Power-aware Proactive Storage-tiering Management for High-speed Tiered-storage Systems

Kazuhisa Fujimoto[†], Hirotoshi Akaike[††], Naoya Okada[†], Kenji Miura[†], and Hiroaki Muraoka[†]
† Research Institute of Electrical Communication, Tohoku University
†† Systems Development Laboratory, Hitachi Ltd.

Abstract

Large-scale high-speed mass-storage systems account for a large part of the energy consumed at data centers. To conserve energy consumed by these storage systems, we propose a high-speed tiered-storage system with a power-aware proactive method of storage-tiering management that minimizes loss of performance, which we have called the energy-efficient High-speed Tiered-Storage system (eHiTS). eHiTS consists of a tiered-storage system with high-speed online storage as the first tier and low-power nearline storage with high capacity as the second tier. All files are always stored in nearline storage when it is created, in which the hard disk drives are usually left powered off. Based on hints from a high-performance computing (HPC) application, only the volume that includes the accessed files (datasets) is copied from nearline to online storage before access. The results obtained from our testbed with 64-TB capacity revealed that eHiTS was able to conserve up to 16% of the energy consumed by an ordinary tiered-storage system with the same capacity. This corresponded to a 55%-energy saving in 1-PB capacity.

1. Introduction

The amount of electrical energy consumed by storage systems has been rapidly increasing at data centers [1]. Large high-speed storage systems, especially, consume more electrical energy. The main source of electrical energy consumed by storage systems derives from the numerous hard disk drives (HDDs) incorporated into them. These days, large high-speed storage systems have PB-class capacity with thousands of HDDs. The electricity charges per year amount to several tens of thousands of dollars[1]. Conserving the electrical energy consumed by large high-speed storage systems is therefore a huge challenge in research to reduce running costs as well as conserve energy at data centers. A representative work on energy conservation in storage systems is the Massive Array of Idle Disks (MAID), in which a spindle motor is stopped so that the HDDs are in a standby state during no-access periods [2]. In MAID systems [3], a few HDDs in the idle state (always accessible) are used for caching data and a large numbe of HDDs in the standby state are used for storing data. When a request misses in the cache HDDs, it is processed after the accessed HDD has spun up. Therefore, the response time in MAID systems is deteriorated in cache misses, resulting in very limited use for high-speed applications. In another approach, low revolutions per minute (rpm) and large capacity HDDs are used as the second tier of tiered-storage systems due to their low-power consumption per capacity, i.e., tiered-storage systems with information lifecycle management (ILM) [4–5]. These systems consist of first-tier high-speed storage with high-speed high-power HDDs and second-tier mass storage with low-rpms and high-capacity HDDs. Rarely accessed data are relocated from first-tier to second-tier storage to control the increase in the first-tier's capacity and reduce the total energy consumed by the systems. An issue with this approach is minimizing the capacity of high-speed storage systems while minimizing reduced performance.

This paper proposes a power-aware proactive method of storage-tiering management to minimize the capacity of first-tier high-speed storage systems in tiered-mass-storage systems. Our proposed method has features as follows: Like the tiered-storage system with ILM, eHiTS consists of a tiered-storage system with high-speed online storage as a first tier and low-power nearline storage with high capacity as the second tier. However, all files are always stored in nearline storage when it is created, in which the HDDs are usually left powered off, not spun down, to save more energy than the MAID systems do. Only the volume that has stored the accessed files is powered on and copied from nearline to online storage before access. The timing to start copying is predicted based on the job-queue status and statistics of job submission and execution in the batch-job scheduler of a batch-processing application. In terms of these, data movement and its timing with our proposed method are managed inversely against ILM where files are stored in online storage when it is

[1] Estimation assuming that an electricity charge is 10-cents/ kWh and the electric power consumed by an HDD in the idle state is 10-15 W.

created and relocated to nearline storage when utilization becomes lower than a predetermined threshold based on the user policy. Even though the first-tier's online storage is used as a data cache in eHiTS, file requests with our proposed method are hit in online storage (cache storage) even during the first access due to the proactive copying of the accessed volume to online storage (cache storage) before access, unlike in ordinary cache management used in general MAID systems. Moreover, the accessed volume is copied back soon after the access period ends. This leads to the capacity of online storage being minimized with minimum loss of performance, resulting in energy savings. Note that the period that files are frequently accessed by the job (which corresponds to the period of the job execution), not each file-access, are predicted in our proposed method.

We selected an HPC system as the first target in the batch-processing applications to evaluate eHiTS because one of the recent major issues in the HPC environment has been to reduce the increasing energy consumed by storage systems due to their much larger capacity as well as higher performance.

This paper is organized as follows. Section 2 summarizes related work and our research is compared with similar investigations. Section 3 proposes the new storage-tiering management for eHiTS. Sections 4 discuss our evaluations of the proposed method. Section 5 concludes the paper and describes future work.

2. Related Work

A great deal of work has been done related to energy conservation in disk-storage systems. One of the main issues with energy conservation in HDDs is how to maximize the time period that HDDs are in a standby state. Methods of maximizing this time period are achieved by exploiting access locality [2], [6–7]. In other methods, the idle time, which means the time period without disk access, is extended to keep HDDs in a standby state longer. Methods to extend the idle time in storage systems have been proposed [8–10]. Also, efficient control of HDD states with hints from the application layer has been proposed to balance performance with energy conservation [11]. A method of using application hints has been proposed to reduce the read latency [12]. However, this cannot be used in eHiTS because it identifies which data will be accessed but not when.

Our approach is similar to that of GreenStor [11] proposed by Mandagere et al. in that it relocates the accessed data in a MAID system efficiently based on hints from applications to conserve energy. Our proposed approach, however, is different from that of Greenstor. The staging timing for the requested data is controlled with application hints but the write-data destaging does not have any specific deadline in Greenstor's disk-cache management. They also assumed that applications provide approximate access timing in the future as hints. But, they do not describe how access timing is predicted. In our proposed method, the accessed volume stays in online storage as short as possible to minimize the size of the disk cache. The accessed volume cached in online storage is copied back soon after the access period ends. We also describe how the timing of the accessed volume copied from nearline to online storage is autonomously predicted and managed in the storage system by exploiting information about the job-queue.

3. Architecture of eHiTS

3.1. Power-aware Storage-tiering Management

Figure 1 outlines the architecture of eHiTS with an HPC system for scientific calculations. The HPC system consists of a supercomputer with an HPC-management server. The eHiTS consists of high-speed online storage and high-capacity nearline storage with low-power consumption, where online storage tiered with nearline storage and network-attached storage (NAS) has been set up in front of the online storage to offer file access to the supercomputer and users. The eHiTS has a storage-management server to control data allocation between the online and nearline storage as well as manage the online and nearline storage and NAS.

Jobs executed in the supercomputer in the HPC sys-

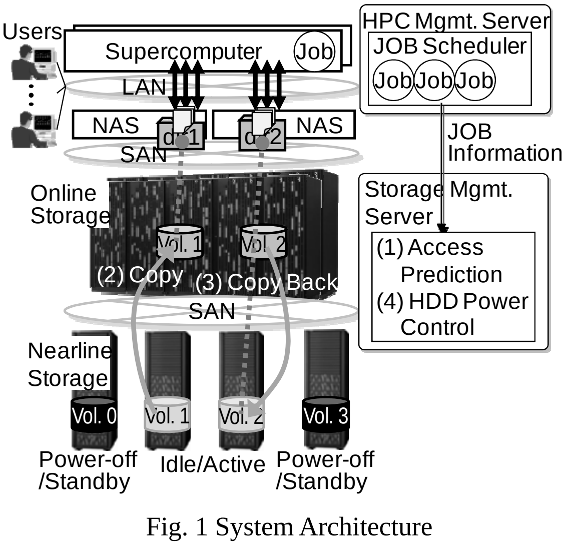


Fig. 1 System Architecture

tem are controlled by a job scheduler that works on the HPC management server. The jobs submitted by users are classified into multiple classes according to the user-specified execution time, the number of CPUs, and the memory allocation size to input them into a queue for each class, as shown in Fig. 2. User jobs are controlled in the queue and executed in sequence from the front of the queue.

The accessed files (datasets) or directories must be copied from nearline to online storage before the job starts execution because the input files for calculation are read soon after the job has started. These are specified from the name described in the job script that users submit. Using the name, the volume mounted to the accessed directory is also specified in the storage management server. By exploiting the job-queue status and statistics of job submission and execution in each queue, the timing to start copying the accessed volume to online storage is predicted in eHiTS for each user-submitted job in each queue. These are labeled "Access Prediction (1)" in Fig. 1.

eHiTS is equipped with another feature where the HDD enclosure is powered off, not spun down, when there is no access to gain larger energy savings than in MAID systems. All files in eHiTS are always stored in nearline storage when they are created, in which the HDD enclosures are usually left powered off. Before the user volume is accessed by the job, it is copied to online storage and the mounting point for the user's directory is changed to the copied volume in online storage to offer high-speed access. These are labeled "Copy (2)" in Fig. 1. In eHiTS, the input files accessed by one user-submitted job are placed together in one user directory. A volume is mounted to the user directory and it must be sufficiently large so that the large output files from the job in HPC application can also be stored together.

Before a volume is copied, the HDD enclosure that includes the volume is powered on. After copying, it is powered off again. This is labeled "HDD Power Control (4)" in Fig. 1.

After a job has completed execution and if the accessed volume will not used by any jobs in the multiple queues, the copy and mounting processes are done inversely. That the job has completed can easily be detected by periodically checking the job status in the job-queueing system of the job scheduler. The volume that had been copied to online storage before the job started is copied back to the former volume in nearline storage and it is remounted to the user directory. These are labeled "Copy Back (3)" in Fig. 1.

The eHiTS is also accessed by users as well as jobs. We have assumed user login is detected in the HPC system to identify user access to eHiTS. The HDD enclosure that includes the user volumes is powered on and stays in an idle state during user login because it is very difficult to predict when the users will access their files during login.

Figure 3 illustrates the power-consumption model of the HDD enclosure in eHiTS (a) and that of an ordinary tiered-storage system with ILM (b). The power consumed by HDD enclosures in online storage usually equals the power consumption in the idle state of HDDs (P_{i_OL}) in both systems. The power consumed by HDD enclosures in online storage in both systems varies between the power consumption in the active state of HDDs (P_{a_OL}) and P_{i_OL} during job execution due to file access from the supercomputer.

In eHiTS, HDD enclosures in nearline storage

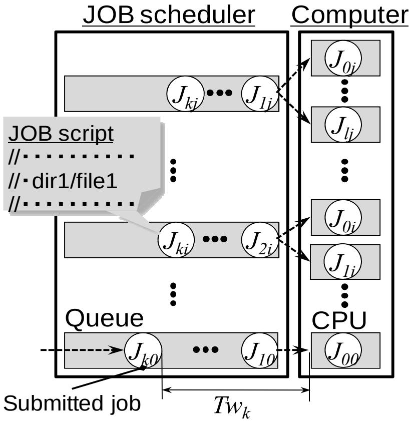
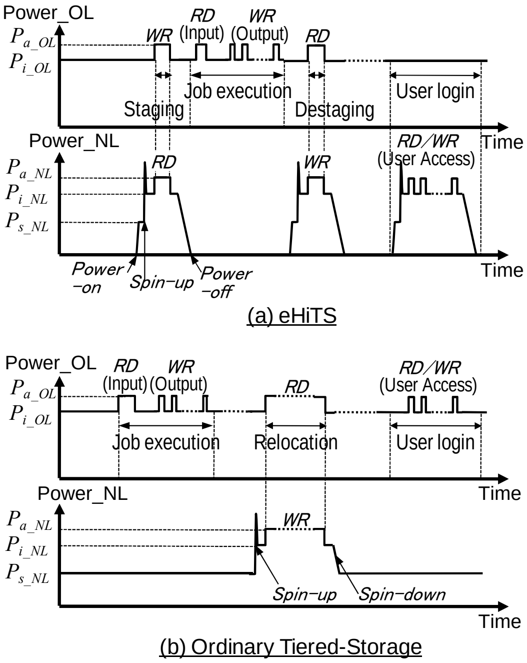


Fig. 2 Model of Job-queues in Job Scheduler



(a) eHiTS

(b) Ordinary Tiered-Storage

Fig. 3 Power Consumption Model

usually consume no power. Power is only consumed when volumes are copied to online storage (staging) and nearline storage (destaging) and during user login. When the HDD enclosure is powered on, the power varies to the power consumption in standby state of HDDs (P_{s_NL}). After this, the power varies to P_{i_NL} with the spin-up of HDDs. Then, the power varies between P_{a_NL} and P_{i_NL} during staging. Synchronized with this, the power consumed by HDD enclosures in online storage varies. During destaging, the power consumed by HDD enclosures in online and nearline storage varies as it does in staging. During user login, the power consumed by HDD enclosures in nearline storage varies between P_{a_NL} and P_{i_NL} due to user accesses.

In contrast to eHiTS, the power consumed by HDD enclosures in the nearline storage of the ordinary tiered storage system usually equals P_{s_NL} with a spin-down feature or P_{i_NL} without the feature. Before rarely accessed data is relocated to nearline storage, the power varies from P_{s_NL} to P_{i_NL} due to spin-up and varies between P_{a_NL} and P_{i_NL} during the relocation to nearline storage. Synchronized with this, the power consumed by HDD enclosures in online storage varies. During user login, the power consumed by HDD enclosures in online storage varies between P_{a_OL} and P_{i_OL}.

Compared with the ordinary tiered storage, the energy consumed by eHiTS can be conserved due to the reduction of the capacity of online storage where there are fewer HDD enclosures and power-off feature of the HDD enclosure in nearline storage. We discuss details of the energy consumption in Section 4.2.

3.2. Timing to Start Copying before Access

In eHiTS, copying an accessed volume from nearline to online storage must be completed before the job starts. The job scheduler in HPC systems generally has multiple job queues, as shown in Fig. 2. In each job queue of the multiple job queues, the job inter-arrival time and the job execution interval, which determine the job-queue state, could be expressed as an independent probability distribution with different parameters [13]. Therefore, we can determine the timing to start copying the accessed volume in each queue of the multiple job queues independently. Here in Fig. 2, we have focused on the job queue at the bottom. The $k+1$-jobs in the job-queueing system have been expressed. J_{00} is the job being executed and J_{k0} is a job submitted at that time. The waiting time for the submitted J_{k0} job to start to execute is T_{wk}. The time for the accessed volume to copy from nearline to online storage outlined in Fig. 1 (2) is T_{copy}, which equals the sum of the time for the HDD enclosure to power up and the time to copy the volume from nearline to online storage. The copying of an accessed volume is successfully completed from nearline to online storage before access if there are sufficient jobs in the job-queue to be $T_{wk} \geqq T_{copy}$. Unlike the prediction with the observed history of previous waiting times [14], we use a threshold value of queue length, (k_{th}), that satisfies $T_{wk} \geqq T_{copy}$. If the number of jobs in the job-queueing system when a job is submitted is equal to or less than k_{th}, the copying of volumes is started and job execution is delayed for T_{copy}. If not, job execution is not delayed and the copying is started when the order of the job is equal to k_{th}.

In this paper, we derived the minimum value of k_{th} for the desirable probability of $T_{wk} \geqq T_{copy}$, which means the copying is successfully completed before access, from simulating the job-queueing system with the job inter-arrival time and the job execution interval followed by a hyper-exponential and hyper-Erlang distribution in a real, large HPC system [13] by using an event-driven simulator. These distributions could be expressed as a mixture of an exponential and Erlang distribution with a mean value of several minutes and those with a mean value of several tens or hundreds minutes. For example, the former distributions often correspond to that for weekday daytime and evening before weekend and the latter ones often correspond to that for night-time and weekend. Figure 4 plots the miss probability of prediction ($P_{miss}=1-P(T_{wk} \geqq T_{copy})$). The average size of the volumes copied between nearline and online storage was 50 GB and the size was followed by a lognormal distribution between 1 GB and 1 TB. We found that k_{th} should be equal to or more than 12 for the lower utilization factor (ρ) of 0.5 as well as the higher ρ of 0.95 if we could allow the miss probability of less than 10^{-4}. We think it is allowable that the accessed volume cannot completely be copied from nearline to online storage before access every 10,000 job executions which means a miss occurs every 416 days if the average of job execution time is one hour. If there are more than twelve jobs in the job-queueing system when a job is submitted, its execution is not

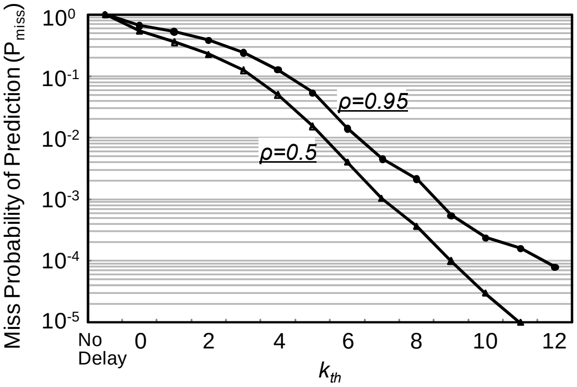


Fig. 4 Miss Probability of Prediction

delayed and the volume accessed by the job starts to copy when the number of jobs ahead of the job reaches twelve. This leads to a maximum of fourteen copied volumes in online storage taking into consideration the volumes copied during staging and destaging. If the average size of directories is 50 GB, which is thought to be sufficiently large in general HPC systems [15-16], the required capacity for online storage is at most 700 GB. If the HPC system has ten job queues, the required capacity of online storage becomes 7 TB.

k_{th} can also be derived dynamically. If we predetermine the desirable probability of $T_{wk} \geqq T_{copy}$ ($P(T_{wk} \geqq T_{copy})$) when the job is submitted, the minimum value of k_{th} can be derived from solving the equation (1) based on queuing-theory numerically. We will study this method in future work.

$$P(T_{wk} \geq T_{copy}) = \sum_{r=0}^{k_{th}-1} v_r(T_{copy}), \quad (1)$$

where $v_r(t)$ is the probability that the r-jobs complete execution during t, which expressed as a function of the mean value of the job execution interval.

4. Evaluation of Energy-conservation Efficiency

4.1. Experimental Setup

Figure 5 illustrates the configuration for the testbed we developed. Table 1 lists the features of the equipment. The tiered-storage system in the testbed consisted of one Hitachi USP VM with 12.6 TB of capacity for online storage and two Hitachi AMS 2500s with 64 TB of capacity in total for nearline storage. The Hitachi AMS 2500s were equipped with a feature where the HDD enclosure was powered on and off by a command from the management software. Two Hitachi Essential NAS Platforms® were set up in front of the online storage. The prototype of the software was equipped with features to predict the timing to start copying the volume, copy the volume between online and nearline storage based on the prediction, and control the power of HDD enclosures in nearline storage and it worked on the storage-management server. It was also equipped with features of ILM in the ordinary tiered storage system to compare the both energy consumption under the same platform. The HPC system consisted of an SGI Altix3700B and a management server with a job scheduler (PBS Pro^{TM}). Two NASs were connected to the Altix3700B via an IP-SW with a 10Gig Ethernet.

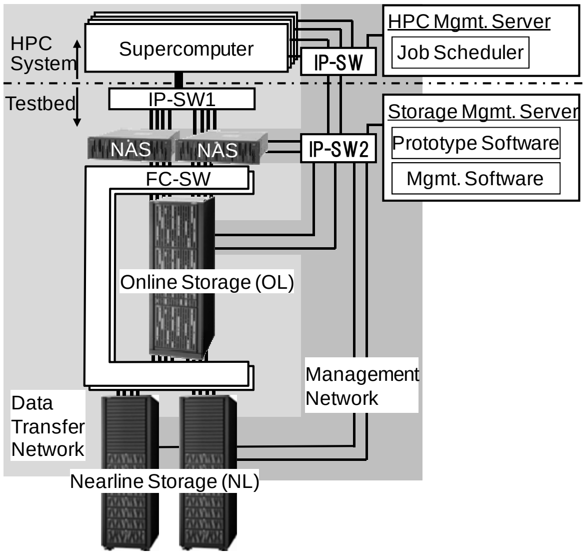


Fig. 5 Configuration of Testbed

The energy consumed by eHiTS was compared in the evaluation with that by an ordinary tiered-storage system with ILM. The power consumed by eHiTS and the ordinary tiered-storage system were measured during the day with a power logger to find how much energy was consumed per day. The power consumed by the controller and HDD enclosures in online and nearline storage were measured independently.

Table 2 lists the parameters for the loads imposed by jobs and users, which affected energy consumption.

Table 1 Features of Equipment

Equipment	Features
NAS	IP: 1 Gbps × 12 ports, FC: 4 Gbps × 4 ports
OL	FC: 4 Gbps × 24 ports, Capacity: 12.6 TB (max.) (HDD: 300 GB-FC/15 krpm × 48, 7D+1P)
NL	FC: 4 Gbps × 8 ports, Capacity: 32 TB (HDD: 1 TB-SATA × 40, 8D+2P)
Storage Mgmt. Server	Windows Server 2003 (Management Software, Prototype Software)

Table 2 Parameters for Evaluation

Parameters	Values
Average Job Inter-arrival Time: 1/λ (min)	*60*
Average Job Execution Interval: 1/μ (min)	*60*
User Login Time: t_{lin} (min)	*480*
File Size (MB)	*100 - 1000*
Accessed Volume Size (GB)	*10*
Relocation Time from online to nearline: t_{relo} (h)	*8*

Table 3 Online and Nearline Storage Capacity for Three Capacity Ratios and Four System Capacities

Capacity ratios (OL:NL)	System Capacities							
	64 TB		256 TB		512 TB		1024 TB	
	OL	NL	OL	NL	OL	NL	OL	NL
8: 120 (Ord. TS)	4.2	60	16.8	240	33.6	480	67.2	960
4: 120 (eHiTS)	2.1	64	8.4	256	16.8	512	33.6	1024
1: 120 (eHiTS)	n/a	64	2.1	256	4.2	512	8.4	1024

The HDD enclosures in nearline storage in the evaluation had 10 HDDs with an 8D+2P RAID6 configuration and each one was only allocated to the fixed multiple users to ease the management of volume copying and power control of HDD enclosures.

We assumed that there would be one job queue for the conditions in the HPC system as mentioned in Section 3.2. To evaluate the energy conserved when all the volume copying were successfully completed before access, the conditions for job submissions and job executions were selected so that T_{wk} was longer than or equal to T_{copy}. That is, both the average inter-arrival time between job submissions and average job execution interval were selected as 60 min and measurement commenced when there were more than three jobs in the job queue so that there would be enough jobs in the queue. Copying the accessed volume was also started when the job submitted.

In the ordinary tiered-storage system with ILM, the relocation of rarely accessed files to nearline storage was carried out regularly, e.g., once a day or once a week. The relocation in the evaluation was carried out once a day. The relocation-time period (t_{relo}) was set to 8 hours overnight.

The most distinct feature of eHiTS is its ability to minimize the capacity of online storage to conserve energy. The eHiTS's energy consumption with a capacity ratio for online to nearline storage of 4 to 120 was compared with that of the ordinary tiered system with a capacity ratio of 8 to 120 to confirm what effect minimizing the capacity of online storage had on conserving energy. The ratio of the ordinary tiered system was determined by assuming that data would seldom be accessed when more than 90 days had elapsed after it had been created [4]. The capacities of online and nearline storage are listed in Table 3 for both capacity ratios and system capacities. Note that the system capacity, which means the usable capacity, is the sum of online- and nearline-storage capacity in an ordinary tiered system and nearline-storage capacity in eHiTS because online storage was used as a cache. The capacity of online storage with 1-to-120 capacity ratio and 1024-TB system capacity is about 8 TB. This value is larger than the required online-storage capacity of 7 TB for the miss probability of less than 10^{-4} mentioned in Section 3.2. Therefore, the miss probability can be sufficiently within reach in this situation.

4.2. Results and Discussion

Figure 6 shows the power consumption measured for nearline storage in eHiTS, where the timings of job submission, execution, and volume copying between nearline and online storage have been indicated. We confirmed that the power-aware proactive storage-tiering method in eHiTS worked successfully. For example, when job1 was submitted, HDD enclosure1 which included the volume accessed by job1 was successfully powered on. After that, the volume copying to online storage started and HDD enclosure1 was powered off after the completion of volume copying. After volume copying was complete, job1 started to execute. This means the supercomputer successfully accessed files included in the copied volume in online storage. After job1 stopped executing, the volume copying to nearline storage started. Before and after that, HDD enclosure1 was also successfully powered on and off.

Figure 7 compares the measured eHiTS's energy consumption per day with that by the ordinary tiered system (Ord. TS) in our testbed. The values indicated in and on the bars in Fig. 7 represent the percentages for the energy consumption against the ordinary tiered-storage system. The bars plot the details on energy conservation. OL-CTL and NL-CTL mean the controllers for online and nearline storage. OL-HDDs and NL-

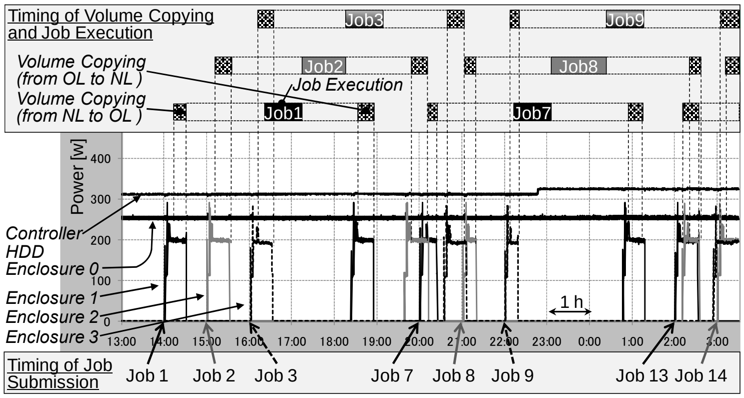
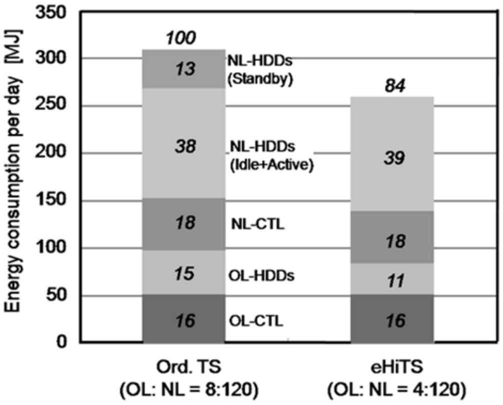


Fig. 6 Experimental Results



* Ord. TS is the ordinary tiered storage system.
* OL:NL means the capacity ratio of online to nearline storage.

Fig. 7 Comparisons of Measured eHiTS's Energy Consumption with Ordinary Tiered System

NDDs mean the HDD enclosures in online and nearline storage. These energy consumptions include the energy consumed by fans, power supplies and switches as well as HDDs. We found that eHiTS with 64-TB capacity and 4-to-120 capacity ratio was able to save up to 16% of the energy consumed by the ordinary tiered-storage system with the same capacity. This 16%-saving came from a 4%-saving due to fewer HDD enclosures in online storage and 12%-saving due to HDD enclosures with power-off feature in nearline storage compared with the ordinary tiered storage system. If the nearline storage of the ordinary tiered-storage system could be equipped with a feature of power-off control, the energy savings of eHiTS decreased to 3% only due to less capacity of online storage.

Figure 8 compares the estimated energy consumed by the eHiTS and ordinary tiered-storage system with 256, 512, and 1024-TB capacities based on the measured energy consumption in our testbed. In the estimates, we assumed that only the number of HDD enclosures was increased to correspond to the increase in system capacity. The controllers were assumed to support up to 1-PB-system capacity. We also assumed that the job scheduler had 10-job-queues and the job execution interval in all the job-queues is the same as that in Table 2.

We found that the energy consumed by eHiTS with a 1-to-120 capacity ratio was 68, 54, and 45% of that by the ordinary tiered system with respective system capacities of 256, 512, and 1024 TB. The energy consumed by OL-CTL and NL-CTL in eHiTS was the same as that by the ordinary tiered-storage system because of the same platform. The energy consumed by the OL-HDDs in eHiTS with 4-to-120 and 1-to-120 capacity ratio were about 70% and 18% of that by the ordinary tiered-storage system in all the system capacities due to the half and eighth capacity of the online storage of the ordinary tiered-storage system. For the 1024-TB system capacity, the energy consumed by the NL-HDDs in eHiTS decreased to 36% compared with 67% in the ordinary tiered-storage system due to power-off control when there was no access. This effect of power-off control became greater for larger system capacity due to the increase in the number of HDD enclosures. Even if MAID systems could also be equipped with a power-off control feature (which corresponds to the case that the energy consumed during standby-state of NL-HDDs is equal to zero in Fig. 8), eHiTS can get larger energy-savings than the ordinary tiered-storage system for larger capacity and capacity ratio.

Figure 9 shows the origin of the energy consumed by eHiTS with the 1-PB-system capacity and 1-to-120-capacity ratio shown in Fig.8. The energy consumed by the NL-HDDs in eHiTS accounted for a significant proportion of 78.2%, in which the largest amount of energy consumption (66.4%) originated from the energy consumption during user login. As mentioned in Section 3.1, the HDD enclosure that included the user volumes stayed in an idle state during user login because it was very difficult to predict when users would access their files during login. Moreover, each enclosure in nearline storage was allocated to the fixed users so that the power-off control could be easily managed in the experiment. Therefore, all the HDD enclosures in nearline storage were powered on during the eight hours of user login. Because of these conditions, the energy consumed by the HDD enclosures in nearline storage accounted for a larger proportion in larger system capacity. To prevent this situation and minimize the number of HDD enclosures that powered on, frequently accessed-volumes by users must be placed together into one or a small number of HDD enclosures. To manage this, the volumes must be moved regularly between HDD enclosures in nearline storage based on the utilization of volumes. We intend to study eHiTS with this kind of management in future work.

On the other hand, the energy consumed by the vo-

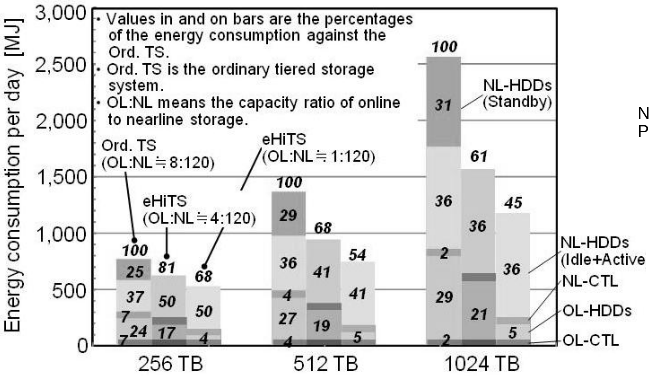


Fig. 8 Comparisons of eHiTS's Energy Consumption with Ordinary Tiered System (Estimation)

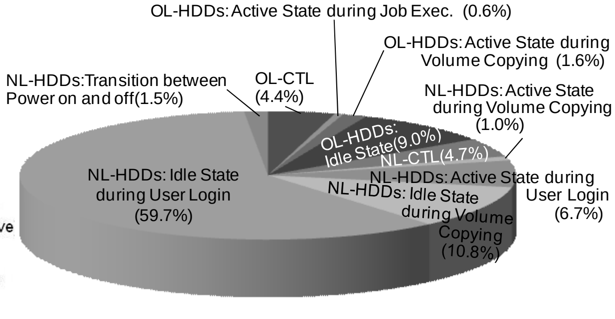


Fig. 9 Origin of Energy Consumed by HDD Enclosures in Nearline Storage

lume copying between online and nearline storage accounted for only 1.6% and 1.0% in online and nearline storage respectively. In the estimates, we assumed that the average size of volumes was 10GB. Both energy consumptions would be about 10% even if the average size is 100GB. It was found that the volume copying between online and nearline storage did not consume much energy in addition to be one of the key features in eHiTS.

5. Conclusions and Future Work

Large-scale high-speed mass-storage systems account for a large part of the energy consumed at data centers. To conserve the energy consumed by these storage systems with minimum degradation in performance, we proposed a power-aware proactive method of storage-tiering management to minimize the capacity of the first tier's online storage with the second tier's nearline storage, which had a feature that lowered power consumption.

We selected an HPC system as the first target application to evaluate eHiTS. We could demonstrate the effectiveness of eHiTS in conserving energy using a testbed developed in a real HPC environment. The results revealed that the energy consumption with eHiTS decreased by as much as 84 % of that of an ordinary tiered-storage system with ILM for a system capacity of 64 TB. This energy conservation corresponded to 55% in 1-PB capacity. This came from 24%-saving due to minimizing online storage capacity and 31%-saving due to HDD enclosures with power-off feature in nearline storage.

We have been improving the energy efficiency and the method of prediction for eHiTS and we intend to evaluate its energy conservation and probability of prediction again. Moreover, we intend to expand eHiTS to other applications that require high speed and energy to be conserved.

Acknowledgements

This work was supported in part by Research and Development for the Next-Generation Information Technology of the Ministry of Education, Culture, Sports, Science and Technology (MEXT). This work has been done in part as a joint research project with Hitachi Ltd.

References

[1] "Report to Congress on Server and Data Center Energy Efficiency Public Law 109-431", U.S. Environmental Protection Agency, ENERGY STAR Program, Aug. 2007.

[2] D. Colarelli and D. Grunwald. Massive Arrays of Idle Disks for Storage Archive. In Proc. ACM/ IEEE Conf. on Supercomputing, pp. 1–11, 2002.

[3] F. Moore and A. Guha. Introducing COPAN Systems MAID Architecture (Massive Array of Idle Disks). White Paper, Copan Systems, 2004.

[4] "Power, Cooling, Space Efficient Storage", ESG Report, Jul. 2007.

[5] M. Peterson. ILM and Tiered Storage. Storage Networking Industry Association, 2006.

[6] E. V. Carrera, E. Pinheiro, and R. Bianchini. Conserving Disk Energy in Network Servers. In Proceedings International Conference on Supercomputing, pp. 86–97, 2003.

[7] E. Pinheiro and R. Bianchini. Energy conservation techniques for disk array-based servers. In Proceedings of the 18th International Conference on Supercomputing, pp. 68–78, 2004.

[8] Q. Zhu and Y. Zhou. Power-Aware Storage Cache Management. IEEE Trans. Computer, 54 (5): 587–602, 2005.

[9] A. E. Papathanasiou and M. L. Scott. Energy Efficient Prefetching and Caching. In Proc. USENIX Tech. Conf., 2004.

[10] A. Weissel, B. Beutel, and F. Bellosa. Cooperative I/O: A novel I/O semantics for energy-aware applications. SIGOPS Oper. Syst. Rev., vol. 36, pp. 117–129, 2002.

[11] N. Mandagere, J. Diehl, and D. Du. GreenStor: Application-Aided Energy-Efficient Storage. In Proceedings of the IEEE/NASA Mass Storage Systems, pp. 16–29, 2007.

[12] R. H. Patterson, G. A. Gibson, and M. Satyanarayanan, "Using transparent informed prefetching to reduce file read latency," in Proceedings of the IEEE/NASA Mass Storage Systems, 1992, pp. 329–342.

[13] J. Jann, P. Pattnaik, H. Franke, F. Wang, J. Skovira, and J. Riodan, Modeling of Workload in MPPs. In Job Scheduling Strategies for Parallel Processing, D. G. Feitelson and L. Rudolph (eds.), pp. 95–116, Springer Verlag, 1997. Lect. Notes Comput.Sci. Vol. 1291.

[14] J. Brevik, D. Nurmi, and R. Wolksi, "Predicting bounds on queueing delay for batch-scheduled parallel machines," in Proceedings of the 11th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming, pp. 110–118, 2006.

[15] G. Gibson et al. PDSI Data Releases and Repository. http://www.usenix.org/events/fast08/ wips_ posters /gibson-poster.pdf

[16] N Nieuwejaar, D. Kotz, A. Purakayastha, C. S. Ellis, and M. L. Best, "File-Access Characteristics of Parallel Scientific Workloads," IEEE Trans. on Parallel and Distributed Systems, Vol. 7, No. 10, 1996, pp. 1075-1089.