Power-Saving Method in Storage Systems for File Sharing Services

Horleang Choeng[†1,a)] Koji Hasebe[†1,b)] Hirotake Abe[†1,c)] Kazuhiko Kato[†1,d)]

Abstract: We present a power-saving method for large-scale storage systems. In particular, our prime target is storage systems for online file sharing services. In our previous study, we proposed a method based on the idea of Popular Data Concentration (PDC). In that study, data were periodically rearranged in the order of frequency of accesses in an environment where a vast amount of files is continuously uploaded. However, if old uploaded files are accessed frequently, the disks storing those files spin up that lead to increase power consumption. Thus, in this paper, we improve our previous method by focusing on file migration. When there are accesses to the old uploaded files, those files are immediately migrated to a small array of disks to avoid many accesses to the disks which are in standby mode. In simulations, we used the real access patterns of approximately 60,000 time-series images with a duration of 3,000 hours. The result of our evaluation showed that this method consumed approximately 12% less energy than the normal system in which there is not any file migration among disks.

1. Introduction

In recent years, with the development and spread of cloud computing services, the number of people using large-scale social networking services such as Facebook and Twitter is rapidly increasing. According to Smart Insights [1], in 2016, the number of people using social media is approximately 70% of the people using the Internet in the world. On the other hand, due to the development of cloud services, management cost in data center is considerably increasing. According to a data center report [2], in 2014, data centers in America consumed about 1.8% of total U.S. electricity consumption. The same report [2] also stated that the U.S. data center electricity consumption increased by approximately 4% from 2010 to 2014 and is expected to increase at the same rate at approximately 4% from 2014 to 2020. Therefore, reduction of management cost has become a huge issue. In particular, since the storage system occupies a high proportion of the power consumption of IT equipment, it is considered that power-saving method of the storage systems is effective in reducing the management cost.

Many methods have been proposed for the power-saving method in storage systems in the past recent years. As representative power-saving methods in storage systems are MAID (Massive Arrays of Idle Disks) [3] and PDC (Popular Data Concentration) [4]. These studies were based on a common idea which skewed the workload toward a small subset of disks, thereby enabling the other disks to be in standby mode.

In our previous study [6], we proposed a method based on the idea of PCD. In that study, data were periodically rearranged in the order of frequency of accesses in an environment where a vast amount of data is continuously uploaded. The rearrangement algorithm helped gathering popular files into a subset of disks and unpopular files into the other subset of disks. However, whenever there are frequent accesses to the old uploaded files which are storing in the disks in standby mode, those disks considerably spin up which leads to the increasing of power consumption.

To address this issue, we come up with an idea which is able to reduce the accesses to the disks in standby mode. The idea is to migrate the frequently accessed files storing in disks in standby mode to another disk which is always running, so that we can decrease the workload on those disks.

Thus, the objective of this study is to improve our previous method by focusing on file migration between a disk array (called Working Disks) and the other disk array (called Archiving Disks) rather than periodically rearranging stored data in a disk array. Whenever there are accesses to the old uploaded files, those files are immediately migrated to Working Disks to avoid many accesses to the disks which are in standby mode.

To evaluate the effectiveness of our new method, we measured the power consumption in our proposing system and compared it with a system without data migration among disks (called a normal system). In our simulations, we use the real access patterns of approximately 60,000 time-series

[†1] Presently with Department of Computer Science, University of Tsukuba, Tsukuba-shi, 305–8573 Japan
[a)] Email:horleang@osss.cs.tsukuba.ac.jp
[b)] Email:hasebe@cs.tsukuba.ac.jp
[c)] Email:habe@cs.tsukuba.ac.jp
[d)] Email:kato@cs.tsukuba.ac.jp

images which were selected from 500px [16] (an online photography community) with a duration of 3,000 hours. The result showed that this method consumed approximately 12% less energy than the normal system.

The rest of this study is organized as following. Section 2 presents related work. Section 3 gives the details of system design. Section 4 describes the migration techniques and Section 5 introduces the power consumption model. Sections 6 represents simulations, while Section 7 presents our simulation results. Finally, Section 8 concludes the study and presents future work.

2. Related Work

There has been many studies on power-saving in storage systems. Many of the approaches are to concentrate the workload into a small array of disks so that the rest of the disks can be shifted to be in standby mode. These approaches can be categorized into the following four groups.

The first group focuses on access frequency of stored files. PDC [4] periodically rearranged data in the storage disk array based on their latest access frequencies. Whereas, MAID [3] skewed workload to a subset of disks which were used as a cache to stored most frequently accessed files.

The second group focuses on redundancy of files. Energy efficient redundant and inexpensive disk array (EERAID) [13] exploits redundant information to directly optimize the disk access distribution by converting requests from one standby mode disk to other active mode disk(s). Power-Aware RAID (PARAID) [14] reduces power consumption by preparing duplication of data in an unused area and varying the number of disks according to the operational workloads.

The third group focuses on caching the data. Pergamum [15] uses NVRAM(Non-Volatile Random Access Memory) as a storage destination for caches and meta-data when writing data to optimize the direct accesses to the disks.

The three previous groups are targeted at small-scale storage systems. When such a method is applied to a storage system of a large-scale file sharing services, it is necessary to take into consideration that a large number of clients sequentially upload large amounts of requests and the like.

Therefore, the last group applies these techniques for large-scale storage systems. Data-classification-driven data placement of GreenHDFS [17] allows scale-down by guaranteeing substantially long periods of idleness in a subset of servers in the data center designated as the Cold Zone. These servers are then shifted to power-saving mode.

In our previous study [6], data were periodically rearranged in the order of frequency of accesses in an environment where a vast amount of data are continuously uploaded. In that study, the rearrangement algorithm helped the system allocate the data in the order of their popularity. The disks which stored less popular data are shifted to be in standby mode. However there is a problem that most of the uploaded data are rarely accessed as time elapsed, but accesses reoccurred occasionally and seem to happen randomly. Thus, merely by sorting data, it was difficult to classify data that was not accessed for a while and arranged these to a subset of disks.

Thus, in this paper, we improve our previous method by focusing on file migration between a disk array (called Working Disks) and the other disk array (called Archiving Disks) in a large-scale system rather than periodically rearranging stored data in a disk array. We use disks in Working Disks as a cache to store frequently accessed files. The reason is that, whenever there are accesses to the old uploaded files, those files can be immediately migrated to Working Disks so that the accesses to the disks in standby mode can be reduced.

3. System Design

In this section, we introduce our proposing system design. Our system is composed of disks with unique IDs, edge servers, an index server and I/O servers. Figure 1 illustrates the overall design (I/O servers are omitted for readability). Each disk is connected to an I/O server and is logically classified into one of three groups: Working Disk, Archiving Disks, and Empty Disks.

We assume that our system is a system to which data are uploaded continuously. Edge servers are responsible for all I/O requests from the clients. Index manager provides a lookup service for data accesses to any I/O servers of disks in Working Disks.

Edge servers only communicate with I/O servers of disks in Working Disks. They do not communicate with the I/O servers of disks in Archiving Disks. Files uploaded by the clients are always written to the disks in Working Disks via Edge server and I/O servers. Index manager assigns each file a unique ID and records it with the disk ID in which the file is stored.

In this study, The number of disks in Working Disk increase one by one as the files are sequentially uploaded by the clients until a maximum number. If the number of disks in Working Disk archives the maximum number, any access to the full disk in Working Disks leads to files migration. If any disk is full, some of the most unpopular files (10% of the most unpopular files inside the disk) in that disk are migrated to a disk in Archiving Disks which is supplied from Empty Disks. The corresponding I/O server records file IDs and disk ID in Archiving Disks to which the file is migrated. If the files are not migrated the disk ID is the corresponding disk itself. In our simulations, we use SSDs (Solid State Drives) for Working Disks but we can also use HDDs(Hard Disk Drives).

The disks in Archiving Disks and Empty Disks are composed of HDDs. At the start of the upload, all files are stored in the disks in Working Disks and the disk in Archiving Disks does not exist. When the migration occurs, Archiving Disks pull a disk from Empty Disks to supply the need to store the migrated files. Another disk is added whenever destination disk in Archiving Disks is full.

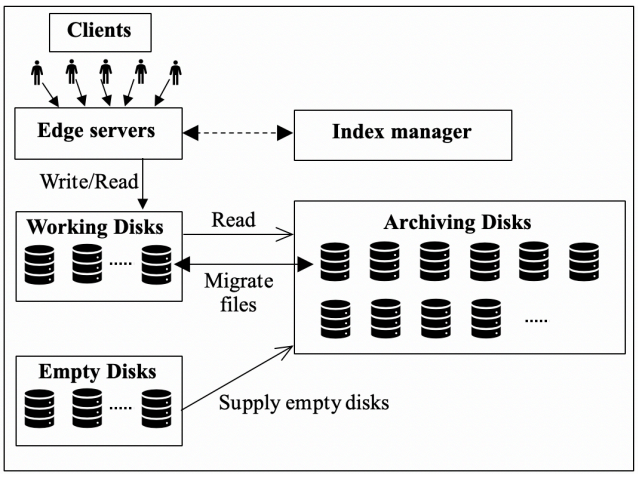

Fig. 1: System Design.

	(1)	(2)
(A)	(1) Files are migrated to an available Archiving Disk (do not leave a copy). (A) Accessed files are migrated to an available Working Disk	(2) Files are migrated to the first moved Archiving Disk (leave a copy). (A) Accessed files are migrated to an available Working Disk
(B)	(1) Files are migrated to an available Archiving Disk (do not leave a copy). (B) Accessed files are migrated to the first uploaded Working Disk	(2) Files are migrated to the first moved Archiving Disk (leave a copy). (B) Accessed file are migrated to the first uploaded Working Disk

Table 1: Four migration techniques. (1) and (2) are the options for file migration from Working Disks to Archiving Disks, while (A) and (B) are the options for file migration from Archiving Disks to Working Disks.

4. Migration Techniques

In this section, we present four migration techniques which are based on our proposing system design. Table 1 shows the four migration techniques. The details of file migration techniques are described below.

There are two options for file migration from Working Disks to Archiving Disks which are described as following.

(1) Files are migrated to an available (free space left) Archiving Disk. In this case, file migrated from a disk in Archiving Disks to Working Disks will be deleted from the disk in Archiving Disks.

(2) Files in Archiving Disks will never be deleted. When file migration occurs, we check whether the files exist in Archiving Disks. Existing files will be migrated to the previous disks in which they were stored. And, Non-existing files will be migrated to a new disk.

Similarly, there are two options for file migration from Archiving Disks to Working Disks.

(A) Files are migrated to an available(free space left) Working disk.

(B) Files are stored in the disks in which they were stored before they were migrated to the disks in Archiving Disks.

We have two possible options each for file migration which results in four options of file migration. We call them 1-A, 2-A, 1-B and 2-B.

We simulated all of these four methods to investigate their power consumption. The detailed results are described in Section 6.4.

5. Power Consumption Model

In this section, we describe power consumption model. For HDD model we refer to the study [7], while SSD model we refer to the study [8].

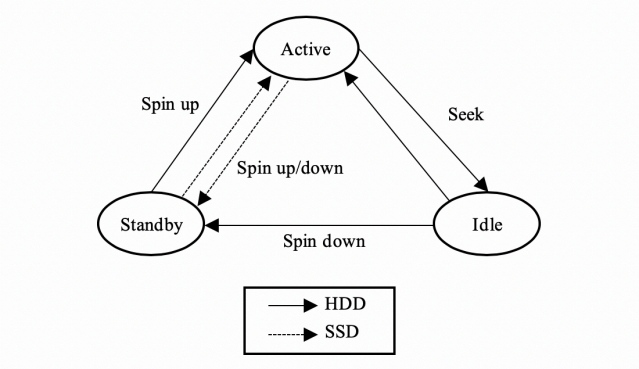

Fig. 2: Relationship among three modes.

5.1 Dynamic Power Management

Figure 2 illustrates relationship among the three modes. Section 5.1.1 and 5.1.2 describe the difference of power management between HDD model and SSD model.

5.1.1 HDD model

HDDs consist of three modes, Idle, Active and Standby mode. The changes between modes are made by three transitions: spin down, spin up, and seek, depending on the I/O request and the power-saving mode transition.

Data transfers occur in the Active mode. When there is not any access to the disk, the disk is waiting for I/O request. During this time, the disk switches to Idle mode, in which the disk spindle is still rotating. After a fixed threshold time (called the idleness threshold) has elapsed, if there is still not any access to the disk, the disk in Idle mode is switched to Standby mode which disk spindle stopped. In Standby mode, the disk workload is at rest which results in power saving.

5.1.2 SSD model

SSDs use a flash memory as a storage element and does not have any motor that drives the platters. Therefore, it does not generate spining down, spining up, or seeking. Since the difference of power consumption between the Idle mode and the Standby mode is very small, we do not distinguish between the Idle mode and the Standby mode. Thus,

there are only two modes, Idle mode and Active mode. In this case, when there is I/O request, the disk is in Active mode. When there is not I/O request, the disk switches to Idle mode.

5.2 Response Time and Power Consumption model

5.2.1 HDD model

Response time of accesses to a disk depends on current mode of the disk. When we set T_{acs} as access time (I/O request processing time), T_{sk} as seek time, T_{rt} as rotational latency, T_{tf} as transfer time, we have an equation as below.

$$T_{acs} = T_{sk} + T_{rt} + T_{tf}$$

Response time of accesses in each mode are:

$$T_{rp} = \begin{cases} T_{acs} & \text{(in Idle mode)} \\ T_{acs} + T_{up} & \text{(in Standby mode)} \\ T_{acs} + T_q & \text{(in Active mode)} \end{cases}$$

When the disk is in Idle mode, there is not any I/O requests to the disk. Therefore, the response time for I/O request is equal to the access time. When the disk is in Standby mode, the response time for platter to spin up (T_{up}) and access time (T_{acs}) are needed. When disk is in Active mode, the response time for current processing in disk (T_q) and access time (T_{acs}) are needed.

In this study, we use Seagate Desktop HDD ST1000DM004 [9] as HDD parameters for our simulations. Table 2 illustrates the HDD parameter setting.

In this parameter setting, T_q and T_{tf} are not fixed values since they depend on file sizes or current processing I/O request.

Idleness threshold T_{th} is the duration of Idle mode until it switches to Standby mode. If there is not any I/O request during this time, the disk spins down which results in power saving. A small value of T_{th} may lead to frequent disk spin up which results in high power reduction but the response time is long due to disk spin up. In contrast, a large value of T_{th} may improve response time but it requires more power consumption. Thus, it is essential to set an optimum value of T_{th}. In this study, we introduce break-even time (T_{be}) [10] to calculate the optimum value of T_{th}.

When we consider the time to the next access to HDD as T_{be}, T_{be} is the time which the power consumption between Idle mode and Standby mode is equal. T_{be} can be calculated as Equation 1 [10].

Here, T_{sb}^{min} is the minimum of time in Standby mode when the power consumption between Idle mode and Standby mode are equivalent, which satisfies Equation 2 [10].

From equation 1 and equation 2, we get T_{be}as Equation 3.

Based on our parameter setting, $T_{th} = T_{be} = 85.6s$, which we use in our simulations.

Symbol	Description	Value
D_{si}	Disk capacity	1,000GB
D_{ra}	Average data transfer rate	125MB/s
P_{id}	Power consumption in Idle mode	3.36w
P_{sb}	Power consumption in Standby mode	0.63w
P_{ac}	Power consumption in Active mode	5.9w
P_{sk}	Power consumption to seek	5.9w
P_{up}	Power consumption to spin up	24w
T_{sk}^{rd}	Average seek time for Read	8.5ms
T_{sk}^{wr}	Average seek time for Write	9.5ms
T_{tr}	Rotational latency	4.16ms
T_{up}	Spin up time	10s
T_q	Time for current processing I/O requests	-
T_{tf}	Data transfer time	-
T_{th}	Idleness threshold	85.6s

Table 2: HDD parameter setting.

Symbol	Description	Value
D_{si}^{SSD}	Disk capacity	1,600GB
D_{rd}^{SSD}	Read speed	3,200MB/s
P_{wr}^{SSD}	Write speed	2,100MB/s
P_{ac}^{SSD}	Power consumption in Active mode	13.3w
P_{id}^{SSD}	Power consumption in Idle mode	5w

Table 3: SSD parameter setting.

$$T_{be} = T_{up} + T_{sb}^{min} \tag{1}$$

$$P_{sb} \cdot T_{sb}^{min} + P_{up} \cdot T_{sb} = P_{id}(T_{sb}^{min} + T_{up}) \tag{2}$$

$$T_{be} = \frac{P_{up} \cdot T_{up} - P_{sb} \cdot T_{up}}{P_{id} - P_{sb}} \tag{3}$$

5.2.2 SSD model

Since spin down, spin up and seek do not exist in SSDs, the access time is:

$$T_{acs}^{SSD} = T_{tf}$$

Again, response time of accesses in each mode are:

$$T_{rp}^{SSD} = \begin{cases} T_{acs}^{SSD} & \text{(in Idle mode)} \\ T_{acs}^{SSD} + T_q & \text{(in Active mode)} \end{cases}$$

We use INTEL SSD DC P4610 SERIES 1.6TB [11] as our SSD parameters for our simulations. Table 3 illustrates SSD parameter setting.

6. Simulation

6.1 Photographs for Simulation

We used 63,204 time-series images from 500px [16] which is an online photography community, for simulations. All photographs include cumulative number of accesses, likes (the number which shows the popularity of the photographs voted by the users), comments and tag information every hour for a duration of 3,000 hours. We analyzed the bias of popularity of each photograph to check the relation with time since they were uploaded.

6.2 Analysis of Photographs for Simulation

Figure 3 shows hourly average number of accesses of images over 3,000 hours. Here, the horizontal and vertical axes

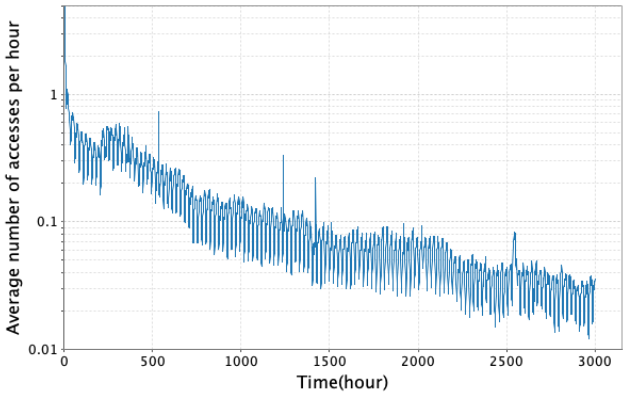

Fig. 3: Hourly average number of access of images.

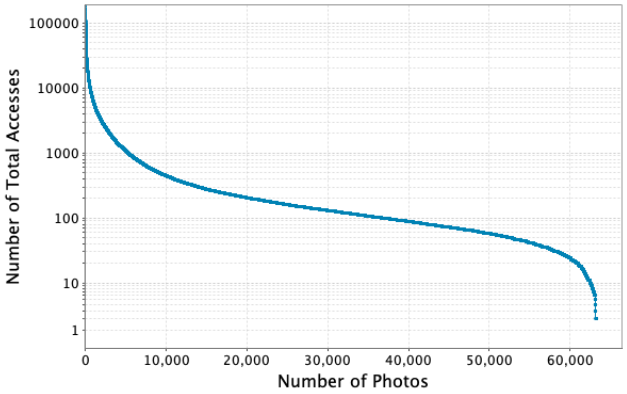

Fig. 4: Distribution of total accesses after 3,000 hours.

represent the elapsed time and the average number of access per hour, respectively. From this result, we can assume that many accesses tend to concentrate on some specific images.

Figure 4 shows the distribution of total accesses of each image after 3,000 hours elapsed. The maximum access is 182,669 times, while the minimum access is only 2 times. We took a further investigation on image accesses which describes in Table 4.

Table 4 shows the rate of total accesses in the top N% of total images compared to total number of accesses. The rate of total accesses in the top 1% of total images holds almost half of the total accesses. Again, the rate of accesses in the top 30% of total images holds almost 90% of the total accesses. From this result we can assume that most of the accesses to the images belongs to top 30% of the total images.

Based on the result we mentioned above, we consider that by storing 30% of the images in Working Disks, and the rest (around 70%) in Archiving Disks, we can reduce power consumption consuming by the whole system. The reason is that, 90% of the accesses will concentrate on the disks in Working Disks, and the workload in Archiving Disks is at rest which results in power consumption reduction.

6.3 Parameters and setting

We developed simulators based on our system design in Section 3 and power consumption model in Section 5. In our simulations, we upload 26 images [12] to the system every minute until 3,000 hours. The number of accesses mimics the access traces of images we mentioned in Section 6.1 and 6.2. Workload setting is described in Table 5.

Top N%	Number of Images	Total accesses	Percentage
1	632	17,151,201	43.4%
10	6,320	30,577,931	77.3%
20	12,640	33,747,531	85.3%
30	18,961	35,434,205	89.6%
40	25,281	36,602,858	92.6%

Table 4: Percentage of total accesses in top N% of total images compared to total accesses.

Type	Value
File size	5MB
Write	26 images per minute
Read data	500px Image access pattern
Simulation duration	3,000 hours

Table 5: Workload Setting.

In our simulations, the disks in Working Disks consists of the maximum of 8 disks, while Archiving Disks consists of unlimited disks supplied from Empty Disks (21 disks are used at the end of simulations). The number of disks in Archiving Disks increases one by one depending on the files migrated from the disks in Working Disks.

We calculate the whole power consumption of our system and compare with a simple system which the same sets of images are uploaded to an array of disks (disks increase one by one) without any file migration. The number and type (SSD and HDD) of disks using in this simple system are the same as those in proposed system.

7. Simulation Result

As we explained in Section 4, we did simulations of all the four methods. Although, the ways we stored the files in the system are different among the four methods, the results of each method showed that there are not much differences of power-saving compared to the normal system we mentioned in Section 6.3. Therefore, in this section we only describes the best result among the four techniques. The best result is 1-B technique.

Figure 5 shows the change of percentage of power consumption. Here, the horizontal and vertical axes represent the elapsed time and percentage of power saving. From 1st to 1,023th hour, the number of disks using in our method and the normal system are the same.Therefore, there is not any different of power consumption between the proposed system and the normal system since there is not any migration of files between Working Disks and Archiving Disks. The power consumption of both methods started showing the different from 1,024th hour. The reason is that the files storing by our proposed method started migrating while the files storing in normal system is static. We can see that from 1,024th hour, the power saving of the proposed method produces a better performance. Our method produced worse performance only at 1,024th hour because there were many file migrations during this hour. After 1,024th hours, our

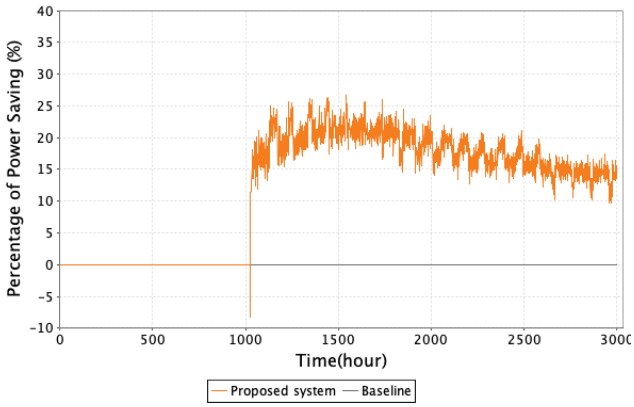

Fig. 5: The change of percentage of power-saving.

method produced better performance with the maximum of 26.6% less power consumption. Also, over 3,000 hours, this method consumed approximately 12% less energy than the normal system.

8. Conclusion and Future Work

In this study, we proposed a power-saving method in storage system for file sharing services. We proposed a system design which composes of disks with unique IDs, edge servers, an index server and I/O servers. Each disk is connected to an I/O server and is logically classified into one of three groups: Working Disks, Archiving Disks, and Empty Disks. We also introduced power consumption model to evaluate our method. In simulations, we used images from 500px for the workload. Again, we developed a normal system to compared with proposed method. The result showed that our system produced better performance when the time elapsed from 1,024th hour with an average of 12% of power saving rate. However, we did not investigate on response time of our proposed system. Thus, investigation on response time of I/O requests is needed.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 17H01718. We would like to express our sincere appreciation to Prof. Koji Hasebe, Prof. Kazuhiko Kato, and Prof. Hirotake Abe for their helpful comments and feedback that helped us improve the paper.

References

[1] D. Chaffey, Global social media research summary 2016, Smart Insights: Social Media Marketing, 2016.
[2] A. Shehbi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, ... and W. Lintner, United states data center energy usage report, 2016.
[3] D. Colarelli, D. Grunwald, Massive arrays of idle disks for storage archives, In Proceedings of the 2002 ACM/IEEE conference on Supercomputing (Supercomputing ' 02), pages 1–11, 2002.
[4] E. Pinheiro, R. Bianchini, Energy conservation techniques for disk array-based servers, In Proceedings of the 18th annual international conference on Supercomputing (ICS ' 04), pages 68–78, 2004.
[5] R.T. Kaushik, M. Bhandarkar, Greenhdfs: towards an energy-conserving, storage-efficient, hybrid hadoop compute cluster, Proceedings of the USENIX annual technical conference, p. 34, 2010.
[6] K. Hasebe, J. Okoshi, K. Kato, Power-saving in storage systems for cloud data sharing services with data access prediction, IEICE TRANSACTIONS on Information and Systems, 98.10: 1744-1754, 2015.
[7] T. Bostoen, S. Mullender, and Y. Berbers, Power-Reduction Techniques for Data-Center Storage Systems, ACM Computing Surveys (CSUR), vol.45, no.3, article no.33, 38 pages, 2013.
[8] J. Park, S. Yoo, S. Lee and C. Park, Power Modeling of Solid State Disk for Dynamic Power Management Policy Design in Embedded Systems, Vol. 5860 of Lecture Notes in Computer Science, pp. 24–35. 2009.
[9] Seagate Technology PLC, Desktop HDD Product Manual, https://www.seagate.com/www-content/product-content/barracuda-fam/desktop-hdd/barracuda-7200-14/en-us/docs/100686584y.pdf, Accessed:2018-12-13.
[10] Y.H. Lu, E.Y. Chung, T. Simunic, T. Benini, G.D. Micheli, Quantitative comparison of power management algorithms, Design, Automation and Test in Europe Conference and Exhibition, Proceedings (pp. 20-26). IEEE, 2000.
[11] Intel Corporation, INTEL® SSD DC P4610 SERIES, Technical Specifications, https://www.intel.com/content/www/us/en/products/memory-storage/solid-state-drives/data-center-ssds/dc-p4610-series/dc-p4610-1-6tb-2-5inch-3d2.html, Accessed: 2018-12-12.
[12] Getty Images, Inc. 500px, https://www.gettyimages.co.jp/collec-tions /500px, Accessed: 2018-12-17.
[13] Li, Dong, and J. Wang, "EERAID: energy efficient redundant and inexpensive disk array.", Proceedings of the 11th workshop on ACM SIGOPS European workshop, ACM, 2004.
[14] C. Weddle, M. Oldham, J. Qian, A. Wang, P. Reiher, and G. Kuenning, "PARAID: A gear-shifting power-aware RAID," ACM Transactions on Storage (TOS),Article No. 13,Volume 3 Issue 3, 2007.
[15] M. Storer, K. Greenan, E. Miller, and K. Voruganti, "Pergamum: replacing tape with energy efficient, reliable, disk-based archival storage", FAST'08 Proceedings of the 6th USENIX Conference on File and Storage Technologies, pp.1–16, 2008.
[16] 500px, https://500px.com, Accessed:2018-12-12.
[17] R.T. Kaushik and M. Bhandarkar, "GreenHDFS: towards an energy- conserving, storage-efficient, hybrid Hadoop compute cluster," Proc. 2010 International Conference on Power Aware Computing and Sys- tems, pp.1–9, 2010.