PUNE INSTITUTE OF COMPUTER TECHNOLOGY

DHANKAWADI, PUNE

DSBDAL CASE STUDY

ON

Health care systems with Hadoop Ecosystem components

Submitted by

Pratik Gokhale (31310)

Isha Maheshwari (31312)

Under the guidance of

Prof. V. V . Bagade

Problem Statement :

Write a case study to process data driven for Health care systems with Hadoop Ecosystem components as shown.

● HDFS: Hadoop Distributed File System

● YARN: Yet Another Resource Negotiator

● MapReduce: Programming based Data Processing

● Spark: In-Memory data processing

● PIG, HIVE: Query based processing of data services

● HBase: NoSQL Database (Provides real-time reads and writes)

● Mahout, Spark MLLib: (Provides analytical tools) Machine Learning algorithm libraries

Solar, Lucene: Searching and Indexing

Learning Objectives :

By performing this case study, we shall be able to:

● HDFS: Hadoop Distributed File System

● YARN: Yet Another Resource Negotiator

● MapReduce: Programming based Data Processing

● Spark: In-Memory data processing

● PIG, HIVE: Query based processing of data services

● HBase: NoSQL Database (Provides real-time reads and writes)

● Mahout, Spark MLLib: (Provides analytical tools) Machine Learning algorithm libraries

● Solar, Lucene: Searching and Indexing

above components of Hadoop ecosystem in Health care system.

Software and Hardware Requirements :

Software:

- Windows 10 OS, 64 bits
- Hadoop

Hardware:

- Processor: Intel i-5 8th gen
- Manufacturer: Lenovo
- Ram: 8 GB/ 16GB Optane memory

Concepts related Theory :

Healthcare Industry is one of the world's greatest and most extensive ventures. Amid, the ongoing years the Healthcare administration around the globe is changing from infection focused to a patient-focused model and volume-based model. Teaching the predominance of Healthcare and diminishing the cost is a guideline behind the creating development towards-esteem based Healthcare conveyance model and patient-focused mind. The volume and interest for huge information in Healthcare associations are developing little by close to nothing. To give successful patient-focused care, it is fundamental to oversee and analyse the huge amount of data sets. The traditional methods are obsolete and are not sufficiently adequate to break down enormous information as assortment and volume of information sources have expanded and a very large rate in previous two decades. There is a requirement for new and creative tools and methods that can meet and surpass the capacity of overseeing such a huge amount of data being generated by the healthcare department.

The social insurance framework of healthcare departments is community in nature. This is since it comprises of a substantial number of partners such as doctors with specialization in different sectors, medical caretakers, research centre technologists and other individuals that cooperate to accomplish the shared objectives of decreasing medicinal cost and blunders and also giving quality healthcare experience. Every one of these partners produce information from heterogeneous sources, for example, physical examination,

clinical notes, patients' meetings and perceptions, research facility tests, imaging reports, medications, treatments, overviews, bills and protection.

The rate at which information is being generated from heterogeneous sources from various healthcare department has incremented exponentially on the daily basis. Therefore, it is becoming hard to store, process and break down this inter related information with traditional dataset handling applications. Nonetheless, new and efficient methods and systems are in addition to provide great processing advancements to store, process, break down and extricate values from voluminous and heterogeneous medical information being generated in a continuous way. Henceforth, the medicinal services framework is quick turning into a major information industry. Generally, medicinal services information has developed enormously in both organized and unstructured way, to a great extent driven by the requests of always extending information parched populace what's more, operational attributes of e-health stages. This dangerous multi-dimensional development has lead scientists, to add numerous more watchwords to portray Healthcare Big Data (HBD). It isn't only the volume however their assortment, specifically the kinds of sources that deliver information and the objective sorts that request them are excessively different and various in Healthcare area. These incorporate medicinal services workforce (doctors, clinical staff, parental figures), benefit giving organizations (counting safety net providers), healing facilities with resources, clinicians, government controllers, drug stores, pharmaceutical makes (with look into groups included), and therapeutic gadget organizations.

In order to process a huge amount of health data records at once we need efficient tools and methodologies. The proposed papers use the Hadoop Framework to handle the data, and the algorithm being used is Map Reduction ("MapReduce").

HDFS:

The Hadoop Distributed File System (HDFS) is the essential information stockpiling framework utilized by Hadoop applications. It comprises of NameNode/The Master and DataNodes/The Slave design to execute a disseminated record framework called Hadoop Distributed File System to get to information crosswise over exceedingly adaptable Hadoop Clusters in an effective way. Hadoop Framework in total consists of 5 daemon processes namely:-

1. NameNode: NameNode is utilized to store the Metadata (data about the area, size of files/blocks) for HDFS. The Metadata could be put away on RAM or Hard-Disk. There will dependably be just a single NameNode in a cluster. The only way that the Hadoop framework can fail is when the NameNode will crash.
2. Secondary NameNode: It is used as a backup for NameNode. It holds practically same data as that of NameNode. On the off chance that NameNode falls flat, this one comes into picture.

3. DataNode: The actual user files or data is stored on DataNode. The number of DataNode depends on your data size and can be increased with the need. The DataNode communicates to NameNode in definite interval of times.
4. Job Tracker: NameNode and DataNodes store points of interest and genuine information on HDFS. This information is likewise required to process according to users' prerequisites. A Developer writes a code to process the information. Processing of data can be done using MapReduce. MapReduce Engine sends the code over to DataNodes, making jobs in multiple nodes running alongside of each other. These employments are to be persistently observed by the Job tracker.
5. Task Tracker: The Jobs taken by Job Trackers are in genuine performed by Task trackers. Each DataNode will have one task tracker. Task trackers communicate with Job trackers to send statuses of the undertaken job status.

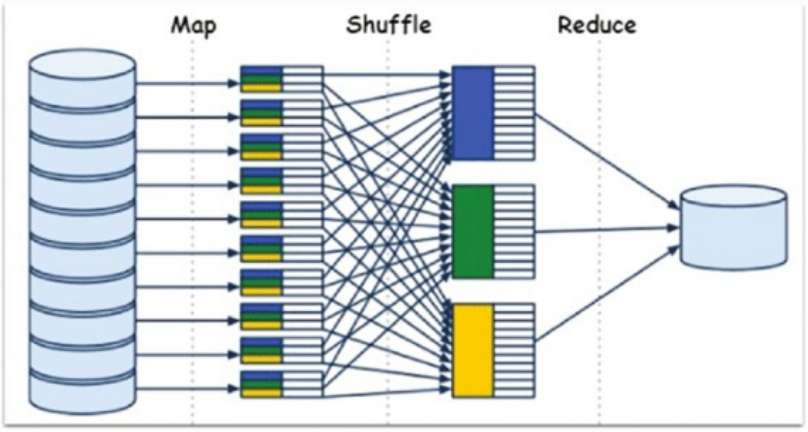
HDFS can be used to maintain the data in a central database as opposed to having it stored individually over several computers.

MapReduce:

Map Reduction algorithm contains two important tasks, namely Map and Reduce.

- Mapping – Attained by Mapper Class
- Reduction – Attained by Reducer Class.

MapReduce utilizes different numerical calculations to separate an errand into little parts and dole out them to various frameworks. MapReduce calculation helps in sending the Map and Reduce errands to proper servers in a bunch. The tasks are executed in parallel in all the different nodes and finally the result is returned to the user. Hadoop uses MapReduce algorithm to create tasks, called jobs which can be executed independently on different clusters (DataNodes) while the result is fetched back to a single node (NameNode) for output. MapReduce is important for several data analysis purposes based on patients' data for better prescriptive analysis in the future.



Apache Spark:

Spark is an open-source cluster computing platform which is widely used for data processing in Hadoop ecosystems. In our framework, Spark is selected among other existing tools due to three major characteristics required in sensing-based healthcare: first, it supports both batch and streaming processing which are necessary to apply various data analytical algorithms; second, it ensures a lower latency level than other tools such as MapReduce, which is strongly required in health applications; and third it guarantees scalability to any number of cluster nodes required for the health application requirements. Therefore, in our platform, we implemented Spark on the master cluster node in order to receive data streaming from Kafka, perform processing, and send data to the Hadoop HDFS for storing purpose. Subsequently, data processing is done via two created scripts, one for the real time decision and other for patient archiving.

Hive:

Hive is software used for data warehousing implemented on top of Hadoop HDFS in order to provide data query and analysis. In addition, Hive allows creating a metadata storage in the form of tables in a relational database system. This makes our platform more efficient in terms of reducing the access time to the patient archive. On the other hand, it helps the medical staff to keep track of the criticality of patients in real-time. In our platform, we installed Hive on the cluster master node and we created an external table located in the HDFS main directory, where the medical staff can explore the data imported to HDFS using HiveQL console. To retrieve the data, they have to write HiveQL queries depending on their requirements.

NoSQL:

Several hospitals use the Hadoop ecosystem’s NoSQL database to collect and manage their huge amounts of real-time data from diverse sources related to

patient care, finances, and a payroll, which helps them identify high-risk patients while also reducing day-to-day expenditures.

MLLib:

Apache Spark's Machine Learning Library (MLlib) is designed for simplicity, scalability, and easy integration with other tools. With the scalability, language compatibility, and speed of Spark, data scientists can focus on their data problems and models instead of solving the complexities surrounding distributed data. MLLib can support decision tree implementation for disease predictions.

HBase:

A storage and processing architecture can be built based on MapReduce and HBase, respectively. First, data generated by health sensors are inserted into HBase using a mass insertion script, and then a data analysis algorithm is proposed in order to retrieve valuable data and help in predicting disease. the authors proposed a Hadoop-based framework in order to secure the transmitted data from biosensors to the server.

Thus, the Hadoop ecosystem plays a significant role in healthcare informatics and greatly influences the healthcare system and the big data four Vs in healthcare. The combination of big data and healthcare analytics can lead to treatments that are effective for specific patients by providing the ability to prescribe appropriate medications for each individual, rather than those that work for most people. As we know, big data analytics is in the early stage of development and current tools and methods cannot solve the problems associated with big data. Big data may be viewed as big systems, which present huge challenges. Therefore, a great deal of research in this field will be required to solve the issues faced by the healthcare system.

Conclusion:

Through the medium of the case study we covered all the Hadoop Ecosystem Components in detail. We can confidently say these Hadoop ecosystem components empower Hadoop functionality in the Healthcare system and will lead to huge improvements in the future.