Hadoop "Hello World" – WordCount example

The main goal for this assignment is to familiarize yourself with running Hadoop in several modes: standalone mode on your own computers (using the local file system for input/output, no HDFS – mainly useful for development/debugging purposes), pseudo-distributed environment (a.k.a., single-node cluster) on your own, and also in a fully distributed mode on the Beocat Hadoop cluster.

To get a Beocat account, please follow the link:
https://account.beocat.cis.ksu.edu

Information about Beocat's Hadoop cluster is available at:
http://support.beocat.cis.ksu.edu/BeocatDocs/index.php/Hadoop

There are many releases of Hadoop available, but at a high level, we can characterize these releases into Hadoop 1.x and Hadoop 2.x. Hadoop 1 was designed with large MapReduce batch jobs in mind (which is what we usually need for many information retrieval tasks). Hadoop 2 was designed to also allow running more interactive/specialized processing models (such as interactive querying and streaming data applications) simultaneously with MapReduce jobs. Two main technical advances in Hadoop 2 are the HDFS federation (multiple Namenode servers manage namespaces) and the resource manager YARN. YARN stands for Yet Another Resource Negotiator, and can be seen as a large-scale, distributed operating system for big data applications. You can read more about Hadoop 1 vs Hadoop 2 differences at:
http://www.tomsitpro.com/articles/hadoop-2-vs-1,2-718.html

The Hadoop version available on Beocat is Hadoop 2.6.0 provided by Cloudera (https://archive.cloudera.com/cdh5/cdh/5/hadoop/).

To run Hadoop on your own machines, you will need to download and install a stable release of Hadoop (Hadoop 2.6.0 if you want to be fully compatible with Beocat, but other Hadoop 2.x.x. releases should also work fine – I used Hadoop 2.7.1 myself). Hadoop releases can be found at any of the mirrors listed at:

http://www.apache.org/dyn/closer.cgi/hadoop/common

To install Hadoop on a Windows machine, you will first need to install Ubuntu VM (make sure that the VM is provided enough RAM and cores for good performance).

You can follow a tutorial at one of the following links:

http://pingax.com/install-hadoop2-6-0-on-ubuntu/
http://www.bogotobogo.com/Hadoop/BigData_hadoop_Install_on_ubuntu_single_node_cluster.php
http://stackoverflow.com/questions/32002432/install-hadoop-2-6-0-in-ubuntu-14-0
https://gist.github.com/piyasde/17d4f7bc97c0f0820d40
http://qursaan-howto.blogspot.com/2014/12/how-to-install-hadoop-as-single-node-on.html
https://www.youtube.com/watch?v=SaVFs_iDMPo

Alternatively, you can also install the Cloudera QuickStart VM from:
http://www.cloudera.com/downloads/quickstart_vms/5-8.html

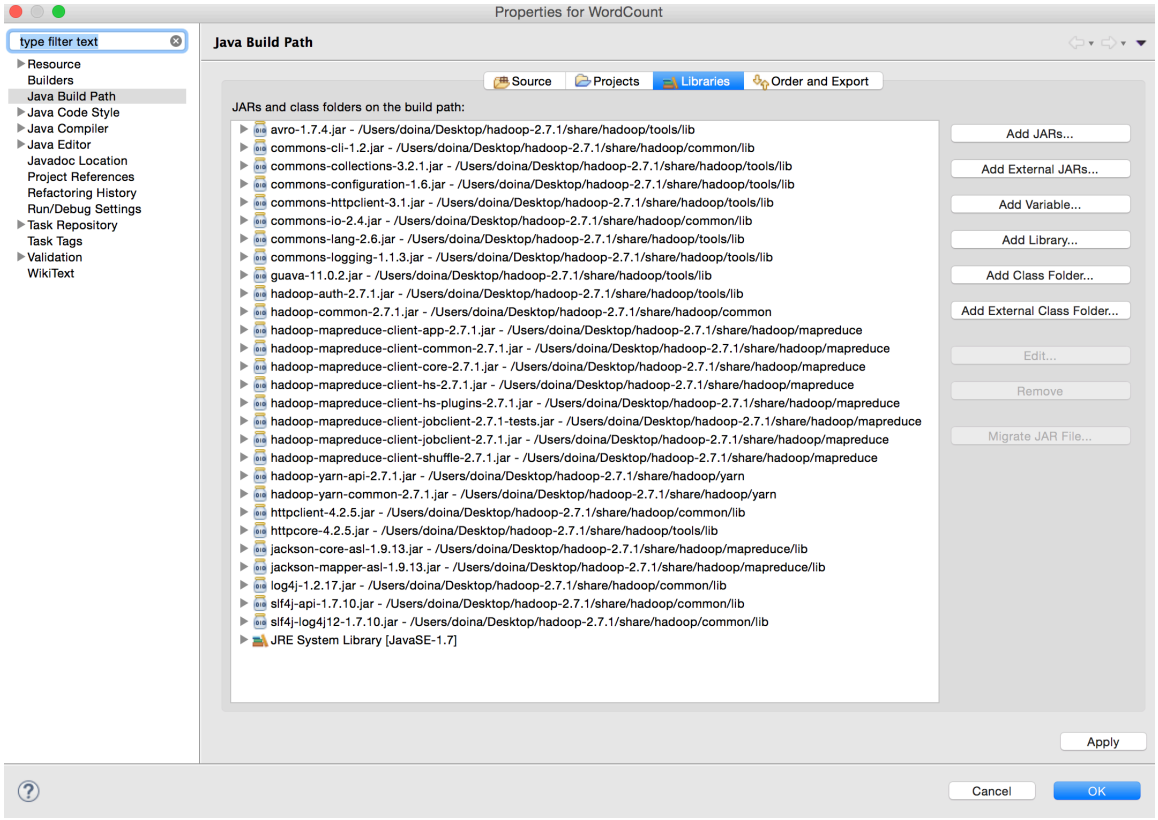
After installing the Ubuntu VM and Hadoop, I recommend installing Eclipse (or your favorite IDE) in Ubuntu.

In what follows, you will use the WordCount example that comes with the Hadoop release that you downloaded. We will run the WordCount on a sample text collection: the complete works of Shakespeare – the corresponding files are available on KSOL Canvas.

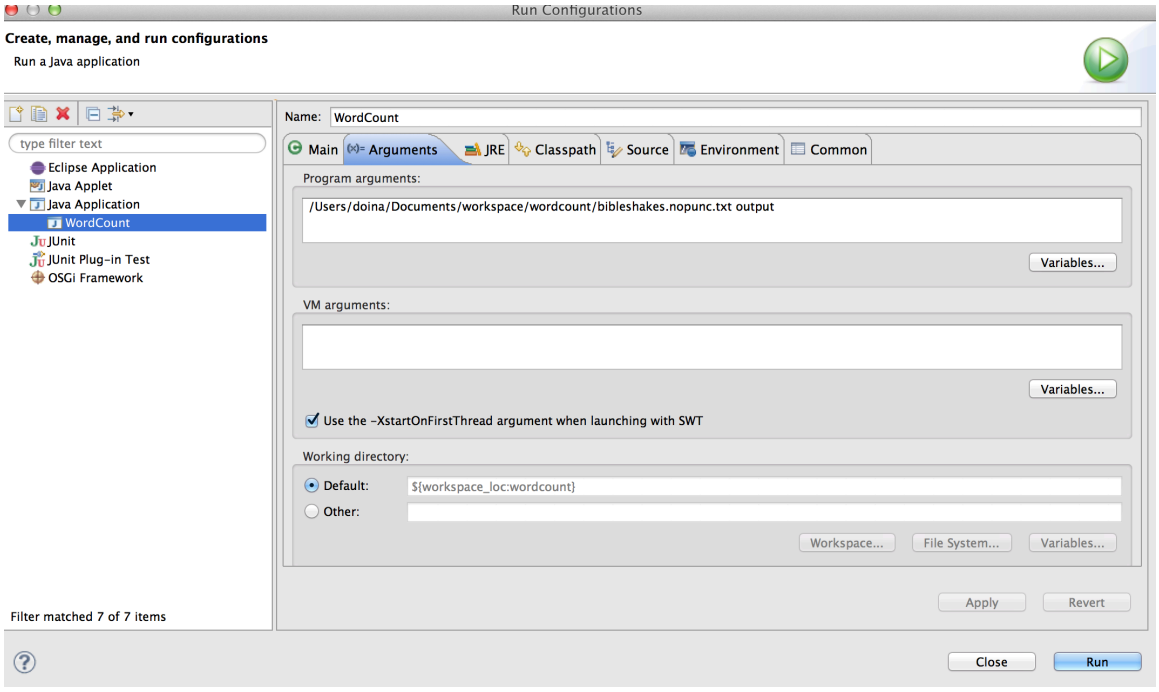
You will need to compile the Hadoop WordCount.java class into a JAR file (the instructions below assume that you are using the Eclipse IDE):

1. **Create a new Java Project.** Launch Eclipse, and from the File Menu select New, then use the Wizard to create a new Java Project. Enter a project name, e.g., WordCount. Click Finish.

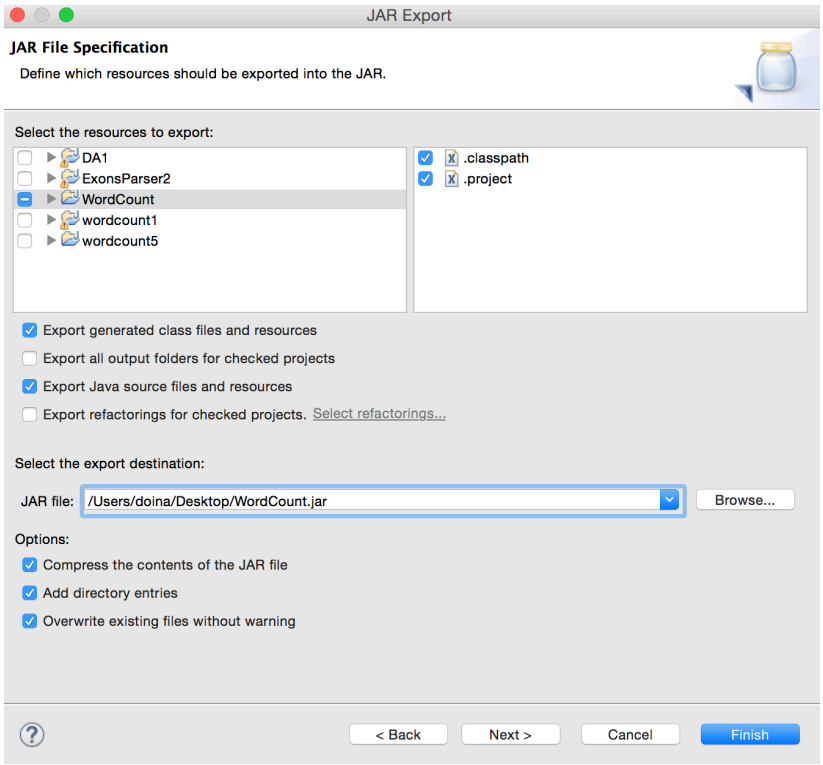
2. **Add Hadoop library to project.** In Eclipse, right-click (control-click), on your project, go to Configure Build Path, then Add External JARs. Browse to the Hadoop folder and select the jars you want to import, then click Open. Please see the screenshot below for a list of jars that are needed for the WordCount example to compile and run.



3. **Add source code file.** Copy the WordCount.java code to the project source directory. Eclipse will compile the file as soon as you save it. At this point, you should be able to run the program in standalone mode from Eclipse, if you specify the input and output directories as shown below. The good thing about running Hadoop in standalone mode using Eclipse is that you can debug your code while developing it. This is useful especially when writing larger programs.



4. **Export JAR file.** From the File Menu, select Export. From under Java select Runnable JAR file, click Next. Make sure the `Extract required libraries into generated JAR` checkbox is checked. Select an export destination for your JAR file - you can use your Desktop, or some other directory. For simplicity, name the file WordCount.jar.



Once the .jar file is created, you can also run the WordCount from command line as shown below (in standalone mode, the local file system is used):

$ bin/hadoop jar WordCount.jar WordCount <in-dir> <out-dir>

Next, you will need run the WordCount program in a single-node/pseudo-distributed mode on the Shakespeare file.

Make sure you start all the Hadoop daemons. Also, put the input data <in-dir> into the HDFS before you run Hadoop and, at the end, copy the output <out-dir> to your local file system.

To run the program, the command syntax is the same as the one for standalone mode, except that <in-dir> and <out-dir> are now HDFS locations:

$ bin/hadoop jar WordCount.jar WordCount <in-dir> <out-dir>

When you are done with the output files for a run, you should delete the output directory. Hadoop will not automatically do this for you, and it will throw an error if you run it while there is an old output directory. To do this, execute: $ hadoop dfs -rmr output

Finally, run the WordCount program on Beocat.

For more details on how to run the WordCount example, please see:

http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Example:_WordCount_v2.7.1

Other Hadoop resources:

Hadoop: http://hadoop.apache.org/
Hadoop wiki: http://wiki.apache.org/hadoop/
FileSystemShell: http://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-common/FileSystemShell.html