings of the 2006 USENIX Annual Technical Conference (USENIX ATC'06)*, pages 97–102, 2006.

[35] M. Rosenblum and J. K. Ousterhout. The design and implementation of a log-structured file system. *ACM Transactions on Computer Systems*, 10(1):26–52, 1992.

[36] M. Saxena, M. M. Swift, and Y. Zhang. FlashTier: a lightweight, consistent and durable storage cache. In *Proceedings of the 7th ACM European Conference on Computer Systems (EuroSys'12)*, pages 267–280, 2012.

[37] G. Soundararajan, V. Prabhakaran, M. Balakrishnan, and T. Wobber. Extending SSD lifetimes with disk-based write caches. In *Proceedings of the 8th USENIX Conference on File and Storage Technologies (FAST'10)*, pages 101–114, 2010.

[38] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 100(3/4):441–471, 1987.

[39] M. Tarihi, H. Asadi, and H. Sarbazi-Azad. DiskAccel: Accelerating disk-based experiments by representative sampling. In *Proceedings of ACM SIGMETRICS*, pages 297–308, 2015.

[40] Tencent. Tencent Block Storage. http://iotta.snia.org/traces/27917, 2022.

[41] A. Verma, R. Koller, L. Useche, and R. Rangaswami. SRCMap: Energy proportional storage using dynamic consolidation. In *Proceedings of the 8th USENIX Conference on File and Storage Technologies (FAST'10)*, pages 267–280, 2010.

[42] M. Wajahat, A. Yele, T. Estro, A. Gandhi, and E. Zadok. Distribution fitting and performance modeling for storage traces. In *Proceedings of the 27th IEEE International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'19)*, pages 138–151. IEEE, 2019.

[43] C. A. Waldspurger, N. Park, A. Garthwaite, and I. Ahmad. Efficient MRC construction with SHARDS. In *Proceedings of the 13th USENIX Conference on File and Storage Technologies (FAST'15)*, pages 95–110, 2015.

[44] H. Wang, X. Yi, P. Huang, B. Cheng, and K. Zhou. Efficient SSD caching by avoiding unnecessary writes using machine learning. In *Proceedings of the 47th ACM International Conference on Parallel Processing (ICPP'18)*, pages 1–10, 2018.

[45] Q. Wang, J. Li, P. P. C. Lee, T. Ouyang, C. Shi, and L. Huang. Separating data via block invalidation time inference for write amplification reduction in log-structured storage. In *Proceedings of the 20th USENIX Conference on File and Storage Technologies (FAST'22)*, pages 429–444, 2022.

[46] S. Wang, Z. Lu, Q. Cao, H. Jiang, J. Yao, Y. Dong, and P. Yang. BCW: Buffer-controlled writes to HDDs for SSD-HDD hybrid storage server. In *Proceedings of the 18th USENIX Conference on File and Storage Technologies (FAST'20)*, pages 253–266, 2020.

[47] J. Wires, S. Ingram, Z. Drudi, N. J. A. Harvey, and A. Warfield. Characterizing storage workloads with counter stacks. In *Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI'14)*, pages 335–349, 2014.

[48] E. Xu, M. Zheng, F. Qin, Y. Xu, and J. Wu. Lessons and actions: What we learned from 10k SSD-related storage system failures. In *Proceedings of the 2019 USENIX Annual Technical Conference (USENIX ATC'19)*, pages 961–976, 2019.

[49] G. Yadgar, M. Gabel, S. Jaffer, and B. Schroeder. SSD-based workload characteristics and their performance implications. *ACM Transactions on Storage*, 17(1):1–26, 2021.

[50] J. Yang, S. Pei, and Q. Yang. WARCIP: Write amplification reduction by clustering I/O pages. In *Proceedings of the 12th ACM International Systems and Storage Conference (SYSTOR'19)*, pages 155–166, 2019.

[51] J. Yang, Y. Yue, and K. V. Rashmi. A large scale analysis of hundreds of in-memory cache clusters at Twitter. In *Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI'20)*, pages 191–208, 2020.

[52] Y. Zhang, P. Huang, K. Zhou, H. Wang, J. Hu, Y. Ji, and B. Cheng. OSCA: An online-model based cache allocation scheme in cloud block storage systems. In *Proceedings of USENIX Annual Technical Conference (USENIX ATC'20)*, pages 785–798, 2020.

[53] Y. Zhang, H. Li, S. Liu, J. Xu, and G. Xue. PBS: An efficient erasure-coded block storage system based on speculative partial writes. *ACM Transactions on Storage*, 16(1):1–25, 2020.

[54] D. Zhou, W. Pan, W. Wang, and T. Xie. I/O characteristics of smartphone applications and their implications for eMMC design. In *Proceedings of the 2015 IEEE International Symposium on Workload Characterization (IISWC'15)*, pages 12–21, 2015.