META MAP-REDUCE USING BOOSTING ALGORITHM.

Ritu Vachhani, Pushpakumar Nishandar, Sushant Gaikwad, Aditya Desai, Rahul Sonkamble

Sanjay Ghodawat Institute, Atigre, Kolhapur, India.
rv.rituvachhani@gmail.com, pmnishandar007@gmail.com, sushantgaikwad9999@gmail.com, adi007desai@gmail.com, sonkamble.rg@sginstitute.in.

Abstract— In the big data age, extracting knowledge from massive data has become a more important concern. Hadoop MapReduce provides two functions namely Map and Reduce help us in implementing machine-learning algorithms using a feasible framework. However, this framework has a weakness that it does not support repetitions. Therefore, algorithms requiring repetitions do not operate at full efficiency in MapReduce framework. Hence, in this paper we propose to apply advanced learning processes, which are meta-learning algorithms to prevent MapReduce from parallelizing machine-learning algorithms. It also improves scaling properties of big data. Our algorithm reduces the computational cost by reducing the complexity. This is achieved by increasing the number of computers whereby decreasing the errors.

Keywords—MapReduce, Big data, Hadoop.

Introduction

Nowadays, it is becoming more and more important to organize and utilize the massive amounts of data as we have been moving from the age of Terabytes to Petabytes. Considering massive data, building classification models using data mining algorithms currently available is difficult. Therefore, to achieve more efficiency and effective models these algorithms are not sufficient. Talking about Big Data, it is something that cannot be easily processed using traditional computing techniques. For example, YouTube which works and manages the data on daily basis. Big Data involves different aspects such as velocity, variety, volume and complexity. Big Data challenges include searching, sharing, transferring, querying, data analysis and information policy.

MapReduce is a programming model which runs in background of Hadoop. It is a programming paradigm which can be used for writing applications that can be processed in parallel for big data on multiple nodes. It is used for analyzing huge data in a simple way.

- *Application*

1) Due to the popularity of non-negative matrix factorization and the increasing availability of huge data sets leads to the problem of factorizing large matrices of dimensions in the order of millions. It is feasible to factorize a million-by-million matrix with billions of nonzero elements on a map reduce cluster.

2) C4.5 is an algorithm which is a successor of ID3 algorithm used to generate a decision tree having a high accuracy in decision making.

3) OLS (Ordinary Least Squares) used to minimize the sum of square difference between the observed and predicted values.

4) Adaboost (Adaptive Boosting) used in conjunction to improve performance, the output of the learning algorithms is combined into a weighted sum that represents final output of the bosteed classifier.

Proposed approach

Hadoop**:** Hadoop is an open source implementation of a large scale batch processing system. Although the Hadoop framework is written in java it allows developers to deploy custom written programs coded in java. [1] The Hadoop ecosystem contains Hadoop kernel, MapReduce and the Hadoop distributed file system (HDFS).

Hadoop useful in: Complex information processing, Unstructured data needs to be turned into structured data, heavily recursive algorithms, Machine learning, Fault tolerance is critical.

Installation of hadoop

Installation of Hadoop Windows:
To install Hadoop, go to Apache site and download tar file from their ftp server.
Download Cygwin application, this helps us to unzip the tar file by providing you the Linux based terminal.
Set the path for java Jdk and Jre respectively for java support environment.
Download Eclipse Java Ide for Map-Reduce manipulation on the Hadoop.
Download Apache ant 1.9.6 which is a java based build tool for automating the build and deployment process.

Installation of Hadoop on Linux
Update $HOME/.bashrc. Excursus: Hadoop Distributed File System (HDFS).
Configuration- hadoop-env.sh .conf/*-si e.xml.
Formatting the Hadoop file system through the help of the Name node.
Starting your single node cluster.

Map Reduce: Map Reduce is a way of distributing data and making huge tasks smaller. It divides input data into smaller and manageable sub-tasks to execute them in parallel.
The Map Reduce algorithm uses the following:
Map Function: Map function is the first step in Map Reduce algorithm. It takes input tasks and converts them into smaller ones to perform computation on each sub-task in parallel.
The Map function takes place in two steps, Splitting and Mapping.
Shuffle Function: It is the next step in Map Reduce algorithm. It works in two steps:
1) Merging. 2) Sorting.
Reduce Function: It is the last stage in Map Reduce algorithm. It takes list of <key, list<value>> sorted pairs from shuffle function and perform reduce operations.
No Abstraction - Hadoop does not have any type of abstraction so Map Reduce developer need to hand code for each and every operation which makes it difficult to work.
Hadoop Map Reduce is a framework for processing large data sets in parallel across a Hadoop cluster.
Map Reduce is divided mainly into two parts viz. [5] Map and Reduce.
Map: This takes a set of data as input works on it and converts it into another set of data and breaks down every single element into tuples (key-value pairs). Using Mark Logic Connector input data is fetched.

Reduce: The reduce task takes the output from the above Map as an input and combines those data tuples into a smaller set of tuples. Using the same Mark Logic Connector, the data is stored. By Default, the Map Reduce framework gets input data from the HDFS. The data then goes through Map Reduce Algorithm where above two tasks are performed. Map Reduce implements various mathematical algorithms to divide a task into small parts and assign them to multiple systems. This algorithm helps in sending the Map and Reduce tasks to appropriate servers in a cluster.

Mathematical algorithms:

1)Sorting

2)Searching

3)Indexing

4)TF-IDF

All types of structured and unstructured data need to be translated before providing it to the Map Reduce model.

Boosting: We develop the boosting algorithms AdaboostPL using mapreduce to boost the overall process and support the running Decision-Tree algorithm which in turn makes the process faster

Block Diagram

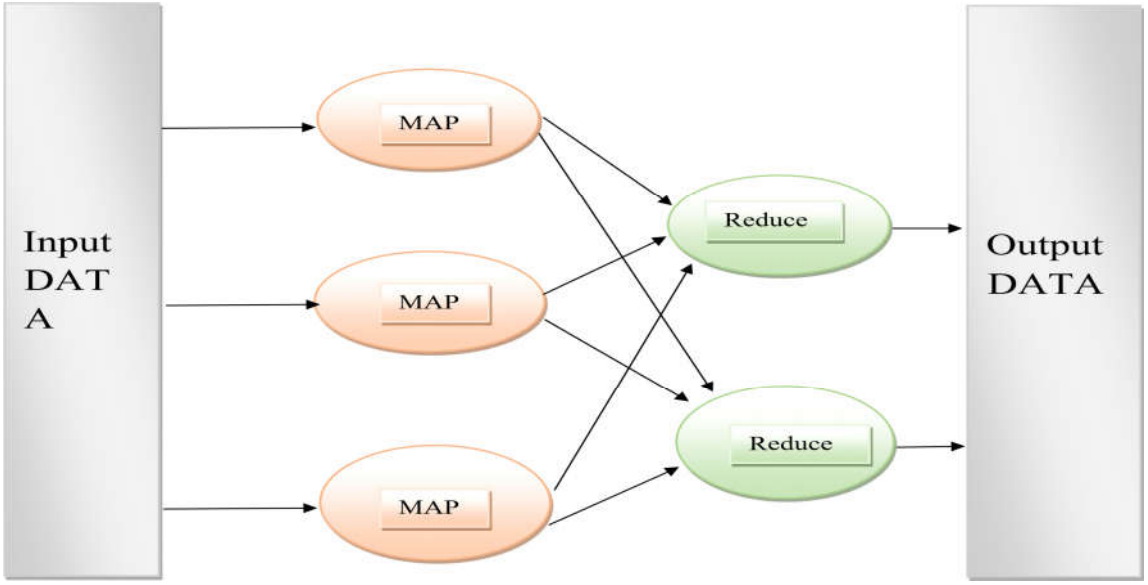

Fig. 1. Mapreduce process

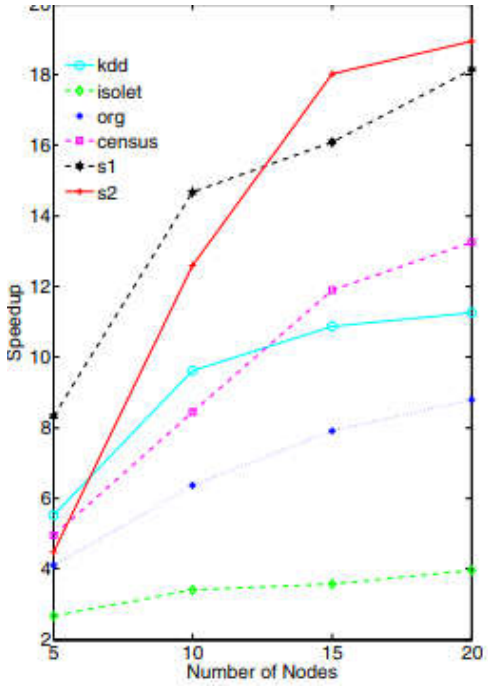

Fig. 2 Result analysis

Conclusion

Hadoop MapReduce is a large scale open source software framework dedicated to scalable, distributed, data intensive computing. [1] The framework breaks up large data into smaller parallelizable chunks and handles scheduling. Maps each price to an intermediate value and reduces it to a solution, User specified partition and combiner option, [5] if you can rewrite algorithms into maps and reduces then your problems can be broken up into small pieces which can be processed in parallel, then Hadoop's MapReduce is the way to go for the distributed problem solving approaches to large database.

Usually it is observed that the MapReduce framework generates a large amount of intermediate data. Such information is dumped when the task finishes, because MapReduce is unable to utilize them. Therefore, we propose a data-aware cache framework for big data application them its task submit their intermediate results to the cache manager. The task asks the cache manager before executing the actual computing work.

REFERENCES

1) Dhole Poonam B, Gunjal Baisa L, "Survey Paper on Traditional Hadoop and Pipelined Map Reduce" International Journal of Computational Engineering Research||Vol, 03||Issue, 12||.

2) Xuan Liu*, Xiaoguang Wang, Stan Matwin1 and Nathalie Japkowicz Meta-MapReduce for scalable data miming Liu et al.journal of big data (2015) 2:14 DOI 10.1186/s 40537-015-00214.

3) Nilam Kadale, U. A. Mande, "Survey of Task Scheduling Method for Mapreduce Framework in Hadoop" International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA 2nd National Conference on Innovative Paradigms in Engineering & Technology (NCIPET 2013) – www.ijais.org.

4) Suman Arora, Dr.Madhu Goel, "Survey Paper on Scheduling in Hadoop" International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014.

5) Wang, F. et al. Hadoop High Availability through Metadata Replication. ACM (2009). B.Thirumala Rao, Dr. L.S.S.Reddy, "Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments", International Journal of Computer Applications (0975 – 8887) Volume 34– No.9, November 2011.

6) Vishal S Patil, Pravin D. Soni, "HADOOP SKELETON & FAULT TOLERANCE IN HADOOP CLUSTERS", International Journal of Application or Innovation in Engineering & Management (IJAIEM)Volume 2, Issue 2, February 2013 ISSN 2319 – 4847.

7) Sanjay Rathe, "Big Data and Hadoop with components like Flume, Pig, Hive and Jaql" International Conference on Cloud, Big Dataand Trust 2013, Nov 13-15, RGPV.

8) Yaxiong Zhao, Jie Wu and Cong Liu, "Dache: A Data Aware Caching for Big-Data Applications Using the MapReduce Framework",TSINGHUA SCIENCE AND TECHNOLOGY ISSNl1007-0214l 05/10l lpp39-50 Volume 19, Number 1, February 2014.

9) B.Thirumala Rao, Dr. L.S.S.Reddy, "Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments", International Journal of Computer Applications (0975 – 8887) Volume 34– No.9, November 2011