**KIMMI KUMARI**

**Experiment 4: Web log analysis**

**// Dataset to be used. Save this dataset with “log\_data.txt” name**

**IP Address Timestamp of request HTTP request Status Code Size in Bytes**

96.7.8.17 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

96.7.1.14 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

96.7.2.14 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

192.168.1.1 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

96.7.4.16 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

91.75.5.14 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

98.21.6.14 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

162.15.16.1 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

8.8.8.8 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

10.99.99.247 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

96.7.1.14 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

96.7.1.14 [24/Apr/2011:04:20:11 -0400] "GET /cat.jpg HTTP/1.1" 200 12433

**//Mapreduce code**

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

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

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

public class LogAnalysis {

public static class LogMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

private final static IntWritable one = new IntWritable(1);

private Text ipAddress = new Text();

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

String[] parts = value.toString().split(" ");

if (parts.length >= 1) {

ipAddress.set(parts[0]);

context.write(ipAddress, one);

}

}

}

public static class LogReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

private IntWritable result = new IntWritable();

public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {

int sum = 0;

for (IntWritable val : values) {

sum += val.get();

}

result.set(sum);

context.write(key, result);

}

}

public static void main(String[] args) throws Exception {

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "log analysis");

job.setJarByClass(LogAnalysis.class);

job.setMapperClass(LogMapper.class);

job.setCombinerClass(LogReducer.class);

job.setReducerClass(LogReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

FileInputFormat.addInputPath(job, new Path(args[0])); // Input file path

FileOutputFormat.setOutputPath(job, new Path(args[1])); // Output directory path

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

**//Execute the Jar Code**

hadoop jar LogAnalysis.jar LogAnalysis /home/cloudera/log\_data.txt loganalysisresult

**//Output**

The output will typically be in the form of text files, with one line per IP address, showing the IP address and the count of requests made by that IP address.

8.8.8.8 1

10.99.99.247 1

91.75.5.14 1

96.7.1.14 2

96.7.2.14 1

96.7.4.16 1

96.7.8.17 1

98.21.6.14 1

192.168.1.1 1

162.15.16.1 1