Scaling File Systems to Support Petascale Clusters: A Dependability Analysis to Support Informed Design Choices

Shravan Gaonkar[1], Eric Rozier[1], Anthony Tong[2], and William H. Sanders[3]

Coordinated Science Laboratory[1,3], Information Trust Institute[1,3],

National Center for Supercomputing Applications[2],

Department of Electrical and Computer Engineering[3], and Department of Computer Science[1]

University of Illinois at Urbana-Champaign

Email: {gaonkar, erozier2, whs}@uiuc.edu, atong@ncsa.uiuc.edu

Abstract

Petascale computing requires I/O subsystems that can keep up with the dramatic computing power demanded by such systems. TOP500.org ranks top computers based on their peak compute performance, but there has not been adequate investigation of the current state-of-the-art and future requirements of storage area networks that support petascale computers. Dependable scaling of an I/O subsystem to support petascale is not as simple as adding more storage servers. In this paper, we present a stochastic activity network model that uses failure rates computed from real log data to predict the reliability and availability of the storage architecture of the ABE supercomputer cluster at the National Center for Supercomputing Applications (NCSA). We then use the model developed to evaluate the challenges encountered as one scales the number of storage servers to support petascale computing. The results present new insights regarding the dependability challenges that will be encountered when building next-generation petascale supercomputers.

Keywords: Simulation, data analysis, modeling techniques, reliability and availability, storage systems.

Contact author: Shravan Gaonkar

1. Introduction

Historically, scientific computing has driven large-scale computing resources to their limits. Towards the end of the current decade we are likely to achieve petascale computing, a development that would bene-

fit many applications, such as climate and environmental modeling, 3D protein molecule reconstruction, aerospace engineering, and nanotechnology. While supercomputer performance has improved by over two orders of magnitude every decade, the performance gap between the individual nodes and the overall processing ability of an entire system has widened drastically [14]. This has led to a shift in the paradigm of supercomputer design, from a centralized approach to a distributed one that supports heterogeneity. While most high-performance computing environments require parallel file systems, there have been several file systems, such as GPFS [11], PVFS2 [17], and Lustre [3], that have been specifically proposed to support very large-scale scientific computing environments.

As the number of individual computing resources and components becomes very large, the frequency of failure of components within these clusters and the propagation of these failures to other resources are important concerns to high-performance computing applications. Failures can be caused by many factors: (a) transient hardware faults due to increased chip density, (b) software error propagation due to a large buggy legacy code base, or (c) manufacturing defects and environmental factors such as temperature or humidity.

Recent literature on failure analysis of BlueGene/L discusses various causes of increased downtime of supercomputers [7]. It has been well-established that elimination of failures is impossible; it is only feasible to circumvent failures and to mitigate their effects on a system's performance. The standard approach to the mitigation of a failure is to checkpoint the application at regular intervals. Long et al., however, showed that checkpointing has a large impact on the performance of very large high-performance computers with large numbers of nodes [16]. In particular, they were able to estimate that more than half the computation time would be spent checkpointing the application state due to the time spent in transferring the application state to the persistent storage.

Increasing the number of compute servers in a cluster almost always increases the size of the desired storage subsystem. Depending on the type of parallel file system, that means an increase in the number of file servers that could accept requests from the compute servers to keep up with I/O requests. Compute servers and file servers have very different characteristics. First, a failure in a file server needs more attention than a failure in a compute node. A compute server might just be marked as unavailable until it is repaired, but a failed file server might have to be reconstructed, or its state might need to be transferred to another file server depending on the replication strategy. Second, file servers are inherently slower due to their I/O characteristics. This generally makes file servers the bottleneck for the reliability and performance of a

cluster. Unfortunately, there has been a trend towards increasing failure rates for I/O subsystems that is similar to that for overall petascale clusters. This increase in failures can be attributed to the usual increase in the number of individual components that are needed to make the whole I/O subsystem work. Recent studies have shown that workload intensity is highly correlated to the failure rates [12, 15]. That emphasizes the need for thorough analysis to understand the impact of the I/O subsystems and their failures on petascale computers, as the I/O subsystem is one of the primary bottlenecks in high-performance computing.

To address the research challenge of providing realistic prediction of petascale file system availability, we take a two-pronged approach. First, we have obtained the failure event log of the ABE cluster from the National Center for Supercomputing Applications (NCSA). The log contains the failures of individual nodes, file server nodes, and the storage area network (SAN). We preprocessed the event logs to determine various reward measures of interest corresponding to the file system, such as the availability of the file system over the lifetime of the log and the failure rate of jobs due to I/O failures and other transient failures. Then, we built and refined stochastic models of the file system used by these clusters that abstracts much of the operations, while generating reward measures that are comparable to the real log events. We then scaled the models to reflect the scale and magnitude of a future petascale computer and estimated the impact of current I/O and file system designs on a petascale computer. Furthermore, we evaluated strategies that could be used to mitigate the bottlenecks due to scaling of I/O file system and cluster designs from current supercomputers to petascale computers. Our analysis will give storage architects support to make informed design choices as they build larger cluster file systems.

The rest of the paper is organized as follows. Section 2 provides a brief overview of the cluster file system (CFS). Section 3 discusses the file system architecture of the ABE cluster at NCSA with the analysis of collected failure log files. Section 4 presents the conceptual stochastic activity network model of the ABE cluster. Section 5 covers results and analysis. Section 6 outlines related work. Section 7 offers our conclusions and plans for future work.

2. Cluster File System (CFS) Architecture

Typical storage architecture for a cluster file system consists of a *metadata server*, multiple *file servers*, and *clients* [3]. The metadata server maintains the file system's metadata, which includes the access control information, mapping of files and directory names to their locations, and mapping of allocated and free space. The metadata server serves the metadata to the clients. The file servers maintain the actual data and information about the file blocks stored on the connected I/O disks and serve these file blocks to the

clients. For reliability/ performance, the file blocks can be replicated/striped over multiple disks. The client communicates first with the metadata server and then with the appropriate file server to perform the read and write operation.

2.1. Metadata Server

The *metadata server* maintains the state of the entire file system. The organizational structure of a single central server could be the main bottleneck for the failure of the file system. However, the bottleneck can be mitigated by internal replication of the node, maintenance of an operation log to reconstruct upon failure, or other strategies. In the case of IBM's GPFS, the metadata can be distributed and handled by a pool of file servers [11]. The metadata server node receives client requests for the location of the file blocks. The metadata server replies with the corresponding location of the file blocks on the file servers.

2.2. File Server

The *file servers* or the *object storage servers* (OSS) are dynamic resources. New file servers can be added. Failed servers are purged automatically. Each file server can be as simple as a Linux file server running on commercial off-the-shelf systems or a dedicated enterprise-class storage server. The file server nodes can further augment reliability mechanisms, using techniques such as RAID, to protect the content on the individual servers. The throughput of a file server depends on the network backbone connecting to the compute node clients and the internal I/O capacity of the server.

2.3. Client

The client nodes (compute nodes) proxy the requests on behalf of the user or applications. Modern cluster file systems cache the requested files on the client side and only transmit or propagate write requests.

3 ABE Cluster: System Configuration and Log File Analysis

The ABE cluster architecture is the current state-of-the-art. ABE consists of 1200 blade compute nodes, i.e., 9600 core CPU Intel 64 (2.33 GHz dual-socket quad-core) processors, 8/16 GBytes shared RAM per node, and an InfiniBand (IB) interface. The cluster runs Red Hat Enterprise Linux 4 (Linux 2.6.9) as its operating system. The cluster can provide a peak compute performance of 89.47 PFLOPS. The Lustre file system supports a 100TB parallel cluster file system for the ABE cluster's compute nodes[2].

Lustre-FS outage time			
Cause of Failure	Start time	End time	Hours
I/O hardware	07/21/07 23:03	07/22/07 12:00	12.95
I/O hardware	07/31/07 01:49	07/31/07 20:01	18.18
I/O hardware	08/22/07 18:08	08/23/07 02:15	08.12
I/O hardware	08/28/07 16:20	08/29/07 18:01	01.67
I/O hardware	09/25/07 18:00	09/26/07 09:30	15.50
I/O hardware	10/04/07 09:30	10/04/07 21:55	12.42
Batch system	10/16/07 17:56	10/16/07 21:24	03.47
Network	10/29/07 11:53	10/29/07 15:15	03.36
File system	11/16/07 09:30	11/16/07 10:00	00.40
File system	11/19/07 09:04	11/19/07 11:00	01.93

Table 1. User notification of outage of the Lustre-FS

3.1. ABE CFS Server Hardware

The ABE Lustre-FS is currently supported by 24 Dell dual Xeon servers that provide 12 fail-over pairs[1]. One OSS serve the metadata of the Lustre-FS, 8 OSSes serve the /cfs/scratch OSS, and the remaining 6 servers handle the remaining partitions of the shared file systems (home, local, usr, etc.) of the cluster. Each server self-monitors its file system's health. The 2 metadata OSSes are connected to the storage I/O through a dual 2Gb fiber channel (FC).

3.2. ABE CFS Storage Hardware

Scratch partition: 2 S2A9550 storage units, from DataDirect Networks Systems, provides the storage hardware for the CFS's scratch partition. Each S2A9550 supports 8 4Gb FC ports. Each port connects to 3 tiers of SATA disks. Each tier has (8+2) disks in RAID6 configuration. Therefore, there are 480 disks, each with a 250GB capacity, that form the scratch partition providing 96TB of usable space.

Metadata: DDN EF2800 provides the I/O hardware to support the metadata of the Lustre-FS. It is connected to the 2 metadata OSSes through a dual 2Gb fiber channel. The EF2800 has one tier of 10 disks in RAID10 configuration.

Other partitions: 10 IBM DS4500s server an approximate total of 40T of usable space over a SAN via 2Gb FC.

Lustre settings: Lustre version 1.4.10.X runs on all of the OSS's hardware. Most of the reliability is provided by the SAN hardware; therefore, the Lustre reliability features are switched off.

[1]We refer to a fail-over pair as an *OSS* in the reminder of the paper.

Date	2	Date	4	Date	6
07/03/07	102	07/19/07	258	08/16/07	375
08/20/07	591	09/05/07	005	09/17/07	002
09/18/07	004	09/19/07	003	09/28/07	463
09/29/07	477	10/01/07	051	10/02/07	035

Table 2. Lustre mount failure notification by compute nodes from 07/01/07 to 10/02/07. Column 2, 4, and 6 are the number of compute nodes that experienced mount failure

Total jobs submitted between 05/13/07 to 10/02/07	44085
Total failures due to transient network errors	1234
Total failures due to other/file system errors	0184

Table 3. Job execution statistics for the ABE cluster

3.3. ABE Log Failure Analysis

All NCSA clusters have elaborate logging and monitoring services built into them. The log data set used in this study was collected from 05/03/2007 to 10/02/2007 for compute nodes (compute-logs) and 09/05/2007 to 11/30/2007 for the SAN (SAN-logs). The compute-logs and SAN-logs are monitored precisely, and the logs provide details about the events taking place in the cluster. Events are reported with the node IP addresses and the event time appended to the log information. To extract accurate failure event information, we filter failure logs based on temporal and causal relationships between events.

Table 1 provides the availability of the ABE cluster based on the notifications provided by the SAN administrators to the users [1]. The availability of ABE's SAN can be estimated to be between 0.97 and 0.98 depending on the dates one chooses as the start and end times for the measure computation. Table 2 shows Lustre-FS mount failures experienced by individual compute nodes aggregated on a per-day basis. Lustre-FS mount failures does not always imply the failure of the CFS as these errors could be caused due intermittent network unavailability. Nevertheless, those errors are perceived as failures from the cluster's perspective.

Table 3 presents the job failure/completion statistics obtained by analyzing the compute-log. The analysis shows that the transient errors causing network unavailability (between the compute nodes and the CFS or between the compute nodes and the login nodes) are 5 times more likely to cause job failures that other errors are (such as software errors, or CFS failures). Earlier clusters had dedicated backplanes connected to compute nodes to provide communication. Current communication in ABE is through COTS network ports and switches. The change in the design choice is mostly intended to lower costs and increase flexibility in maintaining the system.

Table 4 provides the disk failure and replacement log from 09/05/2007 to 11/28/2007 for disks that support

Dates in September 2007	05	06	09	13	23
Number of failed disks	2	1	1	1	1
Dates in October 2007	08	17	24		
Number of failed disks	2	1	1		
Dates in November 2007	08	17			
Number of failed disks	1	1			

Survival analysis of the disk failures ($n = 480$) using Weibull regression (in log relative-hazard form) gives the shape parameter as 0.6963571 with standard deviation of 0.1923109 (95% confidence interval) [5]

Table 4. Disk failure log from 09/05/2007 to 11/28/2007 for disks supporting the ABE's scratch partition

the scratch partition of the ABE's cluster. The authors of [13] estimated the disks' hazard rate function to be statistically equivalent to a Weibull distribution. We performed similar survival analysis on the disk failure data and found that Weibull with $\beta = 0.7$ was a good fit for ABE's disk drive failure logs. The key insights we gained from analyzing failure data and from discussions with cluster system administrators are as follows:

- The disk replication redundancy and replacement have been so well-streamlined that they almost never cause catastrophic failure of the CFS. On average, 0-2 disks are replaced on the ABE cluster per week.

- The ABE cluster's S2A9550 RAID6 (8+2) technology combines the virtues of RAID3, RAID5, and RAID0 to provide both reliability and performance [8]. RAID6 prevents a second drive failure from occurring during disk re-mirroring. The *Blue Waters* petascale computer, which will be built at the University of Illinois, will likely have an (8+3) RAID configuration. That would make the failure of the file system due to multiple individual disk failures highly unlikely.
- Most file system failures are due to software errors, configuration errors, and other transient errors. The software errors take, on average, 2-4 hours to resolve. Most often, the fix is to bring the disks to a consistent state using a file system check (*fsck*). A hardware failure due to a network components or due to a RAID controller might take up to 24 hours to resolve, as these components need to be procured from a vendor.

4. Stochastic Activity Network Model: Cluster File System

The failure data analysis and the insights provide the details necessary to build a stochastic model of the ABE's cluster file system. Here, we describe the details of the stochastic activity network models using Möbius [4].

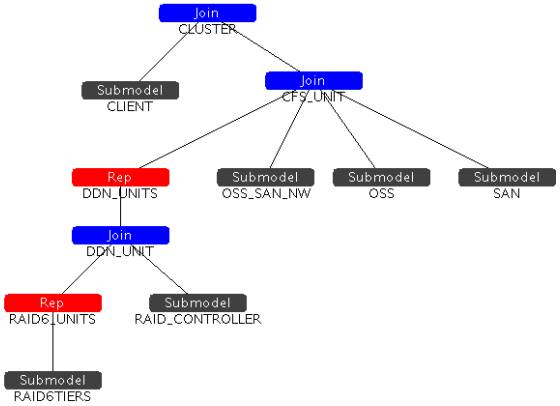

Figure 1. Compositional model of the Cluster File System.

4.1. Overall Model

Figure 1 shows the *composed model* of the ABE cluster using replicate/join composition in Möbius. The leaf nodes in the replicate/join tree are stochastic activity network models which implement the functionalities. Space limitations does not permit detailed descriptions of these submodels. The CLUSTER model has two main submodels connected using a join where the models share states on error propagation from CLIENT to the CFS. The CLIENT represents the behavior and interaction of the compute nodes and the communication network between the compute nodes and the CFS. The CFS_UNIT emulates the ABE's cluster file system. It is composed of the OSS, OSS_SAN_NW, SAN, and the DDN_UNITS. The OSS implements the availability and operational model of the metadata server and the file server. The OSS_SAN_NW implements the failure model of the network ports and switches that connect OSS to the DDN_UNITS. The SAN emulates the operations provided by the network to communicate between OSS and the DDN_UNITS. The OSS, OSS_SAN_NW, SAN and the DDN_UNITS communicate by sharing information about their current state of operation and availability. The DDN_UNITS composes multiple RAID6_UNITS with RAID_CONTROLLER. The failure of disks in RAID6_UNITS is assumed to follow a Weibull distribution. RAID_CONTROLLER emulates the failure and operation of a typical RAID6 architecture. The DDN_UNITS is replicated to emulate multiple S2A9550 units.

Since the goal is to investigate the impact of availability of file systems to petascale computers, the stochastic activity network models do not consider hardware failure in compute nodes. Our model incorporates only the behavior of the scratch partition and the metadata servers of the CFS, because a clusters utility depends mainly on its scratch partition's availability. Finally, hardware and software misconfiguration errors occur in the early deployment phase of the system; therefore, we exclude them from the models. In the

Model parameter	Values (range)
Disk MTBF [2]	100000-3000000
Annualized Failure Rate (AFR)	0.40%–8.6%
Weibull distribution's shape parameter [1]	0.6–1.0
Number of DDN [1]	2–20
Number of compute nodes [1]	1200–32000
Average time to replace disks [3]	1–12 hours
Average time to replace hardware [3]	12–36 hours
Average time to fix software [3]	2–6 hours
Job request per hour [1]	12–15 per hour
Hardware failure rate [1]	1–2 per 720 hours
Software failure rate	1–2 per 720 hours
Annual growth rate of disk capacity [2]	33%
DDN_Units [1]	2–20
OSS Units [1]	8–80

Parameter values obtained from: log file analysis [1], data specification from literature and hardware white papers [2], discussions with NCSA cluster administrators[3]

Table 5. ABE cluster's simulation model parameters

following subsections, we describe the reward measures and the failure model used to represent the ABE's CFS.

4.2. Reward Measures

The **availability** of the cluster file system is defined as the ability of the CFS to serve the client nodes. More precisely, it is defined as the fraction of time when all the file server nodes (OSSes), the DDN, and the network interconnect between the OSSes and the DDN are in the *working* state.

The **disk replacement rate** is defined as the number of disks that need to be replaced per unit of time to sustain the maximum availability of the CFS.

The **cluster utility**, **CU**, is an availability metric from the cluster's perspective. Precisely, it is defined as $CU = \left(1 - \frac{Compute\ cycles\ lost\ due\ to\ unavailable\ file\ system}{Total\ available\ compute\ cycles}\right)$. CU is a metric different from availability metric of the CFS[1]. The cluster users and a SAN administrators tend to notice different levels of availability. This reason is failures in network communication between the compute nodes and the CFS as well as failures due to intermittent transient errors make CFS appear unavailable even though it has not failed.

4.3. Failure model for ABE's CFS

The ABE's cluster suffers from failures mainly because of 3 types of errors: hardware errors, software errors, and transient errors. Each kind of error affects all the CFS's components.

The **hardware errors** in the metadata/file servers (OSSes) occur in the hardware components that are

[1]CU does not distinguish between compute cycles used to perform checkpointing and those used for actual computation.

built to operate the system. These errors include processor, memory, and network errors. Hardware errors are assumed to be less frequent than disk failures, occurring at the rate of 1–2 per month. The RAID controllers in the DDN or network ports/switches that connect DDN to OSS show similar failure rates. The repairs of these components takes 12–36 hours depending upon the severity of the failure (as reported by SAN administrators), as the needed replacement parts have to be shipped from the vendors. Most of the hardware is replicated with fail-over mechanisms. Failure of both members of the fail-over pair causes the unavailability of the CFS system. The replacement of failed disks is modeled as a deterministic event. The repair time is varied between 1 to 12 hours across simulation experiments.

The **software errors** that cause failure of the cluster file systems are mainly due to the corrupted supercomputing applications running on the compute nodes (implemented in the CLIENT submodel) or the Lustre-FS (implemented in the OSS submodel). Since we do not have accurate estimates on software corruption errors, we assume that the rates are similar in the orders of magnitude to hardware error rates. The repair times for software errors modeled as deterministic events. The repair time is varied between 2 to 6 hours across simulation experiments.

Transient errors occur in most components of the cluster model, but mainly in the network components. The error rates are obtained from the failure-log analysis as shown in Table 3. Transient errors are temporary, but hard to diagnose. Our model assume that one of these errors causes a few minutes of unavailability of components under transient failure. The jobs depending on those components fail due to the temporary unavailability.

Past literature has emphasized the importance of modeling **correlated failures** [15]. Most correlated errors occur because of shared resources. Correlated errors propagate to components that have causal or spatial proximity. In the CFS model, hardware errors propagate because other hardware components are connected to each other. Software errors propagate from compute nodes to OSS or from OSS to disk, leading to data corruption. Transient errors propagate errors into software. All failures except disk failures are modeled as exponential distributions. To model correlated failures, we assume that there is small probability, p, that errors can propagate to other connected components.

5. Experimental Results and Analysis

We evaluate the design of the ABE cluster's availability using simulation in the Möbius tool. Table 5 consolidates the parameters collected through failure log analysis, hardware reliability specifications, and discussions with cluster administrators. We scale the parameters values to reflect the size and scale of a

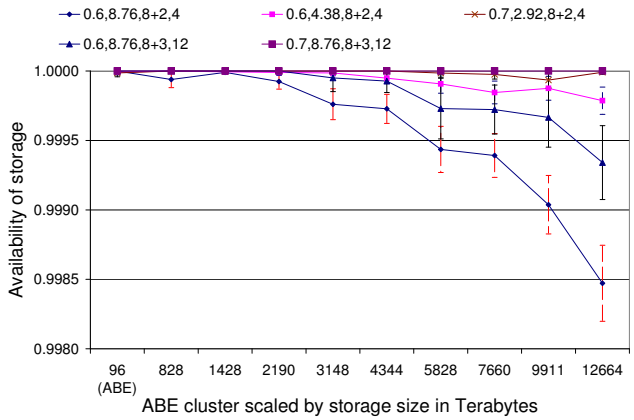

Figure 2. Availability of storage with respect to disk failures. Label with values (0.7,2.92,8+2,4) represents a tuple =(Weibull shape parameter β, AFR in %, RAID configuration, average disk replacement time in hours)

petascale computer, and determine the factors that impede the high availability of the CFS. All the simulation results are reported at 95% confidence level, with intervals.

5.1. Impact of disk failures on CFS

To evaluate the baseline effect of failures of disks on availability of the CFS, we evaluate the DDN_UNITS models associated with the RAID6 tiers and the RAID controllers in isolation from failures of other components of the SAN. Figure 2 shows the availability of the storage hardware as one scales the file system from the current 96TB (supporting the ABE's scratch partition) to 12PB (supporting the petascale Blue Waters computers). The key observation is that the RAID6 architecture provides sufficient redundancy and recovery mechanisms such that the impact of high disk failure rates is mitigated to a very large extent. First, note that all configurations of failure and recovery rates for an ABE sized cluster file system have nearly 100% availability (refer to first data point in Figure 2). However, as the experiments are scaled from the ABE's system to a petascale system, our simulation results show that the RAID6 architecture cannot provide the same level of storage-availability for some of the failure model configurations. The SAN architect's future vision to use (8+3) RAID in Blue Waters is important; it provides better reliability than the (8+2) RAID on petascale systems. While RAID6 provides a larger margin for disk failure rates, i.e., up to 8.6% AFR, it is very important that these rates be contained to lower thresholds by disk manufacturers and vendors to provide the adequate level of availability. If one makes a pessimistic assumption of a higher infant mortality rate in disks (Weibull shape parameter = 0.6), the availability falls below 99.9% for petascale storage.

To better understand the cost of disk replacement, we compute the expected number of disks that need to be replaced per week for the RAID6 tiers. Figure 3 depicts the average number of disks that needs to be

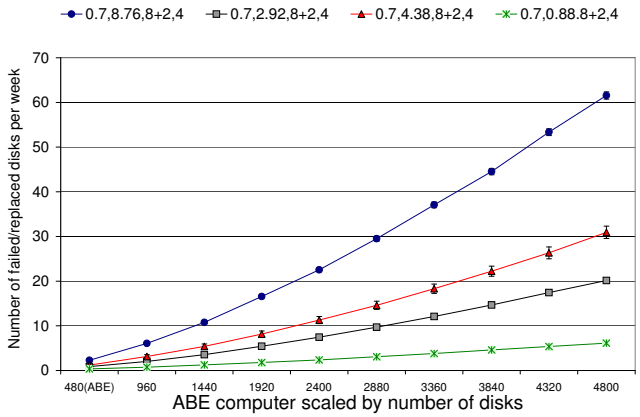

Figure 3. Average number of disks that need to be replaced per week to sustain availability

replaced per week to sustain the availability so that the CFS does not suffer failure due to RAID6 failure. The configuration (0.7,2.92,8+2,4) corresponds to the ABE cluster with 0 to 2 disk replacements per week. Each time a disk fails, there is an operational cost (in dollars) that is borne by the SAN vendors as they provide extended support to their SANs. As the CFS system is scaled to support petascale computers, the number of disks that need to be replaced increases, increasing the labor cost and the replacement cost. Therefore, the SAN vendors have an incentive to increase the disk MTBF to reduce their overall support cost.

Survival analysis of the disk failures data provided a good estimate of the Weibull distributions's shape parameter β, but the estimate for the scale parameter (MTBF) was insignificant [5]. Using simulations, we estimated MTBF that matched the average disk failures per week for the scratch partition and determined MTFF=300,000 hours or annualized failure rate (AFR)=2.92% to be a good fit. These parameter values were used to evaluate petascale system's reward measures.

5.2. CFS availability and CU

To analyze the impact of all components that determine the availability of the CFS, we evaluate the availability and CU of the ABE system (refer to Section 4.2 for the definition of the reward measures). The experiments are scaled to the size of a petascale computer to allow understanding of the impact of failures on those measures. Figure 4 shows that the CFS-availability decreases as one scales the system to support a petascale computer. Since most of the parameter values were obtained through the log data analysis, and times reported by SAN administrators, our measures for CFS-availability matched with ABE's availability as shown in Table 1. Therefore, we have higher confidence of the measures of availability and CU as we scaled the models to represent a petascale computer with a petabyte storage system. The storage-availability in Figure 4 refers to configuration (0.7,2.92,8+2,4), which models the ABE cluster's current environment. We find that the RAID6 subsystem in this configuration continues to provide an availability of 1, but the

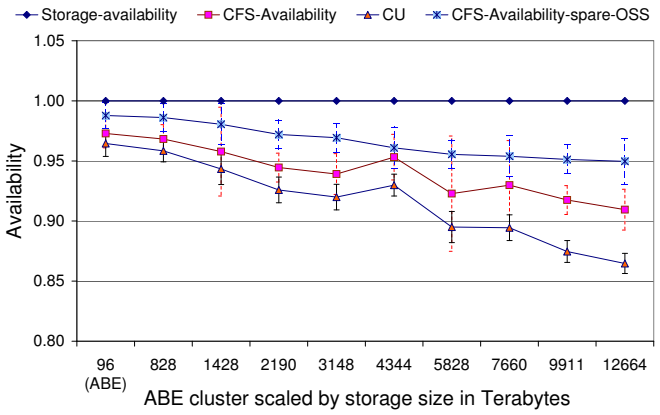

Figure 4. Availability and utility of the ABE cluster when scaled to petaflop-petabyte system

CFS-availability is reduced from 0.972 to 0.909 as one scales the design to support the petascale system. The reduction is mainly due to correlated failures in OSS and hardware. Improving upon ABE's design, the architect could provide an additional standby-spare OSS that can replace the failed OSS. Our evaluation shows that this approach can improve the availability by 3%. Unlike the RAID6 redundancy architecture for reliability, designing fault-tolerant OSS is both technically challenging and expensive due to the complexity of the OSS system. To improve the availability further, the architects have to develop solutions to mitigate correlated errors. For example, improving the robustness of the Lustre-FS can reduce the software-correlated errors. The CU in Figure 4 shows that the clusters network architecture between compute nodes and the CFS has a profound impact on the cluster utility available to the users. The trend to move away from customized backplanes to COTS network hardware (with its complicated software stack) has decreased the CU. The transient errors seen in the network can be mitigated by providing multiple network paths between the compute nodes and CFS.

6. Related work

Cluster model analysis: Past literature describes several attempts to model and analyze different aspects of large-scale supercomputing systems. Wang et al. looked at the impact on the system performance in the presence of correlated failures as the systems are scaled to several hundred thousand processors [16]. Rosti et al. presented a formal model to capture CPU and I/O interactions in scientific applications, to characterize system performance [10]. Oliner et al. investigated the impact of checkpointing overhead using a distribution obtained from a real BlueGene/L log [9]. Some literature has discussed the importance of distributing data across multiple disks to improve performance and reliability of a file system [6]. While [6, 9, 16] have evaluated the cluster from a performance viewpoint (mostly focused on checkpointing and its overhead), our analysis, backed with real cluster failure data, isolates the file system and models its impact

on designs of larger cluster file systems to support petascale computers.

Failure model analysis: The estimation and prediction of failure of file systems are crucial to understanding the overall performance of petascale computers. Recent literature has shown that storage subsystems are prone to higher failure rates than their makers estimate because of underrepresented disk infant mortality rates [13]. Schroeder and Gibson studied the failure characteristics of large computing systems to find that failure rates are proportional to the size of the system and are highly correlated with the type and intensity of the workload running on it [12]. While Liang et al. investigated the failure events from the event logs from BlueGene/L to develop failure prediction models to anticipate future fatal failures [7], our approach builds accurate structural and operational models using the failure data from the logs of the cluster and discussions with the SAN architects to provide finer details about current systems with insights into future cluster file system design.

7. Conclusion and Future Work

Many researchers have focused on developing and understanding reliability of clusters for supercomputing applications. In our paper, we have taken steps to understand the reliability and availability of the ABE cluster through failure data analysis and discussions with administrators at multiple levels of the cluster operation, starting from the lowest level of the SAN's availability, to cluster's availability, and then to user/job perception of cluster utility. Our key findings through analysis and simulation showed that the RAID6 design for a disk's reliability has limited the impact of disk failures on the CFS, even when the model is scaled to evaluate the support for petascale system. On the other hand, transient errors, hardware errors, and software errors contribute significantly to failures, and these components are the limiting factors for the high availability of the CFS. We believe that petascale architects will have to focus on these issues to develop solutions to improve the overall availability of the CFS.

Future work: Our work has mainly focused on evaluating the availability of the ABE's CFS through data collection, analysis, and system modeling. We showed that system modeling combined with data analysis from real systems provides better intuition for designing future systems. NCSA has other operational clusters such as Mercury and Tungsten, with different architectures. There is an opportunity to evaluate and analyze these systems to determine how the ABE architecture was designed based on the lessons learned from the older clusters. That analysis could provide insight into future CFS architectures on petascale systems. The models and data analysis (on larger failure data sets) can be extended to evaluate performance metrics that would complement the reliability measures conducted in this research.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Number CNS-0406351 and a generous gift from HP Labs. The authors would like to thank the security, storage and other affiliated groups at NCSA for their inputs, log files data and time that enabled us to consolidate this practical report. We would also like to thank Jenny Applequist for her editorial comments.

References

[1] Notifications to teragrid users about CFS unavailability `http://news.teragrid.org/user.php?cat=ncsa`.

[2] NCSA ABE cluster technical specification `http://www.ncsa.uiuc.edu/UserInfo/Resources/Hardware/Intel64Cluster/TechSummary/`.

[3] R. L. Braby, J. E. Garlick, and R. J. Goldstone. Achieving order through CHAOS: The LLNL HPC Linux Cluster Experience. In *The 4th International Conference on Linux Clusters: The HPC Revolution 2003*, San Jose, CA, USA, 2003.

[4] D. Deavours, G. Clark, T. Courtney, D. Daly, S. D. J. Doyle, W. H. Sanders, and P. G. Webster. The mobius framework and its implementation. *Transactions on Software Engineering*, 28(10):956–969, Oct 2002.

[5] D. W. Hosmer and S. Lemeshow. *Applied Survival Analysis: Regression Modeling of Time to Event Data*. John Wiley & Sons, Inc., New York, NY, USA, 1999.

[6] K. Hwang, H. Jin, and R. S. Ho. Orthogonal Striping and Mirroring in Distributed RAID for I/O-centric Cluster Computing. *IEEE Transaction in Parallel and Distributed Systems*, 13(1):26–44, 2002.

[7] Y. Liang, Y. Zhang, A. Sivasubramaniam, M. Jette, and R. Sahoo. BlueGene/L Failure Analysis and Prediction Models. In *Proc. of DSN '06*, pages 425–434, 2006.

[8] L. McBryde, G. Manning, D. Illar, R. Williams, and M. Piszczek. Data Management Architecture. Patent number 7127668, Oct 24 2006.

[9] A. J. Oliner, R. K. Sahoo, J. E. Moreira, and M. Gupta. Performance implications of periodic checkpointing on large-scale cluster systems. In *Proc. of IPDPS '05*, page 299.2, Washington, DC, USA, 2005.

[10] E. Rosti, G. Serazzi, E. Smirni, and M. S. Squillante. Models of parallel applications with large computation and I/O requirements. *IEEE Transaction in Software Engineering*, 28(3):286–307, 2002.

[11] F. Schmuck and R. Haskin. GPFS: A shared-disk file system for large computing clusters. In *Proc. of FAST '02*, pages 231–244, 2002.

[12] B. Schroeder and G. A. Gibson. A large-scale study of failures in high-performance computing systems. In *Proc. of DSN '06*, pages 249–258, Washington, DC, USA, 2006.

[13] B. Schroeder and G. A. Gibson. Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? In *Proc. of FAST '07*, pages 1–16, 2007.

[14] E. Strohmaier, J. J. Dongarra, H. W. Meuer, and H. D. Simon. The marketplace of high performance computing. *Parallel Computing*, 25(13–14):1517–1544, 1999.

[15] D. Tang and R. K. Iyer. Dependability measurement and modeling of a multicomputer system. *IEEE Transaction in Computers*, 42(1):62–75, 1993.

[16] L. Wang, K. Pattabiraman, Z. Kalbarczyk, R. K. Iyer, L. Votta, C. Vick, and A. Wood. Modeling coordinated checkpointing for large-scale supercomputers. *In Proc. of DSN '05*, pages 812–821, 2005.

[17] W. Yu, S. Liang, and D. K. Panda. High performance support of parallel virtual file system (pvfs2) over quadrics. In *Proc. of ICS '05*, pages 323–331, 2005.