University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

CSE Technical reports   Computer Science and Engineering, Department of

2-20-2009

SPA: On-Line Availability Upgrades for Parity-based RAIDs through Supplementary Parity Augmentations

Lei Tian
University of Nebraska-Lincoln, tian@cse.unl.edu

Hong Jiang
University of Nebraska-Lincoln, jiang@cse.unl.edu

Dan Feng
Huazhong University of Science and Technology, China

Qiang Cao
Huazhong University of Science and Technology, China

Changsheng Xie
Huazhong University of Science and Technology, China

See next page for additional authors

Follow this and additional works at: http://digitalcommons.unl.edu/csetechreports

Part of the Computer Sciences Commons

Tian, Lei; Jiang, Hong; Feng, Dan; Cao, Qiang; Xie, Changsheng; and Xin, Qin, "SPA: On-Line Availability Upgrades for Parity-based RAIDs through Supplementary Parity Augmentations" (2009). *CSE Technical reports*. 64.
http://digitalcommons.unl.edu/csetechreports/64

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Technical reports by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Authors
Lei Tian, Hong Jiang, Dan Feng, Qiang Cao, Changsheng Xie, and Qin Xin

This article is available at DigitalCommons@University of Nebraska - Lincoln: http://digitalcommons.unl.edu/csetechreports/64

University of Nebraska–Lincoln, Computer Science and Engineering
Technical Report TR-UNL-CSE-2009-0006
Issued Feb. 20, 2009

SPA: On-Line Availability Upgrades for Parity-based RAIDs through Supplementary Parity Augmentations

Lei Tian[1,2], Hong Jiang[1], Dan Feng[2], Qiang Cao[2], Changsheng Xie[2], Qin Xin[3]
[1]*University of Nebraska-Lincoln, USA*
[2]*Huazhong University of Science and Technology, China*
[3]*Data Domain*

Abstract

In this paper, we propose a simple but powerful on-line availability upgrade mechanism, *S*upplementary *P*arity *A*ugmentations (*SPA*), to address the availability issue for 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 on-line 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. In particular, SPA has two supplementary-parity coverage orientations, SPA Vertical and SPA Diagonal, which cater to 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. The SPA with a single supplementary-parity disk can be viewed as a variant of but significantly different from the RAID5+0 architecture in that the former can easily and dynamically upgrade a RAID5 system to a RAID5+0-like system without any change to the data layout of the RAID5 system. Our extensive trace-driven simulation study shows that both SPA orientations can significantly improve the reconstruction performance of the RAID5 system while SPA Diagonal significantly improves the reconstruction performance of RAID5+0, at an acceptable performance overhead imposed in the operational mode. Moreover, our 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.

1. Introduction

In this paper, we try to answer a simple yet intriguing question: By augmenting new spare disks to a RAID[1] system, can we perform an on-line and flexible system upgrade to improve the RAID system availability in a way analogous to conventional on-line 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 are routinely experiencing system upgrades [2]. 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 [3]. Correspondingly, most RAID manufacturers have provided on-line upgrade mechanisms in their RAID products. For example, On-line Capacity Expansion (OCE) [4], which expands the storage capacity on-line, and On-line RAID Level Migration (ORLM) [4], which changes the RAID level on-line by augmenting new disks, respectively offer larger storage capacity, and higher I/O parallelism and reliability.

However, the question of how to upgrade the RAID's availability in production data centers by augmenting new spare disks on-line, while interesting and arguably important, remains unanswered yet. The latest findings and observations from real world by researchers [5, 6] have reported that disk failures and error rates are actually much higher than previously and commonly estimated, which suggest an urgent need to significantly improve the availability of RAID systems. Recently, Jiang et al. [7] 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. [8] analyzed the trend of latent sector errors in the same data set 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) [9]. 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 service providers violate the guaranteed performance of SLAs with unexpected down time or higher latency, 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, the clients will tend to consider those unexpectedly long response times as transient downtime events from users' perspectives even if the services are still available. In general, from the viewpoint of SLA, a transition from the Service Accomplishment state, in which the service is delivered as specified in SLA, to the Service Interruption state, in which the delivered service is different from SLA, is indeed considered a failure [9].

In this paper, 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 on-line 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 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 [4] and ORLM [4], SPA can be very flexibly enabled or disabled on demand. More importantly, SPA can be enabled or disabled without requiring any data re-organization on the original data layout of RAID systems. Thus, existing variants of the schemes rooted in XOR-based parity calculations, such as RAID5, RAID6, RAID5+0 and Parity Declustering [10], can also easily benefit from SPA.

It must be noted that a SPA with a single supplementary-parity disk 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 on-line 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 tradeoff 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 tradeoff 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 [11] and intra-disk redundancy [12, 13] 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 reduce the average user response time during recovery of RAID5, RAID5+0, RAID6, and SPA Vertical by a factor of up to 19.0×, 12.5×, 20.1×, and 14.6× respectively, while decrease their respective reconstruction time by a factor of up to 1.6×, 1.4×, 1.6×, and 1.5×. 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 paper is organized as follows. Motivations and background are presented in Section 2. In Section 3, we describe the SPA approach and its implementation in details. 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 paper in Section 5.

2. Motivations and Background

With rapid advances in the hard disk technology, hard drives have seen their capacity increasing while cost decreases drastically [14]. 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 [15, 16], 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 [17]. Leveraging the idle or lightly loaded periods has been a common practice to enhance performance, reliability and availability of storage systems [18, 19].

Furthermore, the Exclusive Or (XOR) calculation is widely used in parity-encoded RAID systems since any data unit can be regenerated by XORing all the remaining data units and a parity unit that covers all these units, referred to as the P parity. On the other hand, we observe that, given a sub-parity, referred to as the S parity, that covers one half or a portion of the data units, any data unit inside S's coverage 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 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 above observations as well as the performance upgrade mechanisms widely integrated into RAID products such as OCE and ORLM, we propose an on-line 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.

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 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 be common due to the frequent disk failures [20]. 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 [21], EVENODD code [22], Row-Diagonal Parity (RDP) code [23], and Liberation Codes [24], 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 [11] and Interleaved Parity Check (IPC) [12, 13] are proposed to detect or tolerate latent sector errors. Disk scrubbing[11] is an error detection method to scan all disk media in the background to detect latent sector

Table 1. 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 Diag.	Yes	Yes	No	High	Sync. / Async.
SPA Vert.	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.

errors. As a new form of intra-disk redundancy, IPC [12] 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 unrecoverable errors.

To reduce reconstruction time, a live-block recovery approach in D-GRAID [25] rebuilds only those data blocks that are considered live from the perspective of file systems and databases. PRO [26] deploys a popularity-based multi-threaded scheduling algorithm to rebuild the frequently accessed areas prior to rebuilding infrequently accessed areas to exploit access locality and sequentiality. Parity Declustering [10] reduces the additional load on survival disks during recovery by distributing small parity groups over a larger number of disks.

2.2. Distinctive Features of SPA

Although the set of existing reliability mechanisms described 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 above approaches lies in SPA's design principles and goals. SPA is a *supplementary redundancy approach designed as an on-line 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 1.

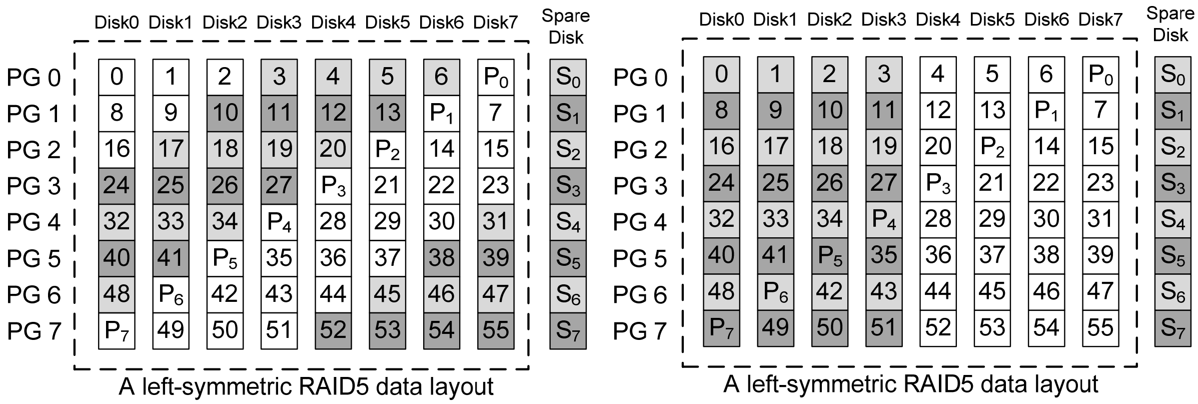
Table 1 illustrates the distinctive features of the two SPA approaches, SPA Diagonal and SPA Vertical detailed in Section 3, that set them apart from other representative availability-enhancing approaches: RAID6, RAID5+0, Parity Declustering and IPC. RAID6 has the best capability of tolerating any two disk failures while IPC has the best capability of tolerating unrecoverable 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 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 on-line 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 on-line upgrade mechanism, since performing data re-organization 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. Supplementary Parity Augmentation

3.1. The Basic SPA Idea

As Figure 1(a) shows, given a RAID5 left-symmetric disk array consisting of eight disks, SPA employs a



(a) SPA Diagonal (b) SPA Vertical

LEGEND: D – Data Unit P – Full Parity Unit S – Supplementary Parity Unit PG – Parity Group

Figure 1. Two examples of SPA coverage and layout. Figure 1(a) depicts an SPA coverage with *Diagonal Orientation*, and Figure 1(b) depicts an SPA coverage with *Vertical Orientation*.

dedicated hot-spare disk to store supplementary parity units that constitute the 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. For example:

$$S_0 = D_3 \oplus D_4 \oplus D_5 \oplus D_6 ; \quad (1)$$

$$S_7 = D_{52} \oplus D_{53} \oplus D_{54} \oplus D_{55} ; \quad (2)$$

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:

$$D_4 = D_3 \oplus D_5 \oplus D_6 \oplus S_0 ; \quad (3)$$

On the other hand, for 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 as well as the full parity unit P and S itself for the same parity. For example:

$$D_{28} = P_4 \oplus D_{29} \oplus D_{30} \oplus S_4 ; \quad (4)$$

Similarly, for any full parity unit on the failed disk that is not covered by S for the corresponding parity group, it can be regenerated by XORing all the data units outside S's coverage and the S parity unit. For example:

$$P_3 = D_{21} \oplus D_{22} \oplus D_{23} \oplus S_3 ; \quad (5)$$

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.

Figure 1(b) illustrates another SPA approach with a different coverage orientation of supplementary parity. There are two forms of supplementary parity distribution: supplementary parity with *Diagonal Orientation* (as shown in Figure 1(a)) and supplementary parity with *Vertical Orientation* (as shown in Figure 1(b)). Diagonal Orientation implies that units covered by SPA are distributed diagonally; while 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 1(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:

$$D_0 = D_1 \oplus D_2 \oplus D_3 \oplus S_0 ; \quad (6)$$

Assume that another disk in Figure 1(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:

$$D_4 = D_5 \oplus D_6 \oplus P_0 \oplus S_0; \quad (7)$$

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 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 sub-groups, in which the units in one sub-group is protected by S, and the units in the other sub-group is independently protected by $P \oplus S$. Therefore, any two simultaneous unrecoverable errors respectively and exclusively occurring in two sub-groups can be tolerated and recovered. For example and also referring to Figure 1, assume that Disk 4 fails at some point, and a latent sector error occurs in unit D_0 (that is also Unit 0 on Disk 0 physically) during the recovery for unit D_4 (Unit 0 on Disk 4 physically), for SPA with Diagonal Orientation, D_0 can be regenerated as follows:

$$D_0 = D_1 \oplus D_2 \oplus P_0 \oplus S_0; \quad (8)$$

And D_4 can be regenerated as in (3).

For SPA with Vertical Orientation, D_0 and D_4 can be regenerated as in (6) and (7).

Obviously, the efficiency of this capability of conditionally tolerating and recovering from unrecoverable errors depends on the occurrence locations of latent sector 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. 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 [27]. 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. 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 (not updated yet) 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.

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 S is only supplementary to the full parity P 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 [28]. Therefore, asynchronous update policies are not suitable to be used in 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 1 depicts a *Half-Parity* coverage range, where the coverage range is 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 tradeoffs. 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 paper 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 [22], RDP [23], and Liberation Codes [24], are also based on XOR operations. To the best of our knowledge, only the Reed-Solomon code [21] 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 [12, 13] 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.

Flexibility. As an on-line 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. Upon enabling, SPA can exploit the idle times or lightly-loaded periods to generate and store the supplementary parity units on the spare disks until all the supplementary parity units are consistent with their corresponding covered data units. SPA's asynchronous update policy helps reduce performance impacts on RAID systems in the operational mode. Even for the synchronous update policy, the performance impacts to RAID systems are shown to be acceptable, as detailed in Section 4.

4. Performance Evaluations

4.1. Evaluation Methodology

We developed an extended version of the DiskSim 4.0 simulator [29] to study the performance impacts of our SPA approaches by first extending DiskSim with two baseline rebuild algorithms, Pipeline Reconstruction (PR) [30] and Disk-Oriented Reconstruction (DOR) [10], 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. 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 sub-modules onto it.

Table 2. Disk and RAID Configuration Parameters.

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

Table 3. The Trace Characteristics.

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

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 sub-module into the logorg module. The main functions of the rebuild submodule include: 1) managing the 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, e.g., RAIDframe in NetBSD

[31], MD in Linux [32]. 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.

We then implemented and integrated our SPA approach into the logorg module, and made it to work together with the rebuild sub-module. The main functions of SPA sub-module are: 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., a 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 sub-module 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 sub-module. 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 is simulated. One of the latest disk models with the Generation-4 layout, Seagate Cheetah 15K.5 [33], is used throughout our experiments, with its main specifications listed in Table 2. We apply two types of workloads to the simulator: WebSearch and Financial [34] obtained from the Storage Performance Council [35]. 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 3.

We only rebuild 5% of the total capacity of our disk model (about 7.4GB) 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 the 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. 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 paper 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. We measure and report the amount of performance improvement by *speedup*, defined as the ratio of the measured response (or reconstruction) times of the old and new schemes.

4.2. Experimental Evaluations

Recovery Performance Study

We first conduct our experiments on a RAID5 disk array consisting of 7 disks and one hot-spare disk with a stripe unit size of 64KB to evaluate the recovery performance of SPA. We incorporate 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. Due to the space constraints, we only report the experimental results on the recovery performance for the Web and Fin2 traces.

Figure 2 and Figure 3 show the comparisons of reconstruction times and average response times of the schemes under the Web and Fin2 workloads, assuming the DOR and PR baseline rebuild algorithms respectively. 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 2, SPA(D) speeds up the reconstruction time by a factor of 1.6× over RAID5, 1.4× over RAID5+0, 1.2× over RAID6(R), 1.6x over RAID6(F) and 1.5x over SPA(V), while RAID5+0 achieves the best response-time performance. The Fin2 trace results of Figure 2 show that RAID5+0, RAID6(R) and SPA(D) outperform RAID5 in reconstruction time by a factor of 1.04×, 1.3× and 1.2× respectively, and in response time by a factor of 1.6×, 0.99×, and 1.8× respectively. On the other hand, SPA(V) outperforms RAID5 in response time by a factor of 1.2× and 1.5× under the Web and Fin2 traces respectively; whereas, reconstruction time is improved rather marginally. The experimental results based on the PR baseline algorithm, shown in Figure 3, 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,

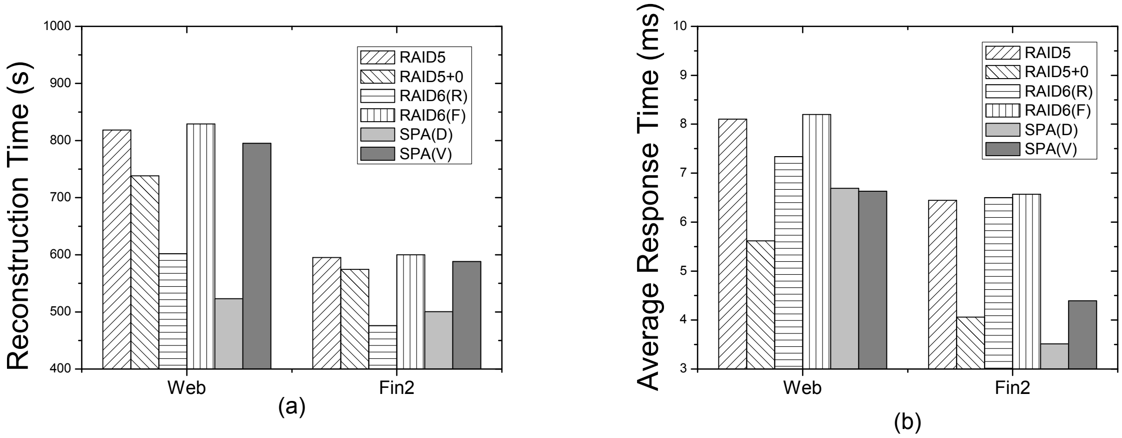


Figure 2. Comparisons of reconstruction times and average response times of the schemes based on the DOR baseline algorithm.

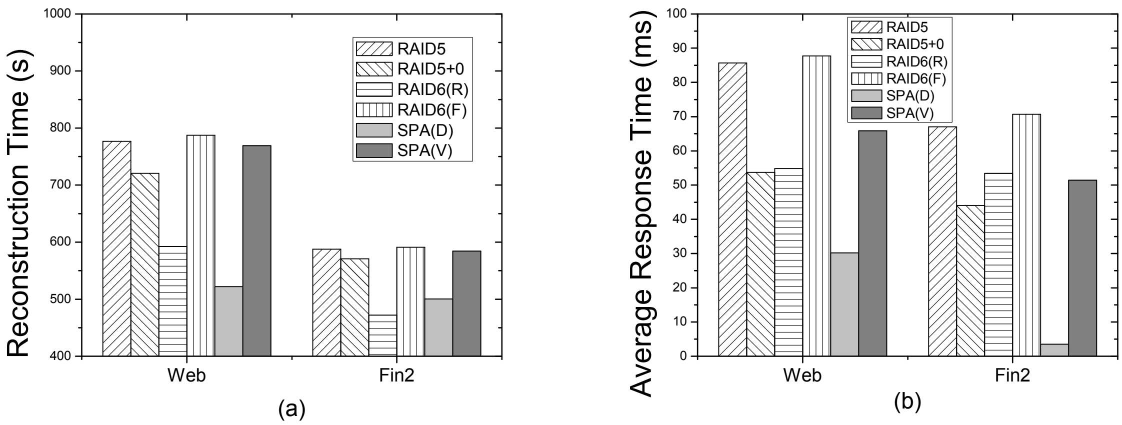


Figure 3. Comparisons of reconstruction times and average response times of the schemes based on the PR baseline algorithm.

RAID6(R), RAID6(F), and SPA(V) by a factor of 2.8×, 1.8×, 1.8×, 2.9×, and 2.2× respectively under the Web trace, and by a factor of 19.0×, 12.5×, 15.2×, 20.1×, and 14.6× respectively under the Fin2 trace.

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 for both traces by up to 25%. This is because SPA(V) does not distribute the reconstruction loads evenly among component disks, so that nearly half disks are under the same reconstruction I/O intensity as the baseline system while the other half 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 conduct 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 4, increasing the number of disks usually shortens the reconstruction time and response time due to the increased disk parallelism. However, one can find 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.

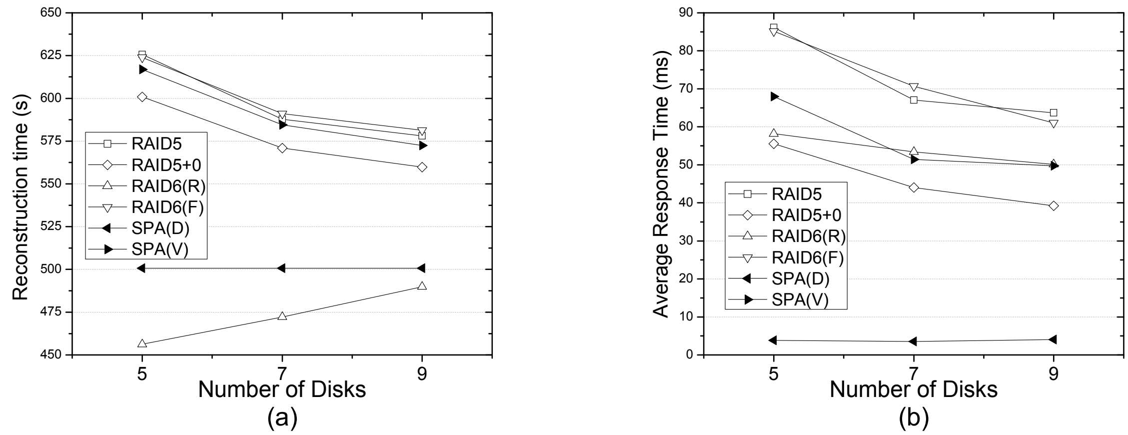


Figure 4. Comparisons of reconstruction times and average response times with respect to the number of disks driven by the Fin2 trace. The baseline reconstruction algorithm is PR.

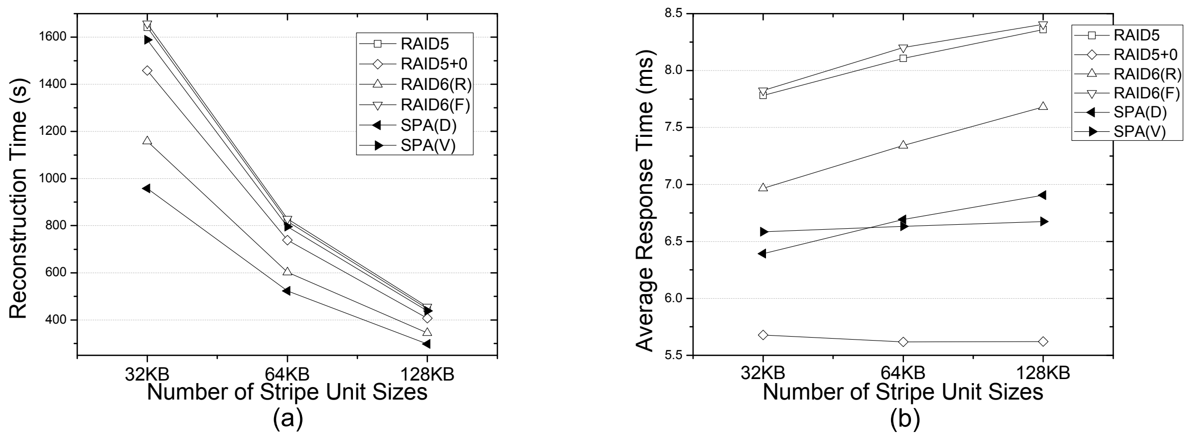


Figure 5. Comparisons of reconstruction times and average response times with respect to the numbers of stripe unit sizes driven by the Web trace. The baseline reconstruction algorithm is DOR.

To examine the performance impact of the stripe unit size, we conduct experiments on a RAID5 disk array consisting of 7 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 5. 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 Figure 2 and 3, 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.

To examine the performance impact of the comparable RAID levels on SPA, we also conduct experiments on a RAID4 disk array consisting of 7 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 paper 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.

Overhead Study

It is very important to understand the performance cost due to parity update operations in the operational mode for SPA and other relevant RAID schemes, since it is this performance cost that will likely impact the choice of an appropriate candidate target system for on-line availability upgrade. Therefore, we conduct

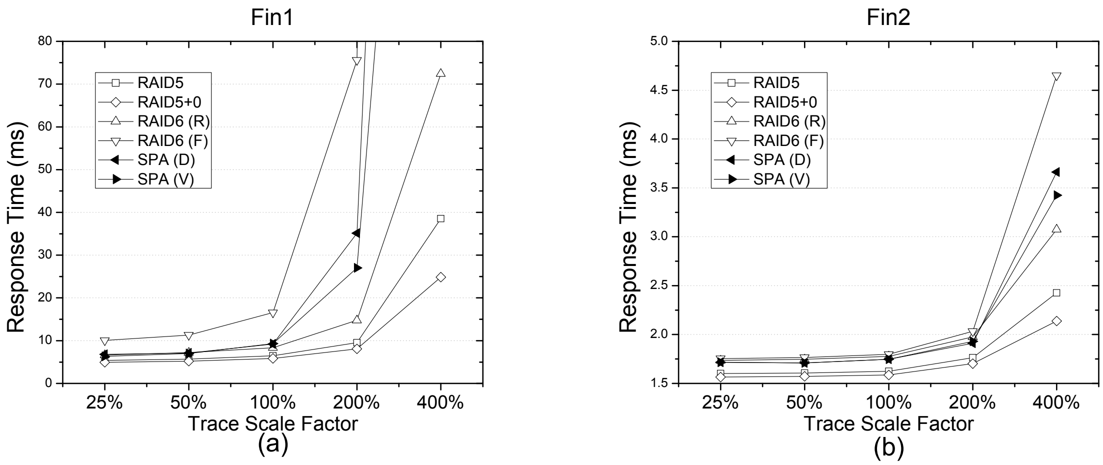


Figure 6. Comparisons of response time of various RAID systems for the traces: Fin1 and Fin2.

performance experiments on a RAID5 disk array consisting of 7 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 on-line 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 use the Fin1 and Fin2 traces as the input workload to the simulator. We introduce 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 the 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 6 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). This indicates that SPA with the synchronous update policy may be not a good candidate of availability upgrades under write-intensive workloads. However, it must be noted that the above 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 Monte-Carlo simulator to obtain comparison results that are more accurate and sound based the more realistic Weibull distributions of the failure process.

Analytical Models

CTMC has been widely applied to analyzing the reliability of storage systems, especially for the RAID-structured systems [12, 13]. Recently, the IPC paper [12] have developed appropriate CTMC models to evaluate MTTDL 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 [5, 36] 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 for its simplicity.

In our model, we consider both disk failures and latent sector errors, and obtain the Mean-Time-To-Data-Loss (MTTDL) estimate of a RAID5 array with the SPA protection, since MTTDL is the *de facto* metric of interest in evaluating storage system reliability instead of the traditional Mean-Time-To-Failure (MTTF) metric [19]. Then we compare MTTDL of our model with those of a conventional RAID5 array and a RAID5+0 array, in order to quantitatively assess SPA's ability to protect against data loss.

We first detail the reliability model for a RAID5 disk with the SPA Diagonal approach, with its state transition diagram shown in Figure 7(a), followed by a brief presentation of the model for the SPA Vertical approach, with its state transition diagram depicted in Figure 7(b) and with a focus on the differences between the two. As shown in Figure 7(a), with the first disk failing and assuming that it is the SPA disk, the RAID5 array enters the degraded mode (state transition from 0 to 1), otherwise, with the failed disk not being the SPA disk, the array enters the critical mode (state transition from 0 to 1'). The rebuild of a stripe of the failed non-SPA disk is performed based on half of the corresponding stripes residing on the remaining disks according to the coverage orientation of SPA Diagonal (i.e., state transition from 1 to 0 or from 1' to 0 for a successful rebuild).

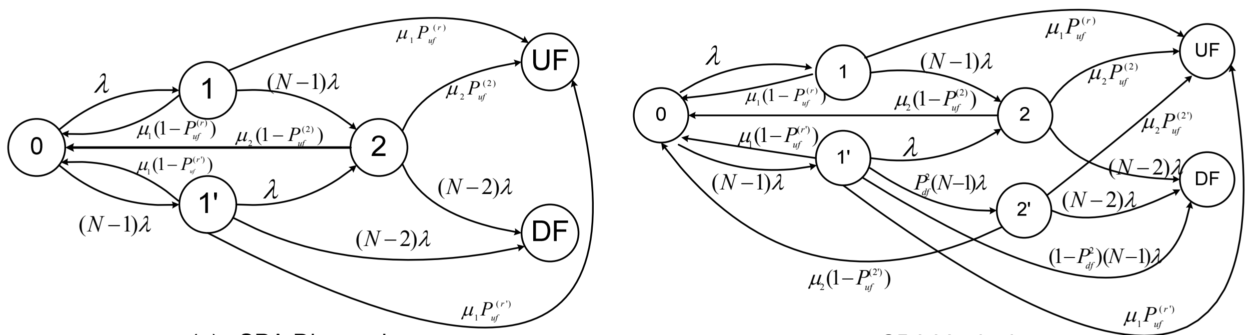
On the one hand, during the rebuild for the failed SPA disk, the probability of an unrecoverable failure, that is, a given stripe cannot be reconstructed, is upper-bounded by the probability that two or more of the corresponding stripes residing in the surviving data disks are in error (state transition from 1 to UF). If any data disk fails during the rebuild for the failed SPA disk, the RAID5 array enters the critical mode, and rebuilds for both failed disks simultaneously (state transition from 1 to 2). The occurrence of any latent sector error during this phase also causes an unrecoverable failure (state transition from 2 to UF). Similarly, any additional disk failure causes a system failure (state transition from 2 to DF). Upon a successful rebuild, the array transits from State 2 to State 0.

One the other hand, during the rebuild for any non-SPA disk failure, the probability of an unrecoverable failure is upper-bounded by the sum of two probabilities (state transition from 1' to UF). One is the probability that one or more of the corresponding stripes residing in half of the remaining data disks are in error given that the occurrence of a latent sector error and the previous disk failure are both in the coverage of SPA or they are both out of the SPA coverage. The other is the probability that two or more of the corresponding stripes residing in half of the remaining data disks are in error given that the occurrence of a latent sector error and the previous disk failure are not in the SPA coverage. If any other data disk fails during the rebuild for the previously failed data disk, it causes a system failure (state transition from 1' to DF). Otherwise, if the SPA disk fails during the rebuild, the RAID5 disk array enters the critical mode, and rebuilds for both failed disks simultaneously (state transition from 1' to 2). During this phase, the occurrence of any latent sector error causes an unrecoverable failure (state transition from 2 to UF). Similarly, any additional disk failure causes a system failure (state transition from 2 to DF). Upon a successful rebuild, the disk array transits from State 2 to State 0.

The related parameters required for the model are detailed in Table 4. We assume that disk failures are independent and exponentially distributed with an average rate of λ, and the rebuild times in the degraded mode and in the critical mode are exponentially distributed with average rates of μ_1 and μ_2, respectively. The upper bound of the probability that an unrecoverable failure occurs because the rebuild of the failed SPA disk cannot be completed before an additional error occurs is denoted by $P_{uf}^{(r)}$. Similarly, the upper bound of the probability that an unrecoverable failure occurs because the rebuild of the failed data disk cannot be completed before an additional error occurs is denoted by $P_{uf}^{(r')}$. The probability that an unrecoverable failure occurs because the rebuild of two simultaneously failed disks cannot be completed before an additional error occurs is denoted by $P_{uf}^{(2)}$.

Similarly, we can develop a reliability model for a RAID5 disk array with the SPA Vertical approach, as depicted in Figure 7(b). The main difference between the Diagonal and Vertical models is the additional state, State 2', in the latter that signifies the fact that the SPA Vertical approach can conditionally tolerate two simultaneous disk failures while the SPA Diagonal approach cannot. Consequently, the SPA Vertical model has three additional state transitions: from 2' to 0, from 2' to UF, and from 2' to DF, as shown in Figure 7(b). The probability that a RAID5 disk array with the SPA Vertical approach can tolerate two simultaneous disk failures is denoted by P_{df}^{2}.

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 [33]. The corresponding parameter values are listed in Table 4. We refer to the IPC paper [12, 13] to directly obtain the reliability model of a RAID5 disk array, and also derive the reliability model of a RAID5+0 array accordingly.



(a) SPA Diagonal (b) SPA Vertical

Figure 7. Reliability model of a RAID5 array with SPA protections. Figure 7(a) depicts the reliability model of SPA Diagonal, and Figure 7(b) depicts the reliability model of SPA Vertical.

Table 4. Disk Drive Parameters

Para-meter	Definition	Value
$1/\lambda$	MTTF for a disk	1,500,000 h
$1/\mu_1$	MTTR for 1 disk failure	12 h
$1/\mu_2$	MTTR for 2 disk failures	12 h
N	Number of disks in a disk array	7 (for RAID5), 8 (RAID5 with SPA, also RAID5+0)
C_d	Capacity per disk	146 GB
S	Sector size	512 bytes = 4096 bits
P_{bit}	UER per bits read	10^{-16} - 10^{-14}

Table 6. Notation of System Parameters

Parameter	Definition
State 0	All disks *work* normally
State 1	Only the dedicated SPA disk fails
State 1'	Any one disk except the SPA disk fails
State 2	The SPA disk and any other disk fail
State 2'	Two disks other than the SPA disk fail and the failures occur exclusively in different coverages
State UF	An unrecovered failure caused by additional latent sector errors
State DF	A system failure caused by additional disk failures
N	Number of the disks in a disk array
$1/\lambda$	Mean time to failure for a disk
$1/\mu_1$	Mean time to rebuild for one disk failure
$1/\mu_2$	Mean time to rebuild for two simultaneous disk failure
$P_{uf}^{(r)}$	The upper bound of the probability that an unrecoverable failure occurs because the rebuild of the failed SPA disk cannot be completed before an additional error occurs
$P_{uf}^{(r')}$	The upper bound of the probability that an unrecoverable failure occurs because the rebuild of the failed data disk cannot be completed before an additional error occurs
$P_{uf}^{(2)}$	The probability that an unrecoverable failure occurs because the rebuild of two simultaneously failed disks cannot be completed before an additional error occurs
P_{df}^{2}	The probability that a RAID5 disk array with the SPA Vertical approach can tolerate two simultaneous disk failures

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 [36] study to generate a sufficient number of sample points of comparative reliability estimates based on the Weibull distribution.

The Sequential Monte Carlo simulation (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 7 is sampled. During that time, the sequence of hard disk failures, repairs, latent error defections, DF (Disk Failures) and UF (Unrecovered Failures) are tracked. For simplicity, a constant unrecoverable bit error rate independent of time and workload was used.

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, η, is 461,386 hours. The shape parameter, β, is 1.12. These parameters are also used in [36] 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 minimum time of six hours is used for the location parameter. 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 and where to) is determined by the outcome 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 [12, 36].).

Results

Table 7. Unrecoverable sector error rate (10^{-14}, 100,000)

Types	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 8. Unrecoverable sector error rate (10^{-15}, 100,000)

Types	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 9. Unrecoverable sector error rate (10^{-16}, 100,000)

Types	DDF	DUF	Total	Normalized
RAID5	16	295	321	1
RAID50	11	109	120	2.68
SPA(D)	4	113	117	2.74
SPA(V)	3	39	42	7.64

Table 10. Comparisons of normalized MTTDLs obtained by SMC and CTMC.

RAID Types	MTTDL(Normalized to RAID5)					
	UER: 10^{-16}		UER: 10^{-15}		UER: 10^{-14}	
	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	3.0	2.4	3.1
SPA(V)	3.4	3.0	7.8	3.1	8.7	3.1

Since both the CTMC model and SMC 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 7, 8, and 9, the UER values are 10^{-16}, 10^{-15} and 10^{-14} respectively. This is because the UERs of the enterprise level, near-line level, and desktop level hard drives are typically 10^{-16}, 10^{-15}, 10^{-14} respectively [37]. The table headers of DDF, DUF, Total and Normalized in the first three tables denote the number of Double Disk Failure, the number of Double Unrecoverable Failure, 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 four RAID approaches. In above three cases, SPA(D) is slightly better than RAID 5+0 and SPA(V) consistently outperforms both RAID 5+0 and SPA(D) significantly. Particularly, both SPA(D) and SPA(V) exhibit increasingly improved performance with the increase of UER. Table 10 shows that the reliability results obtained through CTMC and SMC based on parameters of Table 4, 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 paper. In fact, the results further indicate that CTMC tend to generally underestimate the reliability improvement of SPA over RAID5+0, sometimes 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 RAID 5 and RAID 5+0, their reliability advantages over the latter two should be more pronounced if taking into account of the reduced reconstruction times.

4.4. Discussions

From the above 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 on-line 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 on-line availability upgrade. In what follows we provide a guideline for on-line 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 paper we propose a simple but powerful scheme, Supplementary Parity Augmentation (SPA), to flexibly upgrade the availability of parity-based RAID systems on-line in production environments. The basic idea of SPA is to store and update supplementary parity units on the newly augmented spare disk(s) for on-line 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 on-going research project, SPA, along with its various potential extensions, has a rich design and implementation space to be further explored and prototyped.

References

[1]. David A. Patterson, Garth Gibson, and Randy H. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *SIGMOD '88*, June 1988.

[2]. Ripal Nathuji, Canturk Isci, and Eugene Gorbatov. Exploiting Platform Heterogeneity for Power Efficient Data Centers. In *ICAC '07*, Jacksonville, FL, June 2007.

[3]. Taliver Heath, Bruno Diniz, Enrique V. Carrera, Wagner Meira Jr., and Ricardo Bianchini. Energy Conservation in Heterogeneous Server Clusters. In *PPoPP '05*, Chicago, IL, June 2005.

[4]. Enterprise-class RAID functions: What they are and when to use them. http://searchstorage.techtarget.com/tip/1,289483,sid5_gci996811,00.html

[5]. Bianca Schroeder and Garth A. Gibson. Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You? In *FAST '07*, San Jose, CA, February 2007.

[6]. Eduardo Pinheiro, Wolf-Dietrich Weber, and Luiz Andre Barroso. Failure Trends in a Large Disk Drive Population. In *FAST '07*, San Jose, CA, February 2007.

[7]. Weihang Jiang, Chongfeng Hu, Yuanyuan Zhou, and Arkady Kanevsky. Are Disks the Dominant Contributor for Storage Failures? A Comprehensive Study of Storage Subsystem Failure Characteristics. In *FAST '08*, San Jose, CA, February 2008.

[8]. Lakshmi N. Bairavasundaram, Garth R. Goodson, Shankar Pasupathy, and Jiri Schindler. An Analysis of Latent Sector Errors in Disk Drives. In *SIGMETRICS '07*, San Diego, CA, June 2007.

[9]. John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach, 4th Edition*. Morgan Kaufman Publishers, 2006.

[10]. M. Holland, G. Gibson, and D. Siewiorek. Fast, On-Line Failure Recovery in Redundant Disk Arrays. In *FTCS '93*, 1993.

[11]. Thomas Schwarz, Qin Xin, Ethan L. Miller, Darrell D. E. Long, Andy Hospodor, and Spencer Ng. Disk Scrubbing in Large Archival Storage Systems, In *MASCOTS '04*, October 2004.

[12]. Ajay Dholakia, Evangelos Eleftheriou, Xiao-Yu Hu, Ilias Iliadis, Jai Menon, and KK Rao. Analysis of a New Intra-Disk Redundancy Scheme for High-Reliability RAID Storage Systems in the Presence of Unrecoverable Errors. *ACM Transaction on Storage*, Vol. 4, No. 1, May 2008

[13]. Ilias Iliadis, Robert Haas, Xiao-Yu Hu , and Evangelos Eleftheriou. Disk Scrubbing Versus Intra-Disk Redundancy for High-Reliability RAID Storage Systems. In *SIGMETRICS '08*, Annapolis, MD, June 2008.

[14]. Dave Anderson and Willis Whittington. Hard Drives: Today & Tomorrow. Tutorial in *FAST '07*, San Jose, CA, February 2007.

[15]. Charles Weddle, Mathew Oldham, Jin Qian, and An-I Andy Wang. PARAID: A Gear-Shifting Power-Aware RAID. In *FAST '07*, San Jose, CA, February 2007.

[16]. Dushyanth Narayanan, Austin Donnelly, and Antony Rowstron. Write Off-Loading: Practical Power Management for Enterprise Storage. In *FAST '08*, San Jose, CA, February, 2008.

[17]. Richard Golding, Peter Bosch, Carl Staelin, Tim Sullivan, and John Wilkes. Idleness is not sloth. In *USENIX '95*, 1995.

[18]. Alma Riska, and Erik Riedel. Idle Read After Write - IRAW. In *USENIX '08*, Boston, MA, June 2008.

[19]. Ningfang Mi, Alma Riska, Evgenia Smirni, and Erik Riedel. Enhancing Data Availability through Background Activities. In *DSN '08*, Anchorage, AK, June 2008.

[20]. Garth A. Gibson. Reflections on Failure in Post-Terascale Parallel Computing. Keynote in *ICPP '07*, Xi'an, China, September 2007.

[21]. James S. Plank. A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems, *Software -- Practice & Experience*, 27(9), September 1997.

[22]. M. Blaum, J. Brady, J. Bruck, and J. Menon. EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures. In *ISCA '94*, 1994.

[23]. Peter Corbett, Bob English, Atul Goel, Tomislav Grcanac, Steven Kleiman, James Leong, and Sunitha Sankar. Row-Diagonal Parity for Double Disk Failure Correction. In *FAST '04*, San Francisco, CA, March/April 2004.
[24]. James S. Plank. The RAID-6 Liberation Codes. In *FAST '08*, San Jose, CA, February 2008.
[25]. Muthian Sivathanu, Vijayan Prabhakaran, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau. Improving Storage System Availability with D-GRAID. In *FAST '04*, San Francisco, CA, March/April 2004.
[26]. Lei Tian, Dan Feng, Hong Jiang, Ke Zhou, Lingfang Zeng, Jianxi Chen, Zhikun Wang, and Zhenlei Song. PRO: A Popularity-based Multi-threaded Reconstruction Optimization for RAID-Structured Storage Systems. In *FAST '07*, San Jose, CA, February 2007.
[27]. Drew Roselli, Jacob R. Lorch, and Thomas E. Anderson. A Comparison of File System Workloads. In *USENIX '00*, San Diego, CA, June 2000.
[28]. Stefan Savage and John Wilkes. AFRAID--A Frequently Redundant Array of Independent Disks. In *USENIX '96*, San Diego, CA, January 1996.
[29]. John S. Bucy, Jiri Schindler, Steven W. Schlosser, Gregory R. Ganger, and Contributors. The DiskSim Simulation Environment Version 4.0 Reference Manual. Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-08-101, May 2008.
[30]. Jack Y.B. Lee and John C.S. Lui. Automatic Recovery from Disk Failure in Continuous-Media Servers. *IEEE Transaction On Parallel And Distributed Systems*, Vol. 13, No. 5, May 2002.
[31]. NetBSD RAIDframe. http://www.netbsd.org/docs/guide/en/chap-rf.html
[32]. Linux MD. http://linux-raid.osdl.org/index.php/Linux_Raid
[33]. Cheetah 15K.5 Hard Drives. http://www.seagate.com/www/en-us/products/servers/cheetah/cheetah_15k.5/
[34]. UMass Trace Repository. http://traces.cs.umass.edu/index.php/Storage/Storage.
[35]. Storage Performance Council. http://www.storageperformance.org/home.
[36]. Jon G. Elerath and Michael Pecht. Enhanced Reliability Modeling of RAID Storage Systems. In *DSN '07*, Edinburgh, UK, June 2007.
[37]. Willis Whittington. Desktop, Nearline and Enterprise Disk Drives, Delta by Design. Spring 2007 SNIA Technical Tutorials. 2007.