International Journal of Latest Trends in Engineering and Technology
Vol.(8)Issue(2), pp.433-438
DOI: http://dx.doi.org/10.21172/1.82.058
e-ISSN:2278-621X

PROTOTYPE FOR POLICY RECOMMENDATION SYSTEM BASED ON AADHAR DATA

G.Shanti[1] and Chandrika Telugu[2]

Abstract- The main focused of Watermarking is developing and The project carried is in the field of Big data analytics related to computer science. Data analytics is the process of examining data sets in order to draw conclusions about the information they contain. Big data analytics refers to the techniques that can be used for converting raw data into meaningful information which helps in business analysis and forms a decision support system for the executives in the organization. Big data is the large and complex collection of data that cannot be processed using traditional tools. In this proposed work, web application is designed to help government and people to gain knowledge about the government policies and count of people using policies. Citizens will know about the policies they are eligible for with existing policies. Government will know the count of people who are using policies. To implement this project, we are using Hadoop. Common masses can be benefited by the various governmental policies and they can proceed to recommended policies.
Keywords – Big Data, Hive, Hadoop, Thrift, Aadhar, Ambari

I. INTRODUCTION

Prototype for policy recommendation system based on Aadhar data is developed in the interest of government and citizens. Most of citizens does not know how many policies are existing and what policies they are eligible for, even government does not have consolidate list of users who are using policies. This creates a huge gap between citizens and government in policy usage.This work helps to reduce the gap between government and citizens. Citizens will come to know what are all the policies they are using and to which policies they are eligible for.Government will know count of people who are using policies based on category with respect to state.Every citizen uneducated, educated, rich, poor possess Aadhar card and it is government data as it is linked with Banks, gas connections, pass port, service tax etc,. We assumed that Aadhar data can be helpful.

The rest of the paper is organized as follows. Proposed work andComponents used are explained in section II. Experiment flow and Experimental results are presented in section III. Concluding remarks are given in section IV.

II. PROPOSED WORK AND COMPONENTS

A. Praposed project –

Aadhar dataset of country can be collected for multiple states, states can have multiple zones so we can consider it as cluster, data collected from this is a Big data. To handle big data we are using Hadoop.Data which we are using is static data but not dynamic. We may need to include flume in our eco system for dynamic data. We are analysing on sample data which is generated by us.

[1] *Department of Computer Science and Communication Engineering BvritHyderabad College of Engineering for Women,Hyderabad, Telangana, India*
[2] *Department of Computer Science and Communication Engineering BvritHyderabad College of Engineering for Women,Hyderabad, Telangana, India*

B. Components used –

i) APACHE AMBARI :

Open source management platform for provisioning, managing, monitoring and securing Apache Hadoop clusters. Ambari provides an easy-to-use Hadoop management Web UI . It is the interface where we will run all Hadoop components like HDFS, Hive etc,.

ii)HUE:

This also Hadoop interface in which we select the Hadoop tools and we will write queries and execute them. In this interface you can upload files in HDFS. Here we have some components like Hive, Pig etc. which are tools of Hadoop.

iii) HDFS :

Hadoop Distributed File System Java-based file system which provides scalable and reliable data storage. It is used to store bulk amounts of data like terabytes or petabytes. HDFS is mainly used to store data. HDFS support high throughout mechanism for accessing large amount of data. In HDFS files are stored in sequential redundant manner over multiple machines and this guarantees the Durability to failure,High availabilityas features.

iv) HIVE:

Hive is one of the component of Hadoop to process structured data. It helps to summarize Big Data, and makes querying and analyzing easy. Hive provides a mechanism to project structure on to data and query the data using SQL like queries called HiveQL. This language also allows traditional Map and Reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express logic in HiveQL.

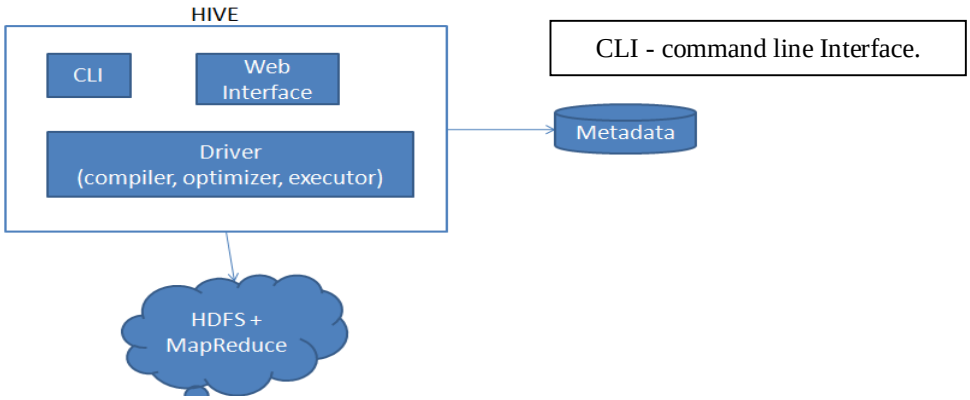


Figure 1. Hive Architecture

Hive is an abstraction on top of MapReduce it allows users to query data in the Hadoop cluster without knowing java and MapReduce. Every hive query is converted into MapReduce job internally. In our project we need Thrift Server as we are connecting through JDBC.

v)Thrift Server :

Thrift server acts as a interface between any language (JDBC) and Driver which sends the request to Hadoop. In our project we are using Web UI for writing and executing our queries.Hivchoosen because because the data which we are analysis is sample and simple data there is no programming logic.

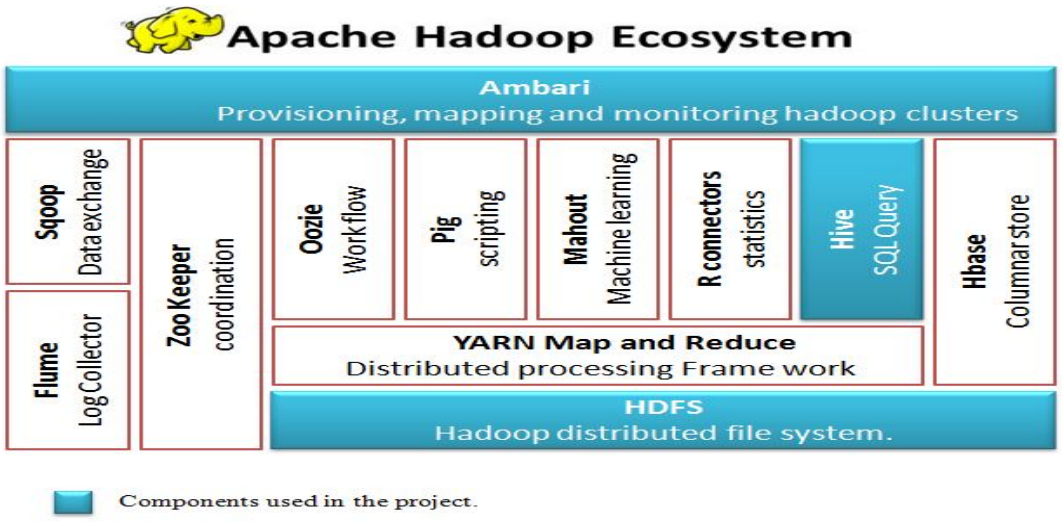


Figure 2. Hadoop Ecosystem with Components usd in this work

III. Experiment Flow and Result

A. Experiment Flow–

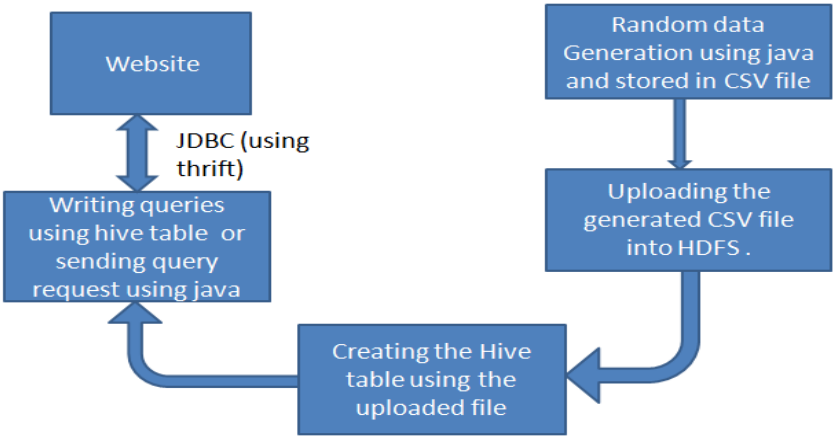


Figure 3. Experiment Flow

1.Data Generation:

Data is not realtime data, it is generated data. We are generation random dataset using java. Random() is used to generate the data randomly in java. The attributes which we are generating are aadhar_no, name , age, gender, state, address, Dwcra, Podhupu laxmi, Pension, Health Insurance, girl child Protection and Child development. As our data is not realtime data we are considering only two polices for three categories.This generated random data is stored in CSV file. CSV(Comma Separated Value) is a simple file format used to store tabular data, like spreadsheets or database.

Pseudo Code:

```
1. long aadharArray[] = new long[1000];
   String dwcra[] = { "dwcra", "NA" };
   String podhupuLaxmi[] = { "NA", "PodhupuLaxmi" };
   String healthInsurance[] = { "Health Insurance", "NA" };
   String girlChildProtection[] = { "Girl Child Protection", "NA" };
   String gender[] =  { "F", "M" };
2. Importing random() method
3.for  i = 0 to aadharArray.length  do
         print aadhr_no,age,gender etc.,.
   end for
```

Flow Chart :

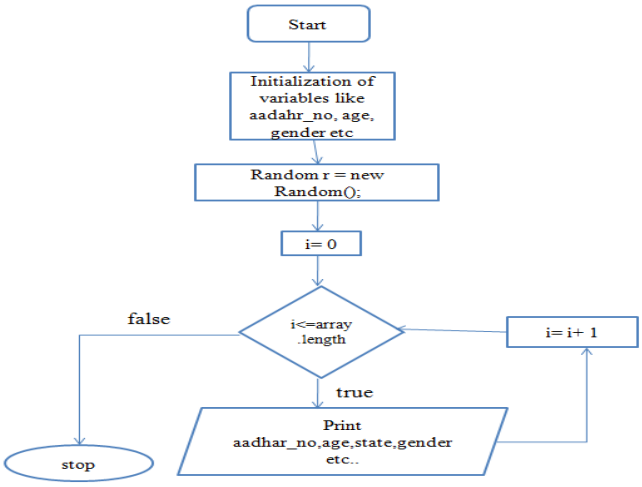


Figure 4. Flow chart for Data Generation

2.Uploading the generated file to HDFS :

Generated CSV data is uploading to HDFS. The data is stored in HDFS as CSV file.

3.Creating Hive Table:

Creating Hive table using the uploading data. Creating table can be done in two ways. Selection of file and Manual Creation.Here we are using selection of file to create hive table.

B. Experiment Procedure–

a) Partioned CSV file :

Complete CSV file is partitioned because the query may take long time to execute due to size of data. So, we are partitioning the into two datasets.Here we are partitioning the data based on states as we are displays the count with respect to state. So that when we are writing queries of state1 will write based on state1 partitioned table.A small query in Hive reads the complete dataset. This becomes a difficult for running MapReduce jobs over a large volume of data. This issue can be solved by implementing partitions in Hive. Hive makes it very easy to implement partition . In Hive's implementation of partitioning, data within a table is split across multiple partitions based on column names in the table. The Partitioned table is stored as a sub-directory in HDFS.

b) Connection from website to HDFS :

A request sent from website to HDFS using Hive queries , hive table is connected using JBDC connection. Queries written in Java are sent to Hive table. It loads data from HDFs and executes the query and returns the results to website. It internally uses thrift server for connection.

C.Experiment Results–

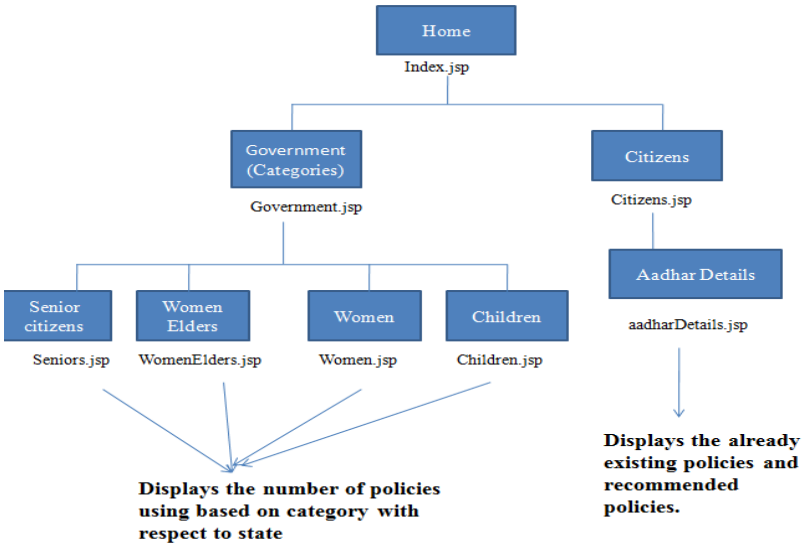
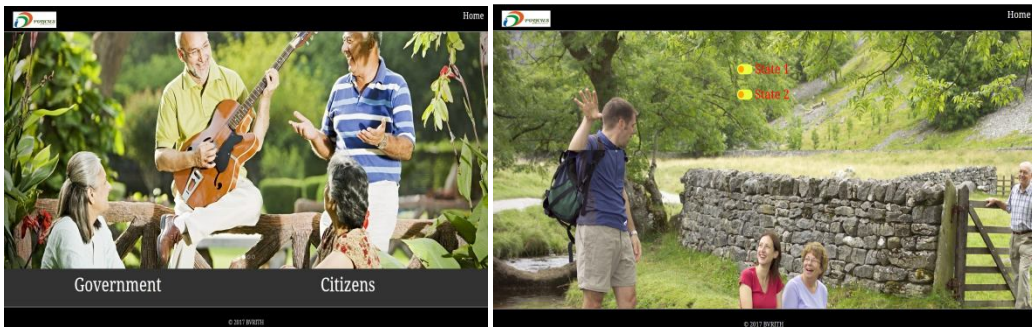


Figure 5. Website Map



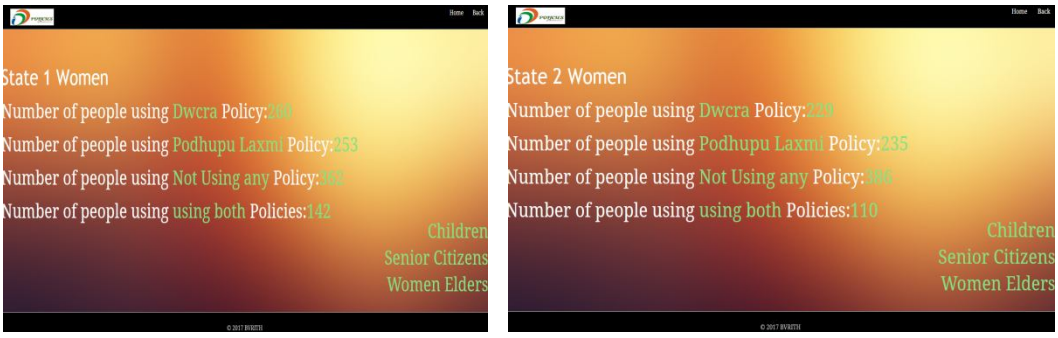


Figure 6. Home page with Governament and citizen links

D.Test cases on Website and queries –

S.No	Input	Expected Outcome	Actual Outcome
1.	Selection of State	Home page – government page(selection of state and category selection) – displays the number of people using policies based on category w.r.to state.	Category name jsp page. Eg: women.jsp
2.	Enter valid Aadhar card no.	Home page—Citizens page (enter aadhar _no)--Recommended policies / Number of policies used	aadharDetails.jsp

S.No.	Input	Expected Outcome	
1.	Partition on state	Checking whether the partitioned data is equal to whole data or not by writing a query.	Count of whole data should be equal to Count of state1 partition + state2 partition.
2.	Query Execution	Checking the query count that we are using in java with hive queries in Hadoop.	Count should be same .

IV.CONCLUSION

- Government will know who are eligible but not using so we can extend it by providing name and address.
- For citizens we can direct provide link to recommended policies website and sending message to user about new policies.
- We can use flume if it is dynamic data.

PERFORMANCE COMPARISION

JAVA	HADOOP
1. It cannot handle large data.	1. It can handle large data.
2. Partitioning of data is not possible .	2. Partition is possible in Hadoop.
3. In java, we use SQL.	3. In Hadoop component Hive we use HiveQL which is similar to SQL queries.
4. It takes long time to process a query.	4. Query processing is fast.

Table -1 Comparision of performance with java

V REFERENCES

[1] P. S. Huang, C[1] Clifford Lynch, “Big data: How do your data grow?” Nature, International weekly journal of science,455, 28, 1974.

[2] Min Chen, Shiwen Mao, Yunhao Liu Big Data: A Survey Springer-Mobile Networks and Applications, 19 , 2014.

[3] Available: http://www.sciencedirect.com

[4] Shadi Ibrahim, Hai Jin, Lu Lu, Li Qi, Song Wu, and Xuanhua Shi, “Evaluating MapReduce on Virtual Machines: The Hadoop Case”, Springer: Cloud Computing Lecture Notes in Computer Science, Volume 5931, 519-528 pp, 2009.