ABSTRACT

The enormous amount of information is being used in regular activities hence it is important to store and analyze such data. This gives the figment that this data is arranged locally on the client's PC. As a general rule, the Internet speaks of a gigantic distributed system that appears as a single resource to the user. A distributed system is an accumulation of autonomous computers connected by communication channels that appears to its clients as a single coherent system. Distributed systems should be fault tolerant, scalable and highly available.

Hadoop is a popular framework with a distributed file system used to store data. It provides fast, reliable and transparent access of data to a large number of heterogeneous and geographically distributed users. The major challenges in large scale distributed systems include availability, scalability, reliability, caching, openness and transparency. This thesis proposes approaches to enhance availability, reliability and caching in a distributed system.

Reliability is the probability of a system functioning correctly over a given period of time. In a good distributed file system, the probability of loss of stored data should be minimized. Caching improves data access speed by storing the recently used information in main memory.

The main goal of this research work is to improve the performance of the Hadoop Distributed File System. The objectives of the research work include minimizing the loss of stored data using erasure codes, improving the data access speed by maintaining the recently accessed information in an associative cache with Least Unified Value eviction policy and enhancing the

availability of NameNode using a Improved Election Algorithm and maintaining metadata in multiple locations.

This research proposes an approach to improve reliability of data on the Hadoop file system through Tornado, Turbo and Reed Solomon Codes. In Hadoop, data are split into blocks that are distributed in the cluster. To maintain reliability, blocks are replicated. NameNode maintains metadata about the size and location of blocks and their replicas. Erasure codes create checksum blocks from data blocks that make up the file.

The existing file system uses Low Density Parity-Check (LDPC) codes for detection and correction of errors. LDPC error correction is restricted to the minimum number of error bits. This research proposes the usage of Tornado codes to detect and correct multiple bit errors in a storage system. Tornado codes are simple to implement using XOR operation. Turbo codes are more accurate and fast for encoding/decoding and for correcting multiple errors. Therefore, Turbo codes are highly desirable for storage and retrieval of data for large clusters. Reed Solomon codes are block-based error correcting codes consisting of two parts, namely data part and parity part. The encoding process adds the parity symbols to message block. Reed Solomon codes can be described as an $\left(s_1', p_1'\right)$ code, where s_1' is the block length in symbols and p_1' is the number of information symbols in the message. There are $\left(s_1' - p_1'\right)$ parity symbols which are calculated based on the data part. This research compares the performance of LDPC with the proposed approach, namely Tornado, Turbo and Reed-Solomon codes for block storage in HDFS. Encoding speed of Reed Solomon codes is faster by 10%, 15% and 30% when compared to that of Tornado, Turbo and LDPC codes respectively.

The second objective of the research work is to improve access speed of information using Associative caching with Least Unified Value (LUV) replacement policy. It works based on the object value which is calculated from past reference history of the data item. The proposed replacement policy LUV on associative caching reduces file access time from 10% to 15% for a single node cluster and 15% to 30% for a multi node cluster.

The third objective of the research work is to improve availability of NameNode using an Improved Election Algorithm. The Improved Election Algorithm elects a new NameNode when the primary NameNode goes down. The election procedure is time efficient and involves less message passing between nodes. The speed of read/write operations increases from 6 to 8 times by using Namespace Partition Table (NPT) in NameNodes when compared to Local File System (LFS).

The proposed reliability, availability and caching mechanism have been tested on a Digital Library Service (DLS) application. Experimental results show that the proposed algorithms improve the overall quality of service on the Hadoop Distributed File System.

To summarize (i) Encoding speed of Reed Solomon codes is faster by 10%, 15% and 30% and decoding of Reed Solomon codes is faster by 10%, 15% and 40% when compared to that of Turbo, Tornado and LDPC respectively. (ii) The proposed LUV replacement policy on associative caching reduces file fetching time by 10% to 15% for a single node cluster and 15% to 30% for a multi node cluster and (iii) The performance of the read/write operations in improved election algorithm to elect a new coordinator (NameNode) when the primary NameNode fails, improves by 6 to 8 times as the Namespace Partition Table (NPT) is maintained in multiple NameNodes when compared to searching data in single NameNode.

Effective adaptive algorithms can be developed to adjust configuration of replication process including metadata buffer size. LUV replacement policy and erasure codes can be used in cloud to enable efficient and reliable data access. Erasure codes can be applied for cloud storage to maintain data in a reliable manner.