Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 50 (2015) 590 – 595

Procedia Computer Science

2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

A NOVEL APPROACH FOR REPLICA SYNCHRONIZATION IN HADOOP DISTRIBUTED FILE SYSTEMS

Miss.J.Vini[a], Rachel Nallathamby[b], C.R.Rene Robin[c]

**P.G scholar,*Research scholar,*Professor and Head of the department*
Department of Computerscience, Jerusalem college of engineering, Chennai

Abstract

The Map Reduce framework provides a scalable model for large scale data intensive computing and fault tolerance. In this paper, we propose an algorithm to improve the I/O performance of the Hadoop distributed file system. The results prove that the proposed algorithm show better I/O performance with comparatively less synchronization

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

Keywords: Big data, distributed file system, Map Reduce,Hadoop, Adaptive replica synchronization

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)
doi:10.1016/j.procs.2015.04.090

1. Introduction

To improve the performance and system scalability in the distributed file system we are using replica synchronization [1]. A replica is nothing but a copy of the original data in the file system. Replica synchronization needs to make sure that the changes in the data sent to one replica are received by other relevant replicas. In Replica synchronization, updates are made to all relevant replicas as many times as the number of write requests, so bottleneck increases.

In HDFS, we use the Meta Data Server (MDS) to manage the whole system and replica synchronization is triggered when any one of the replicas has been updated [2] and the Storage Server (SS) handles the data management.

We use MapReduce to map the input set and get the output set. The input and the output of the MapReduce job are stored in Hadoop Distributed File System (HDFS).

2. Related Work

General parallel file system (GPFS) [4] allocates the space for the multiple copies of data on the different storage server which supports the chunk replication and it writes the updates to all the location. GPFS keeps track of the file which been updated to the chunk replica to the primary storage server. Ceph [5] is the free storage platform has similar replica synchronization technique where the newly written data should be sent to all the replicas which are stored in different storage servers before responding to the client. In Hadoop File System [6] the large data are spitted into different chunks and it is replicated and stored on storage servers. In Google File System (GFS) [7], there are various chunk servers were the MDS manages the location and data layout. For the purpose of the reliability in the file system the chunk are replicated on multiple chunk servers; replica synchronization can be done in MDS. The Lustre file system [8], which is known for parallel file system, which has replication mechanism Mosa Store [9], uses dynamic replication for the data reliability. Here when one new data block is created, the block at one of the SSs is stored in the MosaStore client, and the MDS replicates the new block to the other SSs to avoid the bottleneck when the new data block is created. Replica synchronization is done in the MDS of MosaStore.

The Gfarm file system [10] the replication mechanism is used for data replication for the reliability and availability. In the distributed and parallel file system, the MDS controls the data replication and send the data to the storage servers; this makes pressure to the MDS. Data replication which has the benefits to support for better data access was the data is required and provide data consistency. In the parallel file system [11], this improves the I/O throughput, data duration and availability by data replication. The proposed mechanism, according to the cost of analysis the data pattern are analysed a data replication is done, but replication synchronization is done in the MDS.

In the PARTE file system, the metadata file parts can be replicated to the storage servers to improve the availability of metadata for high service [12]. In the PARTE file system, the metadata file parts can be distributed and replicated to the corresponding metadata into chunks on the storage servers, the file system in the client which keeps the some request of the metadata which have been sent to the server. If the active MDS crashed for any reason, then these client backup requests are used to do the work by the standby MDS to restore the metadata which was lost during the crash.

3. Proposed System Overview

Adaptive replica synchronization is used to improve the I/O throughput, communication bandwidth and performance of distributed file systems. The MDS manages the information in the distributed file system which splits the large data into chunks.

The main aim of using the adaptive replica synchronization is the storage server cannot withstand the large amount of the concurrent read requests to specific replica. The adaptive replica synchronization will be performed to satisfy heavy concurrent reads when the access frequency to the target replica is greater than the predefined