Sensex Log Data POC

Objective:

To study data processing of Sensex Log Data.

Input Data:

Pdf File (With over 3000 Records)

Input Data Format:

SENSEXID SENSEXNAME TYPEOFTRADING SENSEXLOC
OPEN_BALANCE CLOSING_BAL FLTUATION_RATE

Eg: (Tab Separated Records)

121213 NSE_Sensex_Report Daily NewDlihi 29525 29800 10

Task:

Task 1: Use MapReduce Code

The input data from pdf is to extracted and placed into 5 different files according to following conditions:

If TYPEOFTRADING is 'Sip'

- And if OPEN_BALANCE > 25000 and FLTUATION_RATE > 10 then store the details in "**HighDemandMarket**" file.
- And if CLOSING_BALANCE<22000 & FLTUATION_RATE IN BETWEEN 20 and 30 (Both inclusive) then store the details in "**OnGoingMarketStretegy**" file.

If TYPEOFTRADING is 'ShortTerm'

- And if OPEN_BALANCE < 5000 then store the details in "**WealthyProducts**" file.
- And if SensexLoc is "NewYork OR California" then store the details in "**ReliableProducts**" file.

Else

- Store the details in "**OtherProducts**" file.

Task 2: Use Pig Script to write Pig Commands

Develop a PIG Script to filter the Map Reduce Output in the below fashion

- Store Unique data in a HDFS Directory.
- Sort the Unique data based on SensexID and Store it in a different HDFS Directory.

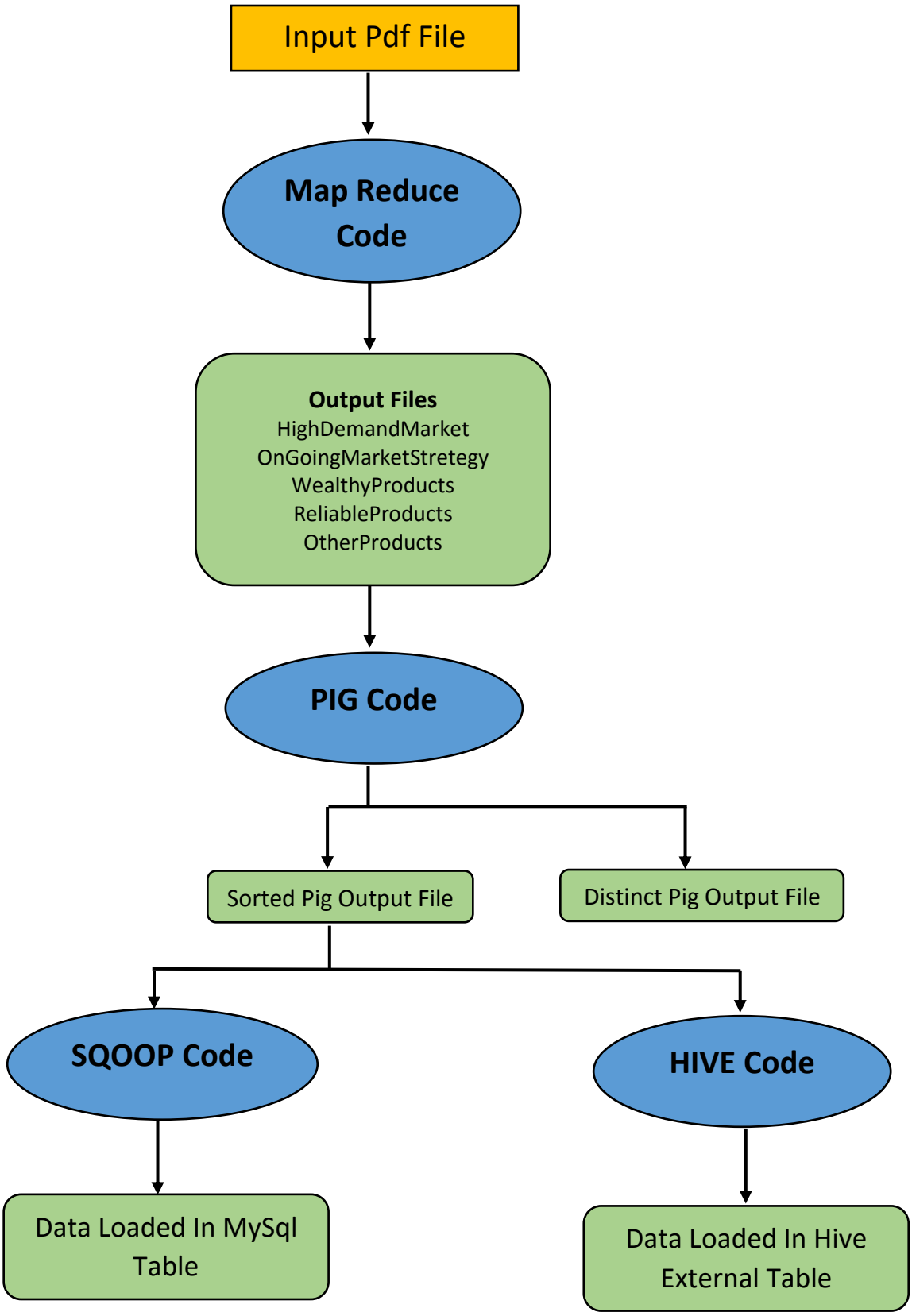
Task 3: Use Sqoop Commands

EXPORT the same PIG Sorted Output from HDFS to MySQL using SQOOP.

Task 4: Use Hive Commands

Store the same PIG Sorted Output in a HIVE External Table.

Architectural Flow Diagram:



Technical Deployment:

The whole Use Case is executed by executing the script file "**sensexPoc.sh**", provided the input file "**Poc2InpuData.pdf**", Pig Script "**PocSensex.pig**", and the Jar file "**PdfProcessing.jar**" of MapReduce Job is present in the same directory as that of the Script.

The commands used inside this script are explained below:

Some variables used are initializes first:

```
#Input File Detail
inputFileName=Poc2InputData.pdf

#DataBase And Tables Created
mysqlDbName=SensexDb
mysqlTableName=sensex_tab
hiveDbName=sensex_db
hiveTableName=sensex_tab

#Input and Output HDFS Location Names
InputPdfLocation=/Poc2Input
MRLogLocation=/SensexLog
MROutputLocation=/SensexOut
PigDistinctOutputLocation=/PigDistinctPoc2
PigSortOutputLocation=/PigSortPoc2

#Mysql Connection Details
mysqlUser=root
mysqlPass=root
```

Actual Code:

```
#Removing HDFS Directories

hadoop fs -rmr $PigDistinctOutputLocation $PigSortOutputLocation $InputPdfLocation
```

This code will remove the mentioned directories enabling us run the same Use Case on a machine again and again, without getting an error that "Directory already exists".

#Placing Input Pdf File On HDFS

```
hadoop fs -mkdir $InputPdfLocation
hadoop fs -put $inputFileName $InputPdfLocation
```

To run our "**Task 1**" which is execution of a MapReduce job using a Jar file, we first need our input file to be on HDFS location. Thus this code will create a directory on HDFS and then place the input file into this directory.

#Running MapReduce Job

```
hadoop jar PdfProcessing.jar com/sensex/poc/PdfDriver $InputPdfLocation/$inputFileName
$MRLogLocation $MROutputLocation
```

This MapReduce Job perform our "**Task 1**", which is extracting the data from pdf file according to the given conditions and place them in respective 5 files.

#Running Pig Script

```
pig -p Input=$MROutputLocation/* -p DistinctLoc=$PigDistinctOutputLocation -p
SortLoc=$PigSortOutputLocation PocSensex.pig
```

This code will perform out "**Task 2**".
This code will run the mentioned pig script, which will filter the distinct data from the output of MapReduce Job and then will Sort the filtered data according to Sensex_Id and will store this in a HDFS location.

The content of Pig Script is described below:

```
inputData = load '$Input' using PigStorage('\t');
```

Above line will load the data, which is the output of MapReduce Job, into a variable *inputData*.

```
distinctData = DISTINCT inputData;
```

Above code will load the Distinct data of variable *inputData* into the variable *distinctData*.

```
sortData = ORDER distinctData BY $0;
```

Above code will load the data present in *distinctData* variable after Sorting it by Patient_Id into the variable *sortData*.

```
store distinctData into '$DistinctLoc';store sortData into '$SortLoc';
```

Above code will Store the data in the variables *distinctData* and *sortData* into the mentioned HDFS directories.

#Creating MySql Table

```
mysql -u $mysqlUser -p$mysqlPass << EOF
drop database if exists $mysqlDbName;
create database $mysqlDbName;
create table $mysqlDbName.$mysqlTableName(
sid int PRIMARY KEY,
sname varchar(20),
strade varchar(20),
sloc varchar(20),
sopen int,
sclose int,
sfluc int);
grant all privileges on $mysqlDbName.* to ''@'localhost';
EOF
```

The above code is the prerequisite of our "**Task 3**".
Yes, before exporting the data into a MySql table, first we need to have a table. This code will first delete the database of MySql if it already exists and then will create the database and the table inside that database.

#Exporting Pig Output to MySql Table

```
sqoop export --connect jdbc:mysql://localhost/$mysqlDbName --table $mysqlTableName --
fields-terminated-by '\t' --export-dir $PigSortOutputLocation/part*;
```

This code will perform our "**Task 3**", that is it will export the output of **Task 2,** that also the sorted output, into the MySql table created above.

#Exporting Pig Output to Hive Table

```
hive << EOF
drop database if exists $hiveDbName cascade;
create database $hiveDbName;
create external table $hiveDbName.$hiveTableName(
sid int,sname string,strade string,sloc string,sopen int,sclose int,sfluc int)
row format delimited
fields terminated by '\t'
lines terminated by '\n'
stored as textfile location '$PigSortOutputLocation';
EOF
```

The above code will perform our “**Task 4**”. This code will create a hive table, and then will load the Sorted Output of **Task 2** (that is pig script execution task), into this hive table.

At the end we will have the following things:

1. 5 Output files of MapReduce Job in a HDFS Directory.
2. 2 Output files of Pig Script execution.
3. A MySql table with Sorted output of Pig task loaded in it.
4. A Hive table with Sorted output of Pig task loaded in it.

--------End--------