Asian Journal of Convergence in Technology **Volume V Issue II**
ISSN NO: 2350-1146 I.F-5.11

Healthcare Data Management and Analysis

Yogita Kale [1]
[1]*Student, Computer Engineering Department, DYPCOE Ambi, Maharashtra, India*

……..***……

Abstract - The Electronic Medical Records (EMRs) are the primary sources to study the enhancement of health and medical care. The rapid development in science and medical technology has produced various methods to detect, verify, prevent, and treat diseases. This has led to the generation of big health-care data and difficulties in processing and managing data. To capture all the information about a patient and to get a more detailed and complete view for insight into care coordination and management decisions big data technologies can be used. A more detailed and complete picture about patients and populations can be identified along with patients at risk before any health issue arises. Optimal strategies to commercialize treatments and the next generation of health care treatments can be identified and developed by it.

Key Words: **Hadoop, Health Care Analysis, Big data, Map Reduce, HDFS, Apache Spark, Apache Hive, Tableau.**

1. Introduction

Historically, large amount of data, driven by record keeping, compliance and regulatory requirements and patient care has been generated by the healthcare industry. The current trend is toward rapid digitization of these large amounts of data, while most data is stored in hard copy form. The big data has the potential to improve the quality of healthcare and on the other hand to reduce the costs. It assures to support wide range of medical and healthcare functions, disease surveillance and population health management. Health care needs to be modernized with the new era of big data, this includes the health care data to be properly analyzed so that we can deduce in which group or regions or age or gender diseases attack the most. Distributed processing, Hadoop can be used for the computation of this gigantic size of analytics. Map Reduce is a popular paradigm in computing for large-scale data processing in cloud computing. However, the slot-based Map Reduce system can suffer from poor performance due to its unoptimized resource allocation. To solve this problem the resource allocation is optimized by the framework in this paper. Many times slots can be severely under-utilized due to the static pre-configuration of distinct map slots and reduce slots which are not fungible. The reason of this is map slots might be fully utilized on the other hand reduce slots may remain empty, and vice-versa. To overcome this problem, we propose an alternative technique which is Dynamic Hadoop Slot Allocation by keeping the slot-based allocation model. It relaxes the slot allocation and depending on their needs allows slots to be reallocated to either map or reduce tasks. Getting the health care analysis in various forms are multipurpose beneficial outputs which will be provided by the framework.

2. LITERATURE SURVEY

In the following we examine some of the reasons why we need the Hadoop technology and his system for the efficient use and management of health care records. In this paper [1] Large amount of data driven by record keeping, compliance & regulatory requirements and patient care is generated by healthcare industries, historically. The current trend is toward rapid digitization of these large amounts of data, though most of the data is stored in hard copy form. Driven by mandatory requirements and the potential to improve the quality of healthcare delivery on the other hand reducing the cost, these huge amounts of data (called as “Big data”) hold the promise of supporting various medical and healthcare functions, including among others clinical decision support, disease surveillance, and management of population health. Reports say in 2011, data from the U.S. healthcare system alone reached, 150 exabytes. Big data for U.S. healthcare will soon reach the zettabyte (1021 gigabytes) scale and, not long after, the yottabyte (1024 gigabytes), at this growth rate. Kaiser Permanente, the California-based health network, which has more than 9 million members, is believed to have between 26.5 and 44 petabytes of potentially rich data from EHRs, including images and annotations. By definition, big data in healthcare refers to electronic health data sets so large and complex that they are difficulty (or impossible) to manage with traditional software and/or hardware; nor they can be easily managed with traditional or common data management methods and tools. Not only because of the diversity of data types and the speed at which it must be managed but also because of its volume big data is overwhelming. In this paper [2] A lot of challenges in terms of data transfer, storage, computation and analysis has been brought by the exponential evolution of data in health care. Ample patient information and historical data, which enclose rich and significant insights that can be exposed using

advanced tools and techniques as well as latest machine learning algorithms for healthcare usage and applications. Though, new big data analytics framework is required for the size and rapidity of such great dimensional data. To show the impact of big data this paper introduces the thought of data in healthcare and the results of various surveys. Some case studies of big data analytics in healthcare are presented. The term "Big data" become popular in last few years, as it represents the hard work of researchers to achieve business intelligence by processing tremendously large amount of data. For typical dataset software tools, it is very difficult to collect, store, manage and analyze. Of course, big data is too large to load into memory and store on a hard-drive and fit in a standard database. In this paper [3] Now-a-days, for large-scale data processing in clusters and data centers MapReduce has become a popular high-performance computing paradigm. Hadoop, an open source implementation of MapReduce, has been deployed in large clusters containing thousands of machines by companies such as Yahoo! and Facebook to support batch processing for large jobs submitted by multiple users (i.e., MapReduce workloads). In this paper [4] Functionality to a Consultant Physician in Geriatric Medicine in terms of storing and analyzing high quality clinical patient data for the purpose of more informed and accurate decision making is provided by the Patient Data Analysis Information System (PDA-IS). To support the Consultant Physician in improving the quality of healthcare delivery is the system's general aim.

3. Proposed System

To deal with big /large amount of data we are going to implement Hadoop ecosystem that overcome all above drawbacks of RDBMS that occur when big data comes in the picture. The medical records are gathered from various organizations and doctors. The data gathered is sent for preprocessing from which patient's wise disease, patient's general information; information about the disease and the survival status is extracted. This data is known as training data. The preprocessed data is given for data mining where for data mining technique is used. After preprocessing is done data is sent to perform analysis clustering. To show clearer picture of the analysis algorithm is applied to the resultant data once the analysis is done. Then the statistics can be represented in various forms and groups.

4. SYSTEM ARCHITECTURE

The medical records are gathered from various organizations and doctors. The data gathered is send for pre-processing from which patient's wise disease, patient's general information; information about the disease and the survival status is extracted. This data is known as training data. The pre-processed data is given for data mining, where for mining the data mining technique is used. After pre-processing is done data is sent to perform analysis clustering. To show clearer picture of the analysis algorithm is applied to the resultant data once the analysis is done. Then the statistics can be represented in various forms and groups.

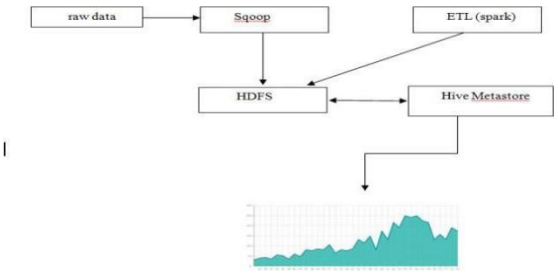


Fig.1 System Architecture

HDFS (Hadoop Distributed File System) - The Hadoop Distributed File System (HDFS) is a distributed file system providing fault tolerance and designed to run on commodity hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. Hadoop provides a distributed file system (HDFS) that can store data across thousands of servers, and a means of running work across those machines, running the work near the data. HDFS has master/ architecture. Large data is automatically split into chunks which are managed by different nodes in the Hadoop cluster.

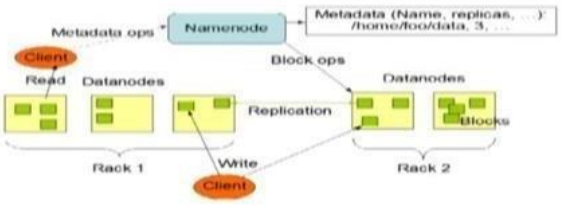


Fig.2 HDFS (Hadoop Distributed File System)

Apache Spark:

Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. It is based on Hadoop MapReduce and it extends the MapReduce model to efficiently use it for more types of computations, which includes interactive queries and stream processing. The main feature of Spark is its in-memory cluster computing that increases the processing speed of an application. Spark is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries, and streaming. Apart from supporting all these workloads in a respective system, it reduces the management burden of maintaining separate tools.

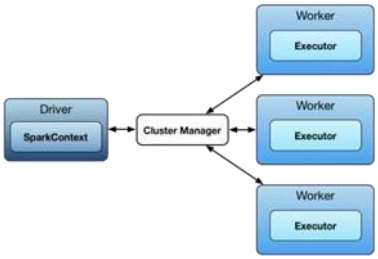


Fig. 3 Spark Architecture

Spark is one of Hadoop's sub project developed in 2009 in UC Berkeley's AMPLab by Matei Zaharia. It was Open Sourced in 2010 under a BSD license. It was donated to Apache software foundation in 2013, and now Apache Spark has become a top -level Apache project from Feb-2014.

Apache Spark has following features.

- **Speed** − Spark helps to run an application in the Hadoop cluster, up to 100 times faster in memory, and 10 times faster when running on disk. This is possible by reducing number of read/write operation ns to disk. It stores the intermediate processing data in memory.

- **Supports multiple languages** − Spark provides built- in APIs in Java, Scala, or Python. Therefore, you can write applications in different languages. Spark comes up with 80 high-level operators for interactive querying.

- **Advanced Analytics** − Spark not only supports 'Map' and 'reduce'. It also supports SQL queries, Streaming data, Machine learning (ML), and Graph algorithms.

- ***Hive Architecture***

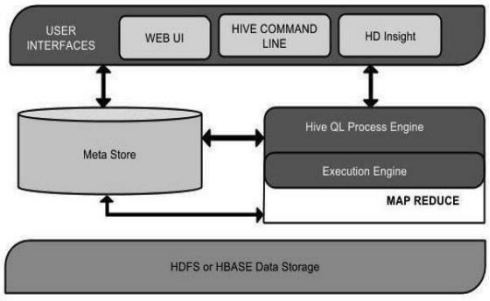


Fig. 4 Hive Architecture

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

Hive is not

- A relational database
- A design for Online Transaction Processing (OLTP)
- A language for real-time queries and row-levelupdates

Features of Hive:

- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.
- It provides SQL type language for querying called HiveQL or HQL.
- It is familiar, fast, scalable, and extensible.

Design and Modules:

1. Data (JSON) file importer:

Input data is a raw data coming from various resources it may contain records in text, csv or json format, using scoop tool we can transfer data from RDBMS to HDFS and vice versa, using hive we can directly import files into hive tables.

2. ETL [Spark to extract, transform and load data]:

Spark is the framework using it, we can apply regular expressions, create dataframes and load into the main hive tables.

3. Hive Metastore:

Hive metastore usage the Mysql RDBMS to store hivemetastore (Metdata) and actual data stores at HDFS.

4. Tableau Visualization:

Tabelau is a tool for visualization, is the tool which data scientist and analyst uses for the visualization, we represents the data into graphical representation , data require for the Tableau is imported from the hive data warehouse , the graphical representation of data is shown on Tableau dashboard and makes querying and analyzing easy.

5. SYSTEM ANALYSIS

Apache Spark vs Hadoop:

Performance: Spark is fast because it has in-memory processing. It can also use disk for data that doesn't all fit into memory. Spark's in-memory processing

delivers near real- time analytics. This makes Spark suitable for credit card processing system, machine learning, security analytics and Internet of Things sensors. Hadoop was originally setup to continuously gather data from multiple sources without worrying about the type of data and storing it across distributed environment. MapReduce uses batch processing. MapReduce was never built for real-time processing, main idea behind YARN is parallel processing over distributed dataset. The problem with comparing the two is that they perform processing differently.

Ease of Use: Spark comes with user-friendly APIs for Scala, Java, Python, and Spark SQL. Spark SQL is very similar to SQL, so it becomes easier for SQL developers to learn it. Spark also provides an interactive shell for developers to query & perform other actions, & have immediate feedback. You can ingest data in Hadoop easily either by using shell or integrating it with multiple tools like Sqoop, Flume etc. YARN is just a processing framework and it can be integrated with multiple tools like Hive and Pig. HIVE is a data warehousing component which performs reading, writing and managing large data sets in a distributed environment using SQL-like interface. You can go through this Hadoop ecosystem blog to know about the various tools that can be integrated with Hadoop.

Costs: Hadoop and Spark are both Apache open source projects, so there is no cost for the software. Cost is only associated with the infrastructure. Both the products are designed in such a way that it can run on commodity hardware with low TCO. Now you may be wondering the ways in which they are different. Storage & processing in Hadoop is disk- based & Hadoop uses standard amounts of memory. So, with Hadoop we need a lot of disk space as well as faster disks. Hadoop also requires multiple systems to distribute the disk I/O. Due to Apache Spark's in memory processing it requires a lot of memory, but it can deal with a standard speed & amount of disk. As disk space is a relatively inexpensive commodity and since Spark does not use disk I/O for processing, instead it requires large amounts of RAM for executing everything in memory. Thus, Spark system incurs more cost. But yes, one important thing to keep in mind is that Spark's technology reduces the number of required systems. It needs significantly fewer systems that cost more. So, there will be a point at which Spark reduces the costs per unit of computation even with the additional RAM requirement.

Data Processing:

Batch Processing vs Stream Processing

Batch Processing: Batch processing has been crucial to big data world. In simplest term, batch processing is working with high data volumes collected over a period. In batch processing data is first collected and then processed results are produced at a later stage. Batch processing is an efficient way of processing large, static data sets. Generally, we perform batch processing for archived data sets. For example, calculating average income of a country or evaluating the change in e-commerce in last decade.

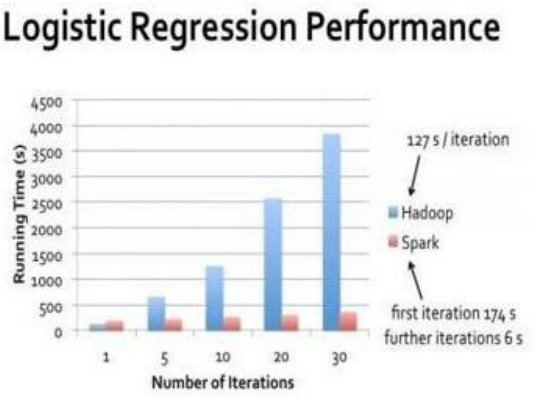
Stream processing: Stream processing is the current trend in the big data world. Need of the hour is speed and real- time information, which is what steam processing does. Batch processing does not allow businesses to quickly react to changing business needs in real time, stream processing has seen a rapid growth in demand. Now coming back to Apache Spark vs Hadoop, YARN is a basically a batch-processing framework. When we submit a job to YARN, it reads data from the cluster, performs operation & write the results back to the cluster. Then it again reads the updated data, performs the next operation & write the results back to the cluster and so on. Spark performs similar operations, but it uses in-memory processing and optimizes the steps. Graph X allows users to view the same data as graphs and as collections. Users can also transform and join graphs with Resilient Distributed Datasets (RDDs).

Fault Tolerance: Hadoop and Spark both provides fault tolerance, but both have different approach. For HDFS and YARN both, master daemons (i.e. Name Node & Resource Manager respectively) checks heartbeat of slave daemons (i.e. Data Node & Node Manager respectively). If any slave daemon fails, master daemons reschedules all pending and in-progress operations to another slave. This method is effective, but it can significantly increase the completion times for operations with single failure also. As Hadoop uses commodity hardware, another way in which HDFS ensures fault tolerance is by replicating data. As we discussed above, RDDs are building blocks of Apache Spark. RDDs provide fault tolerance to Spark. They can refer to any dataset present in external storage system like HDFS, HBase, shared filesystem. They can be operated parallelly. RDDs can persist a dataset in memory across operations, which makes future actions 10 times much faster. If a RDD is lost, it will automatically be recomputed by using the original transformations. This is how Spark provides fault-tolerance.

Security: Hadoop supports Kerberos for authentication, but it is difficult to handle. Nevertheless, it also supports third party vendors like LDAP (Lightweight Directory Access Protocol) for authentication. They also offer encryption. HDFS supports traditional file permissions, as well as access control lists (ACLs). Hadoop provides Service Level Authorization, which guarantees that clients have the right permissions for job submission.

Hadoop MapReduce	Apache Spark
Fast	100x faster than MapReduce
Batch Processing	Real-time Processing
Stores Data on Disk	Stores Data in Memory
Written in Java	Written in Scala

Iterative Machine Learning Algorithms: Almost all machine learning algorithms work iteratively. As we have seen earlier, iterative algorithms involve I/O bottlenecks in the MapReduce implementations. MapReduce uses coarse-grained tasks (task-level parallelism) that are too heavy for iterative algorithms. Spark with the help of Mesos – a distributed system kernel, caches the intermediate dataset after each iteration and runs multiple iterations on this cached dataset which reduces the I/O and helps to run the algorithm faster in a fault tolerant manner.Spark has a built-in scalable machine learning library called MLlib which contains high- quality algorithms that leverages iterations and yields better results than one pass approximations sometimes used on MapReduce



Fast data processing: As we know, Spark allows in-memory processing. As a result, Spark is up to 100 times faster for data in RAM and up to 10 times for data in storage.

Iterative processing: Spark's RDDs allowperforming several map operations in memory, with no need to write interim data sets to adisk.

Near real-time processing: Spark is an excellent tool to provide immediate business insights. This is the reason why Spark is used in credit card's streaming system

6. CONCLUSION

In this paper we are going to give a clear view of the medical record, the pattern extracted from the EMRs, and the results generated by extracting the patterns. These patterns are represented in the graphical and tabular list format. To provide cumulative information, disease and their possible symptoms data is grouped together and analyzed. The system will predict the disease for the symptoms which is provided to the system by us after the analysis is done over it. To show the clearer picture of the analysis algorithm can be applied to the resultant and the grouping can be done. Some grouping categories based on which grouping can be done are Age, Gender, Disease, Region, Survival Status, etc.

ACKNOWLEDGEMENT

When the completion of dissertation report comes to an end, the time comes to acknowledge all persons who have made its success possible. It gives me immense pleasure to express our gratitude to everyone associated directly or indirectly with the successful completion of the seminar report. I would like to take this opportunity to specially thank my guide, Prof. Dhanashree Kulkarni, Department of Computer Engineering, Dr. D Y Patil College of Engineering, Pune, for vesting trust in me. I would like to especially thanks to Dr. Mininath Nighot, Head, Department of Computer Engineering, for inspiring me and providing me all the Internet Lab Facility, which made this seminar work convenient. I would also like to specially thank to Dr. A. A. Pawar, Principal, Dr. Y Patil College of Engineering, and Pune for all required facilities in our M. E. degree course. My thanks are also to all faculty members of my department.

REFERENCES

[1] Shanjiang Tang, Bu-Sung Lee, Bingsheng He, "DynamicMR: A Dynamic Slot Allocation Optimized Framework for MapReduce Clusters: IEEE Transactions, 2013.
[2] Wullianallur Raghupathi and Viju Raghupathi, "Big data analytics in healthcare: promise and potential", Health Information Science and Systems 2014.
[3] Divyakant Agrawal, UC Santa Barbara, Philip Bilstein, Microsoft Elisa Bertino, Purdue Univ. "Big data White pdf", from Nov 2011 to Feb 2012.
[4] International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 7, July 2014).
[5] W. Hersh, "Health-care hit or miss?" Nature, vol. 470, pp.327-329, Feb. 2011.
[6] M.A. Musen and J.H. Bemmel, Handbook of Medical Informatics, HOuten: Bohn Stafleu
[7] E.F. Codd, "A relationalmodel of data for large shared data banks" Column. ACM, vol.13 (6), pp. 377-387, 1970.
[8] NoSQLDatabases, Available: http://www.nosql- database.org/
[9] 10gen. MonogDB, http://www.mongodb.org/
[10] https://en.wikipedia.org/wiki/K-means_clustering
[11] National Health Insurance Research Database, Available: http://nhird.nhri.org.tw/en/index.htm
[12] National Health Insurance Administration, Available: http://www.nhi.gov.tw/english/index.aspx