Data Partitioning for Minimizing Transferred using MapReduce
M. Santhana Lakshmi, D. Sandhiya, A. N. Thasneem, S. Sivashankar

Abstract:
Cloud Computing leverages Hadoop framework for process BigData in parallel. Hadoop has sure limitations that could be exploited to execute the task with efficiency. These limitations area unit principally thanks to knowledge neck of the woods within the cluster, jobs and tasks scheduling, and resource allocations in Hadoop. Economical resource allocation remains a challenge in Cloud Computing MapReduce platforms. We have a tendency to propose H2Hadoop that is AN increased Hadoop design that reduces the computation price related to Big Data analysis. The projected design conjointly addresses the problem of resource allocation in native Hadoop. H2Hadoop provides a better answer for “text data”, like finding polymer sequence and also the motif of a polymer sequence. Also, H2Hadoop provides AN economical Data Mining approach for Cloud Computing environments. H2Hadoop design leverages on NameNode’s ability to assign jobs to the TaskTrackers (DataNodes) among the cluster. By adding management options to the NameNode, H2Hadoop will showing intelligence direct and assign task to the datanode that contain the desired knowledge while not causation the task to the entire cluster. Scrutiny with native Hadoop, H2Hadoop reduces hardware time, variety of browse operations, and another Hadoop factors.

Index term: Cloud Computing, Hadoop, H2Hadoop, Hadoop Performance, Map Reduce, Text Data.

I. INTRODUCTION
Parallel processing in Cloud Computing has emerged as an interdisciplinary doe’s research area due to the heterogeneous character and huge size of data. Translating chronological data to evocative information requires substantial computational power and efficient algorithms to identify the degree of similarity among multiple sequences. Sequential guide mining or data analysis application such as, DNA order aligning and motif finding regularly require large and complex amounts of record dealing out and computational capabilities. Efficiently targeting and arrangement of computational property is required to solve such multipart problems. Although, some of the data sets are understandable by humans, it can be very complex to be unspoken and process using traditional processing techniques.

Availability of open source and commercial Cloud Computing equivalent processing platforms has opened new avenue to explore structured semi-structured or unstructured data. Before we go any further, it is necessary to define certain definitions that are related to Big Data and Hadoop. In this paper we are analyzing Bank data by using hadoop tool along with some hadoop ecosystems like hdfs, mapreduce, sqoop, hive and pig. By using these tools we can process no limitation of data, no data lost problem, we can get high throughput, maintenance cost also very less and it is a open source software, it is compatible on all the platforms since it is Java based.

II. LITERATURE SURVEY
1. Jiong Xie, Shu Yin is proposed Improving MapReduce Performance through Data Placement in Heterogeneous Hadoop Clusters. MapReduce has become an important distributed processing form for large-scale data-intensive application like data mining and net indexing. Hadoop--an open-source implementation of MapReduce is widely used for short jobs requiring low response period. The present Hadoop implementation assumes that compute nodes in a cluster are homogeneous in nature. Data locality has not been taken into account for launching speculative map tasks, as it is believed that generally maps are data local. Unluckily, both the homogeneity and data locality assumptions are not satisfied in virtualized data centers. We show that ignoring the data- area issue in diverse environments can clearly reduce the MapReduce performance. In this paper, we address the problem of how to place data across nodes in a way that every node has a even-handed data giving out load. Given a data-intensive application organization on a Hadoop MapReduce cluster, our data placement scheme adaptively balances the amount of data store in each node to get improved data-processing show. Experimental results on two real data-intensive applications show that our data assignment strategy can forever improve the MapReduce performance by rebalancing data crosswise nodes before the stage a data intensive application in a heterogeneous Hadoop cluster.

2. Xuhui Liu, Jizhong Han is proposed Implementing Web GIS on Hadoop: A Case Study of Improving Small File I/O Performance on HDFS. Hadoop framework has been widely used in various clusters to build large scale, high performance systems. However, Hadoop distributed file scheme (HDFS) is planned to manage large files and suffer performance penalty while managing a large amount of small files. As a consequence, many web applications, like Web GIS, may not take benefits from Hadoop. In this paper, we propose an approach to optimize I/O performance of small files on HDFS. The basic idea is to combine small files into large ones to decrease the file number and build index for every file. Furthermore, a few novel features such as grouping neighboring files and reserving several latest versions of data are considered to meet the characteristics of Web GIS access patterns. Preliminary experiment results show that our approach achieves better performance.
3. Bo Chen, Reza Curtmola, Giuseppe Ateniese, Randal Bums has proposed Improving Reduce Task Data Locality for Sequential MapReduce. In this research, a novel secure and efficient RDC scheme for network coding-based distributed storage systems. RDC-NC mitigates new attacks that stem from the original principle of network code. The scheme is able to preserve in an adversarial setting the minimal communication overhead of the repair element achieved by network coding in a caring setting. We implement our system and experimentally show that it is computationally inexpensive for both clients and servers.

4. Jian Tan, Shicong Meng, Xiaqiao Meng as proposed Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. Improving data locality for MapReduce jobs is critical for the presentation of large-scale Hadoop clusters, embody the standard of moving computation close to data for big data platforms. Scheduling tasks in the vicinity of stored data can significantly diminish network traffic, which is crucial for system stability and efficiency. Though the issue on data locality has been investigated extensively for Map Tasks, most of the existing schedulers discount data locality for Reduce Tasks when smart the intermediary data, cause performance degradation. This problem of reducing the fetching cost for Reduce Tasks has been identified newly. However, the planned solutions are totally based on a greedy approach, relying on the intuition to place Reduce Tasks to the slots that are closest to the majority of the already generate intermediate data. The importance is that, in presence of job arrival and departures, assigning the Reduce Tasks of the current job to the nodes with the lowest fetching cost can avoid a subsequent job with even superior match of data position from being launched on the already taken slots. To this end, we formulate a stochastic optimization framework to improve the records locality for cut Tasks, with the optimal placement policy exhibit a threshold-based structure. In order to ease the implementation, we further propose a receding horizon manage policy based on the optimal result under secret conditions. The improved performance is further validated through simulation experiments and real performance tests on our tested.

5. Radha G. Dobale and Prof.R.P. Sonar as proposed Review of Load Balancing for Distributed Systems in Cloud. Proposed work discusses the load balancing concept in a distributed manner in which nodes execute their load comparison tasks alone without synchronization or global knowledge regarding the system. A fully distributed load balancing algorithm is proposed to hack it with the load difference problem. Instead of partition a file into a no. of chunks and balancing a load by migrating different chunks to different chunk servers, the load balance Nearest Search algorithm roam one user’s one total file into any one node. pack is transferred from heavily loaded node to physically closed lightly loaded node. This proposal strives to balance the loads of nodes and cut the demand movement cost with shrink spending on technology as much as possible.

6. Mohammad Hammoud and Maid F. Sakr has proposed Locality-Aware Reduce Task Scheduling for MapReduce. MapReduce offers a promising programming model for big data processing. Inspired by useful language, MapReduce allow programmers to inscribe functional-style symbols which get automatically divided into multiple maps and/or reduce tasks and scheduled over distributed data transversely several machines. Hadoop, an open source execution of MapReduce, schedule map tasks in the vicinity of their inputs in order to moderate network traffic and advance performance. However, Hadoop schedules ease tasks at request nodes without considering data area leading to performance degradation. This paper describes Locality-Aware Reduce Task Scheduler, a practical scheme for recovering MapReduce performance. LARTS attempt to collocate decrease tasks with the maximum required data computed after recognizing input data network locations and size. LARTS adopts a mutual paradigm seeking an excellent data locality while circumventing arrangement delay, scheduling skew, poor system utilization, and low degree of parallelism. We implement LARTS in Hadoop-0.20.2. Estimate result show that LARTS outperforms the native Hadoop reduces task scheduler by an average of 7%, and up to 11.6%.

7. Nishanth S, Radhikaa B, Ragavendr TJ, Chitra Babu, and Prabavathy B as proposed CoRadoop++: A Load Balanced Data Co- location in Radoop Distributed File System. Hadoop is a popular distributed computing framework, generally used for production data analytics. While Hadoop performs equivalent operations on bulky data sets, it does not co-locate related data by fail to pay. However, performance of log handing out operation such as indexing, grouping join and sessionization on Hadoop can be radically improved if related logs are partitioned and processed as a group. In order to facilitate this, these partition need to be sited on the alike set of nodes in the cluster. To allow this, a grouping key can be used to identify the related logs. CoHadoop , an extension of Hadoop, use this key to co-locate all those files which communicate to the same key. However, it go for the data nodes randomly for every new key. While doing all the mappers, thus, this phase requires data from all mappers be moved to the node where the reducer task is running.

III. SYSTEM ARCHITECTURE

Figure 1. System Architecture

The proposed method performs well in the general populace as well as in sub-populations. Results indicate that the proposed
model extensively improves prediction over established baseline methods analyzing student performance. The goal of this study was to analyze bank data for last year which person is highest deposited and which person highest transfer money another account and which person highest withdraw. For casting for next year.

IV. TECHNOLOGIES USED IN THIS PROJECT

In this project we are analyzing airlines data by using below tools

HADOOP: Hadoop is an open source Apache project. Hadoop framework was written in Java. It is scalable and so can maintain high performance demanding application. Storing very large amounts of data on the file systems of multiple computers is possible in Hadoop structure. It is configured to permit scalability from lone node or computer to thousands of nodes or independent systems in such a way that the individual nodes use local computer storage, CPU, memory and processing power.

HDFS (hadoop distributed file system):
Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is extremely fault open-minded and designed using low-cost hardware. HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across many machines. These files are store in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

MapReduce:
MapReduce is a processing technique and a program model for distributed computing based on Java. The MapReduce algorithm contains two important tasks, to be exact Map and Reduce. Maps take a set of data and convert it into another set of data, where individual elements are broken down into tuple (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuple into a smaller set of tuple. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

Sqoop:
Sqoop is a tool designed to transfer data between Hadoop and relational database servers. It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS, and sell abroad from Hadoop file structure to relational databases. It is provide by the Apache Software Foundation.

Hive:
Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to review Big Data, and makes querying and analyzing easy. Initially Hive was developed by Facebook; later the Apache Software Foundation took it up and industrial it further as an open source below the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

Pig:
Apache Pig is an construct over MapReduce. It is a tool/stage which is used to analyze larger sets of data representing them as data flows. Pig is generally used with Hadoop; we can perform all the data management operations in Hadoop using Apache Pig. To mark data analysis programs, Pig provides a high-level language known as Pig Latin. This language provides various operators using which programmers can improve their own function for reading, writing, and processing data. To analyze data using Apache Pig, programmers need to write scripts using Pig Latin language. All these script are internally changed to Map and Reduce tasks. Apache Pig has a factor known as Pig Engine that accepts the Pig Latin scripts as input and converts those scripts into MapReduce jobs.

Java:
Java is a programming language and a platform. Java is a high level, robust, secured and object oriented programming language. Java is used for storing, analyzing and processing large data sets. The pick of using Java as the programming tongue for the development of hadoop is merely accidental and not thoughtful.

FUTURE ENHANCEMENT

We are using spark we can get result hundred times faster than Hadoop. The secret is that it runs in-memory on the cluster, and that it isn't tied to Hardtop's MapReduce two-stage paradigm. This makes constant access to the same data much faster. Spark can run as a separate or on top of Hadoop YARN, where it can read data directly from HDFS.

V. SOFTWARE REQUIREMENTS

In our Project we use Front End as mysql and Back End as a hadoop.

Jdk 1.7:
In our project java is running in background for that purpose we know about core java only and also java is an open source and platform independent this makes the application more flexible.

MY SQL:
My SQL is a relational database management system developed by Sun Micro systems. As a database, it is a software product whose primary function is to store up and retrieve data as request by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are changed workloads (ranging from small application that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time).

Hadoop:
Hadoop is an open source framework from Apache and is used to store process and analyze data which are very huge in volume. Hadoop is written in Java and is not OLAP (online critical processing). It is use for batch/offline processing. It is being used by Facebook, Yahoo, Google, Twitter, LinkedIn and many more. Moreover it can be scaled up just by adding nodes in the cluster.

Linux:
Main reason behind Linux power and popularity is it opens source nature. If the product is initially good and opensourece then people can take it, alter it and use as per it own mode. Also Linux community is very big. So many people comes with ideas of improvement and lots of people involve in...
implement good ideas. This is the big big reason for the increase and reputation of Linux operating system.

**REAL TIME EXAMPLE**
Real time Example for Our Application:
- https://www.yahoo.com/
- https://www.facebook.com/

**ALGORITHM:**
Input: Ky: the number of clusters
Dy: a data set containing n object
Output: A set of Ky clusters
1. Input the data set and value of Ky.
2. If f Ky=1 then Exit.
3. Else
4. Choose k objects from D randomly as the initial cluster centres.
5. For every data point in the cluster j reissue and define every object into the cluster where the object matches, based on the object’s mean value in the cluster.
6. Update cluster means; after that for each cluster calculate the object’s mean value. 7. Repeat from step 4 until no data point was assigned otherwise stop. The satisfying criteria can be either number of iteration or change of position of centroid in consecutive iterations.

**VI. CONCLUSION**
The proposed method performs well in the general population as well as in sub-populations. Results indicate that the proposed model significantly improves predictions over established baseline methods analyzing student performance. The goal of this study was to analyze bank data for last year which person is highest deposited and which person highest transfer money another account and which person highest withdraw for casting for next year.

**VII. REFERENCE:**
[1]. Ming, M., G. Jing, and C. Jun-jie. Blast-Parallel: The parallelizing implementation of sequence alignment algorithms based on Hadoop platform. in Biomedical Engineering and Informatics (BMEI), 2013 6th International Conference on.