Performance Improvement for Global Banking Based on H2hadoop

P. Meenakshi*, K. Kaveri, S. Kaviya Sindhu, M. Muthuselvi

Dept. of Computer Science and Engineering, Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Anna University, Chennai, INDIA.

***Corresponding Author Email:meenakshi@velhightech.com**

ABSTRACT

An increasing number of demands in a banking environment for the analysis and the prediction of the data in the real world situation. In the banking environment, they used a SQL query for retrieving the information from the database. Due to limitation of memory in a database they moved to concept of Hadoop. In a Hadoop, they overcome the problem of data limitation and the data locality in a cluster. In this paper, we proposed the concept of H2Hadoop, which can able to achieve the performance improvement more than the Hadoop. Thus the H2Hadoop concept is approached to the banking environment in order to enhance the performance by using the common job block table (CJBT). The projected design is ready to consolidate, validate, enrich and method with completely different huge knowledge analytics techniques the information gathered from the various supply systems as encountered within the banking follow, whereas at an equivalent time supporting the various knowledge integration, transmission and method orchestration needs historically encountered during a world financial organization.

KEY WORDS: BigData, Cloud Computing, Hadoop, H2Hadoop, MapReduce, CJBT.

1. INTRODUCTION

The paper deals with the concept of the big data that is related to the hadoop. Big Data can be defined as the huge collection of data or the larger data sets that cannot be processed using the traditional Relational database management system (Jagadish, 2014). Big Data can handles either a Relational database management system such as SQL in a structured manner or it handles the semi-structured/unstructured data such as social media data or a DNA Data sets. The 4V's of Big Data are 1) Volume of the information, which implies the information estimate. 2) Velocity, which implies the speed at which the information is created. 3) Varity of the information, which implies the information frames that distinctive applications manage, for example, arrangement information, numeric information or paired data.4) Veracity of the information, which implies the vulnerability of the status of the information or how clear the information is to these applications. There is a lot of challenges in a Big Data such as Physical Storage and a Data Redundancy. There may be another lo tof challenges such as the process of extracting the information, cleaning data, data integration, data aggregation, and data representation. In order to overcome those challenges they need a framework or an environment which is a hadoop framework.

About Hadoop: Hadoop is an Apache open-source programming structure that is composed in Java for circulated stockpiling and circulated handling (Ming, 2013). It gives answers for Big Data handling and examination. It has a document framework that gives an interface between the clients' applications and the nearby document framework, which is the Hadoop Distributed File System (HDFS). Hadoop circulated File System guarantees solid sharing of the assets for productive information examination (Changqing, 2012). The two principle parts of Hadoop are (i) Hadoop distributed File System (HDFS) that gives the information unwavering quality (conveyed stockpiling) and (ii) Map Reduce that gives the framework examination (conveyed handling). Depending on the rule that "moving calculation towards information is less expensive than moving information towards calculation" Hadoop utilizes HDFS to store vast information records over the group. Map Reduce gives stream perusing access, runs undertakings on a bunch of hubs, and gives an information overseeing framework for a dispersed information stockpiling framework. MapReduce Calculation has been utilized for applications, for example, producing seek files, archive bunching, get to log examination, also, unique different sorts of information investigation. “Compose once and read-many" is an approach that licenses information records to be composed just once in HDFS and afterward permits it to be perused many circumstances over as for the numbers of doled out occupations (Lohr, 2012). Amid the composition procedure, Hadoop separates the information into squares with a predefined piece estimate. The pieces are then composed and copied in the HDFS. The pieces can be copied various circumstances in view of a particular esteem which is set to 3 times as a matter of course (Farrahi, 2014). In HDFS, the bunch that Hadoop is introduced in is partitioned into two primary segments, which are (i) the master hub called Name Node and (ii) the slaves called Data Nodes. In Hadoop bunch, single Name Node is in charge of general administration of the document framework counting sparing the information and guiding the occupations to the fitting Data Nodes that store related application information. Data Nodes encourage Hadoop/Map Reduce to prepare the occupations with gushing execution in a parallel preparing in the environment (Lohr, 2012).

About H2hadoop: In existing Hadoop engineering, Name Node knows the area of the information hinders in HDFS. Name Node is in charge of doling out the employments to a customer and partitioning that occupation into assignments. Name Node additionally doles out the errands to the Task Trackers (Data Nodes) (Nishanth, 2013). Knowing which Data Node holds the squares containing the required information, Name Node ought to have the capacity to guide the occupations to the particular Data Nodes without experiencing the entire bunch. In H2Hadoop, before relegating undertakings to the Data Nodes, we actualized a pre-handling stage in the Name Node (Marx, 2013).

Our emphasis is on distinguishing and removing elements to manufacture a metadata table that conveys data identified with the area of the information obstructs with these components (Jagadish, 2014). Any occupation with similar components ought to just read the information from these particular pieces of the bunch without experiencing the entire information once more (Changqing, 2012). In the proposed concept, there is a CBJT table which store the information about the jobs and the blocks associated with the specific feature. The concept is used only for text data. Using the CBJT we get the result from the specific blocks without checking the entire blocks in the cluster. This enable us to get the performance increased in searching the entire clusters.

Lilterature Survey: In 2010, Map Reduce is used for a small job with low response time. Hadoop is used only in the homogeneous form of computing nodes in a cluster. Data locality is the major problem in the Hadoop. In order to overcome those problem there is a rebalancing of nodes across a heterogeneous Hadoop cluster. In 2011, Map Reduce permits the programmer to write a functional code in java that can divide it's automatically into the multiple maps and reduce task scheduled across multiple machine. Locality Aware Reduce Task Scheduler (LARTS), which practical attend for improving the Map Reduce performance. In LARTS perform the reduce task after analyzing the input data network location and the size. In 2013, Hadoop is widely used for business data analysis. The parallel performance on the cluster is possible by partition of the same set of nodes are grouped into a same cluster. Load Balancing should be possible in the cluster group in order to ensure the performance improvement.

2. EXISTING SYSTEM

In existing system we are using my SQL concept for storing and processing the data with some limitations. It is a relational database and structured query language for manipulate and access the database. It is an American national standards institute(ANSI) used for performing the operations like insert the data into the table, delete the data ,update the date data collection, create through records. Another operations like set permission on tables, procedures and views. It is the back end of the system. It can access only limited amount of data not to process large data. Once the data is lost we cannot retrieved the data. SQL takes more time for processing and getting the result. Due to time consuming problem, we move to concept of hadoop which had an extra feature such as there is no limitation of data and data analysis performed in a specific period of time.

Proposed System: We tried H2Hadoop under these particular conditions, which incorporate number of information records and the size of every record. The proposed arrangement could be executed in two diverse ways. In the first place, in situations where there are many source information documents and everyone is not exactly the default estimation of the square size. Second, in situations where there is a one or a few information source records and where the vast majority of the records are bigger than the default hinder in size. In our execution, we utilized DNA chromosome information also, the information source size is around 24 documents. Every document is less than the default piece estimate in Hadoop. Different employments were actualized utilizing the previously mentioned information. The usage of the proposed arrangement goes in three parts.

Creating the Common Job Block Table (CJBT): Utilizing diverse strategies we can perform outline furthermore, make the CJBT. One of them is utilizing a NoSQL database, on a level plane. For example, HBase. It is a section situated database of which a principle property is extended .The explanation behind utilizing. HBase is that it is an Apache open source programming that is one of NoSQL databases that works on top of Hadoop. We utilize HBase as an ordering table here to finish our exploration and empower the proposed arrangement works effectively. We proposed before the UI ought to contain easy to understand interface so that the client is get the advantages of the upgraded plan while picking normal information from records. For instance, while picking the CJN from a rundown of regular employment names that are identified with the comparative information documents. Distinctive types of UIs can be composed based on the client's needs. One of the regular UIs is, the charge line that is normally utilized when the client knows the charges and the related parameters they will exchanges on Cloud Computing.

System Architecture: The below figure 5.1 explain the work flow h2hadoop as follows.

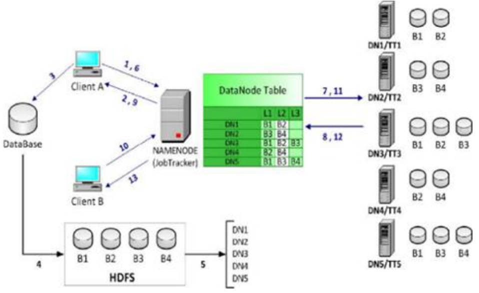

Figure.1. The Work Flow H2hadoop

Map Reduce work process in local Hadoop has been clarified in figure 1 as takes after:

Step 1: Client "A" sends a demand to Name Node. The ask for incorporates the need to duplicate the information documents to Data Nodes. It replays with the IP address of Data Nodes. In the above chart Name Node answers with the IP address of five hubs (DN1 to DN5).

Step 2: Client "A" gets to the crude information for control in Hadoop. Its configurations the crude information into HDFS arrange what's more, partitions pieces in view of the information estimate. In the above case the pieces B1to B4 are dispersed among the Data Nodes.

Step 3: Client "A" sends the three duplicates of every information piece to various Data Nodes.

Step 4: In this progression, customer "A" sends a Map Reduce work (job1) to the Job Tracker daemon with the source information document name(s).

Step 5: Job Tracker sends the errands to all Task Trackers holding the pieces of the information. Each Task Tracker executes a particular assignment on each piece and sends the outcomes back to the Job Tracker.

Step 6: job Tracker sends the last outcome to Client "A". On the off chance that customer "A" has another employment that requires the same datasets it rehashes the set 6-8.

Step 7: In local Hadoop customer "B" with another Map Reduce work (job2) will experience step 1-5 regardless of the possibility that the datasets are as of now accessible in HDFS. In any case, in the event that customer "B" realizes that the information exists in HDFS, it will send job2 specifically to Job Tracker.

Step 8: Job Tracker sends job2 to all Task Trackers. It execute the assignments and send the outcomes back to the Job Tracker. It sends the last outcome to Client "B". Figure.3, demonstrates the work process graph for Native Hadoop.

Module Description:

Data Preprocessing Module: Information preprocessing is a capable apparatus that can empower the client to treat and process complex information, it might expend a lot of handling time.

In this module we need to make Data set for bank dataset it contain set of table to such an extent that cust table elements, account points of interest, exchange points of interest general imprints subtle elements for a year ago and this information first give in MySQL database help of this dataset we investigation this venture.

Data Migration Module with Sqoop: Sqoop is an order line interface application for exchanging information between social databases and Hadoop Apache Sqoop (SQL-to-Hadoop) is a lifeline for any individual who is encountering challenges in moving information from the information stockroom into environment. Apache Sqoop is a successful hadoop device utilized for bringing in information from RDBMS resembles MySQL, Oracle, and so on into HBase, Hive or HDFS.In this module we get the dataset into hadoop (HDFS) using sqoop Tool. Using sqoop we have to perform package of the limit, to such a degree, to the point that if we have to get the particular area or in case we have to carry the dataset with specific condition that will be support by Sqoop Tool and data will be secured in hadoop (HDFS).

Data Analytic Module with Hive: Hive is an information product house framework for Hadoop. It runs SQL like inquiries called HQL (Hive question dialect) which gets inside changed over to delineate occupations. Hive was created by Facebook. Hive underpins Data definition Language (DDL), Data Manipulation Language (DML) and client characterized capacities.

Other features of Hive include:

- Different storage types such as plain text, RCFile, HBase, ORC, and others.
- Metadata storage in an RDBMS significantly
- Reducing the time to perform semantic checks during query execution.
- SQL-like queries (HiveQL), which are implicitly converted into Map Reduce

In this module we need to examination the dataset utilizing HIVE apparatus which will be put away in hadoop (HDFS). For investigation dataset HIVE utilizing HQL Language. Utilizing hive we perform Tables manifestations, joins, Partition, Bucketing idea. Hive investigation the main Structure Language.

Data Analytic Module with Pig: Apache Pig is an abnormal state information stream stage for execution Map Reduce projects of Hadoop. The dialect for Pig will be pig Latin. Pig handles both structure and unstructured dialect. It is additionally top of the guide lessen handle running foundation. In contrast with SQL, Pig

- utilizes lethargic assessment,
- utilizes separate, change, stack (ETL),
- It can store information anytime amid a pipeline.

In this module also used for analyzing the Data set through Pig using Latin Script data flow language.in this also we are doing all operators, functions and joins applying on the data see the result.

Data Analytic Module with Map Reduce: Map Reduce is a handling system and a program show for conveyed registering in view of java. The Map Reduce calculation contains two imperative errands, to be specific Map and Reduce. In this module likewise utilized for investigating the informational collection utilizing MAP REDUCE. Delineate Run by Java Program.

3. CONCLUSION

A Method for predicting the future transactions in banking have been developed by using H2Hadop. The proposed system was to analysis the bank database for the overall transactions and ensure that to provide the performance improvement.

Future Enhancement: In future, we can implement the concept using the spark tool. Spark tool can run in-memory on the cluster. It can run as a standalone or on top of H2Hadoop YARN. It can read data directly from the HDFS (Hadoop Distributed File System).

Spark can run a program up to 100x faster in memory or 10x faster on disc than H2Hadoop. Spark introduces the concept of an RDD (Resilient Distributed Dataset) which is a distributed collections of object operated in parallel. RDDs support two types of operation are Transformations, Actions.

REFERENCES

Amoud Alshammari, Jeongkyu Lee and Hassan Bajwa, H2Hadoop, Improving Hadoop Performance using the Metadata of Related Jobs, 2016.

Changqing J, Big Data Processing in Cloud Computing Environments, in Pervasive Systems, Algorithms and Networks (ISPAN), 2012 12th International Symposium, 2012.

Chen M, Mao S and Liu Y, Big Data, A Survey Mobile Networks and Applications, 2014.

Farrahi K and Gatica-Perez D, A probabilistic approach to mining mobile phone data sequences, Personal Ubiquitous Comput, 2014.

Jagadish H, Big data and its technical challenges, Communications of the ACM, 2014.

Locality-Aware Reduce Task Scheduling for Map Reduce, Mohammad Hammoud and Majd Sakr F, 2011

Lohr S, The age of big data, New York Times, 2012.

Marx V, Biology, The big challenges of big data, Nature, 2013.

Ming M, Jing G and Jun-jie C, Blast-Parallel, The parallelizing implementation of sequence alignment algorithms based on Hadoop platform. InBiomedical Engineering and Informatics (BMEI), 6th International Conference on. 2013.

Nishanth S, Radhikaa B, Ragavendar T J, Chitra Babu, and Prabavathy B, CoRadoop++ A Load Balanced Data Co location in Radoop Distributed File System, 2013.

Radha Dobale G, Sonar R.P, Review of Load Balancing for Distributed Systems in Cloud, 2015.

Schadt E.E, Computational solutions to large-scale data management and analysis, Nature Reviews Genetics, 2010.

Schatz M.C, Langmead B and Salzberg S.L, Cloud computing and the DNA data race, Nature biotechnology, 2010.