International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 2 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT1952362

Analysing and Predicting on Diseases using Data Pipeline in Hadoop

Arpna Joshi[1], Chirag Singla[2], Mr. Pankaj[3]
[1,2]Information Technology Department MAIT, Maharaja Agrasen Institute of Technology, New Delhi, India
[3]Assistant Professor, Maharaja Agrasen Institute of Technology, New Delhi, India

ABSTRACT

A data pipeline is a set of conducts that are performed from the time data is available for ingestion till value is obtained from that data. Such kind of actions is Extraction (getting value field from the dataset), Transformation and Loading (putting the data of value in a form that is useful for upstream use). In this big data project, we will simulate a simple batch data pipeline. Our dataset of interest we will get from https://www.githubarchive.org/ that records the health data of US for past 125years. The objective of this spark project will be to create a small but real-world pipeline that downloads this dataset as they become available, initiated the various form of transformation and load them into forms of storage that will need further use.
In this project Apache kafka is used for data ingestion, Apache Spark for data processing and Cassandra for storing the processed result.

Keywords : Big data, Hadoop, Apache Kafka, Apache Spark, Cassandra.

I. INTRODUCTION

Today real-time analytics for text data on large-scale has become important for many business needs. Comparing to traditional data warehouse applications, the real-time analytic are data intensive in nature and require capturing and processing the data efficiently. However, collecting and processing such a large-scale data had introduced new challenges in terms of storage as well as processing time.

A typical real-time data processing of large-scale data requires building a distributed data pipeline for capturing, processing, storing, and analyzing the data efficiently. The real-time processing system should be capable of capturing high rate data from various streaming sources, process the data near real time, and store data into a persistent database. However, the data processing system should provide minimum latency to process a high throughput data in real time which is a challenging job.

Google introduced the Map Reduce paradigm for parallel and distributed execution of an application over the commodity cluster. Several systems had implemented Map Reduce paradigm for parallel and distributed processing of batch data on multiple machines. Apache Hadoop is one of the most well-known implementations of it.

1.1.1 Purpose

Each year, emergency responders assist in millions of critical events across the country, costing billions of dollars. As emergency events, the time it takes for first responders to arrive on scene is critical, with minutes often making the difference between life and death. Because of these factors, standard staffing and resource use is very high to make sure enough

responders are available at any given time. These factors make emergency response an important potential application for optimization based on predictions. Thus, a model that can learn and make predictions on the location, level of emergency, and type of these events would be extremely useful to government and department management in making staffing and resource allocation decisions.

1.1.2 Scope

This project will help in predicting the frequency with which a disease can reoccur in an area.

1.1.3 Objective

Our objective is to create an application which can predict the occurrence of a disease in a particular area at a particular time, and of what type. This application will predict based on historic emergency incidents in a specific region. We frame this application as a supervised learning problem where training examples will be drawn from historic data on emergencies for the region. This will allow our application to learn the incident likelihood over our region of interest which can then be subsequently turned into a prediction of level of emergency in that region.

II. METHODS AND MATERIAL

2.1 Existing System

Real time as well as batch time processing in available in the present scenario. There are many fields in which real time data streaming is done using big data tools. Our aim in this project is to analysis and predict the health data of US and to get desired outcomes.

2.2 Proposed System

The proposed system is based on big data. It uses Mlib of spark for its predictions. This system is a supervised machine learning model which uses 125 years of health data of US containing epi_week, state, location, disease, event, number, from_date, to_date, etc. as its training data. Based on this training, it predicts about the occurrence of disease and if it is hazardous or not.

2.3 Feasibility Study: A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The following feasibilities are considered for the project in order to ensure that the project is available, and it does not have any major obstructions.

2.3.1 Technical Feasibility

- ✓ It gives an idea of how an existing system (software, hardware) can perform and platform the purposed system handled.
- ✓ The environment required in the development of system is VMware and Cloudera Machine.
- ✓ The tools used in Apache Hadoop along with Spark and Cassandra.

2.3.2 Operational Feasibility

The software is operationally feasible with the help of Cloudera-Quickstart-vm-5.13.0-0 as it provides tools to handle the big data and also to analyse the data and make predictions Also, the data for this project was downloaded from http://www.tycho.pitt.edu/. This data provided us the information of 125years of health data of US.

2.3.3 Economic Feasibility

If certain estimated cost for the project is accepted, then we say the system is economically feasible. The system will be developed and operated in the existing hardware and software infrastructure. The system

developed and installed will be good benefit to the organization hence the proposed system is economically feasible. So there is no need of additional hardware and software for the system. All we just required Libraries for software initially installed in a system which is one-time task.

3.1 Data pipeline

This project mainly consists of four components:

1) Distributed messaging System
2) Distributed and Parallel Processing System
3) Distributed Database System

A) Distributed messaging system

Distributed messaging is based on the idea of steady message queuing. Messages are queued asynchronously between client applications and messaging systems. A distributed messaging system provides the benefits of reliability, scalability, and persistence. There exist many distributed messaging system which can also be used as data ingestion system in a real time processing system. Some example of this sort is Apache Flume, Apache Kafka and RabbitMQ.

We used Apache Kafka as a data ingestion system for a real time processing system. Apache kafka is a publish-subscribe messaging system. In the publish - subscribe system, messages are proceeded in a topic. Message producers are called publishers and message consumers are called subscribers. The major terms used in Apache Kafka are:

Topic: Topic is a category that maintains the number of messages.

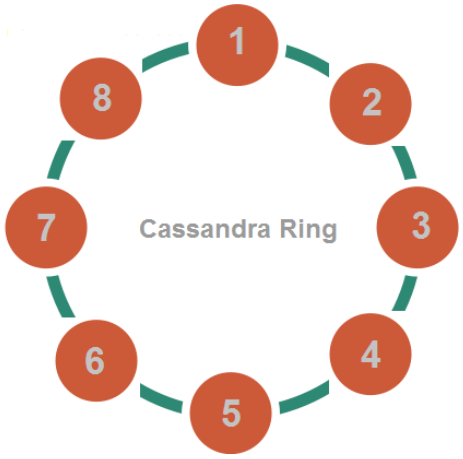
Producer: The application which produces the messages on a topic using Kafka API.

Consumer: The application which reads messages from a topic using Kafka API.

Broker: Broker is a Kafka cluster which consists of multiple nodes.

Apache Kafka internally use Apache Zookeeper to keep a record of various activities across the Kafka cluster.

B) Distributed and Parallel Processing System



As the data is increasing day by day it has become necessary to use distributed and parallel processing system to process and manage the data. There exist many tools that permit us to write parallel and distributed applications such as Apache Hadoop, Apache Spark, and Apache Storm. For this project, we have used Apache Spark as a distributed and parallel processing system for all the processing of data. Apache Spark is in- memory cluster computing framework that was primarily developed to run iterative algorithms based on machine learning. Apache Spark was developed by the University of Berkeley to overcome the limitations of Apache Hadoop. Apache Spark stores all intermediate results in memory rather than storing them on a disk, which makes it 100 times faster than the Apache Hadoop. Apache spark is based on MapReduce and it extends the MapReduce model to adaptively use it for more types of computations. The following diagram shows three ways of how Spark can be built with Hadoop components:

C) Distributed Database System

The NoSQL databases[29] system are non-relational, dis- tributed database system that allows the ad-hoc and fast analysis of high-velocity data with disparate data types. In fact, NoSQL databases system become an alternative of traditional RDMS system with keeping scalability, high availability, and fault tolerance as major key factors. There are a number of NoSQL databases available in the market such as Apache Hbase[8], Apache Cassandra and Mongodb[6]. We use Apache Cassandra as NoSQL distributed database for the real-time system. Apache Cassandra is fully distributed decentralized NoSQL database that provides high availability of data, ease of operations and easy distribution of data across multiple data centers builds on the top of the cluster. Apache Cassandra was initially developed at Facebook for solving slow search in the inbox and later on was converted to open source under Apache in the year 2010. Apache Cassandra architecture makes it easy, scalable and highly available database. Instead of traditional master-slave architecture, Apache Cassandra uses peer to peer distributed architecture, in which each node of the cluster is identical to other nodes in a cluster. The following figure shows the cluster or ring of four Apache Cassandra nodes:

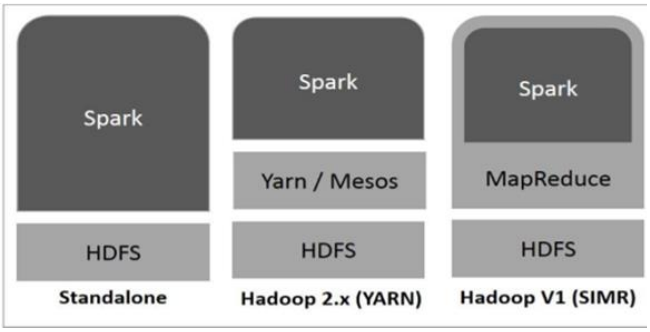


Fig. 2: Apache Cassandra Ring

III. RESULTS AND DISCUSSION

Implementation

Analysing and predicting on diseases using big data pipelines uses various open source big data tools. First the data is extracted from the source using flume and then Kafka is used as data integration system and also works as a distributed message system. Then the data is made into pipelines using spark streaming. The processed data is analysed using Spark sql and then Mlib component of spark is used to make predictions and find whether the disease will reoccur in the particular area or not. The time span after which there are chances of reoccurring of disease can also be predicted.

We didn't used unbounded decision trees as they may so that the classes for the predictions made by decision trees may not have over fitted values. We created a user interface using QT designer and pyqt5, so that this project can be used easily.

IV. CONCLUSION

In this project we have studied use Machine learning library of spark for predicting the frequency with which a disease can reoccur in an area at a given time. This model was based on supervised machine learning model which predicts the results based on data of previous incidents. The project "Analysing and predictions on diseases using hadoop" is very helpful for the doctors as it can predict the level of emergency in an area. The emergency service providers can allocate their resources efficiently using this project which can save lives, money and response time of responders. By predicting emergency level in an area for a time, the emergency service providers can allocate their resource to such areas which are more prone to any kind of emergency. This is GUI based project, which makes it very easier to use.

So, any person with a very little knowledge of computer technology can use it with ease.

Also, this project can plot scatter graph the incidents per hour against pincode/hour of the day/day of the month/month of the year, which make it easier to understand the results of predictions.

V. REFERENCES

[1]. Berestycki, H., and Nadal, J.-P. 2010. Self-organised critical hot spots of criminal activity. European Journal of Applied Mathematics 21(4-5):371–399.

[2]. Romero Tyler, Barnes Zachary and Cipollone Frank "Predicting Emergency Incidents in San Diego" http://cs229.stanford.edu/proj2016/report/BarnesCipolloneRomeroPredictingEmergencyIncidentsInSanDiego.pdf

[3]. https://www.tutorialspoint.com/python/

[4]. http://www.saedsayad.com/decision_tree_reg.htm

[5]. https://www.kaggle.com/

[6]. Andrew Ng, Co-founder, Coursera; Adjunct Professor, Stanford University, "Decision Tree Regression",https://www.coursera.org/learn/machine-learning

Cite this article as :

Arpna Joshi, Chirag Singla, Mr. Pankaj, "Analysing and Predicting on Diseases using Data Pipeline in Hadoop ", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 2, pp. 1288-1292, March-April 2019. Available at doi : https://doi.org/10.32628/CSEIT1952362 Journal URL : http://ijsrcseit.com/CSEIT1952362