Volume 5, Issue 4, April 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Adaptive Replica Synchronization for Distributed File Systems

Shelake Seemadevi M.* M.E CSE, Student SIETC, Paniv, CSE Dept. Solapur University, Solapur, Maharashtra, India

Prof. Yevale Ramesh S. M.E.(CSE) Head of CSE Dept. SIETC, Paniv, CSE Dept. Solapur University, Solapur, Maharashtra, India

Prof. Prakash. B. Dhainje MTech, MBA(IT), PhD(CSE) Principal SIETC, Paniv, CSE Dept. Solapur University, Solapur, Maharashtra, India

Abstract— In this paper we have reviewed and studied various replica synchronization at storage server (ss) and metadata server (MDS) in distributed file systems. There are various functions which must be fulfilled while replica management such as data consistency, reliability, etc. To ensure data consistency data structure is used to store the information about replicas and associated SSs corresponding to the replicas and to achieve reliability system should be low cost. Replica synchronization that is reliant on the MDS to trigger the synchronization, centralized mechanism is used. This contributing to I/O data rate in write-intensive workloads. The evaluation show that this newly presented mechanism can achieve attractive I/O performance enhancement with less synchronization overhead in specific application contexts.

Keywords— Adaptive replica synchronization, deferred update, distributed file systems, I/O data rate, version-based update replay.

I. INTRODUCTION

Now days Modern cloud auditing has been studied. Cloud computing storage provides their service in cloud computing ,like sending of data storage as a service, database-like services and network based storage, often billed on a utility computing basis, e.g., per gigabyte per month. Examples include Amazon SimpleDB1, Microsoft Azure storage2, and so on. In cloud storage services, the customers can access data stored in a cloud easily using any device, without taking care of a large amount of capital investment. To improve I/O performance and system reliability, the file system used in a distributed computing environment is called as a distributed file system. It gives multiple distributed I/O devices by transferring file data to the I/O nodes and uses maximum bandwidth to meet the I/O requirements of distributed applications. Generally the client sends request to the metadata server (MDS), which handles all the functions of file system, to get the permission to operate on the file and the information of the file's layout. Then, the client accesses the corresponding storage servers (SSs), which handle the file data management on storage devices, to perform the real file I/O operations after parsing the layout information obtained from the MDS.

II. REPLICATION IN CLOUD STORAGE

a) MinCopysets: Derandomizing Replication in Cloud Storage:

MinCopysets is a general replication scheme that can be implemented on a distributed file systems that distribute chunks randomly to the data center. This mechanism implements MinCopysets in RAMCloud and HDFS. Randomized node selection is very important task of this mechanism for load balance chunks of data over the cluster and to provide durability, selection of replica in large-scale, distributed system is essential. Selection of node randomly is good for load balancing, but it fails to protect data. Asaf Cidonpresent et al. [1] proposed Min Copysets, a simple, general-purpose and scalable replication technique to improve data durability and to protect data while having benefits of randomized load balancing. In this system the mechanism separates load balancing from data replication, we can use randomized node selection for load balancing but derandomizing node selection for data replication. This improves the data durability. Fig.1 shows the overall illustration of MinCopysets replication scheme.

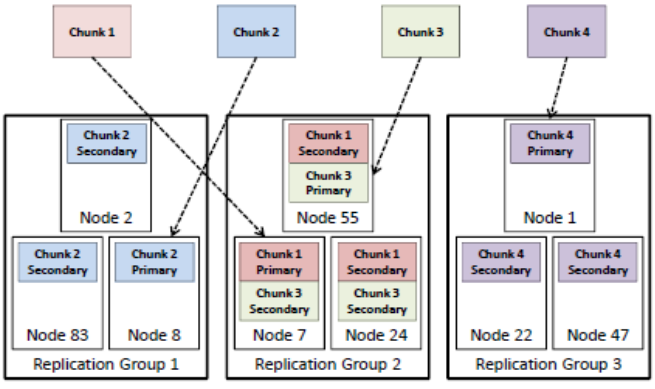

Fig.1.Illustration of the MinCopysets replication scheme

The first replica (primary replica) is chosen at random from the entire cluster. The other replicas (Secondary replicas) are replicated on the statically defined replication group of the first replica.
The main advantage of this system is it doesn't require change in storage system rather MinCopysets provides the storage system with the freedom to decide how to split the nodes into replication groups. Min-Copysets chooses an entire replication group, which consists of a fixed set of nodes. In case of a node failure, the entire replication group is impacted and this becomes overhead for node recovery.

b) Copysets: Reducing the Frequency of Data Loss in Cloud Storage:

To prevent data loss in data center storage systems random replication is used. But random replication is losing data due to node failures. Due to the high fixed cost of each incident of data loss, many data center operators prefer to minimize the frequency of such events at the expense of losing more data in each event. Asaf Cidon et al. [2] present Copyset Replication which minimizes the frequency of data loss events occurred. They implemented and evaluated Copyset Replication on two data center storage systems, HDFS and RAMCloud. Such systems require that each node's data be scattered across several nodes for parallel data recovery and access. Copyset Replication provides between the number of nodes on which the data is scattered and the probability of data loss.

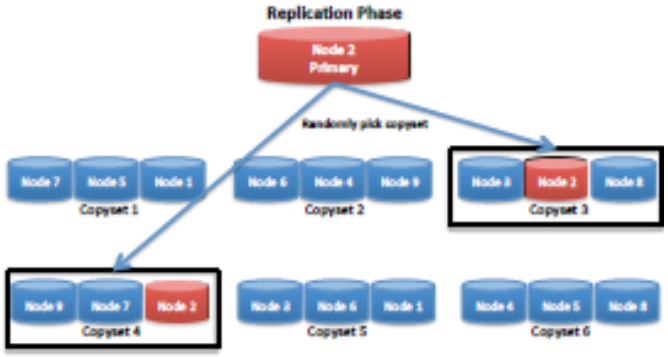

Fig.2 Illustration of the Copyset Replication Replication phase.

Fig.2. shows that design of copyset replication system.It has two phase:permutation and replication. The permutation phase is conducted offline, and whenever chunk needs to be replicated the replication phase is needed.

c) Exploring Data Reliability Tradeoffs in Replicated Storage Systems:

Abdullah Gharaibeh, Matei Ripeanu [3] presents the efficiency of a cost-efficient storage architecture that gives reliability and access performance of a high-end system. This architecture provides two characteristics: 1. Removal of idle storage from LAN connected desktops gives a low-cost storage space and high I/O throughput by taking participation of nodes. 2. The two parts of data reliability – durability and availability – can be separated to control the cost of overall system. The system needs to restore the data the volatile nodes may lose. But these nodes provide a high-throughput. While combining these components no. of terms need to be addressed like high throughput, low cost, and durability, etc. They implemented tool to evaluate the impact of system characteristics and design choices based on availability of data and the associated system costs.

d) The Hadoop Distributed File System:

The Hadoop Distributed File System (HDFS) is implemented to store very large database easily, and to spread those data to user applications. In large cluster, many of servers are directly attached to the storage and execute user application. By spreading storage and computation among servers, the resource can grow with demand while remaining economical at every size. The overall architecture shown in fig.3.

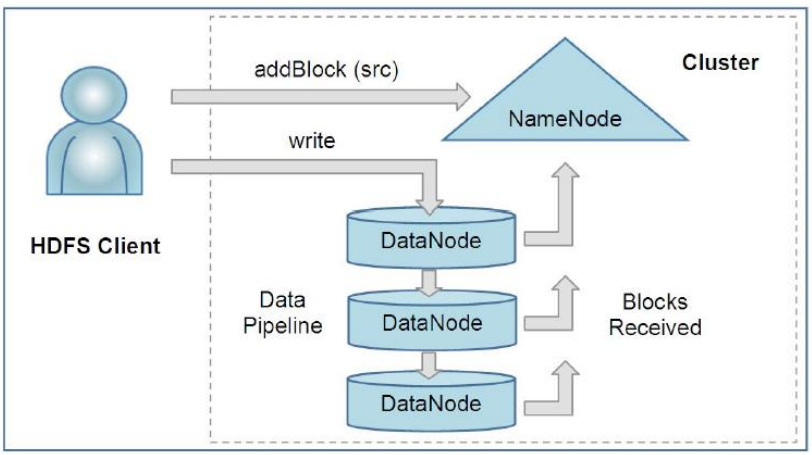

Fig.3 architecture of Hadoop File system

Fig.3. shows an HDFS client gives path of newly created file the NameNode. NameNode returns a list of DataNodes to host For each block of the file,. The client then pipelines data to the chosen DataNodes, which eventually confirm the creation of the block replicas to the NameNode.

e) Ceph: A Scalable, High-Performance Distributed File System:

Sage A. Weil et al.[5] developed distributed file system i.e. Ceph which provides very good performance, reliability, and scalability. It increases the distance between data and metadata management by doing replacement allocation tables.

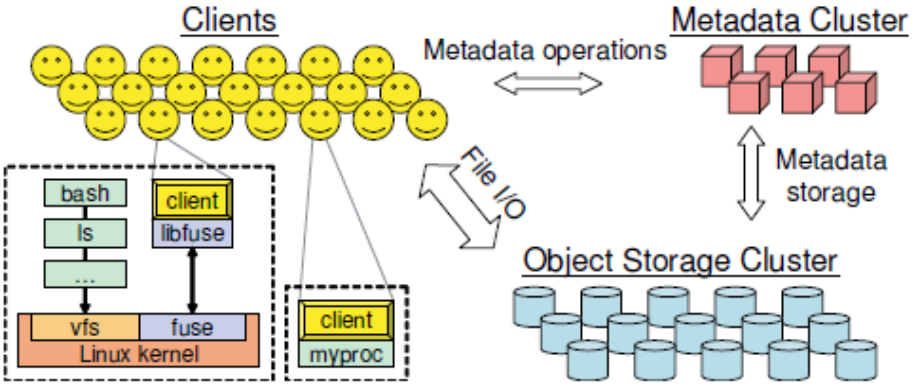

Fig .4. System architecture.

Fig.4. shows Clients communicate with OSDs to perform basic I/O operations. Each process can
Connect directly to a client instance or interact with a mounted file system. The primary goals of the system are to provide characteristics like scalability, reliability and performance. Scalability is in terms of dimensions, overall storage capacity and throughput of the system, and performance with respect to individual
Clients, directories, or files. The main target achieved is when many clients can access read and write operation on system by implementing all the characteristics.

g) Adaptive Replica Synchronization for Distributed File Systems:

Jianwei Liao et al. [6] present adaptive replica synchronization. This mechanism uses Gmei file system to take replica synchronization. Next, concept is version-based update replay which ensures data consistency. This mechanism has the following three contributions:

1. Replication of chunk lists to SSs
2. Lazy and adaptive replica synchronization
3. Version-based update replay

Gmei illustrates the feasibility and applicability of the main purpose of this system. As a matter of fact, these ideas can be applied to other file systems e.g. Lustre, GFS, etc. now adding or removing replicas issued by the MDS results in the synchronization to the related SSs, on which the replicas and relevant chunk lists are stored. So, it can increase robustness of system using adaptive replica synchronization.

III. CONCLUSION

In this paper, we have analyzed and studied various techniques of replication in cloud storage and distributed systems. It is necessary to achieve reliability, efficiency and performance during replica management. Synchronization of replica among various systems can effect on performance of the whole distributed system. To handle distributed system security or robustness is also another important issue. For synchronization of replicas in distributed system Gmei file system is better to manage storage servers. It gives flexibility, applicability of various systems.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Dr. Deshmukh Pradeep K for their guidance and to Prof. Prakash. B. Dhainje for their supervision. We would like to express our appreciation to our parents and all the teachers and coordinators who help us to understand the importance of knowledge and show us the best way to gain it.

REFERENCES

[1] A. Cidon, R. Stutsman, S. Rumble, S. Katti, J. Ousterhout, and M. Rosenblum, *"MinCopysets: Derandomizing replication in cloud storage,"* Stanford Univ., Stanford, CA, USA, Tech. Rep., 2012.

[2] A. Cidon, R. Stutsman, S. Rumble, S. Katti, J. Ousterhout, and M. Rosenblum, "Copysets: Reducing the frequency of data loss in cloud storage," in *Proc. USENIX Conf. ATC*, 2013, pp. 37–48.

[3] A. Gharaibeh and M. Ripeanu, "Exploring data reliability tradeoffs in replicated storage systems," in *Proc. HPDC*, 2009, pp. 217–226.

[4] R. J. Chansler, "Data availability and durability with the Hadoop distributed file system," *USENIX Mag.*, vol. 37, no. 1, pp. 16–22, 2012.

[5] S. Weil, S. Brandt, E. Miller, D. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in *Proc. 7th Symp. OSDI*, 2006, pp. 307–320, USENIX Association.

[6] Jianwei Liao, Li Li, Huaidong Chen, and Xiaoyan Liu," Adaptive Replica Synchronization for Distributed File Systems" 2014 IEEE.

