**BIG DATA FINAL REPORT**

**Log file analysis,Disease finding by symptoms and use case analysis of health care**

**Group members:**

Padegal Sai Giriraj - S4 - 121

Santosh kumar Boyina - S4 - 140

Inagalla Lohiya - S4 -100

M Sai Kushal - S4 - 114

**TABLE OF CONTENTS**

A. PROBLEM STATEMENT

B. INTRODUCTION

C. DATASETS

D. WHY USE BIG DATA

E. WHY USE PIG

F. TECHNOLOGIES USED

G. EVALUATION METRIC

F. CODES AND RESULTS

**A.Problem statement:**

The aim is to manage a health care management system. The various functionalities related to health care will be handled by Map-Reduce system making it more efficient.

This module contains 3 different parts

1. Log file analysis :[Health care website log]
2. Disease finder
3. Other use cases based on the data sets
4. Alive and dead events

**B.Introduction :**

**Log file analysis:**

As a part of our capstone project we generated log file for the website created .We created log files for the health care management system .

The logs were set based on the different levels .We used that file and obtained how many logs have been generated for each level.

**Symptom finder :**

There is a lot of increase in diseases today.These can be identified by symptoms.Doctors generally detect those by making several tests.So based on the previous results of symptoms,diseases can be identified by the symptoms

In the data set.

Final output of this is if you enter the symptoms you will be able to find the disease.

**Use Case analysis of health care :**

We have got different datasets for different types of health care management for example we got a dataset which contains the state wise hospitals list and number of patients based on the diseases they got and the details of the payment’s and others.

Different type of use cases are mentioned below

**Alive & Dead events :**

These two data sets contain the details of the events when the person is alive and dead events contain the dead people id and timestamp of death time.

What we did is we joined the both data files we got the details of the alive events of people before they died and difference between the timestamps of alive events and dead timestamp is obtained.

**C.Datasets :**

**Log file :**

we have created our log file by adding a plugin to our own web application of healthcare. And we have used generated log files as a dataset.

**Symptom analysis**:

We have taken a dataset from kaggle website it consist of lot of diseases with no.of diagnosis count,reason code,complaint\_id and place where disease occurs.

**Link** [**https://www.kaggle.com/plarmuseau/sdsort/data**](https://www.kaggle.com/plarmuseau/sdsort/data)

**Use cases of health care:**

[**https://data.medicare.gov/Hospital-Compare/Payment-and-value-of-care-Hospital/c7us-v4mf**](https://data.medicare.gov/Hospital-Compare/Payment-and-value-of-care-Hospital/c7us-v4mf)

**D. Why Use Big Data:**

* The major questions addressed here are what technologies were being used until now and why do we need to involve Big Data management? The answer is very simple, so far there are different databases to handle such bank related management schemes like heavily normalized SQL, Oracle or Excel for small data sets.
* But Can we improve the effectiveness of the management system by bringing in the biggest Big data concept i.e. the MAP-REDUCE technique, yes and the reason are as follows:
* **Size-** The amount of data is increasing every day, we do not need to mention why the size will increase in the banking field as well, with TBs and PBs of data it is best to use a better technology which will make handling way safer, easier and sophisticated.
* **Diversity-** Ever since mankind has started to grow there have been substantial increase in different functions, services etc. In the field of banking the diversity is also increasing day by day as variety in the information is increasing
* **Complexity-** With variety, veracity and size, it is very obvious that complexity will increase. As in case of a banking system, the information of people, information of banks and how it is linked with each other is a complex enough task to involve a technology like MAP-REDUCE.

**E. WHY PIG:**

* Decrease in development time. This is the biggest advantage especially considering vanilla mapreduce jobs' complexity, time-spent and maintenance of the programs.
* Procedural, not declarative unlike SQL, so easier to follow the commands and provides better expressiveness in the transformation of data every step.
* Since it is procedural, you could control of the execution of every step. If you want to write your own UDF (User Defined Function) and inject in one specific part in the pipeline, it is straightforward.
* Lazy evaluation: unless you do not produce an output, file or does not output any message, it does not get evaluated.

**F.Technologies Used :**

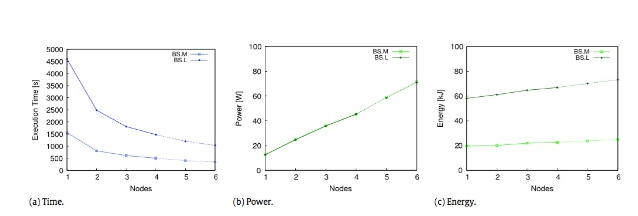
* PIG
* Map Reduce
* Hadoop

## 

## G. The Evaluation Metric :

* The efficiency and Effectiveness of the project will be measured according to the output obtained after the dataset is processed and we are able to identify the accurate result of a given entity.
* ·The accuracy of result can be determined by the time it takes for the entire MapReduce model to execute and generate the output.
* We have broken down the time segments of each major operation and thus made our analysis based on those results.
* Apart from time, the next Metric used is the effectiveness of the code, we have tried to make it very function specific so that unnecessary data can be removed

**The Effectiveness of Map-Reduce:**

****

**H.Detailed Pseudocode for the solution:**

**Log file analysis:(log.pig)**

Logging is done by log levels (TRACE,DEBUG,INFO,WARN,FATAL)

The input file is the log file we generated.and the output is how many fatal and info errors are occured when user is logged in our website

**Code:**

log = LOAD '/home/cloudera/Desktop/log4j-application.log' as line;

LEVELS = foreach log generate REGEX\_EXTRACT(line,'(TRACE|DEBUG|INFO|WARN|FATAL)',1) as LOGLEVEL;

FILTEREDLEVELS = FILTER LEVELS by LOGLEVEL is not null;

GROUPEDLEVELS = GROUP FILTEREDLEVELS by LOGLEVEL;

FREQUENCIES = foreach GROUPEDLEVELS generate group as LOGLEVEL,COUNT(FILTEREDLEVELS.LOGLEVEL) as count;

STORE FREQUENCIES into '/home/cloudera/Desktop/log' using PigStorage

(',');

**output:**



**Disease finding by symptoms:(symptoms.pig)**

We filter the data from dataset which we used if suppose symptom=”fever” the output it will produce the diseases which are occured by the symptoms

**Code:**

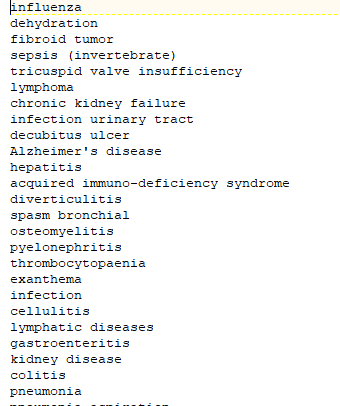
data = load '/home/cloudera/Desktop/symptoms\_bench.csv' using PigStorage(',') as (disease:chararray,symptom:chararray)

result = filter data by ((symptom == 'fever') or(symptom == 'headache'));

result1 = foreach result generate disease;

store result1 into '/home/cloudera/Desktop/result7' using PigStorage(',');

**Output:**



**Use cases analysis of health care :**

**Q.1) analysing heart failure patients and generate hospital name,payment,location,providerid (heartfailure.pig)**

The output it will produce is at which hospital there are heart failure patients and the hospital location and how much amount they paid

**Code:**

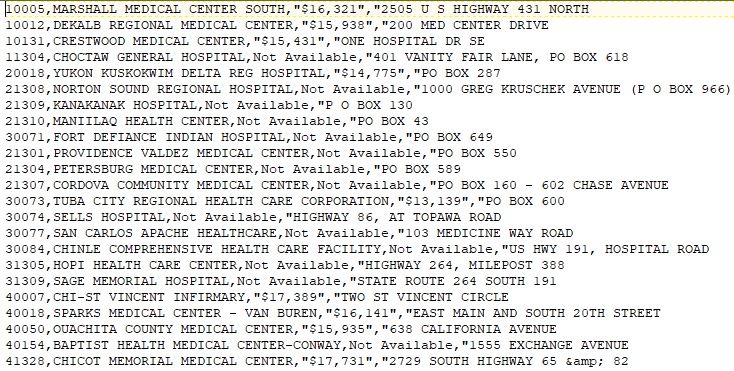
health = load '/home/cloudera/Desktop/Payment\_and\_value\_of\_care\_-\_Hospital.csv' using PigStorage('\t') as (ProviderID:chararray,HospitalName:chararray,Address:chararray,City:chararray,State:chararray,ZIPCode:chararray,Countyname:chararray,Phonenumber:chararray,Paymentmeasurename:chararray,PaymentmeasureID:chararray,Paymentcategory:chararray,Denominator:chararray,Payment:chararray,Lowerestimate:chararray,Higherestimate:chararray,Paymentfootnote:chararray,Valueofcaredisplayname:chararray,ValueofcaredisplayID:chararray,Valueofcarecategory:chararray,Valueofcarefootnote:chararray,MeasureStartDate:chararray,MeasureEndDate:chararray,Location:chararray);

result = filter health by Paymentmeasurename == 'Payment for heart failure patients';

result1 = FOREACH result generate ProviderID,HospitalName,Payment,Location;

STORE result1 into '/home/cloudera/Desktop/heartfailure' using PigStorage(',');

**Output:**

****

**Q.2) Footnote analysis given by patient**

Generally footnote is the remarks or suggestions given by the patient

The hospital management will take the footnote given by the patient

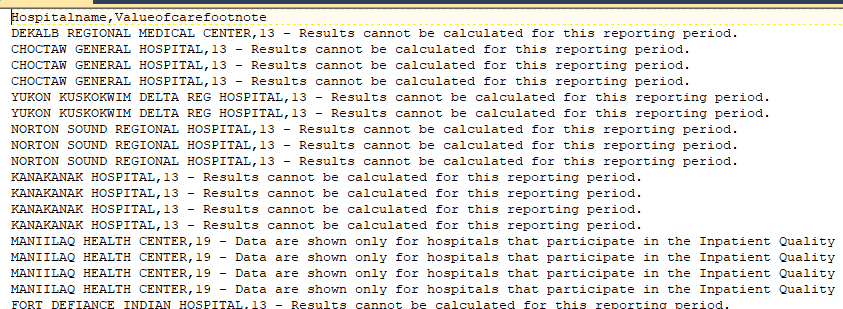
**code:**

result = filter health by Valueofcarefootnote is NOT NULL;

result1 = FOREACH result generate HospitalName,Valueofcarefootnote;

STORE result1 into '/home/cloudera/Desktop/santosh/footnote1' using PigStorage(',');

**Output:**

****

**Q.3) Dividing the heart failure patients by state wise**

Filtering the data by state wise

Suppose state ak contains hospital who treated heart failure patients and there location

**Code:**

result = filter health by Paymentmeasurename == 'Payment for heart failure patients';

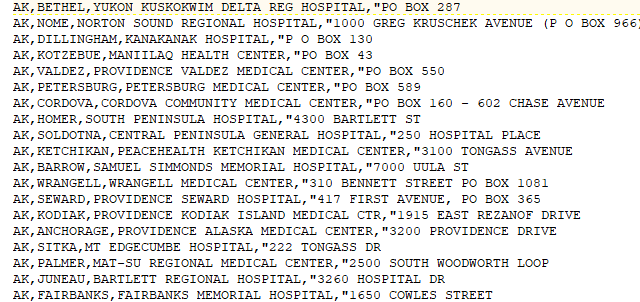
result1 = FOREACH result generate State,City,Hospitalname,Location;

split result1 into st1 if State =='AK',st2 if State =='AL',st3 if State =='AZ',

STORE st1 into '/home/cloudera/Desktop/santosh/st1' using PigStorage(',');

STORE st2 into '/home/cloudera/Desktop/santosh/st2' using PigStorage(',');

STORE st3 into '/home/cloudera/Desktop/santosh/st3' using PigStorage(',');

**Output**

**Q.4) Analysis of alive events and dead events**

Analysing the alive and dead events by patientid and hospitalid which were there in the dataset

**Code:**

events = LOAD '/home/cloudera/Desktop/events.csv' USING PigStorage(',') AS (patientid:int, eventid:chararray, eventdesc:chararray, timestamp:chararray, value:float);

events = FOREACH events GENERATE patientid, eventid, ToDate(timestamp, 'yyyy-MM-dd') AS etimestamp, value;

mortality = LOAD'/home/cloudera/Desktop/mortality.csv' USING PigStorage(',') as (patientid:int, timestamp:chararray, label:int);

mortality = FOREACH mortality GENERATE patientid, ToDate(timestamp, 'yyyy-MM-dd') AS mtimestamp, label;

eventsjoin = JOIN events by patientid LEFT, mortality by patientid;

deadevents = Filter eventsjoin by(mortality::patientid is not null);

deadevents = FOREACH deadevents GENERATE events::patientid as patientid, events::eventid as eventid, events::value as value, mortality::label as label, DaysBetween(mortality::mtimestamp,events::etimestamp)-30 as time\_difference;

aliveevents = Filter eventsjoin by(mortality::patientid is null);

aliveevents = FOREACH aliveevents GENERATE events::etimestamp as timestamp, events::patientid as patientid, events::eventid as eventid, events::value as value, 0 as label;

aliveeventsgrp = GROUP aliveevents BY patientid;

aliveeventsindex = foreach aliveeventsgrp generate group as patientid,MAX(aliveevents.timestamp) as index;

aliveevents = join aliveevents by patientid, aliveeventsindex by patientid;

aliveevents = FOREACH aliveevents GENERATE aliveevents::patientid as patientid, aliveevents::eventid as eventid, aliveevents::value as value, aliveevents::label as label, DaysBetween(aliveeventsindex::index,aliveevents::timestamp) as time\_difference;

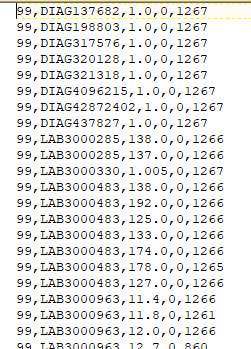
deadevents = ORDER deadevents BY patientid, eventid;

aliveevents = ORDER aliveevents BY patientid, eventid;

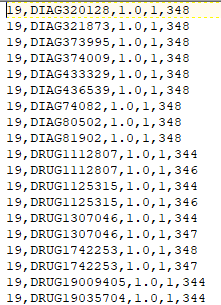
STORE aliveevents INTO '/home/cloudera/Desktop/aliveevents' USING PigStorage(',');

STORE deadevents INTO '/home/cloudera/Desktop/deadevents' USING PigStorage(',');

**Output of dead events:**

****

**Output of alive events:**

****