IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 16, 2017, accepted April 25, 2017, date of publication May 8, 2017, date of current version June 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2701839

Analyzing I/O Request Characteristics of a Mobile Messenger and Benchmark Framework for Serviceable Cold Storage

JAEMYOUN LEE[1], CHANG SONG[2], CHANYOUNG PARK[1], SOYEUN KIM[3], AND KYUNGTAE KANG[1], (Member, IEEE)

[1]Department of Computer Science and Engineering, Hanyang University, Ansan 15588, South Korea
[2]NAVER Labs Corporation, Seongnam 13494, South Korea
[3]School of Business Management, Hongik University, Sejong 30016, South Korea

Corresponding author: Kyungtae Kang (ktkang@hanyang.ac.kr)

This work was supported in part by the NAVER Labs, in part by LINE Corporation, and in part by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (Ministry of Science) (2014-0-00065, Resilient Cyber-Physical Systems Research).

ABSTRACT Cloud computing systems require massive storage infrastructures, which have significant implications on power bills, carbon emissions, and the logistics of data centers. Various proprietary ‘‘cold storage’’ services, based on spun-down disks or tapes, offer reduced tariffs but also lead to extended times to first access. One way to improve cold-storage systems is to build them on a file system that allows for the I/O patterns of storage devices. We have developed a cold-storage test-bed for mobile messenger services, which takes into account the power consumption of each hard disk in the system. We analyzed a trace of I/O requests from a messenger service and found that they had a strongly skewed Zipfian distribution and that most of the stored data is cold. Current cloud benchmarking tools cannot reproduce this pattern of I/O. Therefore, we have developed a tool for benchmarking cold-storage systems that emulates this type of long-tail distribution and can contribute to reducing the power consumption of mobile messenger services.

INDEX TERMS Cold storage, large-scale storage systems, energy-efficiency, statistical analysis, benchmark.

I. INTRODUCTION

Mobile messenger services are widely considered to be the future of social networking. They facilitate real-time personal conversations, and offer a rich environment for media sharing, entertainment, and even commerce. Recent mobile messenger services are superior to traditional messenger services that allow little more than the exchange of typed text, but require complicated infrastructures that use a significant amount of computing power. The increase in the power requirements of these services is unsustainable.

Data centers with massive storage infrastructures for the provision of mobile messenger services use more than 150 million kilowatt hours of energy (equivalent to the power consumption of approximately 13,000 homes); their power consumption is expected to double every year [1], [2]. Large-scale storage systems can consume 25 % of the total power used by data centers. The power consumed by most storage servers is independent of the amount of computing that they carry out. These considerations motivate researchers to improve the energy efficiency of large-scale storage systems.

Certain service providers with massive data centers have attempted to redesign their storage facilities as cold storage systems. Such storage systems reduce power consumption, but cannot immediately access the stored data as near-line systems. Although near-line storage provides total cost of ownership (TCO) benefits, it is only suitable in massive data centers. Service providers who do not have massive data centers would prefer a serviceable cold storage. This would provide the benefits of cold storage while ensuring that the stored data are immediately available. In a serviceable cold storage system, it is important to be power-proportional, meaning that the power consumption should be in step with data-intensive services. To this end, we must clarify the I/O patterns of storage systems that support data-intensive services, and design an accurate and fair benchmark.

The explosive growth of mobile devices such as smart phones and tablets has resulted in a change in the I/O patterns of large-scale storage systems for modern messenger services. Therefore, a thorough analysis of the I/O scenario of current messenger services is needed. We investigated the I/O patterns of image storage servers for the LINE mobile messenger service, run by the LINE Corporation. LINE announced that they had more than 217 million monthly active users worldwide in 2016 [3]. We traced the real I/O requests from the LINE storage servers for seven days. There were billions of logs in the trace data, and we believe that these are sufficient to simulate large-scale storage systems for a messenger service. We conducted various experiments to understand the I/O requests performed in the LINE mobile messenger service [4], [5]. In the experiments, it was found that there was a vast difference between the I/O requests generated by modern messenger services and those generated by other mobile services. The modern messenger service has a strongly skewed Zipfian distribution, and its data confidentiality requirements mean that a cold storage is ideally suited to managing the I/O request distribution.

Originally, we planned to improve serviceable cold storage systems, which are built on a file system that allows for the I/O requests of data-intensive services. However, research on large-scale storage systems is hampered by their cost [6]. Thus, we developed a large-scale storage testbed based on the Open Compute Project (OCP) Cold Storage [7]. This testbed is equipped with power monitors to measure the power consumption of each hard disk, and utilizes spin-down technology to reduce the power consumption. Additionally, current cloud benchmarking tools cannot reproduce the aforementioned I/O patterns [8]. We present a benchmark framework for large-scale object storage systems that facilitates comparisons of the latency and throughput performance of storage systems. A key design goal is sufficient extensibility so that the framework can accommodate commercial workloads. Hence, we extended the Yahoo! Cloud Serving Benchmark (YCSB) framework to interact with a Ceph cluster.

Finally, we implemented the specified workload by imitating the modern messenger service, and used a strongly skewed latest generator (SSLG) to simulate image storage servers. The workload and SSLG were executed on the developed testbed, and the results of read and insert operations were recorded. The contributions of this study are freely available in the vein of open-source projects.[1] We expect these projects to form guidelines for developing serviceable cold storage and reducing power consumption in mobile messenger systems.

The remainder of this paper is organized as follows. In Section II, we introduce the background of this study in terms of OCP Cold Storage, the spin-down technique, and related work. We evaluate the I/O request features of the mobile messenger service in Section III. In Section IV, we describe the spin-down technology and present a power monitor for hard disks. We discuss the implementation of the benchmark framework to compare large-scale storage systems in Section V. Finally, we summarize our findings and conclude this paper in Section VI.

II. BACKGROUND

A. COLD STORAGE

The OCP [9] was founded with the objective of replicating the concepts underlying open source software to create an *open hardware* movement through which commodity systems could be built for hyper-scale data centers. The OCP aims to share more efficient server and data center designs with the general information technology industry; consequently, it has published specifications for various storage servers. In addition, the OCP has proposed Cold Storage, a revised version of an OCP storage server, to satisfy the storage requirements of cold data. Cold Storage is designed to improve the energy efficiency of data centers by exploiting the spin-down technique [10]. It consists of 30 hard disks in two trays. A rack contains 16 Cold Storage units; thus, one rack contains 480 hard disks. Only one of the 15 hard disks in a Cold Storage tray can spin up at any given time; the others spin down to conserve power. Because only two hard disks are in operation at any given time, less power is required because most of the hard disks are spun down.

Although the spin-down technique can significantly reduce the power consumption of data centers, hard disks can only spin down a certain number of times. Thus, a suitable I/O scheduling methodology is required. Furthermore, pathological workloads can completely negate the power-saving benefits of the spin-down technique, causing disks to exceed their duty cycle rating and significantly increasing the aggregate spin-up latency [11]. Although the OCP has published hardware specifications for Cold Storage, its file system specifications have not yet been published. Other open source tiered file systems remain elusive, because file systems are commercially valuable. Thus, established policies for file systems that consider the overheads of the spin-down technique are expected to play an increasingly important role in the future.

B. SPIN-DOWN TECHNIQUE

The spin-down technique, which sets a hard disk into low-power mode while it is *idle*, is used to reduce power consumption. In low-power modes such as *standby*, the spindle motor does not spin and the hard disk head is parked; thus, the power consumption is reduced. Researchers have proposed several spin-down algorithms that can efficiently reduce hard disk power consumption [11]–[14]. Typically, these algorithms are time-out driven, i.e., a hard disk spins down if a time-out occurs before a request is received.

The spin-down technique in the Power Management feature of Advanced Technology Attachment (ATA)/ATA Packet

[1] `https://github.com/jaemyoun/IoDM` and `https://github.com/jaemyoun/YCSB/tree/dev-zipfian-strong-latest`

Interface (ATAPI) Command Set 2 (ACS-2) [15] allows a hard disk to save energy by changing the hard disk state from *active* to *idle*, *standby*, or *sleep*. In the *idle* state, the operations that can be performed are restricted. However, in the *idle* state, the spindle motor of the hard disk continues to spin, and the disk head remains on the platters. Consequently, only a small amount of hard disk power is conserved. In the *standby* state, the spindle motor of the hard disk is spun down and the disk head is parked. Because the spindle motor is not in operation, the hard disk is not able to access data. Naturally, only a few operations can be performed, but the power consumption is reduced significantly. Hard disks typically consume 5–10 times more energy while in the *active* state than in the *standby* state [16]. The *sleep* state is similar to the *standby* state, but only allows a reset operation to be performed, i.e., hard or soft reset. Thus, in theory, the *sleep* state consumes the least amount of power.

Recently, the *idle* state was combined with the *active* state to create the Advanced Power Management feature [17]. This feature allows the hard disk to change its state to either *active* or *idle* automatically. To enter the *standby* and *sleep* states, a special command must be input manually or a time-out must be configured [15]. The hard disk returns from the *standby* or *sleep* state to the *active* state when read or write operations occur. The actual design and implementation of power management features are at the discretion of the hard disk manufacturers.

ACS-2 has been revised to include an Extended Power Conditions (EPC) feature to standardize fine-grained power management controls [18]. The EPC feature set provides a hard disk with additional methods to control the power condition of a hard disk. These methods define some power conditions within the power management feature set—*idle_a*, *idle_b*, *idle_c*, *standby_y*, and *standby_z*—in order of decreasing power requirements. Additionally, the hard disk can translate a power condition immediately, and any of the power condition timers can be initialized so that the device waits for a period of inactivity before transitioning to a specified power condition.

C. RELATED WORK

Albrecht *et al.* [19] conducted an experiment involving thousands of Google users, applications, and services such as content indexing, advertisement serving, Gmail, video processing, and MapReduce jobs owned by individual users. Large applications may include many component jobs. The workload characteristics of jobs in data centers vary considerably among users and jobs. Consequently, the mean read age of the bytes over 15,000 jobs was approximately 30 days, even though jobs accessed very young (one-min-old) to very old (one-year-old) data. Another experiment showed that 50 % of the data stored by a particular user is less than one week old, but this corresponds to more than 90 % of the read activity.

Parikh [20] discussed the necessity for Cold Storage in Facebook's data centers. He argued that 2.8 ZB of data were created in 2012, and that 40 ZB of data would exist globally by 2020. To store these data, data centers would require billions of hard disks, each having the current maximum capacity of 4 TB. Hard disks consume a significant amount of power—a single Facebook data center consumed 153 million kilowatt hours in 2012, roughly equal to the power consumed by 13,000 homes [1]. Data such as photos are hot when they are created, but decrease in relevance over time, becoming warm. Eventually, such data become cold, and reads for such data are rarely requested. More specifically, 82 % of the read traffic services only 8 % of the young photos in Facebook's data centers. These results indicate that the demand for cold data storage such as legal data or backups of data is continuously increasing. Consequently, a tiered system that separates hot, warm, and cold data has been proposed. Furthermore, older data are more likely to become cold data.

Finally, Thereska *et al.* [21] proposed power-proportional distributed storage systems. These allow a large fraction of servers to be powered down during troughs without migrating data or extra capacity requirements, and address the challenges of maintaining read and write availability, performance, consistency, and fault tolerance for general I/O workloads. They discussed live traces from real services (Hotmail, Messenger) on their storage systems. The I/O requests of traditional messenger services differ significantly from those of modern messenger services. The demand for exchanging photos and videos via conversations is growing as smart phones with high-quality cameras become increasingly prevalent. Therefore, the image storage server trace data of the LINE application (a typical modern messenger service) must be analyzed to determine the workload characteristics of modern messenger services.

In our preliminary studies [4], [5], we characterized the power consumed by a 1 TB single hard disk with the aim of using of a cold storage effectively, and proposed an energy-efficient storage policy for mobile messenger services based on that characterization. We then presented a benchmark framework for large-scale object storage servers, which is intended toward facilitating performance comparisons of storage servers in terms of latency and throughput [7], [8]. However, additional experiments and statistical analyses are still required to gain insight into the characteristics of the mobile message services, which is essential in developing energy-efficient storage servers for those services.

III. ANALYSIS OF MOBILE MESSENGER I/O TRACE

We examined actual workloads generated by the LINE instant messenger service, which allows users to exchange text messages, pictures, videos, and audio data; users can also make Voice over Internet Protocol (VoIP) calls, and hold audio and video conferences without incurring charge. We traced requests for the I/O of images arriving at all LINE servers over seven days. This is a large-scale storage system workload—LINE has 20,000 servers, including more than 10 Redis clusters and 10 HBase clusters [22], and the trace of seven days data contains billions of lines. We used various mathematical

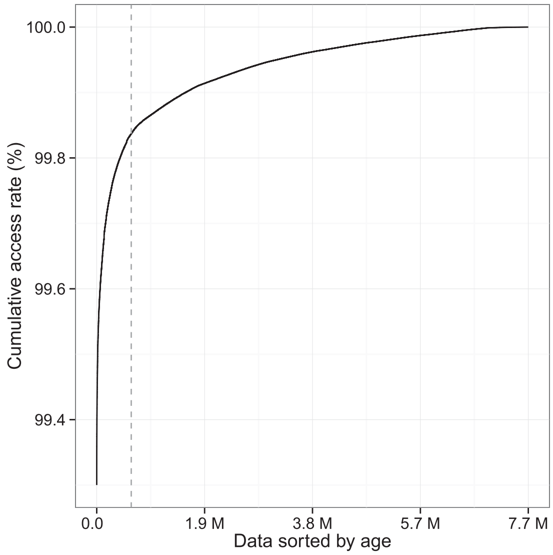


FIGURE 1. Cumulative distribution of read and write operations sorted by the age of the data. The data on the left are younger than those on the right. The vertical dotted line indicates that the youngest 8 % of the data represents 99.83 % of the traffic.

tools such as MATLAB and R, but encountered memory overflow problems and non-deterministic polynomial complete (NP-complete) problems. Therefore, we employed MySQL, an open-source relational database management system, to perform logical and arithmetic operations. MySQL proved capable of handling this large workload with a reasonable transaction processing speed.

A. ACCESS RATE

First, we sorted the workload by file age and calculated the access rate. Figure 1 shows that the youngest 8 % of requests make up 99.83 % of I/O traffic (the horizontal axis represents the number of files sorted by age). The remaining 92 % of older data are unlikely to be read. The younger data with a high probability of being read are called hot data; as the probability decreases over time, these become cold data. The network infrastructure of a mobile messenger service is optimized to serve the 8 % of hot data. Large-scale storage systems for the remaining 92 % serve relatively few read operations. These storage systems consist of several disks per system, making storage one of the biggest sources of power consumption.

If the cold data were migrated to power-proportional storage servers, we would expect a considerable amount of energy to be saved for a very small reduction in performance, manifested as a delay when there is a request for data stored on the hard disk, which is in the *standby* state. Our traces record I/O requests arriving at the system, rather than individual disks, which take no account of caching. However, large-scale storage systems have cache sub-systems, and mobile messenger applications also have cache space in customer devices. These caches enable power-proportional storage systems to avoid routing I/O requests to hard disks that are in the *standby* mode, and the scope for energy saving increases with the size of the cache.

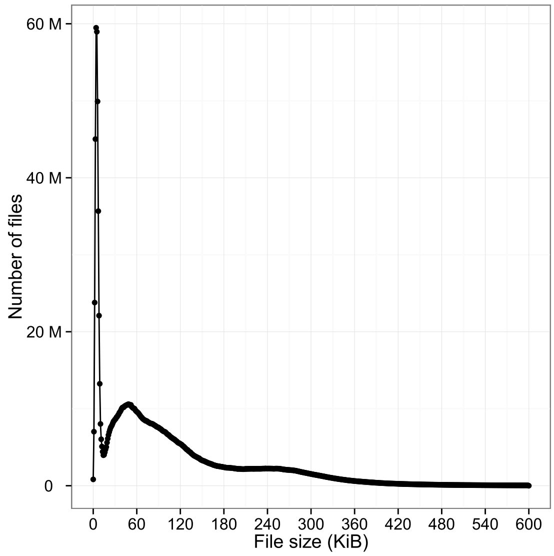


FIGURE 2. Distribution of requested files by size. Files with sizes above 600 KB are excluded, as these were found to be requested infrequently: four times at most.

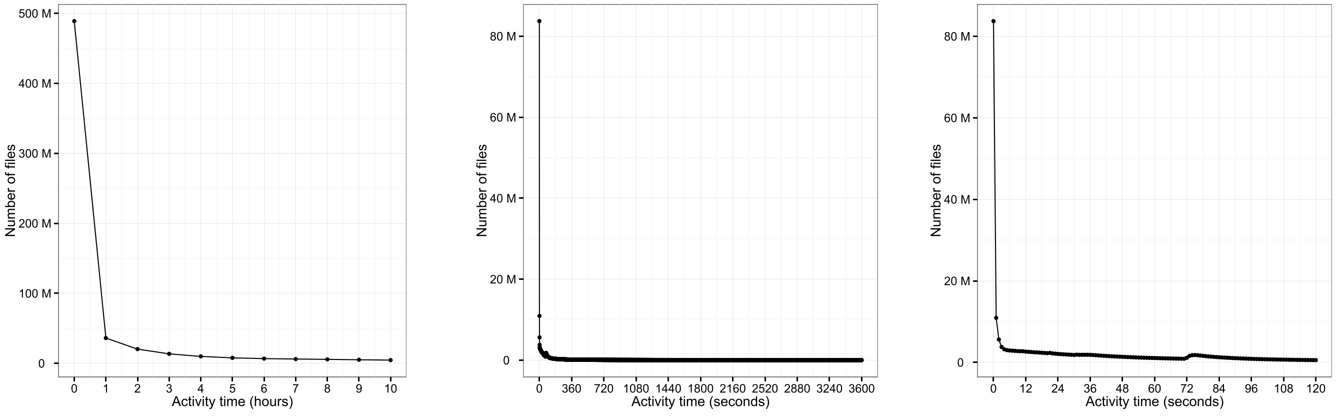
B. ADDITIONAL I/O PATTERNS

We conducted additional experiments to characterize the cold data. We sorted the data by file size and grouped files of the same size in increments of 1 KB. The requests for files in each group were counted. The results are shown in Figure 2. The majority of requests occurred in the range of 4–6 KB, which corresponds to the size of a profile thumbnail image in LINE, and we would expect most of the files in this group to be thumbnails. The I/O patterns suggest that access to the files in this group is largely independent of file age, as longstanding thumbnails are frequently requested. Therefore, it is preferable that the old files in this group are not classified as cold data.

Files with sizes in the range of 40–50 KB are the second-most-requested group, and most of these files are regular image messages. Even if users send images with larger file sizes, the mobile messenger application compresses the files into this range before sending them to the storage servers. The I/O pattern for this class of files suggests that the number of requests corresponds to the number of people in an average chat room. Therefore, these files are best treated as cold data. Files that are larger than 200 KB usually contain video or voice messages, and it is preferable to consider these as cold data.

Even though the file sizes corresponding to these classes vary across mobile messenger applications, all modern services allow users to exchange thumbnail images, image messages, and video messages. Therefore, we would expect similar I/O patterns to be observed in all modern mobile messenger services.

Previous studies have focused on file age and classified files by frequency of usage. This method of identifying cold data is plausible and yields good results. However, Figure 1



(a) (b) (c)

FIGURE 3. Illustration of the number of files remaining live. The number of files that are accessed later is very small, and thus these files are of little interest. (a) At one-hour intervals, up to 10 hours. (b) At one-second intervals, up to one hour. (c) At one-second intervals, up to two minutes.

shows that the threshold used to classify data as cold needs to be appropriate. Therefore, we looked at the length of time for which files remained 'live' (i.e., they were being actively accessed).

Initially, we performed this analysis for periods of one hour, with the results shown in Figure 3(a). Nearly 64 % of all files remain live for an hour or less, and another 4 % are accessed for between 1 and 2 h. Thus, 1 h is a plausible threshold for classifying files as cold. We repeated this experiment, looking at intervals of 1 s over the first hour, with the results shown in Figure 3(b). The first 2 minutes of this graph are magnified in Figure 3(c). The life of many files is less than 3 s, and half of all files are finally accessed within 73 s. These results suggest that files older than 2 minutes have a high probability of being cold data.

However, it is impossible to predict accurately which files will be accessed within 2 minutes of being written; if this prediction is wrong, hard disks that have been spun down must be spun up again, reducing the performance and increasing power consumption. Therefore, determining the threshold based solely on time is not a suitable approach for mobile messenger services.

To establish a more efficient policy of determining cold data, we compared mobile messenger services with other services such as the Web, e-mail, and File Transfer Protocol (FTP). These other services are usually provided to an unspecified number of people. Although the recipients of an e-mail can be limited, the e-mail can be forwarded, whereas mobile messenger services do not allow forwarding because of fundamental privacy concerns. Thus, conversations are shared by a known group of users, and confidentiality is more likely to be preserved.

Figure 4 shows that 31.62 % of files are requested once and 49.24 % are requested twice. This suggests that most files are sent to a single recipient, with the first operation being the write request, required to upload the image from the sender and the second being the read operation to download the image to the recipient's device. Obviously, there is no need to download the image to the sender's device, and the recipient only downloads the image once because it is then stored in the application cache area.

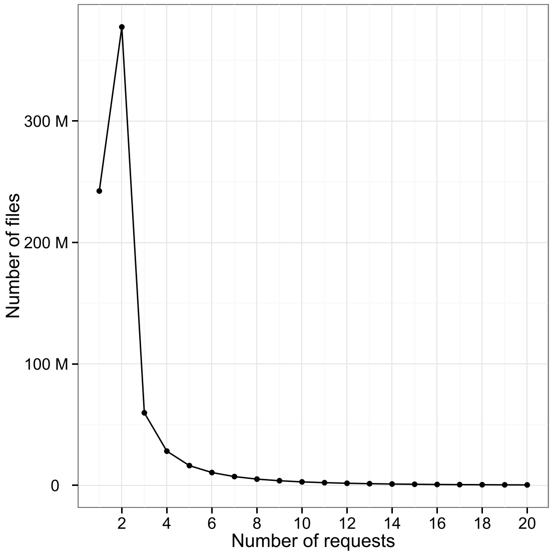


FIGURE 4. Illustration of number of files plotted against number of request for each file.

Our approach toward categorizing data as cold can be made more discernible if we look at the number of recipients, which is known. When the number of times that a file has been accessed equals the number of recipients, further downloads are unlikely; thus, we can label that file as cold data and migrate it to power-proportional storage servers. Even if a file has not been accessed by all its recipients, we must at some time assume that the remaining recipients will not access it, or are unlikely to do so for some time. For example, files that have not been accessed for an hour can be labeled as cold data and moved into power-proportional storage servers.

Files that are less than 73 s old are considered hot data because they have not yet been read. Files that have been requested more than twice have a high probability of being

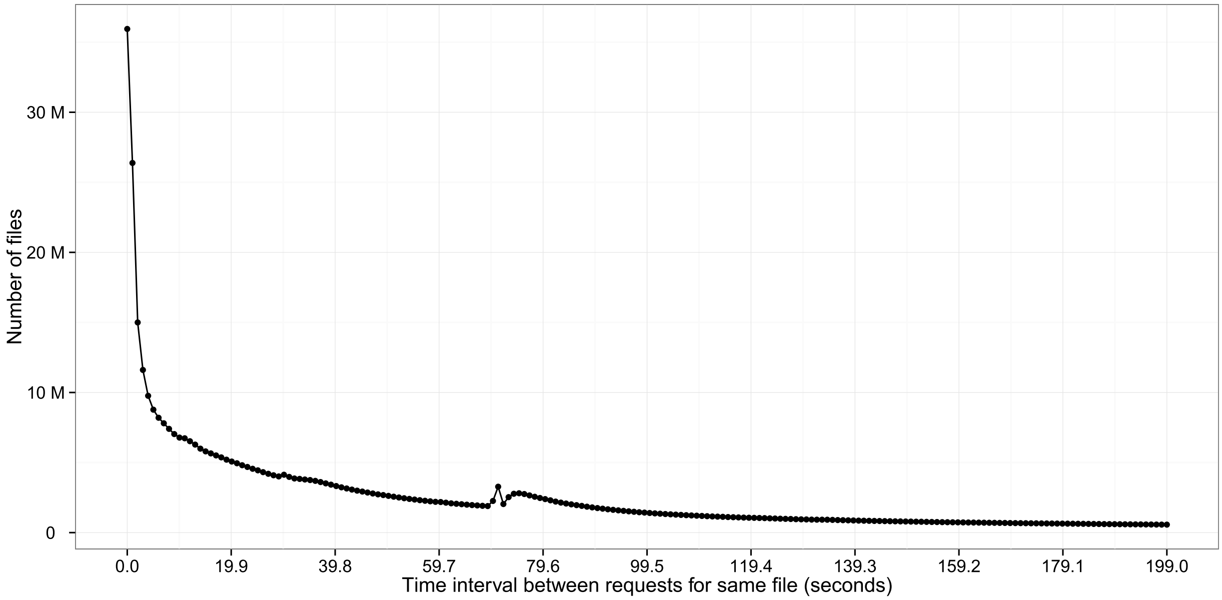


FIGURE 5. Illustration of I/O requests time intervals from 0 to 200 s.

TABLE 1. Descriptive statistics.

Number of Data Counts	Mean	Standard Deviation	Skewness	Kurtosis
1,072,418,419	202.6231	703.26248	6.067	45.745

cold data (as most chat rooms have only two participants). If there are no requests for some time, the file is also classed as cold data. To validate the time threshold for classifying a file as cold data, we performed an additional experiment to examine the distribution of time intervals between requests for the same file, as described in Figure 5. The requests identified write requests with read requests. In other words, the time intervals of the files that are read once are the intervals between a write and read request. Other time intervals where the files are read several times are between read requests. In addition, the distribution excluded files that were only requested once. Most requests occurred at intervals of less than 75 s. In particular, 9.2 % of requests occurred at intervals of less than or equal to 5 s. Considering that the maximum time interval is seven days, the distribution is strongly skewed.

C. STATISTICAL ANALYSIS

We performed a few statistical analyses to show how much the distribution of a time interval between mobile messenger requests was skewed and heavy-tailed. We calculated the basic descriptive statistics and as shown in Table 1 high skewness (6.067) and kurtosis (45.745) imply that the empirical distribution is right-skewed and heavy-tailed.

The log-log plot of the 'number of files divided by the total', which can be regarded as the relative frequency, and the 'time interval' also supports our findings as shown in Figure 6. A log-log plot is a useful way of recognizing whether two variables X and Y have the relationship $Y = aX^k$ (power law). The relationship can be rewritten as $\log(Y) = k\log(X) + \log(a)$ when taking the logarithm and this will exhibit a straight line if X and Y follow a power law. We exclude 0.5 % of the data at the very end of the tail since beyond that point the data is too sparse. The plot exhibits almost a straight line, which implies that the distribution closely follows a power law, whose distribution is of the form $f(x) = ax^k$, and Power law distributions are heavy-tailed.

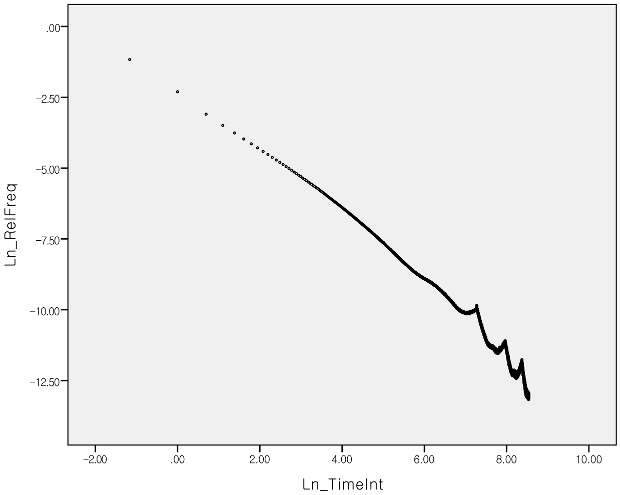


FIGURE 6. Log-Log plot (99.5 % included or Time $\leqq$ 5100).

We further estimated the parameters k and $\log(a)$ from the log-log plot by letting k be the slope and $\log(a)$ be the intercept ($k = -1.431$, $\log(a) = -0.289$) as presented

TABLE 2. Unstandardized coefficients.

Model	B	Standard Error	t	Significance
(Constant)	-0.289	0.032	-9.080	0.000
$\ln(Timeint)$	-1.431	0.004	-341.932	0.000

in Table 2. The fitted distribution of a time interval between mobile messenger requests is

$$f(x) = (e^{-0.289})x^{-1.431} = 0.749x^{-1.431}.$$

D. EVALUATION

Our results show that only a very small amount of data needs to be stored in hot storage servers, because much of the data are cold. The access rate of young data is extremely high in a mobile messenger service. Parikh [20] indicated that 8 % of data constitutes 82 % of the traffic; however, in a mobile messenger workload, 8 % of the data constitutes 99.8 % of the traffic.

We believe that mobile messenger services have a higher proportion of cold data because the associated chat rooms contain a fixed number of users. The number of requests for an image tends to be equal to the number of people in the chat room. An image is sent to all recipients from the application cache, and is rarely sent again. Consequently, the access rate decreases sharply for older files. Hence, a mobile messenger workload has a high proportion of extremely cold data. This is a characteristic of modern mobile messenger services, and the same I/O pattern is not observed in traditional messenger services [21].

To reduce the power consumption of storage servers, they need to be operated according to a policy that reflects this distinctive workload. Previous studies have investigated energy efficiency and low-level dynamic power management, but they focus on data age and the access rate of workloads without considering the number of requests for each data item, even though the number of requests is a more important factor in the mobile messenger workload. These facts indicate that the operation of a power-proportional storage system in a mobile messenger service should be customized to consider data age as well as the number of requests. We propose the following policy:

1) Record the number of accesses to a file and compare it with the number of recipients.
2) If the number of times the file is downloaded is greater than or equal to the number of recipients, it should be migrated into power-proportional storage servers to reduce power consumption.
3) If the file remains in hot storage for too long, it should be migrated into power-proportional storage servers. The threshold time should be set appropriately depending on the workload, but one hour is expected to be a suitable period.

This policy can serve as a guideline to researchers in the field of energy-efficient large-scale storage systems, while providing a structured exposition and discussion of the current low-power hard disk technology and the modern mobile messenger workload.

IV. HARD DISK FEATURES

Research related to the spin-down technique [11], [23], [24] has been predominantly focused on the average power consumption, with no regard for the instantaneous power consumption. In product manuals, hard disk manufacturers specify the power consumption of each hard disk state and the transition periods between states. However, the specified values are averages and maxima, and there is no detailed power consumption information. However, for research on cold storage, the measurement results must be sophisticated, detailed, and precise.

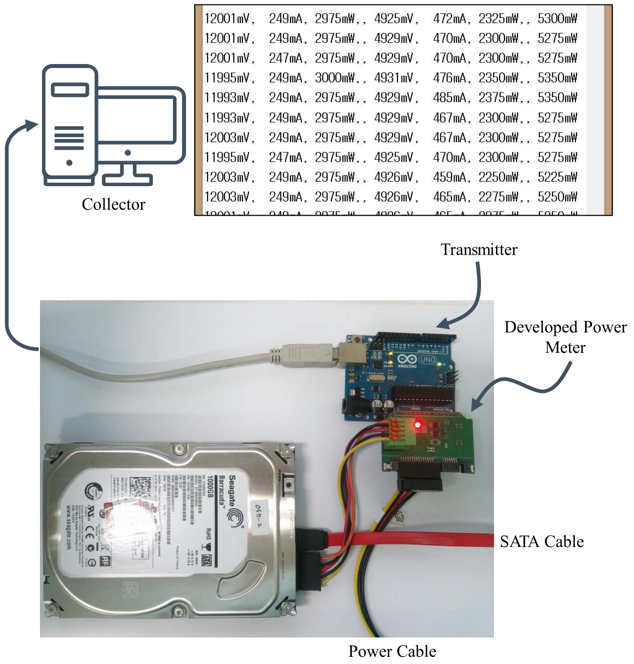
In a series of experiments, we observed that hard disk power consumption increases steeply then flattens out when the hard disk state changes from *standby* to *active*. The peak power consumption is five times greater than the average power consumption. Thus, the instantaneous power consumption must be considered when designing a spin-down scheduler. Additionally, the state transition period is less than a specified average value when there is no load on the hard disk. Most experiments associated with hard disks assume that there are many requests and some data blocks are stored in a cache. Specifically, the hard disk has a heavy load. However, hard disks associated with cold storage are almost never read, so that the load is light and the state transition period is shorter than the average.

A. DEVELOPMENT OF POWER MONITOR AND ANALYSIS OF HARD DISKS

To measure the hard disk power consumption effectively, we developed a power monitor comprising an electric current shunt and power monitors. As a 3.5 inch hard disk is supplied with both 12 V and 5 V DC, both voltages and currents must be measured for the precise determination of hard disk power consumption. Current power monitors cannot measure the power consumption of each hard disk individually in terms of cost, effectiveness, and functionality. The power monitor that we developed is cheap, faithful to a fundamental function, and able to measure power consumption in a number of hard disks.

Figure 7 shows the environment for measuring the power consumption of hard disks. Power is provided through the power monitor, which gathers the voltage and current consumed by the disk. These values are transmitted to a collector by an Arduino Uno microcontroller. As mentioned above, the power monitor measures the current drawn by both 12 V and 5 V supplies to a hard disk using Texas Instruments INA226 power monitors [25]. After gathering this information, the voltage, current, and total power are transmitted to the collector.

The INA226 monitors both the shunt voltage drop and the bus supply voltage. The programmable calibration value, conversion times, and averaging, combined with an internal



```
12001mV,  249mA, 2975mW,, 4925mV,  472mA, 2325mW,, 5300mW
12001mV,  249mA, 2975mW,, 4929mV,  470mA, 2300mW,, 5275mW
12001mV,  247mA, 2975mW,, 4929mV,  470mA, 2300mW,, 5275mW
11995mV,  249mA, 3000mW,, 4931mV,  476mA, 2350mW,, 5350mW
11993mV,  249mA, 2975mW,, 4929mV,  485mA, 2375mW,, 5350mW
11993mV,  249mA, 2975mW,, 4929mV,  467mA, 2300mW,, 5275mW
12003mV,  249mA, 2975mW,, 4929mV,  467mA, 2300mW,, 5275mW
11995mV,  247mA, 2975mW,, 4925mV,  470mA, 2300mW,, 5275mW
12003mV,  249mA, 2975mW,, 4926mV,  459mA, 2250mW,, 5225mW
12003mV,  249mA, 2975mW,, 4926mV,  465mA, 2275mW,, 5250mW
```

FIGURE 7. Developed tools for hard disk power consumption measurement.

TABLE 3. Hard disks used in the experiments.

Manufacturer	Model	Capacity (no. of platters)	Feature	EPC
Seagate	ST1000DM003	1 TB (1)	Basic	X
Seagate	ST4000DM000	4 TB (4)	Basic	X
Seagate	ST6000AS002	6 TB (6)	SMR	O
Seagate	ST10000DM0004	10 TB (7)	Helium-filled	O

multiplier, enable direct readouts of the current in amperes and the power in watts. The INA226 detects on-bus currents of 0–36V V, while the device obtains its power from a single 2.7–5.5 V supply, typically drawing 330 μA of supply current. The INA226 has an operating temperature range of 40–125 °C. The I^{2}C interface features 16 programmable addresses.

Various hard disks must be analyzed, because different products have particular features depending on their purpose. The hard disk products are divided into three main categories: basic, mobile, and enterprise. The mobile category is inadequate for cold storage, and products in the basic category are cheap. Enterprise products have high-density, high-capacity, and advanced power management features.

We used four hard disks in the experiments, as described in Table 3. Each hard disk manufacturer implements a disk controller with specific algorithms and policies. Thus, the experimental hard disks were taken from the same manufacturer. The hard disks were connected by a Serial ATA (SATA) 6 GB/s interface, and the rotation speed was 7200 RPM.

The first hard disk with a capacity of 1 TB is a basic product that is suitable for most customers. The disk is inexpensive, and the performance is appropriate for most workloads. The second hard disk is similar to the first, but has a larger capacity (4 TB). Its cost per unit capacity ($/GB) is less than that of the 1 TB hard disk, and the density is higher.

The third hard disk supports a shingled magnetic recording (SMR) technology. SMR is deployed to increase the areal density of hard disks by recording at a track pitch appropriate for an as-narrow-as-possible reader. Recording a sector at this track pitch with an as-wide-as-necessary writer means that neighboring sectors are affected. This overlapping has a significant impact on the data organization and behavior of data access. Specifically, rewriting a sector on a track that has been shingled over cannot occur without overwriting subsequent down-band tracks [26]. However, a cold storage assumes that there are no rewriting and appending requests, because hot storage will handle such tasks. Thus, an SMR hard disk with a high $/GB ratio is expected to be suitable for cold storage. SMR hard disks with capacities greater than 6 TB are available.

The fourth hard disk is a helium-filled enterprise-class storage device. The disk is filled with helium to reduce internal friction significantly. Helium is much thinner than air, and thus provides much less drag on the rapidly rotating roundels, which in turn drives down power consumption and increases reliability. Helium also permits seven platters to be inserted into a standard-height 3.5 inch hard disk, rather than the usual six, allowing for higher storage capacities [27]. Helium cannot directly increase the maximum areal density of the hard disk platters. However, helium-filled technology is expected to enable higher capacity drive technologies such as SMR and heat-assisted magnetic recording.

Some enterprise hard disks support the EPC feature of ACS-2. EPC enables various cold storage policies, and increases the efficiency of power consumption without significant performance overheads.

B. POWER CONSUMPTION OF VARIOUS HARD DISKS

To execute the spinning down of the hard disks, we used Hdparm and S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology) on Ubuntu. Hdparm is a command-line utility for Linux that sets and views ATA hard disk drive hardware parameters such as the drive caches, sleep mode, power management, acoustic management, and Direct Memory Access (DMA) settings [28]. We conducted experiments on both the *standby* and *sleep* states of the hard disks. However, we focused on hard disks in the *standby* state, because the return time from the *standby* state is slightly less than that from the *sleep* state but the power consumption trends are virtually indistinguishable. In addition, we measured the power consumption of flash memory-based Solid-State Drive (SSD). SSDs use integrated circuit assemblies as flash memory to store data without the spindle motor and head used by hard disks. Therefore, SSD power consumption has a different pattern from that of hard disks.

We measured the hard disk power consumption and state transition period for various scenarios: transitioning from the *standby* state to the *idle* state on the 1 TB and 4 TB hard disks; transitioning from the *idle_b* state to the *idle* state, from the *idle_c* state to the *idle* state, and from the *standby_z* state to the *idle* state on the 6 TB and 10 TB hard disks. The scenarios excluded the spin-down moments, because the power falls down without any singularity when a hard disk is spinning down. Additionally, the state transition period to spin down (less than 1 s) is negligible for cold storage, because the spin-down state is expected to last at least 12 h. The quality of service directly suffers when there is a high probability of a response delay from coincident hard disks being spun down. The command for reading a sector (512 bytes) was used as the spin-up command, and cache memories on the operating system and hard disk controller were flushed out continually. Figure 8 shows the results of 100 executions of the experiments, with the average, maximum, and minimum values represented by the solid line, red shadow, and blue shadow, respectively. Note that there may be differences according to the particular hard disk manufacturers.

Figures 8(a), 8(b), 8(e), and 8(h) depict the results for state transitions from the *standby* state. The four hard disks consume 590 mW, 750 mW, 710 mW, and 920 mW respectively. When the hard disks are transitioning to the *idle* state, the power consumption increases to 19.60 W for 3.54 s, 17.55 W for 7.4 s, 20.89 W for 10 s, and 12.61 W for 13.74 s, for the four hard disks, respectively. These results indicate that the number of platters affects the state transition spin-up period. Thus, a hard disk with several platters, such as a 10 TB hard disk, incurs a significant overhead when using the spin-down technology.

Although it has the longest state transition spin-up period, the 10 TB hard disk has the lowest maximum power consumption among the four hard disks. This shows that the reduction in friction reduces the power requirements. Eventually, the number of platters in a hard disk will increase the state transition period, which will increase the power consumption in transitioning states. However, helium gives a power advantage in the transitioning states for the same number of platters.

The EPC feature appears to be very promising in reducing the power consumption in cold storage. Figures 8(c) and 8(f) describe the state transition from the *idle_b* to the *idle* state. ACS-2 does not specify the *idle_b* state in detail, depending instead on the manufacturers. In general, the *idle_b* state is used to park the head of a hard disk, but the spindle motor remains spinning. Consequently, the *idle_b* state offers reduced power consumption of approximately 1000 mW over the *idle* state, but the transition period to the *idle* state takes approximately 300 ms. This fact encourages the establishment of a cold storage management policy without any overhead.

The *idle_c* state limits the rotation speed based on the *idle_b* state. This approach facilitates a saving of 3 mW, as depicted in Figures 8(d) and 8(g). The transition period to the *idle* state averages 3.7 s, representing a reduction of 6.3 s and 10.1 s, respectively, compared with the *standby* state. The difference in power consumption between the *idle_c* and *standby* states is 1.70 W for the 6 TB hard disk and 1.44 W for the 10 TB hard disk. Thus, if sufficient power is available and the transition period from the *standby* state is too slow, a strategy using the *idle_c* state is a viable alternative.

Unfortunately, we could not identify the difference between the *idle_a* and *idle* states in terms of the transition period, power consumption, and operations.

We found that, when spinning up, a hard disk consumes 5.8 times its average power for a few seconds. This power consumption is immense compared with that of other commands or other devices in a computer. If several hard disks were to spin up at the same time, the computer power supply would fail to meet the demand and would be unable to deliver a stable power supply to the disks as well as the processors. Moreover, this may result in hardware failure and electrical shorts. Therefore, large-scale storage systems must take all reasonable precautions to protect against simultaneous spin up.

We also measured the SSD power consumption. The SSD used in this experiment consumes less than 10 % of the power of a hard disk when waiting for commands. This reduced power consumption is below that when a hard disk is spun down. When the SSD is required to spin down, it carries out an action that we are not able to measure, although there is no spindle motor. However, the power consumed by the SSD after spinning down does not change. Thus, the spinning down on an SSD is a waste of electrical power and is counterproductive, because the amount of power consumed while the SSD is carrying out the spin-down command is five times the average. Accordingly, the SSD may be permitted to receive a spin-down command to ensure compatibility, though the spin-down technique negatively affects the energy efficiency of the SSD. SSD-based large-scale storage is one solution for cold storage, but this is more expensive that hard disk-based storage in terms of $/GB. Additionally, research on flash memory-based storage systems has focused on an all-flash storage system, which does not have tiers for hot and cold storage.

In conclusion, it is clear that the spin-down technique reduces the power consumed by large-scale storage servers. In particular, enterprise hard disks and the spin-down technology have no negative impact on the performance. The only trade-off is in terms of cost, particularly for helium-filled hard disks. In addition, the spin-down technique can significantly reduce the cost of both establishing and operating data centers. Data center cooling systems, which are a primary target for energy efficiency improvements, can be scaled back because the spin-down technique reduces the hard disk operating temperature.

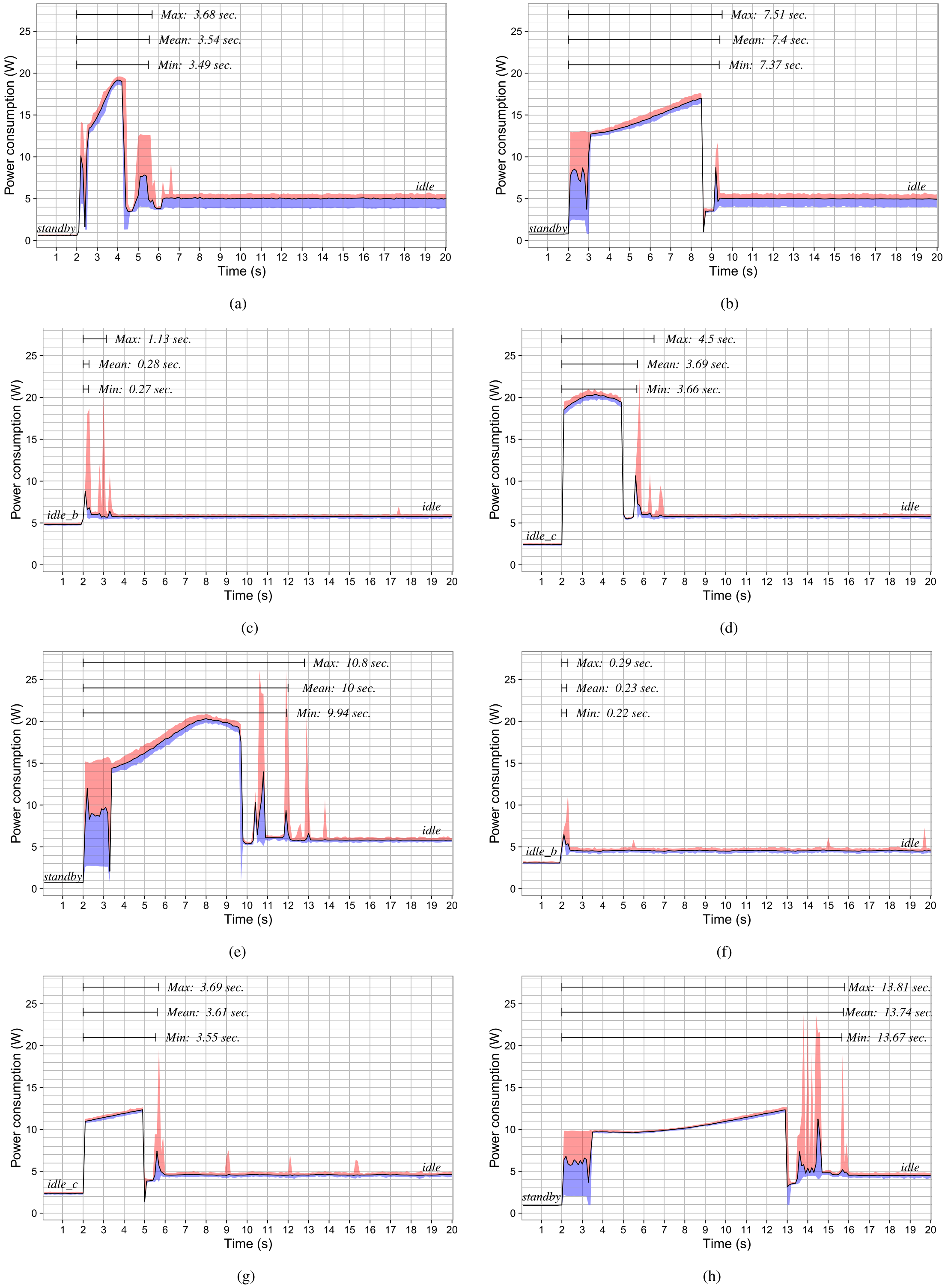


FIGURE 8. Power consumption and state transition period of each hard disks. (a) ***Standby*** mode (1 TB). (b) ***Standby*** mode (4 TB). (c) ***Idle_b*** mode (6 TB). (d) ***Idle_c*** mode (6 TB). (e) ***Standby_z*** mode (6 TB). (f) ***Idle_b*** mode (10 TB). (g) ***Idle_c*** mode (10 TB). (h) ***Standby_z*** mode (10 TB).

C. COLD STORAGE TEST-BED

Research on large-scale storage systems is hampered by their cost [6]. It is therefore desirable to develop a scalable and flexible testbed for evaluating the power consumption of large-scale storage systems. We built a small-scale cold storage testbed that consisteds of ten 4 TB hard disks,

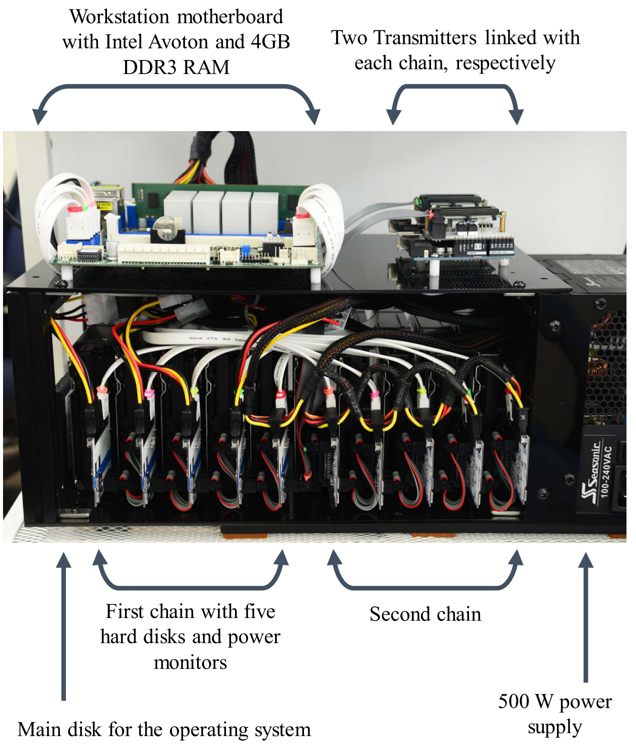

Workstation motherboard with Intel Avoton and 4GB DDR3 RAM
Two Transmitters linked with each chain, respectively
First chain with five hard disks and power monitors
Second chain
Main disk for the operating system
500 W power supply
Seasonic
100-240VAC

FIGURE 9. Our small-scale test-bed of a storage server with 30 hard disks and power monitors for the power consumed.

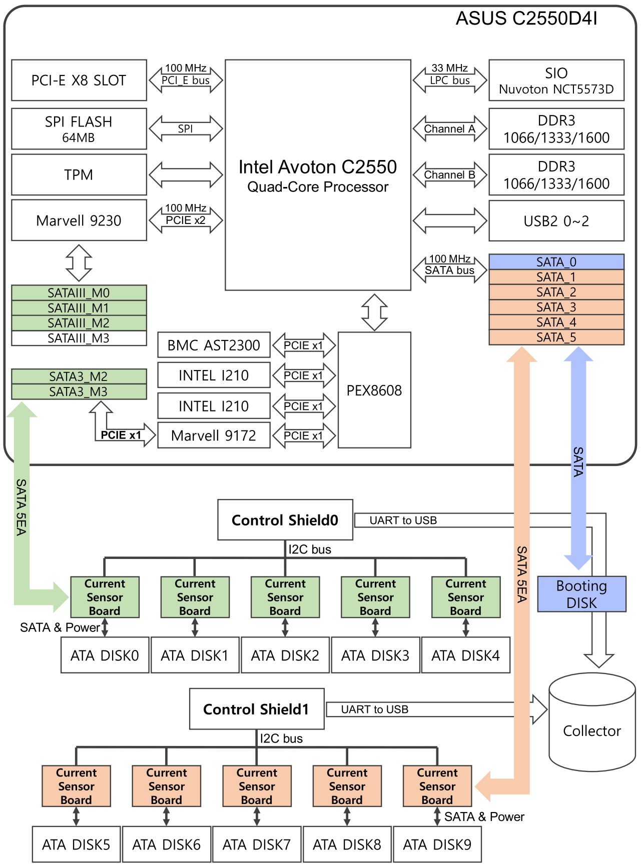


FIGURE 10. Layout of our scalable storage server.

a workstation motherboard with Intel Avoton processors, and a 500 W power supply, as shown in Figure 9.

The testbed uses 10 power monitors that are extended to link themselves as a chain, and two transmitters to gather the measurement values from the chained power monitors. The workstation motherboard has 12 SATA controllers, one as the main disk (SSD) for an operating system and 10 as storage disks for storing data. The remaining controller is reserved for future use. The power supply was selected after considering the request pattern into the hard disk. Although hot storage with intensive requests requires considerable power, cold storage has almost no requests after the hard disks are full. If a distributed file system is appropriate, no high-performance power supply is needed for cold storage. Figure 10 illustrates the testbed in detail.

Our testbed is a small-scale storage system based on the OCP Cold Storage server [10]. There are obviously differences between the testbed and a real storage system. However, it is sufficient for conducting experiments on energy-efficient storage systems with the spin-down technology. We expect the testbed to contribute to the assessment of power consumption in tiered storage systems.

V. BENCHMARKING STORAGE SERVERS

For research on energy-efficient storage systems, it is important to design an accurate and fair benchmark for scaling future computing systems in terms of the power consumption. The YCSB framework facilitates performance comparisons between new-generation cloud database systems. It is implemented as a standard benchmark and benchmarking framework to assist in the evaluation of different cloud systems. The YCSB framework consists of a workload-generating client (YCSB client), which can be used to load datasets and execute workloads across a variety of data serving systems, and a package of standard workloads (YCSB workload) to evaluate different aspects of the performance of cloud serving systems [29].

We extended YCSB to incorporate a benchmarking framework for distributed file systems used in a data center. YCSB is optimized for NoSQL database benchmarking, and many cloud systems are referred to as key-value stores for a highly available storage system [30]. As key-value stores are a simplistic yet powerful NoSQL model, YCSB is most closely related to NoSQL database benchmarks.

We implemented the YCSB client to interact between a YCSB core and a distributed file system, Ceph. The YCSB binding can be written in Java and reliable autonomic distributed object store (RADOS), which is a utility for interacting with a Ceph object storage cluster that has a Java library. RADOS allows various application program interfaces (APIs) to be used for synchronous and asynchronous requests, and consists of key-value stores to address the storage requirements of arbitrary data structures. Thus, YCSB and RADOS are complementary for cold storage research.

TABLE 4. Summary of Ceph object storage benchmarking results (The scan operation was not implemented).

Workload	Operations	Request distribution	Throughput (ops/sec)	Average latency (min, max) (µs)
Update heavy	Read: 50%, Update: 50%	Zipfian	58.123	Read: 4224 (2612, 13967), Update: 30052 (14544, 285183)
Read heavy	Read: 95%, Update: 5%	Zipfian	178.891	Read: 3952 (2102, 13895), Update: 40810(15552, 185215)
Read only	Read: 100%	Zipfian	244.141	Read: 3879 (2388, 13759)
Read latest	Read: 95%, Insert: 5%	Latest	194.250	Read: 3891 (2022, 11639), Insert: 23570 (7532, 46335)
Short ranges	Scan: 95%, Insert: 5%	Zipfian/Uniform	NaN	Scan: NaN, Insert: 20988 (6084, 65055)
Read & modify	Read: 50%, Modify: 50%	Zipfian	53.550	Read: 3999 (2140, 14183), Modify: 34226 (17280, 219391)

We examined the YCSB client for RADOS with our testbed, which comprised three storage servers and ten hard disks per server, and report the results of insert, read, update, and delete operations in the Ceph infrastructure testbed. A key design goal of our storage benchmark framework is extensibility, so that it can accommodate the requirements of arbitrary data structures in commercial storage servers.

A. IMPLEMENTATION OF RADOS-BINDING

The YCSB framework is a Java program that has an interface for representing standard CRUD operations: create, read, update, delete, and scan. YCSB is not fundamentally concerned with the file systems of cloud storage. However, it appears promising in terms of benchmarking storage that accesses objects through key-value stores.

We utilized Ceph as the object storage for cloud systems. Ceph is a massively scalable, open, and software-defined storage platform designed for cloud systems and web-scale object storage. Ceph supports object storage through RADOS, which has the ability to scale to thousands of hardware devices. Although Ceph allows the mounting of block-based storage through the RADOS Block Device (RBD), which is integrated in the Linux kernel, RBD is a utility for manipulating RBD images, which are stored as RADOS objects in the object storage daemons (OSDs). Used in conjunction with RADOS in Ceph, a controlled replication under scalable hashing (CRUSH) algorithm determines the method of data replication and mapping to the individual nodes.

Low-level access to the RADOS service is provided by *librados* in the form of a library. The *librados* API is written in C++, with additional bindings for Java, called *rados-java*. The *librados* API allows both synchronous and asynchronous interaction.

We selected APIs that are ideally suited for each operation as follows:

- **Insert:** Creates an object in a storage cluster. The operation must be synchronized with the storage systems before the system returns *success* to the user.
- **Read:** Reads an object. To improve the Ceph performance, the *read* operation uses asynchronous APIs atomically.
- **Update:** Updates an object by replacing data instead of appending it to a stored object. The implementation of an update operation constitutes the delete and insert operations. Although this implementation affects the performance of the update operation, we are only interested in image storage clusters for a mobile messenger service. Stored data are not updated because the image would already have been transmitted to others.
- **Delete:** Deletes an object.
- **Scan:** Executes a range scan by reading a specified number of records starting at a given record key. This operation is confined to a database, and an object storage without record keys cannot run a range scan. Thus, this operation was not implemented.

Table 4 shows the Ceph object benchmarking results for the insert, read, update, and delete operations. We executed the YCSB standard workloads on a testbed. The environment of the testbed consisted of a Ceph cluster composed of a manager node and three storage nodes. The manager node is in charge of a Ceph monitor daemon, metadata daemon, and gateway. The storage nodes with our three cold storage testbeds were the Ceph object storage daemons.

Although the result does not imply optimized Ceph performance because of the limitations of the small-scale testbed, the YCSB client for RADOS operated successfully with a YCSB core and workloads. We created a pull request to review and merge our contributions in the YCSB official open source repository, and it merged completely at version 0.10 [31]. We hope to extend the YCSB client for RADOS to evaluate storage systems more effectively.

B. STRONG SKEWED WORKLOAD

YCSB currently serves six workloads. Workload A represents a heavy update situation, which has 50 % reads and 50 % updates. The throughput of read operations is about 4.2 ms, which, as mentioned above, is greater than the throughput of update operations. Workload B concerns heavy read operations, and workload C has only read operations. These two workloads assume that the read operations are intensive. The testbed shows almost consistent throughput of read operations in workloads A–C. Workload D requires the latest objects to be read and a few new objects to be inserted. There is no variation in the throughput of read operations among all workloads. The throughput of insert operations is six times that of the read operations. This is because the size of the inserted objects is too small (approximately 1 KB). Workload E is suited to scan operations, hence we did not implement it, and Workload F consists of read operations and read-modify-write operations, which simultaneously read an object and write the modified object. These operations are important in database systems, but are unimportant for our testbed. In particular, read-modify-write operations are not evaluated in the testbed because we assume that there is no modification to the cold storage.

We conducted experiments on all workloads that are appropriate for cold storage. The object size of images in mobile messenger services is approximately 5–50 KB, as mentioned above; thus, the object size specified by the YCSB workloads is too small. Reading the latest objects in workload D appears to be similar to the access distribution of cold data, but cold storage also involves heavier insert operations than those available in Workload D. Large-scale benchmarking generation operations often take longer than the evaluation. The Zipfian distribution requires large datasets to be generated using parallel algorithms and executions. The generation of a Zipfian distribution in YCSB uses the algorithm introduced in [32], which includes the constants α and ζ given by

$$\alpha = \frac{1}{1-\theta},$$

$$\zeta = \sum_{n=1}^{N} \left(\frac{1}{n}\right)^{\theta},$$

where N is the number of stored objects and $0 < \theta < 1$ is the skew. To read or insert an object, the Zipfian generation returns an object number: $OID = base + N * ((\eta * rand()) - \eta + 1)^{\alpha}$, where

$$\eta = \frac{1 - \left(\frac{2}{N}\right)^{(1-\theta)}}{1 - \frac{\zeta_2}{\zeta}},$$

base is the minimum value of the object number, ζ_2 is a constant ζ with $N = 2$, and *rand*() returns a pseudo-random number of *double* type. This facilitates the rapid generation of datasets, but the constraint of θ means that the Zipfian distribution does not recreate the skewness of the access distribution of a mobile messenger service. According to our analysis, cold storage requires a more strongly skewed distribution generator.

Eventually, we created a new workload for cold storage based on our analysis results. The workload had 54.46 % read operations and 45.54 % insert operations, and the object size was greater than 10 KB. Additionally, we added a constant β to the algorithm to generate a strongly skewed latest distribution. The SSLG uses α given by

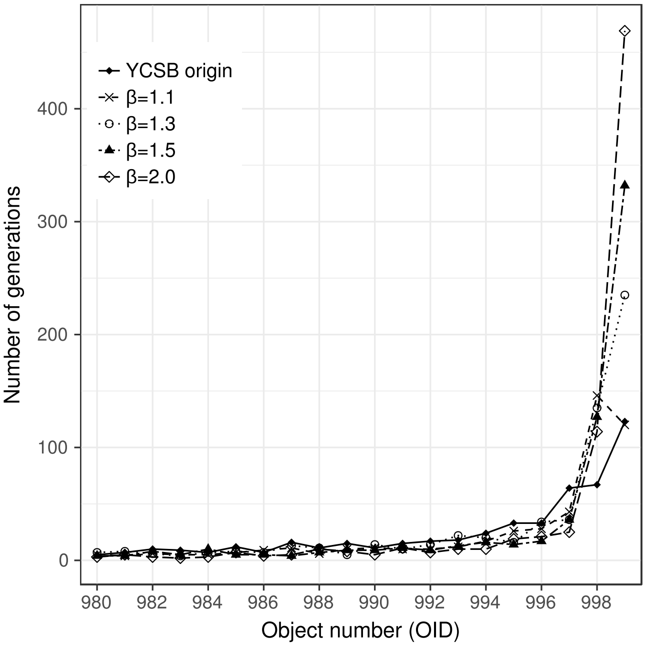


FIGURE 11. Illustration of the skewed distribution of generation results.

$$\alpha = \frac{\beta}{1-\theta},$$

where $\beta > 1$ and larger values produce a stronger skew.

Figure 11 illustrates five skewed distributions given by the YCSB Zipfian generator and SSLG. The experiment considers a storage system with 1,000 objects, and the generators randomly select any object 1,000 times. As the object number increases, the data is considered to be younger and the selection probability increases. The distribution of the YCSB Zipfian generator is skewed to the right in agreement with the Zipfian distribution with $\rho \approx 1$. The value of ρ cannot be calculated exactly because the generator includes a pseudo-random function. We examined the SSLG with $1.1 \leqq \beta \leqq 2.0$. The results indicate that the SSLG generates a distribution that is more than twice as skewed as the YCSB Zipfian generator. The SSLG can overcome the Zipfian generator constraint by applying a carefully improved algorithm. Although adjustments to the algorithm are unobtrusive, this facilitates the emulation of mobile messenger storage systems without any overhead.

In summary, We presented a large-scale object storage benchmark for the performance analysis of energy-efficient storage systems. This benchmark is based on YCSB, and we developed a YCSB client for RADOS that can be used for the interaction between a YCSB core and Ceph object storage. In addition, we defined a new workload for a mobile messenger service and improved the Zipfian generator to recreate the object access patterns of mobile messenger storage systems. This benchmark and the testbed with the developed power monitors are complementary, which suggests that this framework will be useful in the research on energy-efficient large-scale storage systems.

VI. CONCLUSIONS

This study investigated the I/O request characteristics of a real mobile messenger service and validated the need for serviceable cold storage using statistical analysis. It is difficult to avoid the conclusion that modern mobile messenger services have a more strongly skewed distribution than other services. Additionally, we examined the effect of spin-down technology, and presented a cold storage testbed with a hard disk power monitor. Finally, we implemented a benchmark framework for large-scale storage systems based on YCSB. The testbed and framework are expected to accommodate further research on serviceable cold storage. We have identified several directions for future work. First, work is needed to establish object placement strategies in a distributed file system. Second, disk failures need to be considered to ensure data durability and storage reliability. We also plan to use our framework to conduct experiments on distributed file systems with appropriate object placement strategies, which can increase the number of spun-down hard disks to reduce the power consumption significantly.

ACKNOWLEDGMENT

The authors would like to thank the industrial sponsors of NAVER Labs and LINE corporation.

REFERENCES

[1] M. Rogoway. (Jul. 2013). *Facebook: Power Use in Prineville Data Center More Than Doubled Last Year*. [Online]. Available: http://www.oregonlive.com/silicon-forest/index.ssf/2013/07/facebook_power_use_in_prinevil.html

[2] P. Llopis, J. G. Blas, F. Isaila, and J. Carretero, "Survey of energy-efficient and power-proportional storage systems," *Comput. J. Oxfords J.*, vol. 57, no. 7, pp. 1017–1032, Apr. 2013, doi: 10.1093/comjnl/bxt058.

[3] (Jan. 2017). *Number of Monthly Active LINE Users Worldwide as of 4th Quarter 2016 (in millions)*. [Online]. Available: https://www.statista.com/statistics/327292/number-of-monthly-active-line-app-users/

[4] J. Lee, C. Song, and K. Kang, "Analyzing I/O patterns for the design of energy-efficient image servers," in *Proc. IEEE Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2014, pp. 1–8.

[5] J. Lee, C. Song, and K. Kang, "Energy-efficient storage policy for instant messenger services," in *Proc. IEEE 5th Int. Conf. Big Data Cloud Comput.*, Aug. 2015, pp. 38–44.

[6] Y. Wang, M. Kapritsos, L. Schmidt, L. Alvisi, and M. Dahlin, "Exalt: Empowering researchers to evaluate large-scale storage systems," in *Proc. 11th USENIX Conf. Netw. Syst. Design Implement. (NSDI)*, 2014, pp. 129–141.

[7] J. Lee, C. Song, and K. Kang, "A test-bed for the assessment of power management strategies in tiered storage systems," in *Proc. 8th ACM Int. Syst. Storage Conf.*, 2015, p. 19.

[8] J. Lee, C. Song, and K. Kang, "Benchmarking large-scale object storage servers," in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jun. 2016, pp. 594–595.

[9] (2007). *Home > Open Compute Project*, accessed on May 1, 2017. [Online]. Available: http://www.opencompute.org

[10] M. Yan. (Jan. 2013). *Open Vault Storage Hardware V0.5 ST-Draco-Abraxas-0.5*, Open Compute Project. [Online]. Available: http://www.opencompute.org/projects/storage/

[11] T. Bisson, S. A. Brandt, and D. D. E. Long, "A hybrid disk-aware spin-down algorithm with I/O subsystem support," in *Proc. IEEE Int. Perf., Comput., Commun. Conf.*, Apr. 2007, pp. 236–245.

[12] F. Douglis, P. Krishnan, and B. N. Bershad, "Adaptive disk spin-down policies for mobile computers," in *Proc. 2nd Symp. Mobile Location-Independent Comput. (MLICS)*, 1995, pp. 121–137.

[13] Y.-H. Lu and G. de Micheli, "Adaptive hard disk power management on personal computers," in *Proc. 9th Great Lakes Symp. VLSI (GLS)*, 1999, pp. 50–53.

[14] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "DRPM: Dynamic speed control for power management in server class disks," *ACM Special Interest Group Comput. Archit. News*, vol. 31, no. 2, pp. 169–181, 2003.

[15] *AT Attachment-8 ATA/ATAPI Command Set—2 (ACS-2), T13, Rev. 7*, American National Standard T13/2015-D, Jun. 2011.

[16] J. Zedlewski, S. Sobti, N. Garg, F. Zheng, A. Krishnamurthy, and R. Wang, "Modeling hard-disk power consumption," in *Proc. 2nd USENIX Conf. File Storage Technol. (FAST)*, 2003, pp. 217–230.

[17] "Power management on adaptec unified serial RAID controllers and HGST deskstar hard drives," HGST, San Jose, CA, USA, White Paper, Nov. 2008.

[18] *Information Technology-ATA/ATAPI Command Set—3 (ACS-3), T13, Rev. 5*, American National Standard T13/2161-D, Oct. 2013.

[19] C. Albrecht *et al.*, "Janus: Optimal flash provisioning for cloud storage workloads," in *Proc. USENIX Ann. Tech. Conf. (ATC)*, 2013, pp. 91–102.

[20] J. Parikh. (Jan. 2013). *Cold Storage—Jay Parikh in Open Compute Summit IV*. [Online]. Available: http://new.livestream.com/accounts/2462150/events/1790124/videos/9502681

[21] E. Thereska, A. Donnelly, and D. Narayanan, "Sierra: Practical power-proportionality for data center storage," in *Proc. 6th Conf. Comput. Syst. (EuroSys)*, 2011, pp. 169–182.

[22] S. Lee. (Sep. 2014). *Line: How to be a Global Messenger Platform*. [Online]. Available: http://www.slideshare.net/deview/2a1line

[23] J. Chou, J. Kim, and D. Rotem, "Energy-aware scheduling in disk storage systems," in *Proc. 31st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2011, pp. 423–433.

[24] T. Bostoen, S. Mullender, and Y. Berbers, "Analysis of disk power management for data-center storage systems," in *Proc. 3rd Int. Conf. Future Energy Syst., Where Energy, Comput. Commun. Meet (e-Energy)*, 2012, pp. 1–10.

[25] "High-Side Measurement, Bi-Directional Current/Power Monitor W/I2C Interface," Texas Instruments Incorporated, Dallas, TX, USA, Data Sheet, 2011.

[26] T. Feldman and G. Gibson, "Shingled magnetic recording: Areal density increase requires new data management," *USENIX, Login Mag.*, vol. 38, no. 3, pp. 22–30, 2013.

[27] G. Zhang, H. Li, S. Shen, and S. Wu, "Simulation of HGA vibration characteristics inside the helium-filled hard disk drive," *IEEE Trans. Magn.*, vol. 52, no. 4, pp. 1–6, Apr. 2016.

[28] *hdparm download | SourceForge.net*, accessed on May 1, 2017. [Online] Available: https://hdparm.sourceforge.io/

[29] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with YCSB," in *Proc. 1st ACM Symp. Cloud Comput.*, 2010, pp. 143–154.

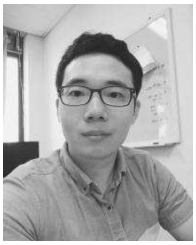
[30] G. DeCandia *et al.*, "Dynamo: Amazon's highly available key-value store," in *Proc. 21st ACM SIGOPS Symp. Oper. Syst. Principles*, 2007, pp. 205–220.

[31] *Release YCSB 0.10.0 ů brianfrankcooper/YCSB*, accessed on Jul. 5, 2016. [Online] Available: https://github.com/brianfrankcooper/YCSB/releases/tag/0.10.0

[32] J. Gray, P. Sundaresan, S. Englert, K. Baclawski, and P. J. Weinberger, "Quickly generating billion-record synthetic databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 1994, pp. 243–252.

JAEMYOUN LEE received the B.S. degree in computer science and engineering from Hanyang University ERICA campus, South Korea, in 2012. He is currently pursuing the Ph.D. degree in computer science and engineering with Hanyang University, Ansan, South Korea. His research interests are in the areas of energy-efficient distributed file systems, scalable and persistent storage systems, high-reliability and resilient systems, and modeling and analysis of cyber-physical systems. His current research is centered on serviceable large-scale cold-storage systems in data centers. He is a Student Member of the IEEE Computer Society.

CHANG SONG received the B.S. degree in computer science from Iowa State University, USA, and the M.S. degree in computer science from Purdue University, USA, in 1996. He was with Digital Equipment Corporation in UNIX file system and was involved in VFS and Cluster file systems. He was with HP, Microsoft, and Apple. He then moved to NAVER working on the service performance team, and later became the CTO of NAVER in 2015. He founded NAVER Labs (NAVER spin-off) in 2017. His research interests include cloud service infrastructure, system performance analysis and improvements, next generation interface, maps, machine learning, autonomous machine, product design, and AR/3D.



CHANYOUNG PARK received the B.S. degree in computer science and engineering from Hanyang University ERICA campus, South Korea, in 2016. He is currently pursuing the M.S. degree in computer science and engineering with Hanyang University, Ansan, South Korea. His current research interests include distributed file systems, intelligent storage systems, storage networking technologies, storage virtualization, and advanced driver assistance systems. He is a Student Member of the IEEE Computer Society.

SOYEUN KIM received the B.S. degree in statistics from Seoul National University, South Korea, in 1999, and the M.Math. and Ph.D. degrees in actuarial science from the University of Waterloo, Waterloo, ON, Canada, in 2001 and 2007, respectively. From 2001 to 2002 and from 2007 to 2009, she was an Associate with Pricewaterhouse-Cooper LLP in Toronto, Canada. From 2009 to 2013, she was with the Korea Insurance Research Institute, Korea. She joined the School of Business Management, Hongik University, South Korea, where she is currently an Assistant Professor. Her research interests include ruin theory and property and casualty insurance claims modeling.

KYUNGTAE KANG (M'05) received the B.S. degree in computer science and engineering and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, South Korea, in 1999, 2001, and 2007, respectively. From 2008 to 2010, he was a Post-Doctoral Research Associate with the University of Illinois at Urbana–Champaign, Illinois, USA. In 2011, he joined the Department of Computer Science and Engineering, Hanyang University, South Korea, where he is currently an Associative Professor. His research interests are primarily in systems, such as operating systems, wireless systems, distributed systems, mobile systems, and interdisciplinary area of cyber-physical systems. He is a member of the ACM.