**Project 2:-**

**Problem Statement:-Develop MR program to parse logs and convert request string into structured format (/a/b/c/d => a b c d)**

**IP**

**Category-1**

**Category-2**

**page**

**status\_code**

**Solution:-**

Created a MAP only job to parse the logs and convert request string into structured format:-

**WeblogMapper class:-**

**package** com.eng.mr;

**import** java.io.IOException;

**import** org.apache.hadoop.io.Text;

**import** org.apache.hadoop.mapreduce.Mapper;

**import** java.util.regex.\*;

**public** **class** WeblogMapper **extends** Mapper<Object, Text, Text, Text>

{

**private** Pattern p =

Pattern.*compile*("^([\\d.]+) (\\S+) (\\S+) \\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(.+?)\" (\\d{3}) (\\d+) \"([^\"]+)\" \"([^\"]+)\"");

**private** Text outputKey = **new** Text();

**private** Text outputValue = **new** Text();

@Override

**protected** **void** map(Object key, Text value, Context context) **throws** IOException, InterruptedException

{

String entry = value.toString();

Matcher m = p.matcher(entry);

**if** (!m.matches())

{

**return**;

}

outputKey.set(m.group(1)); //ip

StringBuilder b = **new** StringBuilder();

b.append(',');

String rh []=m.group(5).split(" ");

String category []=rh[1].split("\\/");

**if** (category.length==4)

{

b.append(category[1]);//category 1 53.49.121.225

b.append(',');

b.append(category[2]);//category 2

b.append(',');

b.append(category[3]);//page

}

**else** **if** (category.length==3)

{

b.append(category[1]);//category 1 53.49.121.225

b.append(',');

b.append(" ");//page

b.append(',');

b.append(category[2]);//category 2

}

**else** **if** (category.length==2)

{

b.append(" ");//category 1 53.49.121.225

b.append(',');

b.append(" ");//category 2

b.append(',');

b.append(category[1]);//page

}

**if** (category.length==5)

{

b.append(category[1]);//category 1 53.49.121.225

b.append(',');

b.append(category[2]);//category 2

b.append(',');

b.append(category[4]);//page

}

**else** **if** (category.length<=1)

{

b.append(" ");//Category 1

b.append(',');

b.append(" ");//Category 2

b.append(',');

b.append(" ");//page

}

b.append(',');

b.append(m.group(6)); //code

outputValue.set(b.toString());

context.write(outputKey, outputValue);

}

}

**WeblogDriver class:-**

package com.eng.mr;

import java.io.IOException;

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 {

Job job = new Job();

job.setJobName("Weblog");

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(Text.class);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

String input=args[0];

String outpath = args[1];

FileInputFormat.setInputPaths(job, new Path(input));

FileOutputFormat.setOutputPath(job, new Path(outpath));

job.setJarByClass(WeblogDriver.class);

job.setMapperClass(WeblogMapper.class);

job.setNumReduceTasks(0);

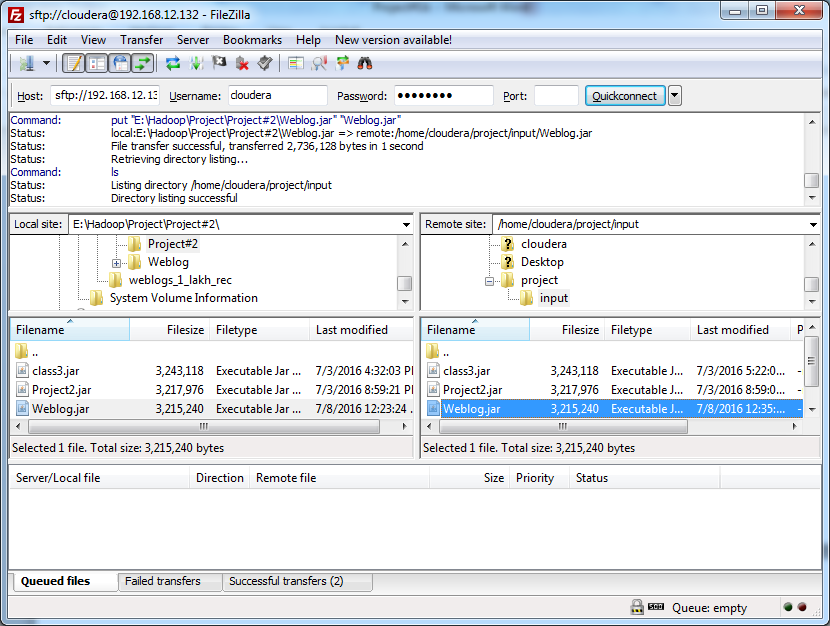
//job.setReducerClass(WordCountReducer.class);

job.waitForCompletion(true);

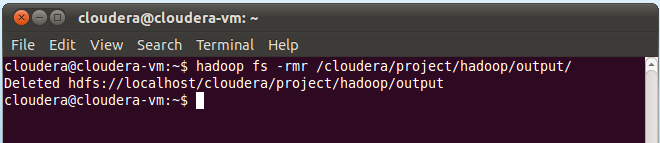
}

}

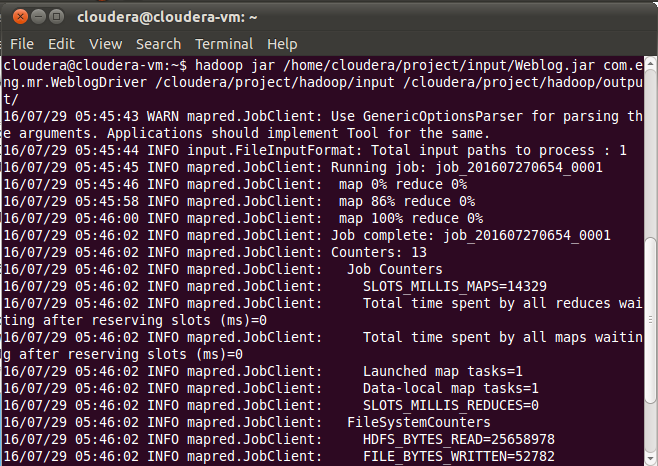
**Transfer of jar file to the VM:-**



**Deleting old output file:-**



**Running MR Program:-**



**Output file:-**

