Analysis of optimal file placement for energy-efficient file-sharing cloud storage system

Fumio Machida, *Senior Member, IEEE,* Koji Hasebe, *Member, IEEE,*
Hirotake Abe, *Member, IEEE,* and Kazuhiko Kato, *Member, IEEE*

Abstract— Popular data concentration is a widely accepted storage energy-saving technique which places frequently-accessed data on a small subset of hard disks and spins-down other infrequently-accessed disks. Many previous studies use intuitive heuristic algorithms for data placement that promote the imbalance in the access frequencies across hard disks. However, the relevance and the optimality of such file placements have not been rigorously investigated. In this paper, we formally define the energy-saving file placement problem under the capacity and performance constraints as a combinatorial optimization problem and show the theory of the optimal file placement where the file access rates in the next period are given. Our analysis based on a stochastic process of disk state transitions gives the theoretical support for the common heuristic placement method. To examine the effectiveness of the optimal file placement, we experimentally evaluate the energy-efficiency of a test storage system using the file access rates generated from the real access traces from Flickr. The experimental results show that the energy consumption can be reduced by 31.8% with the optimal file placement compared to the evenly distributed file placement. We also conduct simulation experiments to confirm the energy-saving impacts in larger-scale storage systems.

Index Terms— Models, Optimization, Power management, Storage management.

1 Introduction

ENERGY-efficiency of cloud storage systems has been explored as an important challenge for service providers and enterprises owning the storage systems. According to the report "Data Age 2025" [1], the amount of world-wide data generated from devices such as IoT sensors will grow from 33 ZB in 2018 to 175 ZB by 2025. A part of those data needs to be saved or archived in data storage somewhere. Present cost-effective cloud or enterprise storage systems are expected to extend their capacity to store even a larger volume of data. With such an increasing demand for data storage, the energy cost of storage systems will be a measurable and persistent issue for the owners of the storage systems. In a data center environment, it was reported that storage systems consumed about 27% of energy, which is the second-largest component after servers and cooling systems [2]. While solid state drive (SSD) and Non-volatile memory (NVM) are gaining popularity as energy-efficient fast storage devices, hard disk drives (HDDs) are still dominant storage components used in data centers [1][3][4]. Therefore, continuous efforts to improve the energy efficiency of large-scale HDD-based storage systems will be essential in the coming data age.

Popular data concentration (PDC) [5] has been a widely adopted technology for saving the storage energy consumption by consolidating popular data into a subset of hard disks so that the other infrequently accessed disks can be shifted to lower-power modes. The main idea behind PDC is to exploit the skewness of the data or file access frequencies, which are often observed in many application domains. For instance, an image file sharing service Flickr, about 70% of files have never been accessed after upload [6]. File access analysis in Yahoo!'s enterprise Hadoop clusters observed that 60% of data was not accessed in 20 days of time window [7]. By laying such infrequently accessed data on a subset of hard disks and spinning down the disks, the storage energy consumption can be reduced significantly. Many studies on storage energy conservation techniques follow the idea of PDC and successfully reduce the energy consumptions of hard disk-based storage systems [6][7][8][9][10][11][12]. Two essential functions in the implementation of PDC are the prediction of data access frequencies and the data placement method. While the prediction is usually made by analyzing historical workload data, the placement method typically relies on a heuristic algorithm that places data on disks in order of data access frequencies [5][6][8][12]. The heuristic is intuitive, and the effectiveness was validated through some simulation studies [5][6][8]. However, none of these studies provide the theoretical background of this heuristic in terms of energy conservation. Indeed, to the best of our knowledge, the relation between the data placement with given file access frequencies and the standby state probabilities of disks that leads to energy conservation has never been theoretically investigated.

In this paper, we address the file placement problem of PDC for an energy-efficient hard disk-based storage system hosting a cloud file sharing service. The service is used to share media contents such as images, photos, and short movies that are posted and accessed by users. For a given set of files to be stored in a cloud storage system, we formulate the energy-saving file placement problem as a combinatorial optimization problem subject to the constraints

- *F. Machida is with the Department of Computer Science, University of Tsukuba, Tsukuba, Japan. E-mail: machida@cs.tsukuba.ac.jp.*
- *K. Hasebe is with the Department of Computer Science, University of Tsukuba, Tsukuba, Japan. E-mail: hasebe@cs.tsukuba.ac.jp.*
- *H. Abe is with the Department of Computer Science, University of Tsukuba, Tsukuba, Japan. E-mail: habe@cs.tsukuba.ac.jp.*
- *K. Kato is with the Department of Computer Science, University of Tsukuba, Tsukuba, Japan. E-mail: kato@cs.tsukuba.ac.jp.*

on the average file access performance and the capacity of individual hard disks. Assuming that the arrival of requests to each media file follows an independent Poisson process in a given period, the relation between the file placement and the standby state probabilities of hard disks is formally analyzed. With this formal model, we derive two important propositions on energy-saving file replacement strategy and the optimal file placement under the capacity constraint, respectively. The first proposition tells us any file migration from an infrequently access disk to a more frequently access disk always reduces the total energy consumption of the storage system. The rule is generic and can be combined with other heuristic-based file placement algorithms. Thus, the rule is useful for designing a file placement algorithm for energy-efficient storage systems. The second proposition gives the proof of the optimal file placement in which files are placed in order of file access frequency under the constraint of hard disk capacity given by the number of files on disks. The result gives a theoretical background of the previously studied heuristic-based method. It also clarifies that the necessary condition where the heuristic method achieves the optimal file placement satisfying the performance constraint. To confirm the effectiveness of the optimal file placement on a real storage system, we conducted experiments on our testbed system consisting of four disks with the real file access frequencies obtained from Flickr file access traces [8]. Our experimental results show the optimal file placement can cut 31.8% of energy overheads steadily compared with the baseline file placement in which file accesses are evenly distributed across the hard disks. Moreover, we also conducted simulation experiments to analyze the influences of the number of files and the disks on the energy-saving by the optimal file placement. Our simulation results are consistent with the results from the previous simulation study [5] that presented the energy-saving increased with the number of hard disks.

The rest of the paper is organized as follows. In Section 2, the related work on energy conservation techniques for hard disk-based storage systems is reviewed. In Section 3, we describe the target cloud storage system, which is supposed to be used for hosting a file-sharing service. In Section 4, we formulate the energy-saving file placement problem and discuss the tradeoff relation between the file access performance and storage energy consumption. In Section 5, we derive the two important propositions for the optimal file placement problem with their proofs. Section 6 shows our experimental results, and finally, Section 7 gives our conclusion and possible future directions.

2 Related work

There has been a rich body of existing studies for energy-saving techniques for data center or cloud storage systems. Earlier studies on these techniques were summarized in a survey paper [13] in which storage systems are not limited to hard disk-based ones. The following review mainly focuses on the energy-saving techniques for storage system consisting of hard disks. In terms of energy-efficiency of data centers, many existing studies focused on server resource management techniques [14][15][16][17], where workload allocation is the key problem to solve. Our study differs from these studies, since energy conservation techniques for hard disks relying on the mechanism of hard disks which are not appropriate for servers [18].

The common idea for reducing the energy consumption in a hard disk-based storage system is to exploit the disk idle time to power-off or speed-down the multi-speed disks. The approach is also referred to as dynamic power management (DPM) [19]. One of the direct methods to increase the disk idle time is to prepare the segregated set of disks for infrequently accessed data. MAID [20] is a storage design that separates the archival data disks from the cache disks to spin-down the data disks when they are in idle states [20]. PDC [5] uses data migration across hard disks periodically according to the data popularity to consolidate the frequently-accessed data. For increasing the disk idle time by data replacement, it is essential to estimate the data access frequencies to find a desirable data placement. Since data access patterns depend on the application workloads, many studies attempted to characterize the application workloads and used them for energy-efficient data management [6][7][8][11][21]. For example, based on the workload analysis of Yahoo!'s Hadoop cluster, GreenHDFS [7] allocates the disks either hot or cold zone, and replace the data according to the age of data. Hasebe et al. [8] used file access traces observed in Flickr to derive the individual file access frequencies and proposed the file exchange algorithm to skew the disk access frequencies. Some recent studies further took into account the correlation of file accesses to determine the file placement for energy saving [36] [37]. In this paper, we consider a cloud-based file sharing service as a target application. However, the proposed model and the optimal file placement analysis are generally applicable to other applications whose file access rates are predictable in advance.

Instead of migrating data, which often imposes additional costs, other studies proposed techniques that control data accesses for the storage system to avoid spinning-up cold disks. The techniques combining data replication and access diversion are presented in many existing studies [9][22][23][24][25][27][28]. For instance, DIV [24] and PARAID [9] presented replica placement techniques to save energy by maintaining high-availability using RAID configuration. SRCMap [25] leveraged a storage virtualization method to replicate the frequently accessed data volume so that data accesses can be diverted to the replicas on the active volumes. Narayanan et al. presented write off-loading [10], which allows write requests on standby disks to be temporarily redirected to other persistent storage. Since our theoretical study mainly focuses on the PDC method, we do not consider the replicated data and diversion of data accesses. However, the impact of data replication and high-availability constraint can be considered in a future extension of our model and analysis.

Compared with the storage energy-saving techniques, the studies on storage power modeling and measurements have received less attention [30]. Allalouf et al. presented the fine-grained storage model to estimate the power consumption of storage workloads [30]. The model of disk idle time under the parallel video-sharing workloads is presented by Yuan et al. [31]. The optimization problems to minimize the power consumption of a disk-based storage system [32][33] and hard disk with SSD storage system [35] were formulated using disk power state models. PREFiguRE [34] is presented as an analytic framework that uses the histogram of disk idle times to determine schedules for power saving modes. None of the work, however, did present the model that associates the data placement and the disk standby state probabilities for estimating storage power consumption. Our model, introduced in Section 5, is the first to describe this association formally.

3 Cloud storage system for a file sharing service

In this section, we describe the requirements and the architecture of a cloud storage system for a file-sharing service.

3.1 System requirements

We consider online file-sharing services where any users can upload files and share the content with other users. The assumed contents are media files such as images, photos, and short movies. YouTube, TikTok, Instagram, and Flickr are well-known popular services providing such a file-sharing service. There are a number of content-specific or community-oriented file-sharing services as well (e.g., Snapchat, 500px, SmugMug, and Vimeo).

To attract many users for such file-sharing services, the service providers need to keep their systems in a good performance. Service response time is one of the key metrics which most users perceive when accessing the content. Users do not continue using the service when they feel impatient in waiting for the responses to their requests. For e-Commerce sites, it is reported that 40% of users would leave the sites if it takes more than two seconds to get the response [38]. Service providers need to define a relevant service level to meet users' expectations.

Besides keeping the service in a good performance, it is also an important challenge for service providers to reduce the total cost of the system. In a data center or a cloud environment, the total system cost can broadly be divided into capital expenditure (CapEx) and operational expenditure (OpEx)[39]. In terms of a storage system, the initial purchase of storage hardware is included in CapEx. The available capacity of the storage system should be constrained by the initial budget for CapEx. On the other hand, the cost of maintaining the storage system appears in OpEx, which may include software license fees, power usage costs, supplies expenses, and human labor costs. Although the energy usage cost cannot be exempted from the OpEx as long as continuing the service, it is important to reduce the wasted power usage that is not necessary for providing the service. Any disks which store infrequently-accessed data must be the target of energy-saving.

3.2 Workload characteristics

We assume that most accesses to a file-sharing service are read accesses, which are randomly requested from service users. Once a user uploads a file such as a photo or a movie taken by users' smart device, the file can be deleted later but is seldom modified or replaced. Under such usages of the service, write accesses to the storage system are limited to the time when files are uploaded. Other accesses are mostly read accesses aiming to look up the uploaded files by different users. The frequency of file accesses depends on the popularity of the content. While a few popular files are accessed very frequently, a large part of uploaded files are seldom or never accessed. According to the previous study [6], for randomly selected 20,000 photos in Flickr, about 70% of photos are never being accessed after upload. Such web access patterns are commonly observed in other web services and are often approximated by Zipf distributions [40][41]. In our experimental study in Section 6, we also use the access traces of Flickr investigated in the previous studies [6][8].

3.3 Cloud storage architecture

From the system requirements and workload characteristics of the file-sharing service described above, we consider a commonly adopted energy-saving cloud storage architecture consisted of hard disks that can shift to standby mode. Following the existing studies [7][20][42][43], we assume the system consists of active (hot) storage and archive (cold) storage. Active storage is mainly used for storing recently uploaded and frequently-accessed files, and thus their hard disks are always in active mode. On the other hand, archive storage hosts less-frequently accessed files so that their hard disks can shift to standby mode for energy-saving. Our target is archive storage, whose hard disks can be switched between active and standby states depending on the file access frequencies.

For conserving the energy by an efficient file placement on archive storage, it is essential to estimate the individual file access frequencies. Energy-saving storage architecture using PDC often maintains the list of files access frequencies and uses the information to determine the file replacement [5][8][12]. PDC can effectively increase the disk idle time in the archive storage if the estimation of file access frequencies is accurate. We can reduce the total energy consumption by spinning down the disks that remain in an idle state for a specific period. The threshold to determine a spin-down for a hard disk is a design parameter of the architecture. A short threshold time may cause frequent spin-ups that require relatively large power consumptions compared to keeping idleness. On the other side, a long threshold time to spin-down may reduce the chance of disk energy conservation. This tradeoff can be considered through the computation of break-even time [42], which derives a reasonable threshold time for energy-saving. In our study, we assume that the threshold time is given as a constant parameter determined by a break-even time analysis. In this architecture, the key to energy conservation is the file placement that is constrained by the performance requirements, the disk capacity limits, the current placement, and the file migration costs. Since our goal in this paper is to analyze the optimal file placement given the information of file access frequencies, we do not consider the file migration costs that will be addressed in future work.

4 Problem formulation

We formally define the energy-saving file placement problem for a file-sharing cloud storage system.

4.1 Definition

Consider a cloud storage system consisting of a set of hard disks $M = \{1,2,\dots,m\}$ that stores a set of files $N = \{1,2,\dots n\}$. The cloud system hosting a file sharing service may have cache servers or dedicated cache storage systems in addition to the target storage system, but hereafter we focus on a hard-disk based nearline storage system which we simply refer to *cloud storage system*. The file placement to hard disks can be represented by a mapping function $\phi: N \to M$. File replacement is performed periodically at a specific time interval. Given predicted file access rates in the next time interval, our goal is to find the optimal file placement ϕ_{opt} that minimizes the total energy consumption of the system under the performance and the capacity constraints. For mathematical tractability, we make the following assumptions to the system.

A1. In the focused time interval, file accesses occur in a Poisson process with a constant arrival rate λ_i which is proportional to the popularity of file i.

A2. The expected energy consumptions in active and standby state of a hard disk for a unit time are given as P_a and P_s, respectively, where $P_a > P_s$.

A3. Average response times for a file in an active and a standby disk are given by T_a and T_s, where $T_a < T_s$.

A4. The total average response time for file accesses to the cloud storage system should be less than T_{req}, where $T_a < T_{req} (< T_s)$.

A5. A disk shifts to a standby mode when a threshold time τ passes after the last file access on the disk, while the disk returns to an active mode when file access occurs in the standby state.

A6. The maximum number of files placed on a disk $j \in M$ is constrained by the capacity c_j, while the total capacity is enough for hosting all the files $\sum_{j \in M} c_j \geq n$, where n is the number of files to be placed.

With the above assumptions and constraints, we define the problem as follows.

Energy-saving file placement problem

Given a set of files N whose accesses are given by Poisson arrivals, and a cloud storage system consisting of a set of disks M whose states are changed by file accesses as well as the threshold time τ to change a disk in standby mode, find the optimal file placement $\phi_{opt}: N \to M$ that minimize the total energy consumption of the system, under the constraints on the required average response time T_{req} and the capacity limit $c_j, j \in M$.

While Poisson arrivals were observed in real systems [45][46], the validity of the first assumption A1 on the constant file access rate is arguable, since the popularity of sharing files can change over time as observed in real traces [8]. The solution to the energy-saving file placement problem only gives the best placement under the predicted file access rates for the next period.

In constructing the disk capacity constraint in A6, we assume that file sizes do not vary much among the files. While the assumption of averaged file size is acceptable when we deal with a large number of files, there may be a correlation between access patters and file sizes. Such correlations may impact on optimal file placement. However, in this paper, we do not consider such a case since the popularity of file content is usually the major factor of file access frequencies, especially in online file-sharing services.

4.2 Response time constraint

It is noted that there is a tradeoff between the response time to file access and disk energy consumption. A long standby period of a disk can reduce the expected total energy consumption, while it has a negative influence on the average response time of the file accesses. Due to this tradeoff, the minimization of energy consumption is constrained by the response time requirement T_{req}. The constraint can be derived from the following tradeoff analysis. Let $\pi_s^{(j)}$ be the probability of standby state for disk j. The expected average response time for disk j with file placement ϕ can be represented as

$$E\left[T_{res}^{(j)} | \phi\right] = T_a\left(1 - \pi_s^{(j)}\right) + T_s \pi_s^{(j)}. \tag{1}$$

Let $w^{(j)}$ be the probability that the requested file is stored in disk j, which satisfies $\sum_j w^{(j)} = 1$. The expected average response time of the cloud storage system is given by

$$E[T_{res} | \phi] = \sum_j w^{(j)} E\left[T_{res}^{(j)} | \phi\right]. \tag{2}$$

From the response time requirement, $E[T_{res} | \phi]$ needs to be less than or equal to T_{req}. Therefore, applying (1) to (2) and rearranging the terms, we obtain the following constraint on the probabilities of standby states for the disks.

$$\sum_j w^{(j)} \pi_s^{(j)} \leq \frac{T_{req} - T_a}{T_s - T_a}. \tag{3}$$

The constraint indicates that the individual standby state probabilities are bounded from above by T_{req}. On the other hand, given the probabilities of standby state for individual disks, the expected total energy consumption of the cloud storage system can be represented by

$$E[P | \phi] = m^+ P_a - (P_a - P_s) \sum_j \pi_s^{(j)}. \tag{4}$$

where $m^+ (\leq m)$ represents the number of disks that host at least one file as a result of the file placement ϕ. To minimize the expected total energy consumption, the aggregated probability of disk standby states should be maximized. However, individual standby state probabilities need to satisfy the response time constraint (3) for the optimal file placement ϕ_{opt}.

In a practical case, the probability that a request meets the minimum response time constraint may be used as a performance indicator instead of the average response time. Even when the service level is specified by such an indicator, the optimal file placement is constrained from the tradeoff between the response time and the energy consumption. We derive a constraint for this case in the Appendix, while we focus on the average response time constraint in the following discussion.

5 Optimal file placement analysis

To derive the optimal solution to the energy-saving file placement problem, first, we analyze the relation between the file access rates and the standby state probabilities of hard disks. The expected total energy consumption of a cloud storage system can be represented by a function of the aggregated file access rates determined by a given file placement. Next, we present a general file migration rule for enhancing the energy conservation of the cloud storage system by file placement. Finally, we theoretically discuss the optimal file placement that minimizes the expected energy consumption of the cloud storage system.

5.1 Disk state analysis

The steady-state probability of a hard disk in a standby state can be derived from given file access rates and a threshold time to become a standby mode. Since our focus is energy conservation by shifting idle disks to standby modes, we simply divide the states of a hard disk into an active and a standby state and assume that the state changes between these states. A state transition from an active state to a standby state occurs when the elapsed time from the last disk access exceeds the threshold value τ. On the other hand, a state transition from a standby state to an active state occurs when any files on the disk are accessed. Since both the state transitions are associated with random arrival of file accesses, the times for state transitions are stochastically distributed. We denote the random variables for the time to a standby state and an active state as X_{as} and X_{sa}, respectively, that are assumed to be independent and identically distributed. Then the stochastic process can be seen as an alternating renewal process [47] whose steady-state probabilities are given by

$$\pi_a = \frac{E[X_{as}]}{E[X_{as}] + E[X_{sa}]}, \qquad \pi_s = 1 - \pi_a, \tag{5}$$

where π_a and π_s represent the steady-state probabilities of active and standby states, respectively.

Let F_j represent the set of files stored in disk j. The file accesses on disk j can be characterized by the Poisson process with rate

$$\lambda^{(j)} = \sum_{i \in F_j} \lambda_i. \tag{6}$$

Thus, the mean time to disk access for disk j is $1/\lambda^{(j)}$, which is independent of the disk state. Since any accesses to a standby disk make the disk spin-up, the expected time in a standby state is given by $E[X_{sa}] = 1/\lambda^{(j)}$. Now we assume that there are r disk accesses before the state transition from an active state to a standby state, the expected state transition time can be given by

$$E[X_{as}] = E\left[r \cdot \frac{1}{\lambda^{(j)}} + \tau\right] = E[r] \cdot \frac{1}{\lambda^{(j)}} + \tau. \tag{7}$$

Let p be the probability that disk access occurs before the threshold value τ. Since the time to next disk access is exponentially distributed with rate $\lambda^{(j)}$, p can be expressed as $1 - e^{-\lambda^{(j)}\tau}$. The probability that the disk accesses occur $r - 1$ times and r-th disk access does not occur before the threshold τ follows a modified geometric distribution of $1 - p$. Therefore, the expected number of disk accesses before a state transition to standby is

$$E[r] = \frac{p}{1-p} = \frac{1 - e^{-\lambda^{(j)}\tau}}{e^{-\lambda^{(j)}\tau}}. \tag{8}$$

Applying expression (8) to (7) and (5) the steady-state probabilities of a hard disk in an active and a standby state are expressed as

$$\pi_a^{(j)} = 1 - \frac{e^{-\lambda^{(j)}\tau}}{1 + \lambda^{(j)}\tau e^{-\lambda^{(j)}\tau}}, \pi_s^{(j)} = \frac{e^{-\lambda^{(j)}\tau}}{1 + \lambda^{(j)}\tau e^{-\lambda^{(j)}\tau}}. \tag{9}$$

The above steady-state probabilities of disk states play a central role in the analysis of optimal file placement since the expression clarifies the relation between the aggregated file access rates on the hard disks and the probability of disk standby states which leads to energy conservation.

5.1 File migration rule

Migrating files among hard disks is a key to save storage energy using PDC. While the conventional file migration algorithms for PDC implicitly attempts to increase the imbalance in file access frequencies, the validity of such a heuristic has not been theoretically investigated. Based on the steady-state probabilities of hard disks derived in the previous section, we present a general file migration rule for reducing the expected energy consumption of a cloud storage system. The formal description of this rule is provided in the following proposition with proof.

Proposition 1

For a pair of disks (j_1, j_2) satisfying $\lambda^{(j_1)} \geq \lambda^{(j_2)}$, any file migration from disk j_2 to disk j_1 always reduces the expected energy consumption of a cloud storage system when the steady-state probabilities of individual hard-disks are given by (9).

Proof

Since the expected energy consumption of the cloud storage system is given by (4), it is enough to show $\sum_{j \in M^+} \pi_s^{(j)}$ is increased by the file migration. Moreover, since the file placement on the other disks are not changed, we can focus on disks j_1 and j_2, and show $\pi_s^{(j_1)} + \pi_s^{(j_2)}$ increases by the file migration. Denote $\pi_s^{(j_1')}$ and $\pi_s^{(j_2')}$ as the steady-state probabilities of standby states for disks j_1 and j_2, respectively, after the file migration. The difference between the total steady-state probabilities before and after the file migration is expressed by

$$\begin{aligned} \Delta\pi_s &= \pi_s^{(j_1')} + \pi_s^{(j_2')} - \left(\pi_s^{(j_1)} + \pi_s^{(j_2)}\right) \\ &= \pi_s^{(j_2')} - \pi_s^{(j_2)} - \left(\pi_s^{(j_1)} - \pi_s^{(j_1')}\right). \end{aligned} \tag{10}$$

Denote λ_x as the access rate of the file to be migrated. Since the disk access rate of j_1 after the file migration is represented by $\lambda^{(j_1')} = \lambda^{(j_1)} + \lambda_x$, from expression (9) we have

$$\begin{aligned} &\pi_s^{(j_1)} - \pi_s^{(j_1')} \\ &= \frac{e^{-\lambda^{(j_1)}\tau}}{1 + \lambda^{(j_1)}\tau e^{-\lambda^{(j_1)}\tau}} - \frac{e^{-(\lambda^{(j_1)} + \lambda_x)\tau}}{1 + (\lambda^{(j_1)} + \lambda_x)\tau e^{-(\lambda^{(j_1)} + \lambda_x)\tau}} \\ &= \frac{e^{-\lambda^{(j_1)}\tau}\left(1 + \lambda_x \tau e^{-(\lambda^{(j_1)} + \lambda_x)\tau} - e^{-\lambda_x\tau}\right)}{\left(1 + \lambda^{(j_1)}\tau e^{-\lambda^{(j_1)}\tau}\right)\left(1 + (\lambda^{(j_1)} + \lambda_x)\tau e^{-(\lambda^{(j_1)} + \lambda_x)\tau}\right)} \end{aligned}$$

$$= \frac{e^{-\lambda^{(j_1)}\tau}\left[1 - e^{-\lambda_x\tau}\left(1 - \lambda_x\tau e^{-\lambda^{(j_1)}\tau}\right)\right]}{\left(1+\lambda^{(j_1)}\tau e^{-\lambda^{(j_1)}\tau}\right)\left(1+(\lambda^{(j_1)}+\lambda_x)\tau e^{-(\lambda^{(j_1)}+\lambda_x)\tau}\right)} \quad (11)$$

Similarly, the disk access rate of j_2 after the file migration is represented by $\lambda^{(j_2')} = \lambda^{(j_2)} - \lambda_x$. From expression (9) we have

$$\begin{aligned}
&\pi_s^{(j_2')} - \pi_s^{(j_2)} \\
&= \frac{e^{-\lambda^{(j_2')}\tau}}{1+\lambda^{(j_2')}\tau e^{-\lambda^{(j_2')}\tau}} - \frac{e^{-(\lambda^{(j_2')}+\lambda_x)\tau}}{1+(\lambda^{(j_2')}+\Delta\lambda)\tau e^{-(\lambda^{(j_2')}+\lambda_x)\tau}} \\
&= \frac{e^{-\lambda^{(j_2')}\tau}\left(1+\lambda_x\tau e^{-(\lambda^{(j_2')}+\lambda_x)\tau} - e^{-\lambda_x\tau}\right)}{\left(1+\lambda^{(j_2')}\tau e^{-\lambda^{(j_2')}\tau}\right)\left(1+(\lambda^{(j_2')}+\lambda_x)\tau e^{-(\lambda^{(j_2')}+\lambda_x)\tau}\right)} \\
&= \frac{e^{-\lambda^{(j_2')}\tau}\left[1-e^{-\lambda_x\tau}\left(1-\lambda_x\tau e^{-\lambda^{(j_2')}\tau}\right)\right]}{\left(1+\lambda^{(j_2')}\tau e^{-\lambda^{(j_2')}\tau}\right)\left(1+(\lambda^{(j_2')}+\lambda_x)\tau e^{-(\lambda^{(j_2')}+\lambda_x)\tau}\right)} \quad (12)
\end{aligned}$$

Note that the expression 0 and (12) are the same function defined by

$$f(z) = \frac{e^{-z\tau}[1-e^{-\lambda_x\tau}(1-\lambda_x\tau e^{-z\tau})]}{(1+z\tau e^{-z\tau})(1+(z+\lambda_x)\tau e^{-(z+\lambda_x)\tau})}, \quad (13)$$
$$z > 0, \tau > 0.$$

Taking the derivative of $f(z)$ and manipulating the expressions, we can show $df(z)/dz < 0$ in $z > 0, \tau > 0$. Since $f(z)$ is a monotonically decreasing function and $\lambda^{(j_1)} > \lambda^{(j_2')}$, we have $f(\lambda^{(j_1)}) < f(\lambda^{(j_2')})$. From expression (10), $\Delta\pi_s = f\left(\lambda^{(j_2')}\right) - f\left(\lambda^{(j_1)}\right) > 0$. Therefore, the total expected energy consumption after the file migration is larger than the expected energy consumption before the migration. ■

This file migration rule provides a very powerful guideline for designing a heuristic file migration algorithm aiming at energy-saving. Indeed, previous studies on PDC approach use this rule without a thorough discussion on this point [5][6][8][12]. Our proposition first gives theoretical support for this rule of thumb.

The file migration rule can be extended to the file exchange rule between two hard disks that can reduce the expected energy consumption of a cloud storage system in the following corollary.

Corollary 1

For a pair of disks (j_1, j_2) satisfying $\lambda^{(j_1)} > \lambda^{(j_2)}$, when we exchange the file x_1 on disk j_1 with the file x_2 on disk j_2 satisfying $\lambda_{x_1} < \lambda_{x_2}$, the expected energy consumption of a cloud storage system always decreases when the steady-state probabilities of individual hard-disks are given by (9).

Proof

Since only disk j_1 and disk j_2 are affected by the file exchange, it is enough to show $\pi_s^{(j_1)} + \pi_s^{(j_2)}$ increases after the file exchange. When we denote $\pi_s^{(j_1')}$and $\pi_s^{(j_2')}$ as the steady-state probabilities of standby states for disk j_1 and disk j_2, respectively, after the file exchange, the difference between the total steady-state probabilities before and after the file exchange is expressed by (10). Using the notation $\Delta\lambda = \lambda_{x_2} - \lambda_{x_1} > 0$, we have $\lambda^{(j_1')} = \lambda^{(j_1)} + \Delta\lambda$ and $\lambda^{(j_2')} = \lambda^{(j_2)} - \Delta\lambda$. Replacing λ_x in the proof of proposition 1 with $\Delta\lambda$, we can derive the same function (13) and reach the same conclusion $\Delta\pi_s > 0$. Therefore, the total expected energy consumption after the file exchange is larger than the expected energy consumption before the file exchange. ■

The rule ensures that any file exchange enlarging the difference of disk access rates between two disks can reduce the total expected energy consumption. The rule is applicable to any file pairs that satisfy the condition on the file access rates (i.e., $\lambda^{(j_1)} > \lambda^{(j_2)}$ and $\lambda_{x_1} < \lambda_{x_2}$). The file exchange rule is used in the proof of optimal file placement discussed in the next section.

5.2 Optimal file placement

To derive the optimal solution to the energy-saving file placement problem, the constraints of hard disk capacity and performance requirements need to be taken into account. First, we incorporate the hard disk capacity constraint and show the file placement that minimizes the total energy consumption without considering the performance requirement given in (3). Assume that the disks in M are sorted in descending order of the capacity, i.e., $c_1 \geq c_2 \ldots \geq c_m$. We present the following proposition for the file placement which minimizes the total energy consumption of the cloud storage system.

Proposition 2

Let $\xi: N \rightarrow \mathbb{N}$ be the function to return the rank of file i by descending order of the access rate λ_i. The file placement function ϕ_{min} defined below minimizes the total energy consumption of a cloud storage system.

$$\phi_{min}(i) = \begin{cases} 1, & \xi(i) \leq c_1 \\ j, & 1 < j \leq m, \quad \sum_{k=1}^{j-1} c_k < \xi(i) \leq \sum_{k=1}^{j} c_k\,. \end{cases}$$

To give a proof of proposition 2, we first clarify the property of the placement ϕ_{min} as in the following lemma.

Lemma 1

In the file placement given by ϕ_{min}, there is no file pair $(x_1, x_2), \phi_{min}(x_1) \neq \phi_{min}(x_2)$ that can reduce the expected energy consumption by exchanging x_1 with x_2 under proposition 1.

Proof of lemma 1

Since the access rate of disk j is given by $\lambda^{(j)} = \sum_{\phi_{min}(i)=j} \lambda_i$, by the definition of ϕ_{min}, for any pairs of disks $(j_1, j_2), j_1 < j_2$, the access rate of each disk satisfies $\lambda^{(j_1)} \geq \lambda^{(j_2)}$. On the other hand, for any pairs of files (x_1, x_2) satisfying $\phi_{min}(x_1) < \phi_{min}(x_2)$, the file access rates must hold the condition $\lambda_{x_1} \geq \lambda_{x_2}$ because the files are sorted in descending order. As a result, under the file placement ϕ_{min}, there does not exist a file pair (x_1, x_2) that satisfies both the conditions $\lambda_{x_1} < \lambda_{x_2}$ with $\phi_{min}(x_1) < \phi_{min}(x_2)$. ■

Lemma 1 shows that the expected energy consumption of the cloud storage system with the file placement ϕ_{min} cannot be reduced by applying corollary 1. This is used in the following proof of proposition 2 by contradiction.

Proof of proposition 2

Assume that there exists a file placement $\phi'(\neq \phi_{min})$ that satisfies $\mathrm{E}[P|\,\phi_{min}] > \mathrm{E}[P|\phi']$. When we sort the disks under the file placement ϕ' in descending order of the disk access rate, ϕ' becomes equivalent to ϕ_{min} if $\phi'(x_1) < \phi'(x_2)$ holds for any file pair (x_1, x_2) on different disks satisfying $\lambda_{x_1} > \lambda_{x_2}$. Since $\phi' \neq \phi_{min}$ by the assumption, there is at least a pair (x_1, x_2) satisfies $\phi'(x_1) > \phi'(x_2)$ and $\lambda_{x_1} > \lambda_{x_2}$. However, in this case, the expected energy consumption can be reduced by exchanging the files x_1 and x_2 from corollary 1. This contradicts the assumption. ■

As proved above, the file placement ϕ_{min} gives the file placement which minimizes the total energy consumption where the capacity constraint is given by the number of files for individual disks (i.e., $c_j, j \in M$). In practice, the capacity constraints are determined by several factors such as I/O bandwidth, hard disk capacity in bytes, and standby threshold time. When we consider these factors, proposition 2 does not guarantee the optimality, and hence we may need to rely on approximate solutions for such cases. Meanwhile, the file replacement rule in proposition 1 is generally applicable as long as file exchange is allowed under the capacity constraints.

Similarly, file placement ϕ_{min} cannot be the optimal solution to the energy-saving file placement problem when it does not satisfy the performance requirement. As discussed in Section 4.2, for the given response time requirement T_{req}, the probabilities of standby states for disks in the cloud storage system need to satisfy the constraint (3). Since the access rate of disk j is $\lambda^{(j)}$, the probability that the requested file is stored in disk j is given by

$$w^{(j)} = \frac{\lambda^{(j)}}{\sum_j \lambda^{(j)}} = \frac{\lambda^{(j)}}{\Lambda}. \tag{14}$$

where we denote $\Lambda = \sum_{i \in N} \lambda_i$. Applying (14) to (3), we obtain the following corollary to show the necessary condition where ϕ_{min} gives the optimal solution.

Corollary 2

The file placement function ϕ_{min} defined in proposition 2 gives a solution to the energy-saving file placement problem, if and only if the expected disk standby state probabilities satisfy

$$\sum_j \lambda^{(j)} \pi_s^{(j)} \leq \frac{T_{req} - T_a}{T_s - T_a} \Lambda. \tag{15}$$

If the condition (15) is not satisfied, ϕ_{min} is not the optimal solution due to the violation of the performance requirement T_{req} while it theoretically minimizes the total energy consumption. It is not a trivial issue to derive a general solution for this case theoretically. However, in practice, there are many alternative solutions one can consider. One of the easiest solutions is to relax some system constraints such as the capacity of the hot storage to accommodate more files on active disks. One can also resort to a nonlinear optimization technique or to develop a heuristic method to find the optimal or near-optimal solutions so as not to violate the constraint (15). Although finding a feasible solution for such a case is not the main scope of this paper, we present how likely the violation of performance requirements can happen in our simulation experiments.

6 Experiments

In this section, we evaluate the effectiveness of the optimal file placement for energy conservation of a cloud storage system through real experiments and simulation studies. First, we conducted workload tests for a testbed storage system with a given set of files and measured the energy consumptions by using different file placements. The experimental results show that the energy consumption of the storage system can be saved by the optimal file placement. To analyze the impact of file placement in larger-scale storage systems, next, we conducted simulation experiments in which the expected energy consumptions by the optimal file placement are evaluated. The impacts of the uncertainty of estimated file access rates are also studied. Finally, we examine the expected average response time given by the optimal file placement.

6.1 Experimental system setup

The purpose of the experimental study is to validate the energy conservation effect by different file placements to the storage system. While many previous studies related to PDC evaluated the energy consumption by fully simulation-based [5][6][7][8][11] or indirect measurement [12], we evaluate the total power consumption of the test storage system directly by a dedicated device Watt Checker. The device taps the power supply to storage nodes at the power outlets and measures the real power usages. The measured data is stored in the device and can be retrieved through Bluetooth connection. The target storage system consists of four storage nodes, each of which has a 500GB of hard disk for file storage. All the storage nodes are connected to the same local area network through which remote file accesses arrive from a client node. We control the states of hard disks by hdparm. To determine the threshold time τ, we conducted a preliminary experiment that traces the power consumption during the state change from standby to active. From the measured power consumptions, we derive the break-even time as 15 seconds and use this value for the following experiments.

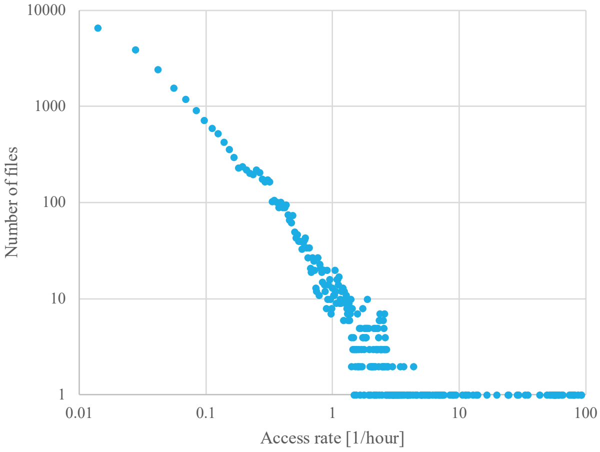

Figure 1. The number of Flickr files against the file access rates

We used Flickr access logs collected in the study [8] to generate the test workloads for file accesses. Specifically, three days of access logs on July 14 – 16, 2012, for 47,835 image files are sampled for creating an empirical distribution of file access frequencies. In Figure 1, the number of files that have the same range of file access rate (i.e., the number of file accesses per hour in three days) is plotted. The plot exhibits how the file accesses are concentrated on a few very popular files, and also shows the majority of files are less frequently accessed. Indeed, about half of the files (23,480 files) are not accessed in this period, although they do not appear in the double logarithmic plot.

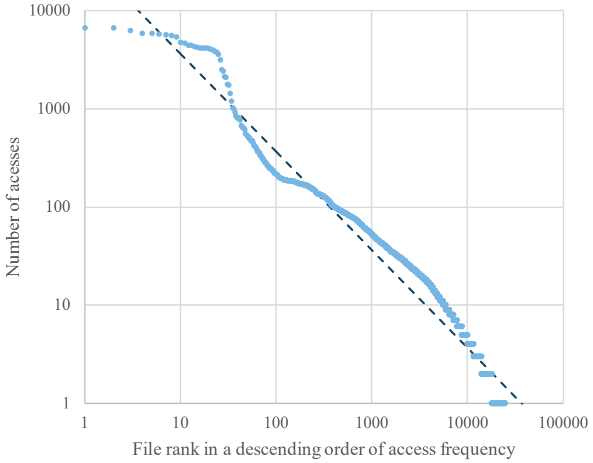

Figure 2. The relation between the file rank of access frequency and the number of actual file accesses

For the same data set, Figure 2 plots the relation between the file rank in descending order of access frequency and the number of actual file accesses during the period. The curve may be roughly approximated by Zipf distribution which is represented by a dotted line. In the experiments, we assume that the individual file access frequencies follow this empirical distribution and used four copies of the same dataset resulting in 191,340 files in total. The capacity of a hard disk is set to 47,835 files so that all the disks need to host an equal number of files. The size of each file is assumed to be 2.65MB, as it was the average file size observed in the Flickr trace [8]. Based on the individual file access frequencies, file access requests are generated on the client node and sent to the corresponding hard disk following to the given file placement. The file read is simply done by `cat` command via `ssh`.

6.2 Energy conservation by file placement

We evaluate the energy consumptions of the storage system under the same workload with different file placements. Besides the optimal file placement ϕ_{min} given in Section 5, *balanced placement* and *bisection placement* are used as the baseline methods. In the balanced placement, the files are distributed across the hard disks such that the workloads are equally balanced. With our test dataset, we simply place one copy of the original file set for each disk. Since there are no differences in file access rates among hard disks, the balanced placement is antithetical to the optimal file placement. The bisection placement is in between the balanced and the optimal placement but is a frequently used method that segregates active/hot disks from archive/cold disks [7][20][42][43]. In our testbed system, we reserve two disks for hot storage on which most frequently access files are stored and assign the other two disks for cold storage. The files in the hot or cold storage are distributed across hard disks such that the workloads are balanced in the individual category (hot or cold). Note that the number of files stored in each disk is equal to 47,835 regardless of the file placements. TABLE I summarizes the individual disk access rates in accesses per second that are the aggregated file access rates by the different file placements.

TABLE I. DISK ACCESS RATES BY DIFFERENT FILE PLACEMENTS (ACCESSES PER SECOND)

	balanced	*bisection*	*optimal*
node 1	1.597546	3.191717	6.079602
node 2	1.597546	3.191717	0.303831
node 3	1.597546	0.003376	0.006752
node 4	1.597546	0.003376	0

Using this file access rates for different file placements, we measured the ten hours of power usages of the storage system. The actual power overheads against the power consumption of the idle storage system are plotted in Figure 3. As can be seen, there is a clear distinction between power overheads among different file placements. Since we use constant file access rates during the experiments, the power overheads are increasing monotonically. The optimal file placement can cut 31.8% of energy overheads steadily compared with the balanced file placement. Compared to the bisection placement, the optimal file placement can reduce 13.3% of energy overheads. The measured energy is not solely due to hard disks but also contains other factors such as CPU power. We emphasize that the experiments are conducted under the same amount of workload, and hence the difference of energy overheads purely comes from the file placements.

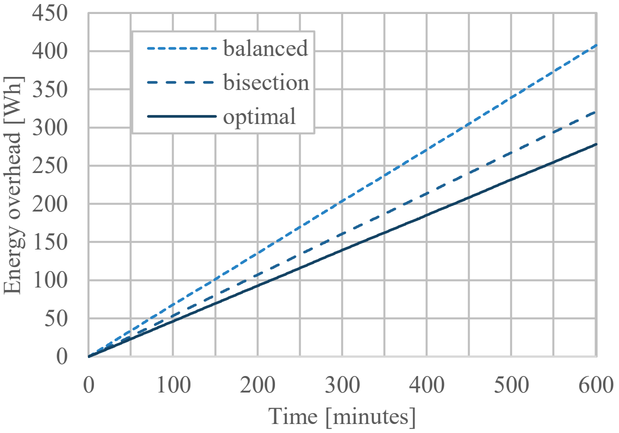

Figure 3. The observed power overheads against the power consumption of the idle storage system

6.3 Simulation experiments

To evaluate the effectiveness of the optimal file placement in larger-scale systems, we conduct simulation experiments using the energy model derived in Section 4. Assuming that the file access rates follow the empirical distribution used in the previous experiment, we synthetically generate n samples ranging from 10^4 to 10^8 to compose the population of files. The number of disks is fixed to a hundred by adjusting the disk capacity to n /100. For the bisection placement, fifty disks are used for hot storage, while the remained fifty disks are assigned for cold storage.

The expected energy consumptions by an active hard disk and a standby hard disk are set to 33.47 W and 19.38 W, respectively, from the observed average values. By varying n, the expected total energy consumptions of the storage system are computed for three file placement methods. For each value of n, simulations are conducted ten times, and then the mean values of the expected energies are plotted in Figure 4.

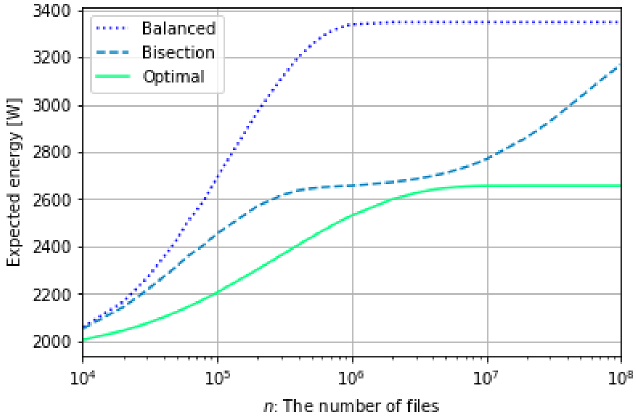

Figure 4. Expected energy consumptions vs. the number of files

The optimal file placement consistently achieves the lowest energy consumptions in the whole range of n. The difference between the energy consumptions by the balanced placement and those by the bisection placement is increasing with the number of files until around $n = 10^5$. When $n > 10^6$, by the balanced placement, all the disks are almost always active states, and hence the values of expected energy reach its upper limit. The expected energy consumption by the optimal file placement does not increase much in the range $n > 5 \times 10^6$. The phenomenon is caused by the given distribution of file access rates where 49% of files are never accessed even in larger n. The optimal file placement can consolidate these cold files into the cold disks and spin-down the cold disks. It is also noted that the bisection placement does not have such property, because the files are distributed evenly in the cold set of disks.

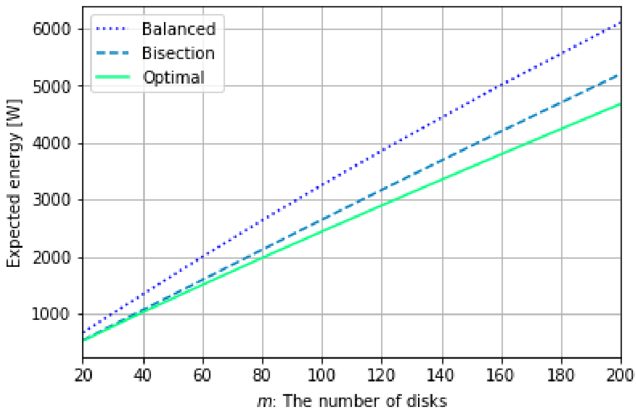

Figure 5. Expected energy consumptions vs. the number of disks

Next, we fix the number of files to 5×10^5 and vary the number of disks from 20 to 200. Note that the capacity of hard disks is changed from 2.5×10^4 to 2.5×10^3 according to the number of disks. The mean values of expected energy consumptions by three file placement methods are plotted in Figure 5. For all the file placement methods, we observed that the expected energy consumptions are proportional to the number of disks. As the scale of the storage system increases, the amount of energy-saving by the optimal file placement also increases. While the simulation settings are different, the results are generally consistent with the previous simulation study that showed the energy-saving by PDC becomes more significant when increasing the number of hard disks [5].

In all of the above experiments, we assumed that the exact file access rates are known before determining the file placement. However, in practice, such a perfect estimation of file access rate is not possible, and hence the effectiveness of the optimal file placement significantly depends on the accuracy of the estimation. To evaluate the consequence of the inaccuracy of file access rates estimations, we conducted another simulation experiment. In this experiment, we compute the optimal file placement by using inaccurate file access rates generated by adding Gaussian noise to the exact file access rates. We fix the number of files to 5×10^5 and set the number of disks to a hundred. By varying the variance of the Gaussian noise in {0, 0.001, 0.01, 0.1}, the expected energy consumptions of the cloud storage system with the optimal file placement are evaluated. Figure 6 plots the accumulated energy consumption over a hundred iterations. For comparative purposes, the accumulated energy consumption by the balanced placement without the Gaussian noise is also presented.

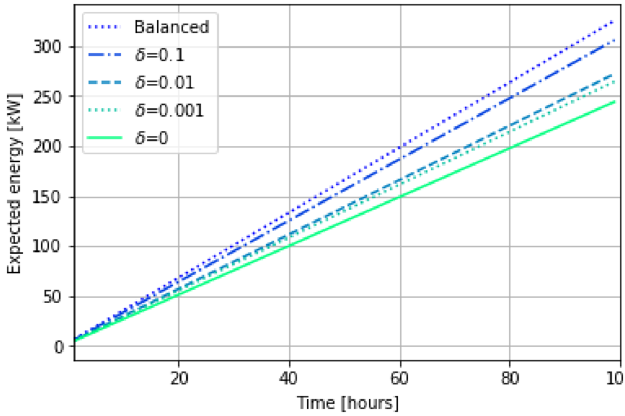

Figure 6. Impacts of the uncertainty of file access rates

The results clearly show that the increased uncertainty (i.e., larger δ) of file access rates significantly impact on the expected energy consumption by the optimal file placement. Even with a smaller variance noise like the case of $\delta = 0.001$, the expected energy consumption is far from the optimal case relying on the perfect estimation of the file access rates. In our test data set, about half of the files in the test data set have never accessed. Accurate predictions of these file accesses are crucial for energy-saving because the effectiveness of the optimal file placement is sensitive to the variance of the access rates of cold files.

6.4 Response time evaluation

The optimal file placement that minimizes the total energy consumption of the cloud storage system may be unacceptable due to the constraint from the average response time requirement (15). Since we assume $T_a < T_{req} < T_s$, the possibility of constraint violation depends on how frequently file requests require the access to standby disks and how long such requests need to wait for disk spin-up. To examine this, we computed the expected average response time for the storage system simulated in the previous scalability experiments. To compute the average response time, we set $T_a = 20$ and vary the values of T_s in {4000, 6000, 8000} milliseconds in reference to [8]. Same as the previous simulation, we vary the number of files n, while the number of disks is fixed to a hundred by adjusting the disk capacity to n /100. The expected average response times are plotted in Figure 7, where we observe that the response time decreases as the number of files increases.

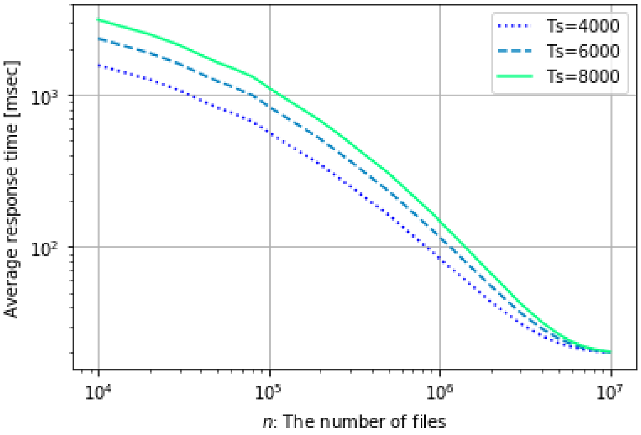

Figure 7. Expected average response times for the storage system with optimal file placement

As can be seen, the average response time is largely influenced by T_s especially when the scale of the system is not too large. However, in a practical range of system scale ($n \geq 10^6$), the average response time is less than a few hundred milliseconds that seems to an acceptable level in terms of the response time requirement T_{req}.

In the above response time model, we do not take into account the throughput capacity of individual disks. Since the optimal file placement attempts to aggregate the accesses to a specific disk, the contention of disk accesses due to the aggregated file accesses may not be negligible. The expected response time drastically gets worse when the disk access rate by the Poisson process exceeds a certain threshold [48]. In order to avoid the performance penalty caused by the contention, it is essential to set disk capacity constraints c_j in consideration of the expected file access rate. The problem is a part of the capacity planning issue for storage systems, which has been studied in the literature [49][50]. Although we do not discuss the capacity planning problem further in this paper, we can avoid such contention by taking a conservative strategy for capacity planning. It means that we can decide disk capacity constraints c_j for the worst-case scenario (i.e., all highly-frequently accessed files are stored in a single disk). For instance, we also used such a conservative policy in our experiments for the optimal file placement shown in TABLE I. In this experimental setting, we limit the disk capacity to 47,835 without considering its efficiency. While the conservative policy is not capacity-efficient, we can avoid an extreme performance penalty due to the concentration of accesses.

7 Conclusion and Future Work

In this paper, we presented a theory of the optimal data placement for an energy-saving cloud storage system for file sharing service. We model the disk state transitions by an alternative renewal process and derive the steady-state probabilities of standby states of hard disks under the given disk access rates. The model can associate the individual file access rates and file placement with the expected energy consumption of hard disks in the storage system. With this model, we formulate the energy-saving file placement problem in which the placement is constrained by the performance requirements and the hard disk capacity. Our theoretical analysis of the optimal solution to the problem derives the two propositions. The first proposition gives a general rule for file migration over the hard disks that reduce the total energy consumption of the storage system. Since the rule does not make any assumptions on hard disk capacity and performance requirements, it is applicable in general for designing heuristic algorithms for data replacement. The second proposition shows that the file placement by order of file access rates can minimize the expected total energy consumption. We clarify the necessary condition where the file placement minimizing the energy consumption is the optimal placement under the requirement for the average response time. Although the heuristic approach has been used in the previous literature, we first give a theoretically proof of the optimality of the heuristic placement method. We further evaluated the effectiveness of the optimal file placement by the experiments on a testbed system and the simulation studies.

Our results have several limitations due to the assumptions introduced for mathematical tractability that needs to be relaxed to some extent in future work. The future extensions of our study can be discussed in the following directions.

- Incorporating varying file access rates
  In this paper, we only consider the optimal file placement assuming that the file access rates are steady at least in the next period. However, in real systems, the frequencies of file accesses are varying over time due to the changing popularity of the content [6][8]. Extending our theory to a more dynamic problem setting is one of the important future research directions.
- Developing efficient file replacement algorithms
  When trying to keep the file placement as close as optimal by file migration across hard disks, we cannot neglect the migration cost. While some existing studies addressed the migration cost in the file placement strategy [6][32][33], a theoretical analysis of the optimal file migration or exchange strategy under the migration cost constraints can be considered in the future.
- Extending the models for high-available storage systems
  Our analysis of optimal file placement did not consider

the redundancy of data placement like by a RAID configuration. Following several existing studies [9][22][23][24], our formal model can be extended to analyze the tradeoff relation among energy-efficiency, performance, and data availability.

- Analyzing the energy-efficiency of hybrid storage systems

  Recent advanced storage systems often employ a tiering configuration that manages a hierarchy of heterogeneous storage devices such as NVRAM, Flash, and SSD [48][49][50], and places the data in consideration with data types and their access patterns. Modeling and analysis of power consumption of such mixed storage architectures can be important future research issues as well.

Acknowledgment

This work was supported in part by a grant of JSPS KAKENHI Grant Number 20H04163.

References

[1] IDC White Paper, Data Age 2025, 2018. [Online] Available https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf

[2] B. Battles, C. Belleville, S. Grabau, and J. Maurier, Reducing data center power consumption through efficient storage, Network Appliance, Inc. 2007.

[3] Horizon, HDD Remains Dominant Storage Technology, 2019. [Online] Available https://www.horizontechnology.com/news/hdd-remains-dominant-storage-technology/

[4] Enterprise Storage Forum, Data Storage Trends 2018, 2019. [Online] Available https://www.enterprisestorageforum.com/storage-management/survey-reveals-tech-trends-reshaping-data-storage.html

[5] E. Pinheiro and R. Bianchini, Energy conservation techniques for disk array-based servers, In Proc. of the 18th annual international conference on Supercomputing, pp. 68-78, 2004.

[6] J. Okoshi, K. Hasebe, and K. Kato, Power-saving in storage systems for internet hosting services with data access prediction, In Proc. of International Green Computing Conference, pp. 1-10, 2013.

[7] R. T. Kaushik and M. Bhandarkar, GreenHDFS: towards an energy-conserving, storage-efficient, hybrid hadoop compute cluster, In Proc. of the USENIX annual technical conference, vol. 109, p. 34, 2010.

[8] K. Hasebe, J. Okoshi, and K. Kato, Power-saving in storage systems for cloud data sharing services with data access prediction, IEICE Transactions on Information and Systems, vol. 98, no. 10, pp. 1744-1754, 2015.

[9] C. Weddle, M. Oldham, J. Qian, A. I. A. Wang, P. Reiher, and G. Kuenning, PARAID: A gear-shifting power-aware raid, ACM Transactions on Storage (TOS), vol. 3, no. 3, p. 13, 2007.

[10] D. Narayanan, A. Donnelly, and A. Rowstron, Write off-loading: Practical power management for enterprise storage, ACM Transactions on Storage (TOS), vol. 4, no. 3, p. 10, 2008.

[11] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes, Hibernator: helping disk arrays sleep through the winter, In ACM SIGOPS Operating Systems Review, vol. 39, no. 5. pp. 177-190, 2005.

[12] S. Yagai, M. Oguchi, M. Nakano, and S. Yamaguchi, Power-effective file layout based on large scale data-intensive application in virtualized environment, IEICE Transactions on Information and Systems, vol. 100, no. 12, pp. 2761-2770, 2017.

[13] T. Bostoen, S. Mullender, and Y. Berbers, Power-reduction techniques for data-center storage systems, ACM Computing Surveys (CSUR), vol. 45, no. 3, p. 33, 2013.

[14] G. S. Aujla, M. Singh, N. Kumar, and A. Zomaya, Stackelberg game for energy-aware resource allocation to sustain data centers using RES, IEEE Transactions on Cloud Computing, vol. 7, no. 4, pp. 1109-1123, 2019.

[15] N. Kumar, G. S. Aujla, S. Garg, K. Kaur, R. Ranjan, and S. K. Garg, Renewable energy-based multi-indexed job classification and container management scheme for sustainability of cloud data centers, IEEE Transactions on Industrial Informatics, vol. 15, no. 5, pp. 2947-2957, 2018.

[16] A. Forestiero, C. Mastroianni, M. Meo, G. Papuzzo, and M. Sheikhalishahi, Hierarchical approach for efficient workload management in geo-distributed data centers, IEEE Transactions on Green Communications and Networking, vol. 1, no. 1, pp. 97-111, 2016.

[17] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, Toward energyefficient cloud computing: Prediction, consolidation, and overcommitment, IEEE network, vol. 29, no. 2, pp. 56-61, 2015.

[18] R. Bianchini and R. Rajamony, Power and energy management for server systems, Computer, vol. 37, no. 11, pp. 68-76, 2004.

[19] L. Benini and G. d. Micheli, System-level power optimization: techniques and tools, ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 5, no. 2, pp. 115-192, 2000.

[20] D. Colarelli and D. Grunwald, Massive arrays of idle disks for storage archives, In Proc. of ACM/IEEE Conference on Supercomputing, pp. 1-11, 2002.

[21] N. Nishikawa, M. Nakano, and M. Kitsuregawa, Energy efficient storage management cooperated with large data intensive applications, In Proc. of the 28th International Conference on Data Engineering, pp. 126-137, 2012.

[22] D. Li and J. Wang. EERAID: Energy-efficient redundant and inexpensive disk array, In Proc. of the 11th ACM SIGOPS European Workshop, pp. 20-22 2004.

[23] X. Yao, J. Wang, RIMAC: A novel redundancy-based hierarchical cache architecture for energy efficient, high performance storage systems, In ACM SIGOPS Operating Systems Review, vol. 40, no. 4, pp. 249-262, 2006.

[24] E. Pinheiro, R. Bianchini, and C. Dubnicki, Exploiting redundancy to conserve energy in storage systems, ACM SIGMETRICS Performance Evaluation Review, vol. 34, no. 1, pp. 15-26, 2006.

[25] A. Verma, R. Koller, L. Useche, and R. Rangaswami, SRCmap: Energy proportional storage using dynamic consolidation, USENIX Conference on File and Storage Technologies, vol. 10, pp. 267-280, 2010.

[26] M. W. Storer, K. M. Greenan, E. L. Miller, and K. Voruganti, Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage, USENIX Conference on File and Storage Technologies, p.1, 2008.

[27] K. Hasebe, T. Sawada, and K. Kato, A game theoretic approach to power reduction in distributed storage systems, Journal of Information Processing, vol. 24, no. 1, pp. 173-181, 2016.

[28] K. Hasebe, T. Niwa, A. Sugiki, and K. Kato, Power-saving in largescale storage systems with data migration, In Proc. of the Second International Conference on Cloud Computing Technology and Science, pp. 266-273, 2010.

[29] C. Karakoyunlu and J. A. Chandy, Exploiting user metadata for energy-aware node allocation in a cloud storage system, Journal of Computer and System Sciences, vol. 82, no. 2, pp. 282-309, 2016.

[30] M. Allalouf, Y. Arbitman, M. Factor, R. I. Kat, K. Meth, and D. Naor, Storage modeling for power estimation, In Proc. of SYSTOR2009: The Israeli Experimental Systems Conference, p. 3, 2009.

[31] H. Yuan, I. Ahmad, and C. C. J. Kuo, Performance-constrained energy reduction in data centers for video-sharing services, Journal of Parallel and Distributed Computing, vol. 75, pp. 29-39, 2015.

[32] P. Behzadnia, W. Yuan, B. Zeng, Y. C. Tu, and X. Wang, Dynamic power-aware disk storage management in database servers, In Proc. of International Conference on Database and Expert Systems Applications, pp. 315-325, 2016.

[33] P. Behzadnia, Y. C. Tu, B. Zeng, and W. Yuan, Energy-aware disk storage management: Online approach with application in DBMS, arXiv preprint arXiv:1703.02591, 2017.

[34] F. Yan, X. Mountrouidou, A. Riska, and E. Smirni, PREFiguRE: an analytic framework for hdd management, ACM Transactions on Modeling and Performance Evaluation of Computing Systems, vol. 1, no. 3, p. 10, 2016.

[35] M. Song, Minimizing power consumption in video servers by the combined use of solid-state disks and multi-speed disks, IEEE Access, vol. 6, pp. 25737-25746, 2018.

[36] X. Lu, C. Sun, C. Yu, J. Sun, M. Che, Z. Xia, Z. Shang, and Y. Hu, A data-aware energy-saving storage management strategy for on-site astronomical observation at Dome A, In Proc. of International Conference on Algorithms and Architectures for Parallel Processing, pp. 551-566, 2018,

[37] C. Hu and Y. Deng, Aggregating correlated cold data to minimize the performance degradation and power consumption of cold storage nodes, The Journal of Supercomputing, vol. 75, no. 2, pp. 662-687, 2019.

[38] Akamai reveals 2 seconds as the new threshold of acceptability for eCommerce web page response times, 2009. [Online] Available https://www.akamai.com/us/en/about/news/press/2009-press/akamai-reveals-2-seconds-as-the-new-threshold-of-acceptability-for-ecommerce-web-page-response-times.jsp

[39] L. A. Barroso and U. Hölzle, The datacenter as a computer: An introduction to the design of warehouse-scale machines, Synthesis lectures on computer architecture, vol. 4, no. 1, pp. 1-108, 2009.

[40] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, Web caching and zipf-like distributions: evidence and implications, In Proc. of INFOCOM'99, vol. 1, pp. 126-134, 1999.

[41] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, Youtube traffic characterization: a view from the edge, In Proc. of the 7th ACM SIGCOMM conference on Internet measurement, pp. 15-28, 2007.

[42] T. Xie, Sea: A striping-based energy-aware strategy for data placement in raid-structured storage systems, IEEE Transactions on Computers, vol. 57, no. 6, pp. 748-761, 2008.

[43] R. T. Kaushik, L. Cherkasova, R. Campbell, and K. Nahrstedt, Lightning: self-adaptive, energy-conserving, multi-zoned, commodity green cloud storage system, In Proc. of the 19th ACM International Symposium on High Performance Distributed Computing, pp. 332-335, 2010.

[44] Y. H. Lu, E. Y. Chung, T. Simunic, T. Benini, and G. De Micheli, Quantitative comparison of power management algorithms, In Proc. of Conference on Design, Automation and Test in Europe, pp. 20-26, 2000.

[45] W. Tang, Y. Fu, L. Cherkasova, and A. Vahdat, MediSyn: A synthetic streaming media service workload generator, In Proc. of the 13th international workshop on Network and operating systems support for digital audio and video, pp. 12-21, 2003.

[46] M. Kim and M. Song, Saving energy in video servers by the use of multispeed disks, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 4, pp. 567-580, 2011.

[47] K. S. Trivedi and A. Bobbio, Reliability and availability engineering : modeling, analysis and applications, Cambridge University Press, 2017.

[48] M. Hofri, Disk scheduling: Fcfs vs. sstf revisited, Communications of the ACM, vol. 23, no. 11, pp. 645-653, 1980.

[49] E. Anderson, M. Hobbs, K. Keeton, S. Spence, M. Uysal, and A. C. Veitch, Hippodrome: Running circles around storage administration, USENIX Conference on File and Storage Technologies, vol. 2, 2002, pp. 175-188.

[50] E. Anderson, S. Spence, R. Swaminathan, M. Kallahalla, and Q. Wang, Quickly finding near-optimal storage designs, ACM Transactions on Computer Systems (TOCS), vol. 23, no. 4, pp. 337-374, 2005.

[51] S. Zheng, M. Hoseinzadeh, and S. Swanson, Ziggurat: a tiered file system for non-volatile main memories and disks, USENIX Conference on File and Storage Technologies, pp. 207-219, 2019,

[52] B. Debnath, S. Sengupta, and J. Li, Flashstore: high throughput persistent key-value store, In Proc. of the VLDB Endowment, vol. 3, no. 1-2, pp. 1414-1425, 2010.

[53] C. Li, P. Shilane, F. Douglis, H. Shim, S. Smaldone, and G. Wallace, Nitro: A capacity-optimized SSD cache for primary storage, USENIX Annual Technical Conference, pp. 501-512, 2014.

Fumio Machida is an associate professor at the Computer Science Department in University of Tsukuba. Before the current position, he was a principal researcher at NEC Corporation. He was a visiting scholar in the Department of Electrical and Computer Engineering at Duke University in 2010. He received the PhD degree from Tokyo Institute of Technology in 2018. He was a recipient of the young scientists' prize of Japan in 2014. His research interests include modeling and analysis of system dependability, software aging and rejuvenation, and cloud and edge computing systems. He is a senior member of the IEEE and the member of ACM.

Koji Hasebe is an associate professor at the Computer Science Department in University of Tsukuba. Before the current position, he was research fellow at the National Institute of Advanced Industrial Science and Technology, Japan. He received his Bachelor's, Master's, and Ph.D. degrees in Philosophy from Keio University in Japan in 1998, 2000, and 2006, respectively. His research interests include multi-agent systems, formal verification, game theory, and computer security. He is a member of the IEEE and ACM.

Hirotake Abe received the B.E. degree in 1999, the M.E. degree in 2001, and the Ph.D. degree in 2004, all from University of Tsukuba, Japan. From 2004 to 2007, he was a research staff of Japan Science and Technology Agency. From 2007 to 2010, he was an Assistant Professor at Toyohashi University of Technology, Japan. From 2010 to 2012, he was an Assistant Professor at Osaka University, Japan. He is currently an Associate Professor at University of Tsukuba, Japan. His research interests include system software, distributed systems and computer security.

Kazuhiko Kato received the BE and ME degrees from the University of Tsukuba, Japan, in 1985 and 1987, respectively. He received the PhD degree from the University of Tokyo, Japan, in 1992. From 1989 to 1993, he was a research associate in the Department of Information Sciences, Faculty of Sciences at the University of Tokyo. He is currently a professor in the Department of Computer Science, Graduate School of System Information Engineering at the University of Tsukuba. His research interests include operating systems, distributed systems, and secure computing. He received the distinguished paper awards from JSSST and IPSJ in 2004 and 2005, respectively.