Energy-efficient Storage Policy for Instant Messenger Services

Jaemyoun Lee*, Chang Song†, Kyungtae Kang*
*Dept. of Computer Science & Engineering, Hanyang University, Korea
Email: {jaemyoun, ktkang}@hanyang.ac.kr
†Naver Labs, Korea
Email: chang.song@navercorp.com

***Abstract*—An instant messenger service is one of the most useful mobile apps for exchanging messages and photos with friends. The infrastructural needs of such services are rapidly growing with user demand for high-quality service, and high-end computing with sufficient resources are required to meet these demands. However, the amount of power required to maintain such a massive infrastructure is impractically high. We present a novel policy to exploit energy-efficient storage servers for instant messenger services. Because instant messages often include personal information, the storage servers must be protected from unauthorized access. This requirement can be expanded to develop a creative policy for reducing power consumption. We validate our proposal by conducting four experiments, the results of which are expected to play an important role in designing energy-efficient servers for instant messenger services.**

1. Introduction

Instant messenger services are widely considered to be the future of social networking. They make it easy to have real-time personal conversations, which can nevertheless be rich environments for media sharing, entertainment, and even commerce. These messenger services are riches in facilitate than previous messenger services, which allow little more than the exchange of typed text. However, these newer services require more complicated infrastructures which in turn use computing power at an increasing rate that must soon become unsustainable. A data center that provides Facebook messenger services uses more than 150 million kilowatt hours of energy, which is roughly as much as 13,000 homes, and the power consumption of this data-centers is expected to double every year [1, 2].

The Open Compute Project (OCP) [3] has designed a storage server, called Cold Storage, for the retention of infrequently used data (cold data) [4]. A Cold Storage server contains 30 shingled magnetic recording (SMR) hard disks, of which only two are maintained in the *active* state while the remaining 28 disks are spun down. This approach to server design has obvious potential for power saving, but the source code that controls disk activity in Cold Storage, and in other similar servers, is not in the public domain. What is clear is that Cold Storage tries to match energy consumption with I/O requirements. A file system of this sort may be described as power-proportional, and they are becoming essential components of more efficient data centers.

In experiment, we have observed a vast difference between the I/O request patterns generated by modern instant messenger services and those generated by previous messenger services, owing to the widespread use of smartphones and tablets to access modern messenger services, which increasingly dealing with photos as well as text. In addition, data confidentiality is a major concern in instant messenger services, and it turns out that meeting this requirement suggests ways of reducing the power consumption of storage servers.

In this paper, we conduct four experiments on the LINE instant messenger service from Line Corporations, which had more than 400 million registrants worldwide in 2014 [18]. Following previous studies [15–17], we analyzed access rates and file ages to determine patterns of I/O requests; but, in addition, we looked closely at last access times and request counts, with the aim of characterizing specific features of instant messenger services. Based on the results of these tests, we propose a novel policy to exploit energy-efficient storage servers for instant messenger services, and the major contributions can be summarized as follows: (1) we provide guidelines for reducing energy consumption in distributed file systems, and (2) we propose a power-proportional policy that ensures reliability.

The remainder of this paper is organized as follows. In Section 2, we provide the background for this study, discussing power-proportional storage servers, the spin-down technique, and power consumption patterns of hard disks; in addition, we present an example showing the workload of a modern storage server. In Section 3, we outline the methodology used in our experiments on hard-disk power consumption and in the analysis of the resulting traces. In Section 4, we evaluate the results of our experiments. Finally, we summarize our findings and conclude the

2. Background

2.1. Cold Storage

The OCP [3] was founded by Facebook in 2011 with the objective of replicating the concepts underlying open source

CFP1552Z-CDR/15 $31.00 © 2015 IEEE
DOI 10.1109/BDCloud.2015.13

software in order to create an *open hardware* movement to build commodity systems 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, in order 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 [4]. A Cold Storage consists of 30 hard disks in two trays. A rack contains 16 Cold Storages; thus, one rack contains 480 hard disks. Only one of the 15 hard disks in a Cold Storage tray is able to spin up at any given time; the others spin down to conserve power. In other words, only two hard disks in a Cold Storage are in operation at any given time [4]. Thus, power is naturally conserved because most of the hard disks are spun down.

However, even though the spin-down technique can significantly reduce the power consumption of data centers, a suitable I/O scheduling methodology is required because hard disks can usually spin down only a limited number of times. Furthermore, pathological workloads can completely negate the power-saving benefits of the spin-down technique, prematurely causing a disk to exceed its duty cycle rating, and significantly increasing the aggregate spin up latency [5]. Although the OCP has published hardware specifications for Cold Storage, its file systems specifications have not been published thus far. Other open sources for tiered file systems remain elusive because file systems have a good commercial value. 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.

2.2. Spin-down technique

The spin-down technique, which sets a disk into a low-power mode while it is idle, is used to reduce hard disk power consumption. In low-power modes, such as *standby*, the spindle motor does not spin and the 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 [5–8]. Typically, these algorithms are time-out driven, i.e., they spin down a disk if a time-out occurs before a request is received.

The spin-down technique in the Power Management feature [9] allows a hard disk to save energy, i.e., it reduces power consumption by changing the hard disk state from *active* to *idle*, *standby*, or *sleep*. In the *idle* state, operations that can be performed are restricted as compared to the *active* state. However, in the *idle* state, the spindle motor of the hard disk continues to spin, and the disk head remains on the platters. Consequently, the amount of hard disk power conserved is very small. 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 as compared to the *standby* state [10]. The *sleep* state is similar to the *standby* state, but only a reset operation can be performed in this state, i.e., hard or soft reset. Thus, in theory, the *sleep* state consumes the least amount of power among all states.

Recently the *idle* state was combined with the *active* state to create the Advanced Power Management feature [11]. This feature allows the hard disk to automatically change its state to either *active* or *idle*. To enter the *standby* and the *sleep* states, a special command must be input manually. This command is specified in [9]. 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 a concrete power management feature are left to the discretion of the drive vendor.

2.3. Power consumption patterns of hard disks

Research related to the spin-down technique [5, 12, 13] is predominantly focused on average power consumption, with no regard for instantaneous power consumption. However, the parts of a hard disk are physical components, the power consumption of which has fickle, irregular patterns. Lee et al. [14] observed that the graph of hard disk power consumption first increased steeply and then flattened out when the hard disk state changed from *standby* to *active*. The peak power consumption was five times greater than the average power consumption. They suggested that the instantaneous power consumption should be considered when designing a spin-down scheduler.

A hard disk consistently consumes $3425\,\mathrm{mW}$ while waiting for a command and $5699.3\,\mathrm{mW}$ for $1.08\,\mathrm{s}$ on average after receiving a mount command or an unmount command. Furthermore, the power consumption is generally maintained at $3850\,\mathrm{mW}$ until the hard disk state changes to *idle*. The time interval between the command and the state change is $1.8\,\mathrm{s}$. It should be noted that the peak power consumed by the hard disk is $2.3\,\mathrm{times}$ greater than the average for a fleeting moment when the state is changed. The spindle motor of the hard disk is able to stop within $1020\,\mathrm{ms}$, after which the hard disk consumes $1079.5\,\mathrm{mW}$ on average. After the hard disk has spun down, it consumes $700\,\mathrm{mW}$, which is five times smaller than the average in the *active* state. Thus, the hard disk spin-down scheduler does not need to consider spinning down all hard disks during a burst.

However, when the hard disk is spinning up, it consumes 5.8 times the average power for $0.45\,\mathrm{s}$, and then it consumes $4050\,\mathrm{mW}$ for around $4.0\,\mathrm{s}$. Thus, the amount of power consumed is immense as compared to that in the case of other commands or 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 would result in accidents

such as hardware failure and short circuits. Therefore, large-scale storage systems must take all reasonable precautions to prevent simultaneous spin up.

2.4. Server I/O patterns

Albrecht et al. [15] conducted an experiment involving thousands of Google users, applications, and services such as content indexing, advertisement serving, Gmail, and video processing, as well as smaller applications, such as MapReduce jobs owned by individual users. A large application may include many component jobs. The workload characteristics and demands of jobs in data centers typically vary considerably among users and jobs. As a result of the variation in mean read age over different jobs in Google's data centers, the mean read age of the bytes read over 15000 jobs is approximately 30 days, even though jobs access very young (one minute old) to very old (one year old) data. Another experiment has shown that $50\,\%$ of the data stored by a particular user is less than one week old, but corresponds to more than $90\,\%$ of the read activity.

Parikh [16] discussed the necessity for Cold Storage in Facebook's data centers. He argued that $2.8\,\mathrm{ZB}$ (zettabytes) of data were created in 2012, and that $40\,\mathrm{ZB}$ of data would exist globally by 2020. To store all these data, data centers require billions of hard disks, each having the maximum capacity (4 TB). Hard disks consume a significant amount of power; 153 million kilowatt hours of power were consumed by a single Facebook data center in 2012, which roughly equals the power consumed by 13,000 homes [1]. However, most 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 hardly requested. More specifically, $82\,\%$ of read traffic is serviced for only $8\,\%$ of young photos in Facebook's data centers.

These results indicate that the requirements for cold data that is stored on disks but almost never read, such as legal data or backups of third copies of data, are continuously increasing. Consequently, a tier system that separates data into hot, warm, and cold storage has been proposed. Furthermore, empirically, aged data is likely to become cold data.

Finally, Thereska et al. [17] discussed the I/O patterns of instant messenger services. The I/O patterns of modern instant messenger services differ significantly from those of traditional instant messengers. The demand for exchanging photos and videos via conversations is growing with the popularity of smartphones having high quality cameras. Therefore, the I/O trace data of LINE's photo servers must be analyzed to determine the workload characteristics of modern instant messenger services.

3. Experiments

We examined actual workloads generated by the LINE instant messenger service, which allows users to exchange text messages, pictures, video, and audio data; users can also make VoIP calls, and hold audio and video conferences without charge. We traced requests for the input and output of images arriving at all LINE servers over seven days. This is certainly a large-scale storage system workload: LINE has 20,000 servers, including more than 10 Redis clusters and 10 HBase clusters [19], and the trace of a week's worth of data contains billions of lines. We attempted to use the R Project software to analyze this workload, but we encountered memory overflow problems, even though the system we used had two Xeon processors and $32\,\mathrm{GB}$ of RAM. Therefore, we started again with MySQL, a widely used open-source relational database management system. It provided capable of handling this large workload, with a creditable transaction processing speed.

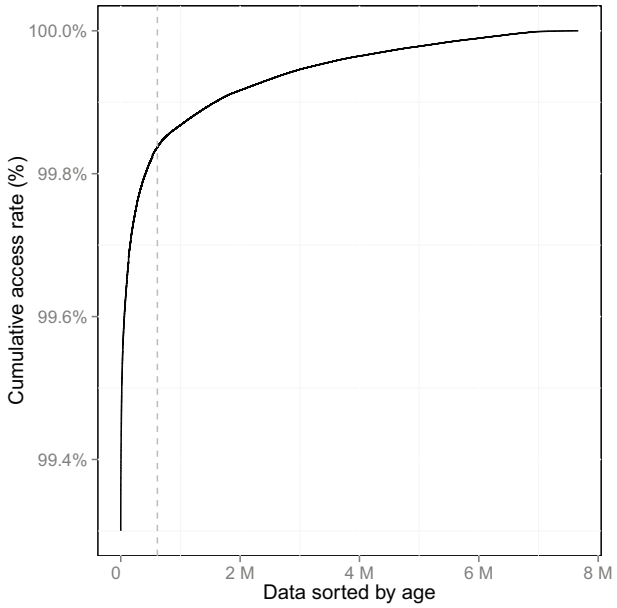

Figure 1. Cumulative distribution of read and write operations sorted by the age of the data. The data on the left are younger than those on the right. The vertical dotted line indicates that the youngest $8\,\%$ of the data represents $99.8\,\%$ of the traffic.

3.1. Analysis of access patterns

First, we sampled $1\,\%$ of the workload, and then sorted this sample workload by file age and calculated the access rate in order to determine the I/O patterns. Figure 1 shows that the youngest $8\,\%$ of requests make up $99.8\,\%$ of I/O traffic, where the horizontal axis is number of files by age. This $92\,\%$ of older data which is unlikely to be read, or may never be read. If this cold data is migrated to power-proportional storage servers, we would expect a considerable amount of energy to be saved for a very small reduction in performance, manifest as a delay which only occurs if there is a request for data stored in a hard disk that is in *standby* state. However, our traces record I/O requests arriving at the system, not its disks, and therefore they take no account

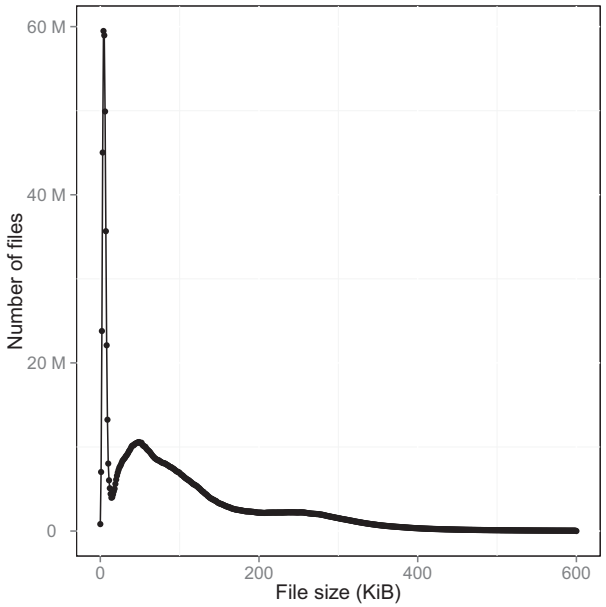

Figure 2. Distribution of requested files by size. Files with sizes above 600 KB are excluded, as these were found to be requested very infrequently: four times at most.

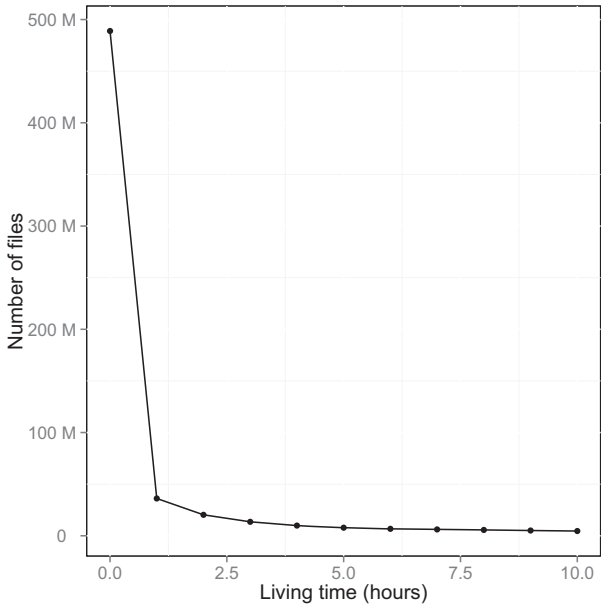

Figure 3. Number of files remaining live, for periods of an hour, up to 10 hours. The number of files that are accessed later is very small, and thus these files are of little interest.

of caching; but large-scale storage systems have cache subsystems, and instant messenger applications also have cache space. These caches enable power-proportional storage systems to avoid routing I/O requests to hard disks that are on standby; and the scope for energy saving increases with the sizes of these caches.

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. Then we counted the number of requests for files in each group. The results are shown in Figure 2. The majority of requests occur 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 the user sends an image with a larger file, the instant messenger compresses the file into this range before sending it to its storage servers. The I/O pattern for this class of files suggests that the number of requests corresponds to the number of people to be found 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 again it is preferable to consider them as cold data.

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

3.2. Selection of cold data

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 shows that this threshold used to classify data as cold needs to be radical. We therefore looked at the length of time for which files remain 'live': meaning that they are actively being accessed.

Initially, we performed this analysis for periods of one hour, with the results shown in Figure 3. We see that nearly 64 % of all files remain live for an hour or less, and another 4 % are accessed between 1 and 2 hours. Thus 1 hour is a plausible threshold for classifying files as cold.

We repeated this experiment, looking at periods of one second over the first hour, with the results shown in Figure 4. The first 2 minutes of this graph are shown magnified in Figure 5. The life of many files is less than 3 seconds, and half of all files are finally accessed within 73 seconds as summarized in Table 1.

However, it is impossible to predict accurately which files will be accessed within 2 minutes of being written; and if our prediction is wrong, hard disks that were spun down will have to be spun up again, reducing performance and increasing power consumption. Therefore, determining

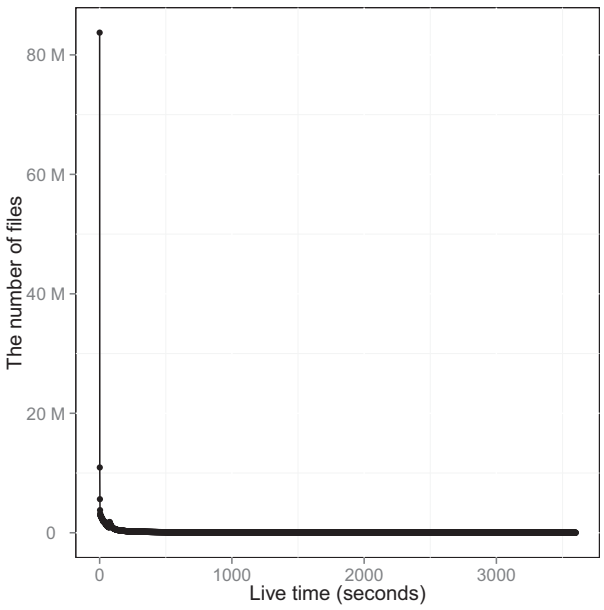

Figure 4. Number of files remaining live, for periods of one second, up to one hours.

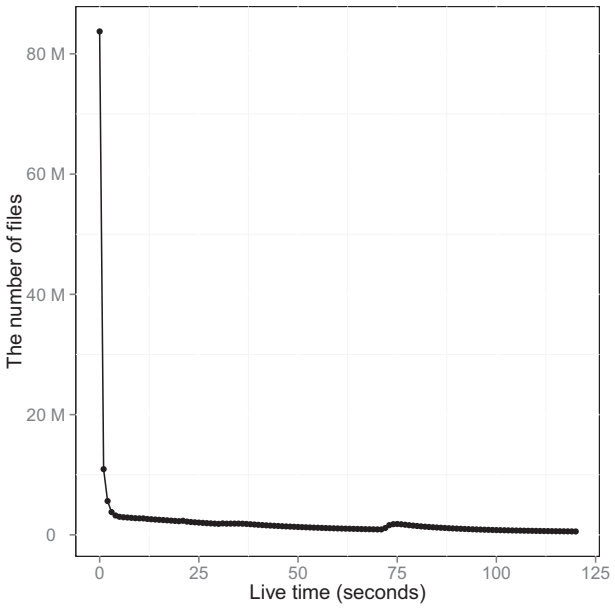

Figure 5. Number of files accessed within 2 minutes.

the threshold based only on time is not a suitable approach for instant messenger services.

To establish a more efficient policy of determining cold data, we compare instant messenger services with other services such as the Web, e-mail, and FTP. These other services are provided to usually unspecified masses. Although the recipients of an e-mail can be limited, an e-mail can be forwarded, whereas instant messenger services do not allow forwarding. Thus, conversations are shared by a known group of users, and confidentiality is more likely to be preserved.

TABLE 1. NUMBER OF FILES WITH FINAL ACCESS TIMES UNDER TWO MINUTES, ANALYZED BY PERIODS OF ONE SECOND.

Second	Ratio	Cumulative ratio
0	18.419 %	18.419 %
1	2.404 %	20.823 %
2	1.237 %	22.060 %
3	0.834 %	22.894 %
4	0.709 %	23.603 %
5	0.661 %	24.264 %
70	0.200 %	49.267 %
71	0.199 %	49.467 %
72	0.252 %	49.718 %
73	0.355 %	50.073 %
74	0.390 %	50.463 %
75	0.397 %	50.860 %

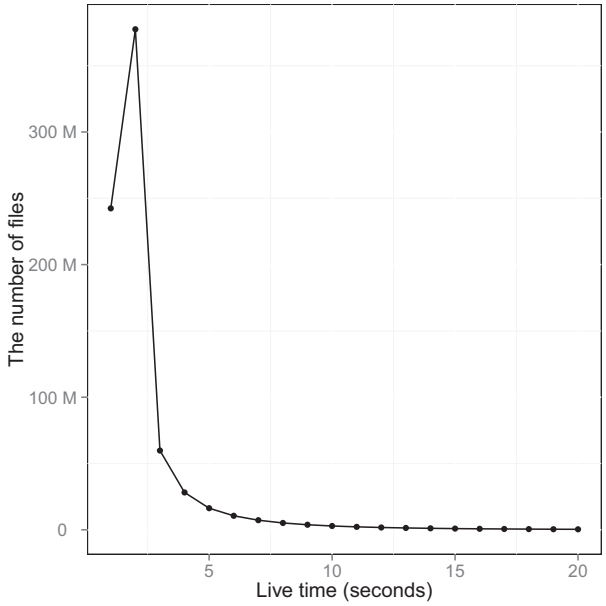

Figure 6. Number of files plotted against number of request for each file.

Figure 6 shows that 31.62 % of files are requested once, and 49.24 % are requested twice, and: thus 80.86 % are requested fewer than three times. This suggests that most files are going to a single recipient, when the first operation is the write request required to upload the image from

the sender, and the second is the read operation which downloads the image to the device of the recipient. There is of course no need to download the image to the sender's device; and the recipient only downloads the image once because their device stores it in the application cache area.

Our approach to categorizing data as cold can be made more discerning 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; so we can label that file as cold data and migrate it into power-proportional storage servers. Even if a file has not been accessed by all its recipients, we must at some time take the view that the remaining recipients will not access it, or are unlikely to so for some time: on that basis, files which have not been accessed for an hour can be labeled as cold data and moved into power-proportional storage servers.

4. 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 is cold. Compared with [16], the access rate of young data is extremely high in an instant messenger service. Parikh [16] indicated that $8\,\%$ of data forms $82\,\%$ of the traffic; however, in an instant messenger workload, $8\,\%$ of data is $99.8\,\%$ of the traffic.

We believe that an instant messenger service has such a high proportion of cold data because the chat rooms provided by an instant messenger service 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. As a result, the access rate of files decreases sharply for older files. For this reason, an instant messenger workload has a high proportion of extremely cold data. This is a characteristic of modern instant messenger services, and the same I/O pattern is not observed in instant messengers which only send text messages [17].

To reduce the power consumption of storage servers, they need to be operated according to a policy which reflects this distinctive workload. Some other 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 instant messenger workload, than data age. All this suggests that the operation of a power-proportional storage system in an instant 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 compares 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 in order to reduce power consumption.

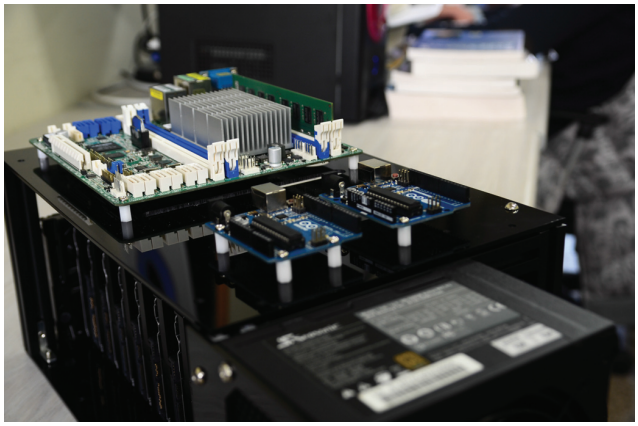

Figure 7. Our small-scale test-bed of storage server with 30 hard disks and consumed power measurement modules

3) If the file remains in hot storage for too long, it should be migrated into the power-proportional storage servers in any case. The threshold time should be set appropriately depending on the workload, but one hour was shown to be a suitable value as described in Section 3.2.

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 current low-power hard disk technology and modern instant messenger workload. We built a small-scale Cold Storage that consists of 30 $4\,\mathrm{TB}$ hard disks, 30 power measurement modules, 6 control modules, and 3 server motherboards with Intel Avoton processors, as shown in Figure 7. We plan to conduct additional experiments to practically test our policy on real trace data of image servers of LINE practically using Ceph [20], a popular open-source distributed file system, by which we expect to significantly reduce power consumption.

5. Conclusion

We proposed a policy to reduce the maintenance costs of an instant messenger service provider using energy-efficient large-scale storage servers that minimize performance degradation. Basically, $99.8\,\%$ of the data stored in the image servers of an instant messenger service is cold data, and power consumption can be reduced drastically by using the proposed policy for early migration of this data into energy-efficient storage servers.

Acknowledgment

This research was supported in part by the MSIP (Ministry of Science, ICT and Future Planning), Korea and NAVER Corp., under ICT/SW Creative research program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2014-H0511-14-1002), in part by Institute for Information & communications Technology Promotion

(IITP) grant funded by the Korea government (MSIP) (No. B0101-15-0557, Resilient Cyber-Physical Systems Research), and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the MSIP (NRF-2013R1A1A1059188). The corresponding author is K. Kang.

References

[1] M. Rogoway, "Facebook: Power use in Prineville data center more than doubled last year," July 2013; 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," *The Computer Journal of Oxfords Journals*, Apr. 2013; doi:10.1093/comjnl/bxt058.

[3] "Open Compute Project"; www.opencompute.org.

[4] M. Yan, "Open Vault Storage Hardware V0.5 ST-draco-abraxas-0.5," Open Compute Project, Jan. 2013; urlwww.opencompute.org/projects/storage/.

[5] T. Bisson, S. Brandt, and D. D. E. Long, "A hybrid disk-aware spin-down algorithm with I/O subsystem support," *Proc. IEEE International Performance, Computing, and Communications Conference* (IPCCC 07), 2007, pp. 236–245.

[6] F. Douglis, P. Krishnan, and B. N. Bershad, "Adaptive disk spin-down policies for mobile computers," *Proc. 2nd Symp. Mobile and Location-Independent Computing* (MLICS 95), 1995, pp. 121–137.

[7] Y.-H. Lu and G. de Micheli, "Adaptive hard disk power management on personal computers," *Proc. 9th Great Lakes Symp. VLSI* (GLS 99), 1999, pp. 50–53.

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

[9] *Working Draft Project American National Standard T13/2015-D, AT Attachment- 8 ATA/ATAPI Command Set - 2 (ACS-2)*, T13, Rev. 7, June 2011.

[10] J. Zedlewski, S. Sobti, N. Garg, F. Zheng, A. Krishnamurthy, and R. Wang, "Modeling hard-disk power consumption," *Proc. 2nd USENIX Conf. File and Storage Technologies* (FAST 03), 2003, pp. 217–230.

[11] "Power management on Adaptec Unified Serial RAID controllers and HGST Deskstar hard drives," White Paper, HGST, Nov. 2008.

[12] J. Chou, J. Kim, and D. Rotem, "Energy-aware scheduling in disk storage systems," *Proc. 31st International Conf. Distributed Computing Systems* (ICDCS 11), 2011, pp. 423–433.

[13] T. Bostoen, S. Mullender, and Y. Berbers, "Analysis of disk power management for data-center storage systems," *Proc. 3rd International Conf. Future Energy Systems: Where Energy, Computing and Communication Meet* (e-Energy 12), 2012, pp. 1–10.

[14] J. Lee, C. Song, and K. Kang, "Analyzing I/O patterns for the design of energy-efficient image servers," *Proc. IEEE International Performance Computing and Communications Conference* (IPCCC 14), 2014, pp. 1–8.

[15] C. Albrecht, A. Merchant, M. Stokely, M. Waliji, F. Labelle, N. Coehlo, X. Shi, and E. Schrock, "Janus: Optimal flash provisioning for cloud storage workloads," *Proc. USENIX Ann. Technical Conference* (ATC 13), 2013, pp. 91–102.

[16] J. Parikh, "Cold Storage - Jay Parikh in Open Compute Summit IV," Jan. 2013; http://new.livestream.com/accounts/2462150/events/1790124/videos/9502681.

[17] E. Thereska, A. Donnelly, and D. Narayanan, "Sierra: Practical power-proportionality for data center storage," *Proc. 6th Conf. Computer Systems* (EuroSys 11), 2011, pp. 169–182.

[18] "LINE now has 400 million registered users! : LINE official blog," Apr. 2014; http://official-blog.line.me/en/archives/1001168967.html.

[19] L. SeokChan, "Line: How to be a global messenger platform," Sept. 2014; www.slideshare.net/deview/2a1line.

[20] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," *Proc. 7th Conf. Operating Systems Design and Implementation (OSDI 06)*, Nov. 2006, pp. 307–320.