Memory and Replica Coordinative Lifetime Enhancement of Flash Storage in Distributed Systems

Narendra Kumar Govinda Raju [Student] **Jishen Zhao** [Professor] **Peter Alvaro** [Professor]
University of California, Santa Cruz, California

Abstract

NAND Flash memory are replacing the traditional storage systems like the magnetic and optical mediums due to their better performance during read and write. However, the major drawback with the flash systems is that they have a very low endurance. The lifetime of the flash memory is less compared to the magnetic medium. It is limited due to the fact that they wear out after few thousands of program-erase (PE) cycles. The magnetic or optical drives support 10^{12} - 10^{14} write cycles whereas flash system support just 10^{6} – 10^{7} write cycles. Hence the flash systems are not suitable for write intensive applications. In this paper, we propose a new solution to reduce the number of writes in flash systems which increases the endurance, thereby increasing their lifetime. On the event of a write, the immediate writes on few of the replicas is delayed and batched at a later stage as a lazy update or through logging. This solution can be deployed on a weak consistent system running write intensive applications. Also in distributed systems, all applications do not have to run as strongly consistent always. Depending on the needs whenever the application does not need strong consistency, our solution could be enabled to increase the system's endurance. This makes the flash systems usable even for work intensive applications as their endurance is increased.

I. Introduction

Flash systems are gaining their presence everywhere from mobile devices to servers with RAID and SAN architectures. SSD has become more popular due to their high speed, low noise, low power consumption and reliability.

There are many reasons for flash storage picking up in the storage industry. The main advantages of NAND flash are: 1) Better performance capabilities, speed in particular. Flash storage can get the system up and running in few seconds. Enterprises that need fast processing of their business applications and retrieve/store data quickly prefer flash storage systems. 2) The durability is very high compared hard drives which have mechanical parts like the spinning disks, head etc. The chances of losing data due to equipment mishandling is low. This feature is very important for the business who are more concerned about the security of their data. The absence of moving parts also contributes to higher performance. 3) They consume very less power, thereby reduces the energy costs greatly.

There are a few disadvantages of the flash system as well and the main disadvantage which is of interest to us is the endurance. Endurance is just a fancy name for life time of the storage systems. Endurance of a storage system is a very important aspect as it directly correlates to the lifetime of data. Compared to the hard drives, the NAND flash has a very low endurance. They have a finite program-erase cycles because of the process involved in the program erase operations for every write. NAND flash uses two methodologies to write data: 1) Quantum Tunneling 2) Hot Electron Injection. Each write to these systems causes a slight physical damage due to the above mechanisms. The damages are

due to the high heat generated. The oxide layer in the flash systems which are used to trap the charges is degraded every time a write is performed. The charge stored representing either a 0 or 1 cannot be differentiated due to the damage and hence the flash systems become unusable after this point. The damage eventually piles up to decrease the endurance time of the flash system. The other disadvantage of the flash systems compared to the hard drives is the cost per GB. NAND flash is much more expensive than the hard drives. The price of flash storage is gradually decreasing but currently SSD are more expensive.

There are many distributed system applications which require a balance out between strong and eventual consistency. For example, consider an online shopping website. Use case such as the billing process should be strongly consistent whereas a recommendation window based on the users' browsing history can be weakly consistent. For such systems, the proposed solution can be deployed for use cases which do not need strong consistency.

II. Motivation

When we consider the flash storage, we know that the number of writes which can be performed on flash storage is in the range of 10^6-10^7. Due to the property of the flash memory, we can't perform in-place update on a block of data. So we have to go through an erase of the block which completely erases the page by making all bits to zero (by having negative charge) and then perform the write cycle. But this leads to a condition called uneven wearing because a part of the storage is accessed rarely where the data is used only for reading (say a copy of the movie image) and the other part of the data is accessed often for writing (say a part of the disk which is used for paging). To solve this problem "Wear Leveling" was introduced. But wear leveling increases the number of writes in a flash storage i.e. the write amplification. The main idea of this research is to reduce the number of writes if the same piece of data is present in the RAM or in the storage replica which is about to change shortly.

III. Design and Implementation

The main objective of our project is to reduce the total number of writes. We have two approaches to achieve this. One is by not replicating data to one of the replica every time there is a write request from the client. The write is delayed for a fixed time and then initiated called the Lazy update. The second approach is not replicating the data to any of the replicas instead log all the writes and then update the replicas after certain time. This avoids considerable program-erase cycles if same blocks are rewritten.

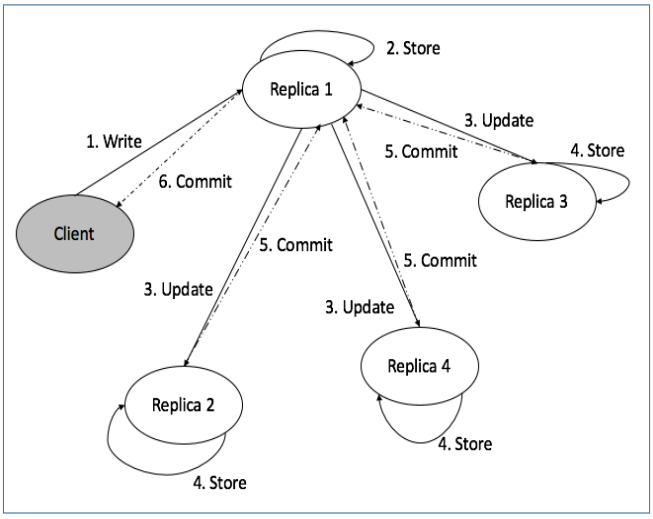

Figure 1 Simple key, value pair Distributed System without our solution

With the first approach the replication is delayed to one of the replica nodes whereas with the second approach the delay is to all the replicas. Thus these approaches are more suitable for weak consistency applications. We wanted to design this approach on CEPH system. But due to time constraints and huge code base of CEPH we have gone ahead to design the model on a simple key, value based distributed system. Without our

solution, the system would update the latest data immediately to all the replicas without any delay as shown in the figure 1. Whenever there is a write request from the client, the data is replicated to all the replicas 1-4 immediately and simultaneously. This leads to more program-erase cycle thus lowering the endurance of the flash system.

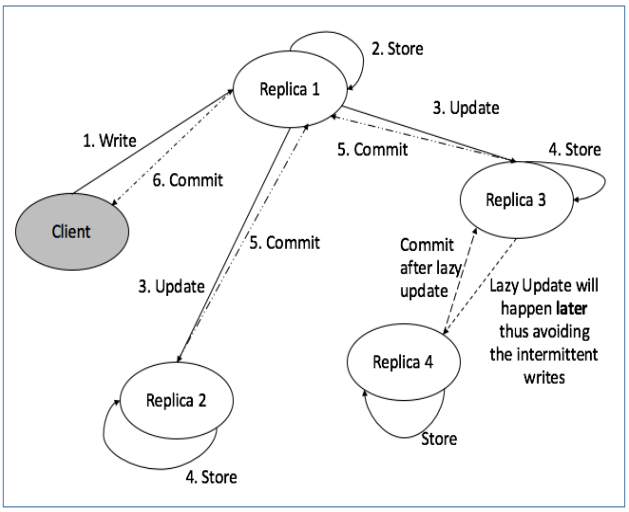

Figure 2 Key, Value pair distributed system with lazy update on Replica 4

The data flow diagram with the first approach of our solution is as shown in the figure 2 above. A single replica is chosen from all the replicas for the lazy update. We have chosen Replica 4 for illustration purpose. On the event of a write request from the client, the data is replicated to all replicas immediately except for replica 4. Once all the other replicas store the new data and committed, the commit signal is sent back to the client. However, the update to replica 4 is delayed up to 2 minutes or until acceptable consistency delay. This way the number of writes to replica 4 is reduced. This increases the endurance of replica 4. With this approach, we will be able to improve the endurance of one node. With the second approach, whenever a client sends a write request, it is logged in and commit signal is sent to the client. A finite number of writes is accumulated until it is actually written to all the replicas as shown in figure 3 below. After certain time, the accumulated writes are sent, all at once to all the replicas. This reduces the program-erase cycle of the flash system. With this approach, the intention is to improve the endurance of all the replicas.

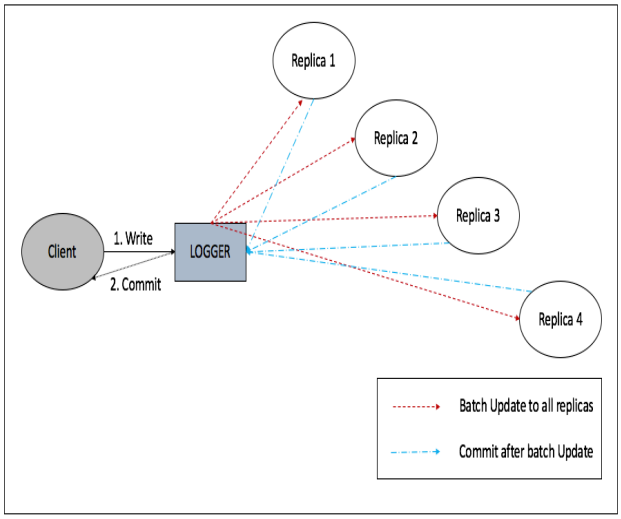

Figure 3 Key, Value distributed system with logging for write events

IV. Result

Simulation of the simple key, value pair distributed system has been done using Python language.

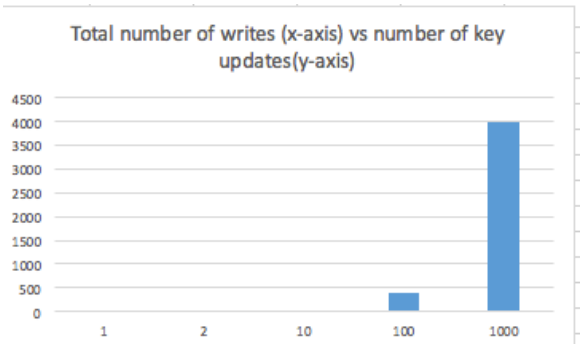

Figure 4 Graph for simple key, value pair distributed system without our solution

The above graph will be compared with the results obtained once the issues are fixed.

Basic skeleton code for the lazy update and logging approach has been written, but facing issues. Trying to fix these issues by second deadline and update this section. I shall submit the updated document by next week.

Code base: https://github.com/Narendrakumarg1728/write_reduction_in_Distributed_replicas

V. Related Work

There is a lot of related research work to improve the endurance of NAND flash systems. Most of research work is related to "Wear Leveling" and "RAID". But as mentioned earlier, wear leveling results in write amplification i.e. it increases the number of writes [9] [16]. In RAID based systems, data is written on one of the mirrored copy whereas the update is avoided on the other mirrors [3]. The data is replicated at a later stage. As per the knowledge of the authors, there is no work related to reducing the number of writes on a flash storage in distributed systems. One of the project done in UCSC under Professor Carlos, aims to improve the read performance in a write intensive workload distributed system. The paper is not related to improving the endurance of the flash system.

VI. Conclusion and Future Work

Will be updated after the results obtained.

VII. References

[1] Reliably Erasing Data From Flash-Based Solid State Drives Michael Wei ∗ , Laura M. Grupp ∗ , Frederick E. Spada † , Steven Swanson University of California, San Diego

[2] STUDY OF BAD BLOCK MANAGEMENT AND WEAR LEVELING IN NAND FLASH MEMORIES Supriya Kulkarni P1 , Jisha P2 1 Student, 2Assistant Professor, Electronics & Communication Dept, MVJ College of Engineering, Bangalore, India supriyakul059@gmail.com, jishahaneesh@gmail.com: (Very basic paper, learnt only about bad block management)

[3] HRAID6ML: A Hybrid RAID6 Storage Architecture with Mirrored Logging Lingfang Zeng †, Dan Feng † , Janxi Chen † Qingsong Wei §, Bharadwaj Veeravalli #, Wenguo Liu † --

[4] CSWL: Cross-SSD Wear-Leveling Method in SSD-Based RAID Systems for System Endurance and Performance Kwanghee Park, Dong-Hwan Lee, Youngjoo Woo, Geunhyung Lee, Ju-Hong Lee†, Deok-Hwan Kim* † Dept. of Electronic Engineering, Inha University, Reliability and Performance Enhancement Technique for SSD array storage system using RAID mechanism Kwanghee Park, Dong-Hwan Lee, Youngjoo Woo, Geunhyung Lee, Ju-Hong Lee†, Deok-Hwan Kim* † Dept. of Electronic Engineering, Inha University,

[5] An Embedded FTL for SSD RAID Alistair A. McEwan and Irfan Mir Department of Engineering University of Leicester, Leicester LE1 7RH, UK

[6] Reliability Management Techniques in SSD Storage Systems Thesis submitted for the degree of Doctor of Philosophy at the University of Leicester by Irfan F. Mir

[7] Building Flexible, Fault-Tolerant Flash-based Storage Systems Kevin M. Greenan † Darrell D.E. Long † Ethan L. Miller † Thomas J. E. Schwarz, S.J. ‡ Avani Wildani † Univ. of California, Santa Cruz † Santa Clara University

[8] Wear levelling using machine learning, this was interesting concept by Coughlin Associates. (Flash Memory Summit)

[9] Wear leveling and few file system concepts at blogs of Elastifile. (Got to know about at Flash Memory Summit)

[10] Software Defined Storage concepts as per Nutanix. (Got to know about at Flash Memory Summit)

- [11] Samsung Magician Brand User_Guide (Samsung SSD disk management and diagnostic features for server and data center usage)
- [12] How Controllers Maximize SSD Life (SNIA publication)
- [13]S olidFire Data Efficiencies Brief article.
- [14] Intel® Rapid Storage Technology article.
- [15] INCITS Technical Committee T10 (SCSI) (TRIM command RFEs (Request for enhancements)

[16] Murugan.M, Du. D, Department of Computer Science, University of MinnesotaRejuvenator:A StaticWear Leveling Algorithm for Flash memory

[17] http://www.zdnet.com/article/enterprise-storage-trends-and-predictions/