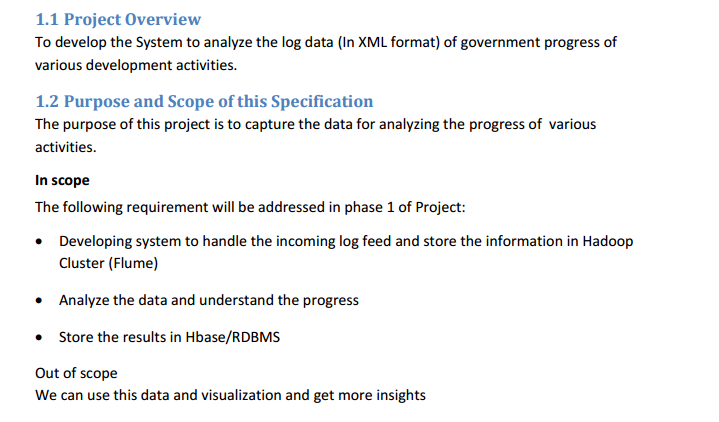
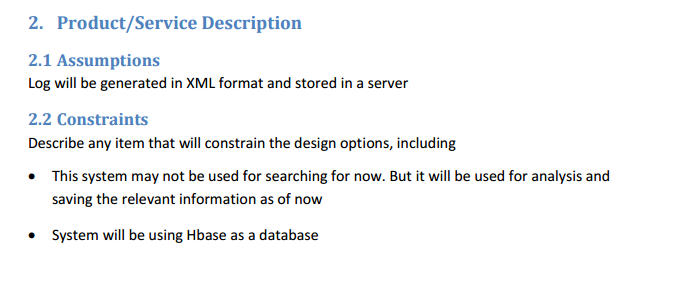
**-: MAJOR PROJECT:-**

**STATE-WISE DEVELOPMENT ANALYSIS IN INDIA**

**PROBLEM STATEMENT:-**

****

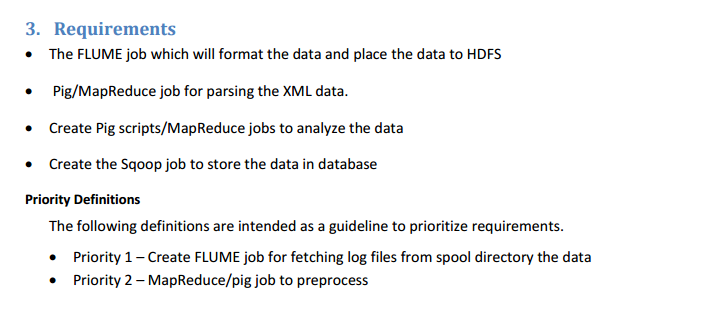
The project gives a glimpse over the overall content of Hadoop ecosystem. It starts with storing the dataset into HDFS through flume. Then the dataset is in xml format, so we can do it in two ways either directly take it as input through pig Latin or giving input format as xml in driver class in map reduce program.



**No we will download the content from the following link,**



The downloaded content would be in local system, now we have to transfer the dataset into HDFS to perform data analytics to get a valuable information over the dataset.

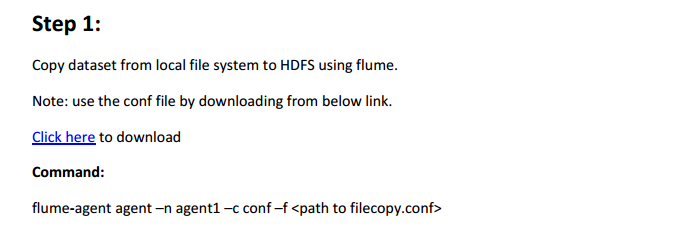
****

Now we have two options to perform analysis, map reduce or pig

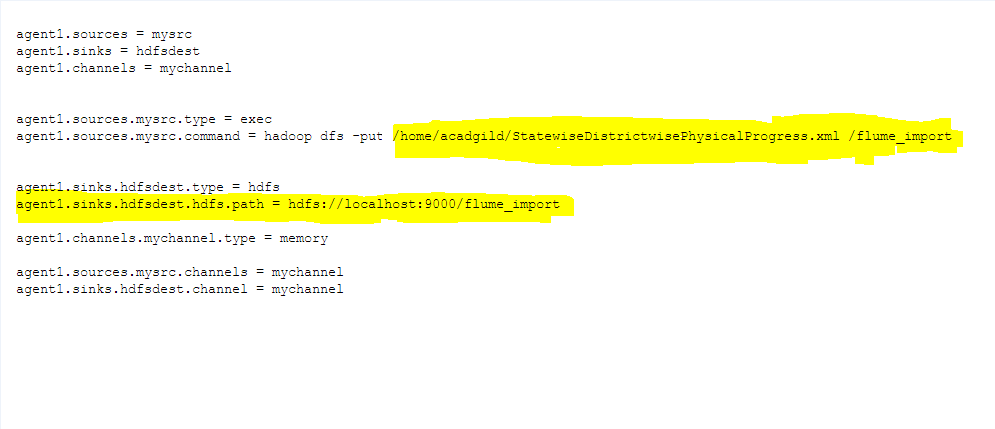
**MAP-REDUCE :** Map-reduce can handle extreme level of complexity, but considering the coding complexity and further use of the obtained result I prefer pig Latin over map reduce

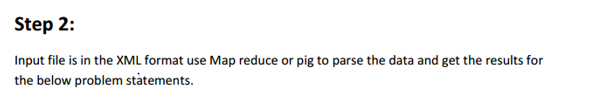
**PIG-LATIN:** In Pig-Latin the commands are easy to execute and in case if the obtained results need a join with other data set then there would be a complexity in map-reduce. So, considering the future performance, I chose pig Latin.

**Now we will download the flume configuration file**

****

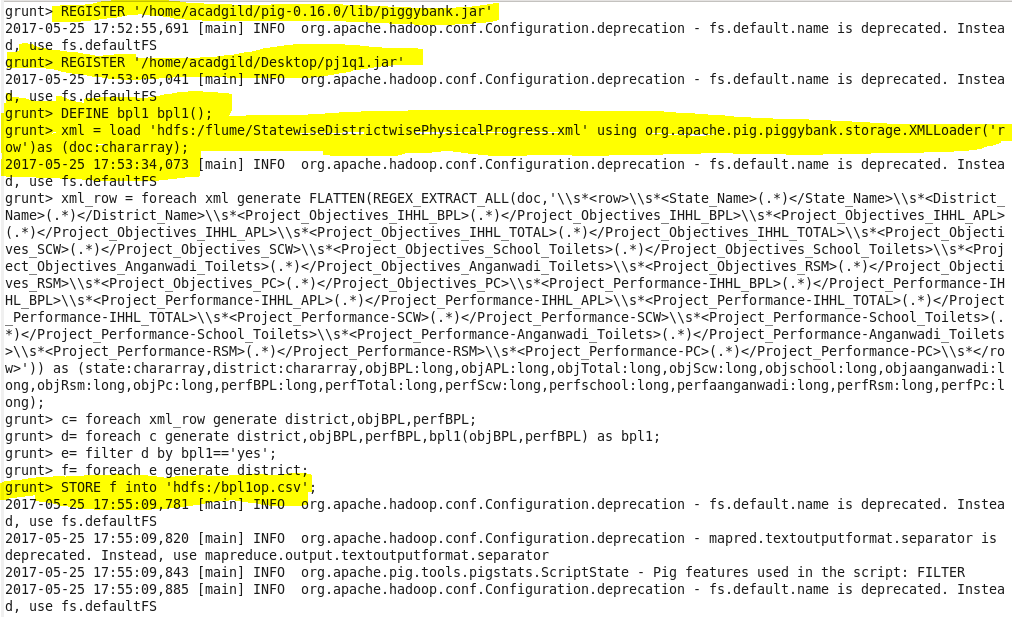
* The flume configuration is downloaded and placed in /apacheflume/conf folder
* Here this configuration file gives us the source ,sink,channel,sink command to be performed and where the content to be stored
* Here it is in local folder /home/acadgild/StatewiseDistrictwisePhysicalProgress.xml which is to be transferred to the hdfs folder /flume

****



**1. Find out the districts who achieved 100 percent objective in BPL cards Export the results to my SQL using SQOOP**

The question is to find the districts which achieved 100 % of its objective in bpl cards

****

At first, we are registering the piggybank jar, this would help in loading XML input file.

Next we are registering the jar file that is used to check whether it has succeeded the objective.

Now we are creating the object from the java class

Then we are loading the xml file from HDFS system that consider each row as a value stored in a x of data type – chararray.

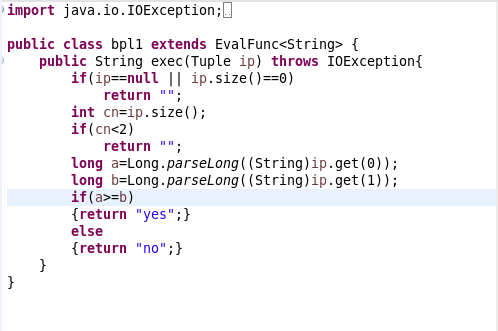
This would help in obtaining the each row value

Next we are using regular expression to obtain the value from each row removing the column names and we are giving each column’s data type

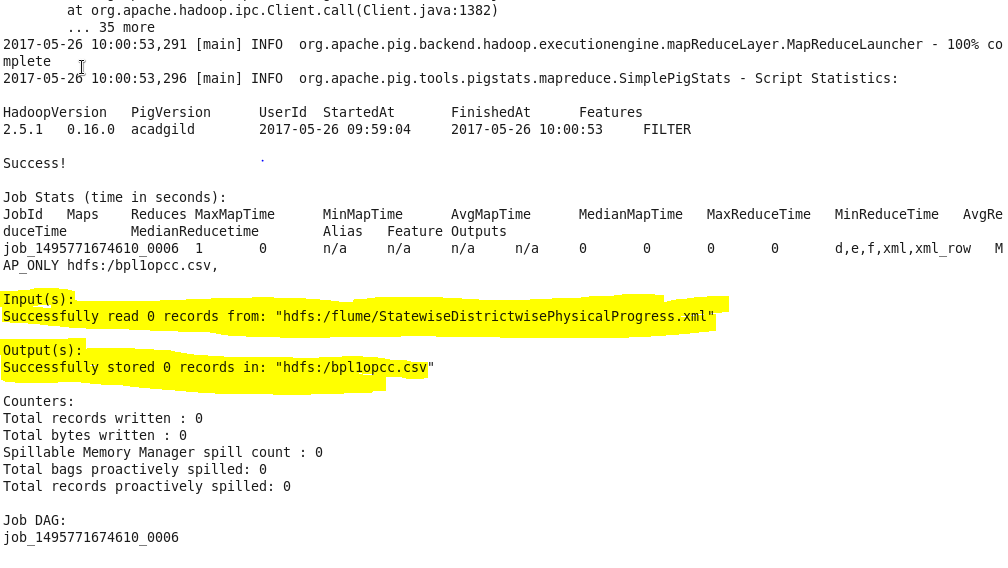
Then from this we are generating only the district name,objective of bpl and performance of bpl

The filtered content is given as input for the jar class

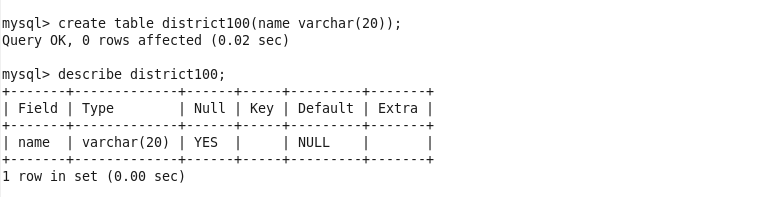
The jar class compares the performance with objective and check whether it is more than that or not .If it is achieved it return ‘yes’ or else ‘no’.



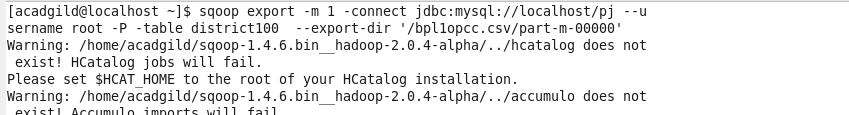
* First the input tuple is checked whether it is null or 0, if yes it returns null value
* Then it checks whether it has two column values also and then we are converting these values into long
* No we are checking whether the performance is more than the objective, if it is it returns ‘yes’.
* Then from the obtained output ‘yes’ content alone is filtered
* Then for each ‘yes’ corresponding district name is obtained
* This is then stored into a hdfs file



We are creating my SQL table to store the district that obtained more than or equal to the objective

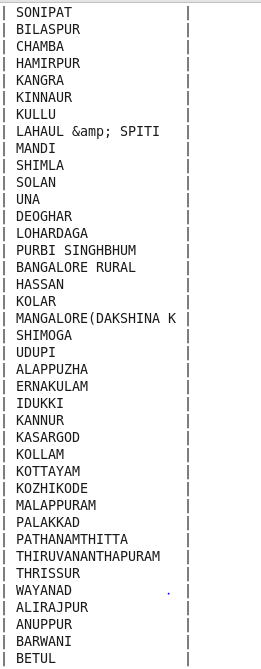


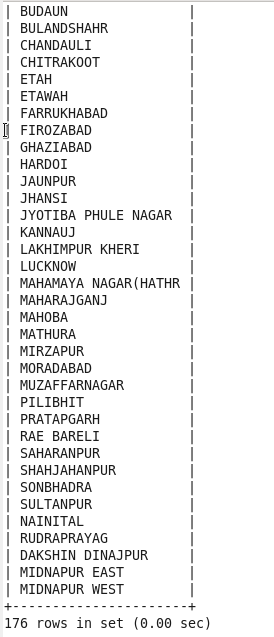
Now the HDFS file is exported to my SQL through SQOOP which connects my SQL and store the content in **pj database , district100 table** from the output HDFS file **‘/bpl1opcc.csv/part-m-00000’**



**The following districts has achieved 100% of objective in bpl**

****

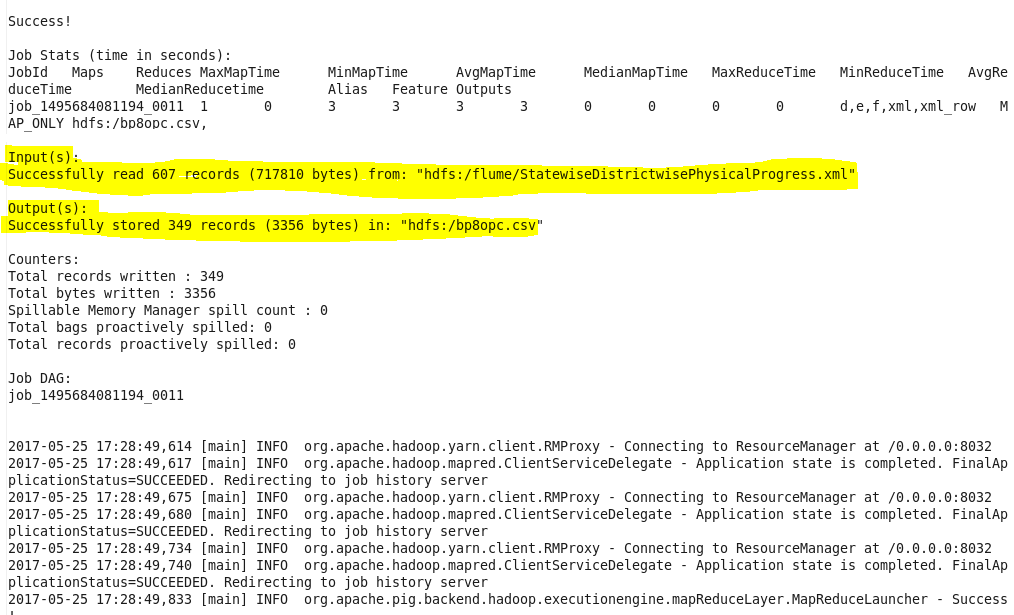
****

****

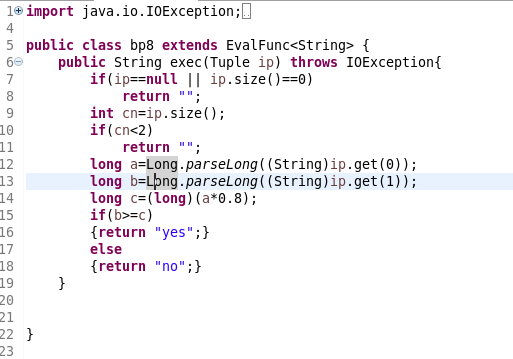
**2. Write a Pig UDF to filter the districts which have reached 80% of objectives of BPL cards. Export the results to MySQL using SQOOP.**

****

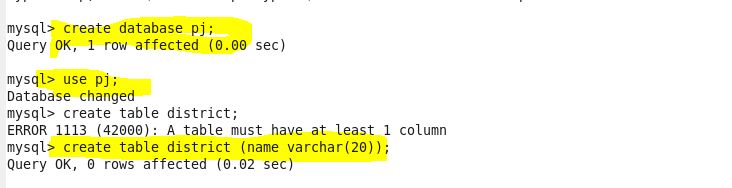
* At first, we are registering the piggybank jar, this would help in loading XML input file.
* Next we are registering the jar file that is used to check whether it is succeeded 80 % of the objective.
* Now we are creating the object from the java class
* Then we are loading the xml file from HDFS system that consider each row as a value stored in x of data type –chararray.
* This would help in obtaining the each row value
* Next we are using regular expression to obtain the value from each row removing the column names and we are giving each column’s data type
* Then from this we are generating only the district name , objective of bpl and performance of bpl
* The filtered content is given as input for the jar class
* The jar class compares the performance with objective and check whether it is more than that or not .If it is achieved it return ‘yes’ or else ‘no’



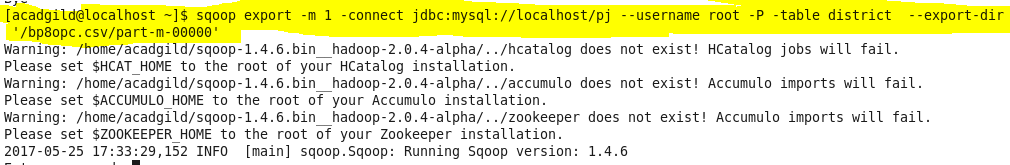
* First the input tuple is checked whether it is null or 0,if yes it returns null value
* Then it checks whether it has two column values also and then we are converting these values into long
* No we are checking whether the performance is 80% of the Objective, if it is it returns ‘yes’.
* Then from the obtained output ‘yes’ content alone is filtered
* Then for each ‘yes’ corresponding district name is obtained
* This is then stored into a HDFS file



We are creating my SQL table to store the districts that obtained 80% of the objective

****

Now the HDFS file is exported to my SQL through SQOOP which connects my SQL and store the content in **pj database ,district table** from the output HDFS file **‘/bp8opc.csv/part-m-00000’**



**The following districts has achieved 80% of objective in bpl**

