Hadoop Framework to Provide Fault Tolerance in the Cluster

Santoshi Kalyani Balasubramanian[1]
Department of Electrical and Computer Engineering
University of New Haven, West Haven,CT
Sbala12@unh.newhaven.edu[1]

Professor Amir Esmailpour[2]
Department of Electrical and Computer Engineering
University of New Haven, West Haven,CT
aesmailpour@newhaven.edu[2]

***Abstract* – With the vast increase in the amount of information, nearly as high as 90% jump in volume of data compare to previously recorded values only in the past two years. As the data size gets increased, it is essential to have proper facilities to handle these data. Therefore, most of the companies use Hadoop in their application. Hadoop is an open source software that is used for reliable, scalable and distributed computing. Yet Hadoop has its own problems in its architecture which results in point of failure in the job tracker. In this paper, we design a solution to handle the problems that may occur if the job tracker fails. The architecture consists of one job tracker and several task trackers within the cluster. A monitoring agent is assigned to monitor the functionalities of job tracker and replace it with the highly efficient task tracker. The newly assigned job tracker can enhance the knowledge from the backup job tracker.**

Keywords – Hadoop, job tracker, point of failure, monitoring agent.

I. INTRODUCTION

Hadoop is an open source implementation of Map Reduce consisting of two components [1]. The two major components are Hadoop Distributed File System (HDFS) and Map Reduce Engine. The HDFS runs on top of the existing file system on each node in cluster. It is designed for very specific data access code. It runs the map reduce programs for distributed data processing, structured and unstructured data. The architecture is similar to master slave model. The map reduce engine has two phases: map and reduce. These phases perform calculation on distributed system. The input data is broken into several pieces of data and the map reduce operations are performed on those data. After performing the essential operation, the data are then sent back. The input data will be replicated and stored in multiple locations.

Although Hadoop handles such huge amount of data in parallel, there are certain drawbacks that may reduce the efficiency in handling the data. One such drawback is the failure of Job Tracker because of the availability of a single job tracker. The job tracker acts as the central heart of the system within the cluster. Hence, productive measures has to be taken to handle this issue. Two factors are to be considered: the recovery time and the ability to maintain the functionality.

In this paper, we provide a solution to enhance the availability of Map Reduce. This proposed scheme introduces a monitoring agent which continually monitors the functionalities of the job tracker (the master node) and the task tracker (the slave node) at regular intervals. There is also a backup job tracker available which maintains the job logs of the current job tracker. By keeping so, we can maintain the consistency of the data within each cluster. The monitoring agent maintains the logs and the efficiency in the performance of the task tracker and allocate a value accordingly. During any discrepancy in the job tracker functions, the monitoring agent will remove the corresponding job tracker and replaces it with the task tracker with high efficiency recorded from the logs. The new job tracker will be updated automatically with all the necessary job logs from the backup job tracker that is available.

The Organisation of this paper is as follows: Section 2 describes the Literature Review, Section 3 covers the scope of the problem and its impact. In Section 4 we propose and describe a potential solution for the problem.

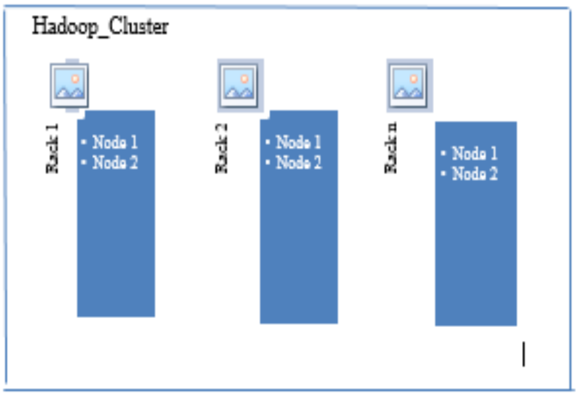

Fig. 1 Architecture of Hadoop

II. LITERATURE REVIEW

Some of the existing solutions proposed to enhance the availability of Map Reduce only implement a standby backup and take a long time to recover the job tracker [2]. Both the master node and the slave node can suffer from any failure. At such situation, all the operations will be frozen and can be continued only after recovery.

Several solutions have been proposed to handle the failure of name nodes [3]. One name node will be in active state whereas the other node will be in passive state. If the active name node fails, then the passive name node encapsulating this and enhance fast recovery. However, they cannot obtain the operation log for the data recovery. A standby job tracker acts as a hot backup node. The standby job tracker synchronises the execution process with the backup job tracker. This solution is implemented in Hadoop 0.20x [4].

The map reduce and the Hadoop framework are designed in such a way that they automatically recover the node failures. Each cluster has only one job tracker which is the daemon service for tracking and executing the map reduce jobs in Hadoop. So this is a single point of failure. If there is any failure, the entire system will be halted [5].

III CLEAR SCOPE OF THE PROBLEM AND ITS IMPACT

In this paper, we focus on a single problem that may occur due to the failure of master node. The map-reduce framework plays a vital role in handling large amount of data. Hence proper recovery measures have to be taken during failure within a short span of time. The job tracker only receives and schedules the job tasks as well as monitoring jobs in task tracker. In case there is a failure in receiving or scheduling the jobs then the whole process will be halted. This leads to bottleneck of jobs scheduling. The time overhead of the recovery should also be considered for successful performance.

IV PROPOSED SOLUTION FOR THE PROBLEM

In general, the client sends the job schedules to the job tracker. It maintains the job logs and schedules as well. There can be more than one slave node under each master node per cluster.

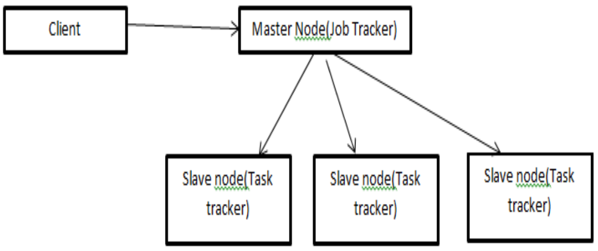

Figure 2

Fig. 2: Flow chart of the general functions within the cluster

We provide a solution by introducing a monitoring agent for each cluster. This monitoring agent will keep an update on the jobs and logs of the master node as well as slave nodes. The efficiency of the slave node can be measured based upon their performance recorded while executing their scheduled jobs. Accordingly we increment a variable for each slave node within the clusters. There is also a backup node which keeps updated about the job schedules.

The main functionality of the proposed monitoring agent is to observe each clusters at specific time intervals. During that time, if there is any mismatch in the functionality of the job tracker it records the information about it. The corresponding job tracker has to be removed from the cluster. The monitoring agent records the efficiency of each of the task tracker and increment a variable based upon their performance. During some failure of the job tracker, the monitoring agent will replace the defective job tracker with the task tracker which has the highest value. The new job tracker will then be updated with the log information and the job schedules from the backup job tracker.

Using this method, we can ensure that the efficiency can be improvised as well as the failure over time can be handled effectively. In addition, if there is any failure during the job execution, then the job tracker will immediately be replaced by the backup job tracker as the information will be updated in this as well.

The sample illustration can be demonstrated in the Figures 3 and 4.

Figure 3

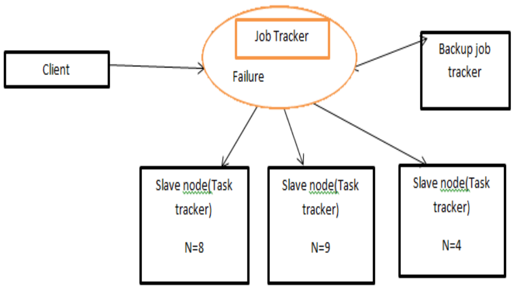
Client
Job Tracker
Failure
Backup job tracker
Slave node(Task tracker) N=8
Slave node(Task tracker) N=9
Slave node(Task tracker) N=4

Fig. 3: Hadoop architecture when job tracker fails

Figure 4

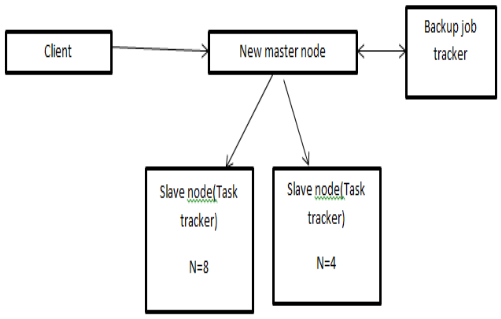

Fig. 4: Hadoop architecture with proposed new master node

V CONCLUSION

Thus the entire framework will be monitored which helps to take care of the failure of master as well as slave nodes.

The implementation will be included in the final version of the paper.

REFERENCES

[1] Apache Hadoop. [Online] Available: http://hadoop.apache.org/ Copyright © 2014 The Apache Software Foundation. Last Published: 03/14/2014 17:31:53

[2] Devarajulu K "High Availability for Job Tracker". [Online] Available: https://issues.apache.org/jira/browse/ZOOKEEPER-1080. Last updated:12/Apr/13 11:48

[3] Huawei, "Achieving high availability for name node using backup name node as hot backup." [Online]. Available: https://issues.apache.org/jira/secure/attachment/12485124/Namenode+HA+using+Backup+Namenode+as+Hot+Standby.pdf

[4] Jian Wan "Dual-JT : Toward the High Availability of Job Tracker in Hadoop" Cloud Computing Technology and Science(CloudCom), 2012 IEEE 4th International Conference 3-6 Dec 2012. Pages 263-268.

[5] Jared Evans "Fault tolerance in Hadoop for Work Migration". [Online] Available: http://salsahpc.indiana.edu/b534projects/sites/default/files/public/0_Fault%20Tolerance%20in%20Hadoop%20for%20Work%20Migration_Evans,%20Jared%20Matthew.pdf. CSCI B34 Survey Paper 3/28/11.