# Online Availability Upgrades for Parity-Based RAIDs through Supplementary Parity Augmentations

LEI TIAN, Huazhong University of Science and Technology/University of Nebraska-Lincoln
QIANG CAO
HONG JIANG, University of Nebraska-Lincoln
DAN FENG and CHANGSHENG XIE, Huazhong University of Science and Technology
QIN XIN



In this article, we propose a simple but powerful online availability upgrade mechanism, *S*upplementary *P*arity *A*ugmentations(*SPA*), to address the availability issue in parity-based RAID systems. The basic idea of SPA is to store and update the supplementary parity units on one or a few newly augmented spare disks for online RAID systems in the operational mode, thus achieving the goals of improving the reconstruction performance while tolerating multiple disk failures and latent sector errors simultaneously. By applying the exclusive OR operations appropriately among supplementary parity, full parity, and data units, SPA can reconstruct the data on the failed disks with a fraction of the original overhead that is proportional to the supplementary parity coverage, thus significantly reducing the overhead of data regeneration and decreasing recovery time in parity-based RAID systems. Our extensive trace-driven simulation study shows that SPA can significantly improve the reconstruction performance of the RAID5 and RAID5+0 systems, at an acceptable performance overhead imposed in the operational mode. Moreover, our reliability analytical modeling and sequential Monte-Carlo simulation demonstrate that SPA is consistently more than double the MTTDL of the RAID5 system and improves the reliability of the RAID5+0 system noticeably.

Categories and Subject Descriptors: C.4 [**Performance of Systems**]: Reliability, availability, and serviceability

General Terms: Design, Reliability

Additional Key Words and Phrases: Availability upgrade, RAID, RAID reconstruction

**ACM Reference Format:**
Tian, L., Cao, Q., Jiang, H., Feng, D., Xie, C., and Xin, Q. 2011. Online availability upgrades for parity-based RAIDs through supplementary parity augmentations. ACM Trans. Storage 6, 4, Article 17 (May 2011), 23 pages.
DOI = 10.1145/1970338.1970341 http://doi.acm.org/10.1145/1970338.1970341

This work was supported by the U.S. NSF under grants CNS-1016609, IIS-0916859, CCF-0937993, and CCF-0621526, the National Basic Research Program of China ("973" program) under grants nos. 2011CB302301 and 2011CB302303, the National High Technology Research and Development Program of China ("863" program) under grants nos. 2009AA01AF401 and 2009AA01A402, Changjiang Innovative Group of Education of China no. IRT0725, and the National NSF of China (NSFC) under grants nos. 61025008, 60703046, 60873028, and 60933002.
Authors' addresses: L. Tian, Department of Computer Science and Engineering, University of Nebraska-Lincoln , Lincoln, NE 60588 and School of Computer Science and Technology, Huazhong University Science and Technology, China; H. Jiang, Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 60588; D. Feng (corresponding author), C. Xie, School of Computer Science and Technology, Huazhong University of Science and Technology, China; email: dfeng@hust.edu.cn.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2011 ACM 1553-3077/2011/05-ART17 $10.00
DOI 10.1145/1970338.1970341 http://doi.acm.org/10.1145/1970338.1970341

## 1. INTRODUCTION

In this article, we try to answer a simple yet intriguing question: By augmenting new spare disks to a RAID [Patterson et al. 1988] system, can we perform an online and flexible system upgrade to improve the RAID system availability in a way analogous to conventional online RAID storage system capacity upgrades that expand capacity and improve I/O parallelism and reliability?

In today's data centers, due to the increasing needs for system maintenance, such as replacing defected components, enhancing system performance, and expanding data capacity, data servers and storage subsystems routinely experience system upgrades [Nathuji et al. 2007]. A recent study shows that 90% of large data centers are expected to upgrade their computing and storage infrastructure in the next two years. This trend has shortened upgrade cycles to be less than two years, as a result of ever more stringent demands on performance, reliability, power efficiency, and ease of management [Heath et al. 2005]. Correspondingly, most RAID manufacturers have provided online upgrade mechanisms in their RAID products. For example, Online Capacity Expansion(OCE) [Techtarget 2008], which expands the storage capacity online, and Online RAID-Level Migration(ORLM) [Techtarget 2008], which changes the RAID-level online by augmenting new disks, respectively, offer larger storage capacity, higher I/O parallelism, and higher reliability.

However, the question of how to upgrade RAID's availability in production data centers by augmenting new spare disks online, while interesting and arguably important, remains unanswered yet. The latest findings and observations from the real world by researchers [Schroeder and Gibson 2007; Pinheiro et al. 2007] have reported that disk failures and error rates are actually much higher than previously and commonly estimated, which suggests an urgent need to significantly improve the availability of RAID systems. Recently, Jiang et al. [2008] analyzed the storage logs covering 44 months and including 1.8 million disks from about $39,000$ storage systems, and concluded that while the annual disk failure rate is about 0.9%, it still contributes to $20-55\%$ of storage subsystem failures. Besides complete disk failures, Bairavasundaram et al. [2007] analyzed the trend of latent sector errors in the same dataset over 32 months across 1.53 million drives, and found that 3.45% of these disks developed latent sector errors.

More importantly, frequent occurrences of disk failures or latent sector errors present a serious challenge to meeting the requirements in certain Service-Level Agreements (SLA) between storage service providers and their clients (end users). SLA commits service providers to a required level of service which often specifies the percentage of time when services must be available, latency per transaction, and so on. Clients pay service fees to obtain their expected services according to the performance/cost ratio and their budgets. If the service is delivered as specified, it is said to be in the service accomplishment state in SLA. If service providers violate the guaranteed performance of SLAs with unexpected down time or higher latency (i.e., in the service interruption state in SLA), they usually have to be penalized economically, typically by a reduction in fees plus some additional compensation and a corrective action plan.

Although RAID systems in production environments tend to utilize extra disk drives to accommodate peak workloads and deliver guaranteed performance to users in the operational mode, hardware or software faults can force these RAID systems to switch from the operational mode to the degraded mode and then to the recovery mode, in which the delivered performance can be significantly reduced due to the I/O-intensive recovery process. Worse still, clients will tend to consider these unexpectedly long response times as transient downtime events from users' perspectives even if the services are still available. From the viewpoint of SLA, a transition from the service accomplishment state to the service interruption state is indeed considered a failure.

In this article, we propose a simple but powerful approach, *S*upplementary *P*arity *A*ugmentation (*SPA*), to upgrade the availability of standard parity-based RAID systems in production data centers online and flexibly. The basic idea behind SPA is to store and update the supplementary parity units on the newly augmented spare disk(s) in the operational mode to achieve the goals of tolerating multiple disk failures and latent sector errors and improving the recovery performance upon a disk failure with an acceptable performance and space cost in the operational mode. In particular, SPA has two partial-parity coverage orientations, SPA Vertical and SPA Diagonal, that cater to the user's different availability needs. The former, which calculates the supplementary parity of a fixed subset of the disks, can tolerate more disk failures and sector errors, whereas the latter shifts the coverage of supplementary parity by one disk for each stripe to balance the workload and thus maximize the performance of reconstruction during recovery. Similar to the storage upgrade mechanisms of RAIDs, such as OCE [Techtarget 2008] and ORLM [Techtarget 2008], SPA can be very flexibly enabled or disabled on demand. More importantly, SPA can be enabled or disabled without requiring any data reorganization on the original data layout of RAID systems.

It must be noted that augmenting one single additional parity disk by SPA to cover a subset of disks in a RAID5 disk array can be considered as a variant of RAID5+0. However, SPA is significantly different from and advantageous over RAID5+0 in the following fundamental ways. Compared with RAID5+0, SPA is much easier to add to a RAID5 system online without any change to the original data layout and it can be executed in an asynchronous mode. Furthermore, SPA Diagonal achieves better reconstruction performance than RAID5+0 while SPA Vertical improves the system reliability of RAID5+0. On the other hand, the proposed SPA, with a more efficient reconstruction mechanism, is designed to strike a sensible trade-off between recovery performance and reliability that lies somewhere between RAID5 and RAID6. In other words, SPA significantly improves the performance of both RAID5 and RAID6 during single-failure recovery and the fault tolerance of RAID5, but at the expense of offering lower reliability than RAID6. The rationale behind this trade-off is that single-disk failures are the most common case (substantially more so than double failures) for high-availability storage systems while the performance during recovery is of critical importance in meeting SLA requirements in data centers. Furthermore, commonly used approaches such as data scrubbing [Schwarz et al. 2004] and intra-disk redundancy [Dholakia et al. 2008; Iliadis et al. 2008] can be easily incorporated to SPA to detect and recover from latent sector errors in the operational or recovery mode, thus mitigating the necessity of recovering from double failures for which the RAID6 codes are designed to address.

Our extensive trace-driven simulation results demonstrate that SPA can significantly improve the recovery performance upon disk failures. The SPA Diagonal approach is shown to improve the average user response times during recovery of RAID5, RAID5+0, RAID6, and SPA Vertical by up to 95%, 92%, 95%, 93%, respectively, while improving their respective reconstruction times by up to 36%, 29%, 37%, and 34%. Furthermore, reliability analytical modeling and sequential Monte-Carlo simulation demonstrate that both SPA orientations consistently more than double the MTTDL of the RAID5 system and improve the reliability of the RAID5+0 system noticeably.

The rest of this article is organized as follows. Background and motivations are presented in Section 2. In Section 3, we describe the SPA approach and its implementation in details, and point out its distinctive features from existing works. Performance results through extensive trace-driven simulations and reliability evaluations through analytical and simulation modeling are discussed and presented in Section 4. We conclude the article in Section 5.

## 2. BACKGROUND AND MOTIVATIONS

### 2.1. Background and Related Work

In general, RAIDs can tolerate one or more disk failures. A RAID operates in one of the following three modes: the *operational mode* when there is no disk failure, the *degraded mode* when one or more disk drives fail while the disk array continues to serve the I/O requests with a performance degradation and risk of data loss, and the *recovery mode* when the disk array is rebuilding the data on the failed disk(s) onto the replacement disk(s) in the background upon disk failure(s). After all the data units are rebuilt, the disk array returns to the operational mode. The period when the disk array is in the degraded or recovery mode is called a "window of vulnerability" because additional disk failures or even a few unrecoverable latent sector errors during this time will cause data loss.

In large-scale RAID-structured data centers composed of tens of thousands of hard drives, multiple concurrent data reconstructions will become common due to the increasingly more frequent disk failures [Gibson 2007]. The degraded performance during recovery contributes to lengthening the response time to the end users, thus likely violating the guaranteed performance specified in SLA and unacceptable to the users. Furthermore, data loss caused by the additional disk failures or latent sector errors during recovery is obviously unacceptable to the end users.

One way to avoid data loss is to tolerate additional disk failures or latent sector errors within the window of vulnerability, while an alternative is to narrow the window of vulnerability by reducing recovery time. Double-parity encoding mechanisms known as the RAID6 level, such as the Reed-Solomon code [Plank 1997], EVENODD code [Blaum et al. 1995], Row-Diagonal Parity (RDP) code [Corbett et al. 2004], and Liberation Codes [Plank 2008], are proposed to tolerate a second disk failure. All these schemes are able to survive and recover from any double-disk failures, but at the cost of notable performance penalty because each write during the operational mode requires two corresponding parity updates on different disks for RAID6. In addition to these solutions, disk scrubbing [Schwarz et al. 2004] and Interleaved Parity Check (IPC) [Dholakia et al. 2008; Iliadis et al. 2008] are proposed to detect or tolerate latent sector errors. Disk scrubbing is an error detection method to scan all disk media in the background to detect latent sector errors. As a new form of intra-disk redundancy, IPC is an error recovery method that adds an additional redundancy level on top of the RAID redundancy across multiple disks by adding parity of segments of blocks in each disk to recover from latent sector errors.

To reduce reconstruction time, a live-block recovery approach in D-GRAID [Sivathanu et al. 2004] rebuilds only those data blocks that are considered live from the perspective of file systems and databases. PRO [Tian et al. 2007] deploys a popularity-based multithreaded scheduling algorithm to rebuild the frequently accessed areas prior to rebuilding infrequently accessed areas to exploit access locality and sequentiality. Workout [Wu et al. 2009] outsources all write requests and popular read requests originally targeted at the degraded RAID set to a surrogate RAID set during reconstruction. MICRO [Xie and Wang 2008] collaboratively leverages storage cache and RAID controller cache to diminish the number of disks accesses for recovery purposes. CORE [Xie and Sharma 2009] is proposed for hybrid disk arrays to improve recovery performance by performing the reconstruction process collaborated by hard disks and flash memory SSDs. Parity declustering [Holland et al. 1993] reduces the additional

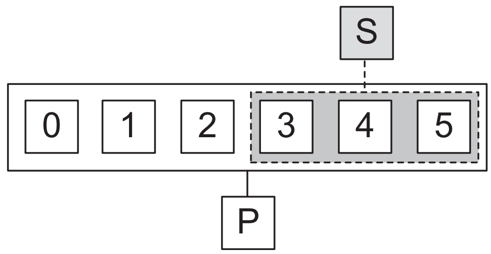



Fig. 1. An example of a full parity unit $P$ and a partial parity unit $S$.

load on survival disks during recovery by distributing small parity groups over a larger number of disks.

### 2.2. Motivations

With rapid advances in the hard disk technology, hard drives have seen their capacity increasing while cost decreases drastically [Anderson and Whittington 2007]. As a result, dedicating a number of spare disks for the sake of availability is no longer a significant cost or resource concern for a large-scale data center. RAID systems in data centers usually have multiple dedicated disks as global or local hot spare disks for their multiple RAID sets. It is thus sensible to trade the capacity and bandwidth of these spare disks for higher system performance, reliability, and availability.

Similarly, workloads of user applications have broadly exhibited a fluctuating property [Weddle et al. 2007; Narayanan et al. 2008], meaning that during the working hours, user workloads tend to be heavy while becoming relatively light during the off hours. Even during the busy times, bursty access patterns have been consistently observed giving rise to many idle periods between I/O bursts [Golding et al. 1995]. Leveraging the idle or lightly loaded periods has been a common practice to enhance performance, reliability, and availability of storage systems [Mi et al. 2008; Riska and Riedel 2008].

Furthermore, the exclusive Or(XOR) calculation is widely used in parity-encoded RAID systems. A parity is regenerated by XORing all the data units in a stripe, so that any data unit can be reconstructed by XORing all the remaining data units in the stripe and the parity unit, referred to as the $P$ parity, as shown in Figure 1.

On the other hand, we observe that, given a subparity, referred to as the $S$ parity, that covers one half or a portion of the data units, any data unit inside $S$'s coverage (the shaded rectangle in Figure 1) can be regenerated by recomputing all the remaining data units inside $S$'s coverage and the $S$ parity unit. At the same time, any data unit outside $S$'s coverage (the hollow rectangle in Figure 1) can be regenerated by recomputing all the remaining data units outside $S$'s coverage and both the $P$ and $S$ parity units. It indicates that if a supplementary parity can be augmented to a parity group (as in standard RAID4/RAID5 levels), approximately half the data reading operations and half the XOR calculations can be avoided during the recovery process. In other words, the overhead of regenerating the lost data on the failed disk can be nearly halved.

Inspired by the preceding observations as well as the performance upgrade mechanisms widely integrated into RAID products such as OCE and ORLM, we propose an online availability upgrade mechanism, SPA, by exploiting an additional level of redundancy on top of the existing parity-based redundancy such as RAID5 across multiple component disks in the form of supplementary spare disks.

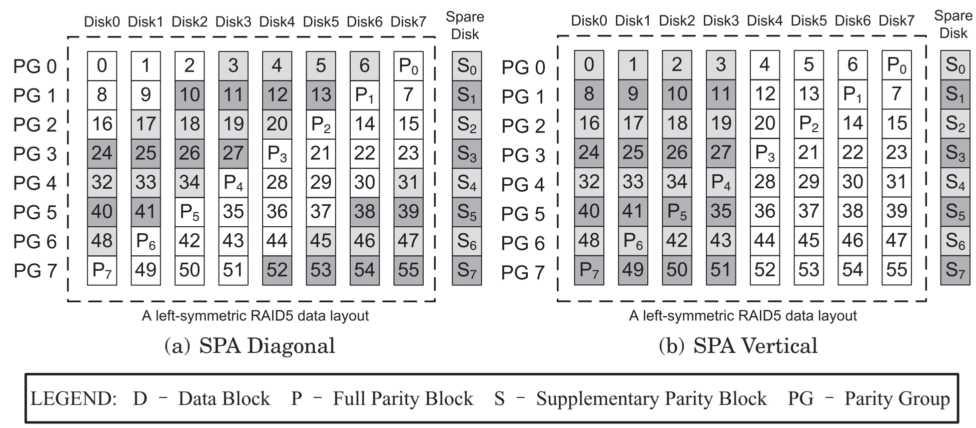



Fig. 2. Two examples of SPA coverage and layout. Figure 2(a) depicts an SPA coverage with *Diagonal orientation*, and Figure 2(b) depicts an SPA coverage with *Vertical orientation*.

## 3. SUPPLEMENTARY PARITY AUGMENTATION

### 3.1. The Basic SPA Idea

As Figure 2(a) shows, given a RAID5 left-symmetric disk array [Lee and Katz 1993] consisting of eight disks, SPA employs a dedicated hot-spare disk to store supplementary parity units that constitute an additional level of redundancy over RAID5 for the availability upgrade.

More specifically, the supplementary parity units from $S_0$ through $S_7$ are calculated to cover half of the disks per parity group (the shaded rectangles in Figure 2).

Assume that one disk, say, Disk 4, fails at some point, we can regenerate the data or parity units on Disk 4 as follows.

For any unit on the failed disk that is covered by the supplementary parity unit $S$ for the corresponding parity group, it can be regenerated by XORing all the remaining data units in $S$'s coverage and $S$ itself for the same parity group. For example,

$$D_4 = D_3 \oplus D_5 \oplus D_6 \oplus S_0. \tag{1}$$

On the other hand, for any full parity (or any data) unit on the failed disk that is not covered by supplementary parity unit $S$ for the corresponding parity group, it can be regenerated by XORing all the remaining data units outside $S$'s coverage and $S$ itself, as well as the full parity unit $P$. For example,

$$D_{28} = P_4 \oplus D_{29} \oplus D_{30} \oplus S_4. \tag{2}$$

Because SPA in this example halves the number of read and XOR operations, it also avoids the negative performance impact of data reconstruction (and reads) on disks that are spared of the recovery intrusion, which is particularly important for user I/O requests under heavy workloads. As a result, SPA can reduce disk bandwidth utilization due to reconstruction, shorten disk I/O queues, mitigate I/O bus bottlenecks, and lower CPU utilization during failure recovery.

There are two forms of supplementary parity distribution: supplementary parity with *Diagonal orientation* (as shown in Figure 2(a)) and supplementary parity with *Vertical orientation* (as shown in Figure 2(b)). From the figure, one can see that SPA with Diagonal orientation implies that units covered by SPA are distributed diagonally. On the other hand, SPA with Vertical orientation signifies that units covered by SPA are distributed among a fixed subset of disks.

The advantage of Diagonal orientation is its ability to balance recovery workload among all the surviving disks, but at the cost of not being able to tolerate a subsequent disk failure during recovery. On the other hand, Vertical orientation can tolerate another disk failure during recovery if exactly one of the two failed disks is covered by SPA, a fault-tolerant ability that is similar to that of RAID5+0. An additional advantage of Vertical orientation is its potential for covering a number of designated disks that may have higher failure rates than the rest. The drawback of this coverage orientation lies in the imbalanced recovery workload.

Assume that one disk in Figure 2(b), say, Disk 0, fails at some point, we can regenerate any data or parity unit on Disk 0 by XORing all the remaining units inside $S$'s coverage and $S$ itself for the same parity group. For example,

$$D_0 = D_1 \oplus D_2 \oplus D_3 \oplus S_0. \tag{3}$$

Assume that another disk in Figure 2(b), say, Disk 4, subsequently fails, we can regenerate any data or parity units on Disk 4 by XORing all the remaining units outside $S$'s coverage and $S$ itself for the same parity group. For example,

$$D_4 = D_5 \oplus D_6 \oplus P_0 \oplus S_0. \tag{4}$$

Although it is intuitively understandable to obtain the SPA idea from the previous figure and corresponding explanation, a general form of the SPA is still very useful to apply SPA to various parity-encoded RAID organizations. We demonstrate an exemplar way to generalize the SPA in the following.

Given a left-symmetric RAID5 disk array with an even number of $N$ disks, we first construct an $N \times N$ table, in which each entry denotes a data unit or a full parity unit. A unit can be denoted by its disk number (denoted as $i$, $0 \le i < N$) and its parity group number (denoted as $j$, $0 \le j < N$) as $U_j^i$. The full parity unit $P$ for a parity group of $j$, $P_j = U_j^{N-j-1}$. A data unit of no. $k$ is denoted as $D_k$, where $0 \le k < N \times (N-1)$. An SPA parity unit $S$ for a parity group $j$ is denoted as $S_j$.

Second, we can generate the distribution of data units according to the data layout of a left-symmetric RAID5, as shown in Figure 2. For example, $D_0 = U_0^0$, $D_7 = U_1^7$, $D_{55} = U_7^7$.

For the SPA with Diagonal orientation for a parity group of $j$, we have

$$S_j = \underbrace{D_j^{j \times (N-1) + \frac{N}{2} - 1} \oplus \cdots \oplus D_j^{(j+1) \times (N-1) - 1}}_{\frac{N}{2}}.$$

For the SPA with Vertical orientation for a parity group of $j$, we have

$$S_j = \underbrace{U_j^0 \oplus \cdots \oplus U_j^{\frac{N}{2}-1}}_{\frac{N}{2}}.$$

From the viewpoint of recovery, SPA Diagonal can be considered a variant of parity declustering by distributing small parity groups evenly over a larger number of disks, while SPA Vertical can be considered a variant of RAID5+0 by converting one RAID5 set into two or more smaller isolated RAID5 sets.

Moreover, regardless of its coverage orientation, SPA has an inherent capability to conditionally tolerate and recover from unrecoverable latent sector errors during the disk failure recovery. This is because for each parity group, the full parity $P$ and the supplementary parity $S$ constitute two isolated parity subgroups, in which the units in one subgroup are protected by $S$, and the units in the other subgroup are independently

Table I. A Comparison of Relevant Schemes from the Perspective of Availability Upgrades for RAID Systems

| Schemes | Applicable to RAID5 level | Applicable to RAID6 level | Needs data layout reorganization | Improves the recovery performance | The extra parity update policy |
|---|---|---|---|---|---|
| Parity Declustering | Yes | No | Yes | High | Sync. |
| IPC | Yes | Yes | Yes | No | Sync./Async. |
| RAID5+0 | Yes | No | Yes | Medium | Sync. |
| RAID6 | Yes | N/A | No* | No | Sync./Async. |
| SPA Diagonal | Yes | Yes | No | High | Sync./Async. |
| SPA Vertical | Yes | Yes | No | Medium | Sync./Async. |

* A RAID5 system can be upgraded to become a RAID6 system without data layout reorganization by using one extra disk as a dedicated second parity disk. However, this will lead to a noticeable performance drop during recovery for write-intensive workloads due to load imbalance.

protected by $P \oplus S$. Therefore, any two simultaneous unrecoverable errors respectively and exclusively occurring in two subgroups can be tolerated and recovered.

Note that the SPA disk also has the probability of a disk failure. However, since SPA disk only stores partial parity units exclusively, it is preferable to reconstruct the content of the SPA disk to a hot-spare disk in the background during idle or very lightly loaded periods. It must also be noted that our SPA approaches will have a negative effect on the reliability of the SPA disk because SPA keeps it active while its non-SPA hot-spare counterparts sit idle. In other words, SPA trades the longevity of hot-spare disks for system reliability, an acceptable trade-off given the trend of ever-decreasing cost of hard drives [Anderson and Whittington 2007].

Obviously, the efficiency of this capability of conditionally tolerating and recovering from unrecoverable latent sector errors depends on the occurrence locations of such errors, thus it is not comparable to the RAID6 or IPC system that can tolerate and recover from unrecoverable errors occurring anywhere in the disk array. However, neither the RAID6 nor the IPC system is capable of improving the reconstruction performance during recovery, one of the main design goals of SPA. Equally important, it is easy to augment a RAID6 system with SPA or integrate the IPC approach into SPA.

### 3.2. Distinctive Features of SPA

Although the set of existing reliability mechanisms described in Section 2.1 is by no means complete, we believe that they are the most representative and closely relevant to our work. The main difference between our SPA and the preceding approaches lies in SPA's design principles and goals. SPA is a supplementary redundancy approach designed as an online availability upgrade mechanism for parity-based RAIDs in production data centers, especially for RAID systems lacking sufficient protection mechanisms, by augmenting new spare disks without any data layout change to the designated RAID set.

Therefore, SPA aims to alleviate the performance degradation, shorten the reconstruction time, and tolerate additional disk failures or unrecoverable errors, thus minimizing the risk of violating SLA for data centers due to disk failures or latent sector errors. Our design philosophy hence underlines the main distinctions between SPA and other availability-enhancing approaches, as summarized in Table I.

Table I illustrates the distinctive features of the two SPA approaches, SPA Diagonal and SPA Vertical detailed in Section 3, that set them apart from other representative availability-enhancing approaches: RAID6, RAID5+0, parity declustering, and IPC. RAID6 has the best capability of tolerating any two disk failures while IPC has the best capability of tolerating latent sector errors, but neither is able to improve the recovery performance. RAID5+0 improves the recovery performance by converting one

RAID5 set into two or more smaller isolated RAID5 sets, while parity declustering achieves high recovery performance by distributing small parity groups evenly over a larger number of disks. However, neither of them, as a variant of RAID5, can be applied to a RAID6 system. On the other hand, SPA Diagonal can achieve a significant improvement in recovery performance that is similar to parity declustering, while SPA Vertical can achieve a recovery performance that is similar to RAID5+0, as shown in Section 4. More importantly, SPA can be enabled online in a RAID system in a production environment without any data reorganization of the designated RAID set. This should be a very desirable and critical feature required of any online upgrade mechanism, since performing data reorganization during upgrade can risk possible data loss in the event of a power supply or disk failure and severely degrade user performance. In addition, the supplementary nature of SPA parity allows SPA to choose either the asynchronous or synchronous parity update policy according to the application workload characteristics.

Furthermore, SPA can be incorporated on top of the aforementioned parity-based approaches, such as parity declustering, RAID5+0, RAID6, and IPC, as long as the parity generation in these schemes is based on the exclusive or (XOR) calculation.

### 3.3. Design and Implementation Issues

**SPA Parity Update Policy**. When a write request arrives at a disk array with an address that falls outside the coverage of any SPA parity unit, no update is needed to any SPA parity unit. Otherwise, the SPA parity unit covering the address of the write request needs to be updated somehow.

SPA provides two update policies: *synchronous* update and *asynchronous* update. The former policy, which updates the corresponding SPA parity unit at the same time as the write operation, ensures the full validity for each parity group, but incurs performance overhead with write-intensive workloads. However, it may be acceptable since write intensity is generally much lower than read intensity and writes tend to congregate around a relatively small proportion of the storage capacity in typical workloads [Roselli et al. 2000]. It must be noted that a write to a data block on a RAID-5 array incurs two parity update operations: one for the full parity update and one for the SPA parity update. The RAID software must guarantee the consistency of the full parity update and data block write for data recovery purposes. For the SPA parity update, it is also updated with the full parity update, and marked "valid". However, it can be cancelled when a disk failure is detected during the update, and marked "invalid" (not updated yet). The inconsistency between the full parity and corresponding SPA parity will not impact the correctness of data recovery. The only loss is that this invalid SPA parity cannot be used to recover its covered data units on a failed disk during recovery.

The advantage of the latter policy, which postpones updates to supplementary parity units until idle or lightly loaded periods, is its ability to minimize performance degradation due to frequent SPA updates. It must be noted that the occurrence of a disk failure before the completion of asynchronous update has no direct impact on the probability of data loss because SPA only adds an additional redundancy on top of the redundancy of the full parity of parity-encoded protection. During the recovery process, the invalid SPA parity units are not updated so as to concentrate available disk IO bandwidth on recovery. However, it may reduce the benefit of the SPA approach to the recovery performance and data reliability during recovery if some supplementary parity units are invalid at the time of recovery. In general, the amount of such decrease in SPA benefit will be proportional to the amount of invalid SPA parity units. To maximize the "best-effort" benefit provided by SPA, the criteria to determine when and how to perform asynchronous update becomes an important issue. Firstly, idle periods should be fully exploited. Fortunately, idle periods have been widely discovered in personal and enterprise computing environments and thus successfully utilized to execute background maintainable tasks such as IPC parity updates, RAID resynchronization, and disk scrubbing as mentioned in Section 2.2. We can deploy methods similar to those described in Mi et al. [2009] to perform asynchronous update. Second, access locality of workloads, even in busy periods, should and can also be explored. Once users access the data units close to the data units whose associated SPA parity units are waiting for update, an SPA parity update operation can be initiated to regenerate those SPA parity units. By doing so, SPA parity update can be made a low-overhead operation feasible for even the I/O-intensive workloads.

On the other hand, RAID5+0 or parity declustering with an asynchronous update policy tends to have much higher probability of data loss than SPA because each data unit in the former is protected by exactly one parity unit, while in the latter the SPA parity is only supplementary to the full parity that is updated synchronously. As a result, any disk failure in the former will lead to data loss due to staled parity, as shown in AFRAID [Savage and Wilkes 1996]. Therefore, asynchronous update policies are not suitable for RAID5+0 or parity declustering.

*SPA Coverage Range Choice*. Besides the coverage orientation choice of SPA, what proportion of disks in a RAID are covered by SPA, which we refer to as SPA *coverage range*, is also an important design issue. More specifically, SPA coverage range refers to the proportion of the component disks in a parity group in the disk array that are covered by the SPA parity. For example, Figure 2 depicts a *half-parity* coverage range, where the coverage range is $\frac{1}{2}$ since one SPA parity unit covers the data units on half of the component disks for each parity group.

SPA provides a family of design options with different space overhead and system availability trade-offs. For example, the *third-parity* option exploits two spare disks to store supplementary parity for two sets of SPA parity units, with each SPA parity disk exclusively covering units on one-third of the component disks. Third-parity reduces the overhead for data regeneration to nearly one-third of that required by the full parity approach. Additionally and in general, if the Vertical orientation coverage distribution is applied, an $n$th-parity approach can tolerate up to $n$ simultaneous disk failures conditionally.

*Extensibility*. Although the examples given in this article are all based on a RAID5 disk array, SPA can also be easily extended to a RAID6 system. While a RAID6 system has the capability to tolerate any two simultaneous disk failures, its recovery performance is nearly the same as a RAID5 system in the event of a single disk failure. In current RAID6 encoding schemes, the first parity $P$ is the same as the one in the conventional RAID5 level that is based on XOR operations, and the second parity $Q$ of most RAID6 schemes, such as the EVENODD [Blaum et al. 1995], RDP [Corbett et al. 2004], and Liberation codes [Plank 2008], are also based on XOR operations. To the best of our knowledge, only the Reed-Solomon code [Plank 1997] uses Galois field algebra to generate its second parity. Therefore, SPA can also be augmented to a RAID6 system in a production environment, and improve the reconstruction performance by virtue of the unique features of SPA. In most cases, with the exception of the Reed-Solomon code, both the $P$ and $Q$ parity can benefit from the augmentation of two SPA disks, with each being dedicated to one of RAID6's two parity groups exclusively.

Similarly, intra-disk redundancy such as IPC [Dholakia et al. 2008; Iliadis et al. 2008] can also be easily integrated into the SPA approach, to further improve its capability of tolerating and recovering from latent sector errors. Of course, the introduction of IPC within SPA will incur extra parity update overheads since the corresponding IPC parity must be updated whenever there is a write request. Since the goal of this article is to reveal the functionality and efficiency of our SPA, we leave the SPA augmentation on other schemes including the RAID6 and IPC as our future work.

*Flexibility*. As an online availability upgrade approach, SPA can be enabled if new spare disks are augmented to parity-based RAIDs in production environments online, and can be disabled if the spare disks are reclaimed. Because applying SPA does not require any change to the original data layout on RAIDs, data loss is unlikely to occur during the operation of enabling or disabling SPA, which is different from OCE and ORLM.

## 4. PERFORMANCE EVALUATIONS

### 4.1. Evaluation Methodology

We developed an extended version of the DiskSim 4.0 simulator [Bucy et al. 2008] to study the performance impacts of our SPA approaches by first extending DiskSim with two baseline rebuild algorithms, Pipeline Reconstruction (PR) [Lee and Lui 2002] and Disk-Oriented Reconstruction (DOR) [Holland et al. 1993], and then augmenting it with SPA. To the best of our knowledge, DiskSim is the most widely used and accurate simulation tool for storage systems and can be easily configured to simulate a hierarchy of storage components such as disks, buses, controllers, as well as some logical organizations such as mirroring and parity-encoded RAIDs. DiskSim can perform accurate simulations, and its disk models have been validated with hard disk products. DiskSim can output the statistical results of various performance metrics of storage subsystem components, for example, average response time, maximal response time, response time distribution, and standard deviation. The excellent hierarchical infrastructure and extensibility of DiskSim 4.0 make it the best evaluation tool for us to develop the baseline rebuild and SPA submodules onto it.

In DiskSim, the *logorg* (logical organization) module is used to simulate logical data organizations, such as various RAID levels or JBOD. We first implemented and integrated the baseline rebuild submodule into the logorg module. The main functions of the rebuild submodule include:

(1) managing rebuild-related events, such as triggering or stopping the rebuild process;
(2) redirecting users' requests on the failed disk to the survival disks in the event of a disk failure, and responding to users' I/O requests with the data regenerated on-the-fly;
(3) reconstructing the full content of the failed disk to the spare disk while servicing users' requests; and
(4) collecting the statistic results such as reconstruction time and user response time.

In particular, we implemented two common RAID rebuild algorithms: the Pipeline Reconstruction (PR) algorithm and the Disk-Oriented Reconstruction (DOR) algorithm as the baseline rebuild algorithms because they or their variants have been most widely integrated into the hardware or software RAID systems, for example, RAIDframe in NetBSD [NetBSD Foundation 2008] and MD in Linux [Østergaard and Bueso 2004]. The basic idea of DOR is to create a number of processes with each being associated with one disk to absorb the available bandwidth of the disks, while PR pipelines the reconstruction procedure to reduce the extra buffer requirement. The implementation of recovery simulation is based on the understanding and analysis of the source code of the DOR and PR algorithms. We believe that our recovery simulation implementation in DiskSim is reasonable and leave its validation to our future work.

We then implemented and integrated our SPA approach into the logorg module, and made it work together with the rebuild submodule. The main functions of the SPA submodule are:

Table II. Disk and RAID Configuration Parameters

| Disk Parameter | Value |
|---|---|
| Disk Model | Cheetah 15*K*.5 |
| Capacity | 146.8 GB |
| Rotation Speed | 15, 000 RPM |
| Disk Cache Size | 16 MB |
| Average | 2.0 ms |
| **RAID Parameter** | **Value** |
| RAID Level | RAID4 / RAID5 |
| Data Layout Scheme | Left Symmetric |
| Number of Disks | 6, 8, 10 |
| Stripe Unit Size | 32, 64, 128 KB |
| Baseline Reconstruction Algorithm | PR / DOR |

Table III. The Trace Characteristics

| Trace | Write Ratio | IOPS | Req. Size |
|---|---|---|---|
| WebSearch1 | 0% | 334.9 | 15.5 KB |
| Financial1 | 76.8% | 122.0 | 3.4 KB |
| Financial2 | 17.6% | 90.2 | 2.3 KB |

(1) to manage the update policy and handle the SPA parity update upon the arrival of each user's write request in the operational mode (e.g., an SPA parity unit update operation is triggered only if a write request is located inside the coverage of this SPA parity unit);
(2) to assist the rebuild submodule in determining the number and locations of data units which need to be read and XORed according to the SPA configuration in the event of a disk failure.

In addition, we also implemented the RAID5+0 and RAID6 levels to make them work with the rebuild submodule. In particular, we implemented two typical second-parity placement strategies for RAID6. The first, called *RAID6 Rotated*, is to rotate the second-parity unit per stripe among the component disks to evenly distribute second-parity units, while the second, called *RAID6 Fixed*, uses a dedicated disk to store all second-parity units.

In our experiments, a RAID5 disk array with a varying number of component disks and varying stripe size was simulated. For the current study, only one disk failure and its corresponding reconstruction process is simulated. One of the recent disk models with the generation-4 layout, Seagate Cheetah 15*K*.5 [Seagate 2007], was used throughout our experiments, with its main specifications listed in Table II. We applied two types of workloads to the simulator: WebSearch and Financial [UMass Trace Repository 2007] obtained from the Storage Performance Council [Storage Performance Council 2009]. The WebSearch1 ("Web" for short) trace was collected from a system running a popular search engine, while the Financial1 ("Fin1" for short) and Financial2 ("Fin2" for short) traces were collected from OLTP applications running at a large financial institution. These three traces have different read/write ratios, access rates, access sizes, and degrees of sequentiality and locality due to their different application characteristics, which represent typical access patterns in real-world production environments. The traces' key workload characteristics are summarized in Table III.

We only rebuilt 5% of the total capacity of our disk model (about 7.4 GB) in all the following rebuild experiments to save simulation time, since our sample results from the full capacity experiments show that conclusions drawn from both reduced capacity and full capacity are consistent.

The average user response time during recovery and the reconstruction time are the two most important metrics in evaluating the recovery performance for RAID systems.

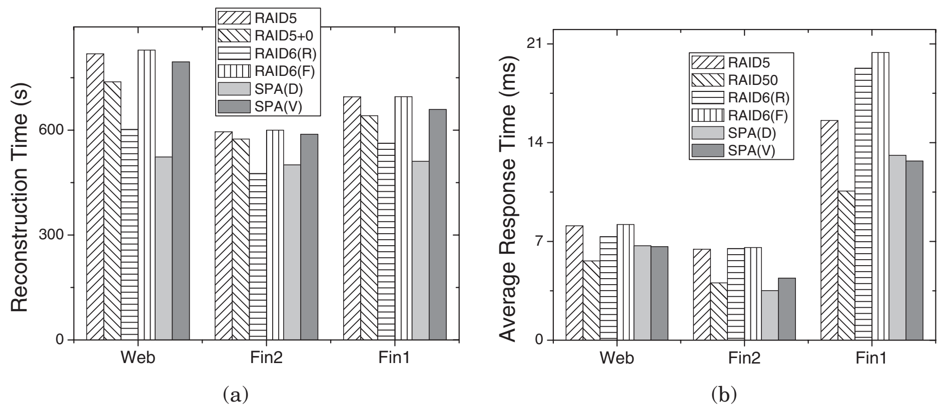



Fig. 3. Comparisons of reconstruction times and average response times of the schemes based on the DOR baseline algorithm.

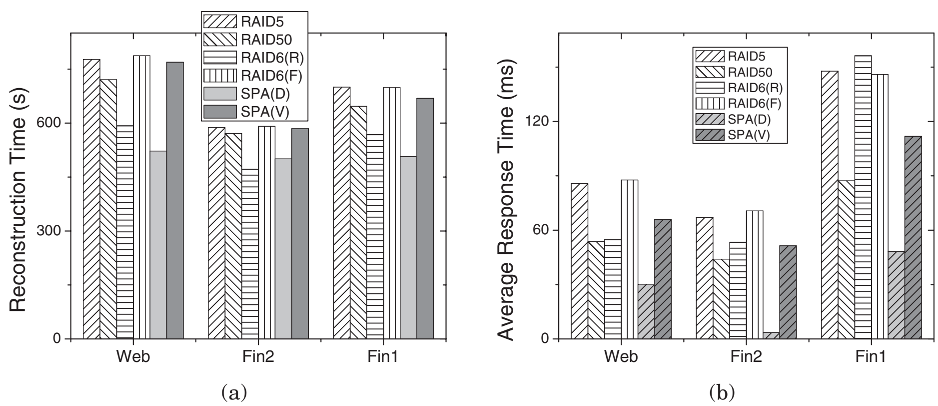



Fig. 4. Comparisons of reconstruction times and average response times of the schemes based on the PR baseline algorithm.

To evaluate the performance impacts of SPA in the operational mode, we also use response time as the performance metric and compare it with the other schemes. For the sake of brevity, the term "response time" will be used in lieu of "average user response time during recovery" in the rest of the article unless otherwise specified. Likewise, we use notations *RAID6(R)*, *RAID6(F)*, *SPA(D)*, and *SPA(V)* to represent RAID6 Rotated, RAID6 Fixed, SPA Diagonal, and SPA Vertical, respectively. The simulation results output by DiskSim are deterministic, that is, each simulation run for the same configuration file gives the same output. Therefore, we only run the DiskSim simulation one time per configuration and report its statistical results.

### 4.2. Experimental Evaluations

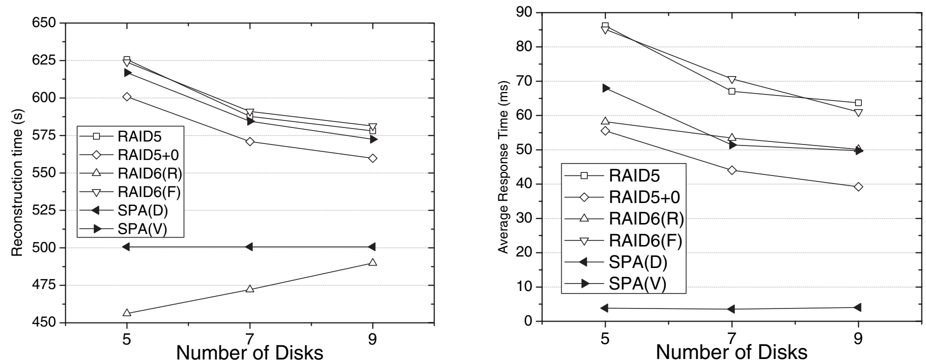
*Recovery Performance Study.* We first conducted our experiments on a RAID5 disk array consisting of seven disks and one hot-spare disk with a stripe unit size of 64KB to evaluate the recovery performance of SPA. We incorporated one spare disk to upgrade a current RAID5 system to one of the following RAID systems of higher availability: RAID5+0, RAID6(R), RAID6(F), SPA(D), or SPA(V) system.

Figure 3 and Figure 4 show the comparisons of reconstruction times and average response times of the schemes under the Web, Fin2, and Fin1 workloads, assuming the DOR and PR baseline rebuild algorithms, respectively. The reason why we only report the preceding results is because the other results we conducted with different traces or baseline rebuild algorithms are comparably consistent with our conclusions. All schemes, except for RAID6(F), outperform the RAID5 scheme in reconstruction time and response time. With the DOR baseline algorithm and under the Web trace, shown in Figure 3, SPA(D) improves the reconstruction time by 36% over RAID5, 29% over RAID5+0, 13% over RAID6(R), 37% over RAID6(F), and 34% over SPA(V), while RAID5+0 achieves the best response time performance. The Fin2 trace results of Figure 3 show that RAID5+0, RAID6(R), and SPA(D) outperform RAID5 in reconstruction time by 3%, 20% and 16% respectively, and in response time by a factor of 37%, −1%, and 45% respectively. The experimental results based on the PR baseline algorithm, shown in Figure 4, indicate similar trends to those based on DOR, except that SPA(D) is shown to have a noticeably more pronounced response-time performance advantage over other schemes. More specifically, SPA(D) outperforms RAID5, RAID5+0, RAID6(R), RAID6(F), and SPA(V) by 65%, 44%, 45%, a factor of 66%, and 54%, respectively under the Web trace, and by 95%, 92%, 93%, 95%, and 93%, respectively, under the Fin2 trace. For the write-intensive Fin1 trace, SPA(D) with the DOR algorithm outperforms RAID5, RAID5+0, RAID6(R), RAID6(F), SPA(V) on reconstruction time performance by 27%, 20%, 9.2%, 27%, 22%, respectively, while SPA(V) outperforms RAID5, RAID6(R), RAID6(F), SPA(D) on response time performance by 18%, 34%, 38%, 3.1%, respectively, but underperforms RAID5+0 by 20%. On the other hand, SPA(D) with the PR algorithm performs the best on both reconstruction time and response time performances. More specifically, SPA(D) outperforms RAID5, RAID5+0, RAID6(R), RAID6(F), SPA(V) on reconstruction time by 28%, 22%, 11%, 27%, 24%, respectively, while on response time by 67%, 45%, 69%, 67%, 57%, respectively.

From the result analysis, one can see that the SPA scheme is much more beneficial to response time performance than to reconstruction time performance. It seems counterintuitive because SPA halves the reconstruction loads and it is expected to translate this into the doubling improvement on the reconstruction time. However, due to I/O parallelism of component disks of RAID, simply reducing reconstruction reads cannot directly contribute to accelerating the recovery process. On the other hand, reducing reconstruction reads can weaken the interference from reconstruction IOs to users' IOs, thus rendering user response time performance being significantly improved.

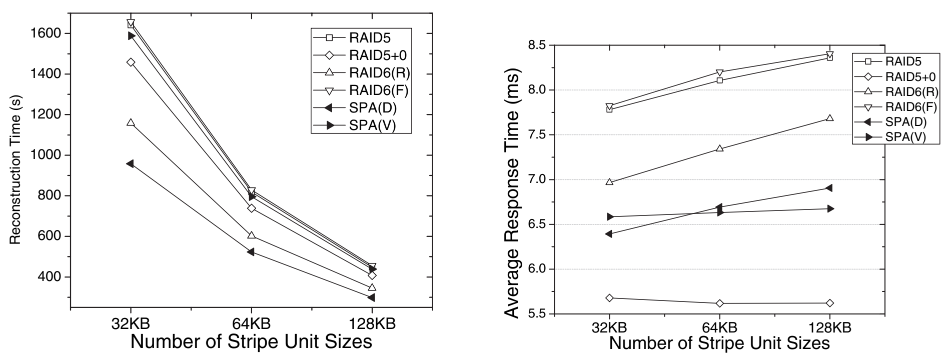
The significant performance advantage of SPA(D) over other schemes stems from the former's ability to halve and evenly distribute the reconstruction workload on all the component disks, and to leverage the bandwidth of the SPA disk in the rebuild. In other words, it mitigates the heavy workload on each disk and reduces the interference from the reconstruction I/O requests, thus successfully minimizing the queuing time for each external I/O request. In contrast, SPA(V) only improves the response time by up to 32% (under Fin2 with DOR) compared with RAID5. This is because SPA(V) does not distribute the reconstruction loads evenly among component disks, so that nearly half of the disks are under the same reconstruction I/O intensity as the baseline system while the other half of the disks have no reconstruction I/O requests at all. The severe load imbalance of the reconstruction I/O causes SPA(V) to underperform SPA(D). On the other hand, RAID6(R) outperforms RAID5+0 in reconstruction time while the opposite is true for response time. It shows that smaller parity groups in RAID5+0 lead to better response time performance.

*Sensitivity Study.* To examine the performance impact of the number of disks, we conducted experiments on a RAID5 disk array composed of a varying number of disks (5, 7, 9) with a fixed stripe unit size of 64KB and one hot-spare disk. As shown in Figure 5, increasing the number of disks by and large shortens the reconstruction time and response time due to the increased disk parallelism. However, one can find




(a) (b)

Fig. 5. Comparisons of reconstruction times and average response times with respect to the number of disks driven by the Fin2 trace. The baseline reconstruction algorithm is PR.




(a) (b)

Fig. 6. Comparisons of reconstruction times and average response times with respect to the numbers of stripe unit sizes driven by the Web trace. The baseline reconstruction algorithm is DOR.

that SPA(D) is insensitive to the change in the number of disks, and consistently outperforms other schemes in response time by a big margin while achieving the second-best reconstruction time performance. Interestingly, RAID6(R)'s advantage in reconstruction time weakens as the number of disks increases, which indicates that SPA(D) may eventually outperform RAID6(R) with a larger number of disks.

To examine the performance impact of the stripe unit size, we conducted experiments on a RAID5 disk array consisting of seven disks and one hot-spare disk with variable stripe unit sizes of 32KB, 64KB, and 128KB. We plot the measured reconstruction times and response times as a function of the stripe unit size in Figure 6. From the figure, we can observe that the recovery performance, especially the reconstruction time, is sensitive to the stripe unit size. Increasing stripe unit size lengthens the response time and shortens the reconstruction time consistently. Similar to Figures 3 and 4, SPA(D) and SPA(V) are shown to consistently improve the baseline RAID5 schemes in both the reconstruction time and response time across all stripe unit sizes. And more importantly, the relative amount of such improvement also remains consistent across all stripe unit sizes. This suggests that SPA is likely to be equally effective when applied to RAID5 of varying stripe unit sizes.

Table IV. SSD Specification Parameters

| Parameter | Value |
|---|---|
| Chip Model | Samsung K9XXG08UXM |
| Page Size | 4KB |
| Block Size | 256KB |
| Chip Size | 2GB |
| Chips Per Package | 8 |
| Page Read | 25$\mu s$ |
| Page Write | 200$\mu s$ |
| Block Erase | 1.5$ms$ |
| Chip Bus | 100$\mu s$ per page |

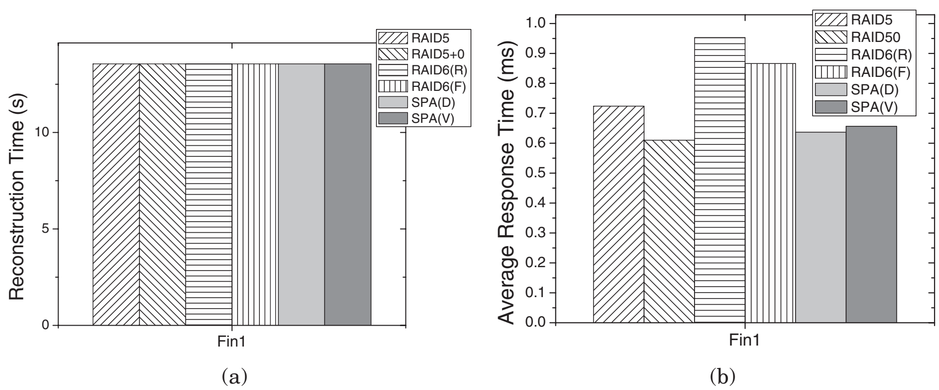



Fig. 7. Reconstruction time and average response time for an SSD-based RAID5 array driven by the Fin1 trace. The baseline reconstruction algorithm is PR.

To examine the performance impact of the comparable RAID levels on SPA, we conducted experiments on a RAID4 disk array consisting of seven disks and one hot-spare disk with a fixed stripe unit size of 64KB. The measured reconstruction times and response times on a RAID4+SPA system, omitted from this article due to space constraints, reveal a very similar performance improvement pattern to a RAID5+SPA system, indicating that SPA is likely to be similarly effective when applied to other RAID schemes such as RAID6 and parity declustering.

Currently, flash memory SSD (SSD for short) is considered as a potential alternative to traditional magnetic disk drive in the future, and has been widely deployed in RAID-structured storage systems in enterprise data centers. Compared with magnetic disk, SSD has advantages of superior random I/O performance, low power consumption, and shock resistance. Therefore, SSD-based RAID organization becomes a new hot research topic [Balakrishnan et al. 2010]. To examine the performance impact of SPA on parity-encoded SSD-based RAIDs, we conducted experiments on a RAID5 disk array consisting of seven SSDs and one hot-spare SSD with a fixed stripe unit size of 64KB. We use an SSD add-on module [Agrawal et al. 2008] for DiskSim 4.0 provided by Microsoft Research to simulate SSDs with a page-mapping FTL [Birrell et al. 2007], and the main specification parameters of SSD [Samsung Corporation 2007] are shown in Table IV.

We found out from simulation results that, for an SSD-based RAID5, reconstruction time performance and average response time performance keep consistent regardless of the fed traces (Web, Fin1, Fin2) or baseline reconstruction algorithms (PR, DOR). Therefore, we only show the experimental results with the PR baseline algorithm under the Fin1 trace. From Figure 7, one can see the reconstruction times for all the

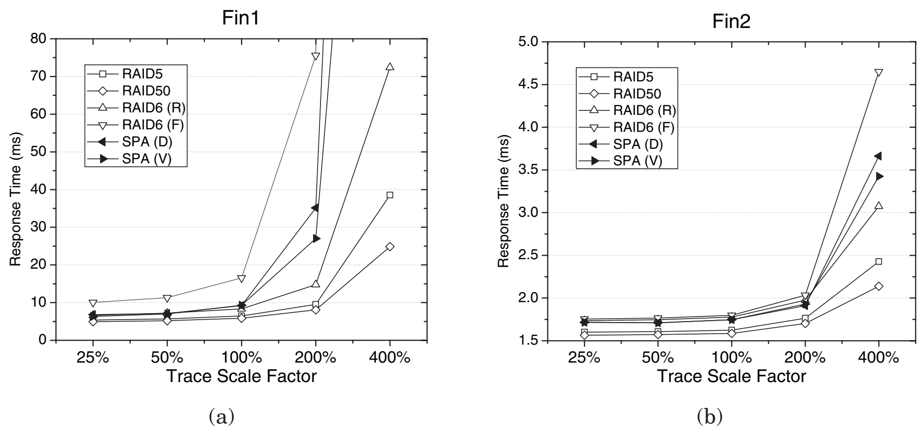



Fig. 8. Comparisons of response time of various RAID systems for the traces: Fin1 and Fin2.

schemes are almost the same. For the average response time performance, SPA(D) outperforms RAID5, RAID6(R), and RAID6(V) by 12.0%, 33.0%, and 26.5%, respectively, and approaches RAID50's performance (−4.4%), while SPA(V) performs a little worse than SPA(D) (−3.1%). We believe that the unique IO characteristics of SSD contribute to the disparity of recovery performance on traditional HDD-based array and the SSD-based array. First, due to millisecond-level page read/write performance of SSD, it can handle IO requests much faster than HDD, and thus shortens the queuing time for each request. Second, due to the uniformity of random page read/write performance of SSD, the interference between users' IOs and rebuild IOs, which significantly negatively impacts the recovery performance on HDD-based arrays, has weak performance impacts on SSD-based arrays. Third, due to the asymmetric read/write performance of SSD (25$\mu$s per page read while 200$\mu$s per page write), reducing rebuild reads on the SSD-based array cannot achieve the same performance gains on the HDD-based array. We think it is more important for SSD-based arrays to offload users' writes rather than reads during recovery. A possible solution is to combine the WorkOut approach [Wu et al. 2009] with our SPA to further improve the recovery performance of SSD-based arrays, and we leave this to our future work.

*Overhead Study.* It is very important to understand the performance cost due to parity update operations in the operational mode for SPA relative to other relevant RAID schemes, since it is this performance cost that will likely impact the choice of an appropriate candidate target system for online availability upgrade. Therefore, we conducted performance experiments on a RAID5 disk array consisting of seven disks and one hot-spare disk with a fixed stripe unit size of 64KB. We compare SPA(D), SPA(V), RAID5+0, RAID6(R), and RAID6(F) by online RAID-level migration (ORLM) from this baseline RAID5 system. Since the Web trace is read-only that hardly causes SPA and other schemes to do any parity update, we instead used the Fin1 and Fin2 traces as the input workload to the simulator. We introduced a scaling mechanism to vary the range of request arrival rates for the traces because the performance cost of the parity update operations is very sensitive to I/O intensity. In this scaling mechanism, a factor of 100% means no scaling while a factor of smaller or greater than 100% implies a proportional scaling down or scaling up in the arrival rates of the original trace. Figure 8 illustrates the response times for various RAID schemes with synchronous parity update policies in the operational mode for a range of trace-scaling factors on the Fin1 and Fin2 traces. It indicates that the RAID5+0 scheme consistently outperforms all other schemes while RAID6(F) consistently performs the worst among all schemes. SPA(D) and SPA(V) underperform RAID6(R), but noticeably outperform RAID6(F). On the other hand, when the trace scale factor is less than or equal to 100%, the difference in response time among all the schemes is very small, with the exception of RAID6(F). Specifically, the average response times of RAID5, RAID5+0, RAID6(R), RAID6(F), SPA(D), and SPA(V) are 6.5ms, 5.8ms, 8.4ms, 16.6ms, 9.2ms, 9.3ms, respectively under Fin1, while they are 1.62ms, 1.59ms, 1.78ms, 1.80ms, 1.75ms, 1.75ms, respectively under Fin2. This indicates that SPA with the synchronous update policy may not be a good candidate of availability upgrades under write-intensive workloads. However, it must be noted that the preceding experiments are meant to study the synchronous update overhead of SPA in the operational mode under heavy workload, which is much higher than its asynchronous counterpart that incurs negligible performance cost.

### 4.3. Reliability Analysis

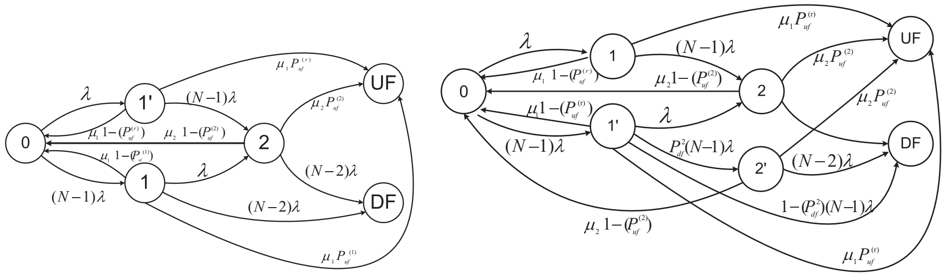
In this section, we first analyze the reliability of a RAID5 disk array incorporated with SPA and other schemes to obtain an intuitive but approximate comparison using a Continuous-Time Markov Chain (CTMC) model. Second, we develop a Sequential Monte-Carlo (SMC) simulator to obtain comparison results that are more accurate and sound based the more realistic assumption of the Weibull distribution for the failure process.

*Analytical Models.* CTMC has been widely applied to analyzing the reliability of storage systems, especially for RAID-structured systems [Dholakia et al. 2008; Iliadis et al. 2008]. Recently, the IPC paper [Dholakia et al. 2008] has developed appropriate CTMC models to evaluate MTTDL (Mean Time To Data Loss) of their proposed approaches to protecting against latent sector errors and demonstrated the applicability of their models. Similar to their models, we develop a reliability model and analyze the reliability of RAID systems that operate with our SPA approach integrated. We must point out that we recognize the recent research findings [Schroeder and Gibson 2007; Elerath and Pecht 2007] revealing the fact that failures in hard drives more closely follow a Weibull distribution than the Poisson distribution assumed by CTMC. However, since our main objective in this reliability analysis is to find the comparative rather than absolute estimates of reliability among SPA, RAID5, RAID5+0, and RAID6, and CTMC is flexible and conducive to simple and closed-form solutions, we choose to use CTMC as a first order of approximation for its simplicity and then use SMC to validate its results.

Due to the space constraints, we have to omit the detailed description on the reliability models of a RAID5 disk array integrated with the SPA Diagonal and Vertical protections, and the corresponding matrix constitution, deduction, and solution processes to obtain their respective MTTDLs, which can be found in our technical report [Tian et al. 2009]. Here, we only present the state transition diagrams of the reliability models of SPA, as shown in Figure 9.

We assess the reliability of the various schemes by considering a RAID system installation using the latest enterprise-level disk drives. Seagate Cheetah $15K.5$ is assumed as the disk model, along with its Annual Failure Rate (AFR) and the Unrecoverable Error Rate (UER) of 0.66% and $10^{-16}$, respectively, as reported in its specification data sheet [Seagate 2007]. The corresponding parameter values are listed in Table V. We refer to the IPC paper [Dholakia et al. 2008; Iliadis et al. 2008] to directly obtain the reliability model of a RAID5 disk array, and also derive the reliability model of a RAID5+0 array accordingly.

*Simulation Study.* In order to validate the CTMC results in light of the unrealistic Poisson assumption of CTMC, we also carry out an extremely time-consuming sequential Monte Carlo simulation [Elerath and Pecht 2007] study (it takes 4 days to complete




(a) SPA Diagonal (b) SPA Vertical

Fig. 9. Reliability model of a RAID5 array with SPA protections. Figure 9(a) depicts the reliability model of SPA Diagonal, and Figure 9(b) depicts the reliability model of SPA Vertical. *State 0* denotes the state in which all disks work normally; *State 1* denotes the state in which only the dedicated SPA disk fails; *State 1'* denotes the state in which any one disk except the SPA disk fails; *State 2* denotes the state in which the SPA disk and any other disk fail simultaneously; *State 2'* denotes the state in which two disks other than the SPA disk fail and the failures occur exclusively in different coverages; *State UF* denotes the state of an unrecovered failure caused by additional latent sector errors; and *State DF* denotes the state of a system failure caused by additional disk failures.

Table V. Disk Drive Reliability Parameters

| Parameter | Definition | Value |
|---|---|---|
| $\frac{1}{\lambda}$ | MTTF for a disk | $1,500,000$ h |
| $\frac{1}{\mu_1}$ | MTTR for one disk failure | 12 h |
| $\frac{1}{\mu_2}$ | MTTR for two disk failures | 12 h |
| $N$ | Number of disks in a disk array | 7 (RAID5), 8 (RAID5 with SPA, also RAID5+0) |
| $C_d$ | Capacity per disks | 146 GB |
| $S$ | Sector size | 512 bytes = 4096 bits |
| $P_{bit}$ | UER per bits read | $10^{-16} - 10^{-14}$ |

one SMC simulation) to generate a sufficient number of sample points of comparative reliability estimates based on the Weibull distribution.

The SMC study was conducted to estimate reliability measures of the RAID5, RAID5+0, RAID6, and SPA-powered RAID5, by simulating $100,000$ RAID sets in $87,600$ hours (10 years) with HDD failures following a Weibull distribution. Each transition distribution in Figure 9 is sampled. During that time, the sequence of hard disk failures, repairs, latent error defects, DF (Disk Failures), and UF (Unrecovered Failures) are tracked. For simplicity, a constant unrecoverable bit error rate independent of time and workload was used, similar to the IPC study [Dholakia et al. 2008].

Basically, our SMC simulator takes as inputs the same parameters as CTMC except for several key assumptions such as the distributions of disk failures and reconstruction times to make the simulation more realistic. A Weibull distribution with a slightly increasing failure rate is used. The characteristic life, $\eta$, is $461,386$ hours. The shape parameter, $\beta$, is 1.12. These parameters are also used in Elerath and Pecht [2007] according to its empirical statistics. The reconstruction times for all RAIDs also follow a Weibull distribution. All the RAIDs have the same parameters as those used in our CTMC analysis. The shape parameter of 2 generates a right-skewed distribution, and the characteristic life is 12 hours.

During the simulation, events such as hard disk failures, rebuilds, latent sector errors, DF (Disk Failures), and UF (Unrecovered Failures) are tracked. The current state of a RAID is sampled in the interval of one hour. The state transition (when

Table VI. Numeric and Normalized Comparisons of Number of Failures (UER = $10^{-14}$).

| Schemes | DDF | DUF | Total | Normalized |
|---|---|---|---|---|
| RAID5 | 14 | 20335 | 20349 | 1 |
| RAID50 | 6 | 10052 | 10058 | 2.02 |
| SPA(D) | 9 | 8750 | 8759 | 2.32 |
| SPA(V) | 3 | 2498 | 2501 | 8.14 |

Table VII. Numeric and Normalized Comparisons of Number of Failures (UER = $10^{-15}$)

| Schemes | DDF | DUF | Total | Normalized |
|---|---|---|---|---|
| RAID5 | 18 | 2596 | 2614 | 1 |
| RAID50 | 7 | 1103 | 1110 | 2.35 |
| SPA(D) | 7 | 947 | 954 | 2.74 |
| SPA(V) | 2 | 308 | 310 | 8.43 |

Table VIII. Numeric and Normalized Comparisons of Number of Failures (UER = $10^{-16}$)

| Schemes | DDF | DUF | Total | Normalized |
|---|---|---|---|---|
| RAID5 | 16 | 295 | 311 | 1 |
| RAID50 | 11 | 109 | 120 | 2.59 |
| SPA(D) | 4 | 113 | 117 | 2.66 |
| SPA(V) | 3 | 39 | 42 | 7.40 |

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 [Dholakia et al. 2008; Elerath and Pecht 2007]).

*Results*. Since both the CTMC model and SMC simulation have shown that MTTDL of RAID6 is four orders of magnitude higher than that of RAID5, RAID5+0, and SPA, it will not be explicitly included in this comparative study.

In Tables VI, VII, and VIII, the UER values are $10^{-14}$, $10^{-15}$ and $10^{-16}$, respectively. This is because the UERs of the desktop-level, near-line-level, and enterprise-level hard drives are typically $10^{-14}$, $10^{-15}$, $10^{-16}$, respectively [Whittington 2007]. The table headers of DDF, DUF, Total and Normalized in these three tables denote the number of double-disk failures, the number of double unrecoverable failures, the sum of both failures, and the ratio normalized to RAID5, respectively. Considering the linear relationship between the total number of failures in a large sampling space and MTTDL, the normalized ratio of number of failures should have the same value as the normalized ratio of MTTDL for the four RAID approaches. In the preceding three cases, SPA(D) is slightly better than RAID5+0 and SPA(V) consistently outperforms both RAID5+0 and SPA(D) significantly. Particularly, both SPA(D) and SPA(V) exhibit increasingly improved reliability performance with the increase of UER. Table IX shows that the reliability results obtained through CTMC and SMC based on parameters of Table V, listing MTTDL values normalized to that of RAID5 under the UER values of $10^{-16}$, $10^{-15}$, and $10^{-14}$. It is evident that the relative MTTDL values of the various schemes being compared remain reasonably consistent under both the CTMC and SMC methods, thus validating our use of CTMC to assess the comparative reliability estimates in the context of this article. In fact, the results further indicate that CTMC tends to generally underestimate the reliability improvement of SPA over RAID5+0, sometimes very significantly.

It must be noted that the same reconstruction time is used in our analysis and simulations for all schemes studied. However, since SPA(D) and SPA(V) improve the reconstruction time of both RAID5 and RAID5+0, their reliability advantages over the

Table IX. Comparisons of Normalized MTTDLs Obtained by SMC and CTMC

| | MTTDL(Normalized to RAID5) | | | | | |
|---|---|---|---|---|---|---|
| | UER: $10^{-16}$ | | UER: $10^{-15}$ | | UER: $10^{-14}$ | |
| Schemes | SMC | CTMC | SMC | CTMC | SMC | CTMC |
| RAID5+0 | 2.6 | 2.7 | 2.9 | 2.7 | 2.1 | 2.7 |
| SPA(D) | 1.9 | 2.5 | 3.6 | 3 | 2.4 | 3.1 |
| SPA(V) | 3.4 | 3 | 7.8 | 3.1 | 8.7 | 3.1 |

latter two should be more pronounced if taking into account of the reduced reconstruction times.

### 4.4. Discussions

From the previous performance evaluations and reliability analysis, it is clear that all of the schemes have their respective advantages and drawbacks as availability upgrade mechanisms for an online RAID5 system. For example, while RAID6 offers the best reliability improvement, it fails to provide the same level of recovery performance offered by SPA. Obviously, detailed availability demands and benefit/cost ratios of the available upgrade approaches must be taken into consideration when choosing an appropriate mechanism. Among the relevant approaches to upgrading a RAID5 system with an additional disk, namely, RAID5+0, RAID6(R), RAID6(F), SPA(D) , and SPA(V), the first two require data layout reorganization and thus should be excluded from the consideration of online availability upgrade. In what follows we provide a guideline for online availability upgrade.

(1) if system reliability is a top priority and there is no performance-centric SLA constraint, then RAID6(F) is the appropriate choice; otherwise,
(2) if reliability is a top priority but there is also a performance-centric SLA constraint, then SPA(V) is the appropriate choice; and finally,
(3) if performance during recovery is a top priority, then SPA(D) is the appropriate choice.

It should be noted from Sections 4.2 and 4.3 that SPA can be advantageous in many cases even when data layout reorganization is tolerable during the upgrade.

## 5. CONCLUSIONS

In this article we propose a simple but powerful scheme, Supplementary Parity Augmentation (SPA), to flexibly upgrade the availability of parity-based RAID systems online in production environments. The basic idea of SPA is to store and update supplementary parity units on the newly augmented spare disk(s) for online RAID systems in the operational mode, thus achieving the goals of improving the reconstruction performance and tolerating multiple disk failures and latent sector errors during recovery.

Our reliability modeling and simulation results demonstrate that SPA can achieve higher system reliability than RAID5 and RAID5+0. More importantly, we implement and integrate our SPA approach into the DiskSim simulator to study SPA's performance improvement and overhead. The trace-driven simulation results show that our SPA approach can significantly improve the reconstruction time and response time performance during recovery, with acceptable performance overheads during the operational mode.

SPA provides a new and effective solution for today's data centers to upgrade storage system availability, for improved performance under recovery and enhanced failure tolerance. As an ongoing research project, SPA, along with its various potential extensions, has a rich design and implementation space to be further explored and prototyped.

## REFERENCES

AGRAWAL, N., PRABHAKARAN, V., WOBBER, T., DAVIS, J. D., MANASSE, M., AND PANIGRAHY, R. 2008. Design tradeoffs for ssd performance. In *Proceedings of the Usenix Annual Technical Conference (USENIX'08).*

ANDERSON, D. AND WHITTINGTON, W. 2007. Hard drives: Today and tomorrow. http://www.usenix.org/events/fast07/tutorial

BAIRAVASUNDARAM, L. N., GOODSON, G. R., PASUPATHY, S., AND SCHINDLER, J. 2007. An analysis of latent sector errors in disk drives. In *ACM SIGMETRICS'07 Joint International Conference on Measurement and Modeling of Computer Systems.*

BALAKRISHNAN, M., KADAV, A., PRABHAKARAN, V., AND MALKHI, D. 2010. Differential raid: Rethinking raid for ssd reliability. In *Proceedings of the 5th European Conference on Computer Systems (EuroSys'10).* 15–26.

BIRRELL, A., ISARD, M., THACKER, C., AND WOBBER, T. 2007. A design for high-performance flash disks. *SIGOPS Oper. Syst. Rev. 41,* 2, 88–93.

BLAUM, M., BRADY, J., BRUCK, J., AND MENON, J. 1995. Evenodd: An efficient scheme for tolerating double disk failures in raid architectures. *IEEE Trans. Comput. 44*, 2, 192–202.

BUCY, J. S., SCHINDLER, J., SCHLOSSER, S. W., GANGER, G. R., ET AL. 2008. The DiskSim simulation environment version 4.0 reference manual. Tech. rep. CMU-PDL-08-101, Parallel Data Lab, Carnegie Mellon University.

CORBETT, P., ENGLISH, B., GOEL, A., GRCANAC, T., KLEIMAN, S., LEONG, J., AND SANKAR, S. 2004. Row-Diagonal parity for double disk failure correction. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST'04).*

DHOLAKIA, A., ELEFTHERIOU, E., HU, X.-Y., ILIADIS, I. , MENON, J., AND RAO, K. 2008. A new intra-disk redundancy scheme for high-reliability raid storage systems in the presence of unrecoverable errors. *ACM Trans. Storage 4,* 1, 1–42.

ELERATH, J. G. AND PECHT, M. 2007. Enhanced reliability modeling of raid storage systems. In *Proceedings of the Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07).*

GIBSON, G. A. 2007. Reflections on failure in post-terascale parallel computing. http://www.cse.ohio-state.edu/∼lai/icpp- 2007/Gibson- PDSI- ICPP07- keynote.pdf

GOLDING, R., BOSCH, P., STAELIN, C., SULLIVAN, T., AND WI LKES, J. 1995. Idleness is not sloth. In *Proceedings of the USENIX'95 Conference.*

HEATH, T., DINIZ, B., CARRERA, E. V., JR., W. M., AND BIANCHINI, R. 2005. Energy conservation in heterogeneous server clusters. In *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming (PPoPP'05).*

HOLLAND, M., GIBSON, G., AND SIEWIOREK, D. 1993. Fast, on-line failure recovery in redundant disk arrays. In *Proceedings of the* $23^{rd}$ *International Symposium on Fault-Tolerant Computing (FTCS'93) Digest of Papers.*

ILIADIS, I., HAAS, R., HU, X.-Y., AND ELEF T HERIOU, E. 2008. Disk scrubbing versus intra-disk redundancy for high-reliability raid storage systems. In *Proceedings of the ACM SIGMETRICS'08 Joint International Conference on Measurement and Modeling of Computer Systems.*

JIANG, W., HU, C., ZHOU, Y., AND KANEVSKY, A. 2008. Are disks the dominant contributor for storage failures? A comprehensive study of storage subsystem failure characteristics. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST'08).*

LEE, E. K. AND KATZ, R. H. 1993. The performance of parity placements in disk arrays. *IEEE Trans. Comput. 42*, 6, 651–664.

LEE, J. Y. B. AND LUI, J. C. S. 2002. Automatic recovery from disk failure in continuous-media servers. *IEEE Trans. Parall. Distrib. Syst. 13,* 499–515.

MI, N., RISKA, A., LI, X., SMIRNI, E., AND RIEDEL, E. 2009. Restrained utilization of idleness for transparent scheduling of background tasks. In *Proceedings of the* $11^{th}$ *International Joint Conference on Measurement and Modeling of Computer (SIGMETRICS'09).* 205–216.

MI, N., RISKA, A., SMIRNI, E., AND RIEDEL, E. 2008. Enhancing data availability in disk drives through background activities. In *Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN'08).*

NARAYANAN, D., DONNELLY, A., AND ROWSTRON, A. 2008. Write off-loading: Practical power management for enterprise storage. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST '08).*

NATHUJI, R., ISCI, C., AND GORBATOV, E. 2007. Exploiting platform heterogeneity for power efficient data centers. In *Proceedings of the IEEE International Conference on Autonomic Computing (ICAC'07).*

NET BSD FOUNDATION. 2008. The netbsd guide. Chapter 16, netbsd raidframe. http://www.netbsd.org/docs/guide/en/chap- rf.html

ØSTERGAARD, J. AND BUESO, E. 2004. The software-raid howto. http://tldp.org/HOWTO/Software- RAID-HOWTO.html

PATTERSON, D. A., GIBSON, G., AND KATZ, R. H. 1988. A case for redundant arrays of inexpensive disks (raid). In *Proceedings of the ACM SIGMOD'88 International Conference on Management of Data.*

PINHEIRO, E., WEBER, W.-D., AND BARROSO, L. A. 2007. Failure trends in a large disk drive population. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST'07).*

PLANK, J. S. 1997. A tutorial on reed-solomon coding for fault-tolerance in raid-like systems. *Softw.Pract. Exper. 27*, 995–1012.

PLANK, J. S. 2008. The raid-6 liberation codes. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST'08).*

RISKA, A. AND RIEDEL, E. 2008. Idle read after write: Iraw. In *Proceedings of the USENIX'08 Conference.*

ROSELLI, D., LORCH, J. R., AND ANDERSON, T. E. 2000. A comparison of file system workloads. In *Proceedings of the USENIX'00 Conference.*

SAMSUNG CORPORATION. 2007. K9xxg08xxm flash memory specification. http://www.samsung.com/global/system/business/semiconductor/product/2007/6/11/NANDFlash/SLC_LargeBlock/8Gbit/K9F8G08U0M/ds_k9f8g08x0m_rev10.pdf

SAVAGE, S. AND WILKES, J. 1996. Afraid: A frequently redundant array of independent disks. In *Proceedings of the USENIX'96. Conference*.

SCHROEDER, B. AND GIBSON, G. A. 2007. Disk failures in the real world: What does an mttf of 1,000,000 hours mean to you? In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST'07).*

SCHWARZ, T. J. E., XIN, Q., HOSPODOR, A., AND NG, S. 2004. Disk scrubbing in large archival storage systems. In *Proceedings of the Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation on Computer and Telecommunication Systems (MASCOTS'04).*

SEAGATE. 2007. Cheetah 15k.5 hard drives. http://www.seagate.com/www/en- us/products/servers/cheetah/cheetah_15k.5

SIVATHANU, M., PRABHAKARAN, V., ARPACI-DUSSEAU, A. C., AND ARPACI-DUSSEAU, R. H. 2004. Improving storage system availability with d-graid. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST'04).*

STORAGE PERFORMANCE COUNCIL. 2009. http://www.storageperformance.org

TECHTARGET. 2008. Enterprise-Class raid functions: What they are and when to use them. http://searchstorage.techtarget.com/tip/1, 289483, sid5_gci996811, 00.html

TIAN, L., FENG, D., JIANG, H., ZHOU, K., ZENG, L., CHEN, J., WANG, Z., AND SONG, Z. 2007. Pro: A popularity-based multi-threaded reconstruction optimization for raid-structured storage systems. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST'07).*

TIAN, L., JIANG, H., FENG, D., CAO, Q., XIE, C., AND XIN, Q. 2009. Spa: On-Line availability upgrades for parity-based raids through supplementary parity augmentations. Tech. rep. TR-UNL-CSE-2009-0006, Department of Computer Science and Engineering, University of Nebraska-Lincoln.

UMASS TRACE REPOSITORY. 2007. Oltp application and search engine storage traces. http://traces.cs.umass.edu/index.php/Storage/Storage

WEDDLE, C., OLDHAM, M., QIAN, J., AND IANDYWANG, A. 2007. Paraid: The gear-shifting power-aware raid. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST'07).*

WHITTINGTON, W. 2007. Desktop, nearline and enterprise disk drives, - delta by design -. http://www.snia.org/education/tutorials/2007/spring/storage/Desktop_Nearline_Deltas_by_Design.pdf

WU, S., JIANG, H., FENG, D., TIAN, L., AND MAO, B. 2009. WorkOut: I/O workload outsourcing for boosting raid reconstruction performance. In *Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST'09)*. 239–252.

XIE, T. AND SHARMA, A. 2009. Collaboration-Oriented data recovery for mobile disk arrays. In *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems (ICDCS'09).* 631–638.

XIE, T. AND WANG, H. 2008. MICRO: A multi-level caching-based reconstruction optimization for mobile storage systems. *IEEE Trans. Comput. 57,* 10, 1386–1398.

Received April 2010; revised August 2010; accepted November 2010