Winter 2016

**CS246: Mining Massive Datasets** 

# Hadoop Tutorial

Due 11:59pm January 12, 2016

# General Instructions

The purpose of this tutorial is (1) to get you started with Hadoop and (2) to get you acquainted with the code and homework submission system. Completing the tutorial is optional but by handing in the results in time students will earn 5 points. This tutorial is to be completed individually.

Here you will learn how to write, compile, debug and execute a simple Hadoop program. First part of the assignment 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/cm5/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 deliverable for sections 1 and 2. In section 3, you are asked to write and submit your own MapReduce job

This assignment requires you to upload the code and hand-in the output for Section 3.

All students should submit the output via GradeScope and upload the code via snap.

GradeScope: To register for GradeScope,

- Create an account on GradeScope if you don't have one already.
- Join CS246 course using Entry Code 92B7E9

Upload the code: Put all the code for a single question into a single file and upload it at 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 https://downloads.cloudera.com/demo\_ vm/virtualbox/cloudera-quickstart-vm-5.5.0-0-virtualbox.zip.
- 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* in the *File* dropdown menu. 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.
- To verify that the VM is running and you can access it, open a browser to the URL: <a href="http://localhost:8088">http://localhost:8088</a>. You should see the resource manager UI. The VM uses port forwarding for the common Hadoop ports, so when the VM is running, those ports on localhost will redirect to the VM.
- Optional: Open the Virtual Box preferences ( $File \rightarrow Preferences \rightarrow Network$ ) and select the Adapter 2 tab. Click the Enable Network Adapter checkbox. Select Host-only Adapter. If the list of networks is empty, add a new network. Click OK. If you do this step, you will be able to connect to the running virtual machine via SSH from the host OS at 192.168.56.101. The username and password are 'cloudera'.

#### The virtual machine includes the following software

- CentOS 6.4
- JDK 7 (1.7.0\_67)
- Hadoop 2.5.0
- Eclipse 4.2.6 (Juno)

The virtual machine runs best with 4096MB of RAM, but has been tested to function with 1024MB. Note that at 1024MB, while it did technically function, it was very slow to start up.

## 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 Eclipse. If you just launched the VM, you may have to close the Firefox window to find the Eclipse icon on the desktop.
- 2. Right-click on the *training* node in the Package Explorer and select *Copy*. See Figure 1.

He Fack	age Explorer 🛿	10.000		
	8	\$ 8	~	
Þ 🚰 tra	Oning			
	Ne <u>w</u>			1
	Go Into			
	Open in New Wind	ow		
	Open Type Hierard	hy		F4
	Sho <u>w</u> In			Shift+Alt+W
	<u>С</u> ору			etri+c
	Copy Qualified Na	me		
	Paste			Ctrl+V
	<u>D</u> elete			Delete
	Remove from Con	text	Shift	+Ctrl+Alt+Dowr

Figure 1: Create a Hadoop Project.

3. Right-click on the *training* node in the Package Explorer and select Paste. See Figure 2.

🚦 Package I	Explorer 🛙	-	8
		8 8 9	▽
👂 🥵 training	l.		
	£		_
	Ne <u>w</u>		>
	Go <u>I</u> nto		
	Open in <u>N</u> e	w Window	
	Оре <u>п</u> Туре	Hierarchy	F4
	Show In		Shift+Alt+W >
	Copy		Ctrl+C
	Copy Quali	fied Name	
	Paste		Ctrl+V
	Delete		Delete
	Delete		

Figure 2: Create a Hadoop Project.

4. In the pop-up dialog, enter the new project name in the *Project Name* field and click *OK*. See Figure 3.

e	Copy Project	×
Project name:	WordCount	
☑ Use <u>d</u> efault I	ocation	
Location: //hom	e/cloudera/workspace/WordC	Count Browse
0	Cancel	ОК

Figure 3: Create a Hadoop Project.

5. Modify or replace the stub classes found in the **src** directory as needed.

#### 2.2 Running Hadoop jobs in standalone mode

Once you've created your project and written the source code, to run the project in standalone mode, do the following:

1. Right-click on the project and select  $Run As \rightarrow Run Configurations$ . See Figure 4.

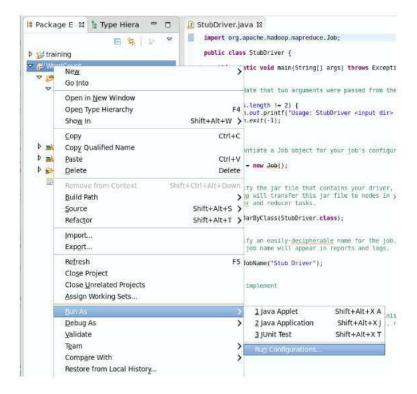


Figure 4: Run a Hadoop Project.

2. In the pop-up dialog, select the *Java Application* node and click the New launch configuration button in the upper left corner. See Figure 5.

	6	Run Configurations	100
Eile Edit Source Refactor	Create, manage, and run Run a Java application	a configurations	
Package Explorer IX     New Ioc     Set training     def WordCount	Inch configuration	Configure launch settings from this dialog:	
	3	Close	949 (J)

Figure 5: Run a Hadoop Project.

3. Enter a name in the *Name* field and the name of the main class in the *Main class* field. See Figure 6.

	Name: Driver	
type filter text 🚽	Main M-Arguments M JRE Classpath Source     Project:     WordCount	Environment Common
E Driver	Main class:	J
U Ju Jurnit Jav StubTest end Maven Build Juj Task Context Test	StubDriver Include system libraries when searching for a main of Include inberited mains when searching for a main of Stgp in main	
Filter matched 7 of 9 items		Apply Revert

Figure 6: Run a Hadoop Project.

4. Switch to the *Arguments* tab and input the required arguments. Click *Apply*. See Figure 7. To run the job immediately, click on the *Run* button. Otherwise click *Close* and complete the following step.

	Name: Driver		
type filter text 🧃	B Main Main Arg	uments 🛛 📷 JRE 🍕 Classpath 🤤 Source 🐻 Environment	Common
Java Applet.  Java Application  Aww_configuration  Je JUnit	Program argun pg100.txt outp		
Ju StubTest	VM arguments	÷	Varjables
Ju Task Context Test			
			Variables
			vanables
	Working direct	ory	vanable <u>s</u>
	Working direct	ory: [s(workspace_loc:WortCount)	vanabie <u>s</u>
	Constant Section 1995		vanables

Figure 7: Run a Hadoop Project.

5. Right-click on the project and select  $Run As \rightarrow Java Application$ . See Figure 8.

Package E      P	E S I V	public class Stu	ie.t	hadoop.mapreduce.lob; river ( id main(String[] args		ť
♥ ## (det ▷ #) £ ▷ #) £ ▷ #) £	Go Into Open in <u>N</u> ew Window Open Type Hierarchy Sho <u>w</u> In	F2 Shift+Alt+W	4	at two arguments were n != 2) { rintf("Usage: StubDri -1);		
▷ ② S ▷ ➡ JRE Sy ▷ ➡ Refere ▷ ➡ conf	<u>C</u> opy Copy Qualified Name <u>P</u> aste <u>D</u> elete	Ctrl+V Delete xt Shift+Ctrl+Alt+Down > Shift+Alt+S >		a Job object for your job's configuration		on.
hadoo	Remove from Context Build Path Source Refactor					
	Import Export			easily-decipherable n me will appear in rep		
	Refresh Close Project Close Unrelated Projects Assign Working Sets	F		("Stub Driver"): ent		
	Bun As	2	5	1 Java Applet	Shift+Alt+X A	
	<u>D</u> ebug As <u>V</u> alidate		>	2 Java Application 3 JUnit Test	Shift+Alt+X] Shift+Alt+X T	m 1
	Tgam Compare With Restore from Local History		>	Run Configurations.	a:	

Figure 8: Run a Hadoop Project.

6. In the pop-up dialog select the main class from the selection list and click OK. See Figure 9.

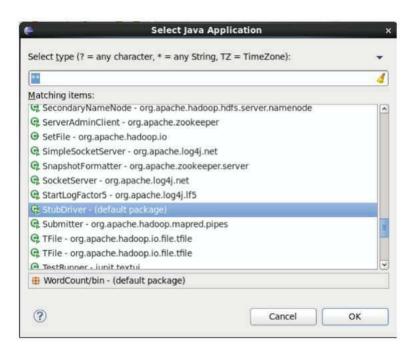


Figure 9: Run a Hadoop Project.

After you have setup the run configuration the first time, you can skip steps 1 and 2 above in subsequent runs, unless you need to change the arguments. You can also create more than one launch configuration if you'd like, such as one for each set of common arguments.

#### 2.3 Running Hadoop in pseudo-distributed mode

Once you've created your project and written the source code, to run the project in pseudodistributed mode, do the following:

1. Right-click on the project and select *Export*. See Figure 10.

Ne <u>w</u> Go Into Open in New Window Ope <u>n</u> Type Hierarchy Sho <u>w</u> In Copy Copy	F4 Shift+Alt+W > Ctrl+C	-	
Ope <u>n</u> Type Hierarchy Sho <u>w</u> In Copy	Shift+Alt+W >		
10000000 0000 0000 00000	Ctrl+C	282	
Paste	Ctrl+V		
Remove from Context	Shift+Ctrl+Alt+Down	2	
<u>S</u> ource Refac <u>t</u> or	Shift+Alt+S > Shift+Alt+T >	8	
Import Exp <u>o</u> rt			
Refresh Clo <u>s</u> e Project Close <u>U</u> nrelated Projects Assian Working Sets	۴5	aration	Co
	Delete Remove from Context Build Path Source Refactor Import Exp <u>o</u> rt Re <u>fresh</u> Clo <u>s</u> e Project	Delete Delete Remove from Context Shift+Ctrl+Alt+Down Build Path  Source Shift+Alt+S  Refactor Shift+Alt+T  Import Export Refresh F5 Close Project Close Unrelated Projects Assign Working Sets	Delete Delete Remove from Context Shift+Ctrl+Alt+Down Build Path Source Shift+Alt+S > Refactor Shift+Alt+T > Import Export Refresh F5 Close Project Close Unrelated Projects Assign Working Sets

Figure 10: Run a Hadoop Project.

2. In the pop-up dialog, expand the Java node and select JAR file. See Figure 11. Click Next  $\,>\,$ 

E Export	
Select	
Export resources into a JAR file on the local file syste	m.
Select an export destination:	
type filter text	4
🕨 🗁 General	
👂 🗁 Install	
🗢 🗁 Java	
a JAR file	
@ Javadoc	
🖓 Runnable JAR file	
<ul> <li>Aun/Debug</li> <li>Tasks</li> </ul>	
(?) < <u>Back</u> <u>Next</u> >	Cancel Einish

Figure 11: Run a Hadoop Project.

3. Enter a path in the JAR file field and click Finish. See Figure 12.

<ul> <li>Export generated class files and resources</li> <li>Export all output folders for checked projects</li> <li>Export Java source files and resources</li> <li>Export refactorings for checked projects. Select</li> </ul>	JAR. .classpath .project hadoop.log	
Select the resources to export:	.classpath .project	
<ul> <li>Export generated glass files and resources</li> <li>Export all output folders for checked projects</li> <li>Export Java source files and resources</li> <li>Export refactorings for checked projects. Select</li> </ul>	.project	
<ul> <li>Export generated class files and resources</li> <li>Export all output folders for checked projects</li> <li>Export Java source files and resources</li> <li>Export refactorings for checked projects. Select</li> </ul>	.project	
<ul> <li>Export generated class files and resources</li> <li>Export all output folders for checked projects</li> <li>Export Java source files and resources</li> <li>Export refactorings for checked projects. Select</li> </ul>	2 Contraction of the second	
<ul> <li>Export generated class files and resources</li> <li>Export all output folders for checked projects</li> <li>Export Java source files and resources</li> <li>Export refactorings for checked projects. Select</li> </ul>	hadoop.log	
<ul> <li>Export all output folders for checked projects</li> <li>Export Java source files and resources</li> <li>Export refactorings for checked projects. Select</li> </ul>		
Select the export destination:	refactorings	
JAR file: /home/cloudera/WordCount.jar	~	Browse
Options:		
☑ Compress the contents of the JAR file		
Add directory entries		
Overwrite existing files without warning		
? < <u>Back</u> <u>Next</u> >		

Figure 12: Run a Hadoop Project.

4. Open a terminal and run the following command:

hadoop jar path/to/file.jar input path output path

After modifications to the source files, repeat all of the above steps to run job again.

### 2.4 Debugging Hadoop jobs

To debug an issue with a job, the easiest approach is to run the job in stand-alone mode and use a debugger. To debug your job, do the following steps:

1. Right-click on the project and select  $Debug As \rightarrow Java Application$ . See Figure 13.

Compare With	> Debug Configurations	
o T <u>e</u> am	> 3 JUnit Test Shift+Alt	+D 7
Validate	Lyava Application Shift (A)	de.
Depuç Ve	1 java Applet Shift+Alt	
Bun As	>	
Assign Working Sets		
Close Unrelated Projects	tis time.	
Close Project		
14	F5	
Export		
impart		
C. P. W. W. W.	/	
Refactor Shift+Alt+T		
Build Path Source shift+Alt+S		
Remove Tram Context Shift+Etri+Alt+Driv	WD .	
	5778 g	
Delete Dele	221	
Copy Qualified Name Poste Cirli-		
Copy Carl-	+0	
Open Type Hierarchy F Show in Shift+Alt+W	F4	
Open in New Window		
Go Into	5	
- WordCount	>	
straining		
E 4		
Package Explo 14 😁 🛛		

Figure 13: Debug a Hadoop project.

2. In the pop-up dialog select the main class from the selection list and click OK. See Figure 14.

2	Select J	ava Application	×
Select type (?	= any character, * = any	y String, TZ = TimeZone):	*
			4
Matching item	s:		
StubDriver	- (default package)		
\$	Work:	space matches	
ANSIBuffer	- jline		
ASMifierCla	assVisitor - org.mockito.a	asm.util	
Q Application	ICLI - org.apache.hadoop	.yam.client.cli	
• ArrayFile -	org.apache.hadoop.io		
BackupNor	le - org.apache.hadoop.h	ndfs.server.namenode	
G Balancer -	org.apache.hadoop.hdfs.	.server.balancer	
Q CLI - org.a	ache.hadoop.mapreduce	e.tools	
CheckClas	Adapter - org.mockito.a	ism.util	
Compressi	onCodecFactory - org.apa	ache.hadoop.io.compress	
H WordCoun	/bin - (default package)	A A A.	
?		Cancel	ОК
100			

Figure 14: Run a Hadoop Project.

You can use the Eclipse debugging features to debug your job execution. See the additional Eclipse tutorials at the end of section 2.6 for help using the Eclipse debugger.

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 8088 of the server, which will the localhost. 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.

#### 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.

- Open Eclipse. If you just launched the VM, you may have to close the Firefox window to find the Eclipse icon on the desktop.
- Right-click on the *training* node in the Package Explorer and select *Copy*. See Figure 15.

🚦 Package Explorer 🛿	-	
	8 \$ 8	~
Ne <u>w</u> Go Into		
Open in <u>N</u> ew Ope <u>n</u> Type Hi Sho <u>w</u> In		Shift+Alt+W
Сору		eti
Copy Qualifie <u>P</u> aste <u>D</u> elete	d Name	Ctrl Del
Remove from	Context	Shift+Ctrl+Alt+Do

Figure 15: Create a Hadoop Project.

• Right-click on the *training* node in the Package Explorer and select *Paste*. See Figure 16.

	- 0	Package Explorer 없
	8 🕸 👂 🔻	
		🚰 training
>		New
		Go Into
	v Window	Open in <u>N</u> e
F4	lierarchy	Ope <u>n</u> Type
Shift+Alt+W >		Sho <u>w</u> In
Ctrl+C		Copy
	ed Name	Copy Qualit
Ctrl+V		Paste
Delete		Delete
Delete		

Figure 16: Create a Hadoop Project.

• In the pop-up dialog, enter the new project name in the *Project Name* field and click *OK*. See Figure 17.

e	Copy Project ×
Project name:	WordCount
☑ Use <u>d</u> efault	location
Location: //hor	me/cloudera/workspace/WordCount
?	Cancel OK

Figure 17: Create a Hadoop Project.

• Create a new package called edu.stanford.cs246.wordcount by right-clicking on the *src* node and selecting  $New \rightarrow Package$ . See Figure 18.

Package ▶ 28 trainin ❤ 18 Words	ng	• •	public class Stu	ne.hadoop.mapreduce.Job;	
▼ ( <b>0</b> ) ▼ <b>(</b>	Ne <u>w</u> Go Into		>	Java Project Project	
0	Open in <u>N</u> ew Wir Ope <u>n</u> Type Hiera Sho <u>w</u> In		F4 Shift+Alt+W 🗲	Package Class Interface	
Þ <mark>mal</mark> ji Þ <mark>mal</mark> i¤ Þ g⇔ c	<u>C</u> opy Cop <u>y</u> Qualified N <u>P</u> aste <u>D</u> elete	ame	Ctrl+C Ctrl+V Delete	Enum Annotation Source Folder Java Working Set	
	Remove from Co Build Path	ntext	Shift+Ctrl+Alt+Down	Folder File Untitled Text File	

Figure 18: Create a Hadoop Project.

• Enter edu.stanford.cs246.wordcount in the *Name* field and click *Finish*. See Figure 19.

E	New J	ava Package	
ava Package Create a new			Ť
Creates folder	s corresponding to package	25.	
Source folder:	WordCount/src		Browse
Name:	edu.stanford.cs246.wordc	ount	

Figure 19: Create a Hadoop Project.

• Create a new class in that package called WordCount by right-clicking on the *edu.stanford.cs246.wordco* node and selecting  $New \rightarrow Class$ . See Figure 20.

<ul> <li>Straining</li> <li>WordCount</li> <li>Src</li> <li>(default pa</li> <li>(b) (2) StubDriv</li> <li>(c) StubDriv</li> <li>(c) StubMag</li> <li>(c) StubRed</li> </ul>	erjava perjava	if (args.length != 2)	String[] args) throws E	om the c
StubTet		>	Java Project	-
and the second se	(MITERS)	24		
🕆 edu.stanfi	Open in New Window		Project	at
🕆 edu.stanfr 👂 📷 JRE System I	The short's an or the state of the state of	S	Project Package	at
👖 edu.stanfi	Open in <u>N</u> ew Window	S	<10400000-040000	at

Figure 20: Create a Hadoop Project.

• In the pop-up dialog, enter WordCount as the Name. See Figure 21.

l <b>ava Class</b> Create a new Java	class.	C
Source fol <u>d</u> er:	WordCount/src	Browse
Pac <u>k</u> age:	edu.stanford.cs246.wordcount	Browse
Enclosing type:		Browse
Na <u>m</u> e:	WordCount	
Modifiers:	public O default O prigate C     abstract O final O status	O projected
<u>S</u> uperclass:	java.lang.Object	Browsg
Interfaces:		Add
		Bernove
Which method stub	s would you like to create?  public static void main(String[] args) <u>C</u> onstructors from superclass  Inherited abstract methods	
Do you want to add	comments? (Configure templates and defa	ult value <u>here</u> )
0		Cancel Finish

Figure 21: Create a Hadoop Project.

• 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 22.

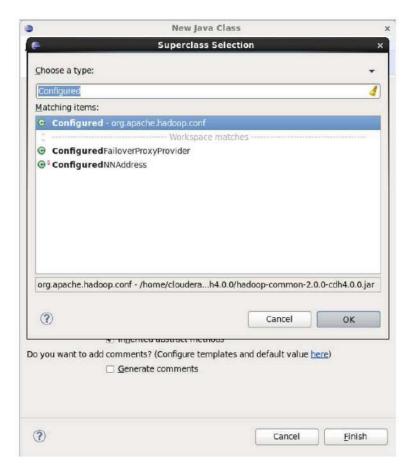


Figure 22: Create a java file.

• 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 23.

•	Implemented Interfaces Selection ×	,
Java Class Create a new	Choose interfaces:	C
	Tool 4	-
Source folder:	Matching items:	Browse
Package:	Tool - org.apache.hadoop.util - /home/cloudera/.m2/rep     Workspace matches	Browse
Enclosing t	<ul> <li>Tool - javax.tools - [jdk1.6.0_32]</li> <li>ToolkitThreadBlockedHandler</li> </ul>	Browse
Na <u>m</u> e:		
Modifiers:		
Superclass:		Browse
Interfaces:		Add
		Remove
Which method		
	org.apache.hadoop.util - /hp-common-2.0.0-cdh4.0.0.jar	
	org.apache.naooop.uur - /np-common-2.0.0-cun4.0.0.jar	
Do you want t	1	
	Add         Cancel         OK	
-		_
(?)	Cancel	Einish

Figure 23: Create a java file.

• Check the boxes for *public static void main(String args[])* and *Inherited abstract methods* and click the *Finish* button. See Figure 24.

ava Class	• -200-	C
Create a new Java	class.	9
Source fol <u>d</u> er:	WordCount/src	Browse
Pac <u>k</u> age:	edu.stanford.cs246.wordcount	Browse
Enclosing type:		Browse.
Na <u>m</u> e:	WordCount	
Modifiers:	public O default O private O project     abstract      final      static	ed
Superclass:	org.apache.hadoop.conf.Configured	Browse
Interfaces:	🔹 org.apache.hadoop.util.Tool	<u>A</u> dd
		Bemove
Which method stub	s would you like to create?	
	Dublic static void main(String[] args)	
	Constructors from superclass	
	Inherited abstract methods	
Do you want to add	comments? (Configure templates and default value	here)
	Generate comments	
	<ul> <li>public static void main(String[) args)</li> <li><u>C</u>onstructors from superclass</li> <li>Inherited abstract methods</li> </ul>	

Figure 24: Create WordCount.java.

• You will now have a rough skeleton of a Java file as in Figure 25. You can now add code to this class to implement your Hadoop job.

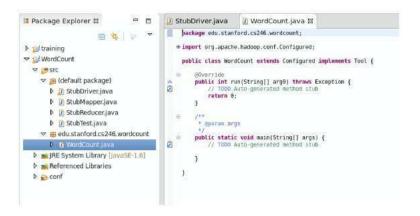


Figure 25: Create WordCount.java.

• 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 I Package Explorer 없 - -🗊 StubDriver.java 👔 WordCount.java 🖾 backage edu.stanford.cs246.wordcount; 8 4 2 7 Ē. package edu.stanford.cs246.wordcount; import org.apache.hadoop.comf.configuration; import org.apache.hadoop.comf.configurad; import org.apache.hadoop.comf.configurad; import org.apache.hadoop.io.lottoritable; import org.apache.hadoop.io.tofwirtiable; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Napper; import org.apache.hadoop.mapreduce.Napper; import org.apache.hadoop.mapreduce.Napper; import org.apache.hadoop.mapreduce.Napper; import org.apache.hadoop.mapreduce.Napper; import org.apache.hadoop.mapreduce.Napper; import org.apache.hadoop.wapreduce.Napper; import org.apache.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoop.hadoo b tal training ▼ Se WordCount e esrc ♥ (default package) 1 StubDriver.iava 0 🗊 StubMapper.java D D StubReducer.java D Ut StubTest Java v III edu.stanford.cs246.wordcount D 🖉 WordCount.java IRE System Library []avaSE-1.6] blic class WordCount extends Configured implements Tool {
 public static void main(String[] args) throws Exception {
 int res = ToolRummer.run(new Configuration(), new WordCount(), args); Referenced Libraries Þ ⊯conf System.exit(res); public int run(String[) args) throws Exception {

file. See Figure 26. The code in WordCount.java calculates the frequency of each word in a given dataset.

Figure 26: Create WordCount.java.

• Download the *Complete Works of William Shakespeare* from Project Gutenberg at http://www.gutenberg.org/cache/epub/100/pg100.txt. You can do this simply with cURL, but you also have to be aware of the byte order mark (BOM). You can download the file and remove the BOM in one line by opening a terminal, changing to the ~/workspace/WordCount directory, and running the following command:

curl http://www.gutenberg.org/cache/epub/100/pg100.txt | perl -pe 's/^\xEF\xBB  $\xBF//' > pg100.txt$ 

If you copy the above command beware the quotes as the copy/paste will likely mistranslate them.

• Right-click on the project and select  $Run As \rightarrow Run Configurations$ . See Figure 27.

Packa	age Explorer 🛿 🐡 🖻	1 👔 StubDriver.ja	iva 🗾 WordCount.java 😫	
▶ 💕 tra	ining	<pre>= import java import org.</pre>	.stanford.cs246.wordcount; .io.IOException; apache.hadoop.conf.Configuration;	
▼ <u>6</u> ▼ 8	Ne <u>w</u> Go Into	,	<pre>pache.hadoop.conf.Configured; pache.hadoop.fs.Path; pache.hadoop.io.IntWritable;</pre>	
	Open in <u>N</u> ew Window Ope <u>n</u> Type Hierarchy Sho <u>w</u> in	pache.hadoop.io.longWritable; pache.hadoop.io.Text; F4 pache.hadoop.mapreduce.Job; pache.hadoop.mapreduce.Mapper; Shift+Alt+W > pache.hadoop.mapreduce.Reducer;		
× A	<u>C</u> opy Cop <u>y</u> Qualified Name Paste <u>D</u> elete	Ctrl+C Ctrl+V Delete	pache hadoop.mapreduce.llb.output.FileOu	
D a D <u>s</u>	Build Path Source Shift+Alt+S	Shift+Ctrl+Alt+Down Shift+Alt+S Shift+Alt+T	<pre>atic void main(String[] args) throws &gt; s = ToolRunner.run(new Configuration &gt;</pre>	
	Import Exp <u>o</u> rt		t run(String[] args) throws Exception {	
	Refresh Close Project Close Unrelated Projects Assign Working Sets	F5	Javadoc 🔍 Declaration 🔛 Console 🛱 splay at this time.	
	Bun As	>	1 Java Applet Shift+Alt+X A	
-	Debug As Yalidate	>	2 Java Application Shift+Alt+X J 3 JUnit Test Shift+Alt+X T	
VordCo	Team Compare With	>	Ru <u>n</u> Configurations	

Figure 27: Run WordCount.java.

• In the pop-up dialog, select the *Java Application* node and click the New launch configuration button in the upper left corner. See Figure 28.

	é išun Configuratione.				
ile Edit Source Refactor	N Create, manage, and run configurations Foun a java application				
Package Explorer II New lat b ⊘itraining b del VontCount	Configure launch settings from this dialog: Press the 'New' button to create a configuration of the Press the 'New' button to create a configuration of the Press the 'Duplicate' button to copy the selected configuration Just Stublest Maxen Build Just Scublest Press the 'Dilect' button to compute filtering options. Edit or view an existing configuration by selecting it. Configure launch perspective settings from the 'Presspective' Filter matched 6 of 8 items	guration. guration.			
	0	Close Bin			

Figure 28: Run WordCount.java.

• Enter a name in the Name field and WordCount in the Main class field. See Figure 29.

<b>2</b> 11	Run Configurations	
Create, manage, and run Run a Java application	configurations	
1 B * B *	Name: Driver	
type filter text. 🧃	G Main 🐶 Arguments 📸 JRE 🕎 Classpath 💱 Source 🖉 Environmen	nt 🖾 Common
😇 Java Applet 😎 🗊 Java Application	Project: MordCount	Browse
Driver	Main class:	
<ul> <li>Jo JUnit</li> <li>Jo StubTest</li> </ul>	edu.stanford.cs246.wordcount.WordCount	Search
«ể Maren Build ፓ <sub>ርያ</sub> Task: Context Test	Include system libraries when searching for a main class Include inperited mains when searching for a main class Stgp in main	y Reyert
Filter matched 7 of 9 items	Apply	<u>Keyert</u>
(?)	Clos	se Bun

Figure 29: Run WordCount.java.

• Switch to the Arguments tab and put pg100.txt output in the Program arguments field. See Figure 30. Click Apply and Close.

e		Run Confi	igurations			8
Create, manage, and run Run a Java application	configurations					
C . × 0 > ·	Name: Driver					
type filter text 🛛 🧃	G Main M- Arg	juments MIRE	👋 Classpa	th 🦆 Source 🛙	Environment	ommon
🔊 Java Applet	Program argun	ments:				
Java Application	pg100.txt out	put				
⊽ JujUnit						Varjables
Ju StubTest. Maven Build Jg Task Context Test	VM arguments					
						Variables
	Working direct	iory:				
	Default:     Other:	\${workspace	Joc.WordCo	unt)		
				Vorkspace)	Elle System	/anabigs)
Filter matched 7 of 9 items					Apply	Reyert
1					Close	Bun

Figure 30: Run WordCount.java.

• Right-click on the project and select  $Run As \rightarrow Java Application$ . See Figure 31.

Pack	age Explorer 🛙 🗢	🗆 🔬 StubDriver		
Þ 🥵 tra	ining	⇒import ja	u.stanford.cs246.wordcount; a.io.IOException; .apache.hadoop.conf.Configurati	
▼ (#-) ▼ (	Ne <u>w</u> Go Into	Import or	<pre>.apache hadoop.conf.Configurati ) ipache hadoop.conf.Configurad; ) pache hadoop.fs.Path; ) pache hadoop.io.IntWritable;</pre>	an;
4	Open in <u>N</u> ew Window Ope <u>n</u> Type Hierarchy Sho <u>w</u> In	F Shift+Alt+W	ipacite inedece inepreduce incodece	ń:
× P x	<u>C</u> opy Copy Qualified Name <u>P</u> aste Delete	Ctrl+ Ctrl+ Delet	<pre>pache.hadoop.mapreduce.lib.ou pache.hadoop.mapreduce.lib.ou pache.hadoop.util.Tool;</pre>	out.TextInputH tput.FileOutpu
D a D a	Remove from Context Build Path	Shift+Ctrl+Alt+Dow	WordCount extends Configured satic void main(String[] args) s - ToolRunner.run(new Config	throws Except
	Source Refactor	Shift+Alt+S Shift+Alt+T	h.exit(res):	
	Import Export		t run(String]] args) throws E	cception {
	Refresh Close Project Close <u>U</u> nrelated Projects <u>A</u> ssign Working Sets	,	Javadoc 🔒 Declaration 🔲 c	onsole 2
20	Bun As		1 Java Applet Shif	t+Alt+X A
	Debug As		2 Zijava Application Shi	Re-Alt-X J
	Validate		<u>3</u> JUnit Test Shif	t+Alt+X T
	Team		Run Configurations	
/ordCoi	Compare With			

Figure 31: Run WordCount.java.

• In the pop-up dialog select *WordCount - edu.stanford.cs246.wordcount* from the selection list and click *OK*. See Figure 32.

	Select Ja	va Application		•
Select <u>t</u> ype (? =	any character, * = any !	String, TZ = TimeZone	):	•
				4
Matching items:				
G StubDriver - (	default package)			
G WordCount -	edu.stanford.cs246.wor	dcount		
S	······ Worksp	ace matches		-
ANSIBuffer - j	line			
ASMifierClass	Visitor - org.mockito.as	im.util		
C ApplicationCl	I - org.apache.hadoop.y	/arn.client.cli		
ArrayFile - or	g.apache.hadoop.io			
G BackupNode	- org.apache.hadoop.hd	fs.server.namenode		
G Balancer - on	g.apache.hadoop.hdfs.s	erver.balancer		
🤹 CLI - org.apad	che.hadoop.mapreduce.	tools		
CheckClassA	dapter - org.mockito.asi	m.util		~
the advertise of and	cs246.wordcount			

Figure 32: Export a hadoop project.

You will see the command output in the console window, and if the job succeeds, you'll find the results in the ~/workspace/WordCount/output directory. If the job fails complaining that it cannot find the input file, make sure that the pg100.txt file is located in the ~/workspace/WordCount directory.

• Right-click on the project and select *Export*. See Figure 33.

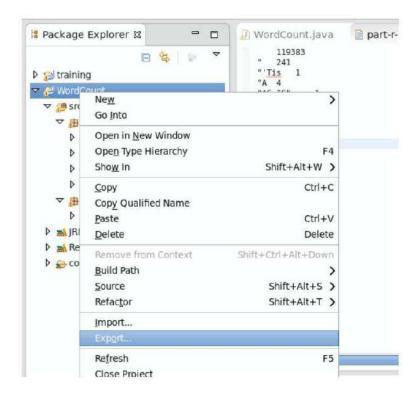


Figure 33: Run WordCount.java.

• In the pop-up dialog, expand the Java node and select JAR file. See Figure 34. Click Next >

	3.42	Export		
elect				X
Export resources in	nto a JAR file on the	local file system.		Ľ
Select an export d	estination:			
type filter text				4
👂 😂 General				
👂 📴 Install				
🗢 📴 Java				
📕 JAR file				
Javadoc				
🔓 Runnable	JAR file			
Run/Debug Analysis				
Team				
(?)	< Back	Next >	Cancel	Finish

Figure 34: Export a hadoop project.

• Enter /home/cloudera/wordcount.jar in the JAR file field and click *Finish*. See Figure 35.

¢	JAR Export	×
JAR File Specification		-
Define which resources should be ex	xported into the JAR.	
Colord the second second		
Select the resources to export:	C D alassath	
Carling      Count      Count	Classpath	
V Eas Wordcount	<ul> <li>✓ I .project</li> <li>✓ hadoop.log</li> </ul>	
Export generated class files and	l resources	
Export all output folders for che	cked projects	
Export Java source files and reso	ources	
Export refactorings for checked	projects. Select refactorings	
Select the export destination:		
JAR file: /home/cloudera/WordCou	ntjar	✓ Browse
Options:		
☑ Compress the contents of the JA	R file	
Add directory entries		
Overwrite existing files without	warning	
(?) < Back	Next > Cancel	Finish
- Eock		

Figure 35: Export a hadoop project.

If you see an error dialog warning that the project compiled with warnings, you can simply click OK.

• Open a terminal in your VM and traverse to the folder /home/cloudera and run the following commands:

hadoop fs -put workspace/WordCount/pg100.txt

hadoop jar WordCount.jar edu.stanford.cs246.wordcount.WordCount pg100.txt
output

• 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 36.

[cloudera@loc	alhost Desktop]\$ hadoop	fs -ls output	
Found 3 items			
-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-rr	3 cloudera cloudera	720989 2014-01-01 16:22 output/part-r-00000	)

Figure 36: 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 37

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

Figure 37: Run WordCount job.

• To view the job's logs, open the browser in the VM and point it to http://localhost: 8088 as in Figure 38

0.0.0.0 Haddoop Hap/Reduce Ad	9	_		0.0.0.0	Hadoop M	lap/Reduc	ce Admini	istration -	Mozilla Fi	refox			- 0
Incachos: 50030jdbrackejsp     Most Visited     Cloudera Manager Hue HDFS NameNode * Hadoop jobiTacker HBase Master Solr      Insteining Running     Total     Sole     S	file Edit	View Hist	ory <u>B</u> ookm	arks <u>T</u> oo	is <u>H</u> elp								
Most Visited (Clouders Clouders Manager Hue HDFS NameNode Hadson JobTracker HBase Master Solr         Muster Summary (Heap Size is 81.06 MB/1021.94 MB)       Containing Reduce Mask Submissions Nedes Sole Sole Sole Sole Sole Sole Sole Sole	0.0.0.0 F	ladoop Map	Reduce Ad	÷									
Numery (Heap Size is 81.06 MB/1021.94 MB)       Concurrent Map Map Map Map Map Map Map Map Map Map	💠 🖻 io	calhost 500	30/jobtracke	isp						合く間	- Google	1	約 1
Numery (Heap Size is 81.06 MB/1021.94 MB)       Concurrent Map Map Map Map Map Map Map Map Map Map	Most Vis		loudera 🖂	Cloudera	Manager	Hue	HDES Nam	eNode I h	iardoon lobi	Tracker	HBase Mast	er Solr	
Map     Reduce     Stots     Nodes     Map     Stots     Stots     Stots     Stots     Capacity     Task, Capacity     Task, Task, Capacity     Task, Task, Capacity     Task, Tas													Quick Link
Cheduling Information         Provide Name State Scheduling Information         Provide Name State Scheduling Information         Provide Name Scheduling Name Scheduling Name Scheduling Name Scheduling Information         Information Info	Map	Reduce ,		Nodes	Map	Reduce	Map	Reduce	Task	Task			Excluded Nodes
State Scheduling information         result       running       N/A         State Scheduling information         Information State Scheduling information         Information State Scheduling information         Information State Scheduling information         Information Information         Information Information	0	0 1	l.	1	0	0	0	0	2	2	4.00	0	0
jobid Priority User Name Map 2, Total Complete C	Running	senth \$200° wi	100 C 100	an(y in the i	Net Held and "X	] 2001 in all field	к.						
Jobid Priority User Name Complete Total Complete Complete Complete Completed Information		ted Jobs			T	1	w Wese I	-	() Trem on our	ur Danne		an ar	Manual
1 1 1 NA NA	9	obid	Priority	User	Name	Comple						, Scheduli	
	jab_20140	1011523_00	01 NORMAL	cloudera	WordCount	100.00%	1	1	100.00%	1	1	NA.	NA
			- 331 3										

Figure 38: Run WordCount job.

• Click on the link for the completed job. See Figure 39.

9			Hadoop	job_2014	01011523	0001	0.0.0.0	Mozilla i	Firefox		- 0.2
Eile Edi	t <u>v</u> iew Hi <u>s</u> to	ory <u>B</u> ookma	arks <u>T</u> ools	<u>H</u> elp							
Hadoo	p job_2014010	011523_00	+								
4 0	localhost 500	0.jobdetails	sp?]obid=	jab_20140	1011523_0	0016ref	resti=0		101	- Google	M 🖌
Most	/isited~C	loudera 🖂 (	Cloudera M	tanager (	Hue OH	DFS Na	meNode /	Hadoop jo	bTracker 🔅 H	Base Master 🔅 So	r
Hado	oop job_	20140	10115	23_00	01 on	0.0	0.0				-
Job File: J Submit H Submit H Job-ACLs Job Sctup Status: S Status: S S Status: S S S S S S S S S S S S S S S S S S S	t: Wed (an 01 16 at: Wed (an 01 1 in: 24sec ap: Successfu)	caldumain 17.0.0.1 diowed 22.09 PST 201 6.22.33 PST 20	19 014				Failed/Kil	leet			
Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Task Atter				
map	100.00%	1	0	0	1	0		0 / D			
reduce	100.00%	1	0	0	Į.	0		0/0			
		1		Count	er			Мар	Reduce	Total	
	-	FILE: Numb	er of bytes	read				1.508.400	983,464	2.491.864	
		FILE: Numb	er 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 I	ead operati	ons			0	0	0	
-	em Counters	FILE: Numb	er of write (	perations				5.27	0	0	
								0	. 0		
	and countries	HDFS: Numi	ber of bytes	s read				0 5,590,012	0	5,590,012	

Figure 39: View WordCount job logs.

• Click the link for the map tasks. See Figure 40.

Hadoop map task list for job_2 🕸							
🔶 🕼 localhost.50030/jobtasks.jsp/job/							49 🐔
Most Visited 🛩 🗌 Cloudera 📄 Clouder					laster	Solr	
Hadoop map task list I	for job_20	1401	011523_000	1 on 0.0.0.0			
All Tasks							
2	Complete	Status	Start Time	Finish Time	0.0000	Counters	
Task	Compiece		Sector Chines	Finish Time	Errors	Counters	
Task (ask_201401011523_0001_m_00	100 000			1-jan-2014 16:22:23 (6sec)	Errors	21	
TO TAR PLANKA PERIOD	100 000				LITOPS		

Figure 40: View WordCount job logs.

• Click the link for the first attempt. See Figure 41.

File Edit View History Bookmarks To	ols <u>H</u> elp								
f Hadoop Task Details 🛛 🚔									
🜪 🔮 kocalhost 50036/task details jsp?tip	id=task_2014010115	23_0001_m_0	00000	<u>_</u> ,	· 61 14-	Google		6	8 🖀
Most Visited~ Cloudera Cloudera	Manager (Hue (	HDFS Name	Node P Had	loop jobTrack	cer ()HBas	e Maste	r 🔿 5	oir	
Job job_201401011523_	0001								
All Task Attempts									
Task Attempts	Machine	Status	Progress	Start Time	Finish Time	Errors	Task Logs	Counters	Action
attempt_201401011523_8001_m_080000_0	/default /ocalhost.localdomain	SUCCEEDED	100.00%	1-jan-2014 16:22:16	1-jan-2014 16:22:22 (6sec)		Last 4KB Last 6KB All	21	
Input Split Locations									
/default/localhest.localdomain									
Go bect to the job Go bect to jobTracker									
Hedoop. 2014.									

Figure 41: View WordCount job logs.

• Click the link for the full logs. See Figure 42.

File Edit View	Task Logs: 'attempt_201401011523_0001_m_000000_0' - Mozilla Firefox	
	v Hi <u>s</u> tory <u>B</u> ookmarks Tools <u>H</u> elp	
Task Logs: 'a	ttempt_20140101	
🔹 🔮 localh	ost.localdomain:50560/tasklog?attemp0d=attempt_201401011523_0001_m_0000000	1
Most Visited	- Cloudera Cloudera Manager Hue HDF5 NameNode Madoop JobTracker HBase Master Solr	
stdeut logs		
stderr logs		_
syslog logs		
2014 01 01 167 2014 01 01 01 01 167 2014 01 01 01 167 2014 01 01 01 01 167 2014 01 01 01 01 00 2014 01 01 00 2014 01 01 00 2014 00 20	22-17,757 WARM mapreduce.(curters) Group org.apache.hadoop.mapred.Task&Counter 15 deprecated. Use org.mpache.hadoop.mapred.22:18,66 FW WARM org.apache.hadoop.comf.comfiguration: ression.id is deprecated. Instead, see dfs.metrics.session.id 22:18,06 FW WARM org.apache.hadoop.comf.comfiguration: ression.id is deprecated. Instead, see dfs.metrics.session.id 22:18,06 FW WARM org.apache.hadoop.metrics.jm.virbetriss: initializing JVM Petrics: with procession.ed 22:18,06 FW Warm.apache.hadoop.metrics.jm.virbetriss: initializing JVM Petrics: with procession.ed 20:18 00 FW org.apache.hadoop.metrics.jm.virbetriss: initializing JVM Petrics: org.apache.hadoop.util.linuffeseurceCalc 20:19 FW org.apache.hadoop.metrics.jm.virbetriss: initializing JVM Petrics.JW Petrics.JW org.apache.hadoop.ongle.JW war/classics.JW org.apache.hadoop.metrics.JW org.apache.hadoop.metrics.JW org.apache.hadoop.metrics.JW org.apache.JW org.apache.hadoop.metrics.JW org.apache.JW org.apache.hadoop.metrics.JW org.apache.JW offst: Los.JW org.apache.JW offst: JW org.apache.JW org.apache.JW org.apache.JW offst: Los.JW of JW org.apache.JW offst: JW offs	ulat

Figure 42: View WordCount job logs.

### 2.6 Using your local machine for development

If you'd rather use your own development environment instead of working in the IDE, follow these steps:

1. Make sure that you have an entry for localhost.localdomain in your /etc/hosts file, e.g.

127.0.0.1 localhost localhost.localdomain

- Install a copy of Hadoop locally. The easiest way to do that is to simply download the archive from http://archive.cloudera.com/cdh5/cdh/5/hadoop-latest.tar. gz and unpack it.
- 3. In the unpacked archive, you'll find a etc/hadoop directory. In that directory, open the core-site.xml file and modify it as follows:

4. Next, open the yarn-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>yarn.resourcemanager.hostname</name>
<value>192.168.56.101</value>
</property>
</configuration>
```

You can now run the Hadoop binaries located in the **bin** directory in the archive, and they will connect to the cluster running in your virtual machine.

#### Further Hadoop tutorials

- Yahoo! Hadoop Tutorial: http://developer.yahoo.com/hadoop/tutorial/
- Cloudera Hadoop Tutorial: http://www.cloudera.com/content/www/en-us/training/library/tutorials.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.