# Amazon Reviews Aggregation

# (Summarization Pattern)

This map reduce job calculates the number of reviews per product. WE are using Reducer as a combiner. Since this operation is associative and commutative.

# OUTPUT:

./hadoop jar /home/sayali/Desktop/AmazonReviewsAggregate.jar
sayali.AmazonReviews5.AmazonReviewsAggregate /AmazonReviews/AmazonReviews.tsv
/AmazonReviewsAggregate
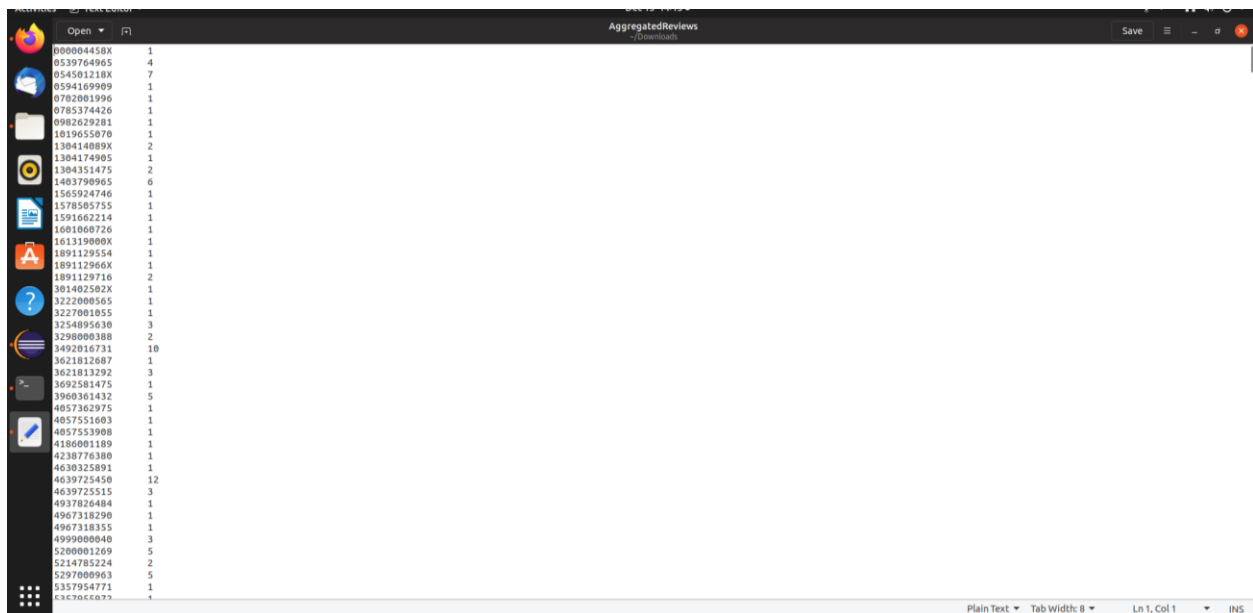


# Map Reduce Code:

```java
package sayali.AmazonReviews5;
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.FloatWritable;
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;
```

```java
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
```

# Mapper:

```java
public class AmazonReviewsAggregateMapper extends Mapper<LongWritable,Text,Text,IntWritable>{

        private static final IntWritable one = new IntWritable(1);
        @Override
        protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context)
                        throws IOException, InterruptedException {
                if(key.get()==0){
                        return;

                } else {

                        String[] line = value.toString().split("\\t");
                        String productId = line[3].trim();
                        context.write(new Text(productId), one); }
        }
}
```

# Reducer:

```java
public class AmazonReviewsAggregateReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
@Override
protected void reduce(Text key, Iterable<IntWritable> values,
                        Reducer<Text, IntWritable, Text, IntWritable>.Context context) throws
IOException, InterruptedException {
                int sum = 0;
                for (IntWritable value : values) {
                        sum += value.get();
                }
                context.write(key, new IntWritable(sum));
        }}
```

# Main:

```java
public class AmazonReviewsAggregate {
        public static void main(String[] args) throws IOException {

                Configuration conf = new Configuration();
                // Create a new Job
                Job job = Job.getInstance(conf, "AmazonReviewsAggregation");
                job.setJarByClass(AmazonReviewsAggregate.class);
                // set the mapper and reducer
```

```java
job.setMapperClass(AmazonReviewsAggregateMapper.class);
job.setReducerClass(AmazonReviewsAggregateReducer.class);

// set the format of mapper and reducer
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);

// set the key nd value format of the output
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);

// set the output and input format
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
// Submit the job, then poll for progress until the job is complete
try {
        job.waitForCompletion(true);
} catch (ClassNotFoundException e) {

        e.printStackTrace();
} catch (InterruptedException e) {

        e.printStackTrace();
}
    }
}
```