isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 4, Issue: 3,Special Issue: 2 ,Apr,2016 ,ISSN_NO: 2321-3337

NETWORK CODING FOR ACCESS IN ERASURECODED STORAGE SYSTEM

Vidhyapriya.M[1], Priyanka.N[2] , Mr.A.Thiyagarajan[3]

Student, Dept. Of Computer Science And Engineering, Agni College Of Technology,India.[1,2]

Asst. Professor, Dept. of Computer Science and Engineering, Agni College of Technology, India.[3]

ABSTRACT- *No single encoding scheme or fault model is optimal for all data. A versatile storage system allows them to be matched to access patterns, reliability requirements, and cost goals on a per-data item basis. Ursa Minor is a cluster-based storage system that allows data-specific selection of, and on-line changes to, encoding schemes and fault models. When using the specialized distributions, aggregate cluster throughput nearly doubled. Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system While the latency metric has been the focus of many research studies, the bandwidth metric has received comparatively little attention. Peer-to-peer systems are positioned to take advantage of gains in network bandwidth, storage capacity, and computational resources to provide longterm durable storage infrastructures.*

Keywords:- Erasure coded storage system, degraded reads, IO parallelism, HDFS

1. INTRODUCTION

To ensure data availability, storage systems usually stripe data redundancy across multiple storage nodes (or servers). Replication is traditionally used to provide data redundancy yet it introduces high storage overhead and becomes a scalability bottleneck. On the other hand, erasure coding provides space-optimal data redundancy while achieving the same fault tolerance as replication. It operates by encoding data into multiple fragments, such that any subset of fragments can sufficiently reconstruct the original data. A node is permanently failed if it loses all its stored data. To preserve the required redundancy level and maintain data availability, a storage system performs failure recovery, which reconstructs all the lost data in another new node. To access the unavailable data, a storage system performs a degraded read, which retrieves data from surviving nodes and reconstructs the unavailable

data. It is expected that degraded reads are slower than normal reads. Field measurements show that temporary failures contribute to the majority of component failures in large-scale storage systems. Thus, degraded reads are more frequently performed than failure recovery operations, and their performance optimizations are critical. It is expected that degraded reads are slower than normal reads. Field measurements show that temporary failures contribute to the majority of component failures in large scale storage system. Thus, degraded read a more frequently performed than failure recovery operations, and their performance optimizations are critical. Due to system upgrades and scaling, distributed storage systems are typically composed of nodes with heterogeneous storage capacities and I/O speeds. Also, files are usually distributed over multiple storage nodes and accessed in parallel. Thus, the degraded read performance is inevitably bottlenecked by the poorly performed surviving nodes.

2. DESIGN REQUIREMENTS

Range accesses: The key objective for permanent failure recovery is to read data from surviving nodes to reconstruct the lost data on a new replacement node. However, degraded reads are different in that a degraded read request typically accesses a range of data units that span not only the unavailable data units on the temporarily failed nodes, but also the available data units on the surviving nodes. To realize a degraded read, it is necessary to read the available data units and extra data units to reconstruct the unavailable data units from the surviving nodes.

Node heterogeneity: Storage systems are often upgraded over time, and thus storage nodes may be composed of heterogeneous hardware resources, such as disks and network interfaces with different bandwidths. On the other hand, the available resources of each storage node may vary from time to time due to dynamic load conditions. Since degraded reads usually use the available data from multiple surviving nodes to reconstruct the unavailable data, the resulting degraded read performance may be bottlenecked by the poorly performed surviving nodes.

Parallel accesses: To further boost the degraded read performance, the degraded read solution should leverage I/O parallelism, which has become an essential property of modern storage systems. Contiguous blocks are striped across different nodes so that a sequential read request can be parallelized. Previous studies also exploit parallel I/Os in failure recovery. A challenge is to extend the parallelism in degraded reads subject to range accesses and node heterogeneity.

Volume: 4, Issue: 3,Special Issue: 2 ,Apr,2016 ,ISSN_NO: 2321-3337

Online decision: The degraded read performance is determined by the read size and node resources, both of which are variable factors depending on the current load conditions. Thus, the degraded read solution must be determined online for each read request based on its read size and the available node resources. In particular, it is infeasible to enumerate on-the-fly the degraded read solutions for all possible failures as in but instead we need to find an efficient degraded read solution in a timely manner. Applicability to general erasure codes: Authors of propose new erasure codes which achieve efficient degraded read performance. However, a storage system has typically deployed a specific erasure code for its stored data, and re-encoding a huge amount of stored data with a new code will introduce significant overheads. Also, different types of data are to be differently encoded based on the fault tolerance and access requirements. Thus, instead of constructing new codes, we aim to develop efficient degraded read strategies that are applicable to a general class of existing erasure codes.

NON FUNCTIONAL REQUIREMENTS

Reliability: To reduce the window of vulnerability and thus improve the system reliability, the reconstruction time must be significantly reduced. Since user I/O intensity severely affects the reconstruction process, WorkOut aims to reduce the I/O intensity on the degraded data set by redirecting I/O requests away from the degraded data set.

Availability: To avoid a drop in user perceived performance and violation of SLAs, the user response time during reconstruction must be significantly reduced. WorkOut strives to achieve this goal by significantly reducing, if not eliminating, the contention between external user I/O requests and internal reconstruction requests, by outsourcing I/O workloads to a surrogate data set.

Extendibility: Since I/O intensity affects the performance of not only the reconstruction process but also other background support tasks, such as re-synchronization and disk scrubbing, the idea of WorkOut should be readily extendable to improve the performance.

Flexibility: Due to the high cost and inconvenience involved in modifying the organization of an existing data, it is desirable to completely avoid such modification and instead utilize a separate surrogate data set judiciously and flexibly. In the WorkOut design, the surrogate data set can be a dedicated data set, a dedicated data set or a live data set that uses the free space of another operational (live) data set. Using the surrogate set ensures that the redirected write data is safe-guarded with redundancy, thus guaranteeing the consistency of the redirected

data. How to choose an appropriate surrogate data set is based on the requirements on overhead, performance, reliability, and maintainability and trade-offs between them.

3. SYSTEM ANALYSIS

Existing System

Regenerating codes introduce more I/Os as they read all stored data and transfer the encoded data. Build a system on HDFS that augments existing optimal regenerating codes to support a general number of failures, and demonstrate the saving of data transfer of regenerating codes over erasure codes in recovery. Existing studies focus on the latter case, and do not account for the read sequence Several studies minimize recovery I/Os (i.e., the amount of data read from surviving disks) for specific erasure codes. Optimal recovery schemes have been proposed for different RAID-6 codes, and achieve around 25 percent of I/O savings compared to simply reconstructing all original data. FARM improves recovery in large-scale storage systems using parity declustering. Some studies leverage workload characteristics to improve the reconstruction through extensive simulations and testbed experiments, we justify the effectiveness of FastDR in achieving efficient degraded reads in a heterogeneous storage environment. A system for boosting degraded reads in XOR-based erasure coded storage systems with heterogeneous storage nodes.To ensure data availability, storage systems usually stripe data redundancy across multiple storage nodes (or servers). Replication is traditionally used to provide data redundancy, yet it introduces high storage overhead and becomes a scalability bottleneck. On the other hand, erasure coding provides space-optimal data redundancy while achieving the same fault tolerance as replication . It operates by encoding data into multiple fragments, such that any subset of fragments can sufficiently reconstruct to access the unavailable data, a storage system performs a degraded read, which retrieves data from surviving nodes and reconstructs the unavailable data. It is expected that degraded reads are slower than normal reads. Field measurements show that temporary failures contribute to the majority of component failures in large-scale storage systems.

SOLVING THE EXISTING SYSTEM PROBLEM

An optimization problem of boosting degraded reads, in which we associate each storage node with a cost of reading per unit of data and propose a model that minimizes the parallel degraded read time. We first formulate the problem of degraded reads for XOR-based erasure codes, and then propose an optimization model for degraded reads that accounts for node

Volume: 4, Issue: 3,Special Issue: 2 ,Apr,2016 ,ISSN_NO: 2321-3337

heterogeneity and I/O parallelism. We further present our EG algorithm that can return an efficient degraded read solution within a reasonable search time. A greedy recovery heuristic to minimize I/Os for any erasure code. Our approach is built on the greedy heuristic in and extends the latter by taking into account node heterogeneity and I/O parallelism in our optimization model. Regenerating codes that minimize the amount of data transfer in distributed storage systems. Note that regenerating codes introduce more I/Os as they read all stored data and transfer the encoded data a new non-MDS local reconstruction code that minimizes I/Os in degraded reads by adding more parity blocks. Recently, other local reconstruction codes are proposed for achieving efficient recovery .These codes are also evaluated on HDFS.A system for boosting degraded reads in XOR-based erasure coded storage systems with heterogeneous storage nodes. We formulate an optimization model the rotated Reed-Solomon (RS) code that reduces I/Os in degraded reads over traditional RS codes. The rotated RS code is designed for sequential read and takes into account the read size .

SYSTEM ARCHITECTURE:

The purpose of the architecture diagram is to represent the type of software architecture that is used by the system, to describe the various hardware and software components that are used for the system implementation.

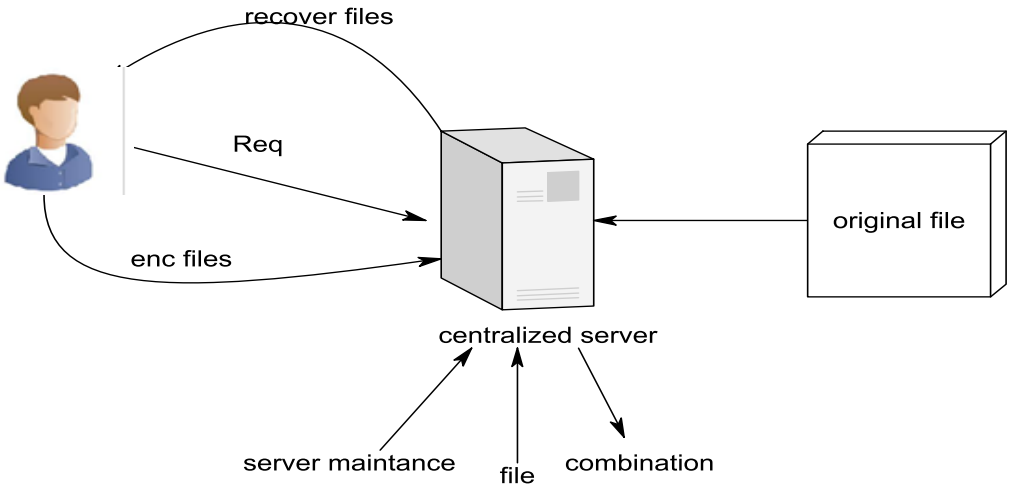

Figure-1 SYSTEM ARCHITECTURE

Volume: 4, Issue: 3,Special Issue: 2 ,Apr,2016 ,ISSN_NO: 2321-3337

4. PROCESS DESCRIPTION

DATA COLLECTOR:

To ensure data availability, storage systems usually stripe data redundancy across multiple storage nodes (or servers). It operates by encoding data into multiple fragments, such that any subset of fragments can sufficiently reconstruct the original data. On the other hand, a node is temporarily failed if its stored data is not lost but is only temporarily unavailable for direct accesses. To access the unavailable data, a storage system performs a degraded read, which retrieves data from surviving nodes and reconstructs the unavailable data. The key objective for permanent failure recovery is to read data from surviving nodes to reconstruct the lost data on a new replacement node.

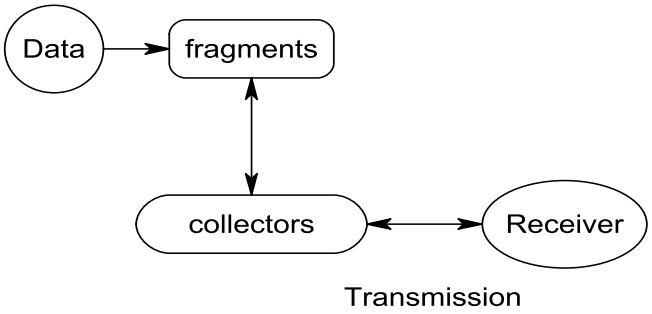

Figure-2 DATA COLLECTOR

REPLICATION MANAGEMENT:

Replication is traditionally used to provide data redundancy, yet it introduces high storage overhead and becomes a scalability bottleneck. HDFS stores data in units of blocks. By default, HDFS achieves fault tolerance via replication, and stores three copies for each block. HDFS stores data in units of blocks. By default, HDFS achieves fault tolerance via replication, and stores three copies for each block. On the other hand, erasure coding provides space-optimal data redundancy while achieving the same fault tolerance as replication. It operates by encoding data into multiple fragments, such that any subset of fragments can sufficiently reconstruct the original data. To ensure data availability, a storage system often introduces data redundancy via replication or erasure coding. As erasure coding incurs significantly less redundancy overhead than replication under the same fault tolerance, it has been increasingly adopted in large-scale storage systems.

Volume: 4, Issue: 3,Special Issue: 2 ,Apr,2016 ,ISSN_NO: 2321-3337

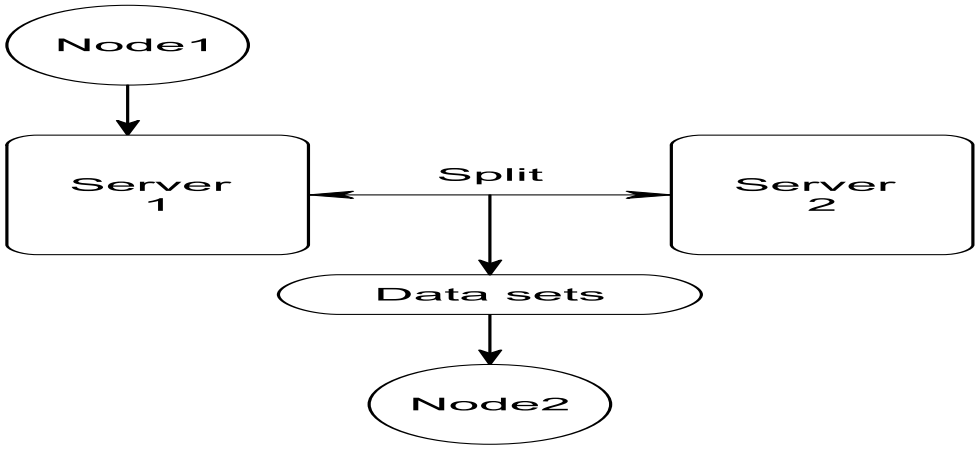

Figure-3 REPLICATION MANAGEMENT

RECOVER ERASURE CODE:

A storage system has typically deployed a specific erasure code for its stored data, and re-encoding a huge amount of stored data with a new code will introduce significant overheads. Also, different types of data are to be differently encoded based on the fault tolerance and access requirements. Each stripe is independently encoded, so our analysis focuses on a single stripe. Note that the identities of data and parity nodes are usually rotated across stripes in the implementation of storage systems for load balancing. New erasure codes which achieve efficient degraded read performance. However, a storage system has typically deployed a specific erasure code for its stored data, and re-encoding a huge amount of stored data with a new code will introduce significant overheads. Also, different types of data are to be differently encoded based on the fault tolerance and access requirements . Thus, instead of constructing new codes, we aim to develop efficient degraded read strategies that are applicable to a general class of existing erasure codes.

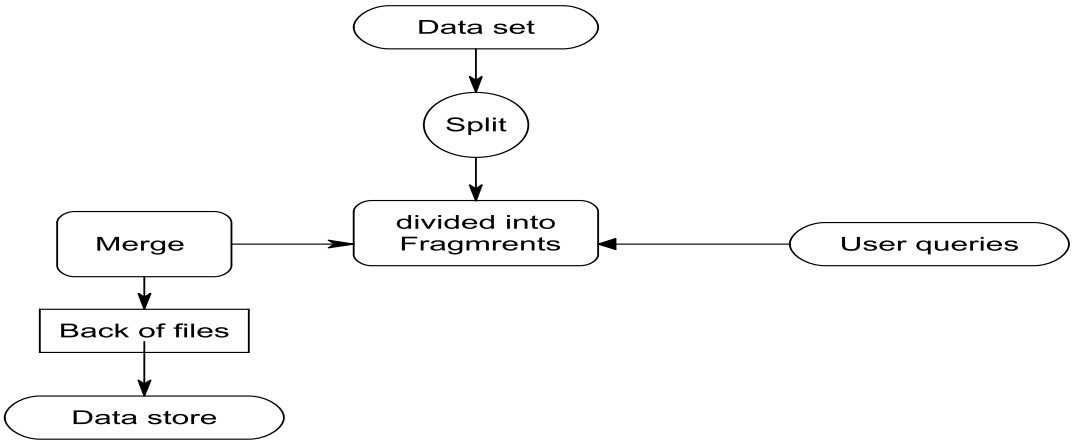

Figure-4 RECOVER ERASURE CODE

Volume: 4, Issue: 3,Special Issue: 2 ,Apr,2016 ,ISSN_NO: 2321-3337

CONCLUSION AND FUTURE WORK

Degraded reads have become performance critical operations, due to the fact that temporary errors account for the majority of failures in modern storage systems. To boost the performance of degraded reads in practical erasure-coded storage systems, it is necessary to take into account parallel I/Os and node heterogeneity when performing degraded reads. In addition, as the parameters of degraded reads vary, it is also crucial that the degraded read solution should be found in a timely manner. In this paper, we propose FastDR, a system for boosting degraded reads in XOR-based erasure coded storage systems with heterogeneous storage nodes. We formulate an optimization model, and further propose an enumerated greedy algorithm to quickly find an efficient degraded read solution in heterogeneous erasure-coded storage systems. Through extensive simulations and test bed experiments, we justify the effectiveness of FastDR in achieving efficient degraded reads in a heterogeneous storage environment.

Instead of proposing a new code construction, we optimize degraded reads for existing erasure codes, and we address heterogeneous storage systems and identify the effective solution with minimal search time.

REFERENCE

[1] Yunfeng Zhu, Jian Lin, Patrick P. C. Lee, and Yinlong Xu, "*Boosting Degraded Reads in Heterogeneous Erasure-Coded Storage Systems*".

[2] Osama khan, Randle burns,James plank, William pierce, "*Rethinking erasurecodes for cloud file system*" , 2005.

[3] Kevin M.greenan, ethan L.Miller, Jay J.Wylie, "*Reliability of flat Xor based erasure codes on heterogenous devices*", 2008.

[4] Alexandros G.Diemakis, Yunnan Wu, Martin O.Wain Wright, "*Network coding for distributed storage system*", 2008.