International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.5., 2015
(Sept.-Oct.)

RESEARCH ARTICLE

ISSN: 2321-7758

A COMPARATIVE ANALYSIS: HIGH AVAILABILITY OF NAMENODE ARCHITECTURE AND REPLICA RECONSTRUCTION

CHETALI PARVE, ABHILASHA SINGH, PRIYANKA TRIPATHI
Department of Computer Technology and Application, NITTTR Bhopal

CHETALI PARVE

ABHILASHA SINGH

PRIYANKA TRIPATHI

ABSTRACT
Hadoop Distributed file system, which manages large amount of data across multiple low cost commodity machines, is an emerging area of research which deals with the big data. HDFS architecture consists of one namenode and multiple datanode and provides features like availability, scalability, reliability, etc. by holding replicas of the blocks. Due to system maintenance or failure, a datanode can be removed from the HDFS cluster and the data blocks that the removed datanode held are lost. If data blocks from the deleted datanode are missing, the accessing load on other datanodes increases and due to this the performance of data processing decreases. Therefore, replica reconstruction becomes an important issue for reallocating the missing data blocks in order to prevent the performance of data processing over distributed file system. So we have considered an existing HDFS architecture where nodes are arranged in a one directional ring structure and data blocks are transferred based on this ring structure so that the difference of the amount of transfer data of each node is minimized. This is done with the help of heuristic scheme that is very effective for replica reconstruction. By replica reconstruction we can reallocate the missing blocks and can also balance the workload on the other datanode and prevent the loss of data blocks in case of deletion of datanode. Namenode is a single point of failure (SPOF) in HDFS cluster. But in case, if the Namenode in HDFS cluster fails, the whole system should be started manually, making the systems on the cluster less available. This paper proposes a highly available architecture in case of both Namenode and datanode failure. And it's working principle will be based on heuristic scheme and Election by bully algorithm for achieving high availability namenode architecture.

Keywords— HDFS, Replica reconstruction, namenode, SPOF, bully algorithm.

INTRODUCTION

Hadoop Distributed File System (HDFS)[1] is a distributed file system that is developed to deploy on low cost commodity hardware to store very large datasets reliably and has high degree of fault tolerance and also stream those data sets at high bandwidth for user applications to run on commodity hardware. It is a master/ slave architecture where master is called namenode and slaves are called datanode. HDFS is has many similarities with existing distributed file system, but they are very different. A distributed file system consists of single namenode and multiple data nodes and provides availability and reliability by holding multiple replicas of data. The availability feature compromises when due to failure or maintenance a node fails, a node can be a datanode or it can be a namenode. Figure 1 shows

the HDFS architecture representing nodes/components.

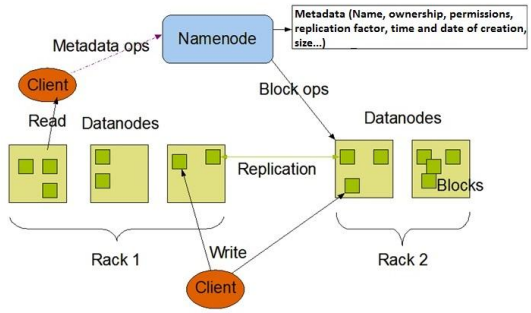

Figure 1: HDFS Architecture

Namenode: HDFS works in a master/slave fashion and namenode is master node. All the metadata that is related to HDFS including the information about datanodes, Replication, and files stored on HDFS, etc. are maintained and stored by the Namenode. A Namenode acts as a master and there is only one Namenode in a cluster.

DataNode: Datanode is the slave node and holds the data of the client in the form of Data Blocks. There can be any number of Datanodes in a Hadoop Cluster.

Data Block: A data/files stored on HDFS is divided into smaller parts which are called Data Block. By default each incoming file is broken into 64 MB of blocks. Any file which is larger than 64 MB of size is broken down into 64 MB blocks. All the data blocks which make a particular file are of the same size i.e. 64 MB except the last block which can be less than 64 MB depending on the size of file.

Rack Awarenes: Data is replicated across different datanodes in the HDFS cluster to ensure fault tolerance /reliability. Based on the location of the datanodes, replication of data blocks is done, to ensure the high degree of fault tolerance. In rack awareness policy, one or two copies of data blocks can be stored on the same rack, one copy should be stored on a different rack within the same data center, one more block on a rack in a different data center, and so on.

Replication: In a distributed file system, data is divided into data blocks and are replicated, data blocks including their replicas, are stored separately across different datanodes for reliability and availability and also ensures a high degree of fault tolerance. The number of replicas which is made/maintained is configurable on the cluster and also for each file. Based on the Replication Factor (by default replication factor is 3), each data is replicated several times on different datanodes in the HDFS cluster.

Data Replication

Data is replicated on different datanodes so as to ensure reliability and fault-tolerance.

- Replication Factor is configurable at a file level and also at a cluster level (Default is 3).
- The data replication is important for various scenarios like the datanode go down, change in Replication Factor, and when the Data Blocks corrupts, etc.
- When the replication factor is increased (at cluster level or for a file), Namenode identifies the datanodes to which the data blocks need to be replicated and then it initiates the replication.
- When the replication factor is decreased (at cluster level or for a file), Namenode identifies the datanodes from which the data blocks need to be deleted and then it initiates the deletion.

Problems Perceived In This Area

Many researchers focus on the namenode failure and datanode failure and many techniques had being implemented to prevent these failures so that high availability of HDFS cluster is maintained. A lot of replication techniques and strategies have been proposed for replica management/reconstruction and for single point of failure of the namenode. And it is important to decide the replication factor properly and also where to store replicas.

Replica reconstruction problems

For achieving high availability and scalability of HDFS architecture, a distributed file system consists of multiple datanode, each can depend on different system requirements, and each datanode manages the blocks of the data and their individual replicas. However, it is difficult to operate all of these datanode without any failures. Some datanode may be unstable due to system failure or maintenance.

When a datanode failure has been detected, the data blocks stored in the datanode are lost and the access load of other datanode, which hold the lost data blocks increases, so that the performance of

the data processing decreases over the distributed file system. Therefore, an important issue is effective replica reconstruction technique that can reallocate the missing data blocks to other stable datanode in order to prevent such performance degradation.

The Hadoop Distributed File System (HDFS) is widely used distributed file system. In HDFS the replica reconstruction process, selects randomly the source and the destination datanode for replication. Through analysis it was found that this default replica reconstruction scheme is inefficient because of data transfer is biased. Many techniques are used by the researchers to solve the problems of replica placement/maintenance. To solve this problem of replica reconstruction an effective replica reconstruction scheme or techniques is required which will aims to balance the workloads of replication processes in case of datanode deletion.

Single Namenode problem

HDFS cluster consist of only one Namenode and among all the nodes/components, the Namenode is the most vulnerable part in the whole system. If one or more datanode fails, it may have a very small performance impact on the users, however, the crash or failure of the single Namenode machine will bring down the whole HDFS cluster as we know that all traffic goes through the Namenode. SPOF [2] of Namenode machine can affect the overall availability of Hadoop. When Namenode goes down the entire system become unavailable and cannot perform any operation until Namenode gets restart manually.

This is an infrastructure issue, which can create a problem of namenode single point of failure (SPOF). Many improvements have been done to solve this issue of SPOF, since the first cluster was implementation, providing with the checkpoint namenode and secondary namenode (backup). All these improvements aim to increase the overall reliability and availability of the system, as theses components work in an active or in a standby way. As per performance, no improvement is provided, since all the clients contact with this single namenode on the filesystem for their operations. In current versions, HDFS federation has provided the ability of multiple namespaces on the same cluster. Each namespace in cluster has its own Namenode. With these approaches, for common storage for blocks, datanodes are used and multiple Namenodes can work in the cluster without any synchronization between each other. This can aim in reducing the load of the single Namenode and splits the namespace in multiple parts, but the clients working concurrently on the same namespace still sends requests to the same Namenode, causing overloading.

Literature review

A. Replication Strategies:

There are some techniques which are used for the replicate the data in multiple nodes which are discussed in detail.

A Dynamic File Assignment Strategy with Replication

In this paper J. Tjioe, R. Widjaja et al. [3] proposed a replication strategy based on a dynamic file assignment according to access load. First, they assign files, which are sorted according to file size, in a round-robin fashion to the disks to distribute the load of all files evenly across all disks. Then, creation and deletion of replicas are occurred according to the load of all files and the load on each disk. From the experiments, load balancing can be achieved in an environment where user access patterns change significantly.

Study of Different Replica Placement and Maintenance Strategies in Data Grid

In this study Rashedur M.Rahman et al. [4] proposed a dynamic replica maintenance algorithm that can re-allocates to the new candidate sites if there is a performance metric degradation significantly over last K time periods. To solve this problem of replica placement, a multi-objective model is used. This multi-objective model considers the objectives of p-center and p-median models simultaneously that will host the replicas to select the candidate sites.

Load balancing using grid-based peer-to-peer parallel I/O

In this paper Y. Wang et al. [5] propose a novel Peer-to-Peer (P2P) storage architecture for MPI applications on Grid systems. Firstly they presented an analytic model of P2P storage architecture. Next, they described a profile-guided data allocation algorithm that can increase the degree of I/O

parallelism present in the system, as well as to balance I/O in a heterogeneous system.

A Study of Effective Replica Reconstruction Schemes at Node Deletion for HDFS

In this paper authors Asami Higai et al. [6] of this paper worked on the HDFS while dealing with large dataset and multiple node architecture where they work on balancing the workloads of replication processes. They worked on replication scheduling strategy assuming that the nodes are arranged in a ring topology and data blocks are transferred based on this one-directional ring structure to minimize the amount of difference in transferring the data of each datanode. Based on this strategy, they proposed two replica reconstruction schemes, an optimization scheme and a heuristic scheme. Also they implemented the proposed schemes in HDFS and evaluated them on an actual HDFS cluster. From the experiments, the replica reconstruction throughput of the proposed schemes showed 45% improvement as compared to the default scheme. They have worked on effective replica reconstruction schemes, whose aim is to balance the workload of copying processes between the source and destination datanodes. The scheme involved by the researchers is optimization and heuristic, that aims to minimize the difference of the amount of transfer data of each datanode, and select a source data node, which holds a missing data block. They have worked and performed their output result parameter on throughput and computation time.

The Model of Data Replica Adjust to the Need Based on HDFS Cluster

In this paper Guangbin Bao et al. [7] The author of this paper presented a technique to work in distributed file system in which HDFS cloud storage cluster and supported by the management technology of data replica and the technique was evaluated using throughput which derive the data distribution efficiently. According to their proposed work they have worked on the energy model which is sensitive to the size of restored file data and the number which cluster can use to build Map tasks. Using this concept to describe the file data will bring about system load for cluster system when the file data are written into the HDFS cluster. Thus the file data should be divided into what size of the data block. They have discussed and work on the factor which affects the system.

They have discussed on various model for the file distribution such as the price-driven self adaptive model, Single-node load state model, Cluster system reliability model. Basically they have introduced the technique for the data replication and further can be worked on the cloud computing for working with large data.

B. High availability of HDFS

A Distributed Namenode Cluster for a Highly Available Hadoop Distributed File System

Yonghwan KIM et al. [8] proposed a new HDFS architecture consisting of several Namenodes to resolve Problems like various name node failure problem, name node limitation to handle the data, and load balancing problem. They had resolved the SPOF problem using multiple namenodes instead of one. When one Namenodes fail, the other Namenodes take their roles immediately. They had extended the maximum size of the namespace by installing additional Namenodes. They had also proposed a system that can be constructed by only commodity hardwares as the original HDFS and needs no special hardware.

NCluster: Using Multiple Active Namenodes to Achieve High Availability for HDFS

Zhanye Wang et al. [9] presented a solution to achieve high availability for HDFS's namenode through efficient metadata replication. They have utilized multiple active namenodes, to build a cluster that serves the metadata request simultaneously. They have also implemented a pub/sub system that can efficiently handle the metadata replication process across the active namenodes. For the solution they have implemented a prototype called NCluster and integrated it with HDFS. They have also evaluated NCluster to exhibit its feasibility and effectiveness. The results showed low replication cost, good throughput and scalability.

Towards A Scalable HDFS Architecture

As we know that metadata of datanode is restricted by the capacity of the RAM of the HDFS's single-point-of-failure Namenode. So Farag Azzedin et al. [9] proposed a highly available, fault tolerant, and widely scalable HDFS architecture in which Namenode is distributed as it is distributed it will not

suffer single Namenode failure in HDFS. For achieving this they have used the Chord protocol and integrated it with Namenode. This architecture has highly improved the scalability and availability of HDFS architecture also including its single-point-of-failure.

III Comparison analysis

No	Author	Paper Title	Methodology	Further Work Can be done
1.	Asami Higai, Atsuko akefusa, Hidemoto Nakada, Masato Oguchi	A Study of Effective replica Reconstruction Schemes at Node Deletion for HDFS	They have worked on HDFC system which is a clone of GFS Google file system and the work is done on heuristic approach on File distribution system.	Further work can be done on working on the condition while namenode get down and still we require to optimize the system such that failure of NN can be optimized.
2.	Jonathan Tjioe, Renata Widjaja, Abraham Lee, and Tao Xie	DORA: A Dynamic File Assignment Strategy with Replication	A Dynamic round robin scheme for file redistribution. They have worked on the disk temperature scheme for load balancing mechanism.	According to the research author DORA can be extended to write-dominant workloads.
3.	Guangbin Bao,Chaojia Yu, Hong Zhao, Hong Zhao	The Model of Data Replica Adjust to the Need Based on HDFS Cluster	They Worked on the energy model and various replication techniques were introduced in their working methodology.	Furthermore their technology can be integrate and utilize with the cloud computing application.
4.	Yonghwan kim, tadashi araragi, junya nakamura and toshimitsu masuzawa	A distributed namenode cluster for a highly-available hadoop distributed file system	They have work on the SPOF problem where the single namenode participate in the distributed system, they introduce the participation of one more namenode which can sync the data with the existing Namenode and can work in the failure condition of primary namenode.	

Problem formulation

Many researchers have focused on the namenode and datanodes failure and many techniques had being implemented to prevent these failures so that high availability of HDFS cluster is maintained. A lot of replication techniques and strategies have been proposed for replica management and placement. In general, data are replicated with replication factor 3 and their replicas are stored on different data nodes

for high availability and reliability. It is important to decide the replication factor properly and also where to store replicas. And it is also important to find the solution for namenode single point of failure to make the systems highly available in HDFS cluster when namenode goes down.

In this paper we have shown the solution for single point of failure (SPOF) of the namenode and the solution for effective replica reconstruction to balance the workload of the replication processes effectively by using replica reconstruction scheme. We have used one of the effective replica reconstruction scheme from the existing work i.e. heuristic scheme that was showing better performance than the default scheme and applied it in the HDFS. According to this the datanode are arranged in ring and transfer of blocks are done in one directional ring structure to balance the workload of the replica reconstruction process when any datanode fails. But as we know that namenode is the single point of failure in the HDFS cluster and if it fails, namenode needs to be started manually making the systems in the cluster less available. This paper proposes a fault tolerance and highly available HDFS architecture, eliminating the drawbacks of the current architecture.

We have thoroughly analyzed and studied the HDFS architecture with the main focus on its Namenode SPOF problem.

Proposed work

In this work replication scheduling strategy assumes that nodes are arranged in a ring and data blocks are transferred on the basis of this one-directional ring topology to minimize the difference of the amount of transfer data of each node. Based on this strategy, there are two existing replica reconstruction schemes, an optimization scheme and a heuristic scheme. As per existing techniques the replica reconstruction throughput of the heuristic schemes showed a good improvement as compared to that of the default scheme, and the load of each datanode can be balanced by eliminating the bias of data transfer. Therefore, as the heuristic scheme was very effective in replica reconstruction at datanode deletion, so we have used this scheme in our proposed work so that we can balance the workload of replica reconstruction when datanode is deleted from the cluster.

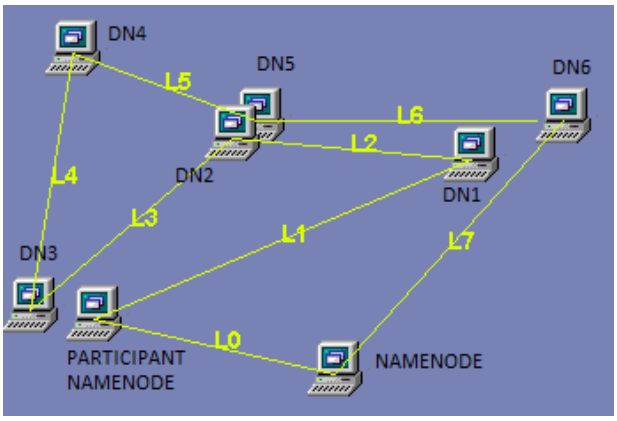

Figure 2: HDFS architecture in ring topology

Figure 2: demonstrates the HDFS architecture where the multiple nodes are arranged in one directional ring network where an efficient technique for workload is required to make process computation time efficient and faster.

We are working on the existing architecture where performance of data processing is increased by using effective replica reconstruction scheme to balance the workload of copying processes in case of datanode failure. We are building a HDFS architecture consisting of two namenodes (coordinator and participant namenode) and six datanodes that are arranged in one directional ring topology. Its working principle will be based on bully algorithm (election algorithm) and data synchronization between the two namenodes. .

To eliminate this problem of HDFS Namenode, we have proposed architecture of HDFS inspired by the existing techniques and its working principle. This HDFS architecture and working principle eliminates Namenode SPOF problem by replicating namenode initially on separate machine only once and then it starts working normally with the Datanode. By this both the Namenodes will be fully updated all time. The idea behind this architecture is that once namenode is replicated after that, it will perform all the operations on its metadata by using data synchronisation. Initially the namenode will act as a Coordinator Namenode, and its replica will be a Participant Namenode machine. Data synchronization between the Coordinator and Participant namenode will perform the updation to their metadata. Due to data synchronization both the

namenodes (i.e. coordinator and participant) will be updated all time, and hence if in the case, the Coordinator namenode goes down or fails completely then the participant namenode becomes the Coordinator and hosts the datanode. Then the new Coordinator from the datanode can be elected using Election by Bully algorithm [11] and the data that the elected datanode holds is distributed among the remaining datanode for reliability and fault tolerance. Therefore, this paper proposes a highly available architecture and its working principle for the HDFS Namenode against its SPOF utilizing Election algorithm and heuristic scheme.

Conclusion and future work

In this paper we have discussed about various replication strategies and the availability problems in HDFS cluster. We have also discussed about the heuristic technique used in recent research paper for HDFS (Hadoop distributed file system). We have taken the idea of replica reconstruction HDFS architecture from the current techniques and analyzed the schemes and techniques proposed by them. The Namenode in HDFS is the single point of failure (SPOF) and if the namenode is down due to any hardware or software failure, the system becomes unavailable until it is restarted manually. To avoid this problem of single point of failure (SPOF) of Namenode, we proposed a highly available HDFS Namenode architecture, where the namenodes and datanodes are arranged in one directional ring structures to balance the access load on each datanodes in case of failure. To eliminate the SPOF problem our working principle will be based on data synchronization and election by bully algorithm [11].

Out of this discussion we can further work on the current techniques and extend the current work, so that single point of failure (SPOF) of name node will get eliminated as well as workload of the replica reconstruction process is also balanced.

In future, we will be deploying this HDFS architecture on Hadoop and will be ensuring its applicability. There is also another SPOF in Job Tracker that is the hadoop programming model which is used for tracking MapReduce tasks. In future, we will extend our proposed work to address the Job Tracker SPOF problem.

REFERENCES

[1]. Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler. “The Hadoop Distributed File System,” 26th Symposium on Mass Storage Systems and Technologies (MSST), pp. 1-10, May 2010.

[2]. Hadoop Wiki, “Namenode Failover”, on Wiki Apache Hadoop, http://wiki.apache.org/hadoop/NamenodeFailover, July, 2012.

[3]. Jonathan Tjioe, Renata Widjaja, Abraham Lee, and Tao Xie,” DORA: A Dynamic File Assignment Strategy with Replication”, 2009 International Conference on Parallel Processing.

[4]. Rashedur M. Rahman Ken Barker Reda Alhajj;” Study of Different Replica Placement and Maintenance Strategies in Data Grid” Seventh IEEE International Symposium on Cluster Computing and the Grid(CCGrid'07).

[5]. Yijian Wang and David Kaeli, “Load Balancing using Grid-based Peer-to-Peer Parallel I/O”.

[6]. Asami Higai, Atsuko Takefusa, Hidemoto Nakada, Masato Oguchi: A Study of Effective Replica Reconstruction Schemes at Node Deletion for HDFS, 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing.

[7]. Guangbin Bao,Chaojia Yu, Hong Zhao, Hong Zhao; “The Model of Data Replica Adjust to the Need Based on HDFS Cluster”, 2012 Fifth International Conference on Business Intelligence and Financial Engineering.

[8]. [8] Yonghwan KIM, Tadashi ARARAGI, Junya NAKAMURA and Toshimitsu MASUZAWA;” A Distributed Namenode Cluster for a Highly-Available Hadoop Distributed File System”, 2014 IEEE 33rd International Symposium on Reliable Distributed Systems.

[9]. Zhanye Wang and Dongsheng Wang;” NCluster: Using Multiple Active Namenodes to Achieve High Availability for HDFS,” 2013 IEEE International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing.

[10]. Farag Azzedin;" Towards A Scalable HDFS Architecture", 2013 IEEE International Conference.

[11]. Andrew S.Tanenbaum and Maarten Van Steen, "Distributed Systems: Principle and Paradigms", Pearson Prentice Hall, Upper Saddle River, NJ 07458.