

Reliability Tradeoffs in Personal Storage Systems

John A. Chandy
john.chandy@uconn.edu

Sumit Narayan
sumitn@engr.uconn.edu

Department of Electrical and Computer Engineering
University of Connecticut

ABSTRACT

RAID has long been established as an effective way to provide highly reliable disk subsystems. However, reliability in RAID systems comes at the cost of extra disks and somewhat lower performance. In this paper, we examine some mechanisms to reduce this cost in the context of integration with backup processes. These methods are most useful in storage systems where complete data protection or availability is not necessary such as in desktop personal computers, laptops, and other mobile storage devices. We will in particular investigate strategies with disk and flash that trade off between availability and reliability, snapshotting tradeoffs of reliability between time and space, and user directed redundancy.

1. INTRODUCTION

Disk arrays have long been used to improve the performance of storage systems [5, 10]. The parallelism inherent in multi-disk systems can significantly boost both the throughput and response times as compared to a single disk system. However, the increased performance comes at the cost of lower reliability. As a result, disk arrays need some form of redundancy to improve reliability. The most common and cost effective solution to improve the reliability of disk systems is the use of Redundant Array of Inexpensive (or Independent) Disks (RAID) [8]. A RAID system stripes data across multiple hard disks that appear to the user as a single disk. The various levels of RAID specify different methods of redundancy, such as parity and mirroring, to provide reliability. The most commonly used forms of RAID are RAID1 for mirroring and RAID5 for parity-rotated striping. Mirroring, or RAID1, entails replication of the data on multiple disks. Parity striping, or RAID5, involves spreading data along with parity across multiple disks. Choosing which RAID level to use is typically determined by cost and application requirements. At the disk array level, the redundancy choice is usually RAID5 as it provides excellent availability, moderate storage overhead, and adequate performance.

All RAID levels require extra disks to provide redundancy. In the case of RAID5, the redundancy overhead is $1/D$ where D is the number of data disks in a redundancy group. With RAID1, the overhead is 100%. This overhead makes RAID cost prohibitive in many environments, particularly single user desktop computers and laptops. RAID5 requires the installation of either SCSI controllers or multiple IDE controllers and enclosures to contain the multiple disks. In a two-disk scenario that is feasible for most single PCs, the 100% overhead of RAID1 mirroring becomes costly.

The reason for the high redundancy cost of RAID is the desire to maintain high data reliability and availability - in other words, the storage system should never lose data and should always be acces-

sible even in the presence of hardware failures. In certain environments, however, that guarantee of data protection is too stringent and not required. Thus, it may be possible to relax these guarantees and reduce the performance and storage cost of redundancy. One possible strategy is to delay parity calculations and writes to improve performance at the risk of small windows of data being unprotected as in the AFRAID system from HP [11].

In the same spirit, in this paper, we present storage system strategies that tradeoff between various aspects of data reliability. First, we explore an architecture that maintains data reliability while reducing the required storage at the cost of reduced system availability. Secondly, we look at data protection and backup mechanisms that improve performance by reducing backup recovery point times. Finally, we propose a mechanism to expose these tradeoffs to the user so that the user can tune redundancy choices on a file by file basis.

2. AVAILABILITY VS. RELIABILITY

RAID is a system that guarantees data protection as well as data availability. However, in many systems data protection is much more important than system availability. For example, in most personal desktops or laptops, if the disk has failed, the user may be able to tolerate the system being unusable for a few hours until the disk is replaced. However, any data loss will be unacceptable. In previous work, we have proposed a system called RAID0.5 [1], a midway point between RAID0 and RAID1 in that it provides equivalent data loss guarantees to RAID1 with just slightly more overhead than RAID0. However, RAID0.5 can not provide the higher system availability of RAID1. The goal of the RAID0.5 architecture is to provide a disk subsystem that can achieve high reliability with low cost with the tradeoff of lower availability.

The key to achieving low redundancy overheads in RAID0.5 is to replicate only a portion of the data on a disk. If we assume that the system is being periodically backed up, then we need to only replicate data that has been changed since the last backup. We define this data as the *active* data. In such a scenario, if a disk fails, active data can be recovered from the remaining disk(s) and inactive data can be recovered from the backup media. The backup window determines how much data needs to be replicated. For example, a weekly backup will likely create a larger active data set size than a daily backup. HP's study of working set sizes showed that, on average, only 2% of available storage is written to over a 24 hour period [9]. The largest observed 24-hour write set size was just over 10%. Thus, assuming daily backups, we need only to replicate 10% of the data in the array. If access patterns are similar over a week and the weekly and daily working set sizes are thus similar, a 15% replication overhead may be sufficient for weekly backups.

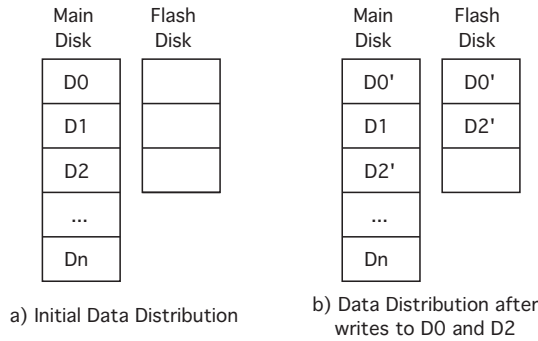


Figure 1: Flash Enabled Active Replication for Low End Survivable Storage (FEARLESS)

In this paper, we present a variation of RAID0.5 appropriate for systems that can not accommodate multiple disks - e.g. personal desktop computers, laptops, MP3 players, etc. In these systems, instead of using a second disk to replicate active data, we use a low cost flash drive such as a USB flash key to hold replicated data. The system is called Flash Enabled Active Replication for Low End Survivable Storage (FEARLESS) and is shown in Figure 1. In Figure 1a, the data distribution is shown for a system where the active data set is empty, i.e. immediately after a backup. After writes to $D1$, and $D3$, we can see in Figure 1b that the writes have been replicated to the flash disk. If the flash disk fills up, the file system or disk driver will respond as if the disk is full, or redundancy can be turned off with a warning. With current flash disk sizes, this could be problematic but with user directed attributes as described in Section 4, the amount of replication could be drastically reduced.

If the main disk fails, any requests to active data can be delivered from the flash disk. However, requests to inactive data can not be serviced because the data is not available online and must be retrieved from backup media. Thus, in a FEARLESS system, as soon as failure is detected, the system must block all future accesses to disk until the main disk can be reconstructed from backup. To prevent the chance of a failure on the flash disk, the system should be powered down. Note that this implies that a FEARLESS system is not available under failure even though there has been no data loss. This behavior is different from a traditional RAID system where data is still available even under a single disk failure. Thus, for a FEARLESS system there is a distinction between availability and data loss protection, whereas no such distinction exists for other RAID levels. A FEARLESS system trades off system availability for low redundancy costs. Therefore, using RAID0.5 is not appropriate for mission critical applications that require 24/7 operation, but it is appropriate for desktop applications that may not require nonstop operation but do require absolute data protection.

2.1 Reliability Analysis

In light of this distinction between system availability and data loss protection, when doing reliability analysis we are more interested in the mean time to data loss (MTTDL) rather than the mean time to failure (MTTF). The analysis is similar but, unlike MTTF, MTTDL tells us nothing about system availability.

We can develop an MTTDL and availability model for a FEARLESS system using a similar analysis methodology to that outlined in [8]. The main disk and flash disk are assumed to have independent and exponential failure rates. The disk and flash have

mean times to failure of $MTTF_{Disk}$ and $MTTF_{Flash}$ respectively. Data loss occurs when both the main disk and flash disk have failed. Since the system is shut off when the first failure occurs, the only chance for double failure is when the main disk or flash disk are being reconstructed after repair.

$$MTTDL_{FEARLESS} = \frac{MTTF_{Disk} * MTTF_{Flash}}{MTTF_{Disk} + MTTF_{Flash}} * \frac{1}{Pr[\text{data loss during repair}]} \quad (1)$$

Data loss during the repair time of the failed disk can happen in two cases: 1) the first failed disk was the flash disk and then the main disk fails, and 2) the first failed disk was the main disk and then the flash disk fails. Thus, the probability of data loss causing failure during the repair time is as follows:

$$Pr[\text{data loss during repair}] = Pr[\text{first failed disk was the flash disk}]p_m + (1 - Pr[\text{first failed disk was the flash disk}])p_f \quad (2)$$

where p_m is the probability that the main disk fails during the repair time and p_f is the probability that the flash disk fails during the repair time. If we define the time to restore the flash disk as $t_{r,Flash}$, then assuming exponential failure rates, $p_m = 1 - e^{-t_{r,Flash}/MTTF_{Disk}}$. Since $t_{r,Flash} \ll MTTF_{Disk}$, $p_m \approx \frac{t_{r,Flash}}{MTTF_{Disk}}$. Similarly, p_f is equal to $\frac{t_{r,Disk}}{MTTF_{Flash}}$.

Substituting into Equation 2, we arrive at:

$$Pr[\text{data loss during repair}] = \frac{MTTF_{Disk}}{MTTF_{Disk} + MTTF_{Flash}} \frac{t_{r,Flash}}{MTTF_{Disk}} + \frac{MTTF_{Flash}}{MTTF_{Disk} + MTTF_{Flash}} \frac{t_{r,Disk}}{MTTF_{Flash}} = \frac{t_{r,Flash} + t_{r,Disk}}{MTTF_{Disk} + MTTF_{Flash}} \quad (3)$$

Substituting into Equation 1, we arrive at:

$$MTTDL_{FEARLESS} = \frac{MTTF_{Disk} * MTTF_{Flash}}{t_{r,Disk} + t_{r,Flash}} \quad (4)$$

By comparison, a mirrored system with 2 disks has a MTTDL of $\frac{MTTF_{Disk}^2}{2 * MTTR_{Disk}}$ where $MTTR_{Disk}$ is the time to repair and restore the disk. Note that $t_{r,Disk}$ from Eq. 4 is only the time to restore the disk. In systems with hot spares, the time to replace the failed disk can be zero, but in the personal systems that we are considering, hot sparing is unlikely to be present. In a FEARLESS system, since the system is powered off after a failure, there is almost no chance of a second failure while the failed disk is being replaced¹. The restore/reconstruction time of a FEARLESS system is determined primarily by the speed of the backup media - tape can be slow but a D2D (disk-to-disk) backup system can be relatively fast.

2.2 Availability analysis

We have presented FEARLESS as a compromise choosing high reliability at the expense of lower availability. Availability is defined as the percentage of time that the system is not available.

¹The probability of failure while the disks are turned off is not absolutely zero, but it is much lower than the chance of failure while the disks are powered up, so for the most part we can ignore it

In reliability analysis terms, this is simply the ratio of the MTTF of the system to the sum of the MTTF and MTTR of the system. For a mirrored system the MTTF of the system is equivalent to the MTDDL. The MTTR of the system is equivalent to the MTTR of a disk since the system will be repaired when the disk is repaired. In practice, though, this is not true since when there is a system failure, the time to recover will probably be longer because of the time required to reinstall software, recover data from backups, etc. For the purposes of this discussion, however, we will assume, in the absence of hot spares, that the majority of the restore time is the time to install the new disk. The availability of a mirrored system can be expressed as follows:

$$A_{Mirror} = \frac{MTDDL_{Mirror}}{MTDDL_{Mirror} + MTTR_{Disk}} \quad (5)$$

$$= \frac{MTTF_{Disk}^2}{MTTF_{Disk}^2 + 2 MTTR_{Disk}^2} \quad (6)$$

For a FEARLESS system, the system is unavailable whenever a disk goes down even though data may not have been lost. Thus, we can derive the availability of a FEARLESS system as follows:

$$A_{FEARLESS} = \frac{\frac{MTTF_{Flash}}{MTTF_{Disk} + MTTF_{Flash}} \frac{MTTF_{Disk}}{MTTF_{Disk} + MTTR_{Disk}} + \frac{MTTF_{Disk}}{MTTF_{Disk} + MTTF_{Flash}} \frac{MTTF_{Flash}}{MTTF_{Flash} + MTTR_{Flash}}}{\frac{MTTF_{Disk}}{MTTF_{Disk} + MTTF_{Flash}} + \frac{MTTF_{Flash}}{MTTF_{Flash} + MTTR_{Flash}}} \quad (7)$$

Comparing Eq. 6 with Eq. 7, we see that the FEARLESS system availability is lower by a factor approximately equal to the $MTTF$. In fact, FEARLESS availability is no better than a non-redundant disk system. This is borne out in Table 1 which shows MTDDL, availability, and available storage for a single disk, a two disk mirrored system, and a single disk with flash replication. $MTTF_{Disk}$ is 300,000 hours which is typical for low-cost consumer disks. $MTTF_{Flash}$ for USB flash drives is roughly 50,000 hours. In spite of the normally high reliability of electronic components, USB flash drives have relatively high failure rates because of the failure propensity of the USB connector [6]. If the flash is not removable, the mean time to failure is around 2,000,000 hours. We also assume that the time to reconstruct the disk is 6 hours for mirrored systems and 12 hours for a FEARLESS systems because of the assumption that the backup tape is slow and the need to reinitialize the replication flash disk. The time to reconstruct the flash disk is only one hour because of the smaller size. Since we are assuming small scale systems that are not mission critical, we do not assume any hot sparing, so the time to replace the disk or flash disk is assumed to be 18 hours. This gives a $MTTR_{Disk}$ of 24 hours for the mirrored systems, and 36 hours for FEARLESS systems. $MTTR_{Flash}$ is assumed to be 19 hours.

The tables show that FEARLESS can offer equivalent or better MTDDL to RAID1 without the need for a second drive. With the high reliability of built-in flash, the MTDDL of FEARLESS is significantly better than mirroring. The disadvantage, however, is the availability of the FEARLESS system. While the availability of FEARLESS is good (99.9%), this is due entirely to the reliability of the disk and not due to the flash replication. Enterprise systems typically demand six nines of reliability which can be delivered with RAID5 and mirrored systems, but is not possible with FEARLESS. Thus, the use of FEARLESS becomes a choice between high avail-

Configuration	MTTDL (years)	Availability
Single Disk	5.7	3.32
Mirror	213895	6.94
FEARLESS with USB Flash	131627	3.47
FEARLESS with Built in Flash	5265097	3.98

Table 1: MTDDL and Overhead. ($MTTF_{Disk} = 300000$ hours, $MTTF_{Flash} = 50000$ hours)

ability with low storage overhead and high availability with high storage overhead. As mentioned before, the ideal environment for such a system is for desktop or single-user systems that contain high value data but do not require 24/7 availability.

3. EXTERNAL VS. SYSTEM FAULTS

In recent years, many file systems and storage arrays have provided a feature called snapshotting which enables users to recreate the state of the file system and some previous point in time. Typical snapshot systems only allow rollbacks to specific points in time when a snapshot was initiated. More recent variations called continuous data protection (CDP) allow for any point in time recovery. In other words, the rollback can be performed to any desired previous point in time rather than arbitrary times decided by the system administrator.

Snapshotting and CDP are forms of backup that are intended to protect against both system failure and external faults. System failure is a failure of the hardware components in the storage system and can be adequately protected by RAID mechanisms. Thus, backup is essentially a protection against external failures such as user error (eg. accidental deletion), viruses, or security violations. Another way to look at it is to say that backup is redundancy in time and RAID is redundancy in space.

When snapshotting or CDP are implemented on a RAID5 system, there is a significant increase in disk I/O. For example consider the scenario in Figure 2. When the snapshot is created, a copy of the file inode is made with all block pointers pointing to the original location of the blocks. In this case, the file inode consists of three blocks at D00, D01, and D02. When we write to block D02, we use a copy on write mechanism to copy the original block and update the snapshot inode to point to the new location of the original block. The whole process, not counting updates to inodes, involves two logical writes and one logical read - i.e. a read to copy the block, a write to store the copy, and then a write of the new block. At the disk array level, each logical write becomes two disk writes and two disk reads in a RAID5 system. Thus, the snapshot write operation translates to four disk writes and four disk reads. One disk read can be saved because the copy read can be cached for use in parity generation on the new data, thereby reducing the disk access count to four disk writes and four disk reads. Thus, adding snapshots incurs a potentially 10% increase in disk accesses when using RAID5. With snapshots, this overhead is incurred on only the first update to a block after a snapshot is taken. With CDP, the overhead is incurred on every update to a block, and as such is much more severe.

Using a similar strategy to weigh tradeoffs in reliability as we did with FEARLESS, we can decrease the performance penalty due to snapshotting by reducing reliability guarantees. As we mentioned

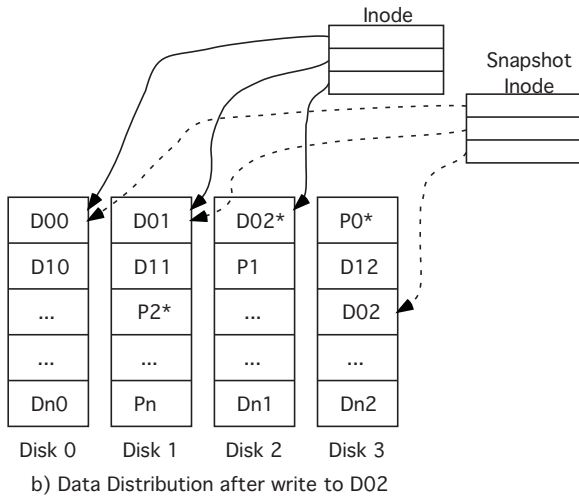
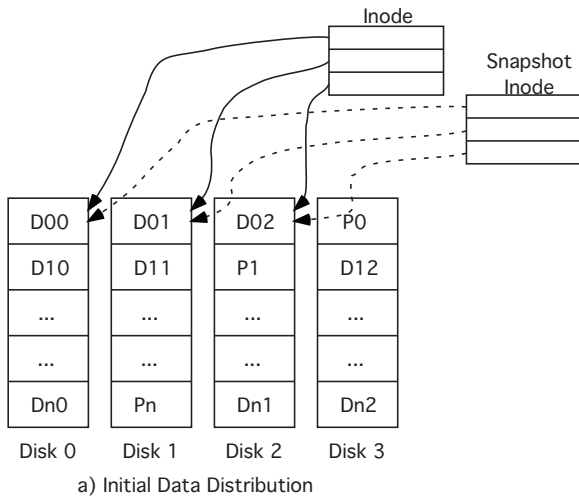


Figure 2: Snapshot Behavior on RAID5

earlier, snapshotting and RAID5 provide protection against external fault and system failures respectively. If we consider the external and system failures separately, we can arrive at a balance that weighs each failure mode independently and potentially improve performance. In most scenarios it is much more important to ensure that the current data is always available than making sure that old data is protecting. In such a scenario, protecting against hardware failure is more important than protecting against external failures. However, since external faults, particularly user errors, are more likely than hardware failures, it makes sense to protect against these errors to some extent as well using CDP, snapshots, or backups.

The strategy is to provide mechanisms to fully protect against hardware failures and thus preserve current data but minimize protection for old data in snapshots. Figure 3 shows how this could be accomplished by placing snapshot data in a region of the disk that is not protected by parity. Thus, the RAID5 penalty for parity generation is not incurred on the copy write, reducing the overall snapshot/CDP write cost to just three writes and two reads - an overhead of just 25%. Note that we have not said anything about snapshot protection, since we have assumed that preserving old data is not important. Thus, the tradeoff we make is much better performance

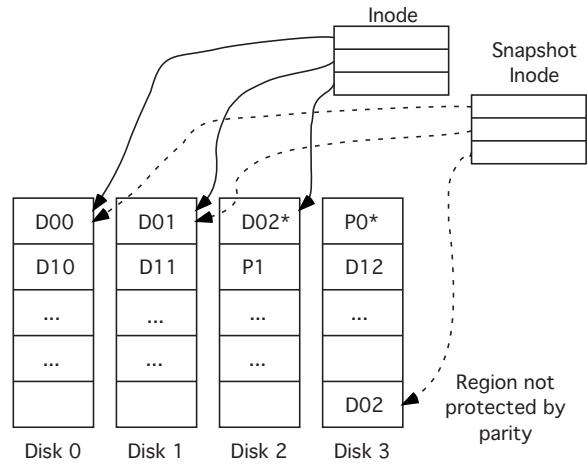


Figure 3: Minimized Snapshot Protection with RAID5

at the cost of lower data protection in time i.e. less protection against external faults.

In order to address this problem, we can use the fact that most systems keep multiple snapshots available on a system - e.g. hourly, daily, weekly, etc. If we rotate the snapshots among the disks, we are always guaranteed to have at least one snapshot available in case of disk failure. Thus, in the presence of failure we may not be able to roll back to the last snapshot, but we will be guaranteed to roll back to at least the snapshot before that. If snapshots are made hourly, we can guarantee that we can recover data from at the most two hours ago. Depending on the environment, lengthening the roll back window or recovery point objective (RPO) is an acceptable penalty to pay given the improved performance cost. In a CDP or near-CDP system, where the granularities of rollback are much smaller, and the update cost is much more severe, the benefits of such a system are even clearer. The performance is significantly improved with little cost in terms of lengthening the RPO.

4. USER DIRECTED REDUNDANCY

In the previous two sections, we have presented mechanisms for improving redundancy cost in terms of storage overhead and performance but to the detriment of availability or reliability in a particular aspect. Both methods use static means to determine the importance of availability or reliability on a storage system wide basis. However, ideally we would like to determine the reliability or availability needs more dynamically. For example, there may be certain files that require high availability and high reliability, while others may not require the same guarantees of data protection. Production databases files are examples of the former and temporary files created during other processes are examples of the latter. The AutoRAID system from HP uses this concept as it dynamically move files from high reliability mirroring to lower reliability parity storage [13]. However, it always presumes high reliability to start before gradually moving files to lower redundancy distributions.

The difficulty with dynamic mechanisms is that the only hint as to the importance of the file is the frequency of access - whereby files that are frequently accessed are presumed to be more important and thus need to have higher levels of protection. In practice, this is not often true. For example, applications may be frequently accessed but need not have high levels of data protection in non 24x7 envi-

ronments, since they can easily be re-installed. Therefore, the only reliable source for importance of files is the user who is able to make judgments of cost in terms of disk space and performance vs. availability, reliability, and RPO guarantees.

The natural expression of the user's wishes is through attributes on a file or directory and modern file systems support extended attributes which can express features other standard file metadata. We propose that these extended attributes be used to be specify such data protection features such as reliability and availability levels, RPO times, and backup intervals. However, since many of these decisions are implemented at layers of the storage system that are far removed from the file system, it is necessary to push this attribute information down the storage system stack. With current virtualized storage systems which abstract away much of the underlying storage architecture, this may be difficult, but it is absolutely necessary in order to fully expose to the user the costs and benefits of each reliability choice.

With such file by file redundancy and availability attributes, it becomes possible to implement the two techniques discussed in the previous sections on a much smaller granularity. Thus, some files could use the lower availability of FEARLESS while others could use no redundancy at all. This is particularly useful in FEARLESS in order to reduce the amount of data replicated on the flash drive. Another setting could guarantee that some files have full snapshot redundancy while other files may specify that no snapshotting is required at all. Obviously, these choices could be intermixed with each other or with additional storage optimizations such as AFRAID. The key is that the tradeoffs are made visible to the user so that the user can control the choice.

5. RELATED WORK

The FEARLESS method is most similar to techniques used to cache RAID5 parity. For example, parity logging caches partial parity in non-volatile memory which is then flushed to a log disk [12]. Comparable methods that use disk instead of memory as the cache include Trail, DCD, and RAPID-Cache [2, 4, 14]. FEARLESS differs from these methods not only in the fact that flash is used as the redundancy store, but also in that the flash acts as a cache to offline backup media rather than RAID5 parity. Thus, FEARLESS is much more appropriate in personal storage systems.

The use of removable storage as a cache has been recognized in distributed storage and pervasive storage research. For example, BlueFS uses removable storage as part of a cache hierarchy in a distributed file system [7]. Similarly, Tolia extended the Coda file system to support portable storage devices through lookaside caching built on file recipe hashing. The PersonalRAID system is a portable storage solution where the storage device is the central personal storage device and provides synchronization with local storage. The distinction between these systems and our system is that FEARLESS caching is designed for redundancy - in other words the cache is a redundancy cache of a backup system rather than a cache of a distributed file system.

FEARLESS is intimately tied into judicious use of backup and systems that provide seamless online backup can ease this process. Tape backed systems are inherently difficult to use and are not often used in personal storage systems. However, there have been recent efforts to automate the backup process including research efforts such as Pastiche [3] which uses peer-to-peer systems to provide backup storage and commercial efforts in disk-to-disk backup.

6. CONCLUSIONS

In this paper, we have presented some strategies to weigh reliability choices balanced with disk overhead and performance costs. In environments where 24/7 high data reliability is not always required, these tradeoffs present opportunities for performance and disk overhead benefits. For example, FEARLESS allows laptop users to have some level of data protection that would not otherwise be possible. Snapshot optimizations can improve performance while sacrificing a bit on data protection of old data. User specified attributes can extend these choices to a file-by-file basis for better control of these tradeoffs. The tradeoffs of these strategies show that reliability should not be a fixed in stone metric and should allow flexibility depending on the situation such that various aspects such availability or RPO can be relaxed.

7. ACKNOWLEDGEMENTS

The authors would like to thank Erik Riedel for his feedback and particularly his suggestion of the FEARLESS acronym.

8. REFERENCES

- [1] J. A. Chandy. RAID0.5: Active data replication for low cost disk array data protection. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, pages 963–969, June 2006.
- [2] T.-C. Chiueh. Trail: A track-based logging disk architecture for zero-overhead writes. In *Proceedings of International Conference on Computer Design*, pages 339–343, Oct. 1993.
- [3] L. P. Cox, C. D. Murray, and B. D. Noble. Pastiche: Making backup cheap and easy. In *Proceedings of the Symposium on Operating Systems Design and Implementation*, Dec. 2002.
- [4] Y. Hu and Q. Yang. DCD - Disk caching disk: A new approach for boosting I/O performance. In *Proceedings of the International Symposium on Computer Architecture*, pages 169–178, 1995.
- [5] M. Y. Kim. Synchronized disk interleaving. *IEEE Trans. Comput.*, C-35(11):978–988, Nov. 1986.
- [6] R. Kozlov and H. Heshes. Extending the warranty period of the DiskOnKey 500 series: Reliability report. White Paper 04-WP-0604-00, Rev 1.0, M-Systems, Newark, CA, Aug. 2004.
- [7] E. B. Nightingale and J. Flinn. Energy-efficiency and storage flexibility in the Blue file system. In *Proceedings of the Symposium on Operating Systems Design and Implementation*, 2004.
- [8] D. A. Patterson, G. A. Gibson, and R. H. Katz. A case for redundant arrays of inexpensive disks (RAID). In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 109–116, June 1988.
- [9] C. Ruemmler and J. Wilkes. A trace-driven analysis of working set sizes. Technical Report HPL-OSR-93-23, Hewlett-Packard, Palo Alto, CA, Apr. 1993.
- [10] K. Salem and H. Garcia-Molina. Disk striping. In *International Conference on Data Engineering*, pages 336–342, 1986.
- [11] S. Savage and J. Wilkes. AFRAID - A frequently redundant array of independent disks. In *Proceedings of the USENIX Technical Conference*, pages 27–39, Jan. 1996.
- [12] D. Stodolsky, G. Gibson, and M. Holland. Parity logging: Overcoming the small write problem in redundant disk arrays. In *Proceedings of the International Symposium on Computer Architecture*, pages 64–75, 1993.
- [13] J. Wilkes, R. Golding, C. Staelin, and T. Sullivan. The HP AutoRAID hierarchical storage system. *ACM Transactions on Computer Systems*, 14(1):108–136, Feb. 1996.
- [14] M. Zhang, X. He, and Q. Yang. Implementation and performance evaluation of RAPID-Cache under Linux. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, June 2002.