Substituting Disk Failure Avoidance For Redundancy In Wide Area Fault Tolerant Storage Systems

Christopher Brumgard, Micah Beck
Dept. of Electrical Engineering and Computer Science
University of Tennessee
Knoxville, TN, United States
e-mail: {sellers,mbeck}@eecs.utk.edu

Abstract— The primary mechanism for overcoming faults in modern storage systems is to introduce redundancy in the form of replication and/or error correcting codes. The costs of such redundancy in hardware, system availability and overall complexity can be substantial, depending on the number and pattern of faults that are handled. In this paper, we describe a system that seeks to use disk failure avoidance to reduce the need for costly redundancy by using adaptive heuristics that predict such failures. While a number of predictive factors such as hard drive utilization rate, age, SMART errors, and model can be used, the initial work we present here focuses on SMART errors. Our approach can predict where near term disk failures are more likely to occur, enabling proactive movement/replication of at-risk data, thus maintaining data integrity and availability. Our strategy can reduce costs due to redundant storage without compromising these important requirements.

Keywords-component; hard drives; logistical networking; SMART errors

I. INTRODUCTION

The amount of data in the world is exponentially increasing every year and magnetic hard disk storage has become the preferred medium for preserving and accessing data. While the storage capacity of hard disks has been following Kyder's Law such that disk areal storage density increases 40% annually, the reliability factor and IO bandwidth of disks have not been keeping pace making it significantly more difficult to preserve and maintain data. The ever-increasing amount of data introduces a huge management problem [1-3].

Combining big data with many distributed users/resources creates severe problems for the scientific community. As the amount of data increases traditional techniques are not scaling well and are economically infeasible to work with [2, 3]. In the past, the scientific community has relied on constant hardware improvements to keep pace with their data processing requirements, but we are entering a critical intersection of Kyder's and Moore's Laws.

While we may have sufficient space to store data, it is going to take a herculean effort to maintain and use the data effectively. Scientific data must be managed constantly and replicated to protect against loss while at the same time being available for processing by users across the globe. Both of these goals require significant processing power and IO bandwidth and these are resources that the scientific community cannot afford to waste.

This paper examines how adaptive data replication and placement heuristics based on hard drive health prediction can improve the resilience of data in Logistical Networking. Utilizing adaptive replication heuristics based on hard drive SMART errors, greater survivability of data can be achieved than with traditional multisite/multisystem replication techniques.

Adaptive replication can make better economic use of hard drives by anticipating when drive failures are likely to occur. Adaptive replication can match perceived reliability of hard drive storage with the criticality of data. Currently drives are replaced when they reach a certain predefined criteria (age, utilization, errors). Adaptive replication can predict which data needs to be migrated to other disks while using the less reliable disk to store less essential or more replicated data. Therefore making it possible to use hard drives until they have complete failure.

II. DISK FAILURE

We have reviewed the literature over the past decade and a half in an effort to understand what important factors these studies have uncovered [4-13]. Despite the importance of the hard disk drive there has been very little large-scale work analyzing their failure trends. Among the factors found to influence reliability are age, utilization, temperature, vintage, failure correlation and SMART errors. While no one has been able to make a perfect predictor of drive failure, we tested whether the prediction of hard drive health based on SMART errors is sufficient to reduce data loss via intelligent data placement and replication.

In recent years some meaningful data and trends have begun to arise from large populations sets. Large-scale user studies are rare because of the number of hard drives that would be required combined with the time and ability to analyze the data. Also users often worry about disclosing data that could be used due to non-disclosure agreements with vendors. The largest study to date was performed by Google and published in "Failure Trends in a Large Disk Population" [9]. Our work is based on the observations in this paper.

In the paper, over 100,000 hard drives were studied using hardware logs that spanned 5 years and 9 months of study on the effects of SMART errors. It was a diverse population of serial and parallel ATA hard drives ranging from 80 to 400

978-0-7695-4844-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ClusterW.2012.39

IEEE computer society

GB in capacity and included several different manufactures and models.

Of particular interest to us were the findings concerning SMART errors which we will summarize. Of all of the error types; scan errors, reallocation counts, offline reallocation and probational counts have the largest impact on failure.

Scan errors were witnessed in roughly 2% of the population and resulted in disks with a 10 times higher annualized failure rate (AFR), 30% failure rate within 8 months of first scan error, and 39 times more likely to fail within 60 days. Younger drives tended to fail more frequently after this error in the first month than older drives. Drives with more than one scan error were also more likely to fail than drives with just one.

Over the 9 months, 9% of the population suffered allocation errors. While not as severe as scan errors, these drives did have 3 to 6 times higher AFR, a 15% failure in the following 8 month period and 14 times higher chance of failure in the first 2 months. It also appeared to be more lethal towards drives in the 10 to 60 month age group. A special subset of these errors called offline allocation errors showed a more substantial impact. There was 21 times greater chance of failure in the first 60 days, but was found in only 4% of the population.

The final SMART error of note was probational count that was detected in 2% of the population over the period of study. Drives were 16 times more likely to fail within 60 days of one of these errors.

For each of these error types, the paper included a set of survival CDF graphs of the 9-month period following the errors. The graphs include a breakdown of the survival curve by disk age groups and by the number of each type of error.

While these SMART errors are potentially useful, the authors of the paper admit these SMART errors are not enough to positively predict all the disk failures. There was a lack of predictive SMART errors in a large portion of the population. Only 56% of failed drives had any of these errors in the months prior to failure. Furthermore not all SMART errors result in a disk failure meaning that prediction could generate a large number of false positives. As a result, our approach attempts to estimate the health of disks and calculate the likelihood of per data object loss, using data movement and replication to keep it above a minimum threshold. Instead of removing or not using disks at risk, we can adaptively continue to use them until failure does occur without endangering data. By incorporating the use of SMART errors, we can achieve this with less redundancy than simple replication schemes.

III. LOGISTICAL NETWORKING

Logistical networking is concerned with the time related positioning of data resources [14], [15]. Simply put how can one arrange things so that required data will be where it needs to be when you need it. When data become large relative to network bandwidth and application timetable, the data's physicality does become a significant problem since computing performance is tethered to data availability. Researchers all over the world need to analyze data too large to be moved much and often. Therefore it becomes necessary to stage data intelligently at desired locations, according to policy and automatically reducing the burden of the end users. It is the imperative of logistical networking to be able to transfer data quickly, efficiently, reliability and easily.

The Logistical Computing and Internetwork research group (LoCI) at the University of Tennessee has been tackling issues concerning logistical networking for over 10 years. LoCI has researched the problems found in traditional storage technologies and applied the lessons from the development of the modern IP networking stack in developing solutions for data logistics. Logistical networking methodology focuses on being cleanly layered, generic and end-to-end for its services.

In addition to introducing a unique approach for how data should be handled, LoCI has created a layered framework of tools and libraries that we believe can address the issues we have raised in this paper. The layering and simplicity of the logistical networking stack makes it easy to utilize and extend. The lower layers are very generic and are designed to scale with the number of servers, users and data sets. The logistical networking tools are fast, enabling multi-server stripping and multiple data streaming transfers from multiple block level replicas. Intelligent algorithms allow end clients and middleware to transparently handle server and network loads and errors.

These layers are described in the following sections.

A. IBP

The Internet Backplane Protocol (IBP) is the fundamental core of logistical networking forming "the narrow waist" of the stack [16]. It is a very generic and lightweight block level transfer and storage service in order to give application as much freedom as possible. Middleware storage servers, called depots, are deployed as part of the network infrastructure. These depots allow end points to allocate temporary storage space. For each allocation, the allocating client is given access role keys to the allocation called capabilities that grant the client certain abilities over the allocation. These capabilities contain both the address information to identify the allocation and the rights to perform actions on the allocation. There are 3 types of capabilities for each allocation: a read capability to read the data on the allocation, a write capability to write data on the allocation and a manage capability to manipulate and retrieve the properties of the allocation.

Clients or other depots at the behest of a higher architectural layer can transfer data to and from depots. Data transfer is done per allocation/block basis and is "best effort" only. Additionally the allocations are themselves a form of "best effort" storage where they are time limited meaning that at any time data can expire can be "lost" on a depot. Forms of reliability and service quality can be achieved via higher layers through techniques like replication and data encoding.

In many ways, logistical networking's IBP is similar to network IP. Both form the common interface that architectural layers above and beneath can understand and adhere. These protocols are best effort only services for data

transfer relying on additional layers for stronger guarantees. Also IP is a datagram with fragmentation protocol similar to how IBP deals with data blocks instead of higher concepts of files and data sets. Higher layers can use IBP and depots to store and forward data across networks in unicast or multicast mode just like IP. In fact, IBP depots are very similar to IP routers in that allocations are treated as being put on a queue where data can be dropped if it has not been moved to the next destination in sufficient time.

The key difference between IP and IBP is the matter of the time scale involved. IP routers focus on the forwarding operation of data as opposed to the storing with data retention lifetime being in the milliseconds on the queue. IBP depots measure data lifetime in weeks and months by comparison with less emphasis on immediate forwarding. This can enable clients to route and transfer data more reliability and avoid frequent end-to-end retransmissions of packets found in the current IP network.

B. Exnode

Since the IBP protocol has no notion of a file at the block level or any formalism for stating a relationship between different allocations, a higher service is required to represent files and complex datasets. A file system is capable of presenting files to the user using a data structure called the inode. An inode can be used to provide a mapping between the physical blocks on disks and the logical file representation. Building on the inode concept, LoCI developed the Exnode service and API [14]. An exnode provides the information necessary to reconstruct a file from set of allocations even where the file has been split up, stripped and replicated multiple times in the wide area on remote depots.

An Exnode is a metadata container mapping byte-level allocations to the logical address space of a file. Every Exnode consists of a series of mappings with each mapping containing the IBP capabilities for that allocation, the physical offset and length within the allocation and the logical offset and length within the file.

Furthermore the Exnode has several advantages over inodes and related techniques that are important in dealing with wide area data. Unlike the inode, the Exnode directly exposes the structure of the file and allocations to the user. By exposing this information it allows the user to better optimize for his needs. For instance, the Exnode reveals the locality of the data needed by the user. This information can used so that data is retrieved from relatively nearby, high availability or high bandwidth depots.

C. LoDN

With the ability to represent files and datasets, an important problem arose with the management of the exnodes. Simply storing them on a local file system made it difficult to for user to keep track of them, share them with collaborators and create understandable relationships between exnodes. Additionally, manually renewing, replacing and controlling for a collection of exnodes is a very tedious task and can be daunting to do reliability and efficiently enough to meet the needs of uses. This lead to the development a Logistical Distribution/Data Network for users and applications to work not only with the data but the metadata associated with IBP.

LoDN, the Logistical Distribution/Data Network, is in part a policy and Exnode metadata repository that offers a hierarchical directory service. Through one of the several LoDN interfaces and APIs a user has a standard POSIX-like control of directory service. Directories and “files” can be created, renamed, moved and removed at will.

D. Implementation and Deployment

The LN software stack has been under development for over 10 years by University of Tennessee, Vanderbilt and the private company Nevoa. It is being actively used by several large organizations including the Compact Muon Solenoid group at CERN, LSST, USGS and TVNA. The largest open deployment of LN is REDDnet consisting of depots at University of California at Santa Barbara, CALTECH, Stephen F. Austin State University, Vanderbilt, University of Florida, University of Michigan and CERN in Switzerland. CMS also maintains a private collection of depots at Vanderbilt consisting of over a petabyte and a half of storage. LN has been deployed through NSF-funded infrastructure projects including the National Logistical Networking Testbed and a PlanetLab-based deployment. Additionally, Nevoa has been deploying depots and related services through Brazil for several years.

The complete LN software stack is available freely from both the University of Tennessee and Vanderbilt. LN currently features 3 different interoperable implementations of the depot server include a Java based implementation from Nevoa. Vanderbilt has also been working on an enterprise level solution called LStore that fulfills some the same functionality as LoDN but targeted for local corporate infrastructures.

IV. Methodology

A. LoDN Simulator

Logistical Networking and hard drives are very complex and modeling them through Markov Models would prove intractably difficult. Additionally, Markov Models are “memory-less” in that future events are solely dependent on the current state of the model. In the case of modeling a set of disks, any failure or repair transitions wipe out concurrent repair operations and reset the likelihood of failure. Given the cost and time involved in conducting a multiyear study to collect such information it is not possible to conduct an empirical study. Therefore, the results of this paper shall use a simulator to determine if and to what degree hard disk drive health prediction can improve data longevity, accessibility and resource utilization within Logistical Networking. Monte Carlo based simulation will provide the greatest flexibility in working with different failure behaviors and developing proactively adaptive data replication heuristics for LoDN.

When this study began there was no simulator available suitable for studying Logistical Networking for this purpose, therefore we developed our own discrete event based

Logistical Networking simulator. This simulator has the ability to simulate several different key elements and allow for enough flexibility for the behavior of hard disks and different adaptive heuristics we are developing for LoDN's data dispatcher.

The simulator is driven via an event queue with events arriving from different components of the simulation including hard drives, depots, buses, network links, LoDN and the simulation environment. The simulator will take as input a configuration file that specifies information about the depots, the local and wide area networks, hard drives and their failure behavior, data arrival patterns and the source clients.

Depots act as separate autonomous units that are interconnected via a set of topological network links. The depots are organized into physical sites much like in the real world where they are connected via a local area network and then have an external link to the outside global network. Each IBP depot contains a set of hard drives for data storage and buses that link the hard drives together and to the external network interface. The depots implement a subset of the IBP protocol including allocation, store, deletion and resource query as outlined in the IBP protocol. Depot and resource query will be extended to allow for further data acquisition than outlined in the IBP protocol. This will include information on the age, manufacturer, model, utilization and any SMART errors that may have occurred for the hard drives so LoDN will be able to analyze the expected reliability of the drive. Depots can be configured to join the Logistical Networking infrastructure at various times within a simulation to indicate real growth of the infrastructure over time. A simplifying assumption about depots in the simulation is that they do not fail or provide false data to the rest of the simulation.

Hard drives are individually configurable including their capacity, bandwidth, manufacturer, model, utilization and failure behavior. The drives maintain their current status including what data currently resides on them, the currently used and free space and the information necessary to calculate their failure time based on the failure model assigned to them at their arrival time in the simulation. The simulator makes two key assumptions regarding hard drives: that there will be no bit rot of data and the drives will operate in fail-stop mode. We believe that these are very reasonable assumptions in that Elerath [6] suggests that bit rot is really not an issue and other failure modes over shadow it. Additionally there are error-correcting codes and methods for disk scrubbing that can mitigate this phenomenon at the disk level. Since we are working from the end users' perspective of disk failure, fail-stop while not accurate does often represent the behavior of users utilization of disks. Once performance or reliability degrade below a certain point, users tend to "junk" the drives and replace them, often without realizing that data recovery is still possible.

Using the analysis work mentioned in the literature review on SMART errors in relation to hard drive failure, we are developing a series of models of disk reliability behavior for the simulator. These models will be implemented and used by the hard drives to present at what time they should stop functioning. When a drive fails all of the data on the disk will be lost and data counters for the simulator will reflect this. However, none of the components of the simulation will be aware of the failure until the drive is used or probed by LoDN. Only at that time will LoDN be able to take action to handle the failure event.

Data transfer within the simulation relies on use of data buses within the depots and a topology of network links connecting the depots. Each bus and network link is a separate simulated component that transfers data using data pulling and a round robin fair queuing between sources. At one second intervals those buses and networks with data packets are allowed to forward data to the next link within the limits of the bandwidth and queue restrictions of the next component. In the case of network links every path between two depots is a series of one or more network links and each network link can have multiple source and next hop links. The configuration file specifies the topology of the network and the characteristics of each link and a static routing table is calculated for each link based on the this information. Data travels through the network of links in data packets similar to the functionality of IP packets in the real world. The data packets include their destination address and the IBP allocation on the depot to which the data is to be written. At each link a data packet can be fragmented into smaller chunks to meet the fair share bandwidth and queue limits of the next link in the path. The data buses and links are simplified in that there is no failure requiring re-routing and retransmission of data.

Data objects are represented via a unique id within the simulation with each data object's details being recorded into a database including their current viable hard drive locations. When a data object no longer has a replica the information pertaining to the loss is also noted including information about the size, duration of time in the simulation and time of the loss. All of this information is invisible to LoDN but recorded for post analysis. The configuration file specifies the number, sizes and arrival time of data objects. This information can be static or dynamically driven with distributions like Poisson and bursty so that a wide range of possibilities could be tested. When data arrives in the system, it appears on a source client. The source client sends a request to LoDN that returns an IBP allocation for the source to send the data object to. Once the data object has arrived on depot, it is then in the hands of the Logistical Networking infrastructure to maintain it for the specified operating time frame.

The final component of the simulator is LoDN and in particular the data dispatcher. LoDN will keep track of virtual exnodes that store the locations of data objects on the depots. Each data object will have an associated replication policy file stating at what sites replicas need to exist. It is the job of the data dispatcher within LoDN to replicate and maintain allocation at the specified sites using heuristics that we will develop and study. The resource manager for LoDN periodically probes the depots and the hard drives collecting information for the data dispatcher to evaluate the estimated reliability of the disks. It is also through this method that LoDN will detect when a disk has failed. This information

will be used by LoDN to determine the potential for data loss of each data object in the system and then take whatever reactive and proactive actions are necessary to preserve the data.

B. Using the Simulator

Using this simulator, the benefits of adaptive hard drive aware data placement for data longevity and accessibility under a range of possible hard drive failure behaviors can be analyzed. The simulation will be run in three modes for the data dispatcher: the first that does traditional, reactive data replication induced by failure; the second with proactive data replication and the third with enhanced proactive data replication with the intelligent, hard drive aware heuristics. The first mode represents how LoDN currently performs replication and responses to failure. The second is another common strategy found in systems like Tempo and Glacier that replicate as much as possible in the background within the constraints of specified resource limits [17]. The final mode represents the set of algorithms under our research.

The simulations will involve a myriad of possible combinations of topologies, depot setups, hard drive characteristics, data arrival distributions and mission times. It would clearly be impossible to exhaustively study the whole range of possibilities. Therefore we plan on developing a few representative configurations. The first among these would be to a setup similar to the current REDDnet infrastructure but extrapolated out to several hundred thousand disks and many petabytes worth of data. This will require a large number of Monte Carlo simulations for meaningful analysis.

C. Failure Models

For modeling failure trends with hard disks and the usefulness of SMART errors for estimating disk health, we used the "Abstract Failure Trends in a Large Disk Drive Population" [9]. To the best of our knowledge this is the most comprehensive study of disk failure and SMART errors ever published. It comprised over 100,000 disks used by Google over 5 years and 9 months of SMART error examination and their impact on disk failure. While neither the raw or refined data is available for direct examination, we reconstructed as much data as possible from the paper as possible.

From the data represented, two different failure models were created for the simulator. The first model is based on the relationship between disk age and the type of SMART errors and the second is based the number and kind of SMART errors that occur within the given time frame. Both models used the four strongest SMART error near-term failure indicators: scan error, reallocation count, offline reallocation count and probational count. The number of disk failures per year is calculated from the AFR's listed over the course of the 5 years. According to Google, 56% of all disk failures had no SMART errors within 8 months before failure, therefore for each year this percentage of disks that were selected to fail had no preceding SMART errors. The number of disks that suffer from SMART errors was calculated from annualizing the distribution percentages over the 9 months of SMART error recording. Each year that many disks were selected to have SMART errors within the interval.

For the disk age based SMART error model, the plots points for the survival CDF were recovered and used to create a spline function. The inverse of this function was computed and used to create random failure generators for each type of error. SMART errors were then distributed randomly across the candidate disks each year and the random failure generator engines were applied using the age of the disk when the SMART error would occur. Each disk profile recorded if and when it would fail as a result of the SMART error. SMART errors were added continually until there were enough SMART error induced failures for each year. In the edge case when the SMART error would occur in the previous year but the resulting failure would occur in the next year, the failure would count towards the number of required SMART error induced failures for the next year but the SMART error event was counted in the distribution for the pervious year. If a disk had a SMART error in the previous year, then that disk could be selected for a non-SMART error induced failure in the next year after 8 months had passed from the time of the SMART errors. It seems reasonable to assume that SMART errors no longer influence the chances of a disk failure after 8 months since each failure curve levels off after that period of time from the occurrence of the error.

The error count based SMART error model was constructed similar to the previous model. SMART error random failure generators were created from the distribution curves where the input to the generator is type of SMART errors and number of errors within the time frame. The SMART errors were added in clusters to the candidate disks using a normal distribution around errors that were already on the disk in that time frame. SMART errors where continually, randomly added to achieve the specified failure counts per year. After each error was the generator would examine the disk to determine if and when the disk would failure. This process resulted in a distribution of the four error types and their counts.

Using these models, 100,000 disk profiles were generated at a time and as disks entered the simulation, they would be randomly assigned a profile to use. Once the profiles were exhausted a new batch of disk profiles would be generated.

Because of the lack of data available, certain assumptions had to be made for each model. There was no information about the impact and probability of a mixture of SMART errors so each type of error was treated mutually exclusively. Only the overall percentage of disks with each type of SMART error was provided. There were no details given about distribution of occurrences as relating to the age of disks. This required annualizing the percentages and having equal percentages per year. The number of errors and size of error clusters where not known therefore errors were added as required to meet the expected failures per year. In the case of the first model, SMART errors were added uniformly in each year. In the second model, a normal distribution was used to cluster errors for each year on a disk. No other factors including utilization, temperature, model and vintage

were included. While model and vintage have been shown to be significant, this can be mitigated by assuming a mixed population of disks. Additionally utilization and temperature have been shown to be far less significant by the source paper.

D. Measuring and Comparing Results

There are several proposed metrics for estimating the time it would take a storage system to have an unrecoverable data loss event. For our research we feel that the best metric to use the Normalize Magnitude of Data Loss, $NOMDL_t$ instead of the more traditional MTTDL, Mean Time To Data Loss, approach. While MTTDL has been the standard reliability metric in most academic work, it suffers from a number of flaws. It is based on Markov Model where each state is the number of working hard drives and transitioning from each state is governed by constant failure and repair rates. It assumes that failure is an exponential distribution, which does not fit with the models we have constructed. MTTDL lacks the ability to cover a given duration of system operation. Instead it states the MTTDL from the start of operation to infinity. It is very unrealistic to assume that a storage system would continue forever. Furthermore, MTTDL only specifies a mean time of data loss but can not provide information as to the degree of that data loss. While two storage systems might have the same MTTDL, the amount of data actually lost could be drastically different.

$NOMDL_t$ address all of these issues and provides a better means of comparison by stating the expected amount of data loss for a given period of expected operation. This is a much more informative metric for understanding data loss as it makes comparison between different scenarios far easier and understandable. $NOMDL_t$ additionally has the advantage of being calculated via a series of Monte Carlo simulations. The fact that $NOMDL_t$ already presupposes simulation data makes it a very natural fit for our research.

V. Results

We have found some preliminary results with complete results being available in the final paper. For the three modes discussed (reactive, proactive and disk health aware proactive replication), disk health aware does perform as well as or better than the other two. When there is enough storage space available, 3 or more copies provide enough resilience for all three modes. However, when storage space relative to the amount data is small, disk health aware does perform better than the previous two methods. In a non-trivial percentage of the disk population when there was only enough room for 1 to 2 copies of data, the enhanced method was able to improve data resiliency.

Based on Google's study, we examined two different models of failure involving SMART errors. The first model based on the age of the disks when SMART errors occurred and the second model based on the number of each kind of SMART error that occur in a given time frame. Factoring in age and the number of errors, significantly improved disk usage and data resiliency.

One drawback though to the results is that it was impossible to have a combined model of effects of disk age and SMART error counts since this information was not made available. In the future, it might be worth examining several potential realistic models that combine these factors.

Proactive replication performs better than reactive replication since proactive replication will utilize nearly all available space for replication while reactive simply attempts to maintain a static number of copies. Of course proactive storage degenerates to reactive storage when there is only enough space for a number of copies matching the static replication requirement of reactive storage.

Disk health aware proactive replication saves significant bandwidth over the wide area. Proactive replication wastes significant bandwidth making as many copies as possible. Because reactive replication waits on a failure, a full copy must be pulled across the wide area to replace the missing data. Proactive disk health aware replication reduces the amount of wide area bandwidth utilized. When a disk is suspected to fail in the near term, data replicates the data on the disk to another disk or disks to maintain the data's health factor. This replication can occur to other depots at the same site or even neighbor disks on the same bus. While disk failure prediction has been shown to be imperfect, it still allows for this optimization in a number of cases. The most dangerous time in the existence of data is after the initial upload from the client. At this point, there is only one copy of the data and if the disk on which the data resides fails, then the data is lost. Further compounding the problem is that it may be some time before the client is made aware of the failure. The client in the interim may have already removed the data locally assuming it to be safe. All three schemes suffer from this problem; however, using disk aware health statistics allow for the initial upload to go what is believed to be a healthier disk. This lessens the odds of suffering a disk failure before additional copies can be replicated.

VI. Conclusions

In this paper, we have described a system that seeks to use disk failure avoidance to reduce the need for costly redundancy by using adaptive heuristics that predict such failures. While a number of predictive factors such as hard drive utilization rate, age, SMART errors, and model can be used, our initial work has focused on SMART errors. We have described an approach that can predict where near term disk failures are more likely to occur, enabling proactive movement/replication of at-risk data, thus maintaining data integrity and availability. Our initial results demonstrate that this strategy can reduce costs due to redundant storage without compromising these important requirements.

References

[1] J. Gantz, D. Reinsel, C. Chute, and V. Schlichting, *The Expanding Digital Universe-A Forecaset of Worldwide Information Growth Through 2010, March 2007*.

[2] J. Gantz, "The Diverse and Exploding Digital Universe."

[3] J. Gantz, *The Digital Universe Decade, Are You Ready?* IDC, 2010.

[4] J. G. Elerath and Sandeep Shah, “Annual Symposium Reliability and Maintainability, 2004 - RAMS,” presented at the Annual Symposium Reliability and Maintainability, 2004 - RAMS, 2004, pp. 151–156.
[5] J. G. Elerath and S. Shah, “Disk drive reliability case study: dependence upon head fly-height and quantity of heads,” *Reliability and Maintainability Symposium, 2003. Annual*, pp. 608–612, 2003.
[6] J. Elerath, “Hard-disk drives,” *Commun. ACM*, vol. 52, no. 6, pp. 38–45, Jun. 2009.
[7] J. G. Elerath, “Elerath, “Reliability Analysis of Disk Drive Failure Mechanisms,” 2005.
[8] W. Jiang, C. Hu, Y. Zhou, and A. Kanevsky, “Are disks the dominant contributor for storage failures?,” *Trans. Storage*, vol. 4, no. 3, pp. 1–25, Nov. 2008.
[9] E. Pinheiro, W. D. Weber, and L. A. Barroso, “Failure trends in a large disk drive population,” *Proceedings of the 5th USENIX conference on File and Storage Technologies*, pp. 2–2, 2007.
[10] Sandeep Shah and J. G. Elerath, “Annual Symposium Reliability and Maintainability, 2004 - RAMS,” presented at the Annual Symposium Reliability and Maintainability, 2004 - RAMS, 2004, pp. 163–167.
[11] B. Schroeder and G. A. Gibson, “The computer failure data repository (CFDR),” *Workshop on Reliability Analysis of System Failure Data (RAF'07), MSR Cambridge, UK*, 2007.
[12] B. Schroeder and G. A. Gibson, “Disk failures in the real world: what does an MTTF of 1,000,000 hours mean to you?,” presented at the FAST '07: Proceedings of the 5th USENIX conference on File and Storage Technologies, 2007.
[13] J. Yang and F.-B. Sun, “A comprehensive review of hard-disk drive reliability,” presented at the Reliability and Maintainability Symposium, 1999. Proceedings. Annual, 1999, pp. 403–409.
[14] J. S. Plank, M. Beck, and T. Moore, “Logistical Networking Research and the Network Storage Stack,” *Proc. Usenix Conf. File and Storage Technologies, work in progress report*, 2002.
[15] M. Beck, T. Moore, J. S. Plank, and M. Swany, “Active Middleware Services (Salim Hariri, Craig A. Lee, and Cauligi S. Raghavendra editors), chapter Logistical Networking: Sharing More Than the Wires,” 2000.
[16] Bassi, “The Internet Backplane Protocol: A Study in Resource Sharing,” *Cluster Computing and the Grid, 2002. 2nd IEEE/ACM International Symposium on*, p. 194, 2002.
[17] Z. Yang, J. Tian, B. Y. Zhao, W. Chen, and Y. Dai, “Protector: A Probabilistic Failure Detector for Cost-Effective Peer-to-Peer Storage,” *IEEE Trans. Parallel Distrib. Syst.*, Nov. 2010.