ISSN:0975-9646

Hadoop Based Analytics on Next Generation Medicare System

Gopal A. Tathe[1], Pratik S. Patil[2], Sangram C. Parle[3],Vishal S. Pathare[4],Prof. Sudarshan S. Deshmukh[5]

Department of Computer Engineering,
Pimpri Chinchwad College of Engineering, Pune, India

Abstract: **The traditional medical system is the field which is not centralized and digitalized. Due to which society has to face immense problems. The problems are caused because of direct interaction between Doctor, Pharmaceutical Store and Medicine Manufacturing Company. Due to this interaction, even though generic medicines are available, doctors suggests the medicines which are not economical. In most of the cases medicines are not available because of decentralized stock management by Pharmaceutical Store. Furthermore, there is no system available which cannot help Government to detect the disease spreading in particular area. As a whole there has to be a decision support system for these three actors for the benefits of society. Hence we are proposing a centralized and digitalized system in which communication between Doctor, Pharmaceutical Store and Medicine Manufacturing Company is performed through a central Decision making server which acts as collaborative centre for hosting all the data from given actors. With the help of this databases we can analyse different situations like region wise sells of medicine, region wise disease analysis etc. Thus our idea aims at solving the problems with the help of DSS based on Distributed environments. For this Decision Support System we will be using Hadoop frame work. Big Data is high volume, velocity and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making. Hadoop is the core platform for structuring Big Data, and solves the problem of making it useful for analytics purposes. Using Hadoop we will be discovering that important predictions can be made by sorting through and analysing Big Data.**

Keywords: ***Decision Support System (DSS), Distributed Environment, Hadoop, Big data***

I. INTRODUCTION

Scientific breakthrough and advancement in medical knowledge have helped to treat conditions which were not curable even a few years back but we are far from our goals of improving the quality of life and to save all the lives that could be saved. Hospitals are striving towards these goals but are challenged to effectively use all their resources – financial, human, information and materials for providing better care, at all times, at affordable costs to all[5].

The health care industry is facing critical questions today such as how can we effectively manage hospitals and provide enhanced services? How can we provide care in a cost-efficient manner? How do you use your resources efficiently to enhance quality of care? While dealing with these challenges, we need to curtail hospital management costs, ensure timely availability of information, improve quality, and extend reach. The traditional medical system is the field which is not centralized and digitalized. Due to which society has to face immense problems. The problems are caused because of direct interaction between Doctor, Pharmaceutical Store and Medicine Manufacturing Company. Due to this interaction, even though generic medicines are available, doctors suggests the medicines which are not economical. In most of the cases medicines are not available because of decentralized

Stock management by pharmaceutical store. Furthermore, there is no system available which cannot help Government to detect the disease spreading in particular area. As a whole there has to be a decision support system for these three actors for the benefits of society. Hence we are proposing a centralized and digitalized system in which communication between doctor, pharmaceutical store and medicine manufacturing company is performed through a central decision making server which acts as collaborative centre for hosting all the data from given actors. With the help of this databases we can analyse different situations like region-wise sale of medicine, region-wise disease analysis etc.

Thus our idea aims at solving the problems with the help of DSS based on Distributed environments. For this Decision Support System we will be using Hadoop frame work. Big Data is high volume, velocity and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making. Hadoop is the core platform for structuring Big Data, and solves the problem of making it useful for analytics purposes. Using Hadoop we can discover important predictions that can be made by sorting and analysing Big Data[3].

II. HDFS DISTRIBUTED STORAGE MECHANISM

HDFS has master/slave architecture. An HDFS cluster consists of a single name node runs on the master server that manages the file system namespace and regulates access to files by clients, and a number of data nodes run on the Slave servers which manage storage attached to the nodes that they run on, The name node and data node are pieces of software designed to run on commodity machines.

HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of more blocks and these blocks are stored in a set of operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to data nodes. The data nodes are responsible for serving read and write requests from the file system's clients. The data nodes also perform block creation, deletion, and replication upon instruction from the name node. The name node makes all decisions regarding replication of blocks. It periodically receives a Heartbeat and a Block report from each of the data nodes in the cluster. A block report contains a list of all blocks on a data node [7].

III. Big Tables Storage Mechanism of Hbase

HDFS provides high fault-tolerant distributed storage, based on HDFS, HBase implemented the column-oriented sparse table storage for big data.

A. HBase Table

HBase stored data in table, the table consists of rows and columns, the column is divided into a number of column families[3], and HBase table has the following characteristics:

Large size: a table can contain billions of rows, millions of columns.

Column-oriented: column (family) is the unit for store and access control, table is queried by column (family).

Sparse: The empty (null) columns do not take up any storage space, so the table can be designed very sparse.

Scalability: The existing table schema can be changed (adding and removing column families).

Row Key: Row key is the table primary key. Hbase maintains data in lexicographic order by row key and every read or write of data under a single row key is atomic.

Column Family: Columns are grouped into sets called column families. A column family must be created before data can be stored under any column in that family. A table may have an unbounded number of columns[3]. Access control and both disk and memory accounting are performed at the column-family level. In practice, these controls allow us to manage several different types of applications: some that add new base data, some that read the base data and create derived column families, and some that are only allowed to view existing data.

Timestamps:

In HBase, cell is the storage unit which uniquely identify by the {row key, column (= <family> + <label>), by the {row key, column (= <family> + <label>), versions of the same data; these versions are indexed by timestamp. Timestamps can be assigned by HBase, or be explicitly assigned by the user when it is inserted.

B. Apache Phoenix

Apache Phoenix is a relational database layer over HBase delivered as a client-embedded JDBC driver targeting low latency queries over HBase data. Apache Phoenix takes your SQL query, compiles it into a series of HBase scans, and orchestrates the running of those scans to produce regular JDBC result sets. The table metadata is stored in an HBase table and versioned, such that snapshot queries over prior versions will automatically use the correct schema. Direct use of the HBase API, along with coprocessors and custom filters, results in performance on the order of milliseconds for small queries, or seconds for tens of millions of rows.

Phoenix provides a JDBC driver that hides the intricacies of the NoSQL store enabling users to create, delete, and alter SQL tables, views, indexes, and sequences; upsert and delete rows singly and in bulk; and query data through SQL. Phoenix compiles queries and other statements into native NoSQL store APIs rather than using MapReduce enabling the building of low latency applications on top of NoSQL stores.

C. Map Reduce

Map-Reduce is commonly used to refer to both a programming model for Bulk Synchronous Parallel Processing, as well as a computational infrastructure for implementing this programming model. From the infrastructure point of view, a Map-Reduce job has three phases: While many good descriptions of Map-Reduce exist, we still would like to present a description since one of the phases (shuffle) is typically given less attention, and this phase is going to be crucial in our complexity measures and in the distinction that we draw with PRAM.

Map: In this phase, a User Denied Function (UDF), also called Map, is executed on each record in a given le. The le is typically striped across many computers, and many processes (called Mappers) work on the le in parallel. The output of each call to Map is a list of {KEY, VALUE} pairs.

Shuffle: This is a phase that is hidden from the programmer. All the {KEY, VALUE} pairs are sent to another group of computers, such that all {KEY, VALUE} pairs with the same KEY go to the same computer, chosen uniformly at random from this group, and independently of all other keys. At each destination computer, {KEY, VALUE} pairs with the same KEY are aggregated together. So if {x,y1},{x,y2},...,{hx,yK} are all the key-value pairs produced by the Mappers with the same key x, at the destination computer for key x, these get aggregated into a large {KEY, VALUE} pair {x,{y1,y2,...,yK}}; observe that there is no ordering guarantee. The aggregated {KEY, VALUE} pair is typically called a Reduce Record, and its key is referred to as the Reduce Key.

Reduce: In this phase, a UDF, also called Reduce, is applied to each Reduce Record, often by many parallel processes. Each process is called a Reducer. For each invocation of Reduce, one or more records may get written into a local output.

IV. Proposed Medical Care System

We are proposing a centralized and digitalized system in which communication between Doctor, Pharmaceutical Store and Medicine Manufacturing Company is performed through a central Decision making server which acts as collaborative centre for hosting all the data from given actors. With the help of this databases we can analyse different situations like region wise sells of medicine, region wise disease analysis etc. Thus our idea aims at solving the problems with the help of DSS based on Distributed environments. The proposed system consist of four systems viz. doctor node, pharmaceutical company node, medical store node and centralised decision making node . Out of which one, centralised decision making node, will act as master (Name node & Data node) and remaining three will act as its slaves (Only Data nodes).

Doctor node will contain SQL information table about the patients from all regions. This table consist of information like Doctor ID, patient's personal information, information about disease, and the prescribed medicines along with the date. Pharmaceutical node will store information about the medicines that are produced by different pharmaceutical companies, the quantity of those medicines produced along with the price fixed by the pharmaceutical company which will also be a SQL table. Similarly, medical node will store information about medicines stock in medical stores.

The centralized node will retrieve the information from these SQL table which will be required for analytical part, and will store it in Hbase table at centralised node. Here, we are using Apache Phoenix on the top of Hbase, as Hbase is NoSQL, by using apache phoenix we will be able to retrieve data from Hbase table using SQL queries, and for normal transactions Apache phoenix gives faster performance than Hbase.

On the centralised Hbase table there will be different MapReduce programs running on different columns to get different results of analytics. These results are as follows:

a. Percentage usage of Generic medicine by a particular doctor.

b. Region wise medicine requirements.

c. Analysis of diseases in a particular region for a given period.

For example, suppose if we want to know how many patients were diagnosed with Swine flu on 1 January then we will apply MapReduce program on Hbase table for two columns i.e. disease name i.e. Swine flu and date i.e. 1 January and fetch those records which matches our requirement and store them into new table and then we will display the result using graphs and charts.

V. CONCLUSION

Hadoop based analytics for the medical system will help us to take the actions related to the region wise disease detection as earliest as possible which will directly help the governing body to take effective and immediate actions and save the lives. This system will help the medical stores and pharmaceutical companies to keep their stocks up to date. Using this system we can recommend the Doctors to use of Generic medicines which will help the people who can't afford the branded medicines.

REFERENCE

[1] Zhou Yinan, Wang Yu. HadoopFile System Performance Analysis [J]. Electronic teachnology, 15-16.

[2] HDFS Architecture[OL]. http://hadoop.apache.org/

[3] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach,M. Burrows, T. Chandra, A. Fikes, and R. Gruber. Bigtable: A Distributed Storage System for Structured Data. In OSDI,205–218. USENIX Association, 2006

[4] Qu Baoli. The Construction of Electronic Health Records in Regional Informatization[J]. Journal of Medical Informatics.2009,(4):13-15

[5] Technology solution for maternal and child health information system based on regional health information platform [OL]. http://61.49.18.65/publicfiles/business/cmsresources/mohbgt/cmsrsdocument/doc9985.pdf