Data Ingestion:

Data Ingestion: flume job

agent1.sources = mysrc

agent1.sinks = hdfsdest

agent1.channels = mychannel

agent1.sources.mysrc.type = exec

agent1.sources.mysrc.command = hadoop dfs -put /home/acadgild/Project/StatewiseDistrictwisePhysicalProgress.xml /flume\_import

agent1.sinks.hdfsdest.type = hdfs

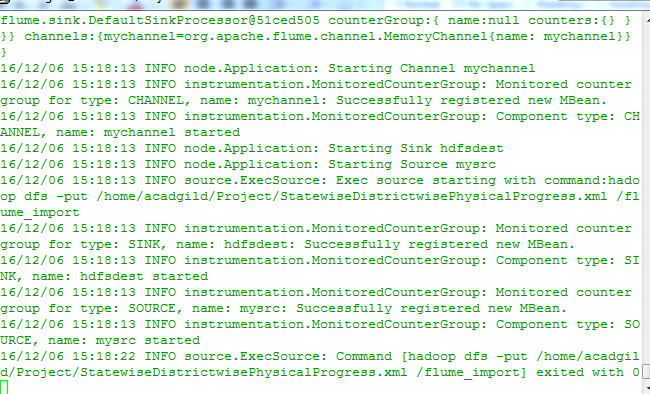
agent1.sinks.hdfsdest.hdfs.path = hdfs://localhost:9000/flume\_import

agent1.channels.mychannel.type = memory

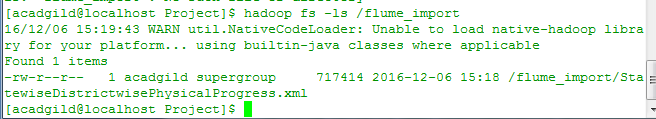
agent1.sources.mysrc.channels = mychannel

agent1.sinks.hdfsdest.channel = mychannel

flume-ng agent -n agent1 --conf-file filecopy.conf



The command is executed. The file is now present in HDFS.

  
Pig/Mapreduce job for parsing the XML data:

1. Find out the districts who achieved 100 percent objective in BPL cards

Export the results to mysql using sqoop

hadoop fs -mkdir -p /project/lib

hadoop fs -put /home/acadgild/project/lib/piggybank.jar /project/lib/piggybank.jar

pig -x mapreduce

REGISTER 'hdfs:///project/lib/piggybank.jar';

DEFINE XPath org.apache.pig.piggybank.evaluation.xml.XPath();

A = LOAD '/flume\_import/StatewiseDistrictwisePhysicalProgress.xml' using org.apache.pig.piggybank.storage.XMLLoader('row') as (x:chararray);

B = FOREACH A GENERATE XPath(x, 'row/District\_Name') AS district\_Name, XPath(x, 'row/Project\_Objectives\_IHHL\_BPL') AS bpl\_objective, XPath(x, 'row/Project\_Performance-IHHL\_BPL') AS bpl\_performance;

result = FOREACH B GENERATE district\_Name, (((double)bpl\_performance/(double)bpl\_objective) \* 100) AS objective\_percent;

filtered\_result = FILTER result by objective\_percent == 100;

dump filtered\_result;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2. Write a Pig UDF to filter the districts who have reached 80% of objectives of BPL cards.

Export the results to mysql using sqoop.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package pigudf;

import java.io.IOException;

import org.apache.pig.FilterFunc;

import org.apache.pig.data.Tuple;

public class FilterUDF extends FilterFunc {

public Boolean exec(Tuple input) throws IOException {

try {

double value1 = (double) input.get(0);

double value2= (double)input.get(1);

if ((value1/value2) \* 100 = 80) {

return true;

}

else {

return false;

}

}

catch (Exception e) {

throw new IOException(e);

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pig -x mapreduce

REGISTER 'hdfs:///project/lib/pig-udf.jar';

DEFINE filterRecords pigudf.FilterUDF;

A = LOAD '/project/lib/result2' USING PigStorage(';') AS (district:chararray, bpl\_performance:double, bpl\_objective:double);

result\_eighty\_percent = FOREACH A GENERATE filterRecords(bpl\_performance, bpl\_objective, district);

dump result\_eighty\_percent;

STORE result\_eighty\_percent into '/project/lib/result\_eighty\_percent' using PigStorage(';');