**Problem statement:**

1. Load data into HDFS using HDFS client.

2. Develop MR programto parse logs and convert request string into structured format (/a/b/c/d =>

a b c d).

50.57.190.149 - - 22/Apr/2012:07:12:41 +0530 GET /computers/laptops.html?brand=819 HTTP/1.0

computers - -laptops.html brand=819 200 12530 - -

Develop MR/Pig/Hive program to extract data for the following KPIs.

3. Count of page views by individual user

4. Top / Bottom 5: catagery-1/ catagery-2 / page /users / entry pages (Exclude status code other

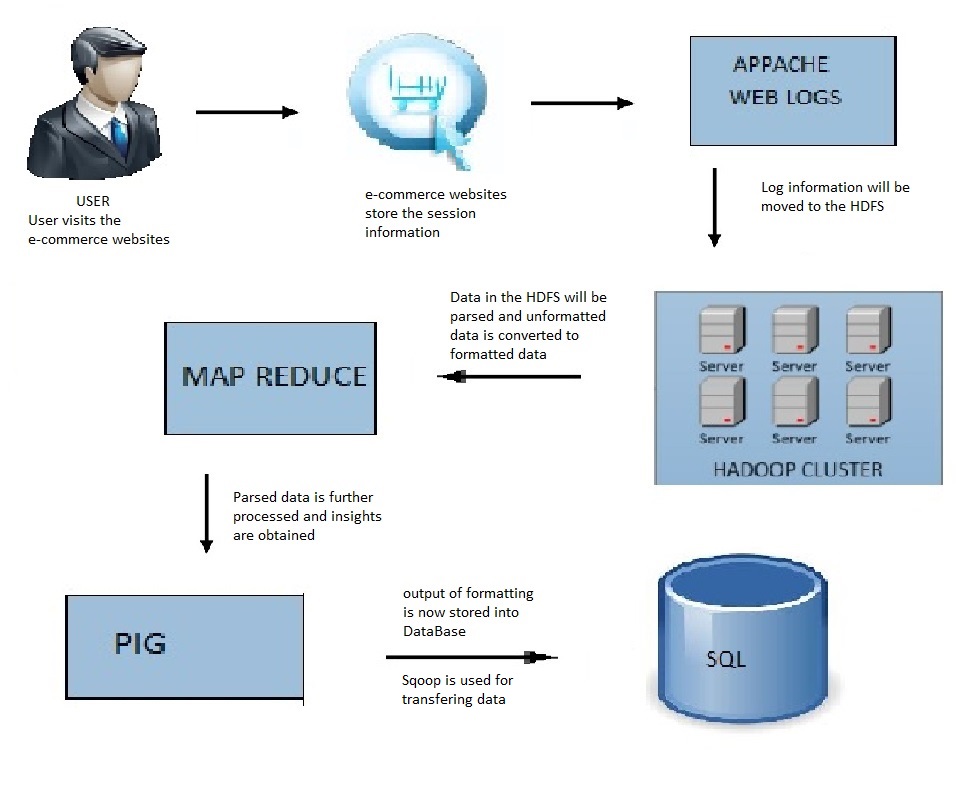
than 200, also exclude record related to css/js/image)

5. Total page views / Category wise pageviews / Unique pageviews.

6. Count of status code = 200 / 404 / 400 / 500.

7. Load results into tables in MySql Database using Sqoop.

**Solution Architecture:**

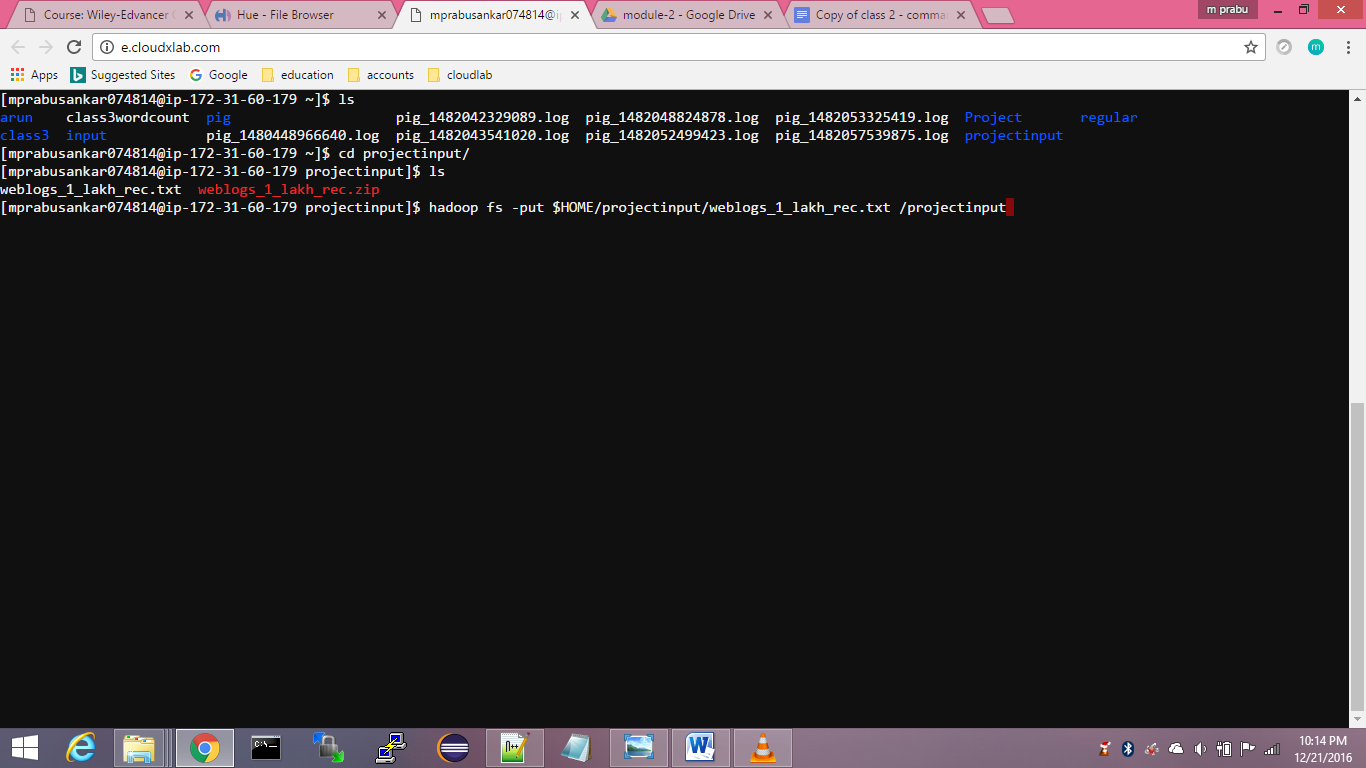


**Software/Tools specification:**

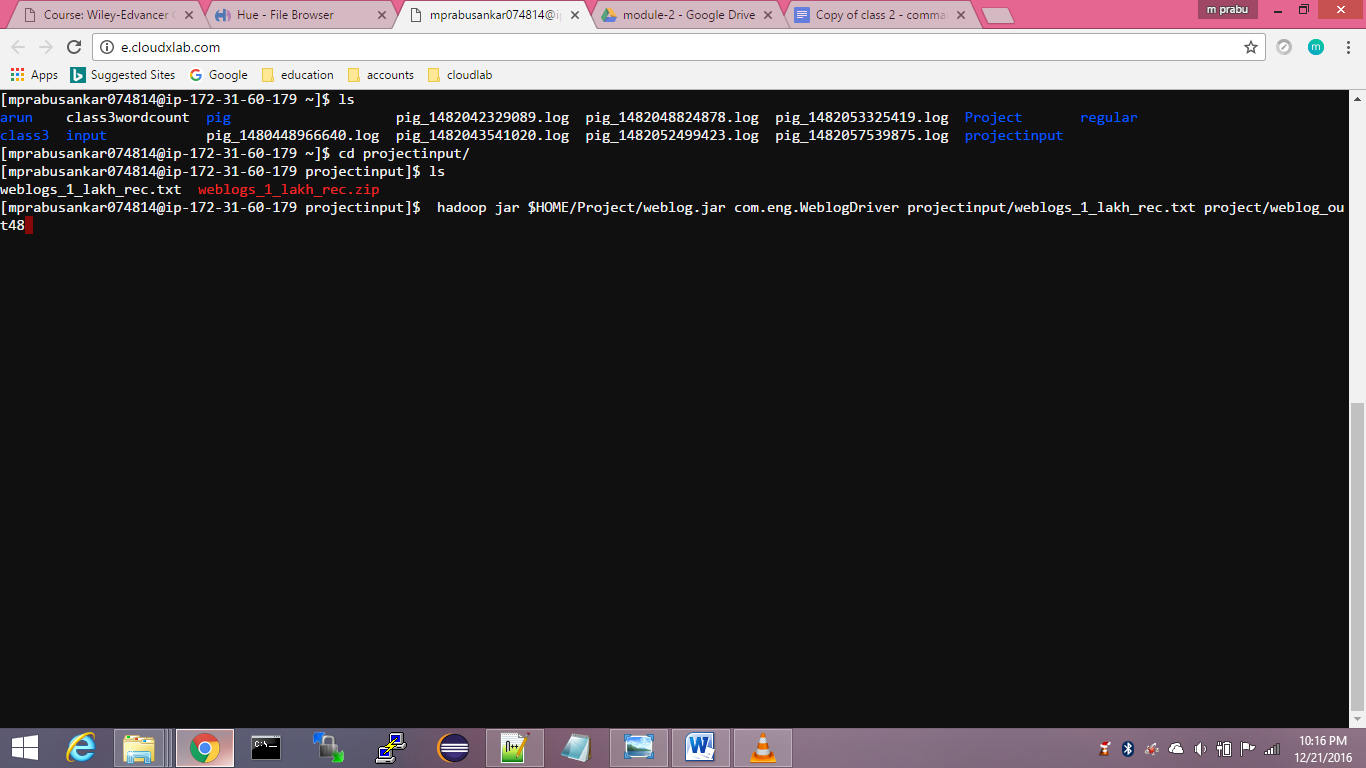
* HDFS
* MAPREDUCE
* PIG
* SQOOP
* SQL

**Solution Description:**

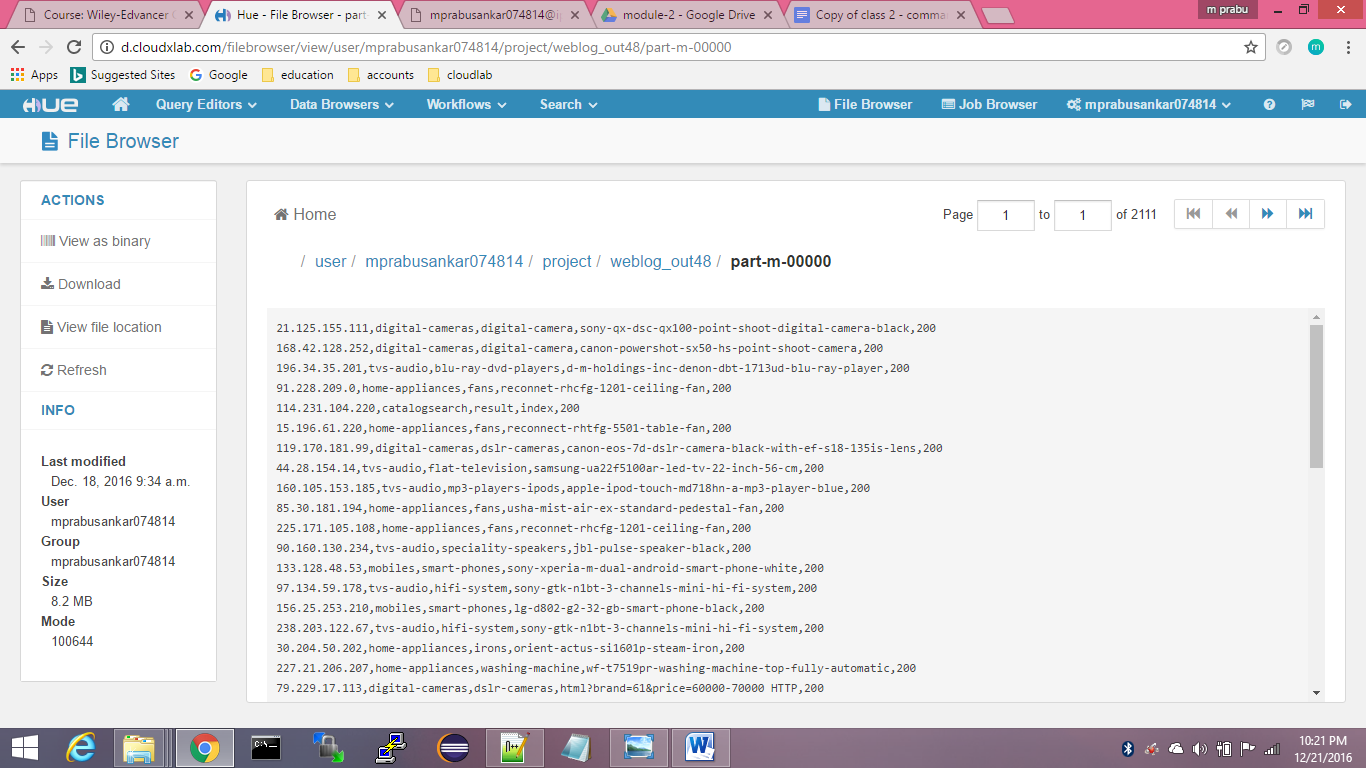
Storing the weblogs into HDFS



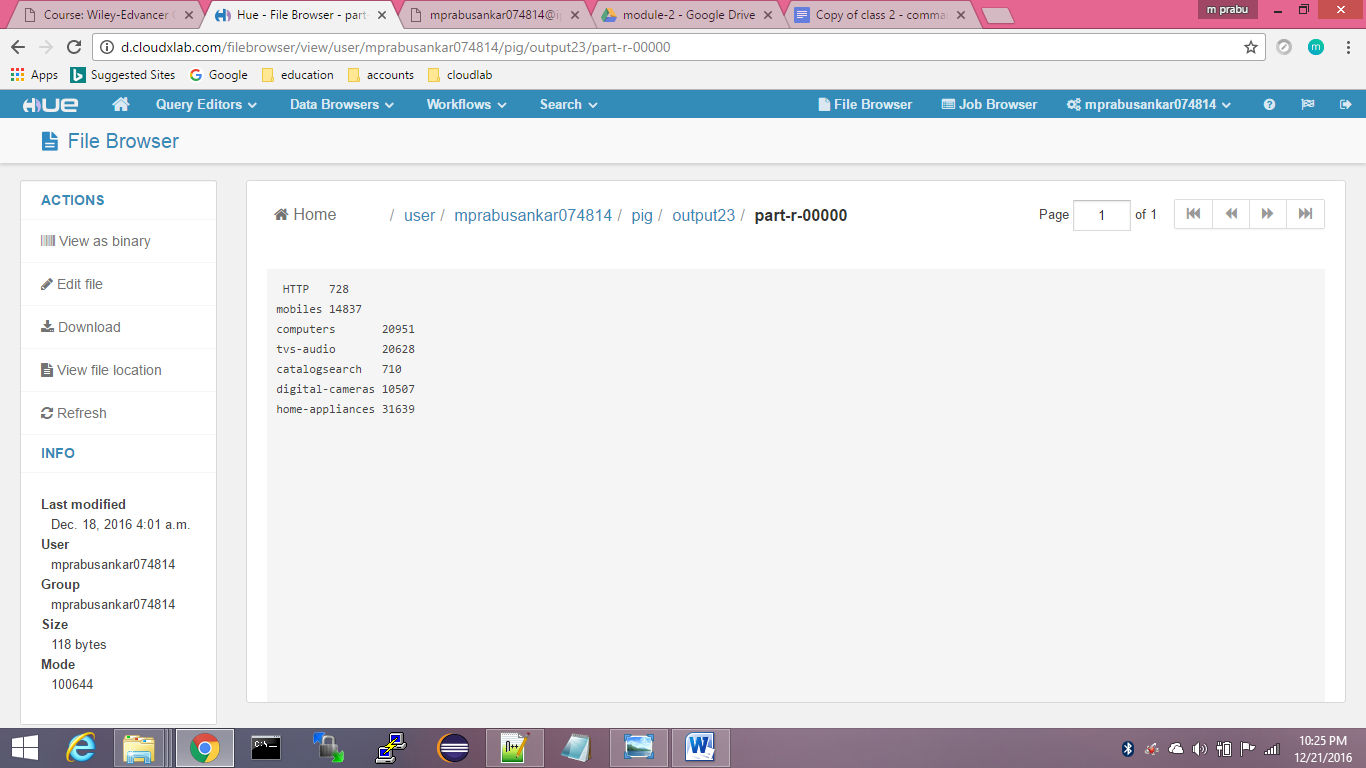
Generating MapReduce for parsing the log files



**Outputs:**







**Coding:**

**Mapper:**

package com.eng;

import java.io.IOException;

import java.util.regex.\*;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.NullWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class WeblogMapper extends Mapper<LongWritable, Text, NullWritable, Text> {

protected void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

String reExpression = "^([\\d.]+) (\\S+) (\\S+) \\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(.+?)\" (\\d{3}) (\\d+) \"([^\"]+)\" \"([^\"]+)\"";

String hostAddr,prodLink,statusCode,primCategory,seconCategory,hostLink,temp,temp1,line;

Pattern pat = Pattern.compile(reExpression);

Matcher matcher = pat.matcher(value.toString());

if (!matcher.matches() ) {

System.err.println("error");

return;

}

hostAddr = matcher.group(1);

prodLink= matcher.group(5);

statusCode= matcher.group(6);

temp = prodLink.replaceAll("/", ",");

temp1= temp.replace(".", ",");

line = temp1.toString();

String[] parts= line.split(",");

primCategory= parts[1];

seconCategory= parts[2];

hostLink=parts[3];

String result = hostAddr +"\t"+primCategory +"\t"+seconCategory+"\t"+hostLink+ "\t" + statusCode;

context.write(NullWritable.get(), new Text(result));

}

}

**Driver:**

package com.eng;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WeblogDriver {

public static void main(String[] args) throws IOException,

InterruptedException, ClassNotFoundException {

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Web Log");

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

job.setJarByClass(WeblogDriver.class);

job.setMapperClass(WeblogMapper.class);

job.setNumReduceTasks(0);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

System.out.println(job.waitForCompletion(true));

}

}

**Pig Script 1:**

A= LOAD 'pig/output7/part-m-00000' using PigStorage() as (host:chararray,webLink:chararray);

B= GROUP A by webLink;

C = FOREACH B GENERATE group,COUNT(A);

STORE C into 'pig/output20/';

**Pig Script 2:**

A= LOAD 'project/weblog\_out47/part-m-00000' using PigStorage() as (host:chararray,primaryCategory:chararray,secondaryCategory:chararray,webLink:chararray,status:int);

B= FOREACH A GENERATE $0 as host,$2 as secondarycategory;

C= GROUP B by secondarycategory;

D= FOREACH C GENERATE group,COUNT(B);

STORE D into 'pig/output23/';

**Pig Script 3:**

A= LOAD 'project/weblog\_out47/part-m-00000' using PigStorage() as (host:chararray,primaryCategory:chararray,secondaryCategory:chararray,webLink:chararray,status:int);

B= FOREACH A GENERATE $0 as host,$1 as category;

C= GROUP B by category;

D= FOREACH C GENERATE group,COUNT(B);

STORE D into 'pig/output32/';

**Pig Script 4:**

A= LOAD 'project/weblog\_out47/part-m-00000' using PigStorage() as (host:chararray,primaryCategory:chararray,secondaryCategory:chararray,webLink:chararray,status:int);

B= FOREACH A GENERATE $0 as host,$4 as status;

C= GROUP B by status;

D= FOREACH C GENERATE group,COUNT(B);

STORE D into 'pig/output25/';

**Pig Script 5:**

A= LOAD 'pig/output29/' using PigStorage() as (host:chararray,count:int);

B= ORDER A by count DESC;

C= LIMIT B 5;

STORE C into 'pig/output34/';

**Pig Script 6:**

A= LOAD 'pig/output32/' using PigStorage() as (secondaryCategory:chararray,count:int);

B= ORDER A by count DESC;

C= LIMIT B 5;

STORE C into 'pig/output33/';

**Pig Script 7:**

A= LOAD 'project/weblog\_out47/part-m-00000' using PigStorage() as (host:chararray,primaryCategory:chararray,secondaryCategory:chararray,webLink:chararray,status:int);

B= FOREACH A GENERATE $0 as host,$3 as webLink;

C= GROUP B by host;

D= FOREACH C GENERATE group,COUNT(B);

STORE D into 'pig/output29/';

**Pig Script 8:**

A= LOAD 'pig/output28/' using PigStorage() as (webLink:chararray,count:int);

B= ORDER A by count DESC;

C= LIMIT B 5;

STORE C into 'pig/output30/';

**Pig Script 9:**

A= LOAD 'pig/output23/' using PigStorage() as (primaryCategory:chararray,count:int);

B= ORDER A by count DESC;

C= LIMIT B 5;

STORE C into 'pig/output31/';

**Conclusion:**

The logs were parsed and the corresponding results were obtained.