Incorporating Proactive Data Rescue into ZFS Disk Recovery for Enhanced Storage Reliability

Zhi Qiao*, Song Fu†, Hsing-bung Chen‡ and Michael Lang§
*†Department of Computer Science and Engineering, University of North Texas
‡HPC-DES Group, Los Alamos National Laboratory
§UltraScaleSystems Research Center, Los Alamos National Laboratory
*zhiqiao@my.unt.edu †song.fu@unt.edu ‡hbchen@lanl.gov §mlang@lanl.gov

I. Introduction

Computations and simulations help advance knowledge in science, energy, and national security. Over the years, they have become more accurate to generate more realistic outcome, and as a result, the demand for computational power and much larger storage system also increases. A typical HPC simulation on LANL's Trinity requires hundreds of PBs of data to be written out on hundreds of thousands of disk drives, in order to capture the entirety of simulation data. At such a scale, disk failures become the norm. Data recovery takes longer time due to increased disk capacity [1]. For a helium-filled 8 TB hard drive, an extensive disk rebuild could last for five days. In a large RAID array, a disk drive may fail while another disk is being rebuilt, which results in data loss and significant performance degradation.

ZFS is a widely used filesystem for HPC systems and local server storage. For example, Lustre uses ZFS as its backend. ZFS is designed from scratch to mitigate fault management problems in storage systems [2] [3]. However, there are many factors that affect the performance of ZFS for failure recovery in a production environment, such as the storage pool utilization, the number of virtual disks in a mirrored group, configuration of software RAID, and the amount of corrupted data. Moreover, host system's hardware configuration also influences ZFS recovery performance [4] [5]. Factors, such as the use of flash storage [6] and the amount of RAM and CPU resources available for ZFS, are crucial.

In this paper, we purpose an analytic model to characterize and mitigate the performance impact from disk failures in storage arrays. We extensively evaluate the recovery performance with a variety of ZFS configurations, including different RAM size and varying RAID array configurations. Due to the wide adoption of flash storage, we conduct our experiments in a heterogeneous storage environment that consists of both hard disk drives and solid state drives. We can obtain the following important observations from our experimental results. *1) CPU performance dominates ZFS recovery time*, which cancels out the performance benefit from using flash storage. *2) Increasing storage utilization causes degraded ZFS recovery performance*. For a storage array that is almost full, the recovery speed is 1.49 times slower than the average speed. *3) Increasing the amount of RAM available to ZFS does not alway lead to better ZFS recovery performance.* Although ZFS is known for being RAM hungry, adding more RAM to the system only improves I/O, but yields no obvious improvement of the recovery time. These observations motivate us to develop alternatives to handle disk failures, such as proactive data rescue with disk failure prediction.

II. Methodology

We extensively evaluate the storage performance with a variety of settings. This helps us characterize system performance in face of disk failures. Then, we analyze ZFS' fault management policies with different storage utilization, and study the optimal solution.

A. Characterizing ZFS Recovery Performance

ZFS has been widely used in production storage systems. It is the backend of Lustre, which has been used in over 60% of Top 100 supercomputers. ZFS is a complex system. Many factors and parameters can affect its runtime performance. The table below shows the configuration parameters and their values that we use in our experiments.

RAM	Disk Media	Utilization%	RAID Type
8-32GB	SATA HDD SATA SSD	15%-75%	RAID5-RAID7 (RAIDZ1-3)

B. Proactive Data Rescue

In addition to disk rebuild after disk drives fail, data can be rescued proactively prior to disk failures. This is enabled by disk failure prediction techniques [7] [8] which can forecast when failures will happen on which drives with a promising accuracy. By incorporating disk failure prediction, ZFS becomes capable of replacing a failing drive before a failure actually occurs. Data on the failing drive are migrated to a spare drive without the expensive disk rebuild and storage disruption. However, some predictive models only work well on drives from certain manufacturers and some may generate excessive false alarms that cost considerable overhead to the system. How to balance the benefit and cost becomes critical to determine the effectiveness and performance of proactive data rescue strategy. To address this issue, we design a finite search space to model runtime recovery performance of ZFS, and explore storage utilization and workload statistics to derive the optimal strategy. As we detail in Section III.B, we invoke

This publication has been assigned an LANL identifier LAUR-xx-17.

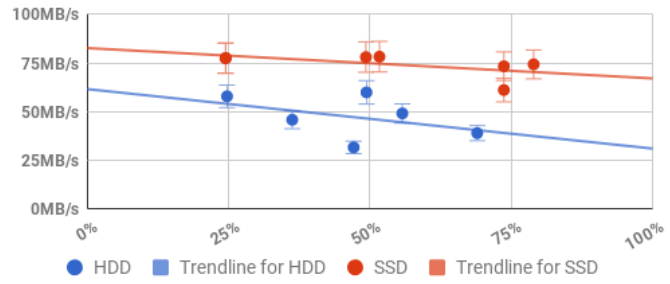

Fig. 1. ZFS recovery throughput under different zpool utilization. Linear regression is used to fit the curves.

ZFS's default recovery process when the storage utilization is low. As more data are stored, we gradually switch to proactively cloning data from the failing disk.

III. Experimental Results

We conduct experiments on HPC servers having multi-core Intel Xeon processors, Ubuntu 16.04 LTS, and ZFS version 0.6.5.11. The servers are also equipped with Seagate BarraCuda ST2000DM hard disk drives (HDD) and Intel DC3520 solid state drives. Some servers have four Seagate HDDs for data storage, one Intel SSD as the boot drive, and 32GB RAM on board. Others use four Intel SSDs for data storage.

ZFS uses storage pools which are called zpool to manage data. We setup zpool on the data storage drives with different RAIDZ configurations. First, we write data to a zpool to reach certain utilization level. Then, a fault is injected to one of the drives in the zpool by flushing the zpool metadata. When ZFS scans the health status of the zpool, it detects the data corruption. ZFS marks that disk as failed and then performs disk recovery.

A. Result Analysis

Our experimental results show that *1) Recovery is a time-consuming process and CPU performance dominates ZFS recovery time*. For example, 900GB data on HDD takes 8 hours to recover. As disk rebuild involves heavy computation, the recovery speed is 5 times slower than I/O throughput. *2) The recovery throughput degrades as the zpool utilization increases.* Figure 1 shows the recovery throughput under different zpool utilization for both HDDs and SSDs. When the storage array is almost full, the recovery speed becomes 1.49 times slower than the average throughput, and 1.89 times slower than the peak throughput. *3) Adding more RAM does not always improve ZFS recovery performance.* The results vary from -5.5% to 7.8%.

B. Data Rescue Strategies

Disk failure prediction technology continues evolving and is capable of accurately forecasting when and where disk failures will occur in storage systems. These prediction results enable ZFS to replace failing drives using spare ones in advance.

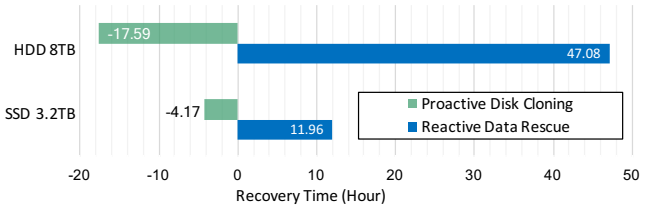

Fig. 2. Time used for proactive disk cloning and post-failure disk recovery for a fully-utilized zpool. *0H* denote the disk failure.

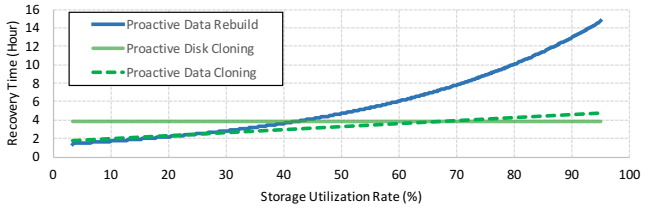

Fig. 3. Time used for proactive disk cloning and proactive data recovery under different zpool utilization.

Several strategies can be used to rescue data. If a failure prediction provides enough lead time, the *disk cloning* strategy copies data from a failing drive to a spare one. However, without knowing which disk sectors have useful data, disk cloning has to copy everything from every sector to the spare drive. In contrast, ZFS tracks active data in a zpool and recovers only the active data when a disk is detected as failed. Although *disk recovery* is compute intensive, recovering only the active data can save time.

Figure 2 shows the performance of proactive disk cloning vs. post-failure data recovery. From the figure, we can see that when the zpool utilization is high, post-failure data recovery takes 47 hours to recover a 8 TB disk drive, while cloning disk uses only about 1/3 of the time.

Figure 3 shows duration of disk cloning and ZFS data recovery under different zpool utilization. Linear regression provides the best fit of the curves. From the figure, we can see that proactive data recovery is more efficient when the zpool is lightly used (i.e., 37%). As more active data are stored, proactive disk cloning becomes a better choice.

Our analytic model uses the collected zpool utilization data and system configuration parameters (described in Section II.A) to derive the optimal data rescue strategy that best suits the storage array in the current state.

IV. Conclusions

Exascale computing imposes a significant challenge on storage reliability. In this paper, we characterize system performance under disk failure with a variety of ZFS configurations. We then develop an analytic model that derives the optimal data rescue strategy using disk cloning and data recovery. We purpose a proactive disk failure management approach to enhance storage reliability.

REFERENCES

[1] Garth A. Gibson and David A. Patterson. Designing disk arrays for high data reliability. *Journal of Parallel and Distributed Computing*, 17(1-2):4–27, 1993.

[2] Jeff Bonwick, Matt Ahrens, Val Henson, Mark Maybee, and Mark Shellenbaum. The zettabyte file system. In *Proc. of the 2nd Usenix Conference on File and Storage Technologies*, volume 215, 2003.

[3] Yupu Zhang, Abhishek Rajimwale, Andrea C Arpaci-Dusseau, and Remzi H Arpaci-Dusseau. End-to-end data integrity for file systems: A zfs case study. In *FAST*, pages 29–42, 2010.

[4] Veerapat Phromchana, Natawut Nupairoj, and Krerk Piromsopa. Performance evaluation of zfs and lvm (with ext4) for scalable storage system. In *Computer Science and Software Engineering (JCSSE), 2011 Eighth International Joint Conference on*, pages 250–253. IEEE, 2011.

[5] Dominique A Heger. Workload dependent performance evaluation of the btrfs and zfs filesystems. In *Int. CMG Conference*, 2009.

[6] Adam Leventhal. Flash storage memory. *Communications of the ACM*, 51(7):47–51, 2008.

[7] Song Huang, Song Fu, Quan Zhang, and Weisong Shi. Characterizing disk failures with quantified disk degradation signatures: An early experience. In *IEEE International Symposium on Workload Characterization (IISWC)*, pages 150–159. IEEE, 2015.

[8] Mirela Madalina Botezatu, Ioana Giurgiu, Jasmina Bogojeska, and Dorothea Wiesmann. Predicting disk replacement towards reliable data centers. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 39–48. ACM, 2016.