Rotated Logging Storage Architectures for Data Centers: Models and Optimizations

Yinliang Yue, Bingsheng He, Lei Tian, Hong Jiang, Fang Wang, and Dan Feng

Abstract—We propose *Ro*tated *Lo*gging (RoLo), a new logging architecture for parallel disk-based mirrored storage systems for enhanced energy efficiency, which is one of the key concerns in modern data centers. By spreading destaging I/O activities among short idle time slots and proactively reclaiming the stale logging space, RoLo rotates loggers among a logical logging space pool formed collectively from the free storage space available among mirrored disks. We develop three flavors of RoLo, that is, RoLo-P/R/E, to emphasize performance, reliability, and energy efficiency respectively. Without the extra dedicated log disks and the corresponding centralized destaging, RoLo eliminates the additional hardware and energy costs, potential single point of failure and performance bottleneck. Furthermore, RoLo-P/R/E, applied to specific scenes correctly, can prolong the lifecycle of the disks and improve the system's energy efficiency by reducing the disk spin up/down frequency. We propose RoLo-S to further alleviate the performance bottleneck and energy consumption caused by frequent disk head seeks in on-duty logger disks. We have implemented RoLo and RoLo-S on real disk systems. Extensive trace-driven evaluations demonstrate the advantages of the three RoLo schemes over both a RAID10 system with centralized logging architecture and a typical RAID10 system, and the advantages of RoLo-S over RoLo.

Index Terms—Logging architecture, destaging mechanism, reliability, performance evaluation, energy efficiency

1 Introduction

Recent studies report that energy costs may increase from 10 percent of the Total Cost of Ownership (TCO) of data centers to over 50 percent in the next few years [8]. Storage subsystems are a major contributor to the energy consumption of data centers. For a typical data center, the disk-based storage subsystem can consume 27 percent of the total energy and this fraction tends to increase rapidly as storage requirements rise by 60 percent annually [27]. Logging architectures are one of the key system components in data centers [15], [16], [20], [24]. Therefore, this paper investigates whether and how we can improve the energy efficiency, performance and reliability of logging architectures.

Hierarchical storage architectures are widely adopted in the modern storage systems to judiciously cache/buffer some data blocks for enhanced I/O performance or energy efficiency [4], [6], [11], [15], [28]. Write requests tend to dominate disk I/O activities, since most read requests are absorbed by multi-level storage caches, while modified data blocks must be written to their targeted disks eventually to ensure data integrity. Logging write requests is one of the main approaches to improving energy efficiency and performance of storage subsystems [4], [6], [11], [15], [16], [20], [24]. Temporarily redirecting write requests to log disks enables the write-targeted disks to switch to and remain in the low-power state for a longer period of time to save energy. Since frequently spinning up/down disks will negatively impact the disk lifecycle and energy efficiency, only when the capacity occupancy of the log disks reaches the predefined threshold can the inconsistent data blocks on write-targeted disks be updated, a process known as *destaging*. We refer to the period during which write requests are redirected to log disks as *logging period*, and the period during which the inconsistent data blocks on write-targeted disks are updated as *destaging period*. There are several inherent shortcomings associated with the conventional centralized logging architecture. On the one hand, the centralized destaging, which incurs bursty I/O activities, prevents the system's energy efficiency from being further improved by simply increasing the available logging space (see Section 2). On the other hand, the extra dedicated log disks not only incur additional hardware and energy costs, but also become a potential performance bottleneck and single point of failure. Unfortunately, to the best of our knowledge, none of the existing logging schemes has overcome these shortcomings, especially for the RAID10 storage architecture that is widely deployed in modern data centers. Recent studies indicate that mirrored disk arrays are advantageous over parity based disk arrays, such as RAID5 and RAID6, in that the former avoids data loss caused by parity pollution and parity inconsistencies [35] that are inevitable in the latter. Besides, parity based disk arrays incur the small write penalty.

In this paper, we propose Rotated Logging (RoLo), a new logging architecture for parallel disk-based mirrored storage systems for enhanced energy efficiency, which is one of

- *Y. Yue is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100049, China. E-mail: yueyinliang@iie.ac.cn.*
- *B. He is with the Nanyang Technological University, Singapore 639798. E-mail: bshe@ntu.edu.sg.*
- *L. Tian and H. Jiang are with the Department of Computer Science and Engineering, University of Nebraska Lincoln, Lincoln, NE 68588. E-mail: {tian, jiang}@cse.unl.edu.*
- *F. Wang and D. Feng are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China. E-mail: {wangfang, dfeng}@hust.edu.cn.*

Manuscript received 23 Apr. 2014; revised 3 Mar. 2015; accepted 17 Mar. 2015. Date of publication 26 Mar. 2015; date of current version 16 Dec. 2015. Recommended for acceptance by J. D. Bruguera.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2015.2417539

the key concerns in modern data centers. RoLo organizes the available free space of all mirrored disks into a logical logging space pool, and employs decentralized destaging to spread and dilute destaging I/O activities among short idle time slots during each logging period, thus proactively and non-intrusively reclaiming the stale logging space. With this simultaneous consumption and reclamation of the logger capacity, logging can be continuously rotated among the logical logging space pool that becomes a virtually unlimited logging capacity. More specifically, three flavors of RoLo are developed in this study, namely, a performance-oriented RoLo (RoLo-P), a reliability-oriented RoLo (RoLo-R) and an energy-oriented RoLo (RoLo-E), to suit the needs of different application environments.

By circularly reusing logical logging space of the existing mirrored disks instead of extra dedicated log disks, RoLo eliminates the additional hardware and energy costs, potential single point of failure and performance bottleneck of the conventional centralized logging architecture. Furthermore and most importantly, RoLo-P/R/E, applied to specific scenes correctly, can prolong the lifecycle of write-targeted disks and improve the system's energy efficiency by reducing the disk spin up/down frequency. Note that RoLo-E differs significantly from RoLo-P/R, and can only be used for write dominated workloads.

Since the mixed interleaved logging I/Os and destaging I/Os in on-duty log disk may cause frequent long-distance disk head seeks and then incur lower performance and worse energy efficiency, we propose RoLo-S to spread logging storage space among effective data regions in a *zebra-crossing* style. By locating logging I/Os with destaging I/Os as near as possible, disk head seeks are reduced significantly and both performance and energy efficiency are further improved. Note that RoLo-S is orthogonal to RoLo-P/R, and RoLo-S's zebra-crossing data layout can be used to further improve the energy efficiency and performance of RoLo-P/R.

Both RoLo and RoLo-S have been implemented and evaluated on real disk systems. Extensive trace-driven evaluations on real disk systems demonstrate that RoLo-P/R saves up to 13.2 percent power than GRAID, meanwhile RoLo-S outperforms RoLo in average response time and energy consumption by 48.9 and 15.7 percent respectively.

Note that RoLo is a general logging model mainly to improve the energy efficiency of storage systems. We conjecture that RoLo is not only suitable for RAID systems, but also for general redundancy based storage models.

The rest of this paper is organized as follows. Background and motivation are presented in Section 2 and the design of RoLo are presented in Section 3. We model and analyze the performance, energy consumption and reliability of RoLo in Section 4. Performance evaluations are presented in Section 5. We present related work in Section 6 and conclude the paper in Section 7.

2 Background and Motivation

2.1 Conventional Logging Architecture

In the mirrored RAID10 systems, immediate in-place updating of data blocks in all the primary disks and mirrored disks prevents any disk from being spun down to save energy. The conventional approach of energy saving is that one extra disk is introduced into standard RAID10 system and used as the dedicated log disk. All the write requests among write-targeted disks are redirected onto the extra dedicated log disk. Thus, all the old data blocks in mirrored disks will be updated, only until the subsequent destaging from primary disk to mirrored disk.

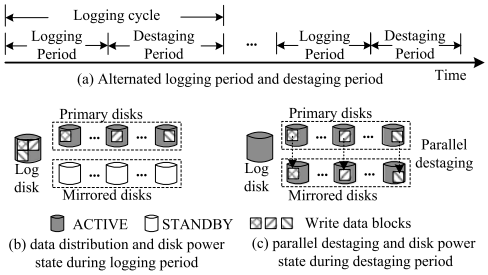


Fig. 1. Working principles of centralized logging and destaging in RAID10 storage architecture.

In such an architecture, shown in Fig. 1a, each *logging cycle* is composed of a *logging period* and its corresponding *destaging period*. During the logging period, all the mirrored disks are set to the STANDBY state, and the two copies of each write data are written to the corresponding location in primary disks and sequential location in the log disk respectively (see Fig. 1b) . The destaging process is triggered once the occupancy level of the log disk reaches a predefined threshold value (e.g., 80 percent). During the destaging period, all the mirrored disks are set to the ACTIVE state and *all the inconsistent data blocks in mirrored disks are updated in parallel from the corresponding primary disks instead of log disk(s)* (see Fig. 1c) for better destaging performance.

To identify the performance and energy bottleneck of the conventional logging architecture, we define *destaging interval ratio* and *destaging energy ratio* to be the fractions respectively of time spent and energy consumed by destaging during each logging cycle. We conducted extensive experiments to study the centralized logging architecture and obtained the following important observations. We present the key results, and more details can be found at our previous work [29], [30].

Observation. Simply increasing the logging space alone will not decrease destaging interval ratio and destaging energy ratio.

Reasons. The increased amount of logged data in a larger logger will prolong both the logging period and destaging period proportionally and thus increase their corresponding energy consumption simultaneously.

Implications. Simply increasing the available logging space is not a viable power saving solution and this inability to improve energy efficiency by increasing logging space stems from the centralized destaging strategy of the conventional logging architecture.

2.2 Free Time and Space Resources

There are two types of "free" resources that have been exploited in storage subsystems of typical data centers: *unused storage space* and *idle time slots*. In Symantec's 2008

State of the Data Center survey [23], it was found that data centers typically utilize 50 percent of their storage capacity. Mark Levin [14] points out that on average the disk storage utilization is 56.6 and 46.4 percent for UNIX and Windows environments respectively under locally attached storage deployment, and this proportion increases to 75.5 and 55.8 percent under SAN deployment. Short idle time slots are abundant for both primary disks and log disk(s) during logging periods under light workloads as indicated in our previous study [29] and other studies [17]. Most idle time slots are much shorter than the break-even time for modern disks to spin down to save power [17].

While either idle time slots or free storage space has been exploited to improve performance [12], reliability [17], or energy efficiency [26] of storage systems, these resources remain to be effectively and fully tapped to optimize the performance and energy efficiency of storage systems with a logging architecture.

2.3 Characteristics of Energy Consumption and Performance of Hard Disks

Hard disk is composed of lots of tracks. Outer tracks correspond to low logical block numbers (LBNs), and inner tracks correspond to high LBNs. One disk I/O operation is comprised of two stages, that are *disk head seek* and *disk read/write*. Both of these two stages have significant influence on the disk energy consumption and I/O performance.

Essary et al. [32] explored the relationship between the first stage, i.e., disk head seek, with the disk energy consumption and I/O performance. The disk power consumption can be expressed as $power = a \times \log(perc + b) + c$, where a, b, and c are constants and $perc$ represents the percentage of the disk traversed, a bounded quantity ranging from 0 to 100. Besides, relatively long disk seek time dominates the I/O response time.

Hylick et al. [31] explored the relationship between the second stage, i.e., disk read/write, with the disk energy consumption and I/O performance. The extensive evaluation results of measuring about ten product level hard drivers get the following conclusions. Reads on inner tracks lead to smaller bandwidth but much more energy consumption than those on outer tracks. In contrast, the bandwidth and energy consumption of writes are irrelevant to track positions.

The above explored relationship between disk energy consumption and I/O performance with both disk head seek and disk read/write motivate us to propose RoLo-S to further improve performance and energy efficiency from RoLo.

2.4 Motivation

2.4.1 Motivation for RoLo

Given the important observations above and the availability of the aforementioned free resources, we are motivated to propose a new rotated logging architecture, RoLo. It integrates the unused free space of redundant mirrored disks in a RAID10 system into a large logical logging space pool, which is circularly recyclable by exploiting the short idle time slots for decentralized destaging to improve both performance and energy efficiency of the system. In other words, one or a few mirrored disks take turns to serve as on-duty log disks, while off-duty mirrored disks can be spun down to save energy. The basic model of RoLo is shown in Fig. 2. At the same time, idle time slots are exploited to spread and dilute the bursty destaging I/O activities. With this decentralized destaging strategy, the stale logging space can be proactively and non-intrusively reclaimed, thus enabling logging to be unlimitedly rotated among the logical logging space pool.

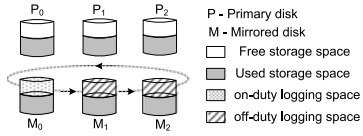


Fig. 2. The basic model of RoLo.

2.4.2 Motivation for RoLo-S

The destaging process reads data from primary disks and then writes them to mirrored disks. The primary disks are with read and write mixed I/Os, meanwhile the mirrored disks are with solely write I/Os, including logging and destaging writes. As revealed by Hylick et al. [31] and discussed in Section 2.3, read energy efficiency and performance with low LBN (i.e., outer tracks) are better than that with high LBN (i.e., inner tracks), and write energy efficiency and performance are not sensitive to the LBN, we are motivated to propose RoLo-S.

In RoLo-S, we set the regions with lower LBN as data region for primary disks, because restricting destaging read I/Os in lower LBN can achieve higher read bandwidth and less energy consumption. Meanwhile, in order to gather write I/Os to reduce the disk head seek distance, we spread logging regions among effective data regions in a *zebra-crossing* style for mirrored logger disks. Although data region is stretched and the write I/O footprints span the entire mirrored disks, the stretched data write I/Os footprints does not decrease the energy efficiency and performance due to the fact that write energy efficiency and performance are not sensitive to the LBN [31], i.e., track positions.

3 RoLo Architecture and Design

3.1 The Basic RoLo Idea

3.1.1 Dynamic Rotated Logging

Any new data is written to two disks in order to prevent data loss, with one copy written to the primary disk as in the standard RAID10 fashion immediately while the second copy is sequentially written to the on-duty logging space. When the available free storage space of the on-duty log disk decreases to a predefined threshold with the logged data, the on-duty logger is rotated to the next mirrored disk, and this will end the current logging period and start a new one, as shown in Fig. 3a. The logger rotation triggers the disk state to change, as shown in Figs. 3b, 3c, and 3d.

For instance, as shown in Fig. 3b, M_0 is used as the on-duty log disk in logging period T_0, and thus M_0 is kept active state. The second copies of newly written data in T_0, i.e., $\mathrm{D}_0\mathrm{T}_0$, $\mathrm{D}_1\mathrm{T}_0$ and $\mathrm{D}_2\mathrm{T}_0$, are logged to the free space of M_0. Besides, M_1 and M_2 are spun down from active state to standby state.

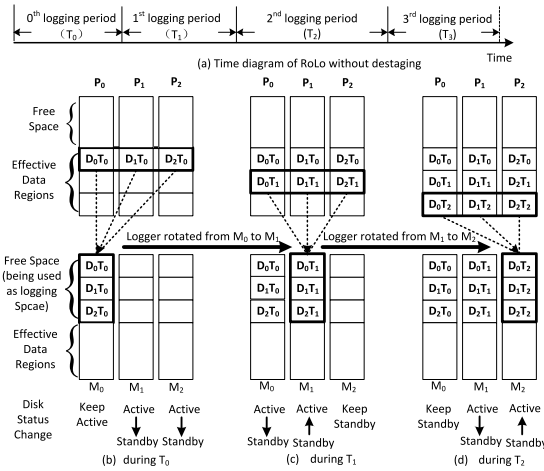


Fig. 3. The dynamic process of logger rotation without destaging. $\mathrm{P_i}$ presents the ith primary disk, and $\mathrm{M_i}$ represents the $\mathrm{P_i}$'s mirrored disk. The $\mathrm{D_mT_n}$ represents the newly written data for the mth mirrored disk pair $(\mathrm{P_m}, \mathrm{M_m})$ during the nth logging period $\mathrm{T_n}$.

Similarly, $\mathrm{M_1}$ is spun up from standby state to active state since $\mathrm{M_1}$ is used as the on-duty logger during $\mathrm{T_1}$. Besides, $\mathrm{M_0}$ is spun down from active state to standby state and $\mathrm{M_2}$ is kept standby state. As shown in Fig. 3c, when entering logging period $\mathrm{T_1}$, $\mathrm{D_0T_1}$, $\mathrm{D_1T_1}$ and $\mathrm{D_2T_1}$ are written to $\mathrm{M_1}$, and $\mathrm{D_0T_2}$, $\mathrm{D_1T_2}$ and $\mathrm{D_2T_2}$ are written to $\mathrm{M_2}$ during $\mathrm{T_2}$.

3.1.2 Decentralized Destaging

Each logger rotation triggers a new destaging process for the newly on-duty log disk to update the inconsistent data blocks from its corresponding primary disk by spreading destaging I/O activities among the short idle time slots, as shown in Fig. 4a. Spreading destaging I/O activities among idle time slots during logging periods is the basic idea of decentralized destaging. A new destaging process is triggered only when the logger rotates to a new log disk and it will not finish until all the inconsistent data blocks in the mirrored disks have been updated. For example, the destaging process for $(\mathrm{P_1}, \mathrm{M_1})$ is triggered immediately when the logger rotates to $\mathrm{M_1}$. The only responsibility of this destaging process is to update all the inconsistent data blocks from $\mathrm{P_1}$ to $\mathrm{M_1}$. As shown in Fig. 4c, only when all the inconsistent data blocks in $\mathrm{M_1}$ have been updated from $\mathrm{D_1T_0}$ in $\mathrm{P_1}$, can this destaging process be terminated. Note that data block $\mathrm{D_1T_0}$ in $\mathrm{M_0}$ becomes invalid and the corresponding stale logging space it occupies is reclaimed when the destaging process for $(\mathrm{P_1}, \mathrm{M_1})$ finishes. Similarly, the destaging process for disk pair $(\mathrm{P_2}, \mathrm{M_2})$ is triggered immediately after $\mathrm{M_2}$ is selected as the on-duty log disk. Data blocks $\mathrm{D_2T_0}$ and $\mathrm{D_2T_1}$ are updated from $\mathrm{P_2}$ to $\mathrm{M_2}$, and the logging space occupied by $\mathrm{D_2T_0}$ and $\mathrm{D_2T_1}$ in $\mathrm{M_0}$ and $\mathrm{M_1}$ are reclaimed after the destaging process for $(\mathrm{P_2}, \mathrm{M_2})$ is completed. The solid lines with arrows and rectangles with twills in Figs. 4c and 4d show the decentralized destaging and proactive reclaiming mechanism respectively. Since most of the stale logging space of $\mathrm{M_0}$ has been proactively reclaimed during periods $\mathrm{T_1}$ and $\mathrm{T_2}$, the logger can be rotated onto $\mathrm{M_0}$ from $\mathrm{M_2}$ again. Note that the logging space occupied by $\mathrm{D_0T_0}$ in $\mathrm{M_0}$ will be reclaimed in $\mathrm{T_3}$, during which destaging I/O activities are issued for $(\mathrm{P_0}, \mathrm{M_0})$.

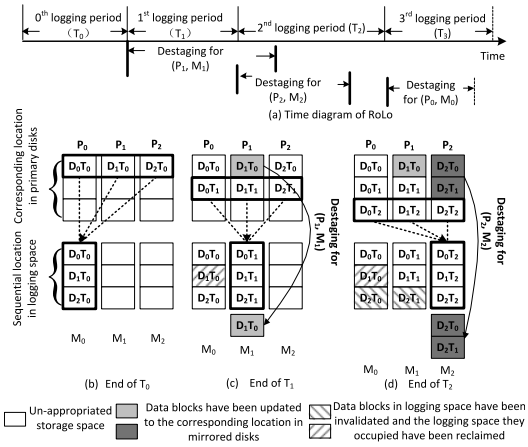


Fig. 4. The decentralized destaging process of RoLo.

As shown in Fig. 4a, the destaging activities may surpass the corresponding logging period due to the intensive foreground I/Os. The prolonged destaging period will induce the increased energy consumption, which trades for the undisturbed foreground I/O performance.

It is possible, in principle, that the destaging of a disk will not terminate before the next destaging process of the same disk is triggered. In this event, we insert the new destaging tasks into this disks unique destaging I/O waiting queue to ensure the destaging tasks being executed in sequence for the simplicity of concurrency control. In our experiments, such events seldom happen.

In our design, mixing logging I/Os with destaging I/Os onto one on-duty log disk represents a tradeoff between performance and energy efficiency. One alternative design is to start destaging the on-duty log after rotating logger to another disk. Our proposal has better energy efficiency than the alternative design. In this paper, we focus on the former and performance penalty can be alleviated when RoLo is enhanced by RoLo-S. The in-depth and detailed comparison between the former and the latter is the future work.

3.2 RoLo-P/R/E

Based on the above principles of logger rotation and decentralized destaging, three flavors of RoLo are proposed in this paper, named the Performance-oriented RoLo (RoLo- P), the Reliability-oriented RoLo (RoLo-R) and the Energy-oriented RoLo (RoLo-E) for respectively specific application scenarios.

3.2.1 RoLo-P

In RoLo-P, all the primary disks are set to the ACTIVE/ IDLE state to guarantee that all the read requests are serviced without any disk spin-up latency, i.e., the latency caused by the disk's state switching from STANDBY to ACTIVE. The disk spin-up latency, which is often more than ten seconds for enterprise-level hard drives [1], is unacceptable for most applications.

In RoLo-P, one or a few *mirrored disks* take turns to serve as on-duty log disks, while off-duty mirrored disks can be spun down to save energy, as shown in Section 3.1. Note that there are *two* copies of each new data block. One copy is written to the corresponding location in the primary disk,

and the other copy is written to the sequential location in the logging space of the on-duty log disk(s), such as M_0.

3.2.2 RoLo-R

Mindful of the fact that the reliability of RoLo-P is slightly lower than that of a RAID10 system (see Section 4.3), RoLo-R is proposed to provide higher reliability by designating one or a few *mirrored disk pairs* as the on-duty logger at a time and writing *three* copies for each new data block. One copy is written to the corresponding location in the targeted primary disk, and the other *two copies* are written to the *two disks* in one mirrored disk pair that serve as the on-duty logger, e.g., (P_0, M_0), respectively. Note that all the primary disks are also set to the ACTIVE/IDLE state for the same purpose as in RoLo-P.

3.2.3 RoLo-E

Motivated by the fact that there are no or very few read requests for some applications, such as storing checkpointing data sets in high performance computing environments [5], we believe that it is not necessary to keep all the primary disks ACTIVE/IDLE all the time. Thus we propose RoLo-E to utilize one or several *mirrored disk pairs*, such as (P_0, M_0), as log disks at a time and spin down all the other mirrored disk pairs to the low-power STANDBY state. *Two* copies of each new data block are written to both the logging space of the primary disk and that of the mirrored disk respectively. To further alleviate the long response time of some reads, we cache popular read data blocks in the on-duty logging space to avoid the passive and expensive disk spin up/down caused by read misses.

3.2.4 Comparison Among RoLo-P/R/E

Note that RoLo-P, RoLo-R and RoLo-E are suitable for different application environments and users can choose the suitable scheme according to their application requirement. RoLo-P/R can be used under read/write mixed workloads, however, RoLo-E can only be used under extremely write dominated workloads, such as checkpointing storage and archival storage. The evaluation results also show that both unaccepted latency and lots of disk spin up/down events are incurred by non-write dominated workloads.

RoLo-P trades both reliability and energy efficiency for performance. RoLo-R trades both performance and energy efficiency for reliability. RoLo-E trades performance and application scene suitability for energy efficiency. From evaluation results in Section 5 one can see that the performance of RoLo-P is better than that of both RoLo-R and RoLo-E. Besides, the energy efficiency of RoLo-P and RoLo-R are almost the same with each other. However, the energy efficiency of both RoLo-P and RoLo-R are worse than that of RoLo-E. As shown in Section 4.3, the reliability of RoLo-R is higher than that of both RoLo-P and RoLo-E.

3.3 RoLo-S

3.3.1 The Basic RoLo-S Idea

Fig. 5 shows the data regions and free regions distribution of primary disks and mirrored disks in both RoLo and RoLo-S. From Fig. 5a, the data region of primary disks is set with low LBN. In Fig. 5b, there is long disk head seek distance and the corresponding long disk head seek time in mirrored disks of RoLo. However, RoLo-S redirects logging I/Os to the nearest available logging region adjacent to the current destaging I/Os' region to minimize the disk head seek distance and thus the average disk head seek time, as shown in Fig. 5c. Note that the data region is stretched, and the write I/O footprints span the entire mirrored disks, however, the stretched data write I/Os footprints does not decrease the energy efficiency and performance. On the contrary, the gathering of write I/Os reduces the disk head seek time and thus improves the energy efficiency and performance.

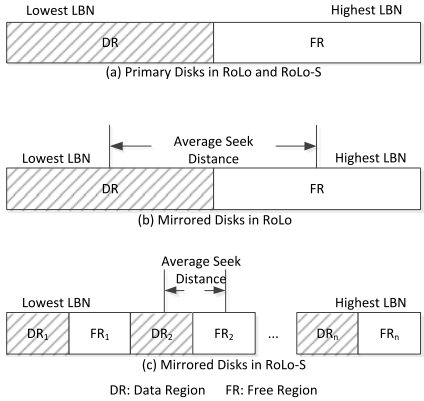


Fig. 5. Data regions and free regions distribution of primary disks and mirrored disks in both RoLo and RoLo-S.

3.3.2 Size of Data Region and Free Region

Modeling and analysis in Section 4.1 and detailed experimental evaluation results in Section 5.3.2 show that the smaller size of data region and free region leads to the shorter disk head seek. For simplicity and without loss of generality, we assume all the data regions are set with the same size d_{dr}, and all the free regions are set with the same size d_{fr}. Supposing that the application workloads do not vary violently among consecutive logging periods, the number of logging I/Os and destaging I/Os in on-duty log disk are approximately equal. Based on the above analysis, we use the same strip unit size for both data region and free region of the RAID0 composed by primary disks. We find that it is a good balancing point for the performance and energy efficiency. Particularly, we use the setting of 64 KB in the evaluation of this paper.

3.4 Enhancing RoLo with RoLo-S

Note that RoLo-S is orthogonal to RoLo-P/R/E. Specifically, RoLo-P/R/E rotate logging among multiple disks, and RoLo-S is a complementary optimization used for each on-duty log disk in RoLo-P/R/E. That means, for each on-duty log disk in our proposed logging scheme, RoLo-S can be used to alleviate the performance bottleneck and improve the energy efficiency via gathering write I/Os and reducing the disk head seek distance and time. Based on the characteristics of energy consumption and performance of hard disks discussed in Section 2.3, we know that the reducing of disk head seek distance and time in RoLo-S can improves the energy efficiency and performance of one single disk.

RoLo-E is mainly used in the extremely write dominated workloads and both of the two disks are used as the on-

duty logger, there is no destaging I/Os activities disturbing the sequentiality of logging I/Os. Thus RoLo-S's zebra-crossing data layout will not contribute to the further performance improvement of RoLo-E and it is not necessary to combine RoLo-E with RoLo-S.

3.5 Discussion

Since the peak bandwidth of one on-duty log disk is limited, RoLo can dynamically adjust the number of on-duty log disks to suit for the performance requirement.

Spinning down redundant disks to low-power state is the main principle to save energy in RoLo model. So RoLo is tightly coupled with the mirrored disk architecture. Since there are only one and two redundant disks in two representative parity based disk architectures, RAID5 and RAID6, we can not build RoLo on parity based disk arrays.

RoLo is a general logging model mainly to improve the energy efficiency of storage systems. Note that RoLo is not only suitable for RAID systems, but also can be applicable to redundancy based storage models, such as distributed file/storage systems and shared log based cloud storage systems.

The original or slightly enhanced RoLo model can be used in logging cloud applications and data stores. For example, RoLo can be integrated with CORFU [38], which is the underlying log platform of Tango [37], to enhance the reliability and energy efficiency of CORFU. RoLo can also be used to provide the reliable and energy efficient log based storage systems for both LogBase [40], a scalable log-structured database system in the cloud, and Hyder [39], a transactional record manager.

4 Performance, Energy Consumption and Reliability Modeling and Analysis

4.1 Performance Modeling and Analysis

Because RoLo spreads destaging I/O activities only among idle time slots during logging periods, we can see that the basic model of RoLo is the similar with that of GRAID. We can further conclude that the average response time of application I/Os of RoLo is approximately equal to that of GRAID. This conclusion is also confirmed by the evaluation results in Section 5.

The number of on-duty log disks in RoLo can be adjusted according to the application response time requirements, RoLo can achieve better balance between application performance requirement and energy saving. However, there is only one dedicated log disk in GRAID, and the application performance will be obviously disturbed due to the limited log disk bandwidth. We compare the average disk-head seek distance of RoLo and RoLo-S. The parameters definition are shown in Table 1 and Table 2.

Modeling of RoLo. The continuous width of data region is $c \times \rho$, and the continuous width of free region is $c \times (1-\rho)$. For both random and sequential destaging I/Os, the destaging I/Os and logging I/Os are redirected to data region and free region separately. Thus, the average disk-head seek distance of RoLo is $\frac{c}{2}$.

Modeling of RoLo-S. The data regions and free regions are interleaved distributed among the physical storage capacity of mirror disks.

TABLE 1
Parameter Definition

Parameters	*Definition*
c	physical storage capacity of primary/mirror disk
p	width of each data region in RoLo-S
q	width of each free region in RoLo-S
ρ	ratio of width of data region
ϕ	destaging I/O dispersion degree
D_{RoLo}	average disk-head seek distance of RoLo
D_{RoLo-S}	average disk-head seek distance of RoLo-S
C_{disk}	storage capacity of disks
N_{prm}	number of primary disks in GRAID and RoLo
N_{log}	number of logger disks in RoLo
R_{free}	free region ratio in RoLo
R_{input}	data write rate in MB/s
$R_{GRAID_{des}}$	destaging rate in GRAID
$T_{GRAID_{des}}$	destaging period length in GRAID
$T_{GRAID_{log}}$	logging period length in GRAID
β	destaging period ratio in GRAID
$T_{RoLo_{des}}$	destaging period length in RoLo
$T_{RoLo_{log}}$	logging period length in RoLo
P_a	active power of disks
P_s	standby power of disks
γ	ratio between standby power and active power
P_{GRAID}	power of GRAID
P_{RoLo}	power of RoLo
P_{saved}	power saved by RoLo from GRAID

For 100 percent sequential destaging I/Os, the average disk-head seek distance of RoLo-S is $\frac{p+q}{2}$, which is less than $\frac{c}{2}$. For 100 percent random destaging I/Os, the average disk-head seek distance of RoLo-S is $\frac{c}{2}$.

The average disk-head seek distance of RoLo-S D_{RoLo-S} is linearly correlation with the destaging I/Os dispersion degree ϕ, $D_{RoLo-S} = \frac{p+q}{2} \times (1-\phi) + \frac{c}{2} \times \phi$, where $\phi \in [0,1]$. According to the relationship between disk-head seek distance and performance, the performance of RoLo-S is always better than that of RoLo.

4.2 Energy Consumption Modeling and Analysis

We estimate the energy consumption for all logging architectures within one logging period.

The total logging space in GRAID is C_{disk}. The log disk in GRAID can be filled in $T_{GRAID_{log}} = \frac{C_{disk}}{R_{input}}$ and the destaging period is $T_{GRAID_{des}}$. There are $N_{prm}+1$ active disks and N_{prm} standby disks during the logging period, and $N_{prm}+N_{prm}+1$ active disks during the destaging period. So the power of GRAID is $P_{GRAID} = (N_{prm}+1) \times P_a + N_{prm} \times P_s + N_{prm} \times (P_a - P_s) \times \beta$.

TABLE 2
Disk Drive Reliability Parameters

Parameter	Definition	Value
$\frac{1}{\mu}$	MTTR for one disk failure	12 h
C_d	Capacity per disk	1 TB
S	Sector size	512 bytes = 4,096 bits
P_{bit}	UER per bit read	10^{-15}
R_{afr}	Annual Failure Rate (AFR)	0.73 percent

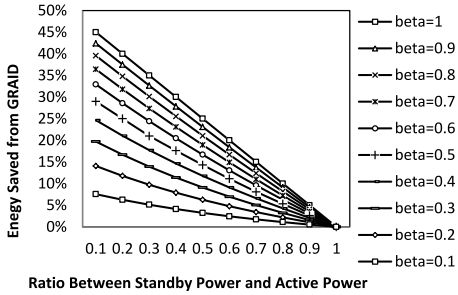


Fig. 6. The Relationship between the Energy of RoLo saved over GRAID and γ and β.

We know that there are always $N_{prm}+1$ active disks and $N_{prm}-1$ standby disks in RoLo, so the power of RoLo is $P_{RoLo}=(N_{prm}+1)\times P_a+(N_{prm}-1)\times P_s$.

The power saved by RoLo from GRAID is

$$P_{saved}=\frac{P_{GRAID}-P_{RoLo}}{P_{GRAID}}=1-\frac{(N_{prm}+1)+(N_{prm}-1)\times\gamma}{(N_{prm}+1)+N_{prm}\times\gamma+N_{prm}\times(1-\gamma)\times\beta}. \quad (1)$$

If we assume $N_{prm}\pm 1\approx N_{prm}$, we can simplify P_{saved} as follows.

$$P_{saved}\approx 1-\frac{1}{1+\frac{1-\gamma}{1+\gamma}\times\beta}. \quad (2)$$

We plot the relationship between P_{saved} and γ in Fig. 6. We can make the following two observations. First, the smaller γ means that the standby disks consumes much less energy compared with the active disks and then the power saved by RoLo from GRAID increases. Second, if the destaging period ratio in GRAID β increases, the power saved by RoLo from GRAID will increase.

4.3 Reliability Modeling and Analysis

The MTTDL model, which excludes latent defects and implicitly assumes that HDD failures follow a homogeneous Possion process (HPP), cannot provide the accurate estimation of reliability[33]. In this paper, we incorporate both disk failures (DF) and unrecoverable failures (UF) (i.e., latent sector errors [3], [33]) into the reliability model of RoLo, GRAID and RAID10 to make a more accurate estimation. We then simulate the time dependent and chronological behavior of the system in an sequential Monte Carlo (SMC) simulation [33] using a three-parameter Weibull distribution [13], [33].

4.3.1 Analytical Model

The failure modes and mechanisms based on HDD electromechanical and magnetic events can be divided into two categories [33]: disk failures and unrecoverable failures, both of which are significant and must be included in the storage systems reliability model.

The following four distributions are required [33], i.e., disk failure distribution (d_{df}), restoring disk failure distribution (d_{rs}), unrecoverable failure distribution (d_{uf}), and scrubbing for latent defects distribution (d_{sc}).

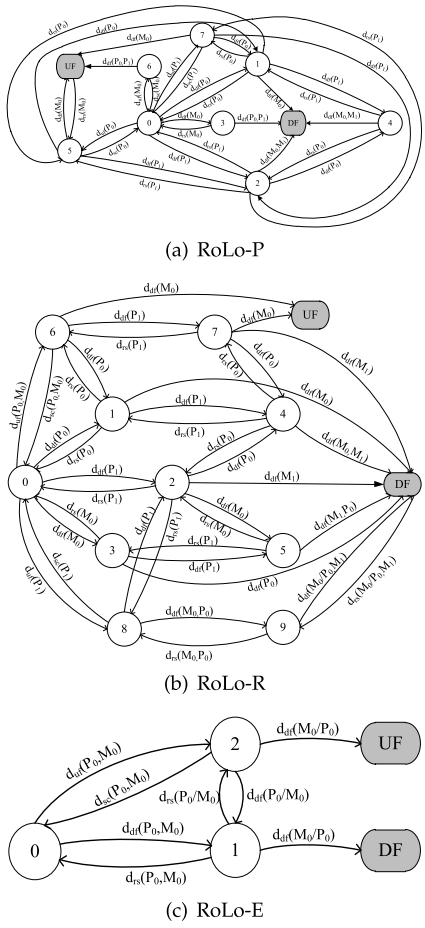


Fig. 7. State diagram for RoLo-P/R/E.

As stated in [33], the order of occurrence of disk failure and unrecoverable failure is significant. The following two scenarios result in data loss. First, two simultaneous disk failures. Second, an disk failure that occurs after a unrecoverable failure has been introduced and before it is corrected. Since the probability of suffering a usage-related data corruption in an unread area during the time of reconstruction is small, multiple simultaneous unrecoverable failures do not constitute failure.

Due to the space constraints, we have to omit the detailed description on the reliability models of RoLo, GRAID and RAID10. Here, we only present the state transition diagrams, as shown in Fig. 7. $d_{df}(Xi)$ demonstrates that some state transition is disk failure distribution with the failed disk X_i. Note that multiple X_i indicate that any one of the disks can trigger the transition. In all the three sub-figures, state 0 denotes the state in which all disks work normally, state DF and UF denote the disk failure and unrecoverable failure state respectively.

4.3.2 Simulation Study

The SMC simulation study was conducted to estimate reliability measures of the RoLo, GRAID and RAID10, by simulating 100,000 RAID sets in 87,600 hours (10 years) with HDD failures following a three-parameter Weibull distribution. Each transition distribution in Fig. 7 is sampled. During the simulation, events such as hard disk failures,

TABLE 3
Failure Distribution Parameters

disk failure & restore distribution						unrecoverable failure & scrub distribution					
d_{df}			d_{rs}			d_{uf}			d_{sc}		
γ	η	β	γ	η	β	γ	η	β	γ	η	β
0	461,386	1.12	7	12	2	0	1,776	1	7	168	3

rebuilds, latent sector errors, disk failures, and unrecoverable failures are tracked. The current state of a RAID is sampled in the interval of 1 hour. The state transition (when and where to) is determined by the outcome of a random test that follows the relevant stochastic processes (e.g., Weibull distribution for disk failures and repairs, and uniform (spatial and temporal) distribution for sector errors).

For simplicity, a constant unrecoverable bit error rate independent of time and workload was used, similar to the interleaved parity check (IPC) [36] study. The three parameters of Weibull distributions, i.e., the location parameter γ, the characteristic life η, and the shape parameter β, are shown in Table 3.

Since both disk failures and unrecoverable failures can occur at any time, so the location parameters γ of these two distributions d_{df} and d_{uf} are 0. Both the disk failure restoration time and the minimum time to scrub an entire HDD are decided by multiple parameters, such as the HDD capacity, the data rate of the HDD and the priority of restoration. Since the minimum time of full speed restoration of one HDD with 1 TB capacity and 150 MB/s Sustained Transfer Rate is 6.7 hours, we set the location parameter γ of both restoring disk failure distribution (d_{rs}) and scrubbing for latent defects distribution (d_{sc}) as seven hours.

In a Weibull distribution, the shape parameter, β, indicates whether the failure rate is decreasing $\beta < 1.0$, constant ($\beta = 1.0$), or increasing ($\beta > 1.0$). Due to the batch-correlated disk failure, Weibull distribution with a slightly increasing failure rate is used in disk failure distribution and thus the shape parameter β of d_{df} is set as 1.12, according to the empirical statistics and used in [33]. The shape parameters β of d_{rs} and d_{sc} are set as 2 and 3 respectively to generate a right-skewed distribution. The shape parameter β of d_{uf} is set as 1 and indicates that the recoverable failure shows uniform (spatial and temporal) distribution.

The characteristic life, η, of d_{df}, d_{rs} and d_{sc} are 461, 386 hours (MTBF), 12 hours (MTTR) and 168 hours (Scrubbing Period) respectively, according to the empirical statistics from a field population of over 120,000 HDDs that operated for up to 6,000 hours, and has been widely used in previous studies. The η of d_{uf} is derived from the bytes read per hour and read errors per byte per HDD and set as 1,776 hours with parameters of Seagate ST1000NM0011 [1].

4.3.3 Results and Discussion

We get the number of disk failure (NDF) and number of unrecoverable failure (NUF) from the SMC simulation study, and then derive the total and normalized comparisons of number of failures. From Table 4, we can conclude that the reliability of RoLo-R outperforms RAID10 due to the fact that there are three copies of each write data written to three independent disks. Meanwhile, the reliability of both GRAID and RoLo-P are slightly worse than that of RAID10, because the second copies of all the write data are written to a single log disk in both GRAID and RoLo-P. RoLo-P outperforms GRAID, because only a small subset of the relevant mirrored disks are spun up for the recovery of the failure of any primary disk in RoLo-P, while all the mirrored disks are spun up for the recovery of the failure of any primary disk in the design of GRAID. The reliability of RoLo-E outperforms all the comparison schemes because there are only two active disks incorporated into the reliability model in RoLo-E.

TABLE 4
Numeric and Normalized Comparisons of Number of Failures (UER $= 10^{-15}$)

Schemes	NDF	NUF	Total	Normalized
RAID10	21	3,018	3,039	1
GRAID	25	3,506	3,531	0.86
RoLo-P	23	3,086	3,109	0.98
RoLo-R	18	2,492	2,510	1.21
RoLo-E	10	1,526	1,536	1.98

We also note that there have been some other approaches to alleviate the increased disk disk failure rate. For example, Paris et al. [34] pointed that enhancing device diversity can protect data against batch-correlated disk failures .

5 Performance Evaluations

5.1 Experimental Setup and Methodology

In order to evaluate the performance and energy efficiency of RoLo and RoLo-S, we have implemented RoLo and RoLo-S prototypes in the Linux Software RAID (MD, Multiple Devices) as independent modules. We have also implemented GRAID [15] as a comparable alternative to RoLo.

In our evaluations, we use average response time to evaluate the performance of all the schemes. A RAID10 disk array consisting of 10 disks is adopted in our evaluation to be a performance and energy consumption baseline [10]. A widely used SATA disk, Seagate ST1000NM0011 [1], is used throughout our experiments. Its main specifications and RAID configuration parameters are listed in Table 5. We

TABLE 5
Disk and RAID Configuration Parameters

Disk Parameter	*Value*
Disk Model	Seagate ST1000NM0011
Capacity/Rotational Speed	1 TB/7,200 RPM
Avg. Seek/Rotational Time	8.5 ms/4.2 ms
Sustained Transfer Rate	150 MB/s
Power Parameters	***Value***
Power(Active, Idle, Standby)	6.82 W, 4.61 W, 0.57 W
Spin down/up time	0.4 s/9.6 s
RAID Configuration	***Value***
RAID level	RAID10
Stripe Unit Size	4 KB, 16 KB, 64 KB
Number of Disks	6, 8, 10 (7, 9, 11 for GRAID)
Free Disk Space	400/300/200 GB(800 GB for GRAID)

TABLE 6
A Summary of Characteristics of Seven Traces

Trace	Write Ratio	IOPS	Avg.Req Size	Data Amount
src2_2	99.6 percent	78.80	63.64 KB	33 GB → 2 TB
proj_0	94.9 percent	23.89	51.42 KB	99.3 GB → 2 TB
mds_0	88.1 percent	2.00	9.20 KB	7.0 GB
wdev_0	79.9 percent	1.89	9.08 KB	7.15 GB
web_1	45.9 percent	0.27	29.07 KB	664 MB
rsrch_2	34.3 percent	0.35	4.08 KB	295 MB
hm_1	4.7 percent	1.02	15.16 KB	553 MB

issue AIO (Asynchronous I/O) to RoLo, GRAID and RAID10 supported MD using a user-mode I/O trace replay tool according to the seven traces.

The traces used in our experiments are collected from a production data center in Microsoft Research UK with a total of 36 different traces [20]. Since the main design goal of RoLo is to significantly benefit write-dominated applications in energy-efficiency, reliability and performance, we select two of the 36 traces, i.e., $src2\_2$ and $proj\_0$, that have the highest write-ratio for use. We choose five additional traces, i.e., mds_0, $wdev\_0$, web_1, $rsrch\_2$ and hm_1, to show the effectiveness of RoLo under non-write-dominated workloads. The traces chosen for our experimental study have different read/write ratios, IOPS (I/O per second), and average request sizes, to represent multiple types of workloads in real enterprise-level data centers, such as source control (*src*), project directories (*proj*), media server (*mds*), test web server (*wdev*), web/SQL server (*web*), research projects (*rsrch*) and hardware monitoring (*hm*) [20]. The summary of the characteristics of the seven traces are listed in Table 6.

Note that $src2\_2$ is extremely write dominated, and can only be used in the special application environments, such as checkpointing data storage and archival storage, which are write once read rarely. There are increasingly extremely write dominated applications. For example, massive video monitoring data in intelligent cities are written to the storage systems sequentially and only read when they are needed.

The RoLo architecture is designed to use in the 7×24 hours running storage systems, and the I/Os are continuously issued to RoLo. The main purpose of the evaluation in this paper is to show the advantage of rotated logging, and so we should make the logger rotating among multiple mirrored disks. Note that the free space of each disk in RoLo is 400 GB and that in GRAID is 800 GB, however, the seven data amounts seen by the traces are only 295 MB, 553 MB, 664 MB, 7.0 GB, 7.15 GB, 33 GB and 99.3 GB respectively. We also expand the data amount of both $src2\_2$ and $proj\_0$ to 2TB by repeatedly playbacking them, in order to fill up the whole logging space in RoLo with 10 mirrored disks.

5.2 Comparison among RoLo, GRAID and RAID10

5.2.1 Main Experimental Evaluation

We first conduct our experiments on a RAID10 disk array consisting of 10 disks with a fixed stripe unit size of 64 KB

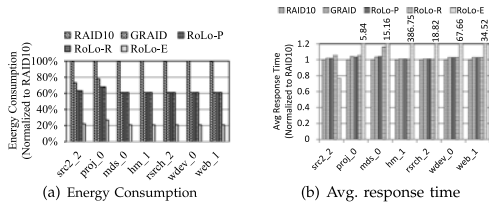


Fig. 8. Energy consumption and average response time, normalized to RAID10, under different traces.

to evaluate the energy efficiency and performance of RoLo compared with RAID10 and GRAID.

Fig. 8 compares the energy consumption and average response time of RoLo with the standard RAID10 and GRAID schemes under two write dominated traces $src2\_2$ and $proj\_0$, and five write non-dominated traces mds_0, $wdev\_0$, web_1, $rsrch\_2$ and hm_1. From Fig. 8a, one can see that the energy consumptions of RoLo-P and RoLo-R are almost the same under all the seven traces. The reason is that the only difference between RoLo-P and RoLo-R is an extra write operation on the primary log disk for write requests in RoLo-R. Since the primary disks in RoLo-P and RoLo-R are set ACTIVE all the time, an extra write operation has negligible impact on energy consumption.

For two write dominated traces $src2\_2$ and $proj\_0$, one can see that RoLo-P and RoLo-R consume 30.8 and 35.4 percent less energy than RAID10 under $proj\_0$ and $src2\_2$ respectively. RoLo-E consumes the least amount of energy under both $proj\_0$ and $src2\_2$, 72.1 and 76.3 percent less than RAID10 under $proj\_0$ and $src2\_2$ respectively. In addition, RoLo-P and RoLo-R consume less energy than GRAID under $proj\_0$ and $src2\_2$ by 12.1 and 13.2 percent respectively. This is because the logical logger space consisting of the free storage space of disks in the former two is much larger than that of GRAID so that the idle periods of disks can be much longer than that of GRAID. RoLo-E consumes less energy than GRAID under $proj\_0$ and $src2\_2$ by 67.4 and 69.7 percent respectively since only one mirrored disk pair is kept ACTIVE and used as logger in RoLo-E.

For five write non-dominated traces mds_0, $wdev\_0$, web_1, $rsrch\_2$ and hm_1, the energy consumptions of RoLo-P and RoLo-R are the same as that of GRAID and the average response time of RoLo-R is worse than those of RoLo-P and GRAID under mds_0, $wdev\_0$, web_1, $rsrch\_2$ and hm_1 by 0.7-5.3 percent. It indicates that when RoLo is deployed in write non-dominated application environments, its negative impact, if any, is negligible.

Fig. 8b shows a comparison of average response times of RAID10, GRAID, RoLo-P, RoLo-R and RoLo-E under all the seven traces. The average response times of GRAID and RoLo-P are nearly the same under all the seven traces because the basic model of RoLo-P is the same as that of GRAID except for the logger management policy. One can see that the average response time of RoLo-P is slightly greater than RAID10 under all the seven traces by 6.4-0.9 percent, while RoLo-R underperforms RAID10 under all the seven traces by 14.3-0.1 percent. The reason behind RoLo-R's inferior performance to RoLo-P, by 3.9-7.5 percent, is because the former issues three copies of each written data to provide higher reliability while the latter issues only two such copies.

TABLE 7
Characteristics of RoLo-E under $src2\_2$ and $proj\_0$

Trace	Read Ratio	Read Hit Rate	Burstiness
proj_0	5.1 percent	26.7 percent	Very Low
src2_2	0.4 percent	90.6 percent	Very High

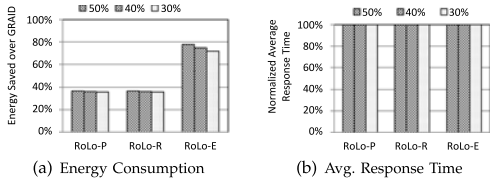
It is interesting to notice the polarization of the average response time of RoLo-E under mds_0, $wdev\_0$, web_1, $rsrch\_2$, hm_1, $proj\_0$ and $src2\_2$. The evaluation results reveals that the average response time of RoLo-E is 5.84×, 15.16×, 67.66×, 34.52×, 18.82× and 386.75× larger than that of RAID10 under $proj\_0$, mds_0, $wdev\_0$, web_1, $rsrch\_2$ and hm_1 respectively, however, the average response time of RoLo-E is only 23.4 percent less than that of RAID10 under $src2\_2$. To investigate the main reason behind this performance disparity of RoLo-E under the seven traces, we analyze the corresponding read hit ratio during our experiments. Table 7 shows that the read ratio of $src2\_2$ approaches to 0 while its read hit ratio is over 90 percent, which sharply contrasts to $proj\_0$ that has a read ratio about 14 times that of $src2\_2$ and a read hit ratio of less than 30 percent. This means that $proj\_0$ results in much more read misses than $src2\_2$, where each missed read request translates into a disk spin-up operation and incurs an expensive latency penalty of 1,000-10,000 times that of a read hit. Because the reason behind the polarization of average response time among mds_0, $wdev\_0$, web_1, $rsrch\_2$, hm_1 and $src2\_2$ is the same with that between $proj\_0$ and $src2\_2$, we only list the comparison results between $proj\_0$ and $src2\_2$ in Table 7 for the space constraint.

5.2.2 Sensitivity Study

Both $src2\_2$ and $proj\_0$ are write dominated workloads. We noticed that RoLo, GRAID and RAID10 show the similar energy consumption and performance comparison results under both $src2\_2$ and $proj\_0$, except for the average response time disparity. There will be lots of disk spins for RoLo-E under $proj\_0$, and thus $proj\_0$ is unsuitable for the sensitivity study of RoLo-E. In order to study the sensitivity of RoLo-P/R/E, we pick up $src2\_2$, although it can only be used in the extremely write dominated environments.

Number of disks. We conduct experiments on different numbers of disks (6, 8, 10 for RoLo and 6 + 1, 8 + 1, 10 + 1 for GRAID) in a RAID10 disk array with a stripe unit size of 64 KB.

As all the primary disks of the RoLo-P, RoLo-R and GRAID schemes are kept ACTIVE, it is expected the more energy is consumed with an increase in the number of disks. The experimental results, not shown here for space constraint, reveal that, the amount of energy saved by GRAID, RoLo-P, RoLo-R and RoLo-E from RAID10 scheme increase by 1.8, 2.6, 2.6 and 8.4 percent respectively as the number of disks increases from 6 to 10 under $src2\_2$. We can conclude that proportionally more energy can be saved as the number of disks increases for GRAID, RoLo-P, RoLo-R and RoLo-E. Note that the amount of the increased energy saving by RoLo-P, RoLo-R and RoLo-E are much larger than that by GRAID as the disk array size increases. The average response times of GRAID, RoLo-P, RoLo-R and RoLo-E decrease as the disk array expands in size, as a result of the increased disk access parallelism.



(a) Energy Consumption (b) Avg. Response Time

Fig. 9. Energy saved over GRAID and average response time as a function of the amount of free storage space under $src2\_2$.

Free storage space. The experimental results, as shown in Fig. 9, reveal that, the amounts of energy saved by RoLo-P, RoLo-R and RoLo-E from GRAID slightly decrease with a reduction in the free storage space. The reason is that the capacity of the active logging space, which is proportional to the free space available on the on-duty logger, decreases with a reduction in the free space of every disk. This in turn means a shorter logging period that requires more frequent rotations of the logger, i.e., more frequent disk spins.

The average response times of RoLo-P, RoLo-R and RoLo-E are almost unchanged as the free storage space changes, suggesting that the average response time of RoLo is not sensitive to the amount of available free storage space. The reason is that the background destaging I/O activities with a lower priority have no or little impact on the foreground I/O performance, even though the decreased free storage space shortens the logging period.

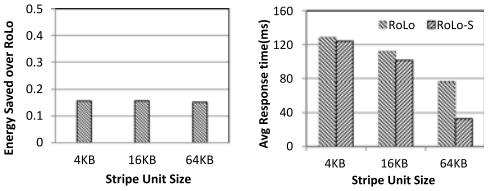
Stripe unit size. We conduct experiments on a 10-disk RAID10 disk array with stripe unit sizes of 16, 32 and 64 KB, respectively. The experimental results, not shown here for space constraint, reveal that, except for RoLo-E that is noticeably sensitive to stripe unit size under $src2\_2$, none of the schemes is sensitive at all to stripe unit size in terms of energy efficiency. To explain the sensitivity of RoLo-E to stripe unit size under $src2\_2$, we find from the trace characteristics that the average read request size is 68.08 KB for $src2\_2$. When the stripe unit size is set to 16 KB under $src2\_2$, most of the read requests are split into more than one sub-requests and more disks have to be spun up. The number of disks spun up for the read miss requests are reduced as stripe unit size increases, which explains why energy saved by RoLo-E from RAID10 increases as the stripe unit size increases from 16 KB to 64 KB.

5.3 Comparison between RoLo and RoLo-S

Since RoLo-S makes the similar performance improvement of RoLo-P and RoLo-R, we only present the experiments between original RoLo-P and enhanced RoLo-P (i.e., RoLo-S) to show the energy efficiency and performance improvement.

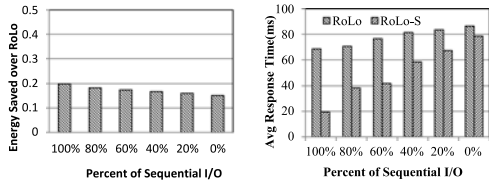
5.3.1 Main Experimental Evaluation

We first conduct our experiments on a RAID10 disk array consisting of eight disks with a fixed stripe unit size of 64 KB to evaluate the energy efficiency and performance of RoLo-S compared with RoLo under $src2\_2$. The experimental results, not shown here for space constraint, reveal that the average response time of RoLo-S is 48.9 percent smaller



(a) Energy Consumption (b) Avg. Response Time

Fig. 10. Energy consumption and performance comparison between RoLo and RoLo-S as a function of the stripe unit size under $src2\_2$.



(a) Energy Consumption (b) Avg. Response Time

Fig. 12. Energy consumption and performance comparison between RoLo and RoLo-S as a function of sequential I/O ratio driven by IOmeter.

than that of RoLo, the energy consumption of RoLo-S is 15.7 percent lower than that of RoLo. The excellent energy efficiency and performance exists on the reduced disk head seeks.

5.3.2 Sensitivity Study

We conduct both open-loop real-life traces driven sensitivity studies and closed-loop IOmeter driven sensitivity studies.

Number of disks. We conduct experiments on different numbers of disks (6, 8 and 10) in a RAID10 disk array with a stripe unit size of 64 KB. The experimental results, not shown here for space constraint, reveal that, the amount of energy saved by RoLo-S from RoLo does not vary with the number of disks. RoLo-S outperforms RoLo in average response time by 17.9-48.7 percent for all the three disk array scale. Moreover, the average response time of RoLo and RoLo-S decreased as the disk array expands in disk numbers, as a result of the increased disk access parallelism.

Stripe unit size. We conduct experiments on a 10-disks RAID10 disk array with stripe unit size of 4, 16 and 64 KB, respectively. From Fig. 10b we can conclude that RoLo-S outperforms RoLo in average time by 3.7-56.5 percent for all the stripe unit size setting. Moreover, the average response time of RoLo and RoLo-S decreases as the increasing of stripe unit size. The experimental results, as shown in Fig. 10a, reveal that, none of the schemes is sensitive at all to stripe unit size in terms of energy efficiency.

Width of zebra-crossing. We set the width of zebra-crossing of RoLo-S as 32, 64 and 128 MB. The experimental results in Fig. 11 reveal that the average response time of RoLo-S decrease along with the reduction of zebra-crossing width, which demonstrates that the smaller zebra-crossing width leads to the shorter disk head seek thus lower average response time and less energy consumption.

In addition to the open-loop real-life trace-driven experiments, we also conduct experiments driven by IOmeter to evaluate the impacts of the sequential I/O ratio and read I/O ratio.

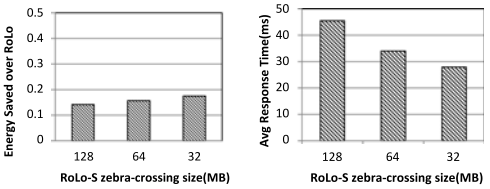
Sequential I/O ratio. We set the percentage of sequential I/O requests as 0, 20, 40, 60, 80 and 100 percent. The experimental results in Fig. 12 reveal that the average response time of both RoLo and RoLo-S decreases along with the increasing of the percentage of sequential I/O requests due to the decreasing of disk-head seek distance and seek time, which is consistent with the modeling and analysis result presented in Section 4. However, RoLo and RoLo-S have different amplitudes of decrease because the disk head seek cost is also obvious with 100 percent sequential I/O workload. We also find that the average response time of RoLo-S is always lower than that of RoLo under different sequential I/O ratio.

Read I/O ratio. We set the percentage of read I/O requests as 0, 50 and 100 percent respectively. Fig. 13 shows that the average response time of RoLo and RoLo-S under 100 percent read requests is much smaller than that under 0 and 50 percent read requests, the reason is that the two write sub-requests in one pair of mirrored disks will wait for each other and the read requests are served by the primary disks only in RoLo and RoLo-S. We find that the average response time of both RoLo and RoLo-S decrease along with the increasing of read request ratio, because the small read requests response time pulls down the total average response time. We also find that average response time reduction of RoLo-S compared with RoLo decrease along with the increase of read ratio, which demonstrates that the advantages of RoLo-S will be highlighted under the condition of write dominated workloads.

6 Related Work

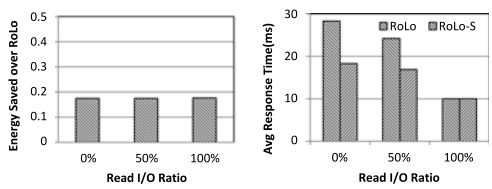
6.1 Logging Techniques

Logging techniques have been widely used in hierarchical storage architectures to improve performance or energy efficiency of storage systems. DCD [11] uses a small log disk as a secondary disk cache beneath a memory cache to optimize write performance. Logging RAID [4] bundles small writes into large writes to overcome the small-write performance problem in parity-based disk arrays. GRAID [15] concentrates the second copy of write data blocks



(a) Energy Consumption (b) Avg. Response Time

Fig. 11. Energy consumption and performance comparison between RoLo and RoLo-S as a function of zebra-crossing size under $src2\_2$.



(a) Energy Consumption (b) Avg. Response Time

Fig. 13. Energy consumption and performance comparison between RoLo and RoLo-S as a function of read I/O ratio driven by IOmeter.

onto an extra log disk to prolong the idle period of mirrored disks to save energy. RoLo rotates the logger among the logical logging space pool formed by free storage space, thus avoiding the dedicated and extra log disks. RoLo is also different from Log-structured File System [21] in how the write data blocks are updated to the targeted permanent location.

6.2 Destaging Algorithms

When and which data blocks to destage are the main concerns of destaging schemes. Different from the fixed periodic update policy in traditional UNIX systems [18], modern destaging schemes used in *MEMORY/NVRAM-DISK* multi-level storage architectures tend to maintain a good tradeoff between exploiting temporal locality and spatial locality for destaging and preventing frequent write buffer overflow [2], [7], [19], [25], [41], [42]. However, RoLo is used in log-disk based storage systems. , for which the capacity of log disk(s) is much larger than that of expensive small capacity NVRAM. RoLo is similar to the previous schemes in scheduling the destaging process as a background task and steals idle periods for destaging. However, in order to prolong the idle periods of as many mirrored disks as possible, RoLo carries out decentralized destaging along with rotated logging to alleviate or eliminate the negative impact on energy efficiency caused by centralized destaging in conventional centralized logging architecture, e.g., GRAID. During the decentralized destaging of RoLo, spatial locality is exploited to bundle as many data blocks with successive location as possible in one destaging I/O operation.

6.3 Energy Efficient Schemes

Single-hard-disk energy efficiency optimization schemes, such as dynamic RPM (DRPM) [9], intra-disk parallelism (IDP) [22] and FS2 [12], are orthogonal to and thus can help extend the power savings of RoLo, which is optimized for write-dominant workloads. Free space is exploited in FS2 and PARAID [26] to save energy, where FS2 utilizes free space to replicate some data blocks to minimize the disk positioning time while PARAID uses it to gather all active data onto a small number of disks in a RAID. In contrast, RoLo utilizes free space to form a large logical logging space.

7 Conclusion

In this paper, we propose a rotated logging RAID10 storage architecture RoLo, which combines decentralized destaging with rotated logging, to enhance the energy efficiency of conventional centralized logging architecture. We design three flavors of RoLo, i.e., RoLo-E/R/P, for respectively specific application scenarios and analytically comparison on their reliability. RoLo-S is developed to further alleviate the performance bottleneck and energy consumption caused by the frequent disk head seek in on-duty logger disks. Extensive trace-driven evaluations on real disk systems demonstrate that RoLo-P/R saves up to 13.2 percent power than GRAID, meanwhile RoLo-S outperforms RoLo in average response time and energy consumption by 48.9 and 15.7 percent respectively.

Acknowledgments

This work was partially supported by National Science Foundation of China under Grant No. 61303056, National High Technology Research and Development Program of China under Grant No.2013AA013203, Bingsheng's work is supported by a MoE AcRF Tier 2 grant (MOE2012-T2-1-126) in Singapore.

References

[1] *Seagate ST1000NM0011 Datasheet*. [Online]. Available: http://www.seagate.com/internal-hard-drives/enterprise-hard-drives/hdd/enterprise-capacity-3-5-hdd/, 2012.
[2] M. Alonso and V. Santonja, "A new destage algorithm for disk cache: DOME," in *Proc. 25th EUROMICRO Conf.*, 1999, pp. 416–423.
[3] L. N. Bairavasundaram, G. R. Goodson, S. Pasupathy, and J. Schindler, "An analysis of latent sector errors in disk drives," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2007, pp. 289–300.
[4] Y. Chen, W. W. Hsu, and H. C. Young, "Logging RAID—An approach to fast, reliable, and low-cost disk arrays," in *Proc. 6th Int. Euro-Par Conf. Parallel Process.*, 2000, pp. 1302–1312.
[5] E. N. Elnozahy and J. S. Plank, "Checkpointing for peta-scale systems: A look into the future of practical rollback-recovery," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 2, pp. 97–108, Apr.–Jun. 2004.
[6] L. Ganesh, H. Weatherspoon, M. Balakrishnan, K. Birman, "Optimizing power consumption in large scale storage systems," in *Proc. 11th USENIX Workshop Hot Topics Oper. Syst.*, 2007, p. 9.
[7] B. S. Gill and D. S. Modha, "WOW: Wise ordering for writes—combining spatial and temporal locality in non-volatile caches," in *Proc. 4th USENIX Conf. File Storage Technol.*, 2005, p. 10.
[8] L. Goasduff and C. Forsling. (2006). "Gartner urges it and business leaders to wake up to it's energy crisis," [Online]. Available: http://www.gartner.com/it/page.jsp?id=496819
[9] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, H. Franke, "DRPM: Dynamic speed control for power management in server class disks," in *Proc. 30th Annu. Int. Symp. Comput. Archit.*, 2003, pp. 169–181.
[10] S. Gurumurthi, J. Zhang, A. Sivasubramaniam, M. Kandemir, H. Franke, N. Vijaykrishnan, and M. J. Irwin, "Interplay of energy and performance for disk arrays running transaction processing workloads," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, 2003, pp. 123–132.
[11] Y. Hu and Q. Yang, "DCD—Disk caching disk: A new approach for boosting I/O performance," in *Proc. 23rd Annu. Int. Symp. Comput. Archit.*, 1996, pp. 169–178.
[12] H. Huang, W. Hung, and K. G. Shin, "FS2: Dynamic data replication in free disk space for improving disk performance and energy consumption," in *Proc. 20th ACM Symp. Oper. Syst. Principles*, 2005, pp. 263–276.
[13] B. Schroeder and G. A. Gibson, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?" in *Proc. 5th USENIX Conf. File Storage Technol.*, 2007, p. 1.
[14] M. Levin. (2006). Storage management disciplines are declining. [Online]. Available: http://www.computereconomics.com/article.cfm?id=1129
[15] B. Mao, D. Feng, H. Jiang, S. Wu, J. Chen, and L. Zeng, "GRAID: A green RAID storage architecture with improved energy efficiency and reliability," in *Proc. IEEE Int. Symp. Model., Anal. Simul. Comput. Telecommun. Syst.*, 2008, pp. 1–8.
[16] J. Menon, "A performance comparison of RAID-5 and log-structured arrays," in *Proc. 4th IEEE Int. Symp. High Perform. Distrib. Comput.*, 1995, p. 167.
[17] N. Mi, A. Riska, Q. Zhang, E. Smirni, and E. Riedel, "Efficient management of idleness in storage systems," *ACM Trans. Storage*, vol. 5, no. 2, p. 4, 2009.
[18] J. C. Mogul, "A better update policy," in *Proc. USENIX Tech. Conf.*, 1994, p. 7.
[19] Y. J. Nam and C. Park, "An adaptive high-low water mark destage algorithm for cached RAID5," in *Proc. Pacific Rim Int. Symp. Dependable Comput.*, 2002, p. 177.

[20] D. Narayanan, A. Donnelly, and A. Rowstron, "Write off-loading: Practical power management for enterprise storage," presented at the *6th USENIX Conf. File Storage Technol.*, San Jose, CA, USA, 2008.
[21] M. Rosenblum and J. K. Ousterhout, "The design and implementation of a log-structured file system," *ACM Trans. Comput. Syst.*, vol. 10, no. 1, pp. 26–52, 1992.
[22] S. Sankar, S. Gurumurthi, and M. R. Stan, "Intra-disk parallelism: An idea whose time has come," in *Proc. 35th Annu. Int. Symp. Comput. Archit.*, 2008, pp. 303–314.
[23] P. Steege. (2009). 50% storage utilization: Are data centers half empty or half full?. [Online]. Available: http://storageeffect.media.seagate.com/2009/0-1/storage-effect/50-storage-utilization-are-datacenters-half-empty-or-half-full/
[24] D. Stodolsky, G. A. Gibson, and M. Holland, "Parity logging overcoming the small write problem in redundant disk arrays," in *Proc. 20th Annu. Int. Symp. Comput. Archit.*, 1993, pp. 64–75.
[25] A. Varma and Q. Jacobson, "Destage algorithms for disk arrays with nonvolatile caches," *IEEE Trans. Comput.*, vol. 47, no. 2, pp. 228–235, Feb. 1998.
[26] C. Weddle, M. Oldham, J. Qian, and A.-I. Andy Wang, "PARAID: A gear-shifting power-aware RAID," *ACM Trans. Storage*, vol. 3, no. 3, p. 13, 2007.
[27] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes, "Hibernator: Helping disk arrays sleep through the winter," in *Proc. 20th ACM Symp. Oper. Syst. Principles*, 2005, pp. 177–190.
[28] Q. Zhu and Y. Zhou, "Power-aware storage cache management," *IEEE Trans. Comput.*, vol. 54, no. 5, pp. 587–602, May 2005.
[29] Y. Yue, L. Tian, H. Jiang, F. Wang, D. Feng, Q. Zhang, and P. Zeng, "RoLo: A rotated logging storage architecture for enterprise data centers," in *Proc. IEEE 30th Int. Conf. Distrib. Comput. Syst.*, 2010, pp. 293–304.
[30] F. Wang, Y. Yue, Q. Zhang, L. Tian, and D. Feng, "Reliability modeling of energy efficient logging architectures based on RAID10 systems," in *Proc. 5th Int. Conf. Frontier Comput. Sci. Technol.*, 2010, pp. 57–62.
[31] A. Hylick, R. Sohan, A. C. Rice, and B. Jones, "An analysis of hard drive energy consumption," in *Proc. IEEE Int. Symp. Model., Anal. Simul. Comput. Telecommun. Syst.*, 2008, pp. 1–10.
[32] D. Essary and A. Amer, "Predictive data grouping: Defining the bounds of energy and latency reduction through predictive data grouping and replication," *ACM Trans. Storage*, vol. 4, no. 1, pp. 1–23, 2008.
[33] J. G. Elerath and M. Pecht, "Enhanced reliability modeling of RAID storage systems," in *Proc. 37th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, 2007, pp. 175–184.
[34] J. Paris and D. D. E. Long, "Using device diversity to protect data against batch-correlated disk failures," in *Proc. 2nd ACM Workshop Storage Security Survivability*, 2006, pp. 47–52.
[35] A. Krioukov, L. N. Bairavasundaram, G. R. Goodson, K. Srinivasan, R. Thelen, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Parity lost and parity regained," in *Proc. 6th USENIX Conf. File Storage Technol.*, 2008, p. 9.
[36] I. Iliadis, R. Haas, X.-Y. Hu, and E. Eleftheriou, "Disk scrubbing versus intra-diskredundancy for high-reliability raid storage systems," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2008, pp. 241–252.
[37] M. Balakrishnan, D. Malkhi, T. Wobber, M. Wu, V. Prabhakaran, M. Wei, J. D. Davis, S. Rao, T. Zou, and A. Zuck, "Tango: Distributed data structures over a shared log," in *Proc. 24th ACM Symp. Operating Systems Principles*, 2003, pp. 325–340.
[38] M. Balakrishnan, D. Malkhi, V. Prabhakaran, T. Wobber, M. Wei, and J. Davis, "CORFU: A shared log design for flash clusters," in *Proc. 9th USENIX Conf. Netw. Syst. Des. Implementation*, 2012, p. 1.
[39] P. A. Bernstein, C. W. Reid, and S. Das, "Hyder-a transactional record manager for shared flash," in *Proc. Conf. Innovative Data Syst. Res.*, 2011, pp. 9–20.
[40] H. T. Vo, S. Wang, D. Agrawal, G. Chen, and B. C. Ooi, "LogBase: A scalable log-structured database system in the cloud," *Proc. VLDB Endowment*, vol. 5, pp. 1004–1015, 2012.
[41] B. He, J. X. Yu, and A. C. Zhou, "Improving update-intensive workloads on flash disks through exploiting multi-chip parallelism," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 152-162, Jan. 2015.
[42] S. T. On, Y. Li, B. He, M. Wu, Q. Luo, and J. Xu, "FD-Buffer: A cost-based adaptive buffer replacement algorithm for flashmemory devices," *IEEE Trans. Comput.*, vol. 63, no. 9, pp. 2288–2301, Sep. 2014.

Yinliang Yue received the bachelor's degree in computer science from the Harbin Institute of Technology, China, in 2005, and the PhD degree in computer architecture from the Huazhong University of Science and Technology, China, in 2011. He is currently an associate professor at the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include computer architecture and storage systems.

Bingsheng He received the bachelor's degree in computer science from Shanghai Jiao Tong University (1999-2003), and the PhD degree in computer science from the Hong Kong University of Science and Technology (2003-2008). He is currently an assistant professor at Nanyang Technological University, Singapore. His research interests include high-performance computing, distributed and parallel systems, and database systems.

Lei Tian received the PhD degree in computer engineering from the Huazhong University of Science and Technology in 2010. He is a senior member of Technical Staff at Tintri. Prior to join Tintri, he was a research assistant professor at the Department of Computer Science and Engineering, University of Nebraska-Lincoln. His research interests include storage systems, distributed systems, cloud computing and big data.

Hong Jiang received the PhD degree in computer science from the Texas A&M University in 1991. He is currently a Willa Cather Professor of computer science and engineering at the University of Nebraska-Lincoln. His research interests include computer storage systems and parallel I/O, big data computing, and cloud computing. He recently served as an associate editor of the *IEEE Transactions on Parallel and Distributed Systems*.

Fang Wang received the BE and master's degrees in computer science and the PhD degree in computer architecture from the Huazhong University of Science and Technology (HUST), China. She is a professor of computer science and engineering at HUST. Her interests include distribute file systems, parallel I/O storage systems, and graph processing systems.

Dan Feng received the BE, ME, and PhD degrees in computer science and technology from the Huazhong University of Science and Technology, China, in 1991, 1994, and 1997, respectively. She is currently a professor and director of Data Storage System Division, Wuhan National Lab for Optoelectronics. Her research interests include massive storage systems, parallel file systems, disk array, and solid state disk.