Paper ID: CSEIT05

DEVELOPING A NOVEL APPROACH FOR IMPROVING AVAILABILITY OF JOBTRACKER IN HADOOP

Prajakta Mane
Prajakta Koshti
Prajakta Patankar
Prof. Ravindra J. Mandale
Prof.Sagar S. Sawan
Rajarambapu Institute of Technology, Rajaramnager, Maharashtra, India

Abstract— Cluster based distributed system contains singles JobTracker and multiple TaskTracker. TaskTracker performs MapReduce operation on DataNode. In this approach, JobTracker facing single point of failure problem. The solution includes the AvatarNode and ZooKeeper. Firstly, AvatarNode uses efficient checkpoint strategy to manage temporary namespace. AvatarNode gives temporary space to store the metadata about current job and flush that data after successful job execution. Secondly, ZooKeeper which is used for session monitoring by communicating with failover manager that is residing in JobTracker and AvatarNode This approach reduces Remote Procedure Call (RPC) traffic between Backup Node and DataNode by direct communication of JobTracker and AvatarNode.

Index Terms — JobTracker, TaskTracker, AvatarNode, ZooKeeper, Hadoop

I. INTRODUCTION

Now a day, increasing large volume of data called Big-data is a problem for all organization. Fortunately, Hadoop framework is available in the market which is a solution to above problem and handles big data efficiently. Hadoop is Free and Open Source Framework (FOSF). Now a day, Hadoop is very popular and it is being used by big companies like Google, Yahoo Inc., Amazon, Twitter, Facebook, etc. [6]. Hadoop is mainly consisting of Hadoop distributed file system (HDFS) and MapReduce components.

HDFS is storage system for all the jobs submitted by client in cluster [5]. HDFS broken down into blocks and these blocks represent different nodes in cluster. HDFS file system is written in traditional hierarchical namespace [6].

HDFS is a distributed file system which provides high throughput access to application data[5]. DataNode consists of MapReduce operation [2]. MapReduce operation contains Map and Reduce, in which Map is responsible for splitting the data provided by DataNode and Reduce is responsible for combining the separated data.

Cluster based distributed file system consists of single JobTracker to handle job and multiple TaskTracker to performs task [6]. This follows Master-Slave architecture. This architecture is simple and efficient, but facing single point of failure problem in JobTracker. The solution includes the Backup Node. It implements efficient checkpoint strategy because it manage own namespace. Backup Node ensures to keep up to date view of namespace but in Backup Node comprises information related to block location and leases [6]

These two things are required to process the client request. For this purpose, Hot Standby node is introduced. This makes the NameNode to propagate any change on replicas to the Hot Standby node and also allow DataNode to send messages to Hot Standby node [6]. Messages are sent in batch manner but there is a problem that it requires change to a great extent to NameNode code and message exchange increases the traffic between NameNode and Hot Standby node [6]. We took the advantage of this fact for developing monitoring framework. This framework is easy and efficient to improve the availability of JobTracker in Hadoop. This framework includes monitoring activity with two important components: AvatarNode and ZooKeeper. AvatarNode is used to keep the replicas of current job files and ZooKeeper is used for session monitoring.

The remaining part of this paper is structured is as follows: Section II discusses various frameworks proposed by different researchers. Section III describes the algorithm and working of our scheme. Section IV presents analysis of existing systems and proposed system. The conclusion and future work is given in last section of this paper.

II. LITERATURE SURVEY

This section describes about work done for making JobTracker and TaskTracker available all the times in Hadoop. Also, this survey gives details of how JobTracker facing single point failure in Hadoop. Amrit Pal and Sanjay Aggarwal have proposed [1] experimental work done on Hadoop cluster. The time required for retrieving and storing data on cluster is decreased by adding nodes in cluster one by one at each step. This method reduces time of retrieving and storing data. The recorded time for storage or retrieval of information is in three forms are real-time, user time and system time. When teragenerate program is generated for storage of data on cluster in real time also increased as number of nodes is increased in each step, where as real time for sorting data is reduced as we increase the number of nodes at each step. As the number of nodes in cluster increases, the amount of time spent by CPU in user mode is also increases partly. There are some variations as well. This variation can be due to network traffic or due to concurrent process running on node. Jisha S Manjaly and Varghese S Chooralil have introduced an algorithm [2] which will not allow a task to run and fail if the load of the TaskTracker reaches to its threshold for the job. In Hadoop, it usually uses FIFO scheduler that allows jobs to utilize

entire cluster capacity. FAIR scheduler and CAPACITY scheduler came with Hadoop. FAIR scheduler is introduced by Facebook and CAPACITY scheduler is introduced by Yahoo. FAIR scheduler aims to give every user a fair share. This system control number of tasks for a particular job in TaskTracker. Also this algorithm supports all the functionalities provided by the Fair Scheduler. This scheduler enables users to configure a maximum load per TaskTracker in the job configuration itself. Jinshuang Yan et al. have proposed [3] optimization methods to improve the execution performance of MapReduce jobs. There are three major optimizations: first, reduce the time cost during initialization and termination stages of a job by optimizing its setup and cleanup task. Second, replace pull-model task assignment mechanism with push-model. Third, replace the heartbeat based communication with instant message communication for event notification between JobTracker and TaskTracker. The performance of improved version of Hadoop is about 23% faster on average than standard Hadoop. Hadoop MapReduce is a successful model and framework for large scale data processing. When a MapReduce job is submitted into Hadoop, the input data of the job would be split into several independent data splits with equal sizes with each map task processing one data split. These map tasks run in parallel and their outputs would be sorted by the framework and then fetched by reduce tasks for further processing. Shyam Deshmukh et al. have attempted [4] to create a scheduler that can learn and adapt itself to any possible application. The system is able to adapt heterogeneous workloads. Heterogeneous workloads are encountered in the real world. A node is overloaded if the ratio crossed a user specified limit. If CPU utilization by job crosses 80% of available CPU capacity then node is overloaded. One machine act as NameNode (JobTracker) and other machines are DataNode (TaskTracker).The Ganglia Monitoring Daemon (gmond) needs to be installed on each slave Hadoop nodes. Ganglia collect metrics, such as CPU and memory usage. Jian Wan et al. has designed [5] solution to resolve the single point of failure of the JobTracker and then enhance its availability. A standby JobTracker is a hot Backup Node of the active JobTracker. The standby JobTracker synchronizes the job execution process with the active JobTracker. The original implementation of the MapReduceframe-work in Hadoop designed with a single JobTracker. Standby JobTracker synchronizes jobs based on the job log obtained from the active JobTracker. The standby JobTracker has a slight delay compared with the active JobTracker because the standby JobTracker needs to recover the rest of the jobs in the failover process. Standby JobTracker monitors the status of the TaskTracker. Andre Oriani and Islene C. Garcia have proposed [6] an algorithm to increase availability of HDFS. Availability has increased by Backup Node and replication has performed by check pointer helper. ZooKeeper is distributed coordination service used for achieving automatic failover. DataNode is sending heartbeat, block report, block received messages to NameNode. This message is used by NameNode to know location of all blocks in HDFS and available replicas of each block. NameNode is performing namespace management, replica management, controlled access to file system, block DataNode mapping. Replication burden is shared among the other elements of HDFS. When active node get fail, Backup Node gets the log information and check pointer from NameNode, further log entries are stored in Backup Node. Weiyue Xu and Ying Lu have proposed [7] a system for energy analysis of cluster failure recovery. For high performance replication is needed that causes more energy consumption. This system focuses on failure recovery using less energy. It maintains set of active nodes for high and immediate availability. Data Redundancy mechanism is achieved by two ways 1) Replication 2) Eraser coding. And it is used for fault tolerance. Replication factor should be define while file creation. Zhanye Wang and DongSheng Wang have proposed [8] an algorithm for improving availability HDFS by using active NameNodes. Pub/sub system is used to improve and handle metadata replication. At a time only one NameNode is active and other NameNode act as Backup Node. Active NameNode performs updation of metadata on local disk drive and then send it to Backup Nodes. NCcluster prototype is used and integrated with HDFS to improve availability of HDFS. This system is advantageous for low replication cost, good throughput and scalability. HDFS clients are communicating with NameNode for accessing the files in HDFS. Inactive NameNode cannot serve the client's request so there is workload on active NameNode. There is large overhead of I/O because of replica synchronization. Qin Zheng has proposed [9] a system for improving MapReduce fault tolerance in the cloud. Redandant copies are investigated for this. This system is targeted at Amazon elastic MapReduce and providing data intensive task for their application. This system is used for improving MapReduce fault tolerance performance and reducing latency. Fault tolerance in cloud is not expensive and reduces the overall cost for fault tolerance. It is preventing revenue loss for users and providers. This approach helps to recover failed tasks faster and reduces their completion time. Tsozen Yeh and Huichen Lee have proposed [10] an algorithm on enhancing availability and reliability of cloud data through syncopy. Syncope scheme is introduced in HDFS to achieve automatic real time synchronization. This synchronization is required for data files which are duplicated among different hadoop cluster. Syncopy reduces required time to achieving availability, reliability by 99.20%. Syncope having three issues such as collect location information, store location information, when to use this information to precede file duplication. Instead of whole file copy syncope only transfers new added data. A lot of time and network bandwidth is required for big files.
By reading and analyzing above papers, we have decided to take this issue of failure of JobTracker in Hadoop.

III. PROPOSED SYSTEM

A. Algorithm

1) Client submit job to JobTracker.
2) JobTracker determines location of data in NameNode.
3) JobTracker assigns the job to selected TaskTracker.
4) TaskTracker nodes are monitored by JobTracker. TaskTracker performs MapReduce operation on DataNode.
5) At a same time, NameNode managing file system related metadata.
6) AvatarNode creates replicas of metadata of current job.

7) If TaskTracker do not submit heartbeat signal to JobTracker, then they are said to be failed.
8) In JobTracker, job get failed is detected by ZooKeeper through Failover Manager (FM) and also detect which point failure occurred.
9) ZooKeeper contact with AvatarNode. AvatarNode sends metadata about current job to JobTracker.
10) JobTracker checks the availability of TaskTracker from NameNode and assign the job to TaskTracker.
11) If job get done completely, JobTracker update its status.
12) JobTracker send job status to ZooKeeper.
13) ZooKeeper sends job status to AvatarNode and AvatarNode flush metadata about current job.
14) JobTracker sends response back to client

B. Block Diagram

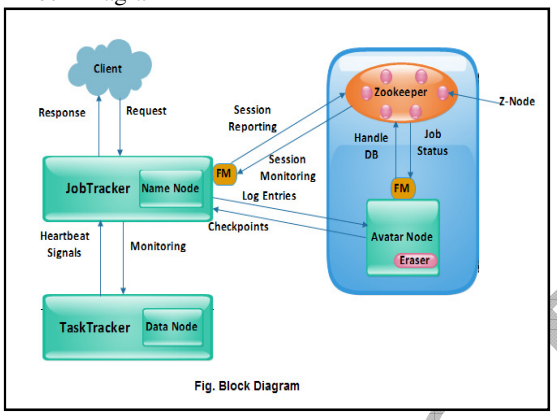

Fig. Block Diagram

C. Working

In this paper algorithm provides easy and efficient approach to increase the availability of JobTracker in Hadoop. Monitoring framework include AvatarNode and ZooKeeper. AvatarNode used to keep the replicas of current job files .ZooKeeper used for session monitoring. This architecture consists of JobTracker, TaskTracker, DataNode, ZooKeeper, AvatarNode, failover manager, NameNode. A NameNode contains serialized form of namespace and it provides as checkpoint also it manages transactional logs. TaskTracker nodes contains DataNodes are monitored by JobTracker. Client assigns the job to JobTracker which contains the NameNode. NameNode are used for managing namespace, controlling access and replica management. JobTracker determines the location of data in NameNode. From that information it decides corresponding TaskTracker and assigns the job to that TaskTracker which are responsible for storing blocks into local file system. DataNode sends their status to NameNode through heartbeat signals. This heartbeat signal tells NameNode that node is still alive and also informs load and free space. ZooKeeper contains Z-node which is used for session monitoring and it determines the job status and also detects the point where failure occurred. When NameNode crashes due to single point of failure then it inform these status to AvatarNode ZooKeeper communicates with AvatarNode and ZooKeeper using failover manager which is used for recovery from failover. AvatarNode create replicas of metadata of current job using primary NameNode. In this, AvatarNode only create temporary replica of current job and keep that replica until job is successfully executed. It communicates with ZooKeeper using failover manager and receives job status from ZooKeeper. If job get fails, AvatarNode send metadata about current job to JobTracker and JobTracker start the execution of job. After job gets successfully executed then AvatarNode flush the temporary stored metadata about job

IV. ANALYSIS

Existing system consist of Hot Standby Node, Backup Node implements efficient checkpoint strategy because it manage its own namespace. Backup node ensures to keep up to date view of namespace but in Backup Node comprises information related to block location and leases. For that hot standby server node is introduced. That makes the NameNode propagate any change on replicas to the hot standby and also make DataNode send message to hot standby server. Message is send in batch manner but there are still another problem that is it requires change to a great extent to NameNode code and message exchange increase the traffic between NameNode and Hot Standby Node.

The proposed system consists of monitoring framework. Monitoring framework contains AvatarNode and ZooKeeper. AvatarNode used to keep the replicas of current job files. ZooKeeper used for session monitoring. This approach reduces RPC traffic between Backup Node and DataNode by direct communication of JobTracker and AvatarNode.

V. CONCLUSION AND FUTURE WORK

The availability of JobTracker is becoming big issue in Hadoop as it is being widely used in real time system. We propose framework which exploits the availability of JobTracker in Hadoop. Thus reduces the chances of node failure in Hadoop cluster. This paper introduces monitoring framework that include AvatarNode and ZooKeeper. The above solution provides high availability of JobTracker in distributed environment of Hadoop.

REFERENCES

[1] Pal, A.; Agrawal, S., "A Time Based Analysis of Data Processing on Hadoop Cluster," in Computational Intelligence and Communication Networks (CICN), 2014 International Conference on , vol., no., pp.608-612, 14-16 Nov. 2014.

[2] Manjaly, J.S.; Chooralil, V.S., "TaskTracker Aware Scheduling for Hadoop MapReduce," in Advances in Computing and Communications (ICACC), 2013 Third International Conference on , vol., no., pp.278-281, 29-31 Aug. 2013.

[3] Jinshuang Yan; XiaoLiang Yang; Rong Gu; Chunfeng Yuan; Yihua Huang, "Performance Optimization for Short MapReduce Job Execution in Hadoop," in Cloud and Green Computing (CGC), 2012 Second International Conference on , vol., no., pp.688-694, 1-3 Nov. 2012

[4] Deshmukh, S.; Aghav, J.V.; Chakravarthy, R., "Job Classification for MapReduce Scheduler in Heterogeneous Environment," in Cloud & Ubiquitous Computing & Emerging Technologies (CUBE), 2013 International Conference on , vol., no., pp.26-29, 15-16 Nov. 2013

[5] Jian Wan; Minggang Liu; Xixiang Hu; Zujie Ren; Jilin Zhang; Weisong Shi; Wei Wu, "Dual-JT: Toward the high availability of JobTracker in Hadoop," in Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th

International Conference on , vol., no., pp.263-268, 3-6 Dec. 2012

[6] Oriani, A.; Garcia, I.C., "From Backup to Hot Standby: High Availability for HDFS," in Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on , vol., no., pp.131-140, 8-11 Oct. 2012.

[7] Weiyue Xu; Ying Lu, "Energy Analysis of Hadoop Cluster Failure Recovery," in Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2013 International Conference on , vol., no., pp.141-146, 16-18 Dec. 2013.

[8] Zhanye Wang; Dongsheng Wang, "NCluster: Using Multiple Active NameNodes to Achieve High Availability for HDFS," in High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on , vol., no., pp.2291-2297, 13-15 Nov. 2013.

9] Qin Zheng, "Improving MapReduce fault tolerance in the cloud," in Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on , vol., no., pp.1-6, 19-23 April 2010.

[10] Tsozen Yeh; Huichen Lee, "Enhancing Availability and Reliability of Cloud Data through Syncopy," in Internet of Things (iThings), IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing(CPSCom), vol., no., pp.125-131, 1-3 Sept. 2014.