IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

Accident Prediction from Traffic Data using Hadoop

Pallavi Dubey[1], Prof. Manaswini Panigrahi[2]

Department of Computer Science and Engineering, IES Institute of Technology and Management, Bhopal, India[1, 2]

Abstract: Accident prediction has been in trend to provide alerts before accidents happen. Traffic on highways is monitored and lots of data is processed daily to predict probability of accidents based on highway conditions like road surface, light on highway, turns etc. In this paper to predict accident based on different queries and process this big data Hadoop has been used. It is found that execution time is very less on Hadoop as compared to sequential techniques.

Keywords: Data analysis, Hadoop, MapReduce, HDFS

I. INTRODUCTION

Hadoop is an open source fault tolerant distributed framework. Hadoop is a platform in which the implementation of Mapreduce is there, which allow the processing of large data set across the low-cost commodity hardware clusters. The main job of the Hadoop server cluster is to store and process data. It provides high availability of data as, it is designed to work which thousands of machines. As it works which thousands of machines due to which the failure of the node in the cluster would also arise. Thus, to handle and monitor this problem with the library itself is designed so.

It consists of two main core components
1. Hadoop Distributed File System
2. MapReduce

Hadoop distributed file system (HDFS) is implemented to store the large set of data. HDFS provides the higher throughput and availability of data, as it basically carries the replication of data.

MapReduce
Mapreduce is a programming, model. It consists of the mapper and reducer phase of generating and processing the large set of data. The function of the mapper is to take the input in a pair of key and value and the function of reducer is to handle that intermediate key and value. The key value is nothing but the data which is related to that particular task, i.e. the value and the group of the no. of value is a key. Then it is forwarded to the reducer. The merge process is carried out by reducer which merges these set of value in order to get the small set of values into the same node. The different values that the reducers phase receive having the same key into nodes.

Hadoop is basically an open source platform consist of the implementation of Mapreduce. A single hadoop cluster consists of no. of task tracker, job tracker, name node, data node, and the processing, function of each are described further.

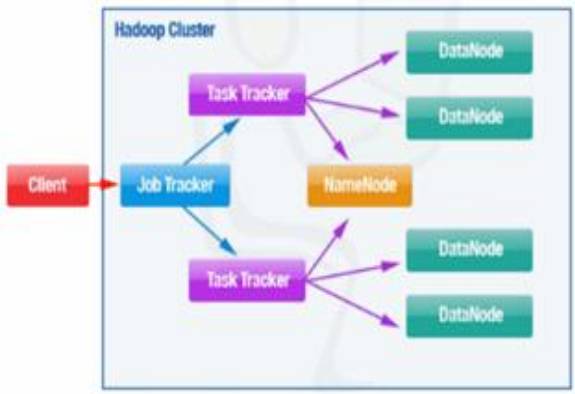

Figure1: Hadoop Cluster

The two main components of apache hadoop are
1) HDFS
2) MapReduce

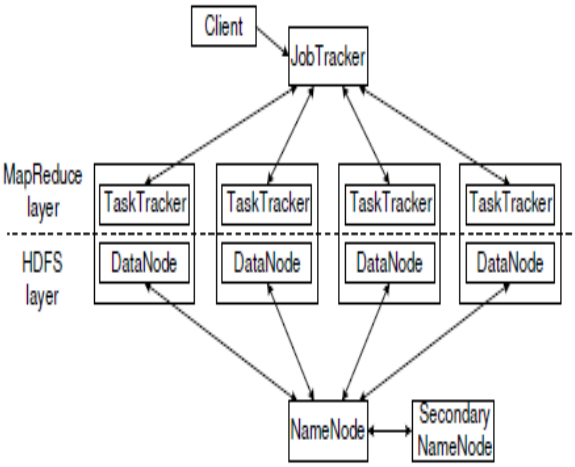

Figure 2: Hadoop Architecture

To manage the HDFS it consists of no. of Name Node, Data Node, and Secondary Name Node, Job Tracker, and Task Tracker are there to perform the MapReduce.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

- HDFS is mainly for storing large sets of a data file. It is so designed to handle the failures on an individual machine. HDFS are typically a workflow, not a primary data storage, for performing the Mapreduce data is copied over HDFS, and to get the result it is again copied form HDFS. To increase the reliability of HDFS, it keeps the data replicas on three machines, with 1 replica on one rack and other replica on the different racks. As in Fig 3, the whole architecture of HDFS is shown.

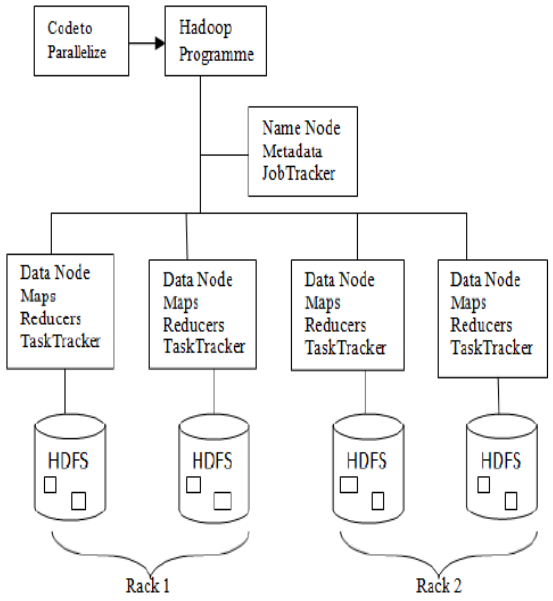

Figure 3: HDFS Architecture

- MapReduce is a framework for performing the parallel processing of a large set of data i.e unstructured and structured. It consist of two phases, which are mapper and reducer phase, it takes the key/ value pair as an input, and perform some operation on this input and produce the relevant result in the form of key/ value. To process these results reduce phase is required as specified by the reduce function. The data from the mapper phase is shuffled, which means the data is exchanged and merge-sorted, to the machine in order to perform the reduce phase.

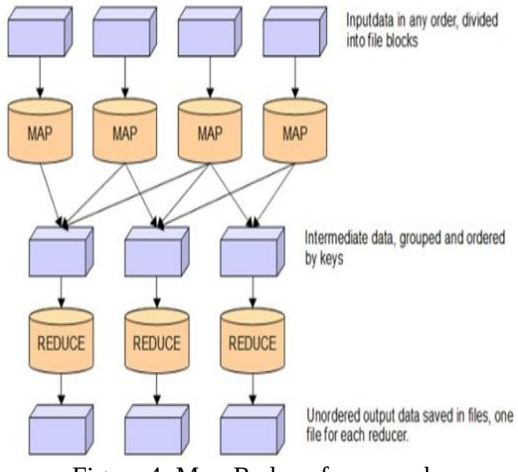

Figure 4: Map-Reduce framework

The architecture of Mapreduce is given in Fig 4, which represent the function of mapper and reducer phase.
In order to describe it further the data is processed in the following 6 steps

1. Input reader: The basic function of input reader is to take the input file, which is a large block and convert it into key/value pairs. The unit of data is split, and the data which is from the input reader is distributed into the splits, and processed by a map task. The usual split size of a block is 64 MB by default, but it is configurable.

2. Map Function: The Map function takes the input from the input reader in the form of key/ value, perform the operation of map function on it, and give the output as a new key/value pair.

3. Combiner Function: This step of Mapreduce is optional, which is performed in the following cases.

- When there is repetition in the keys produced by the map task.
- When the user specified the reduce function is commutative and associative.

In the above cases, the partial reduction will perform the combiner function so that the pair which own the same key will be processed in a group by a reduce task.

4. Partition function: The hashing function is performed in this step by default, whose function is to perform the partition of the intermediate, which are the output from the map task to reduce task. This can be useful by other partition function also, which are known as the user define partition function, generally it provides good balancing.

5. Reduce Function: The pair with the same key will get processed in a group, the reduce function is called once for each different key. It is guaranteed that the input to each reduce task is processed in order to increase the key order. During the sorting process, the user defines its comparison function to be used.

6. Output Writer: The output from the output writer is written in a stable storage. Basically to a file the function can be modified, so as to store the data in a database.

The need for providing the input reader and output writer depends on the sources and destination of data, whereas the need of combiner and partition depends on the data distribution.

The access point for clients is Job Tracker in Hadoop. The main function of Job Tracker is to provide a fair and efficient scheduling of the mapreduce incoming jobs, and the other function is to assign the task for the execution, which is performed by the Task Tracker.

On the basis of available resources no. of a task will be run by the task tracker, when it is ready a new task is allocated.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

II. ACCIDENT PREDICTION USING MAP/REDUCE

We experiment our method based on the MapReduce for solving user asked query from traffic prediction data. The system was built on Apache Hadoop for increasing processing performance using multi-node cluster. In our experiment many queries are processed which are discussed below in table 1.

TABLE 1

Queries	Data set
Q1: Area where maximum accident occurs	Grid Reference Easting and Grid Reference Northing and Road surface
Q2: On which Highway Accidental Timing.	Accident date and Time and Road class
Q3: On which road Maximum Accident Occur due to Road Surface.	Road class and Road surface and Number of causality
Q4: Due to Lighting Problem on which road Maximum Accident occurs	Road class and Lighting condition and Number of causality
Q5: Probability of Accident at any location when drive is male or female.	Sex of causality and Age of Causality and Causality class

We experiment these queries which are discussed in table 1 on different amount of data by using C and Hadoop and generate the result in mm/sec . The experiment data tables are

TABLE 2

USING C					
Data	Q 1	Q 2	Q 3	Q 4	Q 5
500 MB	6412	6790	6127	7116	6342
1 GB	17342	19040	16809	19800	16243
2 GB	679421	732046	650310	792016	639140

TABLE 3

USING HADOOP					
Data	Q 1	Q 2	Q 3	Q 4	Q 5
500 MB	1309	1457	1297	1599	1092
1 GB	2094	2197	1876	2201	1643
2 GB	5197	5407	4837	5917	4561

III. RESULT

These graphs show the CPU execution time for executing individual query for predicting accident from traffic data.

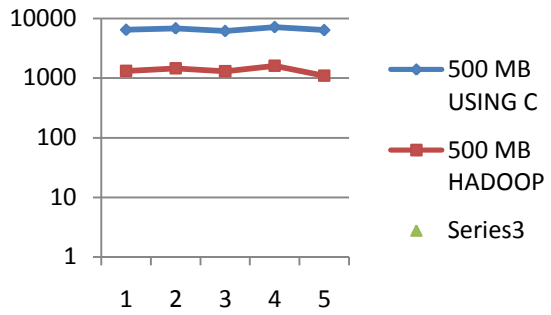

Figure 5: Graph for executing queries on 500 MB data

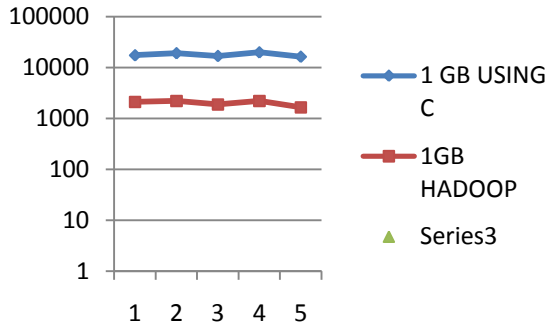

Figure 6: Graph for executing queries on 1 GB data

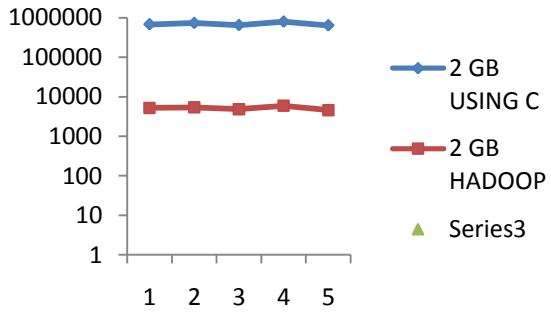

Figure 7: Graph for executing queries on 2 GB data

IV. CONCLUSION

Large amount of data has been generated these days in every domain. Traffic data has been tracked these days in various countries to give alert for accident and highway information. In this paper accident prediction has been done for various queries.

Results are compared for 500 MB, 1 GB and 2 GB for queries using C and Hadoop Map/reduce. It is found maximum of 140 times speedup is achieved by executing Map/reduce on 4 node cluster.

REFERENCES

[1] Seoung -hun Park, Young- guk Ha, "Large Imbalance Data Classification Based on MapReduce for Traffic Accident Predication" , IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing,pp 45-49,2014.
[2] J. Conejero, P. Burnap, O. Rana, and J. Morgan, "Scaling Archived Social Media Data Analysis using a Hadoop Cloud", IEEE, 2013.
[3] S. G. Manikandan and S. Ravi, "Big Data Analysis Using Apache Hadoop," 2014 International Conference IT Converge. Security, , pp. 1–4, Oct 2014.
[4] J. Nandimath, "Big Data Analysis Using Apache Hadoop", pp. 700–703, 2013.
[5] S. Maitrey and C. K. Jha, "Handling Big Data Efficiently by Using Map Reduce Technique", IEEE Int. Conf. Comput. Intell. Commun. Technology, pp. 703–708, Feb. 2015.
[6] S. Humbetov, "Data-Intensive Computing with," 2012.
[7] L. P. Thompson and D. P. Miranker, "Fast Scalable Selection Algorithms for Large Scale Data," pp. 412–420, 2013.
[8] D. Chung, X. Rui, D. Min, and H. Yeo, "Road traffic big data collision analysis processing framework," 2013 7th Int. Conf. Appl. Inf. Commun. Technol., pp. 1–4, Oct. 2013.
[9] J. Shafer, S. Rixner, and A. L. Cox, "The Hadoop Distributed File system : Balancing Portability and Performance."
[10] M. Wang, S. B. Handurukande, and M. Nassar, "RPig : A Scalable Framework for Machine Learning and Advanced Statistical Functionalities", IEEE 4th International Conference on Cloud Computing Technology and Science ,2012, pp. 3–10, 2012.
[11] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. Commun. ACM, 51(1):107–113, 2008.
[12] S. Guha, "Computing environment for the statistical analysis of large and complex data," Ph.D., Purdue University, 2010.
[13] Marcin Jedyk, MAKING BIG DATA, SMALL, Using distributed systems for processing, analyzing and managing large huge data sets, Software Professionals Network, Cheshire Data systems Ltd.
[14] OnurSavas, YalinSagduyu, Julia Deng, and Jason Li,Tactical Big Data Analytics: Challenges, Use Cases and Solutions, Big Data Analytics Workshop in conjunction with ACM Sigmetrics 2013,June 21, 2013.
[15] Kyuseok Shim, "MapReduce Algorithms for Big Data Analysis", 2013, LNCS 7813, pp. 44–48.
[16] Apache™, "Apache™ Hadoop", http://hadoop.apacahe.org.