INTERNATIONAL JOURNAL OF ADVANCE RESEARCH AND DEVELOPMENT

(Volume3, Issue4)

Available online at: www.ijarnd.com

Improving security and performance by implementing FS Algorithm in distributed cloud

Omana M U[1], Nandhini S[2], Priyadharshini R[3]

[1,2]Student, Sree Sakthi Engineering College, Karamadai, Tamil Nadu
[3]Assistant Professor, Sree Sakthi Engineering College, Karamadai, Tamil Nadu

ABSTRACT

The distributed file system is used to store large quantity of files. The availability of the file is considered as the main factor for the availability of the system. The availability of the system is based on the retrieval time of the file. Metadata servers are separated from the data server and the replication of metadata is being implemented in the distributed file system to increase the availability. But there is a security issue in replication of metadata in a single server. When whole data is in sequential order, if a single server gets hacked or failure occurs the entire data will be lost from the server. To overcome this problem, apply FS Algorithm. Scope of the paper is to Detach and reproduce methodology: divide a file into fragments, and replicate the fragmented data over the cloud node in a shuffling order.

Keywords: *Metadata, Replication, Fragment, shuffling and sequential.*

1. INTRODUCTION

Parallel computing is computing in which multiple processors are used to perform multiple operations simultaneously. Parallel computing is derived from serial computing. Parallel computing is also known as parallel programming. The large work is divided into small tasks and workload is shared between more than one processor. Then the solution from the multiple processors is combined and a single solution is obtained. The goal of the parallel computing is to increase the available computation power. Distributed file system involves parallel processing of files. The examples for distributed file systems are HDFS, PVFS.., the parallel processing of data in the distributed file system may provide interoperability. The purpose of the parallel computing is to increase the retrieval time of the data and sharing of the resource efficiently.

2. LITERATURE SURVEY

General purpose architecture for replicated metadata services in distributed file systems reveals the concept of replication of the data in distributed file system [1]

Availability in global distributed storage systems says about the performance of the servers in the storage [3]

Ceph: A scalable, High-performance distributed file system explains how it can be increased [4]

MapReduce: Simplified data processing on large clusters deals with the fragmentation of the files [2]

DRAM error in the wild: A large-scale field study say the working principles of the storage systems [11]

Scalability of Replicated Metadata Services in Distributed File System [6]

3. EXISTING SYSTEM

The two distributed file system (here Hadoop File System and parallel virtual file systems) are taken and they are compared to each other. The separate metadata server and replication of the metadata are implemented in the both the file system. Since both HDFS and PVFS is considered to be prominent file systems they already provide a high available metadata services. The database used here is Oracle Berkeley DB (BDB). The entire architecture is divided into three layers such as

- Network Layer
- Metadata server
- Replicated DB

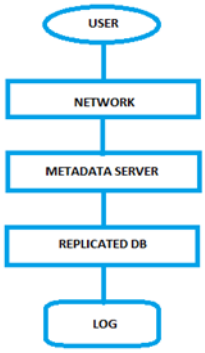
Here the data are store in the data server and their metadata are taken and stored in a separate server. To improve performance the Replication of the data is carried out. The metadata is mapped with its original data. When the data is requested it is directly fetched from the server.

The whole file is stored in a sequential order and it is replicated in multiple servers to avoid the security issues and performance degradation. To provide recovery of the data zookeepers are maintained in the original HDFS.

The drawback of the existing system is:

When the files are stored in a sequential order, if a hackers hacks the system he may get the entire file

- The upload or download a file is a difficult process
- The time taken for retrieval of file may long time since it is fetched from single server



The concept works well for the HDFS and PVFS the only issue is with security and the performance of the system.

When compared to normal HDFS and the HDFS- RMS the HDFS-RMS works more effectively.

4. PROPOSED SYSTEM

The proposed system is designed in such a way that it to overcome the security and performance issue. Here the cloud storage is used instead of the distributed file system. The data to be stored in the servers are first fragmented and they are shuffled. For the fragmentation and shuffling FS algorithm is used. It is used for the purpose of avoiding the security issue. The encryption of the data is also provided, which may also be very efficient. Shuffling the data is carried out in a way that no single server may contain the full data. Even if it contains the full data it will be in a meaningless form.

To increase the performance the concept of load balancing is also implemented. Since the data is stored in multiple servers, if the data is requested it can be retrieved from any one the servers which is free at that time.

As the log servers are maintained in cloud the storage for metadata is reduced. The entire architecture in divided into four module. They are

- User Authentication
- Fragmentation
- Data Replication and Data Encryption
- Data Retrieval and Decryption

a. USER AUTHENTICATION

User authentication verifies identity of the user. The user should provide some kind of proof for identity that the system can understand and trust. The login ID for the user is created. The user id consists of two parameters, first parameter can be name and second can be callback handler used for passing login information to the log server. The call back handler is used transfer the required information to the log server.

b. FRAGMENTATION

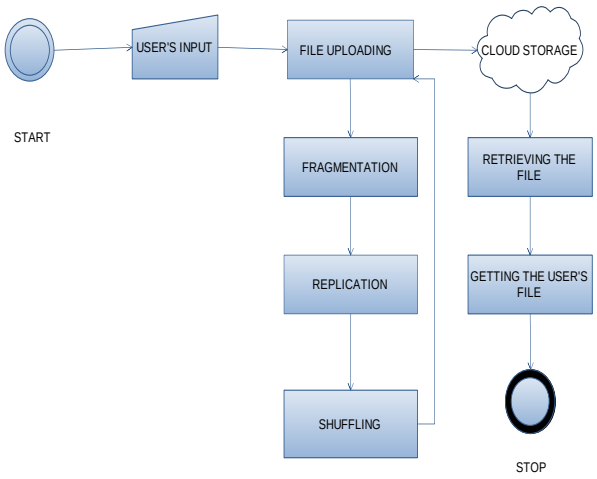
Fragmentation process is carried out for splitting the files into small fragments. When the files are fragmented, the cloud nodes for fragments are been selected. They are placed in such a way that all fragments are placed in a cloud node. The files are fragmented based the length of the files. Partial replication represents the fragment placement methodology. This module mainly focuses on the storage system security during the process. Here the probability of a successful coordinated is extremely less

c. DATA REPLICATION AND DATA ENCRYPTION

This module is used to support the replication mechanism by checking replicas and managing their execution based on the client's requirement. The set of VM are controlled by a single implementation of a replication mechanism as a replica group. Each replica can be identified uniquely a set of rules in which is specified. The replication manager is maintained to make the client perceive a replica group as a single service. It ensures that the fault free replicas exhibit correct behavior during execution time. It analyzes the expected server performance properties and interact with the resource manager to obtain the location of each replica. Client uploaded data are encrypted for securing the data in cloud storage in different virtual server with fragments

Cloned fragments are shuffled by FS- algorithm. They are shuffled like (1-4,2-1,3-2,4-3). Once the user requested for the information, it will retrieve the necessary fragments in the sequential order. Once all the fragments are collected, that will produce entire information to the user.

DESIGN IMPLEMENTATION

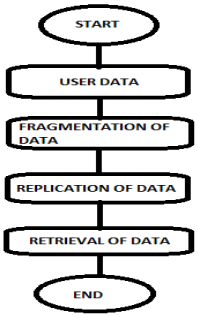


d. SERVER ANALYSIS

The server provider uses generic fault analysis mechanism to offer fault analysis as a service. So that client's applications deployed in VM instance can transparently obtain server status properties. It define ft-unit as the fundamental module that applies a coherent server analysis mechanism to a recurrent system failure at the granularity of a VM-instance. The design stage starts when a client requests the service provider to offer server analysis to offer server analysis support to its applications. In this stage, the service provider must first analyze the client's requirements, match them with available ft−units, and form a complete server analysis solution using appropriate ft−units. We note that each ft−unit offers a unique set of server properties that can be characterized using its functional, operational, and structural attributes.

e. DATA RETRIEVAL AND DECRYPTION

This module describes about the retrieval of data from different virtual server only authenticate user. Data are retrieved from different virtual server and combine the data and convert to decrypted format. Decrypted data are bringing data are bringing to original data for user extraction. The goal of this component is to achieve system-level resilience by minimizing the downtime of the system during failures. To this aim, this component supports ft-units that realize recovery mechanisms so that an error-prone node can be resumed back to a normal operational mode.



11. CONCLUSION

In this paper, a general-purpose architecture of replicated metadata services in distributed file systems, named RMS has been proposed. The evaluation shows that an RMS variant of HDFS performs comparably to native implementations when contrasted in micro benchmarks. However, this has to come at the expenses of durability: Write transactions with synchronous commits to disk

are more expensive in HDFS-RMS (SYNC mode) vs. HDFS-HA with similar, behavior, pointing to the efficiencies possible in special-purpose designs (HDFS-RMS); one can improve HDFS-RMS performance to the levels of HDFS-HA by offering somewhat weaker durability semantics using the NOSYN mode, which we consider as an acceptable tradeoff in replicated setups. Our experience with tuning our HDFS-RMS prototype implementation provides: In most cases, including application level benchmarks, RMS variants do not incur an end-end performance penalty. In terms of availability, the RMS variant of HDFS matches the recovery characteristics of HDFS-HA v2, a state of the art implementation. HDFS-RMS is resilient to the loss of any number of name nodes (assuming sufficient number of replicas), whereas HDFSHA allows only up to two name nodes. Given that the performance of RMS variants is not significantly affected when the degree of replication increases from 3 to 7 using a QUORUM ack policy, leads us to suggest configurations with 5 to 7 replicas as viable alternatives for real-world installations. Overall we have shown that use of RMS is straight forward and yield robust, performance distributed file systems that are easy.

12. RELATED WORKS

This idea can also be implemented without the use of metadata servers. The load balancing concept increases the performance and the scope of the project. The replication can be avoided in a single server. The maintanance of the zookeepers may provide the recovery mechanism of the data.

13. REFERENCES

[1] Stamatakis, D., Tsikoudis , N., Smymaki,O. and Magoutis , K.,"Scalability of Replicated Metadata Services in Distributed File System," in Pronc. Of 12 th IFIP int. Conference on Distributed Applications and Interoperable System (DAIS) 2012, Stockholm, Sweden, 2012.

[2]Ligon, M., Ross, R., "Overview of the Parallel Virtual File System." in Proceedings of USENIX Extreme Linux Workshop, Monterey , CA , USA, 1999

[3] Shvachko , K., Kuang, H., Radia, S. and Chansler , R., "The Hadoop Distributed File System," in Pronc. Of IEEE Conference on Mass Storage Systems and Technologies (MSST), Lake Tahoe, NV, 2010.

[4] Ghemawat, S., Gobioff,H. and Leung, S.-T., "The Google File System," in Pronc. Of 19 th ACM Symposium on Operating Systems Principle (SOSP-19), Bolton Landing, New York,2003

[5] Shepler, S. et al,"Parallel NFS, rfdc 5661-5664,"http://tools.ietf.org/html/rfc5661, IETF

[6] Theimer, M., "Some Lessons Learned from Operating Amazons Web Services" in keynote talk at ACM SIGOPS,MT, USA, October 2009

[7] Peacemaker "A Scalable High Availability Cluster Resource Manager, "http://clusterlabs.org

[8] [online]. Available: "http://github.com/ sambenz/ URingPaxos"

[9] A.Thomson and D.J Abadi, "Calvin FS: consistent bWAN Replication and scalable Metadata Management for Distributed File Systems, in Proc USENIX conference on file and storage Technologies

[10] Srinivas S, " An Introduction to HDFS Federations, " http:// hortonworks.com/blog/an-introduction-to-hdfs-federation

[11] Bianca Schroeder Dept. of Computer Science University of Toronto Toronto, Canada bianca@cs.toronto.edu. Eduardo Pinheiro Google Inc. Mountain View, CA Wolf-Dietrich Weber Google Inc.Mountain View, CA.

