Session 3

Assignment 3

|  |  |
| --- | --- |
| **Prepared For:** | AcadGild |
|  |  |
| **Document Approval:** | **AcadGild** |
|  |  |
|  |  |
|  |  |
|  |  |
| **Project Title:** | Session 3 – Assignment 3 |
|  |  |
| **Prepared By:** | Duncan Burgess |
|  |  |
|  | dburgess@duncb.com |
|  |  |
| **Primary Engineer:** | Duncan Burgess |
|  |  |
| **Document Reference:** | **Session 3 – Assignment 3** |
|  |  |
| **Start Date:** | 02/09/2017 |
|  |  |
|  |  |

# 

# Contents

[Contents 2](#_Toc493044446)

[Change History 3](#_Toc493044447)

[1. Problem Statement 4](#_Toc493044448)

[2. The input file (television.txt) 4](#_Toc493044449)

[3. Solutions 3.1 Calculate the total units sold for each Company 5](#_Toc493044450)

[4. Running Job 3.1 9](#_Toc493044451)

[5. Check for Results 3.1 10](#_Toc493044452)

[6. View Results 3.1 10](#_Toc493044453)

[7. Solutions 3.2 Calculate the total units sold in each state for Onida Company 11](#_Toc493044454)

[8. Running Job 3.2 15](#_Toc493044455)

[9. Check for Results 3.2 16](#_Toc493044456)

[10. View Results 3.2 16](#_Toc493044457)

# Change History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Revision** | **Date** | **Authored By** | **Authorised By** | **Sections Affected** | **Reason for Change** |
| Rev 01 | 02/09/2017 | Duncan Burgess |  | All | Initial release. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Problem Statement

Records look like:

Samsung|Optima|14|Madhya Pradesh|132401|14200

The fields are arranged like:

Company Name|Product Name|Size in inches|State|Pin Code|Price

There are some invalid records which contain 'NA' in either Company Name or Product Name.

3.1 Write a Map Reduce program to calculate the total units sold for each Company.

3.2 Write a Map Reduce program to calculate the total units sold in each state for Onida Company.

# The input file (television.txt)

Samsung|Optima|14|Madhya Pradesh|132401|14200

Onida|Lucid|18|Uttar Pradesh|232401|16200

Akai|Decent|16|Kerala|922401|12200

Lava|Attention|20|Assam|454601|24200

Zen|Super|14|Maharashtra|619082|9200

Samsung|Optima|14|Madhya Pradesh|132401|14200

Onida|Lucid|18|Uttar Pradesh|232401|16200

Onida|Decent|14|Uttar Pradesh|232401|16200

Onida|NA|16|Kerala|922401|12200

Lava|Attention|20|Assam|454601|24200

Zen|Super|14|Maharashtra|619082|9200

Samsung|Optima|14|Madhya Pradesh|132401|14200

NA|Lucid|18|Uttar Pradesh|232401|16200

Samsung|Decent|16|Kerala|922401|12200

Lava|Attention|20|Assam|454601|24200

Samsung|Super|14|Maharashtra|619082|9200

Samsung|Super|14|Maharashtra|619082|9200

Samsung|Super|14|Maharashtra|619082|9200

# Solutions 3.1 Calculate the total units sold for each Company

The required jar file was created in Eclipse.

Driver SalesOfTVDriver

Mapper SalesOfTVMapper

Reducer SalesOfTVReducer

The file and the jar file are copied to hdfs using hdfs fs –copyFromLocal.

**SalesOfTVDriver**

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 OnidaSales

{

@SuppressWarnings("deprecation")

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

{

//Check if input parameters provided appropriately

if(args==null || args.length!=2)

{

System.err.println("Incorrect jar or output directory provided");

System.exit(-1);

}

//Instantiate configuration object

Configuration conf = new Configuration();

//Instantiate job object

Job job = new Job(conf,"OnidaSales");

job.setJarByClass(OnidaSales.class);

/\*

\* Set input pathput

\*/

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

/\*

\* Set output path

\*/

Path outputPath = new Path(args[1]);

FileOutputFormat.setOutputPath(job, outputPath);

//Delete output directory if already existing will fail if present

outputPath.getFileSystem(conf).delete(outputPath, true);

//Set mapper class

job.setMapperClass(OnidaSalesMapper.class);

job.setReducerClass(OnidaSalesReducer.class);

//Set input and output format class

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

//Set output key'value class types

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

//Execute the job and wait until completion and then exit

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

}

}

**SalesOfTVMapper**

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

/\*

\* Input key is the offset of line in file

\* Input value format : CompanyName|ProductName|SizeInInches|State|PinCOde|Prize

\* Output key is the the name of the company

\* Output value is 1

\*/

public class SalesOfTVMapper extends Mapper<LongWritable,Text,Text,IntWritable>

{

private final static String DELIMITER = "|";

private final static String NA = "NA";

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

Text companyName = new Text();

@Override

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

{

String strValue = value.toString();

System.out.println("Current value is "+strValue);

StringTokenizer tokenizer = new StringTokenizer(strValue, DELIMITER);

String strCompanyName = tokenizer.nextToken();

String strProductName = tokenizer.nextToken();

//Check if CompanyName or Product Name is 'NA'

if(!strCompanyName.equalsIgnoreCase(NA) && !strProductName.equalsIgnoreCase(NA))

{

companyName.set(strCompanyName.trim());

context.write(companyName,ONE);

System.out.println("Putting key : "+strCompanyName+" and value : 1 in context");

}

}

}

**SalesOfTVReducer**

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.Reducer;

public class SalesOfTVReducer extends Reducer<Text,IntWritable,Text,LongWritable>

{

@Override

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

{

long lCount = 0;

for(IntWritable value:values)

{

lCount=lCount + value.get();

}

context.write(key, new LongWritable(lCount));

}

}

# Running Job 3.1

[acadgild@localhost Documents]$ yarn jar TVSales.jar television.txt tv\_sales

17/09/03 14:16:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

17/09/03 14:16:55 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032

17/09/03 14:16:56 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.

17/09/03 14:16:56 INFO input.FileInputFormat: Total input paths to process : 1

17/09/03 14:16:56 INFO mapreduce.JobSubmitter: number of splits:1

17/09/03 14:16:56 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1504029604882\_0017

17/09/03 14:16:57 INFO impl.YarnClientImpl: Submitted application application\_1504029604882\_0017

17/09/03 14:16:57 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application\_1504029604882\_0017/

17/09/03 14:16:57 INFO mapreduce.Job: Running job: job\_1504029604882\_0017

17/09/03 14:17:03 INFO mapreduce.Job: Job job\_1504029604882\_0017 running in uber mode : false

17/09/03 14:17:03 INFO mapreduce.Job: map 0% reduce 0%

17/09/03 14:17:08 INFO mapreduce.Job: map 100% reduce 0%

17/09/03 14:17:14 INFO mapreduce.Job: map 100% reduce 100%

17/09/03 14:17:14 INFO mapreduce.Job: Job job\_1504029604882\_0017 completed successfully

17/09/03 14:17:14 INFO mapreduce.Job: Counters: 49

File System Counters

FILE: Number of bytes read=204

FILE: Number of bytes written=212533

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=848

HDFS: Number of bytes written=38

HDFS: Number of read operations=6

HDFS: Number of large read operations=0

HDFS: Number of write operations=2

Job Counters

Launched map tasks=1

Launched reduce tasks=1

Data-local map tasks=1

Total time spent by all maps in occupied slots (ms)=3067

Total time spent by all reduces in occupied slots (ms)=3181

Total time spent by all map tasks (ms)=3067

Total time spent by all reduce tasks (ms)=3181

Total vcore-seconds taken by all map tasks=3067

Total vcore-seconds taken by all reduce tasks=3181

Total megabyte-seconds taken by all map tasks=3140608

Total megabyte-seconds taken by all reduce tasks=3257344

Map-Reduce Framework

Map input records=18

Map output records=16

Map output bytes=166

Map output materialized bytes=204

Input split bytes=115

Combine input records=0

Combine output records=0

Reduce input groups=5

Reduce shuffle bytes=204

Reduce input records=16

Reduce output records=5

Spilled Records=32

Shuffled Maps =1

Failed Shuffles=0

Merged Map outputs=1

GC time elapsed (ms)=124

CPU time spent (ms)=1400

Physical memory (bytes) snapshot=417820672

Virtual memory (bytes) snapshot=4161044480

Total committed heap usage (bytes)=343932928

Shuffle Errors

BAD\_ID=0

CONNECTION=0

IO\_ERROR=0

WRONG\_LENGTH=0

WRONG\_MAP=0

WRONG\_REDUCE=0

File Input Format Counters

Bytes Read=733

File Output Format Counters

Bytes Written=38

# Check for Results 3.1

[[acadgild@localhost Documents]$ hadoop fs -ls tv\_sales

17/09/03 14:18:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 2 items

-rw-r--r-- 1 acadgild supergroup 0 2017-09-03 14:17 tv\_sales/\_SUCCESS

-rw-r--r-- 1 acadgild supergroup 38 2017-09-03 14:17 tv\_sales/part-r-00000

Note r - indicates it is a reduced job.

# View Results 3.1

[acadgild@localhost Documents]$ hadoop fs -cat tv\_sales/part-r-00000

17/09/03 14:18:37 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Akai 1

Lava 3

Onida 3

Samsung 7

Zen 2

# Solutions 3.2 Calculate the total units sold in each state for Onida Company

The required jar file was created in Eclipse.

Driver onidasalesdriver

Mapper onidasalesmapper

Reducer onidasalesreducer

The file and the jar file are copied to hdfs using hdfs fs –copyFromLocal.

**Onidasalesdriver**

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 OnidaSales

{

@SuppressWarnings("deprecation")

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

{

//Check if input parameters provided appropriately

if(args==null || args.length!=2)

{

System.err.println("Incorrect jar or output directory provided");

System.exit(-1);

}

//Instantiate configuration object

Configuration conf = new Configuration();

//Instantiate job object

Job job = new Job(conf,"OnidaSales");

job.setJarByClass(OnidaSales.class);

/\*

\* Set input pathput

\*/

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

/\*

\* Set output path

\*/

Path outputPath = new Path(args[1]);

FileOutputFormat.setOutputPath(job, outputPath);

//Delete output directory if already existing will fail if present

outputPath.getFileSystem(conf).delete(outputPath, true);

//Set mapper class

job.setMapperClass(OnidaSalesMapper.class);

job.setReducerClass(OnidaSalesReducer.class);

//Set input and output format class

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

//Set output key'value class types

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

//Execute the job and wait until completion and then exit

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

}

}

**onidasalesmapper**

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

/\*

Input key is the offset of line in file

Input value format : Company|Product|Size|State|Zip|Price

Output key is the the name of the state for Onida

Output value is 1

\*/

public class OnidaSalesMapper extends Mapper<LongWritable,Text,Text,IntWritable>

{

private final static String DELIMITER = "|";

private final static String NA = "NA";

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

Text stateName = new Text();

private final static String ONIDA = "ONIDA";

@Override

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

{

String strValue = value.toString();

System.out.println("Current value is "+strValue);

StringTokenizer tokenizer = new StringTokenizer(strValue, DELIMITER);

String strCompanyName = tokenizer.nextToken();

String strProductName = tokenizer.nextToken();

//Check if CompanyName or Product Name is 'NA'

if(!strCompanyName.equalsIgnoreCase(NA) && !strProductName.equalsIgnoreCase(NA))

{

if(strCompanyName.equalsIgnoreCase(ONIDA))

{

tokenizer.nextToken();

String strStateName = tokenizer.nextToken();

stateName.set(strStateName.trim());

context.write(stateName,ONE);

System.out.println("Putting key : "+strCompanyName+" and value : 1 in context");

}

}

}

}

**onidasalesreducer**

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.Reducer;

/\*\*

\* Input key is the the name of the state

\* Input value is 1

\* Output key is the name of the state

\* OUtput value is the number of televisions in the state of Onida

\*

\*/

public class OnidaSalesReducer extends Reducer<Text,IntWritable,Text,LongWritable>

{

@Override

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

{

long lCount = 0;

for(IntWritable value:values)

{

lCount=lCount + value.get();

}

context.write(key, new LongWritable(lCount));

}

}

# Running Job 3.2

[acadgild@localhost Documents]$ yarn jar OnidaSales.jar television.txt Onida\_Sales

17/09/03 12:36:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classs where applicable

17/09/03 12:36:25 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032

17/09/03 12:36:26 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface an execute your application with ToolRunner to remedy this.

17/09/03 12:36:26 INFO input.FileInputFormat: Total input paths to process : 1

17/09/03 12:36:26 INFO mapreduce.JobSubmitter: number of splits:1

17/09/03 12:36:26 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1504029604882\_0015

17/09/03 12:36:26 INFO impl.YarnClientImpl: Submitted application application\_1504029604882\_0015

17/09/03 12:36:27 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application\_1504029604882\_0015/

17/09/03 12:36:27 INFO mapreduce.Job: Running job: job\_1504029604882\_0015

17/09/03 12:36:33 INFO mapreduce.Job: Job job\_1504029604882\_0015 running in uber mode : false

17/09/03 12:36:33 INFO mapreduce.Job: map 0% reduce 0%

17/09/03 12:36:39 INFO mapreduce.Job: map 100% reduce 0%

17/09/03 12:36:45 INFO mapreduce.Job: map 100% reduce 100%

17/09/03 12:36:45 INFO mapreduce.Job: Job job\_1504029604882\_0015 completed successfully

17/09/03 12:36:45 INFO mapreduce.Job: Counters: 49

File System Counters

FILE: Number of bytes read=66

FILE: Number of bytes written=212261

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=848

HDFS: Number of bytes written=16

HDFS: Number of read operations=6

HDFS: Number of large read operations=0

HDFS: Number of write operations=2

Job Counters

Launched map tasks=1

Launched reduce tasks=1

Data-local map tasks=1

Total time spent by all maps in occupied slots (ms)=3438

Total time spent by all reduces in occupied slots (ms)=3428

Total time spent by all map tasks (ms)=3438

Total time spent by all reduce tasks (ms)=3428

Total vcore-seconds taken by all map tasks=3438

Total vcore-seconds taken by all reduce tasks=3428

Total megabyte-seconds taken by all map tasks=3520512

Total megabyte-seconds taken by all reduce tasks=3510272

Map-Reduce Framework

Map input records=18

Map output records=3

Map output bytes=54

Map output materialized bytes=66

Input split bytes=115

Combine input records=0

Combine output records=0

Reduce input groups=1

Reduce shuffle bytes=66

Reduce input records=3

Reduce output records=1

Spilled Records=6

Shuffled Maps =1

Failed Shuffles=0

Merged Map outputs=1

GC time elapsed (ms)=183

CPU time spent (ms)=2090

Physical memory (bytes) snapshot=423550976

Virtual memory (bytes) snapshot=4156125184

Total committed heap usage (bytes)=347078656

Shuffle Errors

BAD\_ID=0

CONNECTION=0

IO\_ERROR=0

WRONG\_LENGTH=0

WRONG\_MAP=0

WRONG\_REDUCE=0

File Input Format Counters

Bytes Read=733

File Output Format Counters

Bytes Written=16

# Check for Results 3.2

[acadgild@localhost Documents]$ hadoop fs -ls Onida\_Sales

17/09/03 12:37:23 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classs where applicable

Found 2 items

-rw-r--r-- 1 acadgild supergroup 0 2017-09-03 12:36 Onida\_Sales/\_SUCCESS

-rw-r--r-- 1 acadgild supergroup 16 2017-09-03 12:36 Onida\_Sales/part-r-00000

[acadgild@localhost Documents]$ hadoop fs -cat Onida\_Sales/part-r-00000

Note r - indicates it is a reduced job.

# View Results 3.2

17/09/03 12:37:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classs where applicable

Uttar Pradesh 3