

Chapter4 Project to be done in local machine - all steps taken

Installing hadoop local machine

```
...624d542a25 — @sandbox-hdf:/ — com.docker.cli < sudo | ~ — -bash | +
```

```
Reset branch 'master'
Your branch is up to date with 'origin/master'.

==> Fetching /usr/local/Homebrew/Library/Taps/homebrew/homebrew-core...
==> Resetting /usr/local/Homebrew/Library/Taps/homebrew/homebrew-core...
Branch 'master' set up to track remote branch 'master' from 'origin'.
Reset branch 'master'

[Sandras-MacBook-Pro:~ sandra$ brew cask doctor
Error: Unknown command: cask
[Sandras-MacBook-Pro:~ sandra$ brew tap homebrew/cask-versions
[Sandras-MacBook-Pro:~ sandra$ brew upgrade
[Sandras-MacBook-Pro:~ sandra$ brew install hadoop
==> Downloading https://ghcr.io/v2/homebrew/core/openjdk/manifests/15.0.2
#####
  100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/openjdk/blobs/sha256:fca110fb6caad1228156b587a3ca9fa
==> Downloading from https://pkg-containers-az.githubusercontent.com/ghcr1/blobs/sha256:fca110fb6caad
#####
  100.0%
==> Downloading https://www.apache.org/dyn/closer.lua?path=hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tgz
==> Downloading from https://downloads.apache.org/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tgz
#####
  100.0%
==> Installing dependencies for hadoop: openjdk
==> Installing hadoop dependency: openjdk
==> Pouring openjdk-15.0.2.catalina.bottle.tar.gz
==> Caveats
For the system Java wrappers to find this JDK, symlink it with
  sudo ln -sfn /usr/local/opt/openjdk/libexec/openjdk.jdk /Library/Java/JavaVirtualMachines/openjdk.jdk

openjdk is keg-only, which means it was not symlinked into /usr/local,
because macOS provides similar software and installing this software in
parallel can cause all kinds of trouble.

If you need to have openjdk first in your PATH, run:
  echo 'export PATH="/usr/local/opt/openjdk/bin:$PATH"' >> /Users/sandra/.bash_profile

For compilers to find openjdk you may need to set:
  export CPPFLAGS="-I/usr/local/opt/openjdk/include"

==> Summary
🍺 /usr/local/Cellar/openjdk/15.0.2: 614 files, 324.9MB
==> Installing hadoop
🍺 /usr/local/Cellar/hadoop/3.3.0: 21,819 files, 954.7MB, built in 47 seconds
==> Caveats
==> openjdk
For the system Java wrappers to find this JDK, symlink it with
  sudo ln -sfn /usr/local/opt/openjdk/libexec/openjdk.jdk /Library/Java/JavaVirtualMachines/openjdk.jdk

openjdk is keg-only, which means it was not symlinked into /usr/local,
because macOS provides similar software and installing this software in
parallel can cause all kinds of trouble.

If you need to have openjdk first in your PATH, run:
  echo 'export PATH="/usr/local/opt/openjdk/bin:$PATH"' >> /Users/sandra/.bash_profile

For compilers to find openjdk you may need to set:
  export CPPFLAGS="-I/usr/local/opt/openjdk/include"

Sandras-MacBook-Pro:~ sandra$
```

Installing spark in local machine

```
sandra — java < spark-shell — 101x61
...624d542a25 — @sandbox-hdf:/ — com.docker.cli < sudo | ~ — java < spark-shell

[Sandras-MacBook-Pro:~ sandra$ Java -version
java version "14.0.1" 2020-04-14
Java(TM) SE Runtime Environment (build 14.0.1+7)
Java HotSpot(TM) 64-Bit Server VM (build 14.0.1+7, mixed mode, sharing)
[Sandras-MacBook-Pro:~ sandra$ brew install scala
==> Downloading https://downloads.lightbend.com/scala/2.13.5/scala-2.13.5.tgz
#####
100.0%
==> Caveats
To use with IntelliJ, set the Scala home to:
/usr/local/opt/scala/idea
==> Summary
🍺 /usr/local/Cellar/scala/2.13.5: 42 files, 23.4MB, built in 4 seconds
[Sandras-MacBook-Pro:~ sandra$ scala -version
Scala code runner version 2.13.5 -- Copyright 2002-2020, LAMP/EPFL and Lightbend, Inc.
[Sandras-MacBook-Pro:~ sandra$ brew install apache-spark
==> Downloading https://ghcr.io/v2/homebrew/core/openjdk/11/manifests/11.0.10
#####
100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/openjdk/11/blobs/sha256:6dd0a8c323dd861d68d43b6cce0
==> Downloading from https://pkg-containers-az.githubusercontent.com/ghcr1/blobs/sha256:6dd0a8c323dd
#####
100.0%
==> Downloading https://www.apache.org/dyn/closer.lua?path=spark/spark-3.1.1/spark-3.1.1-bin-hadoop3
==> Downloading from https://downloads.apache.org/spark/spark-3.1.1/spark-3.1.1-bin-hadoop3.2.tgz
#####
100.0%
==> Installing dependencies for apache-spark: openjdk@11
==> Installing apache-spark dependency: openjdk@11
==> Pouring openjdk@11--11.0.10.catalina.bottle.tar.gz
==> Caveats
For the system Java wrappers to find this JDK, symlink it with
  sudo ln -sfn /usr/local/opt/openjdk@11/libexec/openjdk.jdk /Library/Java/JavaVirtualMachines/openjdk-11.jdk
Ma
openjdk@11 is keg-only, which means it was not symlinked into /usr/local,
because this is an alternate version of another formula.

If you need to have openjdk@11 first in your PATH, run:
  echo 'export PATH="/usr/local/opt/openjdk@11/bin:$PATH"' >> /Users/sandra/.bash_profile

For compilers to find openjdk@11 you may need to set:
  export CPPFLAGS="-I/usr/local/opt/openjdk@11/include"

==> Summary
🍺 /usr/local/Cellar/openjdk@11/11.0.10: 654 files, 297.3MB
==> Installing apache-spark
🍺 /usr/local/Cellar/apache-spark/3.1.1: 1,361 files, 242.6MB, built in 9 seconds
==> Caveats
==> openjdk@11
For the system Java wrappers to find this JDK, symlink it with
  sudo ln -sfn /usr/local/opt/openjdk@11/libexec/openjdk.jdk /Library/Java/JavaVirtualMachines/openjdk-11.jdk
Ma
openjdk@11 is keg-only, which means it was not symlinked into /usr/local,
because this is an alternate version of another formula.

If you need to have openjdk@11 first in your PATH, run:
  echo 'export PATH="/usr/local/opt/openjdk@11/bin:$PATH"' >> /Users/sandra/.bash_profile

For compilers to find openjdk@11 you may need to set:
  export CPPFLAGS="-I/usr/local/opt/openjdk@11/include"

[Sandras-MacBook-Pro:~ sandra$ Spark-shell
21/04/16 17:08:27 WARN Utils: Your hostname, Sandras-MacBook-Pro.local resolves to a loopback address: 127.0.0.1; using 172.17.0.2 instead
```

Accessing Spark local machine

Inserting Data in hadoop

```
sandra@Sandras-MacBook-Pro ~ % hdfs dfs -mkdir /Data
2021-04-19 20:39:24,879 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
sandra@Sandras-MacBook-Pro ~ % hdfs dfs -mkdir /Data/LeftOuterJoins
2021-04-19 20:39:39,503 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
sandra@Sandras-MacBook-Pro ~ % hadoop fs -put /usr/local/bin/Data/LeftOuterJoin/users.txt /Data/LeftOuterJoins
2021-04-19 20:40:00,661 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
sandra@Sandras-MacBook-Pro ~ % hadoop fs -cat /Data/LeftOuterJoins/users.txt
2021-04-19 20:40:22,795 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
u1 UT
u2 GA
u3 CA
u4 CA
[u5 GA]
sandra@Sandras-MacBook-Pro ~ % hadoop fs -cat /Data/LeftOuterJoins/transactions.txt
2021-04-19 20:40:38,601 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
cat: '/Data/LeftOuterJoins/transactions.txt': No such file or directory
sandra@Sandras-MacBook-Pro ~ % hadoop fs -put /usr/local/bin/Data/LeftOuterJoin/transactions.txt /Data/LeftOuterJoins
2021-04-19 20:41:09,297 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
sandra@Sandras-MacBook-Pro ~ % hadoop fs -cat /Data/LeftOuterJoins/transactions.txt
2021-04-19 20:41:13,549 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
t1 p3 u1 3 339
t2 p1 u2 1 400
t3 p1 u3 606
t4 p2 u2 10 1000
t5 p4 u4 9 90
t6 p1 u1 4 120
t7 p4 u1 8 160
t8 p4 u5 2 40%
sandra@Sandras-MacBook-Pro ~ % hadoop fs -put ~/Desktop/myLeftOuterJoin.java /Data/LeftOuterJoins
2021-04-19 20:41:32,018 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
sandra@Sandras-MacBook-Pro ~ %
```

Hadoop localhost active

The screenshot shows the HDFS Health Overview page at localhost:9870/dfshealth.html#tab-overview. The top navigation bar includes tabs for Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The Overview tab is selected.

Overview 'localhost:9000' (active)

Started:	Sat Apr 17 10:56:04 -0400 2021
Version:	3.3.0, raa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled:	Mon Jul 06 14:44:00 -0400 2020 by brahma from branch-3.3.0
Cluster ID:	CID-4aa2994f-43ff-46b1-aa7d-3a5627547a1c
Block Pool ID:	BP-750351908-127.0.0.1-1618671291283

Summary

Security is off.
Safemode is off.
6 files and directories, 3 blocks (3 replicated blocks, 0 erasure coded block groups) = 9 total filesystem object(s).
Heap Memory used 103.56 MB of 142 MB Heap Memory. Max Heap Memory is 4 GB.
Non Heap Memory used 66.25 MB of 68.69 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	465.63 GB
Configured Remote Capacity:	0 B
DFS Used:	36 KB (0%)
Non DFS Used:	444.97 GB
DFS Remaining:	2.42 GB (0.52%)
Block Pool Used:	36 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%

Inserting into spark scala

```
[scala] sc.textFile("users.txt")
res20: org.apache.spark.rdd.RDD[String] = users.txt MapPartitionsRDD[7] at textFile at <console>:31
[scala] res20.take(5)
res21: Array[String] = Array(u1 UT, u2 GA, u3 CA, u4 CA, u5 GA)
[scala] sc.textFile("transactions.txt")
res22: org.apache.spark.rdd.RDD[String] = transactions.txt MapPartitionsRDD[9] at textFile at <console>:31
[scala] res22.take(8)
res23: Array[String] = Array(t1 p3 u1 3 330, t2 p1 u2 1 400, t3 p1 u1 3 600, t4 p2 u2 10 1000, t5 p4 u4 9 90, t6 p1 u1 4 120, t7 p4 u1 8 160, t8 p4 u5 2 40)
scala> 
```

Spark local host active :

The Executors page displays the following summary table:

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(1)	0	135.4 KIB / 434.4 MB	0.0 B	8	0	0	4	4	0.2 s (7.0 ms)	223 B	0.0 B	0.0 B	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(1)	0	135.4 KIB / 434.4 MB	0.0 B	8	0	0	4	4	0.2 s (7.0 ms)	223 B	0.0 B	0.0 B	0

Below the summary, there is a detailed table for the single active executor:

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Thread Dump
driver	10.0.0.91:61393	Active	0	135.4 KIB / 434.4 MB	0.0 B	8	0	0	4	4	0.2 s (7.0 ms)	223 B	0.0 B	0.0 B	Thread Dump

Showing 1 to 1 of 1 entries

The Spark Jobs page displays the following information:

- User: sandra
- Total Uptime: 16.3 h
- Scheduling Mode: FIFO
- Event Timeline** (checkbox checked): Enable zooming

Legend for Executors:

- Added (blue square)
- Removed (red square)

Legend for Jobs:

- Succeeded (blue square)
- Failed (red square)
- Running (green square)

Timeline grid showing executor and job status over time (X-axis: 700 to 500, Y-axis: 19:54:58 to 19:55:00).

Looks successful so far:

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
3	take at <console>:32	+details 2021/04/20 13:49:59	10 ms	1/1	60.0 B	0 B	0 B	0 B
2	take at <console>:32	+details 2021/04/20 13:49:59	10 ms	1/1	119.0 B	0 B	0 B	0 B
1	take at <console>:32	+details 2021/04/20 13:49:12	12 ms	1/1	15.0 B	0 B	0 B	0 B
0	take at <console>:32	+details 2021/04/20 13:49:12	0.2 s	1/1	29.0 B	0 B	0 B	0 B

Java Code (attached too)

```
package myId.LeftOuterJoin;
//Step 1 import required classes and interfaces
import scala.Tuple2;
import java.util.Iterator;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.apache.spark.api.java.function.Function;
import org.apache.spark.api.java.function.PairFlatMapFunction;
import org.apache.spark.api.java.function.FlatMapFunction;
import org.apache.spark.api.java.function.PairFunction;
import java.util.Set;
import java.util.HashSet;
import java.util.Iterator;
import java.util.Arrays;
import java.util.List;
import java.util.ArrayList;
import java.util.Collections;
```

```
public class myLeftOuterJoin {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

```

//Step 2 Read input parameters
if(args.length < 2) {
    System.err.println("Usage: LeftOuterJoin <users> <transactions>");
    System.exit(1);
}
String usersInputFile = args[0]; //HDFS text file
String transactionsInputFile = args[1]; //HDFS text file
System.out.println("users="+ usersInputFile);
System.out.println("transactions"+ transactionsInputFile);

//Step 3 create a JavaSparkContext object
JavaSparkContext ctx = new JavaSparkContext();

//Step 4 create a JavaRDD for users
JavaRDD<String> users = ctx.textFile(usersInputFile, 1);

//<K2, V2> JavaPairRDD<K2, V2> mapToPair(PairFunction<T, K2, V2> f)
//Return a new RDD by applying a function to all elements of this RDD.
//PairFunction<t, K, V> where T => Tuple2<K, V>
JavaPairRDD<String, Tuple2<String, String>> usersRDD =
    users.mapToPair(new PairFunction<
        String, //T
        String, //K
        Tuple2<String, String> //V
    >(){
        public Tuple2<String, Tuple2<String, String>> call(String s) {
            String[] userRecord = s.split("\t");
            Tuple2<String, String> location =
                new Tuple2<String, String>("L", userRecord[1]);
            return new Tuple2<String, Tuple2<String,
String>>(userRecord[0], location);
        }
    });
//Step 5 create a JavaRDD for transactions
JavaRDD<String> transactions = ctx.textFile(transactionsInputFile, 1);

//mapToPair
//<K2, V2> JavaPairRDD<K2, V2> mapToPair(PairFunction<T, K2, V2> f)
//Return a new RDD by applying a function to all elements of the RDD.
//PairFunction<T, K, V>
//T => Tuple2<K, V>
JavaPairRDD<String, Tuple2<String, String>> transactionsRDD =
    transactions.mapToPair(new PairFunction<

```

```

        String, //T
        String, //K
        Tuple2<String, String> //V
    >(){
    public Tuple2<String, Tuple2<String, String>> call(String s) {
        String[] transactionRecord = s.split("\t");
        Tuple2<String, String> product =
            new Tuple2<String, String>("P",
transactionRecord[1]);
        return new Tuple2<String, Tuple2<String,
String>>(transactionRecord[2], product);
    }
});

//Step 6 create a union of the RDDs created in step 4 and step 5
JavaPairRDD<String, Tuple2<String, String>> allRDD =
    transactionsRDD.union(usersRDD);

//Here we perform a union() on usersRDD and transactionsRdd
//JavaPairRDD<String, Tuple2<String, String>> allRDD =
//    usersRDD.union(transactionsRDD);

//Step 7 create a JavaPairRDD (userId, List<T2>) by calling groupBY()
//group allRDD by userID
JavaPairRDD<String, Iterable<Tuple2<String, String>>> groupedRDD =
    allRDD.groupByKey();
//now the groupedRDD entries will be as follows:
//<userId, List[T2("L", location),
//              //T2("P", Pi1),
//              //T2("P", Pi2),
//              //T2("P", Pi3),...
//              //]
// >

//Step 8 create a productLocationRDD as JavaPairRDD<String, String>
//PairFlatMapFunction<T, K, V>
//T => Iterable<Tuple2<K, V>>
JavaPairRDD<String, String> productLocationsRDD =
    groupedRDD.flatMapToPair(new PairFlatMapFunction<
        Tuple2<String, Iterable<Tuple2<String, String>>>, //T
        String, //K
        String>(){
        //V
    public Iterator<Tuple2<String, String>>
    call(Tuple2<String, Iterable<Tuple2<String, String>>> s){

```

```

        //String userID = s._1; //NOT needed
        Iterable<Tuple2<String, String>> pairs = s._2;
        String location = "UNKNOWN";
        List<String> products = new ArrayList<String>();
        for(Tuple2<String, String> t2: pairs) {
            if(t2._1.equals("L")) {
                location = t2._2;
            }
            else {
                //t2._1.equals("P")
                products.add(t2._2);
            }
        }

        //now emit (K, V) pairs
        List<Tuple2<String, String>> kvList =
            new ArrayList<Tuple2<String, String>>();
        for (String product : products) {
            kvList.add(new Tuple2<String, String>(product, location));
        }
        //Note that edges must be reciprocal; that is,
        //every {source, destination} edge must have
        // a corresponding {destination, source}
        return (Iterator<Tuple2<String, String>>) kvList;
    }

});

//Step 9 find all locations for a product;
//result will be JavaPAirRDD <String, List<String>>
JavaPairRDD<String, Iterable<String>> productByLocations =
    productLocationsRDD.groupByKey();

//debug3
List<Tuple2<String, Iterable<String>>> debug3 = productByLocations.collect();
System.out.println("--- debug3 begin ---");
for(Tuple2<String, Iterable<String>> t2 : debug3){
    System.out.println("debug3 t2._1=" + t2._1);
    System.out.println("debug3 t2._2+" + t2._2);
}
System.out.println("---debug3 end ---");

//Step 10 finalize output by changing "value" from List<String>
//to Tuple2<Set<String>, Integer>, where you have a unique

```

```

        //set of locations and their count
        JavaPairRDD<String, Tuple2<Set<String>, Integer>>
productByUniqueLocations =
            productByLocations.mapValues(
                new Function<Iterable<String>, //input
                Tuple2<Set<String>, Integer> //output
                >(){
                    public Tuple2<Set<String>, Integer> call(Iterable<String> s){
                        Set<String> uniqueLocations = new HashSet<String>();
                        for(String locations : s) {
                            uniqueLocations.add(locations);
                        }
                        return new Tuple2<Set<String>, Integer>(uniqueLocations,
                            uniqueLocations.size());
                    }
                });
        //Step 11 print the final result RDD
        //debug4
        System.out.println("== Unique Locations and Counts ===");
        List<Tuple2<String, Tuple2<Set<String>, Integer>>> debug4 =
            productByUniqueLocations.collect();
        System.out.println("---debug4 begins---");
        for(Tuple2<String, Tuple2<Set<String>, Integer>> t2 : debug4) {
            System.out.println("debug4 t2._1="+t2._1);
            System.out.println("debug4 t2._2="+t2._2);
        }
        System.out.println("---debug4 end---");
    }
}

```

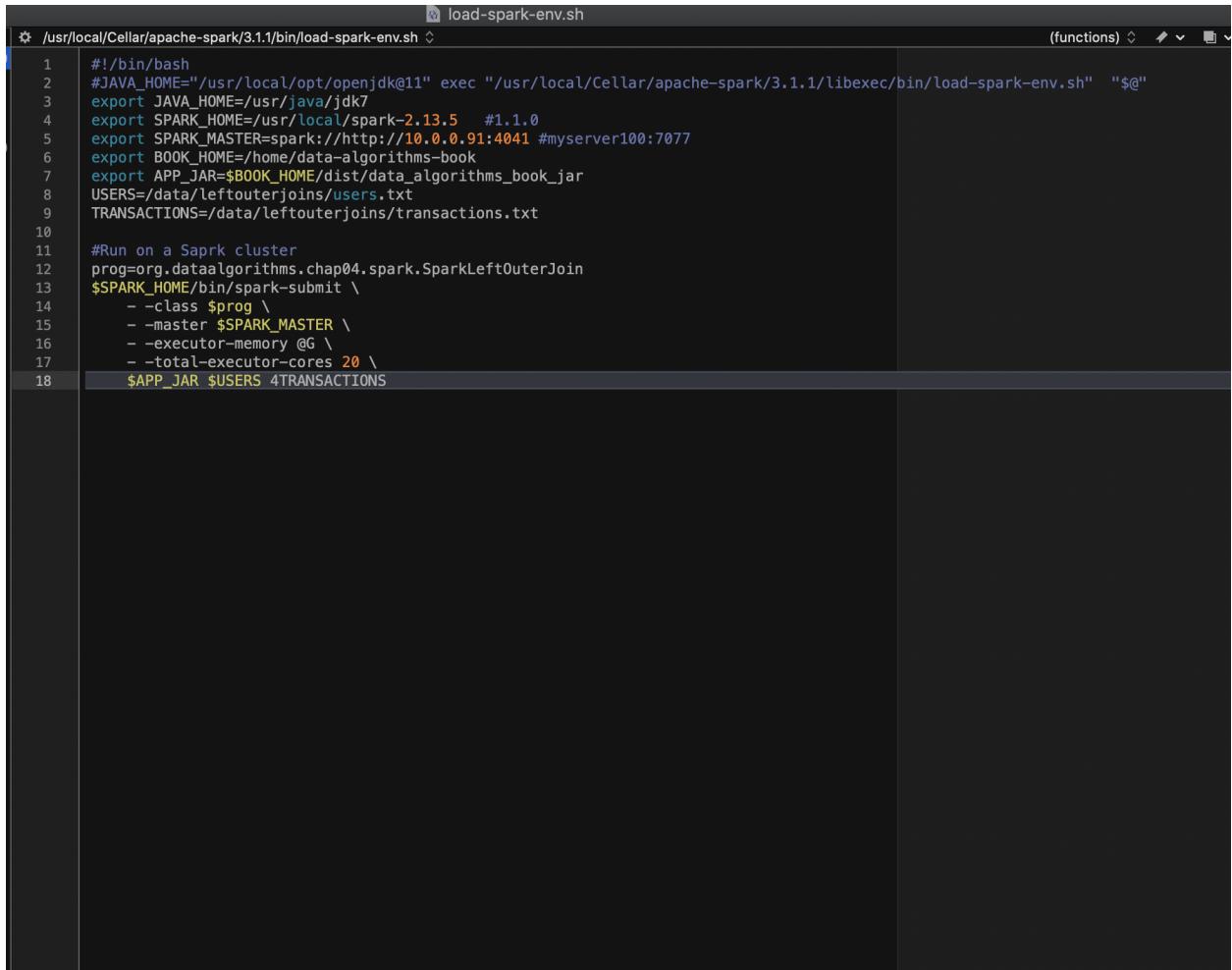
pom.xml

Adding Apache dependencies (creatingfile) (next page):

The screenshot shows the Eclipse IDE interface with the title bar "Desktop - LeftOuterJoin/pom.xml - Eclipse IDE". The left side features the "Package Explorer" view, which displays the project structure for "LeftOuterJoin". The "src/main/java" folder contains a package named "myd.LeftOuterJoin" with a single file "myLeftOuterJoin.java". The "src/test/java" folder is empty. The "JRE System Library [JavaSE-1.7]" and "Maven Dependencies" entries are also visible. The right side of the interface is the "LeftOuterJoin/pom.xml" editor, showing the XML code for the Maven project's build configuration.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
4   modelVersion="4.0.0">
5   <groupId>myId</groupId>
6   <artifactId>LeftOuterJoin</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8   <name>LeftOuterJoin</name>
9   <url>http://www.example.com</url>
10  <properties>
11    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
12    <maven.compiler.source>1.7</maven.compiler.source>
13    <maven.compiler.target>1.7</maven.compiler.target>
14  </properties>
15  <dependencies>
16    <dependency>
17      <groupId>org.apache.spark</groupId>
18      <artifactId>spark-sql_2.12</artifactId>
19      <version>3.1.1</version>
20    </dependency>
21    <dependency>
22      <groupId>junit</groupId>
23      <artifactId>junit</artifactId>
24      <version>4.11</version>
25      <scope>test</scope>
26    </dependency>
27  </dependencies>
28  <build>
29    <pluginManagement><!-- lock down plugin versions to avoid using Maven defaults (may be moved to
30      <plugins>
31        <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean
32        <plugin>
33          <artifactId>maven-clean-plugin</artifactId>
34          <version>3.1.0</version>
35        </plugin>
36        <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/
37      </plugins>
38    </pluginManagement>
39  </build>
```

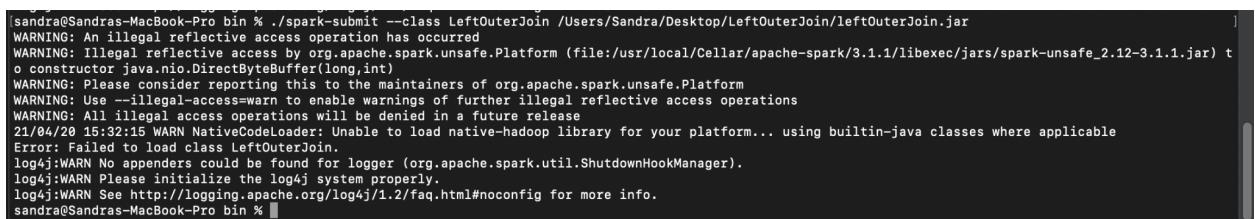
Updating #!/bin/bash



```
load-spark-env.sh
/usr/local/Cellar/apache-spark/3.1.1/bin/load-spark-env.sh
functions

1 #!/bin/bash
2 #JAVA_HOME="/usr/local/opt/openjdk@11" exec "/usr/local/Cellar/apache-spark/3.1.1/libexec/bin/load-spark-env.sh" "$@"
3 export JAVA_HOME=/usr/java/jdk7
4 export SPARK_HOME=/usr/local/spark-2.13.5 #1.1.0
5 export SPARK_MASTER=spark://http://10.0.0.91:4041 #myserver100:7077
6 export BOOK_HOME=/home/data-algorithms-book
7 export APP_JAR=$BOOK_HOME/dist/data_algorithms_book.jar
8 USERS=/data/leftouterjoins/users.txt
9 TRANSACTIONS=/data/leftouterjoins/transactions.txt
10
11 #Run on a Spark cluster
12 prog=org.dataalgorithms.chap04.spark.SparkLeftOuterJoin
13 $SPARK_HOME/bin/spark-submit \
14   --class $prog \
15   --master $SPARK_MASTER \
16   --executor-memory @G \
17   --total-executor-cores 20 \
18   $APP_JAR $USERS 4TRANSACTIONS
```

Creating .jar file Inserting java code into scala → error!



```
sandra@Sandras-MacBook-Pro bin % ./spark-submit --class LeftOuterJoin /Users/Sandra/Desktop/LeftOuterJoin/leftOuterJoin.jar
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/usr/local/Cellar/apache-spark/3.1.1/libexec/jars/spark-unsafe_2.12-3.1.1.jar) to
o constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
21/04/20 15:32:15 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Error: Failed to load class LeftOuterJoin.
log4j:WARN No appenders could be found for logger (org.apache.spark.utilShutdownHookManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
sandra@Sandras-MacBook-Pro bin %
```

Other approach in Spark-shell (running commands but getting nowhere):

8 Read text into Spark

```
from local filesystem:  
text_RDD =  
sc.textFile("file:///home/cloudera/testfile1")  
from HDFS:  
text_RDD =  
sc.textFile("/user/cloudera/input/testfile1")  
text_RDD.take(1) #outputs the first line
```

9 Wordcount in Spark: map

```
def split_words(line):  
    return line.split()  
  
def create_pair(word):  
    return (word, 1)  
  
pairs_RDD=text_RDD.flatMap(split_words).map(create_pair)
```

10

```
pairs_RDD.collect()
Out[1]: [(u'A', 1),
          (u'long', 1),
          (u'one', 1),
          (u'ago', 1),
          (u'is', 1),
          (u'galaxy', 1),
          (u'far', 1),
          (u'far', 1),
          (u'away', 1)]
```

11 Wordcount in Spark: reduce

```
def sum_counts(a,b):  
    return a+b  
  
wordcounts_RDD=pairs_RDD.reduceByKey(sum_counts)  
  
wordcounts_RDD.collect()
```

12

```
Out[1]:
```

sandra — java - spark-shell — 108x60

```
Type in expressions to have them evaluated.  
Type :help for more information.
```

```
[scala] sc.textFile("users.txt")
res0: org.apache.spark.rdd.RDD[String] = users.txt MapPartitionsRDD[1] at textFile at <console>:25
```

```
[scala] res0.take(5)
res1: Array[String] = Array(u1 UT, u2 GA, u3 CA, u4 CA, u5 GA)
```

```
[scala] sc.textFile("transactions.txt")
res2: org.apache.spark.rdd.RDD[String] = transactions.txt MapPartitionsRDD[3] at textFile at <console>:25
```

```
[scala] res2.take(8)
res3: Array[String] = Array(t1 p3 u1 3 330, t2 p1 u2 1 400, t3 p1 u1 3 600, t4 p2 u2 10 1000, t5 p4 u4 9 90,
t6 p1 u1 4 120, t7 p4 u1 8 160, t8 p4 u5 2 40)
```

```
[scala] def create_pair(word):
<console>:1: error: ';' expected but ')' found.
    def create_pair(word):
        ^
[scala] def create_pair(word): return(word, 1)
<console>:1: error: ';' expected but ')' found.
    def create_pair(word): return(word, 1)
        ^
[scala] def split_words(line): return line.split()
<console>:1: error: ';' expected but ')' found.
    def split_words(line): return line.split()
        ^
[scala] pairs_RDD=text_RDD.flatMap(split_words).map(create_pair)
<console>:23: error: not found: value pairs_RDD
    pairs_RDD=text_RDD.flatMap(split_words).map(create_pair)
        ^
<console>:24: error: not found: value pairs_RDD
    val $ires6 = pairs_RDD
        ^
[scala] val users = sc.textFile(path + "/users.txt").
[|   | map(rec => (rec.split(",")(0), rec.split(",")((1))))
```

```
<console>:24: error: not found: value path
    val users = sc.textFile(path + "/users.txt").
        ^
[scala] val users = sc.textFile(path + "users.txt").
[|   | map(rec => (rec.split(",")(0), rec.split(",")((1))))
```

```
<console>:24: error: not found: value path
    val users = sc.textFile(path + "users.txt").
        ^
[scala] val usersLeftOuterJoin = users.leftOuterJoin(transactions)
<console>:23: error: not found: value users
    val usersLeftOuterJoin = users.leftOuterJoin(transactions)
        ^
<console>:23: error: not found: value transactions
    val usersLeftOuterJoin = users.leftOuterJoin(transactions)
        ^
[scala]
```

```
[scala] val users = sc.textFile(path + users.txt).
[|   | map(rec => (rec.split(",")(0), rec.split(",")((1)))))
<console>:24: error: not found: value path
    val users = sc.textFile(path + users.txt).
        ^
<console>:24: error: value txt is not a member of org.apache.spark.rdd.RDD[(String, Array[String])]
    val users = sc.textFile(path + users.txt).
        ^
[scala] val users = sc.textFile(path + hdfs://Data/LeftOuterJoins/users.txt).
[|   | map(rec => (rec.split(",")(0), rec.split(",")((1)))))
<console>:2: error: ')' expected but '(' found.
    map(rec => (rec.split(",")(0), rec.split(",")((1))))
        ^
[scala] val users = sc.textFile(path + "/hdfs://Data/LeftOuterJoins/users.txt").
[|   | map(rec => (rec.split(",")(0), rec.split(",")((1)))))
<console>:24: error: not found: value path
    val users = sc.textFile(path + "/hdfs://Data/LeftOuterJoins/users.txt").
        ^
[scala]
```

Unable to proceed or figure how to write the path for val users and val transactions so that i can try performing leftOuterJoin command

```
val usersLeftOuterJoin = users.leftOuterJoin(transactions)
```

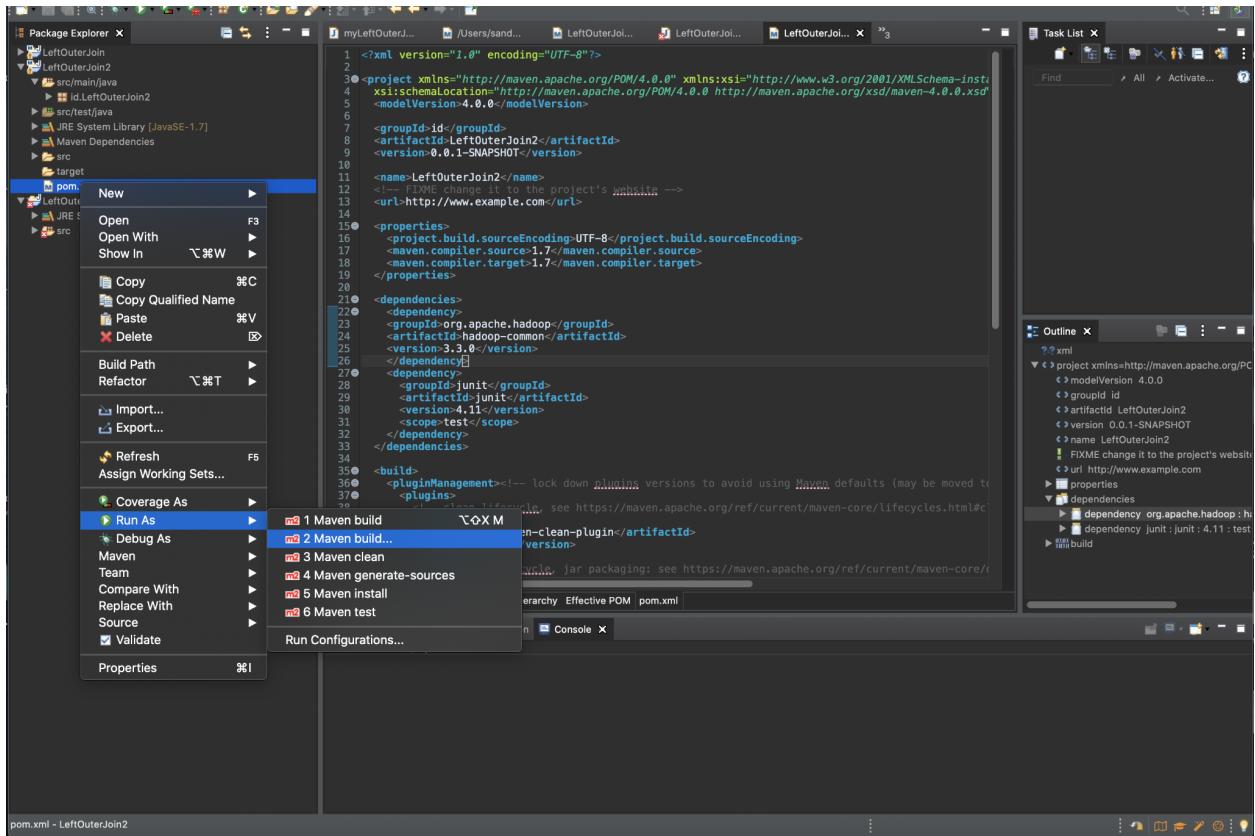
Attempt with Method hadoop for extra credit:

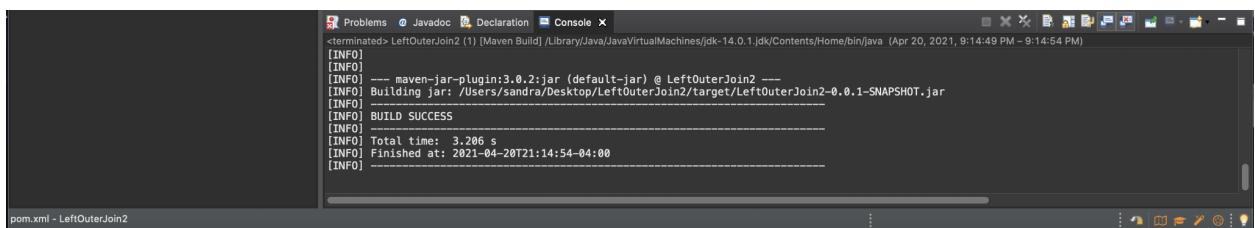
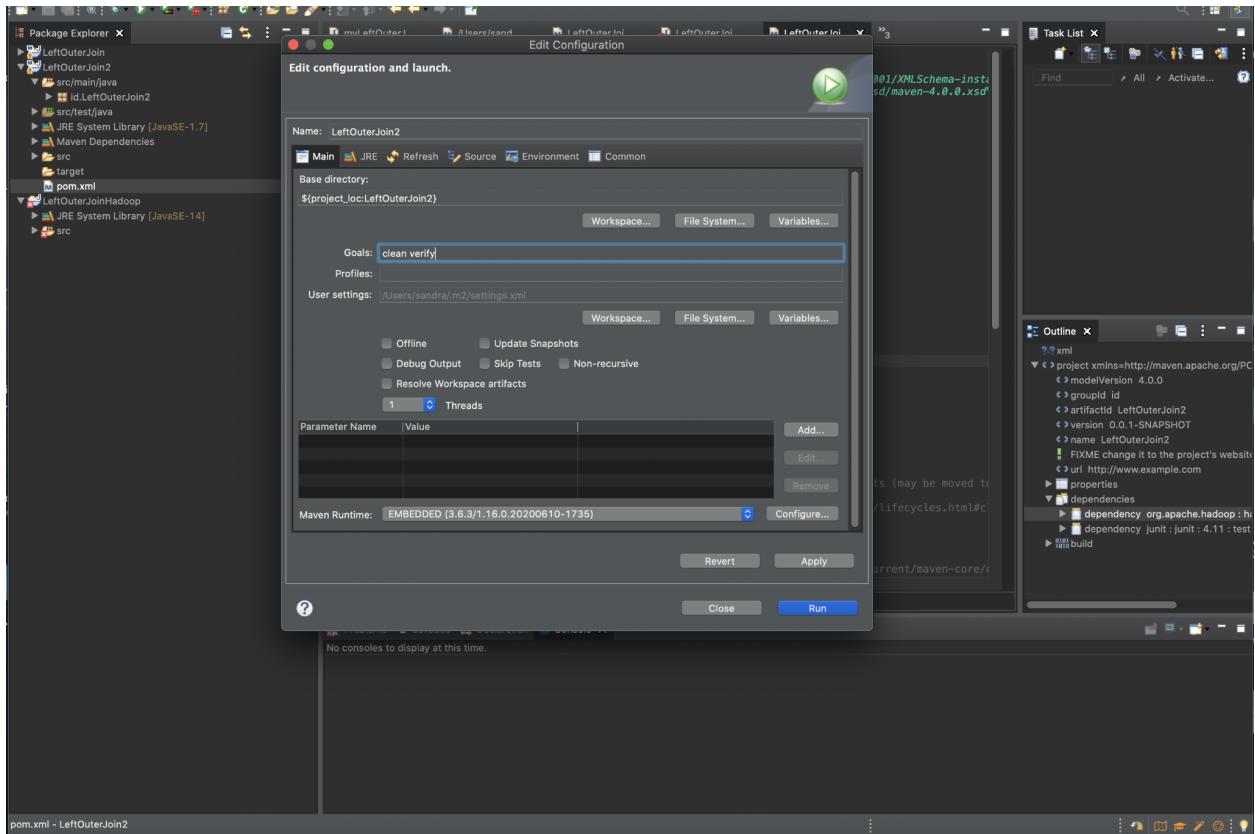
Java Code attached(LeftOuterJoinHadoop, LeftOuterJoinHadoopMapper, & LeftOuterJoinHadoopReducer)

Data in place:

```
sandra@Sandras-MacBook-Pro ~ % hdfs dfs -mkdir /Data
2021-04-19 20:39:24,879 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[sandra@Sandras-MacBook-Pro ~ % hdfs dfs -mkdir /Data/LeftOuterJoins
2021-04-19 20:39:39,503 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[sandra@Sandras-MacBook-Pro ~ % hadoop fs -put /usr/local/bin/Data/LeftOuterJoin/users.txt /Data/LeftOuterJoins
2021-04-19 20:40:00,661 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[sandra@Sandras-MacBook-Pro ~ % hadoop fs -cat /Data/LeftOuterJoins/users.txt
2021-04-19 20:40:22,795 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
u1 UT
u2 GA
u3 CA
u4 CA
[u5 GA
sandra@Sandras-MacBook-Pro ~ % hadoop fs -cat /Data/LeftOuterJoins/transactions.txt
2021-04-19 20:40:38,601 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
cat: '/Data/LeftOuterJoins/transactions.txt': No such file or directory
[sandra@Sandras-MacBook-Pro ~ % hadoop fs -put /usr/local/bin/Data/LeftOuterJoin/transactions.txt /Data/LeftOuterJoins
2021-04-19 20:41:09,297 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[sandra@Sandras-MacBook-Pro ~ % hadoop fs -cat /Data/LeftOuterJoins/transactions.txt
2021-04-19 20:41:13,549 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
t1 p3 u1 3 330
t2 p1 u2 1 400
t3 p1 u1 3 600
t4 p2 u2 10 1000
t5 p4 u4 9 90
t6 p1 u1 4 120
t7 p4 u1 8 160
[t8 p4 u5 2 40
sandra@Sandras-MacBook-Pro ~ % hadoop fs -put ~/Desktop/myLeftOuterJoin.java /Data/LeftOuterJoins
2021-04-19 20:41:32,018 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
sandra@Sandras-MacBook-Pro ~ %
```

Adding hadoop dependency in pom.xml file and running maven build for success:





Java code presented error the import org.apache.hadoop.mapreduce cannot be resolved
So switch pom.xml to the following dependency to get rid of that.

The screenshot shows the Eclipse IDE interface with the following details:

- Menu Bar:** Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Package Explorer:** Shows the project structure:
 - LeftOuterJoin
 - LeftOuterJoin2
 - src/main/java
 - id.LeftOuterJoin2
 - LeftOuterJoin2.java
 - LeftOuterJoin2Mapper.java
 - LeftOuterJoin2Reducer.java
 - src/test/java
 - JRE System Library [JavaSE-1.7]
 - Maven Dependencies
 - src
 - (default package)
 - LeftOuterJoinHadoop.java
 - LeftOuterJoinHadoopMapper.java
 - LeftOuterJoinHadoopReducer.java
 - target
 - pom.xml:** The current file being edited, showing Maven XML code.
 - Code Editor:** Displays the following Maven POM XML code:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
  modelVersion="4.0.0">
  <groupId>id</groupId>
  <artifactId>LeftOuterJoin2</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>LeftOuterJoin2</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-client</artifactId>
      <version>3.3.0</version>
    </dependency>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to -->
      <plugins>
        <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean_lifecycle -->
        <plugin>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.1.0</version>
        </plugin>
        <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/lifecycles.html#jar_lifecycle -->
      </plugins>
    </pluginManagement>
  </build>

```

Below the code editor, there are tabs: Overview, Dependencies, Dependency Hierarchy, Effective POM, and pom.xml.

Java code still present with errors i wasn't not able to fix

The screenshot shows the Eclipse IDE interface with the following details:

- Menu Bar:** Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Package Explorer:** Shows the project structure with packages like id.LeftOuterJoin, id.LeftOuterJoin2, and id.LeftOuterJoinHadoop.
- Code Editor:** Displays the Java code for `LeftOuterJoin2.java`. The code implements a `Mapper` and `Reducer` for a Hadoop job, reading from HDFS paths for transactions and users and writing to a user mapper.
- Task List:** Empty.
- Outline:** Shows the class definitions: `id.LeftOuterJoin2` and `LeftOuterJoin2`.
- Console:** Shows the Maven build output:

```
<terminated> LeftOuterJoin2 (1) [Maven Build] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java [Apr 20, 2021, 9:14:49 PM – 9:14:54 PM]
[INFO] 
[INFO] --- maven-jar-plugin:3.0.2:jar (Default-jar) @ LeftOuterJoin2 ---
[INFO] Building jar: /Users/sandra/Desktop/LeftOuterJoin2/target/LeftOuterJoin2-0.0.1-SNAPSHOT.jar
[INFO] 
[INFO] BUILD SUCCESS
[INFO] 
[INFO] Total time: 3.206 s
[INFO] Finished at: 2021-04-20T21:14:54-04:00
[INFO] 
```

Mapper:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "LeftOuterJoin2". It includes packages like "id.LeftOuterJoin" and "id.LeftOuterJoin2", and files such as "LeftOuterJoin2.java", "LeftOuterJoin2Mapper.java", and "LeftOuterJoin2Reducer.java".
- Code Editor:** Displays the "LeftOuterJoin2Mapper.java" file. The code defines a Mapper class that implements the `Mapper<LongWritable, Text, Text, IntWritable>` interface. It overrides the `map` method to split the input value by a tab character and emit key-value pairs based on user ID and location ID.
- Console:** Shows the Maven build output for the "LeftOuterJoin2" project. The log indicates a successful build with a total time of 3.206 seconds, completed at 2021-04-20T21:14:54-04:00.

Reducer:

The screenshot shows the Eclipse IDE interface with the following details:

- Menu Bar:** Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Package Explorer:** Shows the project structure:
 - id.LeftOuterJoin
 - LeftOuterJoin2 (selected)
 - src/main/java
 - id.LeftOuterJoin2
 - LeftOuterJoin2.java
 - LeftOuterJoin2Mapper.java
 - LeftOuterJoin2Reducer.java
 - src/test/java
 - JRE System Library [JavaSE-1.7]
 - Maven Dependencies
 - src
 - target
 - JRE System Library [JavaSE-14]
 - src
 - (default package)
 - LeftOuterJoinHadoop.java
 - LeftOuterJoinHadoopMapper.java
 - LeftOuterJoinHadoopReducer.java
- Editor:** Displays the code for `LeftOuterJoin2Reducer.java`. The code implements a reducer for a left outer join, handling both undefined and defined right values.
- Task List:** Shows a single task: "Find All" and "Activate...".
- Outline:** Shows the outline of the selected class: `id.LeftOuterJoin2`, `LeftOuterJoin2Reducer`, and the method `reduce(Text, Iterable<IntWritable>)`.
- Console:** Shows the Maven build output:

```
<terminated> LeftOuterJoin2 (1) [Maven Build] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java [Apr 20, 2021, 9:14:49 PM – 9:14:54 PM]
[INFO] [INFO]
[INFO] --- maven-jar-plugin:3.0.2:jar (Default-Jar) @ LeftOuterJoin2 ---
[INFO] Building jar: /Users/sandra/Desktop/LeftOuterJoin2/target/LeftOuterJoin2-0.0.1-SNAPSHOT.jar
[INFO] [INFO] BUILD SUCCESS
[INFO] [INFO] Total time: 3.206 s
[INFO] Finished at: 2021-04-20T21:14:54-04:00
[INFO] [INFO]
```

Unable to proceed!