Winter 2014

CS246: Mining Massive Datasets

Problem Set 0

Due 9:30am January 14, 2014

General Instructions

This homework is to be completed individually (no collaboration is allowed). Also, you are not allowed to use any late days for the homework. This homework is worth 1% of the total course grade.

The purpose of this homework is to get you started with Hadoop. Here you will learn how to write, compile, debug and execute a simple Hadoop program. First part of the homework serves as a tutorial and the second part asks you to write your own Hadoop program.

Section 1 describes the virtual machine environment. Instead of the virtual machine, you are welcome to setup your own pseudo-distributed or fully distributed cluster if you prefer. Any version of Hadoop that is at least 1.0 will suffice. (For an easy way to set up a cluster, try Cloudera Manager: http://archive.cloudera.com/cm4/installer/latest/ cloudera-manager-installer.bin.) If you choose to setup your own cluster, you are responsible for making sure the cluster is working properly. The TAs will be unable to help you debug configuration issues in your own cluster.

Section 2 explains how to use the Eclipse environment in the virtual machine, including how to create a project, how to run jobs, and how to debug jobs. Section 2.5 gives an end-to-end example of creating a project, adding code, building, running, and debugging it.

Section 3 is the actual homework assignment. There are no deliverables for sections 1 and 2. In section 3, you are asked to write and submit your own MapReduce job.

This homework requires you to upload the code and hand-in a print-out of the output for Section 3.

Regular (non-SCPD) students should submit hard copies of the answers (Section 3) either in class or in the submission box (see course website for location). For paper submission, please fill the cover sheet and submit it as a front page with your answers. You should upload your source code and any other files you used.

SCPD students should submit their answers through SCPD and also upload the code. The submission must include the answers to Section 3, the cover sheet and the usual SCPD routing form (http://scpd.stanford.edu/generalInformation/pdf/SCPD_HomeworkRouteForm.pdf).

Cover Sheet: http://cs246.stanford.edu/cover.pdf

Upload Link: http://snap.stanford.edu/submit/

Questions

1 Setting up a virtual machine

- Download and install *VirtualBox* on your machine: http://virtualbox.org/wiki/ Downloads
- Download the *Cloudera Quickstart VM* at http://www.cloudera.com/content/dev-center/ en/home/developer-admin-resources/quickstart-vm.html
- Uncompress the VM archive. It is compressed with 7-Zip. If needed, you can download a tool to uncompress the archive at http://www.7-zip.org/.
- Start *VirtualBox* and click *Import Appliance*. Click the folder icon beside the location field. Browse to the uncompressed archive folder, select the .ovf file, and click the *Open* button. Click the *Continue* button. Click the *Import* button.
- Your virtual machine should now appear in the left column. Select it and click on *Start* to launch it. Username and password are "cloudera" and "cloud era".
- Optional: Open the network properties for the virtual machine. Click on the Adapter 2 tab. Enable the adapter and select Host-only Adapter. If you do this step, you will be able to connect to the running virtual machine from the host OS at 192.168.56.101.

Virtual machine includes the following software

- CentOS 6.2
- JDK 6 (1.6.0_32)
- Hadoop 2.0.0
- Eclipse 4.2.6 (Juno)

The login user is cloudera, and the password for that account is cloudera.

2 Running Hadoop jobs

Generally Hadoop can be run in three modes.

1. Standalone (or local) mode: There are no daemons used in this mode. Hadoop uses the local file system as an substitute for HDFS file system. The jobs will run as if there is 1 mapper and 1 reducer.

- 2. **Pseudo-distributed mode:** All the daemons run on a single machine and this setting mimics the behavior of a cluster. All the daemons run on your machine locally using the HDFS protocol. There can be multiple mappers and reducers.
- 3. Fully-distributed mode: This is how Hadoop runs on a real cluster.

In this homework we will show you how to run Hadoop jobs in Standalone mode (very useful for developing and debugging) and also in Pseudo-distributed mode (to mimic the behavior of a cluster environment).

2.1 Creating a Hadoop project in Eclipse

(There is a plugin for Eclipse that makes it simple to create a new Hadoop project and execute Hadoop jobs, but the plugin is only well maintained for Hadoop 1.0.4, which is a rather old version of Hadoop. There is a project at https://github.com/winghc/hadoop2x-eclipse-plugin that is working to update the plugin for Hadoop 2.0. You can try it out if you like, but your milage may vary.)

To create a project:

1. Open or create the ~/.m2/settings.xml file and make sure it has the following contents:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
  <profiles>
    <profile>
      <id>standard-extra-repos</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <repositories>
        <repository>
          <id>central</id>
          <url>http://repo.maven.apache.org/maven2/</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>true</enabled>
          </snapshots>
        </repository>
        <repository>
          <id>cloudera</id>
          <url>
```

```
https://repository.cloudera.com/artifactory/cloudera-repos
</url>
<releases>
<releases>
<renabled>true</enabled>
</releases>
<snapshots>
<renabled>true</enabled>
</repository>
</repository>
</repositories>
</profile>
</profiles>
</profiles>
```

- 2. Open Eclipse and select $File \rightarrow New \rightarrow Project...$
- 3. Expand the Maven node, select Maven Project, and click the Next > button.
- 4. On the next screen, click the Next > button.
- 5. On the next screen, when the archetypes have loaded, select maven-archetype-quickstart and click the Next > button.
- 6. On the next screen, enter a group name in the *Group Id* field, and enter a project name in the *Artifact Id*. Click the *Finish* button.
- 7. In the package explorer, expand the project node and double-click the pom.xml file to open it.
- 8. Replace the current "dependencies" section with the following content:

```
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>jdk.tools</groupId>
            <artifactId>jdk.tools</artifactId>
            <version>1.6</version>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-hdfs</artifactId>
            <version>2.0.0-cdh4.0.0</version>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-auth</artifactId>
            <version>2.0.0-cdh4.0.0</version>
```

```
</dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-common</artifactId>
            <version>2.0.0-cdh4.0.0</version>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-core</artifactId>
            <version>2.0.0 -mr1-cdh4.0.1</version>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit-dep</artifactId>
            <version>4.8.2</version>
        </dependency>
    </dependencies>
</dependencyManagement>
<dependencies>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-hdfs</artifactId>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-auth</artifactId>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-common</artifactId>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-core</artifactId>
    </dependency>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.10</version>
        <scope>test</scope>
    </dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
```

```
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>2.1</version>
<configuration>
<source>1.6</source>
<target>1.6</target>
</configuration>
</plugin>
</plugin>
</build>
```

- 9. Save the file.
- 10. Right-click on the project node and select $Maven \rightarrow Update Project$.

You can now create classes in the **src** directory. After writing your code, build the JAR file by right-clicking on the project node and selecting $Run \ As \rightarrow Maven \ install$.

2.2 Running Hadoop jobs in standalone mode

After creating a project, adding source code, and building the JAR file as outlined above, the JAR file will be located at /workspace/< project >/target directory.

Open a terminal and run the following command:

hadoop jar ~/workspace/< project >/target/< project >-0.0.1-SNAPSHOT.jar \

```
-D mapped.task.tracker=local -D fs.defaultFS=local < args >
```

You will see all of the output from the map and reduce tasks in the terminal.

2.3 Running Hadoop jobs in pseudo-distributed mode

Open a terminal and run the following command:

hadoop jar ~/workspace/< project >/target/< project >-0.0.1-SNAPSHOT.jar < args >

To see all running jobs, run the following command:

hadoop job -list

To kill a running job, find the job's ID and then run the following command:

hadoop job -kill < id >

2.4 Debugging Hadoop jobs

To debug an issue with a job, the easiest approach is to add print statements into the source file and run the job in standalone mode. The print statements will appear in the terminal output. When running your job in pseudo-distributed mode, the output from the job is logged in the task tracker's log files, which can be accessed most easily by pointing a web browser to port 50030 of the server. From the job tracker web page, you can drill down into the failing job, the failing task, the failed attempt, and finally the log files. Note that the logs for stdout and stderr are separated, which can be useful when trying to isolate specific debugging print statements.

If you enabled the second network adapter in the VM setup, you can point your local browser to http://192.168.56.101:50030/ to access the job tracker page. Note, though, that when you follow links that lead to the task tracker web page, the links point to localhost.locadomain, which means your browser will return a page not found error. Simply replace localhost.locadomain with 192.168.56.101 in the URL bar and press enter to load the correct page.

2.5 Example project

In this section you will create a new Eclipse Hadoop project, compile, and execute it. The program will count the frequency of all the words in a given large text file. In your virtual machine, Hadoop, Java environment and Eclipse have already been pre-installed.

• Edit the ~/.m2/settings.xml file as outlined above. See Figure 1



Figure 1: Create a Hadoop Project.

• Open Eclipse and create a new project as outlined above. See Figures 2-9.

										Java	- Eclipse
<u>F</u> ile	<u>E</u> dit	<u>S</u> ource	Refac <u>t</u> or	<u>N</u> avigate	Se <u>a</u> rch	<u>P</u> roject	<u>R</u> un	<u>W</u> indow	<u>H</u> elp		
N	ew					lt+N >	Jav	a Project			V 51 V X
0	pen Fil	e <u>.</u>					P <u>r</u> o	ject			0-1
C	ose					Ctrl+W	Pac	:kage			
C	ose Al	1			Shift+	Ctrl+W	Cla	SS			
Si	ave					Ctrl+S	Inte	erface			
Sa	ave As						Enu	um			
Sa	av <u>e</u> All				Shift-	+Ctrl+S	Anı	notation			
R	evert						Sou	urce Folde	r		

Figure 2: Create a Hadoop Project.

E New Project	×
Select a wizard	
Create a Maven Project	
<u>W</u> izards:	
type filter text	4
👂 🗁 General	
D 🔁 CVS	
👂 🗁 Java	
🗢 🗁 Maven	
🗊 Checkout Maven Projects from SCM	
🖄 Maven Module	
👺 Maven Project	
👂 🗁 Examples	
(?) < Back Next > Cancel	nish

Figure 3: Create a Hadoop Project.

ŧ	New Maven Project		×
New Maven project		f	
Select project name and location			M
Create a simple project (skip arch	etype selection)		
✓ Use default <u>W</u> orkspace location			
Location:		→ Br	ows <u>e</u>
<u>Add</u> project(s) to working set			
Wo <u>r</u> king set:		\$ M	lor <u>e</u>
► Ad <u>v</u> anced			
(?)	< Back Next >	Cancel	sh
<u> </u>			

Figure 4: Create a Hadoop Project.

€	New Maven Project	
New Maven project		
Select an Archetype		IAI
Catalog: All Catalogs		≎ <u>C</u> onfigure
<u>F</u> ilter:		
Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-marmalade-mojo	RELEASE
org.apache.maven.archetypes	maven-archetype-mojo	RELEASE
org.apache.maven.archetypes	maven-archetype-plugin	1.2
org.apache.maven.archetypes	maven-archetype-plugin-site	1.1
org.apache.maven.archetypes	maven-archetype-portlet	RELEASE
org.apache.maven.archetypes	maven-archetype-profiles	RELEASE
org.apache.maven.archetypes	maven-archetype-quickstart	1.1
An archetype which contains a sample Maven project	ct.	
Show the last version of Archetype only	Include snapshot archetypes	Add Archetype
Advanced		
-		
0	< Back Next >	Cancel Finish

Figure 5: Create a Hadoop Project.

e	New Ma	aven Project			×
New Mave	en project				
Specify Are	chetype parameters				IAI
Group Id:	edu.stanford.cs246				*
Artifact Id:	wordcount				~
Version:	0.0.1-SNAPSHOT				
Package:	edu.stanford.cs246.wordcount				~
Properties	available from archetype:				
Name	Value				<u>A</u> dd
					Bemove
 Advance 	d				
0		- Deels	Neutra	Canad	et - t - h
T		< <u>B</u> ack	Next >	Cancel	Einish

Figure 6: Create a Hadoop Project.



Figure 7: Create a Hadoop Project.

ŧ	E Java - wordcount/pom.xml - E						
<u>F</u> ile <u>E</u> dit <u>S</u> o	ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> i	rch <u>P</u> roject <u>R</u> un <u>W</u> in	dow <u>H</u> elp				
1 . 5 .	🛛 🕤 🔺 🔯 × 🗿	· • • • • •	🙆 🖨 🖌 🖌 언 - 전 - 🍫 -				
Package	Explorer 🛛 🖳 🚍	wordcount/pom <project xmlns<br="">xsi:schema <modelvers< pre=""></modelvers<></project>	.xml 8 ="http://maven.apache.org/POM/4.0. Location="http://maven.apache.org/H ion=4.0.0				
 ▷ @ src ▷ @ src ▷ ➡ JRE ▷ ➡ Ma 	Ne <u>w</u> Go Into Open in <u>N</u> ew Window Ope <u>n</u> Type Hierarchy Show In	F4	u.stanford.cs246 J>wordcount 0.1-SNAPSHOT >Jar count				
v ≽sit ≥tar ₪ po	<u>C</u> opy Copy Paste	Ctrl+C Ctrl+V	//maven.apacne.org >> t.build.sourceEncoding>UTF-8=>>				
	Delete Remove from Context Build Path Source Refactor	Delete Shift+Ctrl+Alt+Down Shift+Alt+S Shift+Alt+T	<pre>/Managementb jencies> >pendency> <groupid>jdk.tools</groupid> <artifactid>jdk.tools<version>1.6</version> fependency> pendency></artifactid></pre>				
_	Import Exp <u>o</u> rt Re <u>f</u> resh Clo <u>s</u> e Project	F5	<pre><groupid>org.apache.hadoop<artifactid>hadoop-hdfs<version>2.0.0-cdh4.0.0</version> jependency> egroupId>org.apache.hadoop<artifactid>hadoop-auth<version>2.0.0.04.0.0</version></artifactid></artifactid></groupid></pre>				
	Assign Working Sets Bun As Debug As Validate Tgam	>	<pre>setSatir_icon during version dependency> egroupId>org.apache.hadoop<artifactid>hadoop-common</artifactid>hadoop-commonhadoop-commonhadoop-commonhadoop</pre>				
	Comp <u>a</u> re With Restore from Local Histor <u>y</u>	>	Add Dependency Add Plugin				
	<u>Maven</u> P <u>r</u> operties	Alt+Enter	New <u>M</u> aven Module Project				
<		Overview Depende	Download JavaDoc Download Sources				
wordcount) ə Java - w	ordcount/pom 🗵 [cloudera@	واocalhost:~	Disable Workspace <u>R</u> esolution Disable Maven <u>N</u> ature				

Figure 8: Create a Hadoop Project.

🗧 Update Maven Pro	ject	×
Update Maven Project		
Select Maven projects and update options		
Available Maven Codebases		
🗹 😓 wordcount	Se	lect All
	Des	
	Exi	band All
	Coll	apse All
□ Offline		
☑ Update dependencies		
Force Update of Snapshots/Releases		
✓ Update project configuration from pom.xml		
✓ Clean projects		
?	Cancel O	К

Figure 9: Create a Hadoop Project.

• The project will contain a stub source file in the src/main/java directory that we will not use. Instead, create a new class called WordCount. From the *File* menu, select $New \rightarrow Class$. See Figure 10

•									Java - Ecli
<u>F</u> ile	<u>E</u> dit	<u>S</u> ource	Refac <u>t</u> or	<u>N</u> avigate	Se <u>a</u> rch	<u>P</u> roject	<u>R</u> ur	n <u>W</u> indow <u>H</u> elp	
<u>N</u> e							>	Java Project	
Ор	en Fil	e <u>.</u>						P <u>r</u> oject	0
	se					Ctrl+	W	Package	
Clo	ose All	I			Shi	ft+Ctrl+	w		
Sa	ve					Ctrl+	-S	Interface	
Sa	ve As.							Enum	
Sav	v <u>e</u> All				Sh	nift+Ctrl+	-S	Annotation	
Rei	vert							Source Folder	

Figure 10: Create java file.

• On the next screen, enter the package name (e.g, the group ID plus the project name) in the *Package* field. Enter WordCount as the *Name*. See Figure 11.

🖨 Java Class	New Java Class	×
Create a new Java c	lass.	C
Source fol <u>d</u> er:	wordcount/src/main/java	Br <u>o</u> wse
Pac <u>k</u> age:	edu.stanford.cs246.wordcount	Bro <u>w</u> se
Enclosing type:		Bro <u>w</u> se
Na <u>m</u> e:	WordCount	
Modifiers:	<u>public</u> O default O private O protected	
	abstract final static	
<u>S</u> uperclass:	java.lang.Object	Brows <u>e</u>
Interfaces:		<u>A</u> dd
		<u>R</u> emove
Which method stubs	s would you like to create?	
	public static void main(String[] args)	
	<u>Constructors from superclass</u>	
	Inherited abstract methods	
Do you want to add	comments? (Configure templates and default value here	<u>e)</u>
	<u>G</u> enerate comments	
?	Cancel	<u>F</u> inish

Figure 11: Create java file.

• In the *Superclass* field, enter Configured and click the *Browse* button. From the popup window select *Configured* — *org.apache.hadoop.conf* and click the *OK* button. See Figure 12.

New Java Class	×
E Superclass Selection	×
<u>C</u> hoose a type:	
Configured	ſ
Matching items:	
Configured - org.apache.hadoop.conf	
🗘 🗘 Workspace matches	
ConfiguredFailoverProxyProvider	
G * ConfiguredNNAddress	
org.apache.hadoop.conf - /home/clouderah4.0.0/hadoop-common-2.0.0-cdh4.0.0.jar	•
Cancel OK	
minentee add commente2 (Confirms tomolotics and default value bare)	
	_
Cancel <u>F</u> inish	

Figure 12: Create java file.

• In the *Interfaces* section, click the *Add* button. From the pop-up window select **Tool** — *org.apache.hadoop.util* and click the *OK* button. See Figure 13.

۲	F Implemented Interfaces Selection ×	>
Java Class Create a new	Choose interfaces:	C
Source fol <u>d</u> er:	Matching items:	Br <u>o</u> wse
Pac <u>k</u> age:	 ioor - org.apache.nadoop.uur - /nome/croudera/.mz/rep Workspace matches 	Bro <u>w</u> se
Enclosing t	 Tool - javax.tools - [jdk1.6.0_32] ToolkitThreadBlockedHandler 	Bro <u>w</u> se
Na <u>m</u> e: Modifiers:		
Superclass:		Brows <u>e</u>
Interraces:		<u>A</u> dd <u>R</u> emove
Which method	org anache badoon util - /h. n-common-2.0.0-cdh4.0.0 iar	
Do you want t	Add Cancel OK	
?	Cancel	<u>F</u> inish

Figure 13: Create java file.

• Check the boxes for *public static void main(String args[])* and *Inherited abstract methods* and click the *Finish* button. Se Figure 14

ŧ	New Java Class		×
Java Class	lass		C
Create a new Java o	.1d55.		
Source fol <u>d</u> er:	wordcount/src/main/java		Br <u>o</u> wse
Pac <u>k</u> age:	edu.stanford.cs246.wordcount		Bro <u>w</u> se
Enclosing type:			Bro <u>w</u> se
Na <u>m</u> e:	WordCount		
Modifiers:	<u>public</u> O default O private	O pro <u>t</u> ected	
	□ abs <u>t</u> ract □ fina <u>l</u> □ stati <u>c</u>		
<u>Superclass:</u>	org.apache.hadoop.conf.Configured		Brows <u>e</u>
Interfaces:	org.apache.hadoop.util.Tool		<u>A</u> dd
			Remove
which method stub	s would you like to create?		
	<u>Constructors from superclass</u>		
	☑ Inherited abstract methods		
Do you want to add	comments? (Configure templates and d	efault value <u>her</u>	<u>e)</u>
	<u>Generate comments</u>		
?		Cancel	<u>F</u> inish

Figure 14: Create WordCount.java.

• You will now have a rough skeleton of a Java file as in Figure 15. You can now add code to this class to implement your Hadoop job.



Figure 15: Create WordCount.java.

• Rather than implement a job from scratch, copy the contents from http://snap. stanford.edu/class/cs246-data-2014/WordCount.java and paste it into the WordCount.java file. Be careful to leave the package statement at the top intact. See Figure 16. The code in WordCount.java calculates the frequency of each word in a given dataset.

File Edit Source Refactor Navigate Search Project Run Window Help	F Java	- wordcount/src/main/java/edu/stanford/cs246/wordcount/WordCount.java
Package Explorer S Package Explorer S WordCount.java S Image: Size and the second s	<u>File Edit Source Refactor Navigate Sear</u>	ch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp
<pre> Package Explorer</pre>	📫 * 🖬 * 💷 🕼 📥 🔌 🏇 * 💽 *	 Q × B B C C A F F A F <l< th=""></l<>
<pre>job_SetUutputKeyLiass(lext.class); job_setOutputKeyLiass(lext.class); job_setOutputValueClass(IntWritable.class); job_setMapperClass(Map.class); job_setReducerClass(Reduce.class); job_setInputFormatClass(TextInputFormat.class); job_setOutputFormatClass(TextOutputFormat.class); job_setOutputFormatClass(TextOutputFormat.class); FileInputFormat.addInputPath(job, new Path(args[0])); FileOutputFormat.setOutputPath(job, new Path(args[1])); job_waitForCompletion(true); return 0; }</pre>	 Package Explorer IS Package Explorer IS Image: Sector Sect	<pre> WordCount.java 83 package edu.stanford.cs246.wordcount; import java.io.IOException; public class WordCount extends Configured implements Tool { public static void main(String[] args) throws Exception { ToolRunner.run(new Configuration(), new WordCount(), args); } @ @Override public int run(String[] args) throws Exception { Job job = new Job(getConf(), "WordCount"); job.setJarByClass(WordCount.class); job.setOutputKeyClass(Text.class); job.setOutputKeyClass(IntWritable.class); job.setReducerClass(Reduce.class); job.setInputFormatClass(TextInputFormat.class); job.setOutputFormatClass(TextOutputFormat.class); job.setOutputFormatClass(TextOutputFormat.class); job.setOutputFormat.setOutputPath(job, new Path(args[0])); FileOutputFormat.setOutputPath(job, new Path(args[1])); job.waitForCompletion(true); return 0; } } </pre>

Figure 16: Create WordCount.java.

• Build the project by right-clicking the project node and selecting $Run \ As \rightarrow Maven$ install. See Figure 17.

Eile Edit So	Java	- wordcount/src/main	/java/edu/stanford/cs246/wordcount/Word
<u>rite Edit 5</u> 0	arce herac <u>ion h</u> avigate se <u>a</u> n	en <u>r</u> ioject <u>R</u> an <u>w</u> inde	w Help
		· 🎴 · 🕆 🞯 · 2	● ▲ ペ·199 3 ○ ■ ■ 14 ○ ○
🛱 Package B	Explorer 🛿 🗖 🗖	🕖 WordCount.java	x
	 □ <u>\$</u> ≥ ▼	package edu.stan	nford.cs246.wordcount;
🗢 🖼 wordcor		a import invo in I	0Exception;
⊽ / src/r	Ne <u>w</u> Go Into	>	dCount extends Configured implements Tool {
🗢 🌐 e(c void main(String[] args) throws Exception {
▶ 🧾	Open In <u>N</u> ew Window	E4	er. / an (new configuration(), new wordcount(),
▶ 🧵	Show In	Shift+Alt+W 🕽	
▷ 🥭 src/t	Silo <u>w</u> III		un(String[] args) throws Exception { = new Job(getConf(), "WordCount"):
D May	<u>C</u> opy Copy Qualified Name	Cli+C	arByClass(WordCount.class);
	Paste	Ctrl+V	utputValueClass(IntWritable.class);
Image Site Site Site Site Site Site Site Sit	Delete	Delete	apperClass(Map.class);
pom	Remove from Context	Shift+Ctrl+Alt+Down	educerClass(Reduce.class);
	Build Path	>	nputFormatClass(TextInputFormat.class);
		Shift+Alt+S >	utputFormatclass(TextoutputFormat.class);
	Refac <u>t</u> or	Shift+Alt+T ゝ	<pre>tFormat.addInputPath(job, new Path(args[0])); utFormat.setOutputPath(job, new Path(args[1]))</pre>
	Import		<pre>ForCompletion(true);</pre>
	Export		
	Re <u>f</u> resh	F5	'
	Clo <u>s</u> e Project		c class Map extends
	<u>A</u> ssign Working Sets		er <longwritable, intwritable="" text,=""> { final static IntWritable ONE = new IntWritable</longwritable,>
	<u>R</u> un As	\rightarrow	1 Java Applet Shift+Alt+X A
	<u>D</u> ebug As	>	2 Java Application Shift+Alt+X J
	<u>V</u> alidate		<u>3</u> JUnit Test Shift+Alt+X T
	T <u>e</u> am	>	<u>4</u> Maven build Shift+Alt+X M
	Comp <u>a</u> re With	>	5 Maven build
	Restore from Local History		b Maven clean
		>	8 Mayen install
	P <u>r</u> operties	Alt+Enter	9 Maven test
			Pup Configurations
<		<	Run Configurations

Figure 17: Create WordCount.java.

- Download the *Complete Works of William Shakespeare* from Project Gutenberg at http://www.gutenberg.org/cache/epub/100/pg100.txt.
- Open a terminal and change to the directory where the dataset was stored.
- Run the command:

```
hadoop jar ~/workspace/wordcount/target/wordcount-0.0.1-SNAPSHOT.jar \
edu.stanford.cs246.wordcount.WordCount -D mapred.job.tracker=local \
-D fs.defaultFS=local dataset output
```

See Figure 18

[cloudera@localhost Desktop]\$ wget http://www.gutenberg.org/cache/epub/100/pg100.txt --2014-01-01 16:16:00-- http://www.gutenberg.org/cache/epub/100/pg100.txt Resolving www.gutenberg.org]152.19.134.47 Connecting to www.gutenberg.org]152.19.134.47[:80... connected. HTTP request sent, awaiting response... 200 OK Length: 5589890 (5.3M) [text/plain] Saving to: "pg100.txt" 100%[=========>] 5,589,890 1.24M/s in 5.8s 2014-01-01 16:16:06 (942 KB/s) - "pg100.txt" saved [5589890/5589890] [cloudera@localhost Desktop]\$ hadoop jar ~/workspace/wordcount/target/wordcount-0.0.1-SNAPSHOT.jar edu.stanford.cs246.wordcount.W ordCount -D mapred.job.tracker=local -D fs.defaultFS=local pg100.txt output 14/01/01 16:16:35 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:35 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:35 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:35 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:35 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:35 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:35 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:35 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:36 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:36 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:36 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:36 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:16:36 WARN fs.FileSystem: "local" is a deprecated filesystem name. Use "file:///" instead. 14/01/01 16:1

Figure 18: Run WordCount job.

• If the job succeeds, you will see an output directory in the current directory that contains a file called part-00000. The part-00000 file contains the output from the job. See Figure 19

```
[cloudera@localhost Desktop]$ head output/part-r-00000
        119383
        241
"'Tis
        1
"Α
        4
"AS-IS".
                 1
"Air,"
       1
"Alas,
        1
"Amen"
        2
"Amen"? 1
"Amen," 1
```

Figure 19: Run WordCount job.

• Run the command:

hadoop fs -ls

The command will list the contents of your home directory in HDFS, which should be empty, resulting in no output.

• Run the command:

hadoop fs -copyFromLocal pg100.txt to copy the dataset folder into HDFS.

• Run the command:

hadoop fs -ls

again. You should see the dataset directory listed, as in Figure 20 indicating that the dataset is in HDFS.

```
[cloudera@localhost Desktop]$ hadoop fs -ls
[cloudera@localhost Desktop]$ hadoop fs -copyFromLocal pg100.txt
[cloudera@localhost Desktop]$ hadoop fs -ls
Found 1 items
-rw-r--- 3 cloudera cloudera 5589890 2014-01-01 16:19 pg100.txt
[cloudera@localhost Desktop]$ ■
```

Figure 20: Run WordCount job.

• Run the command:

hadoop jar ~/workspace/WordCount/target/WordCount-0.0.1-SNAPSHOT.jar \ edu.stanford.cs246.wordcount.WordCount pg100.txt output

See Figure 21. If the job fails, you will see a message indicating that the job failed. Otherwise, you can assume the job succeeded.

[cloudera@localhost Desktop]\$ hadoop jar ~/workspace/wordcount/target/wordcount-0.0.1-SNAPSHOT.jar edu.stanford.cs246.wordcount.W ordCount pg100.txt output 14/01/01 16:22:08 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool f or the same. 14/01/01 16:22:09 INFO input.FileInputFormat: Total input paths to process : 1 14/01/01 16:22:09 INFO mapred.JobClient: Running job: job_201401011523_0001 14/01/01 16:22:11 INFO mapred.JobClient: map 0% reduce 0% 14/01/01 16:22:24 INFO mapred.JobClient: map 100% reduce 0% 14/01/01 16:22:32 INFO mapred.JobClient: map 100% reduce 100% 14/01/01 16:22:34 INFO mapred.JobClient: Job complete: job 201401011523 0001 14/01/01 16:22:34 INFO mapred.JobClient: Counters: 32 14/01/01 16:22:34 INFO mapred.JobClient: File System File System Counters 14/01/01 16:22:34 INFO mapred.JobClient: FILE: Number of bytes read=2491864 14/01/01 16:22:34 INFO mapred.JobClient: FILE: Number of bytes written=3797929 14/01/01 16:22:34 INFO mapred.JobClient: 14/01/01 16:22:34 INFO mapred.JobClient: FILE: Number of read operations=0 FILE: Number of large read operations=0 14/01/01 16:22:34 INFO mapred.JobClient: FILE: Number of write operations=0

Figure 21: Run WordCount job.

• Run the command:

hadoop fs -ls output

You should see an output file for each reducer. Since there was only one reducer for this job, you should only see one part-* file. Note that sometimes the files will be called part-NNNNN, and sometimes they'll be called part-r-NNNNN. See Figure 22

[cloudera@lo	ca	lhost Desl	<pre><top]\$ hadoop<="" pre=""></top]\$></pre>	fs -ls	output		
Found 3 item	S						
- rw-rr	3	cloudera	cloudera	0	2014-01-01	16:22	output/ SUCCESS
drwxr-xr-x	-	cloudera	cloudera	0	2014-01-01	16:22	output/logs
- rw- r r	3	cloudera	cloudera	720989	2014-01-01	16:22	output/part-r-00000

Figure 22: Run WordCount job.

• Run the command:

hadoop fs -cat output/part * | head

You should see the same output as when you ran the job locally, as shown in Figure 23

```
[cloudera@localhost Desktop]$ hadoop fs -cat output/part\* | head
       119383
н
       241
"'Tis
       1
"A
       4
"AS-IS".
               1
"Air," 1
"Alas,
       1
"Amen" 2
"Amen"? 1
"Amen," 1
cat: Unable to write to output stream.
```

Figure 23: Run	WordCount	job.
----------------	-----------	------

• To view the job's logs, open the browser in the VM and point it to http://localhost: 50030 as in Figure 24

🥹 0.0.0.0 Hadoop Map/Reduce Administration - Mozilla Firefox _ 💷 🗙													
<u>F</u> ile <u>E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp													
🕴 0.0.0.0 ł	🕴 0.0.0.0 Hadoop Map/Reduce Ad 🕀												
🔶 💽 localhost:50030/jobtracker.jsp													
[™] Most Visited → □ Cloudera □ Cloudera Manager □ Hue □ HDFS NameNode ℓ Hadoop JobTracker □ HBase Master □ Solr													
Cluster Summary (Heap Size is 81.06 MB/1021.94 MB)										nks 🛆			
Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Exclude Node	ed s
0	0	1	1	0	0	0	0	2	2	4.00	0	0	
Scheduling Information Queue Name State Scheduling Information default running N/A Filter (Jobid, Priority, User, Name)									=				
	Jobid	Priority	User	Name	Map % Complet	6 Map te Total	Maps Completed	Reduce Comple	% Redu te Tota	ce Reduces I Complete	job Scheduli Informat	ng Dia Info	gnc >
job_20140	1011523_0	001 NORMAL	cloudera	WordCount	t 100.00%	1	1	100.00%	1	1	NA	NA	
													~

Figure 24: View WordCount job logs.

• Click on the link for the completed job. See Figure 25.

2			Hadoop	job_2014	01011523	0001	on 0.0.0	.0 - Mozi	illa F	irefox		>	ĸ
<u>File E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp													
🧨 Hadoo	op job_2014010	011523_00	43-										
	localhost:5003	30/jobdetails	jsp?jobid=	job_20140	1011523_0	001&ref	resh=0			ి 🛛 🕄	Google	🔬 🖆	ŀ
🛅 Most V	/isited∨ □C	loudera 🖂	Cloudera M	anager []Hue []H	DFS Na	meNode	(Hado	op Jol	bTracker 🗍	Base Master	Solr	
Hado	oop job	20140	10115	23 00	01 on	0.0.	0.0					(~
User: clou	ıdera –	•		-									
Job Name	: WordCount												
Submit H	ost: localhost.lo	caldomain:80. caldomain	20/user/clou	dera/.stagin	g/job_201401	011523_	0001/Job.x	mi					Ξ
Submit H	ost Address: 12	27.0.0.1											
Job-ACLs: lob Setup	: All users are a : Successful	llowed											
Status: Succeeded													
Started at: Wed Jan 01 16:22:09 PST 2014													
Finished	in: 24sec	0.22.551512	514										
Job Clean	up: Successful												
Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed, Task At	Killed tempts					
map	100.00%	1	0	0	1	0		0 / 0					
reduce	100.00%	1	0	0	1	0		0 / 0					
	Counter Map Reduce Total												
		FILE: Number of bytes read 1,508,400 983,464 2,491,864											
		FILE: Number of bytes written 2,653,323 1,144,606 3,797,929											
		FILE: Numb	er of read o	perations					0	0	0		
		FILE: Numb	er of large r	ead operati	ons				0	0	0		
File Curt	C	FILE: Numb	er of write o	perations					0	0	0		
File Syste	em counters	HDFS: Num	ber of bytes	read				5,590,	012	0	5,590,012		
		I .											~

Figure 25: View WordCount job logs.

• Click the link for the map tasks. See Figure 26.

۷	Hadoop map task l	ist for job_2	2014010	11523_0001 on 0.0	.0.0 - Mozilla Firefox			_ 0	×
<u>F</u> ile <u>E</u> dit	t <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools	<u>H</u> elp							
🦿 Hadoo	p map task list for job_2 문								
	localhost:50030/jobtasks.jsp?jobid=job	20140101152	23_00018	type=map&pagenum	n=1 🖙 😭 🛃 Go	ogle		#	
📷 Most V	/isited∽ □Cloudera □Cloudera Mar	nager 🗌 Hue	HDF	S NameNode 🦿 Had	loop JobTracker 🔅 HBase M	laster	Solr		
Hado	oop map task list for	job_201	1401	011523_000	1 on 0.0.0.0				
All Tas	iks			1	1			1	
	Task	Complete	Status	Start Time	Finish Time	Errors	Counters		
	task_201401011523_0001_m_000000	100.00%		1-Jan-2014 16:22:16	1-Jan-2014 16:22:23 (6sec)		21	I	
Go back to	JobTracker								
Hadoop, 2	014.								

Figure 26: View WordCount job logs.

• Click the link for the first attempt. See Figure 27.

2	Hadoop Task	Details - Mo	zilla Firefo	¢				_	×
<u>File Edit View History B</u> ookmarks <u>T</u> oo	ols <u>H</u> elp								
【 Hadoop Task Details 日本 日本									
Iocalhost:50030/taskdetails.jsp?tipi	id=task_20140101152	23_0001_m_0	00000	<u>ن</u>	<u>'</u> 🕄 🛃 🖌	Google		ś	
ⓑ Most Visited ∽ □ Cloudera □ Cloudera	Manager 🗌 Hue 🗌	HDFS Name	Node 🦿 Had	oop JobTrack	er 🗌 HBas	e Maste	r 🗍 S	olr	
Job job_201401011523_	0001								
All Task Attempts									
Task Attempts	Machine	Status	Progress	Start Time	Finish Time	Errors	Task Logs	Counters	Action
attempt_201401011523_0001_m_000000_0	/default /localhost.localdomain	SUCCEEDED	100.00%	1-Jan-2014 16:22:16	1-Jan-2014 16:22:22 (6sec)		Last 4KB Last 8KB All	21	
Input Split Locations /default/localhost.localdomain Go back to the job Go back to JobTracker									
Hadoop, 2014.									

Figure 27: View WordCount job logs.

• Click the link for the full logs. See Figure 28.

8 Task Logs: 'attempt_201401011523_0001_m_000000_0' - Mozilla Firefox	_ = ×
<u>F</u> ile <u>E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp	
🗍 Task Logs: 'attempt_20140101 🕀	
🜪 🛞 localhost.localdomain:50060/tasklog?attemptid=attempt_201401011523_0001_m_000000_ 🔂 🛩 😰 🚺 😽 Google	e 🔮 🐴
Most Visited 🗸 🗍 Cloudera 🗋 Cloudera Manager 🗌 Hue 📄 HDFS NameNode 🦿 Hadoop JobTracker 🗌 HBase Mast	er 🗍 Solr
Task Logs: 'attempt_201401011523_0001_m_000000_0'	=
stderr logs	
<pre>syslog logs 2014-01-01 16:22:17,737 WARN mapreduce.Counters: Group org.apache.hadoop.mapred.Task\$Counter is deprecated. Use org. 2014-01-01 16:22:18,667 WARN org.apache.hadoop.conf.Configuration: session.id is deprecated. Instead, use dfs.metric 2014-01-01 16:22:18,667 WARN org.apache.hadoop.metrics.jvm.JvmMetrics: Initializing JVM Metrics with processName=WAR 2014-01-01 16:22:18,982 INFO org.apache.hadoop.util.ProcessTree: setsid exited with exit code 0 2014-01-01 16:22:18,966 INFO org.apache.hadoop.mapred.Task: Using ResourceCalculatorPlugin : org.apache.hadoop.mapred.Task: Using ResourceCalculatorPlugin : org.apache.hadoop.mapred.MapTask: Processing split: hdfs://localhost.localdomain:8020/u 2014-01-01 16:22:19,199 INFO org.apache.hadoop.mapred.MapTask: io.sort.mb = 50 2014-01-01 16:22:19,268 INFO org.apache.hadoop.mapred.MapTask: cecord buffer = 39845888/49807360 2014-01-01 16:22:19,268 INFO org.apache.hadoop.mapred.MapTask: spliting map output: record full = true 2014-01-01 16:22:19,141 INFO org.apache.hadoop.mapred.MapTask: spliting map output: record full = true 2014-01-01 16:22:20,141 INFO org.apache.hadoop.mapred.MapTask: buffert = 0; bufferd = 1172876; bufford = 49807360 2014-01-01 16:22:20,141 INFO org.apache.hadoop.mapred.MapTask: kvstart = 0; bvferd = 131072; length = 163840 2014-01-01 16:22:20,141 INFO org.apache.hadoop.mapred.MapTask: buffert = 0; bvferd = 131072; length = 163840 2014-01-01 16:22:20,141 INFO org.apache.hadoop.mapred.MapTask: buffert = 0; bvrend = 131072; length = 163840 2014-01-01 16:22:20,141 INFO org.apache.hadoop.mapred.MapTask: buffert = 0; bvrend = 131072; length = 163840</pre>	.apache.hadoop.mapreduce cs.session-id P, sessionId= L.LinuxResourceCalculatc user/cloudera/pg100.txt: d.MapTask\$MapOutputBuffe

Figure 28: View V	WordCount job	logs.
-------------------	---------------	-------

2.6 Using your local machine for development

If you enabled the second network adapter, you can use your own local machine for development, including your local IDE. If order to do that, you'll need to install a copy of Hadoop locally. The easiest way to do that is to simply download the archive from http:// archive.cloudera.com/cdh4/cdh/4/hadoop-2.0.0-cdh4.4.0.tar.gz and unpack it. In the unpacked archive, you'll find a etc/hadoop-mapreduce1 directory. In that directory, open the core-site.xml file and modify it as follows:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. --->
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://192.168.56.101:8020</value>
</property>
</configuration>
```

Next, open the mapred-site.xml file in the same directory and modify it as follows:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Put site-specific property overrides in this file. --->
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>192.168.56.101:8021</value>
</property>
</configuration>
```

After making those modifications, update your command path to include the bin-mapreduce1 directory and set the HADOOP_CONF_DIR environment variable to be the path to the etc/hadoop-mapreduce1 directory. You should now be able to execute Hadoop commands from your local terminal just as you would from the terminal in the virtual machine. You may also want to set the HADOOP_USER_NAME environment variable to cloudera to let you masquerade as the *cloudera* user. When you use the VM directly, you're running as the *cloudera* user.

Further Hadoop tutorials

- Yahoo! Hadoop Tutorial: http://developer.yahoo.com/hadoop/tutorial/
- Cloudera Hadoop Tutorial: http://www.cloudera.com/content/cloudera-content/ cloudera-docs/HadoopTutorial/CDH4/Hadoop-Tutorial.html
- How to Debug MapReduce Programs: http://wiki.apache.org/hadoop/HowToDebugMapReducePrograms

Further Eclipse tutorials

- Genera Eclipse tutorial: http://www.vogella.com/articles/Eclipse/article.html.
- Tutorial on how to use the Eclipse debugger: http://www.vogella.com/articles/ EclipseDebugging/article.html.

3 Task: Write your own Hadoop Job

Now you will write your first MapReduce job to accomplish the following task:

- Write a Hadoop MapReduce program which outputs the number of words that start with each letter. This means that for every letter we want to count the total number of words that start with that letter. In your implementation ignore the letter case, *i.e.*, consider all words as lower case. You can ignore all non-alphabetic characters.
- Run your program over the same input data as above.

What to hand-in: Hand-in the printout of the output file and upload the source code.