26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing

DuoFS: A Hybrid Storage System Balancing Energy-Efficiency, Reliability, and Performance

Bing Jiao[†‡], Shu Yin[*†‖], Xiaojun Ruan[§], Si Chen[¶], Zhuo Tang[‡], and Xiaomin Zhu[‖],
[†]School of Information Science and Technology, ShanghaiTech University, China, Email: yinshu@shanghaitech.edu.cn
[‡]Hunan University, China
[§]California State University, East Bay, USA
[¶]West Chester University, PA, USA
[‖]National University of Defense Technology, China
[*]Corresponding Author

***Abstract*—As the Energy Wall and the Reliability Wall become unavoidable, it is a demanding and challenging task to reduce energy consumption in large-scale storage systems in modern data centers while retaining acceptable systems reliability. We propose a reliable energy-efficient storage system called DuoFS, which aims at balancing the energy efficiency, the reliability and the performance of parallel storage systems by seamlessly integrating one HDD-based file system and one SSD-based file system. At the heart of the DuoFS is a transformative middleware layer that dispatches files to the one of the two independent parallel file systems based on the files' I/O access popularity. By replicating popular files to the SSD-based file system and pushing the HDD-based file system into the low-power mode under light workload conditions, DuoFS can reduce significant energy consumption, avoid major factors that harm the storage systems reliability, and extract SSDs good I/O performance. Experimental results show that the DuoFS system saves up to 40% of energy, achieves up to 50% better I/O performance while only sacrificing less than 15% of the system's reliability.**

***Index Terms*—hybrid parallel storage system, multiple file systems, PVFS, PLFS, energy- efficient, reliability**

I. Introduction

Traditional energy-saving techniques in disk array systems do not take into account adverse impacts of energy conservation mechanisms on system reliability. Existing solutions tend to frequently spin up and down hard disk drives (HDDs) or replace HDDs with solid state disks (SSDs) [1] [2]. These solutions may cause severe mechanical malfunction of HDDs and lead to permanent data loss. Moreover, heavy write I/O loads shorten the lifetime of SSDs due to the limited numbers of erasure cycles of flash memory. SSDs are more expensive than HDDs; therefore, it is non-cost-effective to build a large-scale storage system using SSDs. In this paper, we propose a hybrid parallel storage system call DuoFS aiming to improve reliability and energy-efficiency of storage systems by introducing a middleware I/O layer so as to apply the hybrid disk technique without modification of existing file systems. At the heart of DuoFS is the hybrid storage technique that exploits flash- based storage devices' low power consumption as well as tradition disk-based storage appliances' large capacity and long lifetime to improve energy efficiency and reliability of parallel storage systems under fluctuating I/O workloads.

The following three factors motivate us to develop the DuoFS system that is reliable and energy efficient.

- the high energy consumption of storage systems in data centers;
- the adverse impacts of existing energy-saving techniques on parallel storage system reliability; and
- the difficulty in modifying the existing parallel file systems.

Motivation 1. Increasing evidence indicates that as much as 27% of the energy in a modern data center is consumed by storage devices [3]. Even worse, such a fraction tends to go up as data storage capacity is dramatically rising by 60% annually [4]. Energy spent to operate disks leads to high heat dissipation, which poses a serious obstacle to the development of energy-efficient cooling systems. Improving energy efficiency of storage systems can substantially reduce operating costs of large-scale data centers; reducing energy consumption of storage systems offers potential economic and environmental benefits [5]

Motivation 2. Existing energy conservation techniques can yield significant energy savings in disks. While several energy conservation schemes like cache-based energy-saving approaches normally have the marginal impact on disk reliability, many energy-saving schemes (e.g., dynamic power management and workload skew techniques) inevitably have noticeable adverse impacts on storage systems [6][7]. For example, dynamic power management (DPM) techniques save energy by using frequent disk spin-downs and spin-ups, which in turn can shorten disk lifetime [8][9]. Furthermore, in order to retain the data consistency, extra effort should be encountered, which leads to performance overhead.

Motivation 3. Existing parallel file systems are very large software projects that must support general use cases and ensure data integrity. In many cases, they support full POSIX compliance, which is not necessary for HPC checkpoint workloads. Relaxation of POSIX semantics or reorganization of I/O workloads within these parallel file systems can negatively impact the entire range of their supported workloads. Additionally, those sophisticated parallel file systems- take PVFS and Lustre as examples- usually have at least 200,000 lines of

0-7695-6380-5/18/$31.00 ©2018 IEEE
DOI 10.1109/PDP2018.2018.00082

code, which set obstacles for modification [10]. Furthermore, any modifications should be carefully addressed based on the knowledge of global picture of the file systems. Otherwise, a simple change to a file system module could sabotage the stability of the whole system. Due to such reasons, it is very difficult to find a large-scale system for the real-world testing since no administrators dare to gamble systems stability on implementing an experimental project. Current studies also taught us that the data management policy of a file system needs to be tailored for both SSDs and HDDs [11]. The configuration will be much more complicated when a file system is mounting a hybrid storage system that consists of SSDs and HDDs.

The benefits of DuoFS are as follows:

- **Better Balance-** With the help of hot/cold data placement policy, DuoFS offers a better balance among energy consumption, reliability, and hardware budget under a fluctuating system workload;
- **A Practical Solution-** DuoFS retains a better system reliability compared to flash-based storage systems and provides a practical solution during the transition period from full-HDD storage to full-NVM storage.
- **Portability-** As DuoFS re-organizes the I/O workload using a middleware layer, it can be implemented without additional kernel modification of existing file systems.
- **Multi-Storage Architecture Support-** DuoFS supports multiple APIs, which makes it possible to mount and manage two different-purpose storage architectures at the same time.
- **Scalability-** Since DuoFS mounts multiple file systems underneath, it makes the storage architecture expandable as the systems go upscaling.
- **Data Consistency-** DuoFS retains data consistency without introducing much overhead as each of the file systems mounted on DuoFS is managed independently.

The remainder of this paper is organized as follows. The overview framework and the design issues of DuoFS are described in Section II. We discuss our prototype and evaluation methodology in Section III. Section IV presents experimental results as well as performance evaluations. Finally, Section VI concludes the paper with future research directions.

II. DuoFS: A Hybrid Energy-Efficient Storage System Framework

A. Framework Overview

The overview framework of DuoFS is expressed in Fig. 1. The DuoFS consists of four major components– a Data Marker, an I/O Re-organizer, a File System Selector, and underlying file systems. The I/O access pattern will be first analyzed to identify the data popularity so that the data will be categorized into two: Hot and Cold data, indicating the frequently accessed data and less frequently accessed data. The popularity information is attached to the data by the Hot/Cold Data Marker. By reading the popularity information, the lower I/O Re-organizer will rearrange the I/O into two groups– hot and cold– accordingly. The File System Selector uses the popularity information to determine the destination file system underlying to retrieve/store the data– hot data on the File System-Hot, which mounts flash-based storage nodes while cold data on the File System-Cold, which mounts hard disk storage nodes. In DuoFS, the underlying File System-Hot acts as a large cache holding most of the frequently accessed data while File System-Cold serves as a large archiving unit that stores less frequently accessed data and can be put into low power mode when I/O workload is not very high. The DuoFS is designed especially to fit the write-once-read-multiple-times scenarios, where frequently read files is replicated to File System-Hot to achieve better performance while the rest of files are stored on File System-Cold. The File System-Cold is pushed to idle mode to saving energy during the light workload.

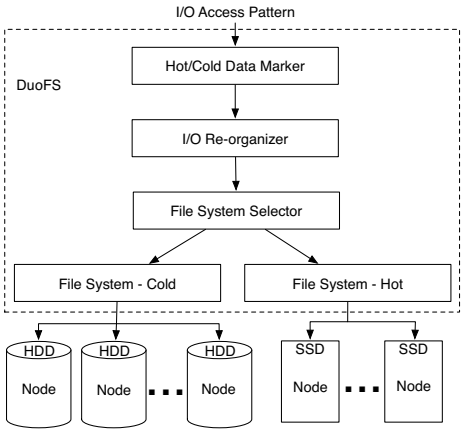


Fig. 1: An Overview of DuoFS Framework

B. Hot/Cold Data Marker

The major purpose of the Hot/Cold Data Marker is to attach the popularity information to the original file so that the lower I/O Re-Organizer and File System Selector can use such information to reorganize and to distribute a file to the target underlying file system. The marker traps the I/O requests and adds an additional tag on the target data. The tag uses one bit for the identification purpose (1 for Hot and 0 for Cold). The popularity of the data, a.k.a the hotness of the data is determined by the LRU mechanism. Once the data hotness is identified from the I/O access pattern that is collected via I/O traces, the Hot/Cold Data Marker module sets the tag bit and passes the I/O requests along with the tag to the lower level module –I/O Re-Organizer. Since the additional tag bit is only used before the file system is selected, such additional tag information will not affect the correctness of the actual data stored on the nodes. In order the manage the tag, a hash table is built to store the paired information as <filename><tag>. The table is generated after the Hot/Cold Data Marker and resides in the memory permanently. Such table is flushed to a separate HDD once the DuoFS is shut down. Furthermore, once the target file system is selected, the I/O requests will

be released by the File System Selector and will be processed via APIs such as POSIX.

C. I/O Re-organizer

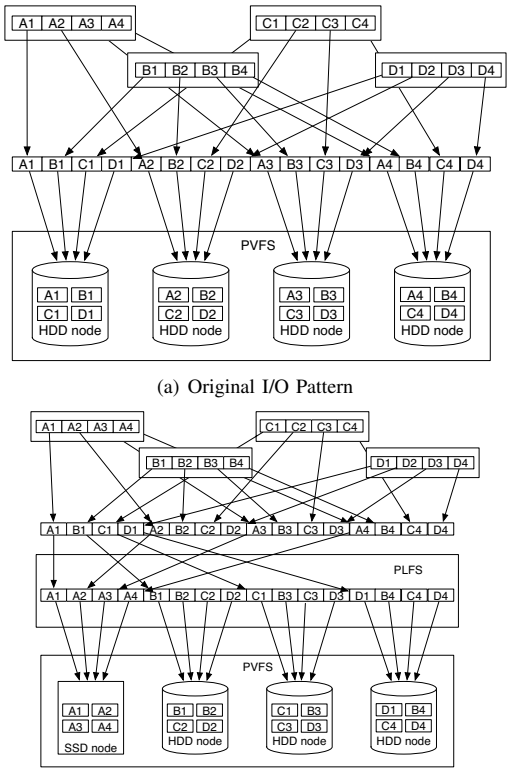
The I/O re-organizer is a middleware layer that is applied on the parallel log-structured file system (PLFS) [12]. Developed by the Los Alamos National Lab, the original design purpose of PLFS is to improve the I/O performance via transformation from N-N pattern to N- 1 pattern while avoiding the serializations. As shown in Fig. 3, by decoupling a file into non-shared component pieces PLFS eliminated serializations on the underlying parallel file system that was responsible for the large performance discrepancy between N- 1 and N-N I/O workloads. In other words, I/O pattern was re-organized by PLFS before it is forwarded to the actual file systems [12]. With the help of a container structure, this reorganization process transforms a virtual serial file into a group of parallel subsets and redirects each subset to the target storage node. Thanks to PLFSs transparency to the actual underlying parallel file system, the real file system processes the I/O request without realizing that the order of the I/O pattern is altered. We utilize the I/O re-organization characteristic of PLFS to redirect blocks or data to the destination node. With the modification of PLFS, the IO Re-organizer determines the target node for all the blocks or data that belongs to the hot data and aggregate the data into, for example, SSD nodes under the light workload. Fig. 2 show a example of original I/O pattern(Fig. 2(a)) and the re-organized I/O pattern(Fig. 2(b)) with the modified PLFS. We can see in Fig. 2(b) that with the small amount of data re-organization, the hot data (File A in the example) will be aggregated in SSD node so that under the light workload, the HDD nodes can be switched to idle mode.

D. File System Selector

The File System Selector chooses the underlying target file systems based on the data popularity tag. Since the PLFS supports multiple backends, once the backend is determined, the underlying file system is selected automatically. The selector first reads the tag, which is passed by the Hot/Cold Marker and maps to the file system backend. According to the design, File System Selector has two options in backends: one is representing File System-Hot, which is mounting flash-based storage nodes; and the other is representing File System-Cold, which is managing HDD storage nodes. If the tag bit is "1", then the first backend is picked, the data will be redirected to the File System-Hot, otherwise, the data is directed to the File System-Cold.

E. Data Transition & Consistency

As the DuoFS is majorly in charge of appointed parallel file system selection and files dispatching, each of the underlying file systems can manage the data layout independently. Furthermore, the SSD based file system only holds the replicas of hot data, the data transition direction between the underlying file systems is only copying hot data from the HDD based file



(a) Original I/O Pattern

(b) I/O Pattern Re-Organization

Fig. 2: Original I/O Pattern vs. Re-Organized I/O Pattern

system to the SSD based file system. When the SSD based file system is filled with the hot data, the LRU mechanism is applied to the data replacement. This operation has two benefits: 1. due to the single direction of the data transition, the I/O overhead can be reduced; and 2. as the underlying file systems manage the data on their own, the data consistency is achieved.

F. Energy-Efficiency

DuoFS leverages flash-based storage nodes, which are highly energy efficient, to cache and buffer data stripes in a skewed fashion. In doing so, DuoFS strives to push File System-Cold and the subset nodes that belong to it into low-power state (e.g. idle) while maintaining the File System-Hot organization. When File System-Cold nodes are switched to the idle state, DuoFS services I/O requests via a combination of blocks residing on power-on File System-Hot nodes. DuoFS identifies blocks that have been frequently accessed (a.k.a., hot data); replicas of hot data blocks are placed and managed into SSD nodes. Hence, in the case where File System-Cold nodes are in idle state, requests accessing hot data can naturally and quickly be served by File System-Hot nodes. Evidence begins to accumulate showing that when the workload is light, requests are likely to access hot data. Although a request might access data that are less popular than hot data, File System-Cold nodes may be temporarily switched to the active state to

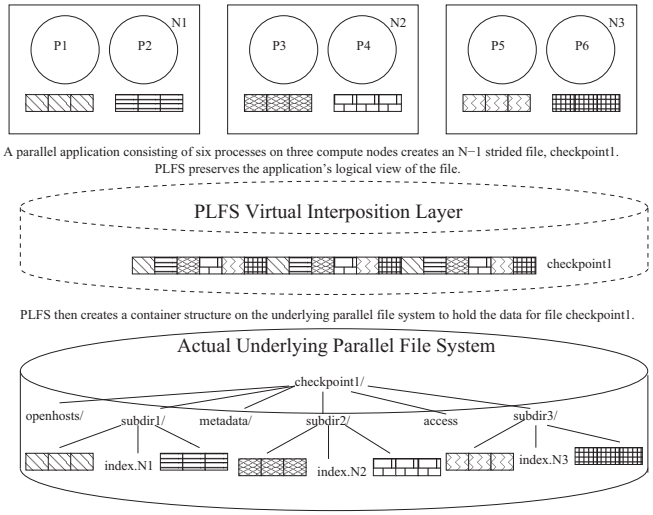


Fig. 3: PLFS Virtual Layer Architecture

provide an I/O service under the light load. Recently accessed cold data will be cached into File System-Hot nodes after File System- Cold nodes are switched back to idle again, thereby reducing unnecessary disk power- state transitions when the cold data is accessed in the not- long-distant future.

By replicating an appropriate amount of data strips to SSD nodes based on system workload, DuoFS aims to aggressively conserve energy consumption by preventing the File System-Cold, which is HDD-based nodes from the active mode. DuoFS heritages our prior work REEDs [13] high energy efficiency thanks to SSDs low power consumption, which is one-third of that of hard drives or HDDs. Additionally, since the DuoFS switches the nodes instead of disks into idle mode, the energy conservation is so significant that the expensiveness of SSDs could be counteracted.

G. Reliablity

To improve the reliability of parallel storage systems, we ensure that DuoFS is capable of tolerating disk failures. Previous studies indicate that life expectancy of server-class disks is likely to be reduced due to power-state transitions [9]. Parallel file systems such as PVFS originally does not replicate data as it is assumed to be running on top of a storage system where RAID-like mechanisms are applied on each node. However, PVFS does not tolerate the inter-node data failure as the organization of PVFS nodes are configured as a RAID-0, only that the storage unit is a node in stead of a disk. This problem is addressed in DuoFS by facilitating PVFS with data redundancy (i.e., placing replicas of popular data into SSD nodes).

One challenge we are facing is how to maintain a high reliability of SSD nodes in DuoFS. SSDs have limited erasure cycles; frequent data updates under the light I/O load may shorten the lifetime of DuoFS when it is in the SSD-HDD hybrid state. Furthermore, an excessive number of power-state transitions sabotages the reliability of hard disk based nodes [14]. For example, suppose the disks transition frequency is 300 per month, the annual failure (AFR) rate is increased by 0.13% [13].

In order to maintain high reliability of SSDs, DuoFS limits the number of erasure cycles of SSDs by disabling the majority writes directly to SSD nodes under heavy I/O load. We mount an additional HDD to the File System-Hot to build a light-weight hybrid storage system. The HDD in File System-Hot will be in charge most of the updates to hot data. The updates will be later flushed to the HDDs based nodes for permanent storage. An enormous number of writes issued under high workloads tends to substantially shorten the lifespan of SSDs. Reducing the number of erasure cycles is achieved by keeping SSDs active when workload conditions are light. Apart from improving the reliability of SSD nodes, DuoFS is capable of maintaining high reliability of HDD nodes. Reliable HDDs become possible in DuoFS because under the light workload, the File System-Cold will be put to low power mode and the number of power-state transitions of HDDs is reduced by SSD nodes under light and fluctuating workloads.

III. Prototype and Evaluation Methodology

A. Prototype Implementation

We develop a proof-of-concept prototype of DuoFS using Parallel Log-structured File System (PLFS) on the parallel virtual file system (PVFS). Since PLFS supports multi-backend, we choose two backends during the PLFS configuration. One of the backends serves as the directory that holds hot data and is mounted to the File System-Hot while the other backend that holds cold data will be mounted to the File System-Cold.

Whether the data is hot or cold can be identified by the tag that is generated via Hot/Cold Data Marker, and such tag is passed to PLFS as an additional parameter at the beginning of calling PLFS. In order to enable PLFS to utilize this tag information, we modified the PLFS source code by adding one new function. This newly added function checks the Hot/Cold data tag before PLFS obtains the backend information. If the tag indicates the data is hot, then the function passes the File System-Hot backend to PLFS, otherwise, the PLFS gets the backend that belongs to the File System-Cold. In our experiments, we focus on the comparisons between DuoFS and original PLFS+PVFS file systems, which achieve high I/O performance by bridging the gap between the logical N-1 workload and the physical N-N workload. The drawback of PLFS-PVFS lies in the lack of a node-level data recovery mechanism.

B. Experimental Setup and Benchmark Characteristics

The prototype DuoFS is implemented with the help of PLFS on two parallel file systems environment, both of which are configured as PVFS. One of the PVFS consist of two storage nodes that consist of one 1TB WD HDD drive and one Samsung 240GB SSD drive each. One of the SSD based nodes also acts as the primary server that has a quad core 3.0

GHz Xeon CPU, 8GB RAM. The other PVFS consist of three storage nodes that have two 1TB WD HDD drives each. The details of the system's characteristics can be found in Table I.

TABLE I: File System and Disk System Parameters

Operating System	Debian MINT 32-bit
File System	OrangeFS 2.4.8 (PVFS)
Node Quantity	5
Node Arrangement	HDD node *3 SSD node *2
Average Power per Node	400W
Hard Disk	
Brand	Western Digital
Capacity	1TB
Quantity of Devices	8
Data Transfer Rate	126MB/s (MAX)
Solid State Disk	
Brand	Samsung
Capacity	240 GB
Quantity of Devices	2
Data Transfer Rate	Read: 540MB/s(MAX) Write: 520MB/s(MAX)

In terms of the I/O performance, we consider the read performance by testing the prototype DuoFS from the following aspects: read open time, IOPS, read bandwidth, and I/O time. We have collected three standard I/O benchmark available to the research community as shown in Table II.

TABLE II: Benchmarks Used in Performance Evaluations

Name	Description
FIO	Flexible IO Tester Synthetic Benchmark
MPIIO Test	LANL Synthetic Checkpoint Benchmark

The first benchmark, FIO, is a versatile IO workload generator that can be used both for benchmark and stress/hardware verification. Another benchmark that we are using is the MPI-IO Test, which is a LANL synthetic checkpoint benchmark tool. The MPI-IO test is built on top of MPI's I/O calls and is used to gather timing information for reading from and writing to file(s) using a variety of I/O profiles; N processes writing to N files, N processes writing to one file, N processes sending data to m processes writing to m files, or n processes sending data to m processes to one file.

In order to consider the data popularity (hot/cold) issue and for the purpose of simplification, we apply the Pareto Principle (a.k.a. the 80/20 rule) to the prototype, which indicates that there will be 80% of I/O requests that are asking for 20% of the entire data (a.k.a. Hot Data) while 20% of requests that are asking for the rest of 80% of the data (a.k.a Cold Data) [15]. Such data popularity module will be replaced by a dynamic hot/cold data identification module when our ongoing project is completed.

Furthermore, as one of the major goals of DuoFS is to achieve storage systems energy conservation while retaining their reliability, we build a simple method to determine the energy consumption as follows (in Eq. 1):

$$E = P * N_{active} * T_{I/O} \tag{1}$$

where E represents the energy consumption of a storage system when dealing with certain I/O requests, P is the average power of a single storage node, N_{active} indicates the number of active storage nodes that are involved in the I/O requests, and $T_{I/O}$ is the total I/O time lapse from the first I/O request starts to the completion of the last I/O request.

When it comes to the reliability analysis, we are using the Weibull-based evolution methods that are discussed in MREED [13]. The system reliability can be expressed via Eq. 2:

$$R = R_{util} * \tau + \alpha * R_{freq} \tag{2}$$

where R_{util} is the baseline failure rate derived from disk utilization, which is examined by the Weibull distribution. R_{freq} represents an adder to the base annual failure rate [13].

IV. Experimental Results

Figs. 4–10 illustrate the general I/O performance of the DuoFS using aforementioned benchmarks in addition to the energy consumption evaluations. Noted that the "SSD-Only" indicates the situation that only SSD-based nodes are involved while the HDD-based nodes are managed to the low-power mode by the FileSystem-Cold. The purpose of presenting the "SSD-Only" is to provide a full flash-based storage case. We can see that the full flash-based storage system out-stands most of the performance tests and the energy consumption evaluations. However, our argument is that the hybrid storage mechanisms such as DuoFS balance the price-performance ratio better due to the significant price difference between SSDs and HDDs.

A. I/O Benchmark Performance Analysis

FIO Tests From Figs. 4–5, we can see that the DuoFS beats the traditional HDD-based PLFS file systems for most of the time. Since most of the frequently accessed data is distributed to SSD-based nodes, the DuoFS can benefit significantly from the I/O performance advantages of SSDs. Noticed that as the number of concurrent processes is beyond four, DuoFS behaves around 10% worse than the SSD-only case. The inefficiency of the inner file system determination process is the main restraint that DuoFS prototype cannot handle as much simultaneous I/Os as the SSD-only does.

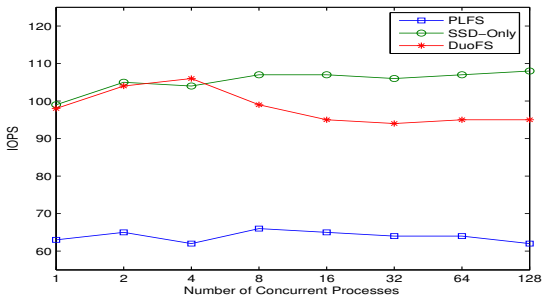


Fig. 4: IOPS Comparison under FIO Testing

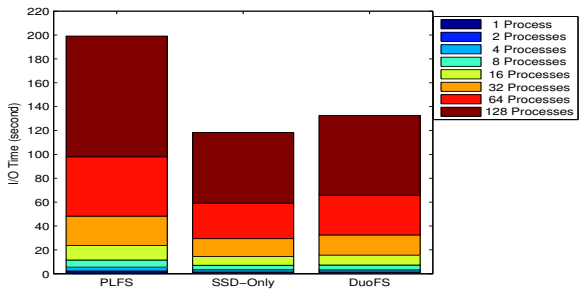


Fig. 5: I/O Time Comparison under FIO Testing

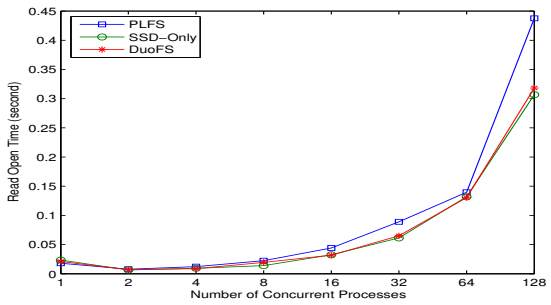


Fig. 8: Read Open Time Comparison under MPI-IO Testing DuoFS

MPI-IO Tests Figs. 6–8 show the test results in the I/O read bandwidth, the total I/O time, and the read open time via the MPI-IO test tools. We can observe that DuoFS works well when the number of concurrent processes is not very large (less than 64). The major reason is also that DuoFS has a lower parallelism level compare to the traditional PLFS setup with the total same number of nodes. It is also noticed that as the number of concurrent processes increased, the read open time of DuoFS is slightly quicker than that of the original HDD based PLFS thanks to the better read performance of flash based devices.

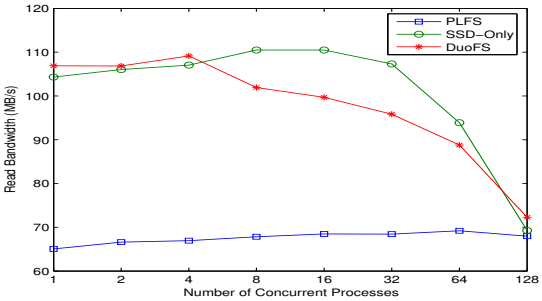


Fig. 6: Read Bandwidth Comparison under MPI-IO Testing

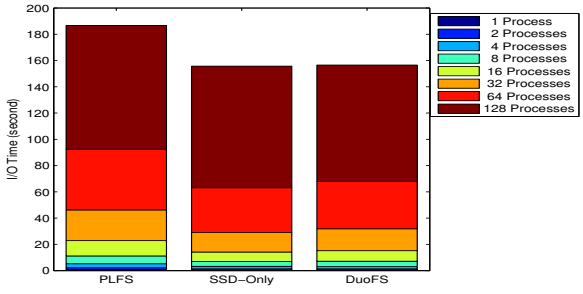


Fig. 7: I/O Time Comparison under MPI-IO Testing

B. *Energy Consumption*

Figs. 9 through 10 plot the energy consumption of PLFS and DuoFS under three different benchmarks – FIO and MPI-IO test. We can observe that when the number of processes is smaller than 32, the DuoFS can save more than 60% of the energy compared to the traditional PLFS. This is majorly because the number of active storage nodes is much less than that of PLFS during the entire testing time period. In addition, due to the good read bandwidth of SSDs, the DuoFS can provide most of the I/O requests with shorter time lapse, which further improves the energy conservation.

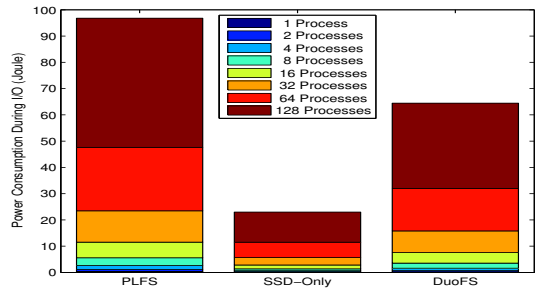


Fig. 9: Energy Consumption Comparison under FIO Testing

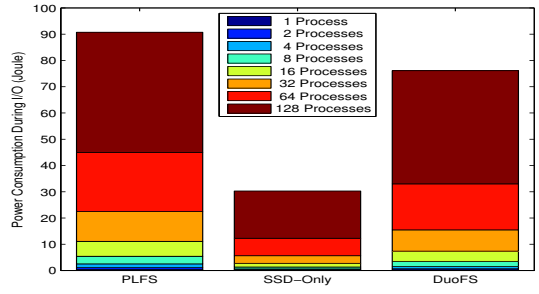


Fig. 10: Energy Consumption Comparison under MPI-IO Testing

C. Reliability

We apply MREED [23] to quantify the system reliability of our proposed DuoFS. It is noteworthy that MREED is a reliability modeling framework for parallel disk systems coupled with energy conservation techniques. One critical module in MREED is to model the impact of energy-efficient schemes on the utilization and power-state transition frequency of each disk in a parallel disk system. A second important module of MREED is to calculate the annual failure rate of each disk as a function of the disks utilization and power-state transition frequency. Given the annual failure rate of each disk in a parallel disk system, MREED derives the reliability of the system. Fig. 11 illustrates the annual failure rates (AFR) of PLFS and DuoFS during the entire testing time. PLFS's AFR behavior is better than the DuoFS primarily due to the lack of energy-saving mechanisms. However, compared to the energy consumption results in the aforementioned section, the DuoFS only trades nearly 5% of the reliability for more than 60% energy conservation.

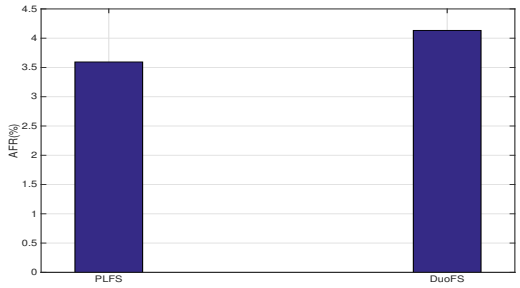


Fig. 11: Reliability Comparison between PLFS and DuoFS

V. Related Works

Solid State Storage: Solid State Drives or SSDs have emerged in the last few years as a viable replacement for hard drives in a wide variety of application settings. Commodity SSDs offer promising random read and write performance, potentially eliminating I/O bottlenecks in high-performance data centers while driving down energy consumption [16]. OCZ reports that SSDs speed up the performance of conventional hard drives (HDDs) by a factor of more than 100 [17]. More importantly, the power efficiency of SSDs is significantly higher than that of HDDs; for example, the power consumption of an SSD and an HDD at a peak load are 2W and 6W, respectively. Unfortunately, SSDs have severe disadvantages that prevent them from being widely applied in modern data centers. These drawbacks include low endurance limits and high cost. Very recently, a study demonstrates that the erasure limit of MLC devices is typically ranging anywhere from 5,000 to 10,000 cycles per block [18]. Although the SSDs cost per GB sharply drops down to 80¢, the cost-efficiency of SSDs is nowhere near that of HDDs, which cost as low as 5¢/GB [19][20].

Hybrid Storage: Due to the performance benefits, SSDs are commonly implemented as cache pools for HDDs in parallel file systems, such as iBridge [1] and its predecessor iTransformer [21]. Hybrid Storage with SSDs is another popular technique to exploit their merits, e.g. iCash [22], HM [2] and HAS [23]. Although effective, the vast majority of researches focus on extracting performance benefits of SSDs without the consideration the reliability issues of the flash devices.

Energy Efficiency vs. Reliability: Evidence on the reliability of RAID systems began to accumulate indicating that existing energy conservation techniques are inadequate for RAID systems due to the following three reasons [24]. First, no opportunity is offered to spin down any disk in a conventional RAID system due to the I/O load balancing across all the disks for maximized disk parallelism and performance. Second, hard disks were not designed for frequent power cycles, which significantly reduce HDD life expectancy. Third, energy-efficient caching and dynamic power management are inapplicable for RAIDs deployed in server systems, because the servers are too busy to have any long idle time period.

A reliability analysis (see, for example, [9]) on energy-efficient RAID systems suggests that an energy-saving mechanism greatly affects the healthiness of parallel disks, and the reason is two-fold: (1) the power-state-transition frequency caused by spinning up and down disks may lead to mechanical malfunctions in disks [25]; (2) flash-based disks are likely to exceed the erasure limits and have a high risk of breaking down with high frequent updates.

Middle-ware Layer: There are several transformative I/O middleware layers currently developed for HPC environments. Reaching exascale I/O performance is likely to rely on these middleware layers capable of managing parallel I/O workloads. Work has been conducted on matching the user view of parallel I/O to optimize workloads on a parallel file system, but PLFS takes this further and attempts to mask I/O workload and system configuration parameters from users [12]. Similar to PLFS project, the Adaptable I/O System (ADIOS) from Oak Ridge National Laboratory is an I/O library and API for scientific codes that efficiently groups scientific array data and is capable of writing the data in a log-structured format [26]. The Distributed Application Object Storage (DAOS) from Sandia National Laboratory serves as the persistent storage interface and translation layer between the user-visible object model and the requirements of the underlying storage infrastructure [27].

VI. Conclusion

In this paper, we discussed a parallel storage system framework called DuoFS, which aims at balancing the issue of energy-efficiency and reliability. DuoFS presents a hybrid idea of applying two separate file systems (File System-Hot and File System- Cold) that mount different types of storage nodes. File System-Hot mounts SSD nodes to achieve better I/O performance and active power consumption where File System-Cold mounts HDD nodes to gain larger capacity and longer lifetime. DuoFS distributes hot data to File System-Hot and pushes File System-Cold, which holds cold data, to idle mode

under the light workload to achieve energy conservation. More importantly, we implemented a DuoFS prototype on a real file system– PVFS – with the help of a transformative layer called PLFS and make the large-scale real world testing possible. Our ongoing work on this project is performed in the following few aspects. First of all, a hot/cold data identifier should be carefully designed and implemented. The current study only assumes that the hot/cold data has already been defined based on the analysis of existing traces. An efficient hot/cold data identifier could enhance the accuracy of data placement, hence further reduce power management overhead. Second, the power management policy should be more sophisticated. The present policy only groups the storage nodes into two: active nodes for hot data and idle nodes for cold data. The improved power management algorithm should address the placement of warm data more carefully. Thirdly, we intend to address the write issue carefully, to cover more I/O scenarios such as fault tolerance using checkpointing, which generate a lot of writes periodically but only issues reads under failure. Last but not the least, the data de-duplication case is also needed to be studied to further improve the systems energy conservation.

ACKNOWLEDGMENTS

We are indebted to Dr. Xiao Qin for his invaluable feedback and suggestions that greatly improved this work. Shu Yin's research is supported by the National Natural Science Foundation of China under Grant 61402158, the China Postdoctoral Science Foundation under Grant 2015M572708, and ShanghaiTech University under a start-up grant. Xiaomin Zhu's work is supported by the National Natural Science Foundation of China under Grant 61572511, and the Scientific Research Project of National University of Defense Technology under Grant ZL16-03-09.

REFERENCES

[1] X. Zhang, K. Liu, K. Davis, and S. Jiang, "ibridge: Improving unaligned parallel file access with solid-state drives," in *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, May 2013, pp. 381–392.

[2] S. Wu, G. Chen, K. Chen, F. Li, and L. Shou, "Hm: A column-oriented mapreduce system on hybrid storage," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 12, pp. 3304–3317, Dec 2015.

[3] B. Battles, C. Belleville, S. Grabau, and J. Maurier, "Reducing data center power consumption through efficient storage," *Netapp white paper*, February 2007.

[4] E. Pinheiro, R. Bianchini, and C. Dubnicki, "Exploiting redundancy to conserve energy in storage systems," *SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 1, pp. 15–26, Jun. 2006.

[5] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A survey of green networking research," *Communications Surveys Tutorials, IEEE*, vol. 14, no. 1, pp. 3–20, 2012.

[6] K. Bellam, A. Manzanares, X. Ruan, X. Qin, and Y.-M. Yang, "Improving reliability and energy efficiency of disk systems via utilization control," in *Proc. IEEE Symp. Computers and Comm.*, 2008.

[7] T. Xie and Y. Sun, "Sacrificing reliability for energy saving: Is it worthwhile for disk arrays?" in *Proc. IEEE Symp. Parallel and Distributed Processing*, April 2008, pp. 1–12.

[8] S. Yin, X. Ruan, A. Manzanares, and X. Qin, "How reliable are parallel disk systems when energy-saving schemes are involved?" in *Proc. IEEE International Conference on Cluster Computing (CLUSTER)*, 2009.

[9] S. Yin, Y. Tian, J. Xie, X. Qin, M. Alghamdi, X. Ruan, and M. Qiu, "Reliability analysis of an energy-aware raid system," in *Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International*, 2011, pp. 1–8.

[10] A. Manzanares, J. Bent, M. Wingate, and G. Gibson, "The power and challenges of transformative i/o," in *Proceedings of the 2012 IEEE International Conference on Cluster Computing*, ser. CLUSTER '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 144–154. [Online]. Available: http://dx.doi.org/10.1109/CLUSTER.2012.86

[11] M. Maas, T. Harris, K. Asanovic, and J. Kubiatowicz, "Trash day: Coordinating garbage collection in distributed systems," in *Proceedings of the 15th USENIX Conference on Hot Topics in Operating Systems*, ser. HOTOS'15. Berkeley, CA, USA: USENIX Association, 2015, pp. 1–1. [Online]. Available: http://dl.acm.org/citation.cfm?id=2831090.2831091

[12] J. Bent, G. Gibson, G. Grider, B. McClelland, P. Nowoczynski, J. Nunez, M. Polte, and M. Wingate, "Plfs: A checkpoint filesystem for parallel applications," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, ser. SC '09. New York, NY, USA: ACM, 2009, pp. 21:1–21:12. [Online]. Available: http://doi.acm.org/10.1145/1654059.1654081

[13] S. Yin, X. Li, K. Li, J. Huang, X. Ruan, X. Zhu, W. Cao, and X. Qin, "Reed: A reliable energy-efficient raid," in *2015 44th International Conference on Parallel Processing*, Sept 2015, pp. 649–658.

[14] B. Jiao, X. Zhu, X. Ruan, X. Qin, and S. Yin, "Duofs: An attempt at energy-saving and retaining reliability of storage systems," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, June 2017, pp. 2634–2635.

[15] G. Yadgar and M. Gabel, "Avoiding the streetlight effect: I/o workload analysis with ssds in mind," in *8th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 16)*. Denver, CO: USENIX Association, 2016. [Online]. Available: https://www.usenix.org/conference/hotstorage16/workshop-program/presentation/yadgar

[16] M. Balakrishnan, A. Kadav, V. Prabhakaran, and D. Malkhi, "Differential RAID," *ACM Transactions on Storage*, no. 2, pp. 1–22, Jul. 2010.

[17] OCZ, "Ocz vector sepcifications," http://www.ocz.com/consumer/vector-7mm-sata-3-ssd/specifications.

[18] G. Wu and X. He, "Delta-ftl: improving ssd lifetime via exploiting content locality," in *Proceedings of the 7th ACM european conference on Computer Systems*, ser. EuroSys '12, 2012, pp. 253–266.

[19] A. Richi, "Ssd price crash:80 ¢per gb!" http://h30565.www3.hp.com/t5/Mobility-Matters/SSD-price-crash-80-per-GB/ba-p/5032, February 2012.

[20] T. Iyer, "Hdds return to pre-flood prices," http://www.tomshardware.com/news/Thailand-Flood-Storage-Price-SSD,22150.html, April 2013.

[21] X. Zhang, K. Davis, and S. Jiang, "itransformer: Using ssd to improve disk scheduling for high-performance i/o," in *2012 IEEE 26th International Parallel and Distributed Processing Symposium*, May 2012, pp. 715–726.

[22] Q. Yang and J. Ren, "I-cash: Intelligently coupled array of ssd and hdd," in *2011 IEEE 17th International Symposium on High Performance Computer Architecture*, Feb 2011, pp. 278–289.

[23] S. He, X. H. Sun, and A. Haider, "Has: Heterogeneity-aware selective data layout scheme for parallel file systems on hybrid servers," in *2015 IEEE International Parallel and Distributed Processing Symposium*, May 2015, pp. 613–622.

[24] C. Weddle, M. Oldham, J. Qian, A.-I. A. Wang, P. Reiher, and G. Kuenning, "Paraid: a gear-shifting power-aware raid," in *FAST '07: Proceedings of the 5th USENIX conference on File and Storage Technologies*, Berkeley, CA, USA, 2007, pp. 30–30.

[25] S. Yin, X. Ruan, A. Manzanares, X. Qin, and K. Li, "Mint: A reliability modeling framework for energy-efficient parallel disk systems," *Dependable and Secure Computing, IEEE Transactions on*, vol. 11, no. 4, pp. 345–360, July 2014.

[26] J. F. Lofstead, S. Klasky, K. Schwan, N. Podhorszki, and C. Jin, "Flexible io and integration for scientific codes through the adaptable io system (adios)," in *Proceedings of the 6th International Workshop on Challenges of Large Applications in Distributed Environments*, ser. CLADE '08. New York, NY, USA: ACM, 2008, pp. 15–24. [Online]. Available: http://doi.acm.org/10.1145/1383529.1383533

[27] J. Lofstead, I. Jimenez, C. Maltzahn, Q. Koziol, J. Bent, and E. Barton, "Daos and friends: A proposal for an exascale storage system," in *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov 2016, pp. 585–596.