Volume 5, No. 4, April 2014 (Special Issue)

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Novel Data Replication Strategy in HADOOP

[1]B.Sudheer Kumar, [2]D.Sraveena, [3]K.Durga Natasha*, [4] K.Sravanthi and [5]V.Harika
Dept. of Information Technology
ACE Engineering College, Hyderabad, India.

Abstract: Data management is the big issue in today's world where data is the collection of vast information. So fast and accurate retrieval of data is an important task. Not only the fast retrieval of data but also the recovery of lost data is also an important task. This can be resolved by using "REPLICATION" (Replication can be defined as the process of maintaining multiple copies of same data and sharing the data with multiple devices). The replication strategy in Hadoop is to store three replicas, two in one rack and third one in a rack that is nearest to the previous rack. So there is a 0.1% chance of losing two rack servers at a time so in order to resolve the issue we proposed a novel technique of data replication strategy to decrease the chance of losing data and also to increase the availability of data.

Keywords: HDFS, AFS, NFS, CODA, Replication factor, Replication, Blobseer.

I. INTRODAUCTION

Replication can be defined as the process of managing the copies of data and also arranging the copies of data in such a way that a change made in any copy reflects other copies too[1]. In simple terms, it can be defined as storing of similar data/information in multiple devices, which may be located in same rack or different rack depending on the algorithm designed for different file systems. It is based on 'Replication factor' which is referred to as the number of times data is replicated and the number of places the data is stored. The default replication factor depends on the different algorithms designed for different file systems like the three default replicas for Hadoop and eight for Blobseer and so on. The replication factor cannot be changed dynamically. In case if it needs to be changed, whole cluster needs to be restarted. Placing of replicas is another important factor which is also different according to the algorithms used for different file systems. For example, Blobseer places eight replicas in eight different servers by default. In hadoop, for pre-file replication value, a new file must be used in data pool with replication factor.

The cases considered are: when configuration value is not mentioned, then default pool is always used; when the configuration value is mentioned and replication factor is exactly found then make use of pool; when the configuration value is mentioned and inaccurate replication factor is found then make use of closest match. The importance of replication is to provide availability, efficiency and fault-tolerance, load balancing, performance, response time and accessibility between software and hardware components. The other important use of replication is that it reduces the network traffic, improves the system throughput, scalability and autonomous operations. This paper is organized as follows section II describes various replication strategies in different file system, section III Novel Technique , section IV Comparison table and section V conclusion.

II. DISTRIBUTED FILE SYSTEM

Distributed File System (DFS), as the name suggests, it is defined as the information/data distributed across different nodes making processing time less thereby increasing the throughput [2]. DFS supports an excellent feature called 'parallel processing' which means dividing a task among several nodes and executing all sub parts simultaneously. This feature made DFS much popular. DFS is a client-server service which provides redundancy which means duplication of data to improve data availability in case of failure due to data lose or due to heavy load. DFS helps the clients to access and share the data from anywhere round the globe within the cluster. The DFS serves the clients by considering different factors such as distance, performance etc. among the replicas. The DFS provides three types of services namely: storage service, true file service and name space. The storage service provides a logical view of the storage system. It provides the service for allocating and managing the space on secondary storage device. The file system should provide file sharing semantics, file caching mechanism, file replication mechanism, concurrency control, etc. The following sub-sections will describe about each of the above mentioned DFS with respect to replication strategy.

A. AFS:

File System (AFS) was under development since 1983' at CMU with the main goal is to create a scalable DFS which targets 5000 workstations[3]. The implementation is done in the UNIX environment. In AFS, local files are served like normal UNIX files. Only the authorized users can access AFS file space.AFS is most widely used in academics and research environments [4]. The organizational unit of AFS is called "CELL". A cell is the collection of server and client machines.

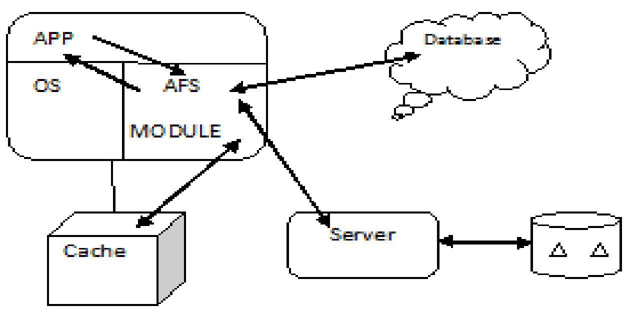

Figure 1. Replication strategy in AFS

CONFERENCE PAPER
Two day National Conference on Innovation and Advancement in Computing
Organized by: Department of IT, GITAM UNIVERSITY Hyderabad (A.P.) India
Schedule: 28-29 March 2014

Servers and clients can only belong to a single cell at a given time, but a user may have accounts in multiple cells. The cell that is first logged in is called the home cell and the remaining cells are called foreign cells.AFS uses random strategy for file replication. This strategy follows the principle that, it selects the data nodes randomly from the existing data nodes. The concept of invalidation message is not present in this distributed file system. That is, if one replica is modified or updated it does not affect other copies of the same data. That means other copies of the same data are not updated. The data that is stored in the AFS is stored in the form of cells. The data that is to be stored in the Andrew file system selects the location randomly for their storage. If any of the space is left empty then the file to be stored occupies the suitable location and stores the data.

B. CODA:

CODA is one of the distributed file systems, developed in the year 1990's [5]. The implementation is done in UNIX based Operating System such as LINUX. The main goal of CODA is to achieve high degree of naming and location transparency. By this, the system will appear to the users as similar as to a pure local file system. CODA is the advanced version of AFS. The architectural features of AFS are used in CODA. The additional concept included is the invalidation message. This concept explains that when two users are accessing the same data and one user has updated the existing data, then the second user gets the message about the updations. This concept is called as invalidation message.

This concept is also called as optimistic strategy.

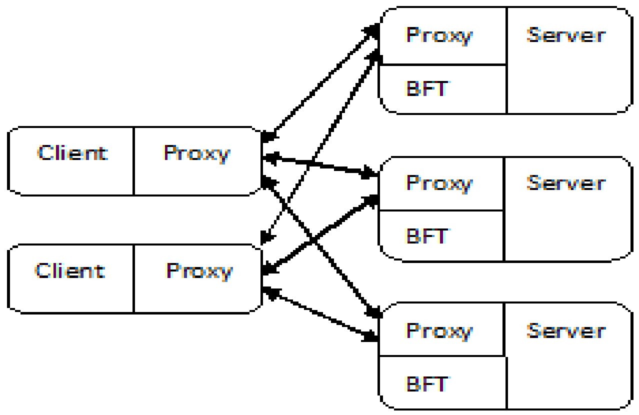

Figure 2. Replication strategy in CODA

CODA allows the file servers to be replicated. The unit of replication in CODA is called "VOLUME". The collection of servers that have a copy of volume is known as Volume Storage Group (VGS).This group contains the data which the clients can contact and access the required data. The client's Accessible Volume Storage Group (AVSG) for a volume consists of the servers in the VGS that the clients can contact. If the required data of the client is present in the AVSG then the sever responds to the client. If the AVSG is empty, then the client is said to be disconnected. The client can read the files from the AVSG. The process of reading a file goes on like, it first contacts with one of the AVSG members to which the file belongs to. Then the client sends the request to access the particular file and then gets the response back from the server. And in between if there are any updations and modifications made in that particular file then the client gets the message about the updations made and the files are updated automatically.

C. NFS:

Network File System (NFS) is one of the distributed file systems, developed by Sun Microsystems in 1984[6]. NFS allows the system administrator to distribute data across multiple servers in a way that is transparent to the users of the data. It uses virtual file system layer to handle local and remote files. It uses mount protocol to access remote files. It gives local names to the remote files and access those remote files through the local names given. NFS supports file locking.

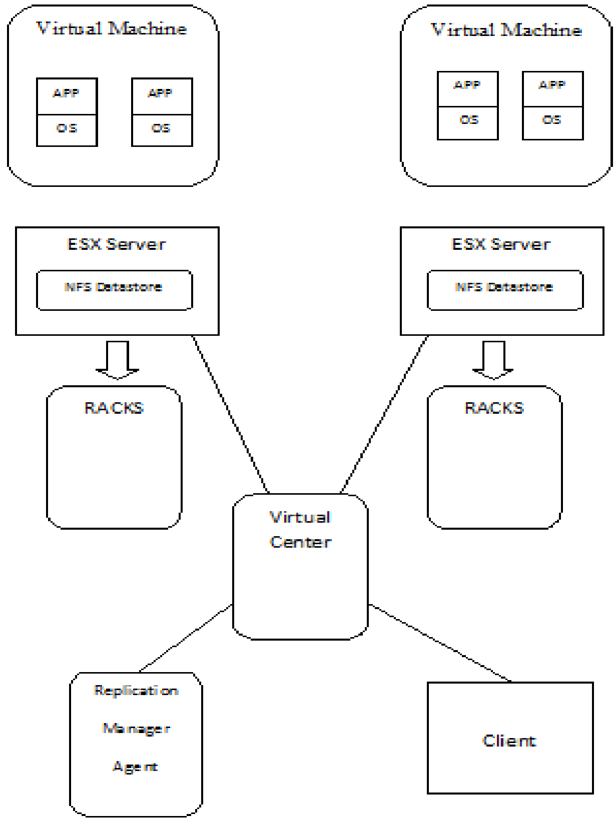

Figure 3. Replication strategy in NFS

It has two main features i.e., referral and replica [7]. Referral is an object in a namespace of the server. This referral contains the location information where the information is exactly present. It directs the clients to the data or location of information. Replica is a copy of file system. One NFS server can be placed on several other NFS servers. If one replica is lost, then the client switches to another replica. Replication allows the admin to place different copies of data on multiple servers and informs clients where the replica/copy of data is present. We can set up to 8 replica locations.

D. HADOOP:

HADOOP is one of the distributed file systems, developed in 1990's. In hadoop the data is stored in different racks of a server [8]. The main features are the name node and the data node. The name node contains the information about the locations of the data nodes and the data node contains the actual data what the user requires [9]. The data is stored in the form of racks. The total number of copies can be minimum of 3. The first two copies are placed in one rack and the third copy is placed in the next closest rack of the previous rack [10]. The drawback in this system is that if 2 racks are crashed at a time then all replicas are lost.

CONFERENCE PAPER
Two day National Conference on Innovation and Advancement in Computing
Organized by: Department of IT, GITAM UNIVERSITY Hyderabad (A.P.) India
Schedule: 28-29 March 2014

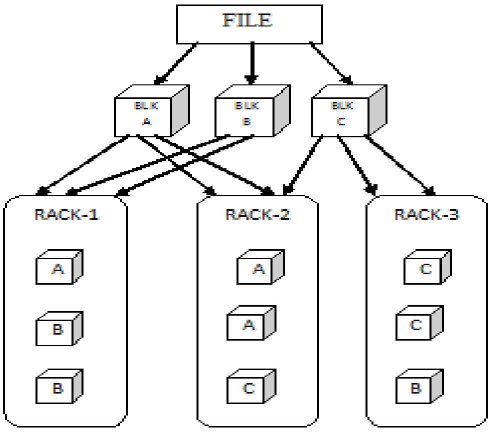

Figure 4.Replication strategy in Hadoop

III. NOVEL TECHNIQUE

The proposed algorithm is used for the efficient replication procedure. The data/information given is stored in minimum of three replicas at three different locations. The location where the data is present is provided in the name node and in the data node the actual information is present.

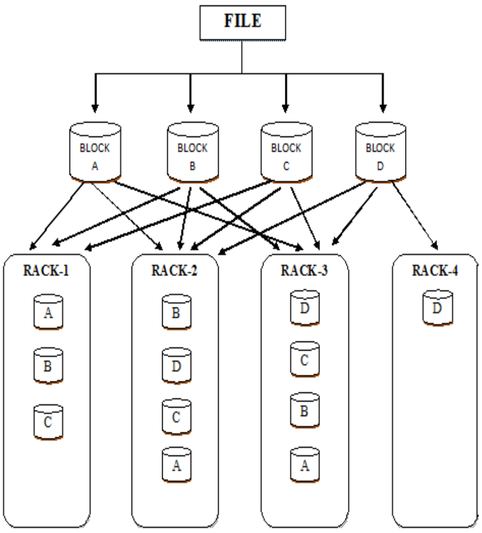

Figure 5. Replication strategy in Novel Technique

First and second copies of replicas are stored in any of the locations available and the third copy is stored based on the comparison of the highest memory and processor configurations. Different copies of the same data are placed in different racks. So the probability of data to be lost will be reduced. And proposed algorithm also explains that the memory and processor configurations are checked and the last replica is placed in other farther rack for fast and efficient retrieval of the data. The below figure is the block diagram of how the replication process is carried out in proposed algorithm.

Novel Technique Data Replication:

procedure Data management

while a file is requested to be saved until whole file is saved

Divide the file into blocks of 128kb

Prepare a list of available nodes for every rack server independently

for keep comparing available nearest nodes for configurability for every rack server independently

Arrange nodes in decreasing order of configurability for every rack server

if nodes in any rack servers have equal configurability

select nearest node from that rack servers

Sort the rack servers according to the nearest distance

Assign first node in first three different rack server to a, b, c respectively

save three copies of a file in a, b, c

end if

else if nodes of all rack servers have equal configurability

select nearest node from each rack server

Sort the rack servers according to the nearest distance

Assign randomly chosen nodes in first three differen rack server to a, b, c respectively

save three copies of a file in a, b, c

end else if

end for

end while

end procedure

When a file is asked to be saved, it is divided into blocks of 128kb. These blocks are one by one saved. The strategy used is first a list of available nodes is listed for every rack server independently. These nodes are arranged in descending order of configurability for every rack server. The rack servers are sorted according to nearest distance. The first node of first three rack servers is used to save the file. If the nodes have equal configurability, then nearest node is given preference.

Self adaptive:

Self adaptive technique in Hadoop:

There are minimum of three replicas in two racks, two in one rack server and other replica in other nearest rack server.

Case-1

Suppose rack with single replica is lost, the other two replicas in another rack will retain that lost replica.

Consider an example: Let RS1, RS2 be two rack servers and R1, R2, R3 be three replicas of a block. Now let us assume that R1, R2 be present in RS1 and R3 in RS2.When RS2 is lost or corrupted, then R3 will be retained either from R1 or R2 which is present in RS1.

Case-2

Suppose rack with two replica is lost, the other replica in another rack will retain that lost replica.

Consider an example: Let RS1, RS2 be two rack servers and R1, R2, R3 be three replicas of a block. Now let us assume that R1, R2 be present in RS1 and R3 in RS2.When RS1 is lost or corrupted, then R1 and R2 will be retained by R3 which is present in RS2.

Case-3

Suppose two racks are lost, then there is no chance to retain the replica.

Consider an example: Let RS1, RS2 be two rack servers and R1, R2, R3 be three replicas of a block. Now let us assume that R1, R2 be present in RS1 and R3 in RS2.When RS1 with R1, R2 and RS2 with R3 are completely lost or corrupted then we cannot retain them.

Self adaptive technique in Novel technique:

CONFERENCE PAPER
Two day National Conference on Innovation and Advancement in Computing
Organized by: Department of IT, GITAM UNIVERSITY Hyderabad (A.P.) India
Schedule: 28-29 March 2014

There will be minimum of three replicas in three different rack servers.

Case-1

If any rack is lost, the other two replicas in different racks will retain that lost replica.

Consider an example: Let RS1, RS2, RS3 be three rack servers and R1, R2, R3 be three replicas .Now let us assume that R1, R2,R3 be present in RS1, RS2, RS3 respectively. When RS1 is lost or corrupted, then R1 will be retained either from R2 present in RS2 or R3 present in RS3.

Case-2

If two racks are lost, then the replica in third rack will retain the lost replicas.

Consider an example: Let RS1, RS2, RS3 be three rack servers and R1, R2, R3 be three replicas .Now let us assume that R1, R2,R3 be present in RS1, RS2, RS3 respectively. When RS1, RS2 are lost or corrupted, then R1,R2 will be retained from R3 present in RS3.

Case-3

If all the three racks are lost which is rare to happen, then there is no possibility to get back the data.

Consider an example: Let RS1, RS2, RS3 be three rack servers and R1, R2, R3 be three replicas .Now let us assume that R1, R2,R3 be present in RS1, RS2, RS3 respectively. When RS1, RS2, RS3 are lost or corrupted, then R1,R2,R3 cannot be retained.

IV. COMPARISION TABLE

Table I. Analysis table for DFS

Distributed File System	*Replication Factor*	*Replication Strategy & Algorithm*	*Data Called*	*Self Adaptive*
AFS	-	Random Strategy and random algorithm	Cell: first logged in is home cell and remaining are foreign cells.	NO
CODA	-	Optimistic Strategy and optimistic algorithm	Volumes	YES
NFS	Minimum 8	Multiple server and Distributed algorithm	Data	NO
Hadoop	Minimum 3	Distributed Strategy and Nearest neighbor node algorithm	Blocks	YES
Novel technique	Minimum 3	High configuration and different racks	Blocks	YES

V. CONCLUSION

This paper has given an account of the present working strategies and the drawbacks of the various distributed file system. This paper also gives the importance of the efficient distributed system, the effective algorithm and self adaptive technique to overcome the existing problems. The reasonable approach to tackle the issue is the “Novel System”. This research has thrown up many questions in need to implement the “Novel System” for better and effective retrieval of data. In present Hadoop replication strategy, two replicas are placed in one rack and the third replica in the nearest rack to that of previous rack. Therefore there is a chance of losing two rack servers at the same time and in this novel technique even the performance of the self adaptive replication strategy can be improved. So to overcome this issue, ”Novel technique” is proposed. This helps to overcome the drawback and decrease the chance of losing the data.

VI. REFERENCES

[1] Salman Abdul Moiz, Sailaja.P, Venkataswamy.G, Supriya N.Pal. Database Replication: A survey of Open Source and Commercial Tools, Centre for Development of Advanced Computing, Banglore, 2011.

[2] Smita Hedge. Replication in Distributed File System. Department of Computer Science, University of Texas at Arlington, 2008.

[3] John H.Howard. An Overview of the Andrew Distributed File system. Information Technology Center, Carnegei Mellon University, 1988.

[4] Eliezer Levy, Abraham Silberschatz, ACM computing surveys, University of Texas , Austin , 1990.

[5] Peter J. Braam, M.Satyanarayanan. The Coda Distributed File System. School of Computer Science, Carnegie Mellon University.

[6] T.Haynes.Ed, D.Noveck.Ed. Network File System Version-4 Protocol, 2014.

[7] Russel Berg. The Sun Network File system: Design, Implementation and Experience, 1986.

[8] D.Borthakur. The Hadoop Distributed File System: Architecture and Design. The Apache Software Foundation, 2007.

[9] Konstantin Shvachko , Hairong Kuang , Sanjay Radia , Robert Chansler , IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST) ,USA , 2010.

[10] Debessay Fesehaye Kassa , Klara Nahrstedt, SCDA: SLA-aware cloud datacenter architecture for efficient content storage and retrieval, Proceedings of the 22nd international symposium on High-performance parallel and distributed computing, June 17-21, 2013, New York, USA.

Short bio data for the AUTHORS


Battula Sudheer Kumar did his M.Tech in Software Engineering from Jawaharlal Nehru Technological University (JNTU), Hyderabad in 2012, B.Tech from Jawaharlal Nehru Technological University (JNTU), Hyderabad in 2009.He is having 2 years of research experience and presently working as an Assistant Professor of IT department in ACE Engineering College, Hyderabad. His areas of interest include Distributed Systems and Cloud Computing.


D.Sraveena pursuing her graduation in Information Technology from ACE Engineering College, Ghatkesar and my research interests on Distributed Computing and Cloud Computing.


K.Durga Natasha pursuing her graduation in Information Technology from ACE Engineering College, Ghatkesar and my research interests on Distributed Computing and Cloud Computing.


K.Sravanthi pursuing her graduation in Information Technology from ACE Engineering College, Ghatkesar and my research interests on Distributed Computing and Cloud Computing.


V.Harika pursuing her graduation in Information Technology from ACE Engineering College, Ghatkesar and my research interests on Distributed Computing and Cloud Computing.