ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 2, May 2014

International Conference On Advances in Computer & Communication Engineering (ACCE - 2014)
on 21st & 22nd April 2014, Organized by
Department of CSE & ISE, Vemana Institute of Technology, Bengaluru, India

Fault Tolerance Techniques in Big Data Tools: A Survey

Manjula Dyavanur[1], Kavita Kori[2]

Asst. Professor, Dept. of CSE, SKSVMACET, Laxmeshwar-582116, India[1,2]

ABSTRACT: Big Data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process the data within a tolerable elapsed time. Fault tolerant system is one that can provide continue correct performance of its specified tasks in presence of failure.This paper is based on a survey of different kind of fault tolerance techniques in big data tools such as Hadoop and MongoDB.

KEYWORDS: Big Data, Big data Tools, Fault tolerance, Hadoop, MongoDB

I. Introduction

In simplest terms, the phrase Big Data refers to the tools, processes and procedures allowing an organization to create, manipulate, and manage very large data sets and storage facilities. Does this mean terabytes, petabytes or even larger collections of data? The answer offered by these suppliers is "yes." Big data tools are Hadoop,Splunk,MongoDB, FlockDB,Hibari and so on.

Fault tolerance is an important issue in big data; it is concerned with allthe techniques necessary to enable a system to tolerate software faults remaining in the system after its development. The main benefits of implementingfault tolerance in big data include failurerecovery, lower cost, improved performance etc. [1].When multiple instances of an application are running on several machines and one of the servers goesdown, there exists a fault and it is implemented by fault tolerance.

Here we considered two big data tools; Hadoop and MongoDB. In Hadoop Replication and Check point methods are used and in MongoDB Replication method is used for fault tolerance which are described briefly in section 3 and 4. Other Big data tools use one of these or both fault tolerance approaches.

II. BACKGROUND TERMINOLOGY

A fault tolerance is a setup or configuration that prevents a computer or network device from failing in the event of an unexpected problem or error [2]. To make a computer or network fault tolerant requires that the user or company to think how a computer or network device may fail and take steps that help prevent that type of failure. The path of generation of fault isshown in a figure 1.

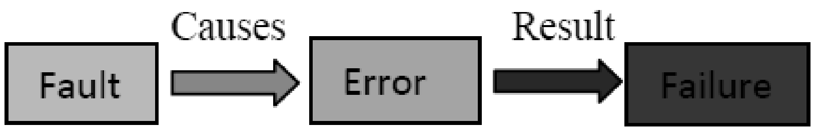

Fig. 1 Generation path of failure

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 2, May 2014

International Conference On Advances in Computer & Communication Engineering (ACCE - 2014)
on 21st & 22nd April 2014, Organized by
Department of CSE & ISE, Vemana Institute of Technology, Bengaluru, India

A system is said to fail when it will not fulfillthe requirements.An error is part of the system state that maylead to a failure.The cause of an error is a fault [3].The faults may be transient,intermittent or permanent faults, design faultsor operational faults.

III. FAULT TOLERANCE IN MONGODB

Replication provides redundancy and increases data availability. With multiple copies of data on different database servers, replication protects a database from the loss of a single server. Replication also allows you to recover from hardware failure and service interruptions. With additional copies of the data, we can dedicate a server to disaster recovery, reporting or backup.

In some cases, replication can be used to increase read capacity. Clients have the ability to send read operations to different servers. We can also maintain copies in different data centers to increase the locality and availability of data for distributed applications.

A. Replication in MongoDB

A replica set is a group of instances that host the same data set. One of them isprimary, receives all write operations. All other instances, secondaries, apply operations from the primary so that they have the same data set.

The **primary**as shown Figure 2.accepts all write operations from clients. Replica sets can have one and only one primary at any given moment. Because only one member can accept write operations, replica sets provide **strict consistency**. To support replication, the primary records all changes to its data sets in its oplog.

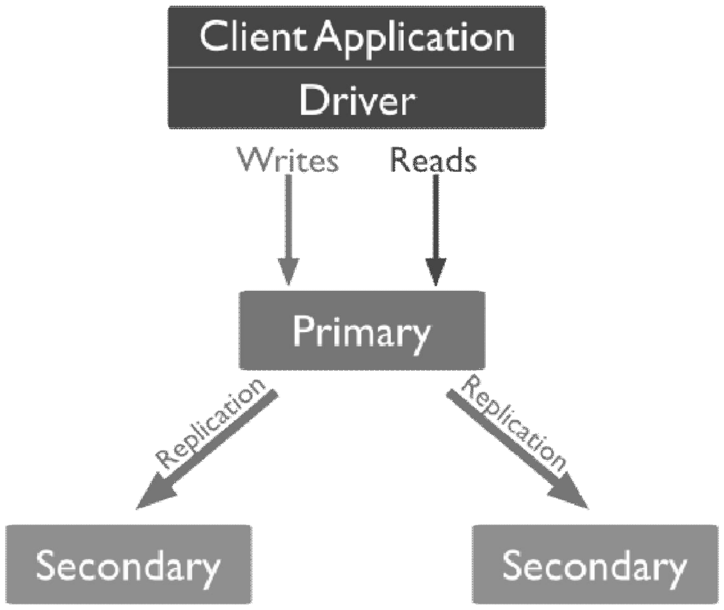

Fig. 2 Default routing of reads and writes to the primary

The **secondaries** replicate the primary's oplog and apply the operations to their data sets. Secondaries' data sets reflect the primary's data set. If the primary is unavailable, the replica set will elect a secondary to be primary. By

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 2, May 2014

International Conference On Advances in Computer & Communication Engineering (ACCE - 2014)
on 21st & 22nd April 2014, Organized by
Department of CSE & ISE, Vemana Institute of Technology, Bengaluru, India

default, clients read from the primary; however, clients can specify a *read preference* to send read operations to secondaries.

Here we can add an extra instance to a replica set as an **arbiter**. Arbiters do not maintain a data set. Arbiters only exist to vote in elections. If your replica set has an even number of members, add an arbiter to obtain a majority of votes in an election for primary. Arbiters do not require dedicated hardware.

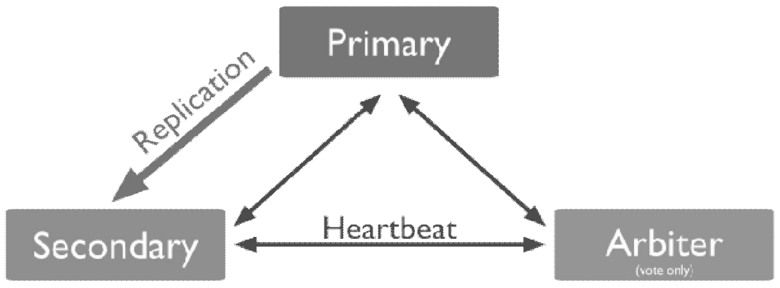

Fig. 3 Replica set that consists of a primary, a secondary, and an arbiter

The **primary** may, under some conditions, step down and become a **secondary**. A **secondary** may become the primary during an election. An **arbiter**, however, will never change state and will always be an arbiter.

B. Asynchronous Replication

Secondaries apply operations from the primary asynchronously. By applying operations after the primary, replica sets can continue to function without some members. However, as a result, secondaries may not return the most current data to clients.

C. Automatic Failover

When a primary does not communicate with the other members of the replica set for more than 10 seconds, as shown in Figure 4.the replica set will attempt to select another member to become the new primary. The first secondary that receives a majority of the votes becomes primary.

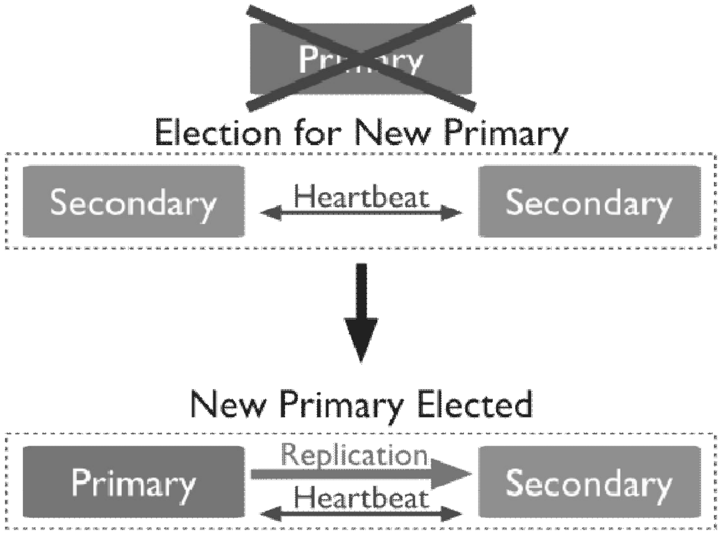

Fig. 4. An election of a new primary. In a three member replica set with two secondaries, the primary becomes unreachable. The loss of a primary triggers an election where one of the secondaries becomes the new primary

IV. FAULT TOLERANCE IN HADOOP

Each Hadoop cluster contains variety of nodes as shown in figure 5, hence HDFS architecture is broadly divided into following three nodeswhich are,

- Name Node.
- Data Node.
- HDFS Clients/Edge Node.

Name Node

It is centrally placed node, which contains information about Hadoop file system [5]. The main task of name node isthat it records all the metadata & attributes and specific locations of files &data blocks in the data nodes [6]. Namenode acts as the master node as it stores all the information about the system [7]. As name node acts as the masternode it generally knows all information about allocated and replicated blocks in cluster. It also has information about the free blocks which are to be allocated next. The clients contacts to the name node for locating information within the file system and provides information which is newly added, modified and removed from data nodes[8].

Data Node

The second type of node in HDFS architecture is data node. It works as slave node. Hadoop environment may contain more than one data nodes based on capacity and performance [5]. A data node performs two main tasks storing a block in HDFS and acts as the platform for running jobs. During the initial startup each data node performs handshakes with name node. It checks for accurate namespaces ID if found then it connects data node to name node, and if not then it simply close the connection [9] [6]. Each data node keeps the current status of the blocks in its node and generates block report. After every hour data node sends the block report to name node hence it always has updated information about the data node. During this handshaking process data node also sends heartbeats to name node after every 10 minutes, due to this action the name node knows which nodes are functioning correctly and which not. If name node doesn't receive heartbeats from data nodes it just assumes that data nodes are lost and it generates the replica of data node [7].

HDFS Clients/Edge node

HDFS Clients sometimes also know as Edge node [5]. It acts as linker between name node and data nodes. These are the access points which are used by user application to use Hadoop environment [6]. In the typical Hadoop cluster there is only one client but there are also many depending upon performance needs [5]. When any application wants to read a file it first contacts to the name node and then receive list of data nodes which contains the required data , hence after getting that list the clients access the appropriate data node requesting the data node which can hold that file also location of replica's which is to be written. Then the name node allocates the appropriate location for that file.

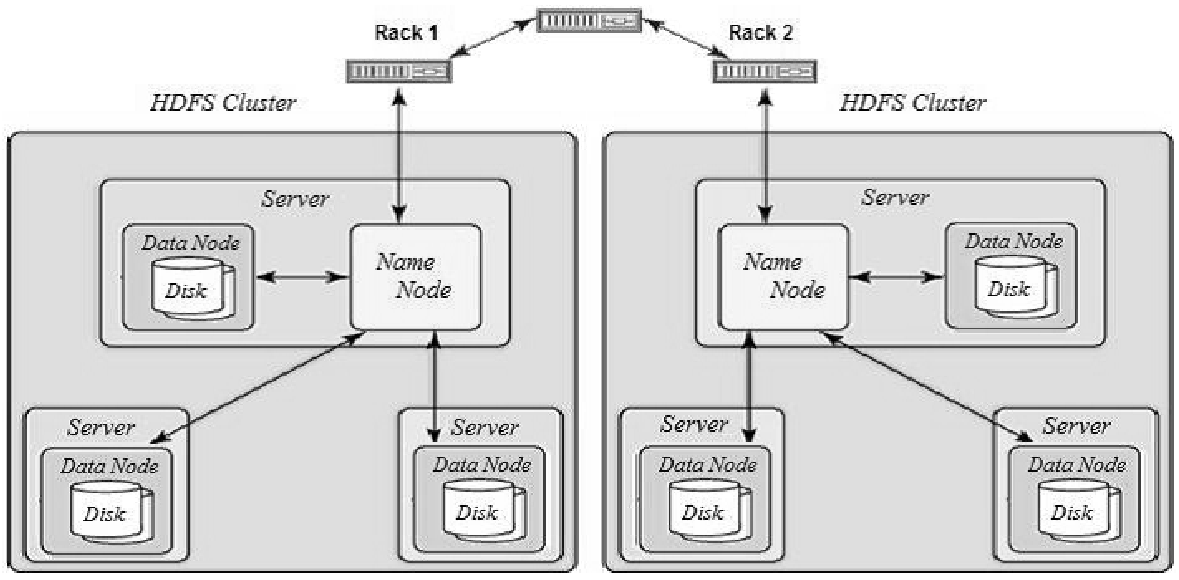

Fig 5.Skeleton of HDFS of Hadoop

When the system continues to functions correctly without any data loss even if some components of system have failed to perform correctly. It is very difficult to achieve 100% tolerance but faults can be tolerated up to some extent. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets [10]. The main purpose of system is to remove common failures, which occurs frequently and stops the normal functioning of system. When a single node causes whole system to crash and fails such node are known as single point failure nodes. In faults tolerance system its primary duty is to remove such nodes which causes malfunctions in the system [11]. Fault tolerance is one of the most important advantages of using Hadoop. There are mainly two main methods which are used to produce fault tolerance in Hadoop namely Data duplication and Checkpoint & recovery.

A. Data Duplication

In this method, the same copy of data is placed on several different data nodes so when that data copy is required it isprovided by any of the data node which is not busy in communicating with other nodes. One major advantage of thistechnique is that it provides instant recovery from failures. But to achieve such type of tolerance there is very largeamount of memory is consumed in storing data on different nodes i.e. wastage of large amount of memory & resources.As data is duplicated across various nodes there may be possibility of data inconsistency. But as this technique provideinstant and quick recovery from failures hence it is frequently used method compared to checkpoint and recovery.

B. Checkpoint & Recovery(Rollback)

In the second method, similar concept as that of rollback is used to tolerate faults upto some extent. After a fixed spanof time interval the copy report has been saved and stored. If the failure occurs then it just rollback upto the last savepoint and from there it start performing the transaction again. This method uses concept called rollback that is a
rollback operation brings the system to its previous working condition. But this method increases overall execution timeof system, because the rollback operations need to go back and check for the last saved consistent stages which increasethe time. Also there is one major drawback of this method is that it is very time consuming method compared to firstmethod but it requires less additional resources.

V.CONCLUSION

This survey paper includesBig data tools and also fault tolerance techniques used to Hadoop and MongoDB. we discussed about the architectural framework of Hadoop andalso some of the strategies to overcome the faults tolerance in the HDFS and MongoDB that includes data duplication and checkpoint and automatic recovery. This research can be extended by providing mechanism for handling breakdown in name node of HDFS of Hadoop. Also arrange some alternative and backup recovery for name node failure.

REFERENCES

1. Y.M. Teo, B.L. Luong, Y. Song, T. Nam,"Cost- Performance of Fault Tolerance in Cloud Computing, International Conference on Advanced Computing and Applications, (Special Issue of Journal of Science and Technology, Vol. 49(4A), pp. 61-73), Ho Chi Minh, Vietnam, October 19-21, 2011.
2. Ravi Jhawar, Vincenzo Piuri, Marco Santambrogioy," A Comprehensive Conceptual System-Level approach to Fault Tolerance in Cloud Computing,DOl 10.1109/SysCon.2012.6189503.
3. A. Christy Persya,Sr.Lecturer, T.R.Gopalakrishnan Nair," Fault tolerant real time systems, International Conference on Managing Next Generation Software Application (MNGSA-08), Coimbatore, 2008.
4. www.mongodb.org
5. Joey Joblonski. Introduction to Hadoop. A Dell technical white paper.
http://i.dell.com/sites/content/business/solutions/whitepapers/en/Documents/hadoop-introduction.pdf
6. Jared Evans CSCI B534 Survey Paper. *"Fault Tolerance in Hadoop for Work Migration"*
http://salsahpc.indiana.edu/b534projects/sites/default/files/public/0_Fault%20Tolerance%20in%20Hadoop%20for%20Work%20Migration_Evans,%20Jared%20Matthew.pdf
7. T. White. Hadoop: The Definitive Guide. O'Reilly, 2009.
8. Shvachko, K., *et al*. (2010) the Hadoop Distributed File System. *IEEE*.
http://storageconference.org/2010/Papers/MSST/Shvachko.pdf
9. Borthakur, D. (2007) The Hadoop DistributedFile System: Architecture and Design.
http://hadoop.apache.org/common/docs/r0.18.0/hdfs_design.pdf

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 2, May 2014

International Conference On Advances in Computer & Communication Engineering (ACCE - 2014)
on 21st & 22nd April 2014, Organized by
Department of CSE & ISE, Vemana Institute of Technology, Bengaluru, India

10. “The Hadoop Distributed File System: Architecture and Design” by DhrubaBorthakur,
11. Selic, B. (2004) Fault tolerance techniques for distributed systems. *IBM*.
http://www.ibm.com/developerworks/rational/library/114.htm