A Case Study on Reliability of Spider Storage System

Feiyi Wang, Sarp Oral, Galen Shipman, David Dillow
{*fwang2, oralhs, gshipman, dillowda*}*@ornl.gov*
National Center for Computational Sciences
Oak Ridge National Laboratory

Abstract

Spider storage system runs one of the world's largest and fastest POSIX-compliant parallel filesystem. With over 10 petabytes of unformatted capacity and 240 gigabytes per second throughput, Spider supports the world's most powerful open science capability supercomputer, Jaguar XT5, at Oak Ridge National Laboratory (ORNL). Necessitated by both scale and uniqueness of such system, this paper aims to provide a detailed discussion on its hardware characteristics from reliability perspective, establish a baseline failure model and develop a quantitative expectation on system's reliability and availability.

1 Introduction

The dominant factor when designing a capability HPC system is scalability and performance - be it about raw CPU, I/O, or networking. Availability and reliability, though generally regarded important, are often coped with best effort so long as it is within the tolerance of scientific users. This resignment from high availability comes with a recognition of scale and complexity of such high performance system and it makes a reasonable case for taking system offline regularly for scheduled and un-scheduled maintenance and repair.

According to a recent study [10], failure rate per processor per year is more or less the same and system's failure rate is proportional to the scale. Furthermore, there is no obvious trend indicating that as technology progresses, the reliability indicator will get better. Since the Jaguar XT5 at ORNL is at the vanguard of large-scale scientific computing with rapidly expanding scale and computing power, we have all the more reasons to be concerned: it is important and imperative to gain deeper understanding on system failures and develop the right and quantitative expectation where we can, and engineer the system for better reliability and availability. After all, any gain of such directly translates into productivity and make the machine more *productive*.

Spider system is one of the world's fastest and largest POSIX-compliant parallel file system, designed to work with both Jaguar and other computing resources at the National Center for Computational Sciences at ORNL. At the backend is built around 48 DDN S2A9900 storage platforms. Each S2A9900 is configured with five ultrahigh density 4U, 60-bay disk drive enclosures (56 drives are actually used), which gives us

a total of 280 1-TB hard drives per S2A9900. The system as a whole has 13,440 TB or over 10 PB of unformatted capacity. On top of this, Spider is configured with 192 Lustre storage servers into a single global filesystem namespace, designed to deliver up to 240 GB/s. These characteristics provides a both unique and interesting case study for reliability and availability.

The rest of the paper is organized as follows. In Section 2, we present the physical layout of the storage system. Then, we dive into reliability analysis of a single disk array in one couplet. In Section 3, we investigate a DDN9900 couplet with its peripheral components to build failure model. Based on this failure model, we derive quantitative expectation on system failure rate and summarize the insights gained through this analysis in Section 4. Finally, we discuss some limitations of this analysis and the future work.

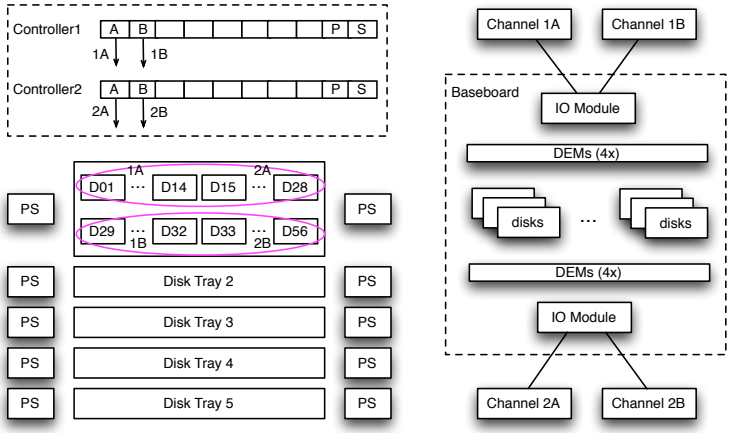


Figure 1: Illustration of Disk Physical Layout

2 Physical Layout of Storage System

The following sections frequently refer MTTF (Mean Time to Failure) and MTTR (Mean Time to Repair), which are the primary statistical parameters on estimating reliability of any system. As we are concerned not only with the RAID disks itself, but also the reliability of its peripheral components and hence the combined impacts on system reliability, it is necessary to expose some level of details on disk physical grouping and logical layout. This layout information is also the foundation on which we define the failure model for the overall Spider disk subsystem.

The basic unit in such storage system is *tiers*. Each tier contains 10 drives: 8 are data drives (channel A through H), plus one parity drive, channel P, and a second parity drive, known as channel *S*. A tier is built by taking one disk from each channel.

Figure 1 illustrates physical layout of the disk system. There are two power sources

(PS) for each controller and disk tray: one is regular house AC power, the other one is from UPS. The data transfer link from controller to disk tray is through Serial Attached SCSI (SAS). As this is a point-to-point serial connect, it further goes through an I/O module which is connected to 4 Disk Expansion Modules (DEMs), and each DEM can connect up to 15 disks. Therefore, the total number of disks for a single tray is up to 60.

As shown in the right hand side of Figure 1, there are two I/O modules on the baseboard, and each is shared by two channels. The 60 disks exported by the I/O module are split between two channels: each channel has accessibility for up to 30 disks. Hardware redundancy is built in as the pair of channels can access the same group of disks. In the illustration, $1A$ and $2A$ can both access disks from D_1 to D_{28}. Should either controller fails, the disks are still accessible.

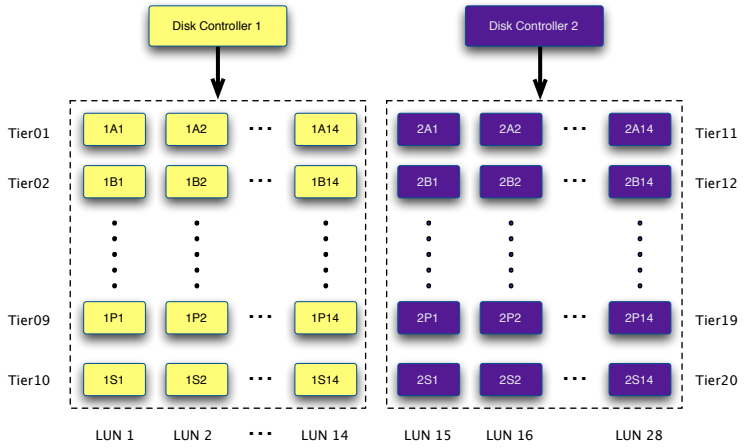


Figure 2: DDN S2A9900 Disk System Layout

Based on the physical layout given in Figure 2, we can present a logical layout of the tier structure: 28 tiers and each tier with 10 drives. The label of the disk is a tuple of $<$*controller number, channel number, disk number*$>$. The disk number itself is not named in linear fashion as in its physical layout, instead it wraps to one following the channel changes.

Note that disk drives used by Spider system are SATA-2 disk instead of enterprise class disks for cost reasons. The study by Schroeder and Gibson [9] found no significant difference on failure properties between the consumer and enterprise class drives. This also provided the reasoning and basis of some of the reliability parameters we practiced in the follow-on calculation.

3 Reliability of Disk Array in One Couplet

This section examines the reliability of a single couplet, a 280-disk drive system. A couplet is defined as two tandem S2ADDN9900 raid controllers working together in an active-active mode. The spider system as a whole has a total of 48 couplets and 13,440 disk drives. The disk drives are largely from the same vendor and thus homogeneous enough to be assumed to have the similar failure characteristics. We will discuss more on the definition of Spider failure from our point of view later.

3.1 JBD Scenario

In the following discussion, we first present a simplistic estimate on failures with no consideration of RAID, so called just a bunch of disk scenario (JBD). Then we discuss the RAID 6 employed in the Spider system, particularly in the context of three sources of failures: independent disk failures, uncorrectable bit errors, and the combination of two. We have deliberately ignored another possible source of failure: system crash, be it caused by misconfiguration, operation error, power loss or any environmental impact etc. Also, as mentioned before all couplets and disk trays are dual-powered. If the house power goes off, the UPS power kicks in and can power the whole system for another 7 seconds. This 7 second window is sufficient, such that any residue data held at the controller cache can be flushed to disks in milliseconds. And we consider the probability of *both* house power and UPS power went down being so small so it is safe to ignore in this context of discussion.

First, we consider JBD scenario. Assume each individual drive has a MTTF of 1.2 million hours as stated in manufacturer's data sheet, and it operates 24 hours per day, 365 days per year. That will give us AFR (Annual Failure Rate) of: $(365 \times 24)/(1.2 \times 10^6) = 0.73\%$

If we take recent disk failure study [9] into account: the real world AFR is four times of calculated AFR from vendor's data sheet, then we have the following expected number of disk failures per year: $48 \times 280 \times 4 \times 0.73\% = 336$ disk failures.

3.2 RAID 6: P+Q Redundancy

Now, we consider RAID 6 (P+Q redundancy) case following the basic premises setup by Patterson RAID framework [3, 7], assuming disks fail independently.

Let X_1 represents the time to first disk failure, X_2 and X_3 represent time to the 2nd and 3rd disk failures within the $MTTR$ of the remaining group, respectively, then we have:

$$
\begin{aligned}
MTTF_{group} &= E(X_1)\cdot E(X_2)\cdot E(X_3) \\
&= \frac{MTTF/S}{G+C}\cdot\frac{MTTF/S}{MTTR\cdot(G+C-1)} \\
&\quad\times\frac{MTTF/S}{MTTR*(G+C-2)} \\
&= \frac{MTTF^3}{S^3(G+C)(G+C-1)(G+C-2)MTTR^2}
\end{aligned}
\tag{1}
$$

Here, $MTTF$ is for individual disk drive; S is a scale factor introduced to take into account of recent findings that manufacturer-provided MTTF is several factors off from the real world observation, D is the total number of disks with data; G is the number of data disks in a group; and C is the number of check disks in a group. Therefore, a RAID 6 MTTDL (Mean Time To Data Loss) is $\frac{MTTF_{group}}{N_g}$, where N_g is the number of disk groups. For a couplet in the Spider system, N_g is the total number of disks with data excluding check disks divided by the number of data disks in a group excluding check disks, which amounts to $N_g = 28*8/8 = 28$ disk groups.

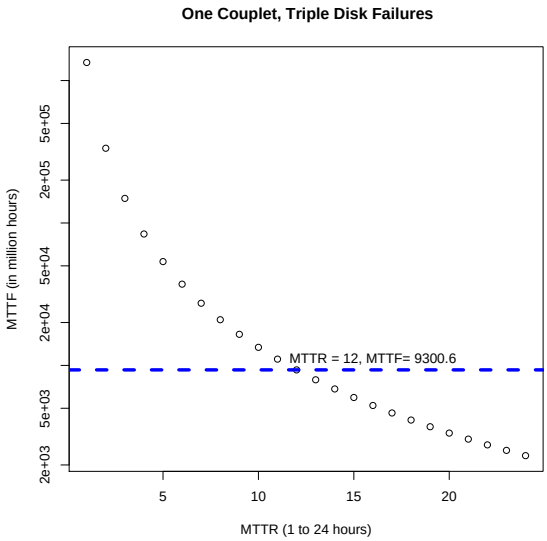


Figure 3.2 shows the MTTF of disk array within a couplet against the varying MTTR ranging from 1 hour to 24 hours. Naturally, MTTR has a significant impact on overall MTTF of the disk array. We assumed individual disk MTTF is 1.2 million hours and S is a conservative 4. If we consider the median value of MTTR to be 12 hours, as shown by the horizontal dashed line in the figure, the MTTF for disk array is an astonishing 9,300 million hours or approximately 1 million years. It should be clear that this idealistic picture only considered fail-stop mode for disks, denoted as triple disk failure case.

3.3 Uncorrectable Bit-Errors

Another source of failure is the so-called *uncorrectable bit-errors*, also known as "Non-recoverable Read Errors Per Bits Read" in disk manufacturer's data sheet. The usual number quoted is in the neighborhood of 10^{14}. A few studies have been conducted on this source of failure [2, 13], especially on double disk failure with single uncorrectable bit error, with assumption that the I/O is sector-based. We follow a similar line of reasoning and we consider the Spider system in particular. We also briefly discuss two other cases including one disk failure with double uncorrectable bit error and triple uncorrectable bit errors in the Spider system context. The probability of failure of reading all stripes on a disk, P_{error}, can be calculated as:

$$\begin{aligned} P_{error} &= P_{stripe\_error} \times N_{stripes} \\ &= \frac{S \times 512 \times 8}{10^{14}} \times \frac{D}{S} \end{aligned} \tag{2}$$

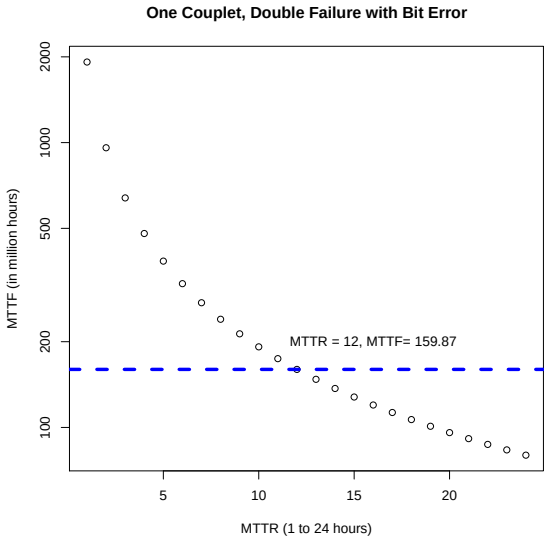
- P_{stripe_error}: probability of error of reading a RAID stripe.
- N_{stripe}: total number of stripes.
- D: disk capacity.
- S: RAID stripe size.

The above equation assumes that both disk capacity and RAID stripe size have a unit of sectors. Though DDN RAID stripes and cachelines are aligned on 1MB boundaries, as it shows, the RAID stripe size is elimated in the context. Failure probability is only related to bit error rate and the disk capacity. We take 1TB as disk capacity, this translates into that there is 8 percent of disk failures resulting in data loss can perhaps be attributed to the uncorrectable bit error. As the individual disk capacity continues to grow while the uncorrectable bit error rates remain steady for more than a decade now, the impact of this particular source of failure is only getting bigger.

Given that $P_{\text{error free}} = 1 - P_{error}$, RAID 6 (P+Q) failure due to a double disk failure plus uncorrectable bit error can be calculated by:

$$\begin{aligned} MTTF_{group} &= \frac{MTTF/S}{G+C} \cdot \frac{MTTF/S}{MTTR(G+C-1)} \cdot \frac{1}{1-(P_{\text{error free}})^{G+C-2}} \\ &= \frac{MTTF^2}{S^2 MTTR(G+C)(G+C-1)(1-(P_{\text{error free}})^{G+C-2})} \end{aligned} \tag{3}$$

The last item is the inverse of the probablity of at least one disk with a bit error anywhere on the disk within MTTR. Based on the Spider's configuration and (3), we can revisit the MTTDL for one couplet, but still using MTTR as the varying factor. The results show that with the increasing capacity of each drive while the uncorrectable bit error rate is not getting better, this source of failure can have a very dramatic impact on reliability. Taking the median value of 12 for MTTR as an example, the MTTDL is approximately 698 years on Spider. This is still a great number to have on reliability, but it is an order magnitude smaller than the triple disk failure, which implicates that this source of failure is far more common than the case of triple independent disk failure. Although the MTTDL number is indeed very high, it still translates into 1.4% of the probability that you will lose data in a 10-year operation period.



4 Reliability of a Couplet with Peripheral Components

As shown in Figure 1, there are three other major components besides disk array within our consideration: I/O module, DEM, and baseboard. Vendor supplied MTTF for these three components are shown in the following table. Our approach is to first define the *Spider failure*, then take into account of the physical layout illustrated by Figure 1 and deduce the reliability graph for the couplet system: the composite system is composed of a mix of series and parallel component connections based on the failure model we defined, and we can reach the estimation for the reliability of composite system (one couplet). First, we define the following cases to be considered as Spider system failures (note that more than two disk failure case is not considered here as it is encapsulated in the disk array reliability estimation):

Component	**MTTF**
I/O Module	1,263,856
DEM	1,552, 437
Baseboard	356,143

Case 1: If any two out of the five baseboards fail, then the system fails;

Case 2: If any three out of ten I/O modules fail, then the system fails;

Case 3: If any one baseboard fails, and another I/O module not connected with failed baseboard also fails, then the system fails.

Case 4: If any two I/O modules fail, then any other baseboard failure fails the system.

Case 5: If any three out of 10 DEMs (x4) fail, then the system fails.

Case 6: If a disk array fails, then the system fails.

We consider above failure definitions are mutually exclusive and exhaustive as a whole. With this failure model in place, we deduct the system reliability combinatorially using the following the notations:
- N_b: number of baseboards in total
- f_b: number of failed baseboards that cause system failure
- N_c: number of I/O modules (with Channel Controller) in total
- f_c: number of failed I/O modules that cause system failure
- N_d: number of x4 DEMs in total
- f_d: number of failed x4 DEMs that casue system failure
- $P_b(t)$,$P_c(t)$ and $P_d(t)$ represent probability of time to failure for baseboard, I/O module, and DEMs, respectively.

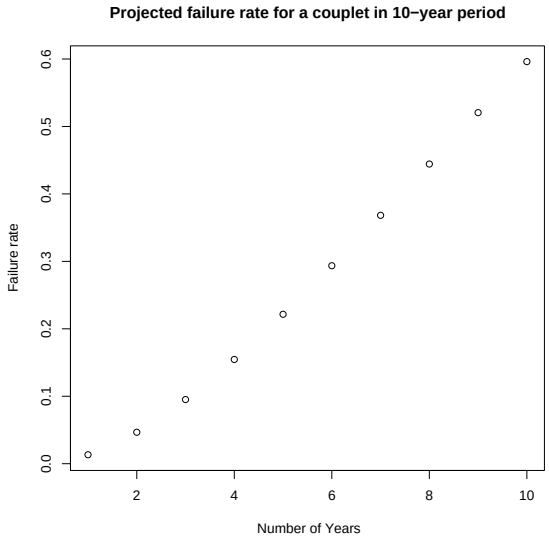


Figure 3: 10-year Projection of Failure Rate

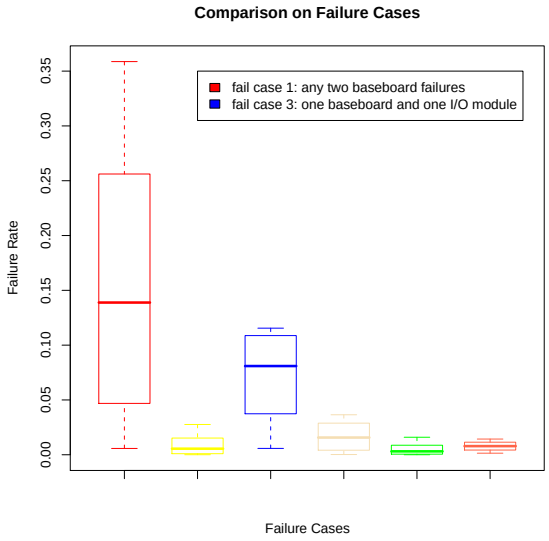


Figure 4: Failure Rate Distribution on Each Case

$$
\begin{aligned}
R(t) = 1 - \Bigg[& \sum_{i=f_b}^{N_b} \binom{N_b}{i} P_b(t)^{(N_b-i)} [1-P_b(t)]^i \\
& + \sum_{j=f_c}^{N_c} \binom{N_c}{j} P_c(t)^{(N_c-j)} [1-P_c(t)]^j \\
& + \binom{N_c}{1} P_c(t)^{N_c-1} (1-P_c(t)) \\
& \times \binom{N_b-1}{1} (1-P_b(t)) P_b(t)^{(N_b-1)} \\
& + \binom{N_c}{2} P_c(t)^{N_c-2} (1-P_c(t))^2 \\
& \times \binom{N_b}{1} (1-P_b(t)) P_b(t)^{(N_b)} \\
& + \sum_{k=f_d}^{N_d} \binom{N_d}{k} P_d(t)^{(N_d-k)} (1-P_d(t))^k \\
& + (1-P_{\text{disk array}}) \Bigg]
\end{aligned}
\tag{4}
$$

To calculate Eq. 4 based on the reliability estimates we have deducted so far, including those provided by the vendors, we have made an approximation of $p = \frac{\lambda}{MTTF}$, if $\lambda \ll 1$. The details on this approximation are shown in the appendix followed. Based on this failure model as well as known component MTTF listed in the table above, we can project the failure rate of one couplet over a 10-year period, as illustrated in Figure 3. We can also sketch the 10 year spread on each of the failure case in the form of the boxgraph in Figure 4. It shows that on average (indicated by the middle bar in the box), case 1 and case 3 have the most significant impact on the overall failure rate.

Conclusions

In this paper we introduced our reliability analysis on Spider disk subsystem. It is the backend disk system of one of the largest and fastest (possibly the largest and fastest at the time of writing this paper) POSIX-compliant parallel file system on world, serving the parallel I/O needs of the world's fastest open science parallel supercomputer, Jaguar at 1.6 PFLOPS. With more than 13,000 SATA II disks, 96 hardware RAID controllers, 10 PB of formatted disk capacity and 240 GB/s aggregate throughput, Spider really is a complex system to deploy, maintain, and of course to analyze, be it experimentally or analytically. Our effort was target to grasp a better understanding of what would be the expected reliability and failure rate in coming days once Spider is rolled into the full production mode.

To summarize, as a result of our work, we gained following insights on reliability of Spider:

- Though on average we can expect the number of disk replacements ranges anywhere from 98 to 393 per year (the later number considered the 4x scale factor observed from real world data), due to RAID 6 design and sensible monitoring and time to replacement, the probability of data loss due to triple disk failure is *extremely* rare, as shown in section 3.
- The disk failures combined with uncorrectable bit error that can cause overall data loss is more tangible than triple disk failure. However, we still have a very respectable overall reliability of disk array itself.
- The reliability of peripheral components present the most severe impact on the overall reliability, with baseboard being the weakest link. It contributes approximately 50% of the possible failure scenarios.
- For the first year, we can expect the number of failed couplets to be 0.64. However, by the end of year two, the number is approaching to 2.42, indicating a high probability that we will experience some couplet failures.

This work is just the beginning of our realiability analysis on Spider and we will update and correct our model with real-world data once the system is rolled into the full production mode. We also believe, our model, results, and findings will help the broader HPC community in terms expected disk subsystem complexity and reliability.

5 Future Work

Spider storage system is at its early stage of life: all hardware components have been recently procured and tested at the base level. The shake out of earlier defects have been done and the no-single-point-of-failure design has been verified. Several test filesystem have been created and benchmarked to evaluate the impact of different configurations and ready to go into full production. Going further, much elaborate data collection methods must be devised and put into place in order for us to develop a long-term and meaningful understanding of system failure as well as its reliability. We also need an effective correlation engine to gain valuable yet lacking insight on if and how hardware faults, system software and application I/O jobs are related.

Appendix

This appendix provides details on the approximation of $P(t) = \frac{t}{MTTF}$. The density function for a exponential distribution is:

$$f(t) = \lambda e^{-\lambda t}$$

Where λ is the rate of failure. The probability of an unit will fail between t_0 and t_1 is just the integral of $f(t)$ over that interval:

$$P(t_0 < \text{time for unit to fail} < t_1) = \int_{t_0}^{t_1} e^{-\lambda t} dt$$
$$= e^{\lambda t_0} - e^{-\lambda t_1}$$

If $t_0 = 0$, then first item is 1, and we have cumulative failure distribution:

$$P(\text{time for unit to fail} \leq t) = F(t) = 1 - e^{-\lambda t}$$

Also, if $\lambda t \ll 1$, then $e^x \sim 1 + x$, and we have:

$$F(t) = 1 - (1 + (-\lambda t)) \sim \lambda t$$

References

[1] John A. Chandy and Prithviraj Banerjee. Reliability evaluation of disk array architectures. pages 263–267, 1993.

[2] Peter M. Chen, Edward K. Lee, Garth A. Gibson, Randy H. Katz, and David A. Patterson. Raid: high-performance, reliable secondary storage. *ACM Comput. Surv.*, 26(2):145–185, 1994.

[3] Garth A. Gibson and David A. Patterson. Designing disk arrays for high data reliability. *Journal of Parallel and Distributed Computing*, 17:4–27, 1993.

[4] William V. Courtright II, Garth Gibson, Mark Holland, LeAnn Neal Reilly, and Jim Zelenka. RAIDframe: A Rapid Prototyping Tool for RAID Systems. `http://www.pdl.cmu.edu/RAIDframe/`, August 1996.

[5] Israel Koren. The RAID Tutorial. `http://www.ecs.umass.edu/ece/koren/architecture/Raid/raidhome.html`.

[6] Henry Newman. Linux File Systems: Ready for the Future? `http://www.enterprisestorageforum.com/sans/article.php/3749926`, May 2008.

[7] David A. Patterson, Garth Gibson, and Randy H. Katz. A case for redundant arrays of inexpensive disks (raid), 1988.

[8] Drew Robb. Choosing the Right High-Performance File System. `http://www.enterprisestorageforum.com/sans/article.php/3777326`, 2008.

[9] 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 *5th USENIX Conference on File and Storage Technologies*, 2007.

[10] Bianca Schroeder and Garth A. Gibson. Understanding Failures in Petascale Comptuers. In *SciDAC Workshop*, volume 78 of *Journal of Physics: Conference Series*, 2007.

[11] Martin Schulze, Garth Gibson, Randy Katz, and David Patterson. How reliable is RAID? In *Proceedings of COMPCON*, pages 118–123, 1989.

[12] North River Solutions. Calculating the True Reliability of RAID Systems. Technical report, 2007.

[13] Xuefeng Wu, Jie Li, and Hisao Kameda. Reliability Analysis of Disk Array Organizations by Considering Uncorrectable Bit Errors. In *The Sixteenth Symposium on Reliable Distributed Systems*, pages 2–9, 1997.