```
package Access;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class ALMapper extends Mapper <LongWritable,Text,Text,IntWritable> {

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

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

		con.write(new Text(Log[0]),new IntWritable(1));
	}
}

package Access;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class ALReducer extends Reducer <Text,IntWritable,Text,IntWritable> {

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

		int sum=0;

		//find sum of occurrences
		for( IntWritable value:values) {
			sum+=value.get();
		}


		con.write(t, new IntWritable(sum));
	}
}


package Access;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FileStatus;
import org.apache.hadoop.fs.FileSystem;
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.output.FileOutputFormat;

public class ALDriver {

        @SuppressWarnings("deprecation")
        public static void main(String[] args) throws IOException, ClassNotFoundException,
InterruptedException {


                Configuration conf=new Configuration();
                Job job=new Job(conf);

                job.setJarByClass(ALDriver.class);
                job.setMapperClass(ALMapper.class);
                job.setReducerClass(ALReducer.class);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(IntWritable.class);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(IntWritable.class);


                FileInputFormat.addInputPath(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job, new Path(args[1]));

                 job.waitForCompletion(true);

                   FileSystem fs=FileSystem.get(conf);
                   FileStatus[] status=fs.listStatus(new Path("hdfs://localhost:9000"+args[1]));
                        FSDataInputStream fd=fs.open(status[1].getPath());
                        System.out.println(status);

                int max=0;
                String ip="";
                String str=fd.readLine();
                do {
                        String parts[]=str.split("        ");
                        //find most occurred IP
                        if(max<Integer.parseInt(parts[1])) {

                                max=Integer.parseInt(parts[1]);
                                ip=parts[0];
                        }
                str=fd.readLine();


                 }while(str != null);

                System.out.println("IP address: " + ip);
                System.out.println("No. of occurrences: " + max);

        }
}
```