IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 28, 2018, accepted April 11, 2018, date of publication May 2, 2018, date of current version June 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2832221

Minimizing Power Consumption in Video Servers by the Combined Use of Solid-State Disks and Multi-Speed Disks

MINSEOK SONG, (Member, IEEE)
Department of Computer Engineering, Inha University, Incheon 22212, South Korea

(e-mail: mssong@inha.ac.kr)

This work was supported by the Inha University Research Grant.

ABSTRACT Disks use a considerable portion of the total energy consumed by a video server. Solid-state disks (SSDs) can be used as a cache to reduce disk speeds, thereby decreasing the power consumed by disks. However, effective SSD bandwidth management is essential to cope with the high request rates for popular video files stored on SSDs. To address this problem, we propose a new SSD cache management scheme for a video server that uses multi-speed disks. First, we introduce a storage allocation scheme to examine how the presence of an SSD cache affects disk bandwidth consumption. We then propose an SSD bandwidth allocation algorithm to minimize energy consumption by allowing disks to run at lower speeds while achieving jitter-free disk speed transitions and by limiting the number of transitions to ensure disk reliability. Our experimental results demonstrate that our proposed bandwidth allocation algorithm achieves appreciable power savings under various workloads while limiting the number of disk speed changes.

INDEX TERMS Multimedia systems, data storage systems, energy efficiency.

I. INTRODUCTION

Recent advances in network and system technologies have made it feasible to provide video-on-demand (VoD) services for a range of applications including digital libraries, education-on-demand, distance learning, and user-created content. Naturally, this increases the amount of Internet video traffic, which then requires a large server structure. For example, it has been reported that nearly five billion videos are watched everyday on YouTube [1].

This growing demand for video data increases the energy consumed by servers, which represents a major problem. In particular, because of their high bandwidth requirements, servers are typically built as a redundant array of independent disk (RAID), which may comprise thousands of disks, thereby making it one of the biggest energy consumers in a server. Recent studies have shown that the energy consumed by storage systems may constitute 40% of that of the entire data center [2], [3]. Considerable energy consumption also represents a serious economic concern for service providers. For instance, a data center with a provisioned 10-MW peak power capacity costs between 100 and 200 million dollars per year [4].

A major consequence of this high power consumption is heat, which negatively affects system reliability. For example, it has been shown that running at 15°C above ambient temperature can double the failure rate of disk drives [2], [5], [6]. However, cooling systems for high heat densities are prohibitively expensive and the cooling system itself adds considerably to the power requirement [2]. For example, some studies have shown that the cooling system is the largest power consumer in a data center, where it accounts for 50% of the total power used [2]. High energy consumption also hinders the expansion of servers and has a negative effect on the environment [2], [6].

Reducing the speed at which a disk spins decreases its power consumption. Thus, Gurumurthi [7] and Gurumurthi *et al.* [8] proposed using disks that can run at several different speeds. Using multiple disks running at different speeds can greatly reduce power consumption, even under video server workloads, as the disks spin slowly in response to fewer video requests. By contrast, a single-speed disk always spins at full speed. Multi-speed disks are commercially available. These include drives produced by Western Digital that use an IntelliPower technique to adjust the rotations per minute (RPM) and transfer rate to reduce the disk power consumption [9]. They have also been used to build more energy-efficient storage systems (e.g., Nexsan storage servers [10]). These developments have motivated further research into effective power reduction techniques based on multi-speed disks.

Flash-based solid-state disks (SSDs) have many technical merits such as non-volatility, low power consumption, shock resistance, and excellent random read performance, and thus are now being used for enterprise storage systems such as video servers [11]–[15]. The main reasons for using SSDs in video servers [13]–[15] include: 1) low latency when reading data, which is crucial in video servers; 2) good throughput for large random read operations, which is important for supporting large numbers of simultaneous video streams; and 3) good performance for sequential write operation, which is a necessary attribute of a video server.

The high cost of SSDs prohibits their use for storage in video servers because of the amount of data in videos (e.g., storing 10,000 2 h movies at 9 Mb/s requires 78 TB). The cost of flash memory has been declining, but it is still an order of magnitude more expensive than disk storage [16], [17]. However, the pattern of access to a set of videos tends to be highly skewed because most requests are for only a few popular videos. Thus, we can use an SSD as a cache to improve the throughput of a video server while also reducing the disk power consumption rate. However, this also means that making effective use of the limited SSD bandwidth is essential due to the high request rates for popular video files on the SSD.

Several studies have attempted to use SSDs for video servers. For example, a file-level caching scheme was recently introduced to increase the number of simultaneous clients [13], and a dynamic data replication scheme was developed for allowing popular videos to be replicated on SSD, so as to improve the SSD caching hit ratio [18]. However, to the best of our knowledge, the present study is the first to consider SSD bandwidth management issues in order to minimize power consumption for a video server that uses SSDs and multi-speed disks.

We present an SSD cache management scheme for video servers that use multi-speed disks. First, we examine how the allocation of SSD storage affects disk bandwidth. We then examine how the power consumption varies with the number of requests served by the SSD and propose an algorithm that determines the requests to be served from the SSD in order to minimize the disk energy consumption subject to the SSD bandwidth. This algorithm also considers disk reliability issues by limiting the number of speed changes and by providing speed transitions without jitter.

The remainder of this paper is organized as follows. First, we review related studies in Section II and present the basic idea and system model in Section III. We describe how SSD storage can be allocated in Section IV and propose an algorithm for SSD bandwidth allocation in Section V. We assess our schemes in Section VI and present our conclusion in Section VII.

II. RELATED WORK

A. DISK POWER MANAGEMENT FOR SERVERS

Techniques for managing disk power consumption in servers have been studied widely, mainly by extending the periods when the disk can be in a low-power mode. Several schemes redistribute the workload to allow more disks to remain in low-power modes. For example, Pinheiro and Bianchini [19] proposed a data concentration technique in which the workloads are dynamically migrated so the load becomes skewed toward only a few disks in order to allow other disks to enter low-power modes. Khatib and Bandic [4] proposed a power capping scheme that uses different disk scheduling algorithms to limit disk power consumption.

Redundancy is essential for ensuring fault tolerance in all practical storage systems, and this approach has been exploited by several energy conservation schemes. Pinheiro *et al.* [20] presented a request-redirection scheme to leverage the redundancy in storage systems. Weddle *et al.* [21] proposed a new RAID architecture that employs skewed striping techniques to maximize the number of disks in the low-power mode.

To reduce the amount of energy used when server workloads are intense, Gurumurthi [7] and Gurumurthi *et al.* [8] proposed a speed-changing scheme called dynamic RPM and suggested that single-speed disks are not suitable for reducing the energy used when processing server workloads. Several schemes have been presented for servers based on this method. Zhu *et al.* [5] proposed a flexible disk placement architecture called Hibernator to reduce disk speeds as often as possible. Xie [22] described a data placement technique in which the loads are distributed such that some disks can run at low speeds.

All of these techniques attempt to resolve disk power issues using various methods, but they cannot guarantee real-time data retrieval, thus making them unsuitable for video servers. Lee *et al.* [23] presented a prefetching scheme in which the amount of data prefetched for each video stream is dynamically adjusted to minimize the power consumption by a disk array with heterogeneous disks. Song [24] suggested using multi-speed disks for video servers by guaranteeing continuous data retrieval when disk speeds change. Kim and Song [25] proposed a technique that adjusts the retrieval period to allow more disks to run at low speeds. Yuan *et al.* [26] presented energy management techniques to satisfy delay constraints for large-scale video-sharing services. However, none of these methods use SSD.

B. USE OF SSD FOR SERVERS

Many studies have used an SSD cache to improve the performance of disk-based storage systems. Li *et al.* [27] proposed a caching architecture with adjustable deduplication, compression, and replacement granularities in order to minimize the response time. Arteaga *et al.* [17] developed schemes for allocating flash memory across virtual machines to balance the cache loads in cloud computing systems. Meng *et al.* [28] developed schemes for partitioning flash memory to optimize the allocation of flash memory to virtual machines.

To utilize the limited SSD space effectively, various methods of storing popular data on the SSD have been studied. Kim and Kim [29] suggested a method of storing the prefix parts of popular videos onto SSD to improve overall data bandwidth. Lin *et al.* [30] introduced a scheme called hot random offloading (HRO) that stores frequently accessed random data on an SSD to handle as many random I/O requests as possible from the SSD. Yin *et al.* [31] presented an SSD cache architecture that stores popular data on the SSD in order to allow disks to spin down when the workloads are light, thereby reducing the amount of energy consumed by the disk. He *et al.* [32] proposed a new SSD cache placement scheme in which performance-critical data are cached on an SSD to fully utilize SSD-based file servers with the aim of improving I/O performance of the parallel file systems.

SSDs can also be used to reduce the power consumed by disk arrays [11]. Kgil *et al.* [33], [34] studied various types of cache architecture in order to reduce power consumption by the main memory and disk. Useche *et al.* [35] presented several schemes for an energy-aware SSD cache. These included data popularity prediction, I/O indirection, and reconfiguration schemes. Hui *et al.* [36] introduced a hybrid storage system that uses an SSD as a cache to allow under-utilized disks to be spun down.

None of the methods previously mentioned consider multimedia workloads. To address this, Ryu *et al.* [13] analyzed the advantages and disadvantages of using SSDs in video servers, where it was shown that low-end SSDs can allow essentially constant throughput. They also extended this approach to the use of SSDs for HTTP adaptive streaming [14], [15]. Manjunath and Xie [18] considered an architecture for an SSD-based server in which a video file can be dynamically replicated from disk to SSD in order to reduce the disk load. Lee and Song [37] developed a video cache management scheme that considers the popularity of video segments in order to minimize disk bandwidth consumption. Chen *et al.* [38] developed an error correction technique to use low-cost flash memory for video servers by considering video coding and flash memory characteristics. In all of these previous studies, however, disk power issues were not considered.

III. BASIC IDEA AND SYSTEM MODEL

A. BASIC IDEA

If I/O requests are served by the SSD cache, then disk bandwidth utilization at each speed level is effectively reduced, which also decreases the power consumed by the disk. However, effective SSD bandwidth management is essential to serve many popular requests from the cache. To address this problem, we propose an algorithm that determines requests served by the SSD based on the following steps:

1) The problem formulation step examines how the power consumed at each disk speed level depends on the number of requests served by the SSD.
2) The bandwidth allocation step determines the requests served by the SSD to minimize the overall power consumption while also limiting the number of disk speed changes.

Frequently starting and stopping disks is known to affect disk drive longevity [5]. Therefore, many disk specifications provide their duty cycles, which are the maximum number of times that the drive can be spun down [5], [25]. The disk speed change also involves this spin-down operation, which means that minimizing the number of speed changes is essential [5]. Therefore, we limit the number of simultaneous speed changes when the algorithm is executed.

B. SYSTEM MODEL

Consider a video server composed of an SSD and an array of multi-speed disks. When the server receives video requests, it first checks whether requested data is located on the SSD; otherwise, video data can be served from multi-speed disks.

We organize the retrieval of data from a video server using round-based scheduling [23]–[25]. The time is divided into periods called rounds which are of an equal length of T_r, where each client is served once during each round. (Table 1 summarizes the important symbols used in this paper.) Video file i has a bit rate of b_i. When a server receives requests from clients, it allocates bandwidth to each request based on its bit rate. The amount of data that must be read for each video i during round T_r is $b_i T_r$ in order to maintain the playback rate. Therefore, to stream at 1.5 Mbps with a round length of 2 s, the server must read 3 Mbits of data during every round.

Since the data transfer rate of a single disk is significantly higher than the playback rate of a single video stream, each disk in a video server is able to provide the data for multiple streams. Disk bandwidth utilization can be defined as the ratio of the total service time that a disk spends retrieving all the streams during a round relative to the round length, and it must be less than or equal to 1 [24], [25].

We consider a server that contains N_d multi-speed disks, in which each disk supports N_s speed levels. Let $r(l)$ be the data transfer rate for a disk running at speed level l, $(l = 1, \ldots, N_s)$. We use a typical seek-time model in which a constant seek time T_s is required for one read of contiguous data [24], [25]. A disk is composed of multiple tracks. For video workloads, each disk has an effective rotational delay of 0 because each video segment can be sufficiently large that it can span one or more complete disk tracks [39]. The disk head can start reading as soon as it reaches the destination track; it then reads the entire track rather than waiting to go to the location in the track where data is placed. Therefore, we assume that the rotational delay is 0.

Reading the video file incurs a seek overhead of T_s and a reading time of $\frac{b_i T_r}{r(l)}$, which increases the service time by $T_s + \frac{b_i T_r}{r(l)}$. Thus, serving video i increases the disk bandwidth

TABLE 1. **Important symbols used in this paper.**

Notation	Meaning
N_s, N_d, N_v	Number of speeds, disks, and videos, respectively
θ	Skewness parameter in Zipf distribution
$P_s(l)$, $P_a(l)$, $P_i(l)$	Seek, active, and idle power for speed level l, respectively
T_r, T_s	Round length and typical seek time, respectively
T_t	Transition time between two different speed levels
P_t	Power required for speed transition
$r(l)$	Disk transfer rate at speed level l
S_i, b_i, p_i, L_i	Size, bit rate, access probability, and length of video i, respectively
D_i	Average disk bandwidth utilization over all speed levels for a video i
G_i	Proportion of disk bandwidth utilization reduction for a video i over all videos
S_{ssd}, B_{ssd}	Capacity and bandwidth of SSD, respectively
Y_i	Value denoting whether the first Y_i proportion of a video i is cached
$V(j)$	Video index for request j
A_k	Array of possible request combinations served by SSD for a disk k
$C_{k,m}$	Set of requests in the mth combination in array A_k
N_k	Number of elements in array A_k
$X_{k,m}$	Binary variable that determines whether requests in the mth element of A_k are served by an SSD
$D_{k,m}(l)$	Disk bandwidth utilization of disk k at speed level l when $X_{k,m} = 1$
$E^{\text{seek}}_{k,m}(l)$, $E^{\text{active}}_{k,m}(l)$, $E^{\text{idle}}_{k,m}(l)$	Energy consumption in seek, active, and idle phases at speed level l during a round when $X_{k,m} = 1$
$P_{k,m}(l)$	Power consumption of disk k at speed level l when $X_{k,m} = 1$
$B_{k,m}$	Amount of SSD bandwidth required for a set of requests in $C_{k,m}$
$L_{k,m}$	Lowest speed level for disk k when $X_{k,m} = 1$
N_p	Number of rounds allocated for prefetching
$D^{\text{cont}}_{k,m}(l)$	Disk bandwidth utilization for disk k at speed level l reserved for prefetching during the speed transition when $X_{k,m} = 1$
$L_{k,m}$	Lowest speed level for disk k where $\{l \mid D_{k,m}(l) + D^{\text{cont}}_{k,m}(l) \leq 1\}$
$ch_{k,m}$	Variable indicating whether speed changes occur when $X_{k,m} = 1$
N_c	Number of simultaneous speed changes allowed
U	Upper bound on speed changes
α	Every αth execution of the algorithm, where N_c is set to U; otherwise, $N_c = 0$

utilization at speed l by $\frac{T_s+\frac{b_iT_r}{r(l)}}{T_r}$. Let D_i be the average disk bandwidth utilization over all speed levels for video i, which can be calculated as follows:

$$D_i = \frac{\sum_{l=1}^{N_s} T_s + \frac{b_iT_r}{r(l)}}{T_rN_s}.$$

IV. SSD STORAGE ALLOCATION

The arrival of client requests follows a Poisson distribution [40], [41] and the access probability follows a Zipf distribution. Therefore, the probability of requiring video i, p_i, is calculated as follows:

$$p_i = \frac{1}{\sum_{m=1}^{N_v} m^{\frac{1}{1-\theta}}} N_v \frac{1}{i^{1-\theta}},$$

where N_v is the number of videos and θ is a skewness parameter [40]. Let L_i be the length of video i in seconds. The proportion of requests for video i over all the videos can then be expressed as p_iL_i [42]. Let G_i be a parameter that denotes the expected proportion of disk bandwidth utilization reduction, which can be expressed as:

$$G_i = p_iL_iD_i.$$

Let S_i be the size of video i in MB. Let Y_i denote whether the first Y_i proportion of video i is cached, $(0 \leq Y_i \leq 1)$. Obviously, our aim is to cache video files in order to maximize the disk bandwidth reduction, $\sum_{i=1}^{N_v} Y_iG_i$. If $Y_i = 1$, then video i is entirely cached. By contrast, if $Y_i = 0$, video i is not cached at all. Let S_{ssd} be the size of SSD in MB, and the total SSD storage requirement must not exceed S_{ssd}. Thus, $\sum_{i=1}^{N_v} Y_iS_i \leq S_{\text{ssd}}$. We can formulate the SSD allocation problem that determines Y_i as follows:

$$\begin{aligned} &\text{Maximize} \sum_{i=1}^{N_v} Y_iG_i \\ &\text{subject to} \sum_{i=1}^{N_v} Y_iS_i \leq S_{\text{ssd}}, \\ &\qquad 0 \leq Y_i \leq 1. \end{aligned}$$

This problem is a linear version of the 0/1 knapsack problem (LKP) [43]. In the LKP, each object has a weight and profit, and the problem involves selecting a fractional proportion of each object such that the total profit is maximized while satisfying the capacity constraints [43]. We can treat the storage allocation problem as the LKP by considering the SSD as a knapsack and by regarding each video as an object.

The optimal solution to the LKP can be found using a greedy approach [43]. Thus, in the storage allocation problem, each video is sorted in non-ascending order of the values of $\frac{G_i}{S_i}$, and the video files having higher values for $\frac{G_i}{S_i}$ are cached first subject to the SSD storage limit.

V. SSD BANDWIDTH ALLOCATION

A. POWER MODELING

A disk has three power phases–seek, idle, and active–where the disk seeks during the seek phase, spins without read, write, or seek operations during the idle phase, and actually reads or writes during the active phase [23]. A disk seek phase can consist of four steps: a speedup step, when the arm is accelerated; a coast step, when the arm moves at its maximum velocity for long seeks; a slowdown step, when the arm decelerates to reach the desired track; and a settle step, when the disk controller adjusts the head to access the desired location [44], [45]. The power consumption required for the seek phase includes all these four steps.

The seek phase consumes the most energy. Therefore, reducing the number of seek operations is critical [23]. In order to minimize the number of seek operations, it is advantageous to process low bit-rate requests first from the SSD. During each round T_r, the exact amount of data, $b_i T_r$ needs to be read for video i. For example, consider two videos with bit-rates of 1.5 and 3 Mbps; to read 3Mbits of data, the former requires two seek operations, whereas the latter requires one seek during a round of 1 s. Therefore, we give a higher priority to requests for videos with low bit-rates. Let $R_{k,m}$ be a request for a video cached on the SSD with original files stored on disk k and with the mth lowest bit rate. Then, we have an array of request combinations served from the SSD for disk k, A_k as follows:

$$A_k = \{\phi, \{R_{k,1}\}, \{R_{k,1}, R_{k,2}\}, \ldots, \{R_{k,1}, \ldots, R_{k,N_k-1}\}\},$$

where N_k represents the number of elements in this array.

Let $V(j)$ be the video index for request j. We introduce a binary variable $X_{k,m}$ as follows: if videos in the mth combination in an array A_k (e.g., $C_{k,m}$) are served by SSD, then $X_{k,m} = 1$; otherwise, $X_{k,m} = 0$. For example, if $X_{k,3} = 1$, then two requests ($R_{k,1}$ and $R_{k,2}$) are served by SSD. We then calculate the disk bandwidth utilization at each speed level l when $X_{k,m} = 1$, $D_{k,m}(l)$ as follows:

$$D_{k,m}(l) = \sum_{j \in C_{k,N_k} - C_{k,m}} \frac{T_s + \frac{b_{V(j)} T_r}{r(l)}}{T_r}.$$

Let $P_s(l)$ be the power required during the seek phase for disk speed level l. $P_a(l)$ is the active power required when the disk is reading or writing, while $P_i(l)$ is the power consumed when the disk is rotating without disk activity when the speed level is l. We then derive the energy consumption at speed level l when $X_{k,m} = 1$ during T_r for each power phase as follows:

1) The total seek time is $\sum_{j \in C_{k,N_k} - C_{k,m}} T_s$. Therefore, the energy required in the seek phase, $E^{\text{seek}}_{k,m}(l)$, is calculated as:

$$E^{\text{seek}}_{k,m}(l) = \sum_{j \in C_{k,N_k} - C_{k,m}} T_s P_s(l).$$

2) The total time taken to read data for T_r is $\sum_{j \in C_{k,N_k} - C_{k,m}} \frac{b_{V(j)} T_r}{r(l)}$. Therefore, the energy required for reading data, $E^{\text{active}}_{k,m}(l)$, is calculated as:

$$E^{\text{active}}_{k,m}(l) = \sum_{j \in C_{k,N_k} - C_{k,m}} \frac{b_{V(j)} T_r}{r(l)} P_a(l).$$

3) If no disk activity is occurring, the disk is rotating without reading or seeking, which requires a power of $P_i(l)$. We calculate the total idle time during a round of length T_r by subtracting the seek and read times from T_r. Thus, the energy required in the idle phase, $E^{\text{idle}}_{k,m}(l)$, can be calculated as:

$$E^{\text{idle}}_{k,m}(l) = (T_r - \sum_{j \in C_{k,N_k} - C_{k,m}} \frac{b_{V(j)} T_r}{r(l)}) P_i(l).$$

We can now determine the power consumption by disk k when the speed level is l and $X_{k,m} = 1$, $P_{k,m}(l)$ as:

$$P_{k,m}(l) = \frac{E^{\text{seek}}_{k,m}(l) + E^{\text{active}}_{k,m}(l) + E^{\text{idle}}_{k,m}(l)}{T_r}.$$

B. COPING WITH DISK SPEED CHANGES

A considerable amount of time may be required to transition from one speed to another [25]. Disk requests cannot be handled during this speed transition period, which causes playback jitter. Prefetching data, which requires contingent disk bandwidth [24], is necessary during the speed transition period to prevent this problem.

Let T_t be the transition time between two different speed levels [24], [25]. The server has to prefetch a sufficient amount of data in T_t s. In order to ensure real-time playback, the data to be consumed during the round immediately after the transition period must also be prefetched [24], [25]. Thus, a total of $T_t + T_r$ s of data can be prefetched for each stream if the speed changes. If the prefetching starts N_p rounds before speed transition, then the amount of data for request j, $(T_t + T_r) b_{V(j)}$, needs to be prefetched across N_p rounds. Therefore, the contingent disk bandwidth for a speed change when $X_{k,m} = 1$ for speed level l, $D^{\text{cont}}_{k,m}(l)$ can be calculated as:

$$D^{\text{cont}}_{k,m}(l) = \sum_{j \in C_{k,N_k} - C_{k,m}} \frac{T_s + \frac{(T_t + T_r) b_{V(j)}}{r(l)}}{N_p T_r}.$$

Let $L_{k,m}$ be the speed level selected when $X_{k,m} = 1$. Because some contingent disk bandwidth may be reserved to prepare for disk speed changes, we select the lowest speed level in $\{l \mid D_{k,m}(l) + D^{\text{cont}}_{k,m}(l) \leq 1\}$ for $L_{k,m}$ to maximize the reduction in disk energy consumption. We also introduce a binary variable, $ch_{k,m}$, to indicate whether speed changes occur in disk k when $X_{k,m} = 1$. If the selected speed level is the same as the current speed level, then $ch_{k,m} = 0$; otherwise, $ch_{k,m} = 1$.

C. PROBLEM FORMULATION

Let B_{ssd} be the SSD total bandwidth in MB/s. Let $B_{k,m}$ be the SSD bandwidth in MB/s required to serve all of the requests in $C_{k,m}$. Let P_t be the power required for speed transition. We introduce a variable, N_c, to denote the number of simultaneous speed changes. We then have two constraints, (**C1** and **C2**), for SSD bandwidth allocation as follows:

1) **C1**: The total bandwidth needed to serve requests from the SSD must not exceed the SSD bandwidth; $\sum_{k=1}^{N_d}\sum_{m=1}^{N_k} X_{k,m}B_{k,m} \leq B_{\mathrm{ssd}}$.
2) **C2**: The number of speed change must not exceed N_c; $\sum_{k=1}^{N_d}\sum_{m=1}^{N_k} X_{k,m}ch_{k,m} \leq N_c$. This constraint also limits the amount of energy consumption required for the speed transition by $N_cP_tT_t$.

We can then formulate the bandwidth allocation problem ($\mathcal{BAP}$), which minimizes power consumption for determining $X_{k,m}$, ($k = 1, \ldots, N_d$, and $m = 1, \ldots, N_k$) as:

$$\begin{aligned}
&\text{Minimize} \sum_{k=1}^{N_d}\sum_{m=1}^{N_k} X_{k,m}P_{k,m}(L_{k,m}) \\
&\text{subject to} \sum_{k=1}^{N_d}\sum_{m=1}^{N_k} X_{k,m}B_{k,m} \leq B_{\mathrm{ssd}}, \\
&\qquad\qquad \sum_{k=1}^{N_d}\sum_{m=1}^{N_k} X_{k,m}ch_{k,m} \leq N_c, \\
&\qquad\qquad \sum_{m=1}^{N_k} X_{k,m} = 1, \\
&\qquad\qquad X_{k,m} \in 0, 1.
\end{aligned}$$

D. BANDWIDTH ALLOCATION ALGORITHM (BAA)

$\mathcal{BAP}$ is a variant of the multi-dimensional multiple-choice knapsack problem (MMKP), which is NP-hard [43]. In the MMKP, each object consists of a set of items, with each item having a weight and profit. The problem involves selecting exactly one item from each object in order to maximize the total profit while satisfying multiple constraints [43]. We can convert the $\mathcal{BAP}$ into the MMKP by considering the SSD as a knapsack and by regarding each array for disk k, A_k as an object. The objective of the $\mathcal{BAP}$ is to minimize power consumption while satisfying the two constraints of **C1** and **C2**.

Because $\mathcal{BAP}$ is NP-hard, we propose a heuristic solution that runs in polynomial time. In general, greedy algorithms exhibit good performance with multiple-choice knapsack problems [43]. Therefore, we use a greedy approach in which we define parameters denoting the ratio of the power relative to the bandwidth and try to select that with the best ratio. Our solution consists of two phases, in which we consider **C1** alone in the first phase and **C2** step by step in the second phase. For this purpose, we introduce the temporary variables I_k, ($k = 1, \ldots, N_d$) to indicate that the I_kth element in an array of A_k is selected for disk k.

The details of the bandwidth allocation algorithm (BAA) are presented in Fig. 1. For the first phase, we define a series of parameters $Q^{\mathrm{F}}_{k,m}$ for each disk k, ($k = 1, \ldots, N_d$ and $m = 1, \ldots, N_k - 1$) as:

$$Q^{\mathrm{F}}_{k,m} = \frac{P_{k,m}(L_{k,m}) - P_{k,N_k}(L_{k,N_k})}{B_{k,N_k} - B_{k,m}}.$$

```
1: List of values, $Q^{F}_{k,m}$: $G^{F}$;
2: List of values, $Q^{S}_{k,m}$: $G^{S}$;
3: Temporary variable: $I_k$ $(k = 1, ..., N_d)$;
4: Temporary variable: $V$ and $W$;
5: $X_{k,m}$ are all initialized to 0, $(k = 1, ..., N_d$ and $m = 1, ..., N_k)$;
6: $I_k$ are all initialized to $N_k$, $(k = 1, ..., N_d)$;
7: $V \leftarrow \sum_{k=1}^{N_d} B_{k,I_k}$;
8: while $V > 1$ do
9:     Find the lowest value of $Q^{F}_{k,L} \in G^{F}$ and remove $Q^{F}_{k,L}$ from $G^{F}$;
10:    if $L < I_k$ then
11:        $V \leftarrow V + B_{k,L} - B_{k,I_k}$;
12:        $I_k \leftarrow L$;
13:    end if
14: end while
15: $W \leftarrow \sum_{k=1}^{N_d} ch_{k,I_k}$;
16: while $W > N_c$ do
17:     Find the lowest value of $Q^{S}_{k,L} \in G^{S}$ and remove $Q^{S}_{k,L}$ from $G^{S}$;
18:     if $V + B_{k,L} - B_{k,I_k} \leq B_{ssd}$ and $ch_{k,I_k} = 0$ then
19:         $V \leftarrow V + B_{k,L} - B_{k,I_k}$;
20:         $W \leftarrow W - 1$;
21:         $I_k \leftarrow L$;
22:     end if
23:     if $G^{S} = \phi$ then
24:         Break the loop;
25:     end if
26: end while
27: for $k = 1$ to $N_d$ do
28:     $X_{k,I_k} \leftarrow 1$;
29: end for
```

FIGURE 1. **Bandwidth allocation algorithm (BAA).**

The values of $Q^{\mathrm{F}}_{k,m}$ represent the ratio of power difference to bandwidth difference, and the values with the best ratio are selected one by one.

I_k are all initialized to N_k, which requires the lowest power consumption but the highest SSD bandwidth. Thus, some of the I_k values can be decreased to meet the bandwidth requirement, **C1**. We first select the most profitable change so that the lowest value of $Q^{\mathrm{F}}_{k,L}$ is searched in order to replace the value I_k by L, which minimizes the power consumption while maximizing the decrease in the SSD bandwidth. This operation is repeated until constraint **C1** is satisfied (lines 8–14 in Fig. 1).

In the second phase, if condition **C2** does not hold, then some of the I_k values may be changed to reduce the number of speed changes. Thus, we establish new parameters $Q^{\mathrm{S}}_{k,m}$ as follows:

$$Q^{\mathrm{S}}_{k,m} = P_{k,m}(L_{k,m}) - P_{k,I_k}(L_{k,I_k}),$$

which are only defined when $ch_{k,m} = 0$ and $ch_{k,I_k} = 1$. We find the lowest value, $Q^{\mathrm{S}}_{k,L}$, from a set of $Q^{\mathrm{S}}_{k,m}$ values and change the value of I_k to L if $ch_{k,L} = 0$. This minimizes the increase in power consumption while the speed of the disk k remains unchanged. These steps are repeated until constraint **C2** is satisfied (lines 16–26 in Fig. 1).

The BAA runs when a new client is accepted for video service in order to accommodate new workload changes immediately. However, executing the BAA may require that the speeds of some disks change, resulting in frequent speed changes in the long term. Thus, speed changes are allowed only periodically in order to prevent this problem, where with every αth execution of the BAA, the value of N_c is set to an upper bound value, U, $(U > 0)$; otherwise, N_c is set to 0.

VI. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

We evaluated the effectiveness of our scheme through simulations. The disk had a maximum data transfer rate of 170 MB/s [4] and a typical seek time of 8.5 ms [25]. However, we considered a range of up to five speeds between 2880 and 7200 RPM in order to analyze the effects of various speed configurations. For this purpose, we assume that disk power has a quadratic relationship with speed [6]–[8], [25], and calculated the power at a lower speed as shown in Table 2.

TABLE 2. Performance and power characteristics at each RPM level.

speed level	1	2	3	4	5
RPM	2880	3960	5040	6120	7200
$r(l)$	68 MB/s	93.5 MB/s	119 MB/s	144.5 MB/s	170 MB/s
$P_i(l)$	4.08 W	5.11 W	6.48 W	8.17 W	10.2 W
$P_a(l)$	7.38 W	8.41 W	9.78 W	11.47 W	13.5 W
$P_s(l)$	7.38 W	8.41 W	9.78 W	11.47 W	13.5 W

We compared the BAA with two other bandwidth allocation methods based on our model as follows:

1) Lowest bit-rate first selection (LS): As described in Subsection V-A, for disk power reduction, it is advantageous to serve low bit-rate requests from the SSD. Thus, the LS scheme selects the request with the lowest bit-rate first, subject to the SSD bandwidth limitation.
2) Uniform selection (US): Suppose that I_k is an index of the selected element in A_k. The US scheme increments the I_k values of all the disks one by one until the SSD bandwidth limit is not exceeded.

We considered a disk array consisting of 50 multi-speed disks, each of which corresponded to one of the following five sets of RPM configurations: {2880, 7200}, {5040, 7200}, {2880, 5040, 7200}, {2880, 5040, 6120, 7200}, and {2880, 3960, 5040, 6120, 7200}. Unless stated otherwise, in a Zipf distribution, θ was set to 0.271, which was measured for a real VoD application [40]. Videos typically last between 1 and 2 h. Therefore, the length of each video was selected randomly from this range.

A server was assumed to serve 1000 video files with randomly chosen bit rates between 10.4 and 20.8 Mb/s, which is typical of HD quality videos. The idle power of the SSD was 0.03 W, whereas power for the SSD read operation was 2.1 W, based on real measurements [47]. The round length was 2 s and N_p was 12 s [25]. The speed transition time was 6 s and the transition power was 21 W [7], [25]. The arrival of client requests was modeled as a Poisson process [41], [46] with an average arrival rate for requests of 0.4/s. U was set to 1. We profiled the disk power consumption over all the disks for 10 h.

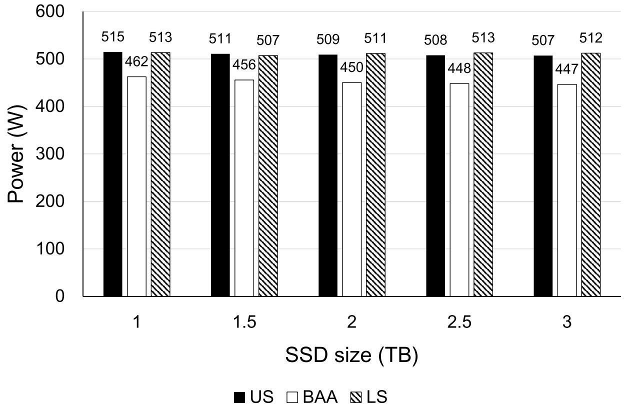

FIGURE 2. Disk power consumption against SSD size.

B. EFFECT OF SSD SIZE

Fig. 2 shows the variations in disk power consumption with each algorithm versus the SSD size when $B_{\text{ssd}} = 1$ GB/s, where the results indicate the following:

1) BAA always performed the best. Specifically, it reduced the power consumption from 10 to 13% compared to the LS scheme and from 10 to 12% compared to the US scheme. To support the high request rates for popular video files stored on the cache, all the algorithms attempted to serve as many requests from the SSD as possible, allowing the algorithm to be executed until SSD bandwidth could be utilized fully. However, BAA performed better than the other schemes because it chose disk speed to minimize the total disk power consumption.
2) When using the BAA and US schemes, the reduction in power consumption increased with the SSD size. However, the effect of the SSD size was negligible for the LS scheme.
3) The difference in power consumption between the BAA and the other algorithms increased with the SSD size, suggesting that the effect of the BAA was more pronounced, especially when many videos were stored on the SSD.
4) The difference in the power consumption between US and LS schemes was below 1%.

C. EFFECT OF SSD BANDWIDTH

Fig. 3 shows the disk power consumption versus the SSD bandwidth when $S_{\text{ssd}} = 2.5$ TB, where the results indicate the following:

1) BAA always performed the best. Specifically, it reduced the power consumption from 11 to 14% when compared to the other two schemes. If the storage

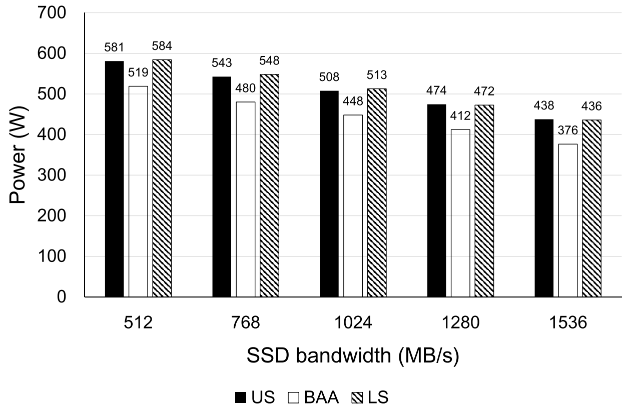

FIGURE 3. **Disk power consumption against SSD bandwidth.**

accounts for 40% of the total cloud power consumption [2], [3], this reduction corresponded to a reduction of 4.4 to 5.6% of total power consumption.

2) Clearly, the reduction in the disk power consumption increased with the SSD bandwidth, which allowed more requests to be served by the SSD cache.
3) The difference in power consumption between the BAA and the other schemes was maximized when the SSD bandwidth was moderate ($B_{\text{ssd}} = 1.25$ GB/s and $B_{\text{ssd}} = 1.5$ GB/s). With sufficient SSD bandwidth, the lowest speeds could be selected for most of the disks, whereas most of the disks required the highest speeds to handle the high demand for disk requests when the SSD bandwidth was low. However, when the bandwidth was moderate, the selection of the disk speed had a tremendous effect on power consumption. The BAA selected the speed levels with the aim of minimizing power consumption, thereby obtaining greater power savings compared with the other schemes, particularly when the bandwidth was moderate.

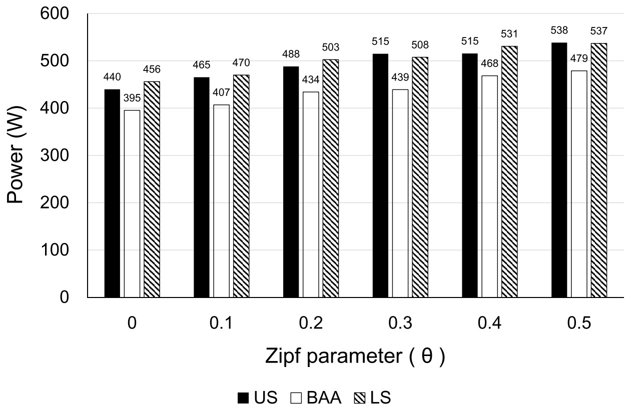

FIGURE 4. **Disk power consumption against Zipf parameters.**

D. EFFECTS OF ZIPF PARAMETERS

In a Zipf distribution, the degree of popularity skewness decreases with the value of the skewness parameter, θ [40]. Fig. 4 shows the disk power consumption when each algorithm was used with different values of θ when $S_{\text{ssd}} = 2$ TB and $B_{\text{ssd}} = 1$ GB/s. Based on the results, we make the following observations:

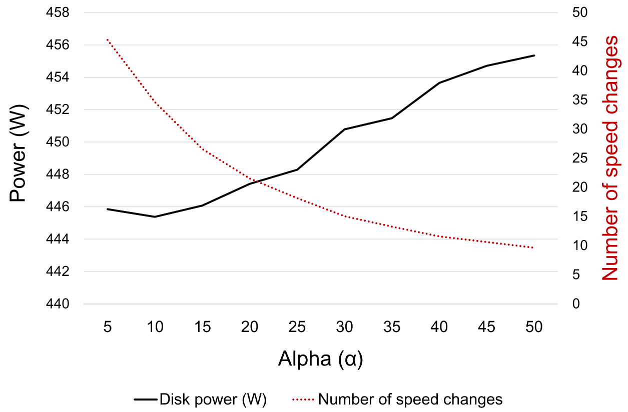

FIGURE 5. **Number of speed changes against α.**

1) BAA always performed the best. Specifically, it reduced power consumption from 11 to 14% when compared to the LS scheme and from 9 to 15% when compared to the US scheme.
2) The disk power consumption increased with the values of θ because low θ values make a few videos very popular, which improves the effectiveness of SSD caching.
3) The difference between the power consumption of LS and BAA decreased with the value of θ, suggesting that the BAA can handle skewed access patterns efficiently.
4) When the degree of popularity skewness was high, US performed better than LS.

E. EFFECTS OF α VALUES

Executing algorithms frequently may increase the number of disk speed changes, thereby affecting disk longevity. Therefore, we examined how the number of speed changes and disk power consumption varied with respect to the value of α when $S_{\text{ssd}} = 2$ TB and $B_{\text{ssd}} = 1$ GB/s, as shown in Fig. 5. Based on these results, we make the following observations:

1) The number of disk speed changes decreased with the value of α because the number of BAA executions, which may require disk speed changes, decreased with the value of α.
2) The power consumption generally increased with the value of α. The BAA attempted to minimize disk power consumption. Therefore, more executions of the BAA resulted in greater energy savings.
3) When the value of α decreased from 10 to 5, the energy consumption was rather increased as a result of an increase in energy for speed transition. For example, the amount of transition power was 6.06 W when $\alpha = 10$ and 7.93 W when $\alpha = 5$. However, the power reduction achieved by a greater number of executions of the algorithm did not offset this increase.

4) There was a trade-off between power consumption and the number of speed changes with respect to the value of α. This provides a guideline for determining the value of α. For example, if the number of speed changes is limited to 20 for 10 h in order to guarantee the specified disk lifetime shown in Fig. 5, then the value of α is determined to be 25.

VII. CONCLUSION

In this study, we developed a new power management scheme for a video server in which an SSD is used as a cache for a disk array with multi-speed disks. First, we examined how the power consumption at each disk speed level varies with the amount of SSD bandwidth allocated to each disk. We then formulated an optimization problem with the purpose of minimizing the overall disk power consumption subject to the SSD bandwidth while limiting the number of speed changes. We then proposed a two-phase algorithm to solve this problem. The first step was to find the best configuration of the SSD bandwidth allocation for each disk in order to determine the requests that can be served by the SSD. The second step was to ensure that the number of disk speed changes is limited.

Our experimental results showed that the proposed BAA could reduce the power consumption by between 10 and 15% as compared to two standard algorithms that we derived from our model. Our results also demonstrated that: 1) the effectiveness of the proposed scheme is pronounced with a large SSD size, adequate SSD bandwidth, and skewed access patterns; and 2) a trade-off exists between variations in power consumption and disk speed changes based on the number of times the algorithm is executed. In an SSD cache, making effective use of its bandwidth is essential to support the high request rates for popular video files stored on the cache. We believe that our results provide useful guidelines for constructing power-efficient video servers by employing an SSD cache and multi-speed disks.

Frequent cache replacements produce many write operations, thus causing the SSD to wear out at a quicker rate. As a future study, we will extend our scheme to consider this wear-out effect. One possible method is to use a throttling technique by limiting the number of cache replacements with the aim of maximizing cache hit ratio. We also plan to extend our scheme to support various types of storage architecture, including hot and cold disk farms.

REFERENCES

[1] *37 Mind Blowing YouTube Facts, Figures and Statistics—2018*. Accessed: Apr. 26, 2018. [Online]. Available: https://fortunelords.com/youtube-statistics/

[2] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 732–794, 1st Quart., 2016.

[3] Z. Li, K. Greenan, A. Leung, and E. Zadok, "Power consumption in enterprise-scale backup storage systems," in *Proc. USENIX Conf. File Storage Technol.*, Feb. 2012, pp. 1–7.

[4] M. Khatib and Z. Bandic, "PCAP: Performance-aware power capping for the disk drive in the cloud," in *Proc. USENIX Conf. File Storage Technol.*, Feb. 2016, pp. 227–240.

[5] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes, "Hibernator: Helping disk arrays sleep through the winter," *ACM Operat. Syst. Rev.*, vol. 39, no. 5, pp. 177–190, 2005.

[6] Q. Zhu and Y. Zhou, "Power-aware storage cache management," *IEEE Trans. Comput.*, vol. 54, no. 5, pp. 587–602, May 2005.

[7] S. Gurumurthi, "Power management of enterprise storage systems," Ph.D. dissertation, Dept. Comput. Sci. Eng., Pennsylvania State Univ., State College, PA, USA, 2005.

[8] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "Reducing disk power consumption in servers with DRPM," *Computer*, vol. 36, no. 12, pp. 59–66, Dec. 2003.

[9] *IntelliPower*. Accessed: Feb. 8, 2018. [Online]. Available: https://support.wdc.com/knowledgebase/answer.aspx?id=3665

[10] *Automaid Technical Brief*. Accessed: May 8, 2018. [Online]. Available: https://www.nexsan.com/wp-content/assets/white-papers/AutoMAID-Technical-Brief-White-Paper.pdf

[11] T. Bosten, S. Mullender, and Y. Berber, "Power-reduction techniques for data-center storage systems," *ACM Comput. Surv.*, vol. 45, no. 3, 2013, Art. no. 33.

[12] S. Lee and J. Kim, "Effective lifetime-aware dynamic throttling for NAND flash-based SSDs," *IEEE Trans. Comput.*, vol. 65, no. 4, pp. 1331–1334, Apr. 2016.

[13] M. Ryu, H. Kim, and U. Ramachandran, "Impact of flash memory on Video-on-Demand storage: Analysis and tradeoffs," in *Proc. ACM Multimedia Syst. Conf.*, pp. 175–186, 2011.

[14] M. Ryu, H. Kim, and U. Ramachandran, "Why are state-of-the-art flash-based multi-tiered storage systems performing poorly for HTTP video streaming?" in *Proc. ACM NOSSDAV Conf.*, Jun. 2012, pp. 3–8.

[15] M. Ryu and U. Ramachandran, "FlashStream: A multi-tiered storage architecture for adaptive HTTP streaming," in *Proc. ACM Multimedia Conf.*, 2013, pp. 313–322.

[16] W. Li, G. Baptise, J. Riveros, G. Narasimhan, T. Zhang, and M. Zhao, "CacheDedup: In-line deduplication for flash caching," in *Proc. USENIX Conf. File Storage Technol.*, Feb. 2016, pp. 301–314.

[17] D. Arteaga, J. Cabrera, J. Xu, S. Sundararaman, P. Machines, and M. Zhao, "CloudCache: On-demand flash cache management for cloud computing," in *Proc. USENIX Conf. File Storage Technol.*, Feb. 2016, pp. 355–369.

[18] R. Manjunath and T. Xie, "Dynamic data replication on flash SSD assisted Video-on-Demand servers," in *Proc. IEEE Int. Conf. Comput., Netw. Commun.*, Feb. 2012, pp. 502–506.

[19] E. Pinheiro and R. Bianchini, "Energy conservation techniques for disk-array-based servers," in *Proc. ACM/IEEE Conf. Supercomput.*, Jun. 2004, pp. 88–95.

[20] E. Pinheiro, R. Bianchini, and C. Dubnicki, "Exploiting redundancy to conserve energy in storage systems," *ACM Perform. Eval. Rev.*, vol. 4, no. 1, pp. 15–26, Jan. 2006.

[21] C. Weddle, M. Oldham, J. Qian, A. Wang, P. Reither, and G. Kuenning, "PARAID: A gear-shifting power-aware RAID," *ACM Trans. Storage Syst.*, vol. 3, no. 3, pp. 13–33, 2007.

[22] T. Xie, "SEA: A striping-based energy-aware strategy for data placement in RAID-structured storage systems," *IEEE Trans. Comput.*, vol. 57, no. 6, pp. 748–761, Jun. 2008.

[23] Y. M. Lee Song and E. Kim, "Saving disk energy in video servers by combining caching and prefetching," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 10, no. 1s, p. 15, 2014.

[24] M. Song, "Energy-aware data prefetching for multi-speed disks in video servers," in *Proc. ACM Multimedia Conf.*, Oct. 2007, pp. 755–758.

[25] M. Kim and M. Song, "Saving energy in video servers by the use of multispeed disks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 567–580, Apr. 2012.

[26] H. Yuan, I. Ahmad, and C. J. Kuo, "Performance-constrained energy reduction in data centers for video-sharing services," *J. Parallel Distrib. Comput.*, vol. 75, no. 1, pp. 29–39, 2015.

[27] C. Li, P. Shilane, F. Douglis, H. Shim, S. Smaldone, and G. Wallace, "Nitro: A capacity-optimized SSD cache for primary storage," in *Proc. USENIX Annu. Tech. Conf.*, Jun. 2014, pp. 501–512.

[28] F. Meng, L. Zhou, X. Ma, S. Uttamchandani, and D. Liu, "vCacheShare: Automated server flash cache space management in a virtualization environment," in *Proc. USENIX Annu. Tech. Conf.*, Jun. 2014, pp. 133–144.

[29] T. Kim and E. Kim, "Hybrid storage-based caching strategy for content delivery network services," *Multimedia Tools Appl.*, vol. 74, no. 5, pp. 1697–1709, 2015.

[30] L. Lin, Y. Zhu, J. Yue, Z. Cai, and B. Segee, "Hot random off-loading: A hybrid storage system with dynamic data migration," in *Proc. IEEE Modeling, Anal., Simulation Comput. Telecommun. Syst. Symp.*, Jul. 2011, pp. 133–144.

[31] S. Yin *et al.*, "REED: A reliable energy-efficient RAID," in *Proc. IEEE Int. Conf. Parallel Process.*, Sep. 2015, pp. 649–658.

[32] S. He, Y. Wang, and X.-H. Sun, "Improving performance of parallel I/O systems through selective and layout-aware SSD cache," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 10, pp. 2940–2952, Oct. 2016.

[33] T. Kgil, D. Roberts, and T. Mudge, "FlashCache: A NAND flash memory file cache for low power Web servers," in *Proc. IEEE Int. Conf. Compil., Archit. Synth. Embedded Syst.*, Nov. 2006, pp. 103–112.

[34] T. Kgil, D. Roberts, and T. Mudge, "Improving NAND flash based disk caches," in *Proc. IEEE Int. Symp. Comput. Archit.*, Aug. 2008, pp. 186–190.

[35] L. Useche, J. Guerra, M. Bhadkamkar, M. Alarcon, and R. Rangaswami, "EXCES: External caching in energy saving storage systems," in *Proc. IEEE 14th Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2008, pp. 89–100.

[36] J. Hui, X. Ge, X. Huang, Y. Liu, and Q. Ran, "E-HASH: An energy-efficient hybrid storage system composed of one SSD and multiple HDDs," in *Proc. Int. Conf. Swarm Intell.*, Jun. 2012, pp. 527–534.

[37] J. Lee and M. Song, "Cache management for video servers by the combined use of DRAM and SSD," in *Proc. IEEE Int. Symp. Multimedia*, Dec. 2016, pp. 537–540.

[38] C. Chen, D. Xiong, K. Zhao, C. W. Chen, and T. Zhang, "Realizing low-cost flash memory based video caching in content delivery systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 4, pp. 984–996, Apr. 2018.

[39] J. Schindler, J. Griffin, C. Lumb, and G. Ganger, "Track-aligned extents: Matching access patterns to disk drive characteristics," in *Proc. USENIX Conf. File Storage Technol.*, 2002, pp. 175–186.

[40] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic batching policies for an on-demand video server," *Multimedia Syst.*, vol. 4, no. 3, pp. 112–121, 1996.

[41] Y. Zhou, L. Chen, C. Yang, and D. Chiu, "Video popularity dynamics and its implication for replication," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1273–1285, Aug. 2015.

[42] L. Kleinrock, *Queueing Systems: Theory*, vol. 1. Hoboken, NJ, USA: Wiley, 1975.

[43] D. Pisinger, "Algorithms for knapsack problems," Ph.D. dissertation, Dept. Manage. Eng., Univ. Copenhagen, København, Denmark, 1995.

[44] C. Ruemmler and J. Wilkes, "An introduction to disk drive modeling," *Computer*, vol. 27, no. 3, pp. 17–28, Mar. 1994.

[45] A. Hylick and R. Sohan, "A methodology for generating disk drive energy models using performance data," in *Proc. ACM Workshop Power Aware Comput. Syst. (Hotpower)*, 2009, p. 100.

[46] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," in *Proc. ACM Eur. Conf. Comput. Syst.*, Apr. 2006, pp. 333–344.

[47] *250GB SSD Review: 33 Models Compared*. Accessed: Feb. 10, 2018. [Online]. Available: https://us.hardware.info/reviews/7142/19/250gb-ssd-review-33-models-compared-power-consumption

MINSEOK SONG (M'07) received the B.S., M.S., and Ph.D. degrees in computer engineering from Seoul National University, South Korea, in 1996, 1998, and 2004, respectively.

Since 2005, he has been with the Department of Computer Engineering, Inha University, Incheon, South Korea, where he is currently a Professor. His research interests include embedded systems and multimedia systems.