**8.Aim: Hadoop commands**

Using the command line interface

In this part, we will explore some basic HDFS commands. All HDFS commands start with ***hadoop*** followed by ***dfs*** (distributed file system) or ***fs*** (file system) followed by a dash, and the command. Many HDFS commands are similar to UNIX commands. For details, refer to the *Hadoop Command Guide* and *Hadoop FS Shell Guide*.

We will start with the **hadoop fs** **–ls** command which returns the list of files and directories with permission information.

Ensure the Hadoop components are all started, and from the same Gnome terminal window as before (and logged on as *biadmin*), follow these instructions:

1. List the contents of the root directory.



**hadoop fs -ls /**

****

1. To list the contents of the /user/biadmin directory, execute:



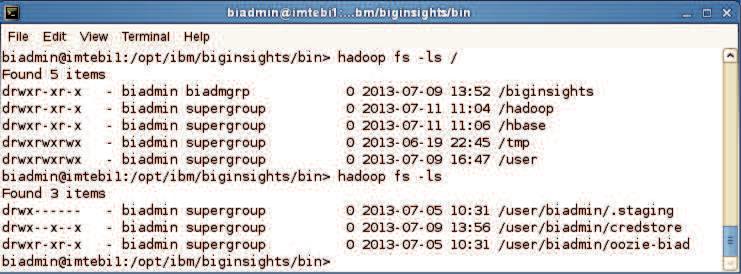
**hadoop fs -ls**

or



**hadoop fs -ls /user/biadmin**

Note that in the first command there was no directory referenced, but it is equivalent to the second command where /user/biadmin is explicitly specified. Each user will get its own home directory under /user. For example, in the case of user biadmin, his home directory is /user/biadmin. Any command where there is no explicit directory specified will be relative to the user’s home directory.



1. To create the directory ***myTestDir*** you can issue the following command:



**hadoop fs -mkdir myTestDir**

Where was this directory created? As mentioned in the previous step, any relative paths will be using the user’s home directory.

1. Issue the ls command again to see the subdirectory myTestDir:

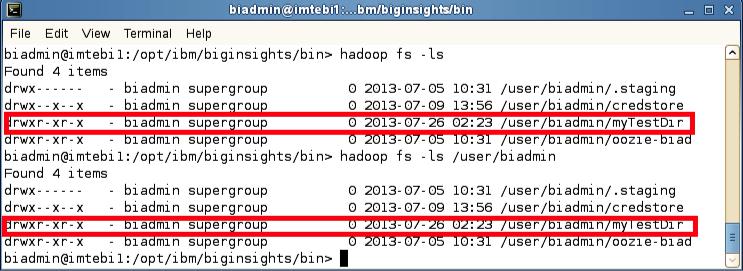


**hadoop fs -ls**

or



**hadoop fs -ls /user/biadmin**

****

To use HDFS commands recursively generally you add an “r” to the HDFS command (In the Linux shell this is generally done with the “-R” argument).

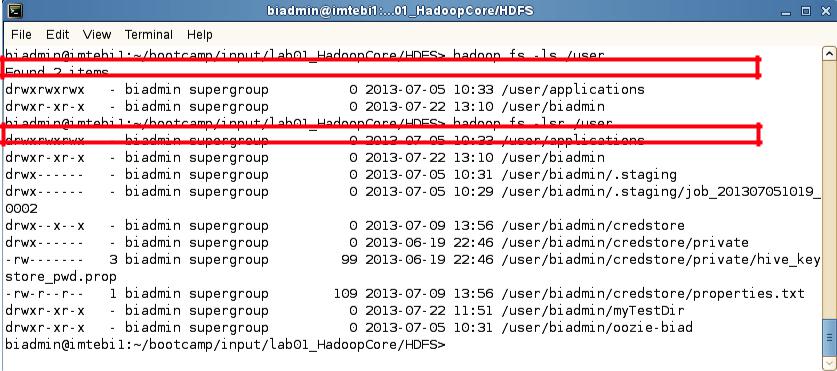
1. For example, to do a recursive listing we’ll use the –lsr command rather than just –ls, like the examples below:



**hadoop fs -ls /user**

****

**hadoop fs -lsr /user**

****

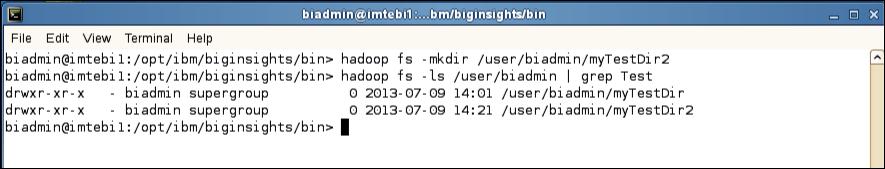
1. You can pipe (using the | character) any HDFS command to be used with the Linux shell. For example, you can easily use *grep* with HDFS by doing the following:



**hadoop fs -mkdir /user/biadmin/myTestDir2**

****

**hadoop fs -ls /user/biadmin | grep Test**

****

As you can see the grep command only returned the lines which had test in them (thus removing the “Found x items” line and the .staging and oozie-biad directories from the listing

1. To move files between your regular Linux filesystem and HDFS you can use the put and get commands. For example, move the text file README to the hadoop filesystem.



**hadoop fs -put /home/biadmin/bootcamp/input/lab01\_HadoopCore/HDFS/README**

****

**README**

****

**hadoop fs -ls /user/biadmin**

****

You should now see a new file called /user/biadmin/README listed as shown above. Note there is a ‘1’ highlighted in the figure. This represents the replication factor. By default, the replication factor in a BigInsights cluster is 3, but since this laboratory environment only has one node, the replication factor is 1.

1. In order to view the contents of this file use the –cat command as follows:



**hadoop fs -cat README**

You should see the output of the README file (that is stored in HDFS). We can also use the linux diff command to see if the file we put on HDFS is actually the same as the original on the local filesystem.

1. Execute the commands below to use the diff command:



**cd /home/biadmin/bootcamp/input/lab01\_HadoopCore/HDFS/**

****

**diff <( hadoop fs -cat README ) README**

Since the diff command produces no output we know that the files are the same (the diff command prints all the lines in the files that differ).

To find the size of files you need to use the –du or –dus commands. Keep in mind that these commands return the file size in bytes.

10. To find the size of the README file use the following command:



**hadoop fs -du README**

****

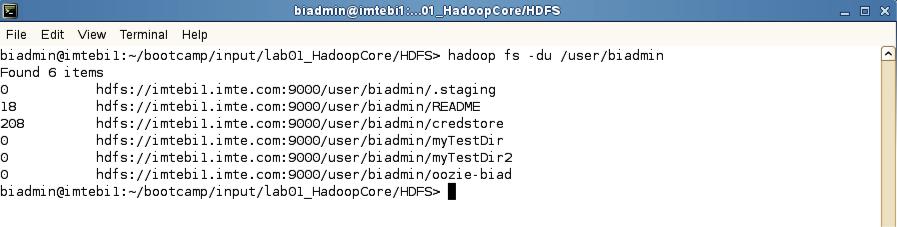
**Inspecting README file size**

In this example, the README file has 18 bytes.

11. To find the size of all files individually in the /user/biadmin directory use the following command:

****

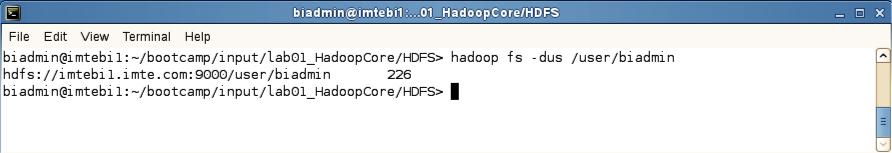
**hadoop fs -du /user/biadmin**

****

12. To find the size of all files in total of the /user/biadmin directory use the following command:



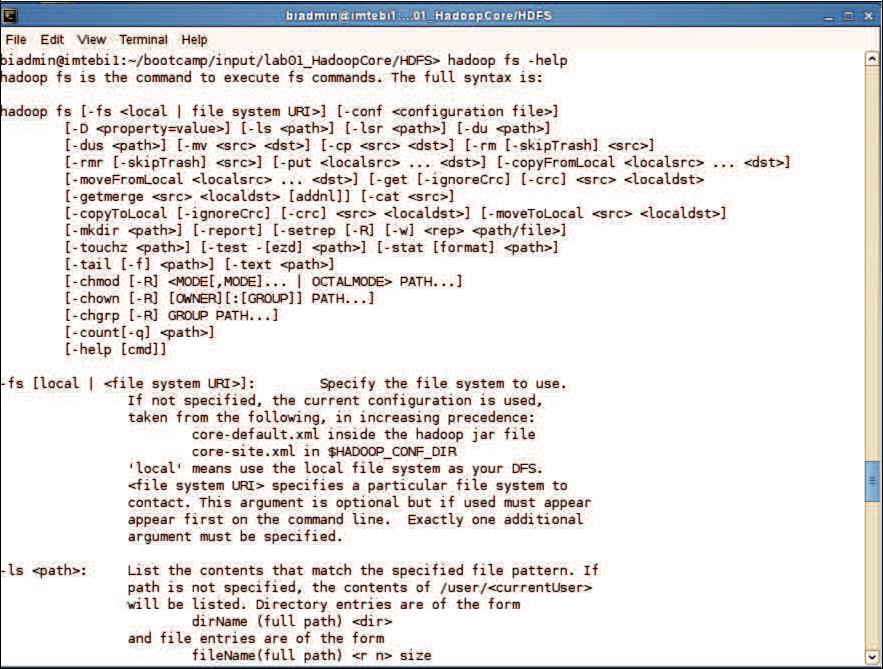
**hadoop fs -dus /user/biadmin**

****

13. If you would like to get more information about hadoop fs commands, invoke –help as follows:



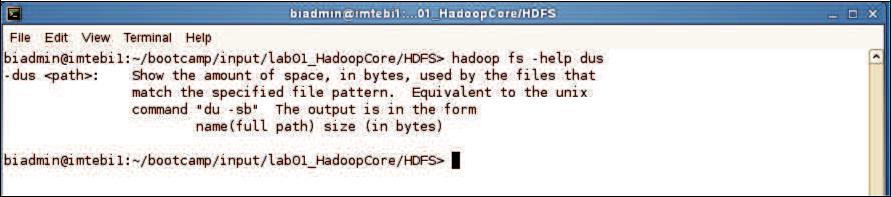
**hadoop fs -help**

****

1. For specific help on a command, add the command name after help. For example, to get help on the dus command you’d do the following:



**hadoop fs -help dus**

****

**Help for specific Haoop commands**

**9.Aim: Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.**

Now that we’ve seen how the FileSystem (fs) shell can be used to execute Hadoop commands to interact with HDFS, the same fs shell can be used to launch MapReduce jobs. In this section, we will walk through the steps required to run a MapReduce program. The source code for a MapReduce program is contained in a compiled .jar file. Hadoop will load the JAR into HDFS and distribute it to the data nodes, where the individual tasks of the MapReduce job will be executed. Hadoop ships with some example MapReduce programs to run. One of these is a distributed WordCount program which reads text files and counts how often words occur.

**Running the WordCount program**

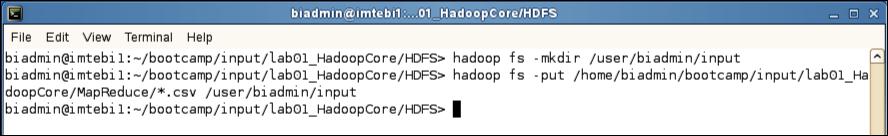
First we need to copy the data files from the local file system to HDFS.

Step 1:Execute the commands below to copy the input files into HDFS.



**hadoop fs -mkdir /user/biadmin/input**

**hadoop fs -put /home/biadmin/bootcamp/input/lab01\_HadoopCore/MapReduce/\*.csv /user/biadmin/input**

****

**Copy input files into HDFS**

Step 2: Review the files have been copied with the following command:



**hadoop fs -ls input**

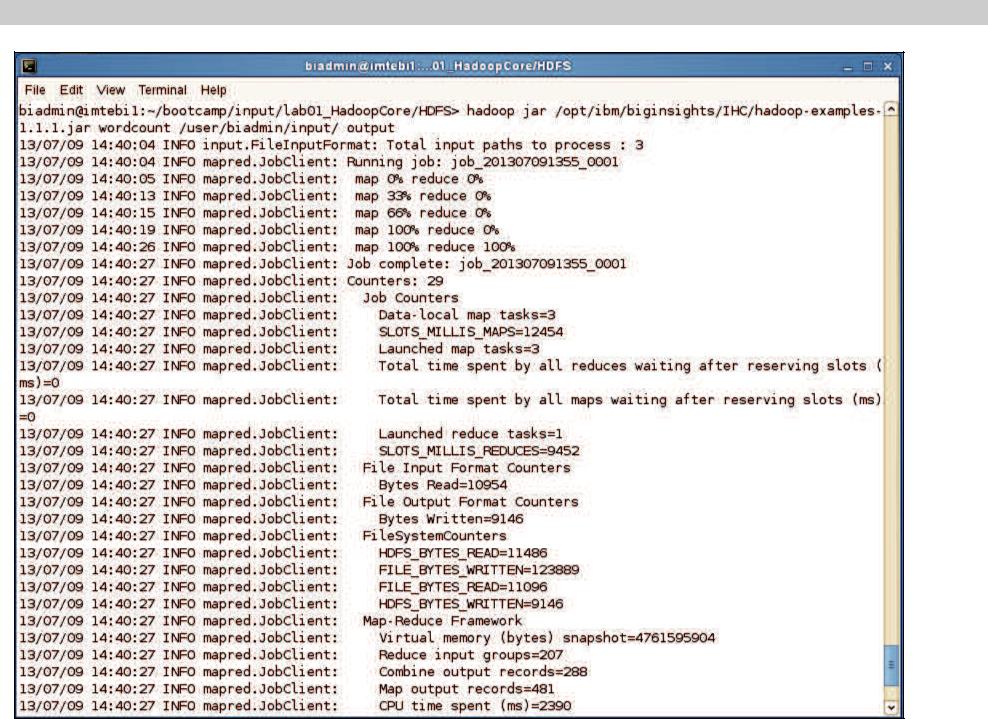
****

**List copied files into HDFS**

Step 3: Now we can run the wordcount job with the command below, where “/user/biadmin/input/” is where the input files are, and “output” is the directory where the output of the job will be stored. The “output” directory will be created automatically when executing the command below.

****

**hadoop jar /opt/ibm/biginsights/IHC/hadoop-examples-1.1.1.jar wordcount /user/biadmin/input/ output**

****

**WordCount MapReduce job running**

Step 4: Now review the output of step 3:

In this case, the output was not split into multiple files.



**hadoop fs -ls output**

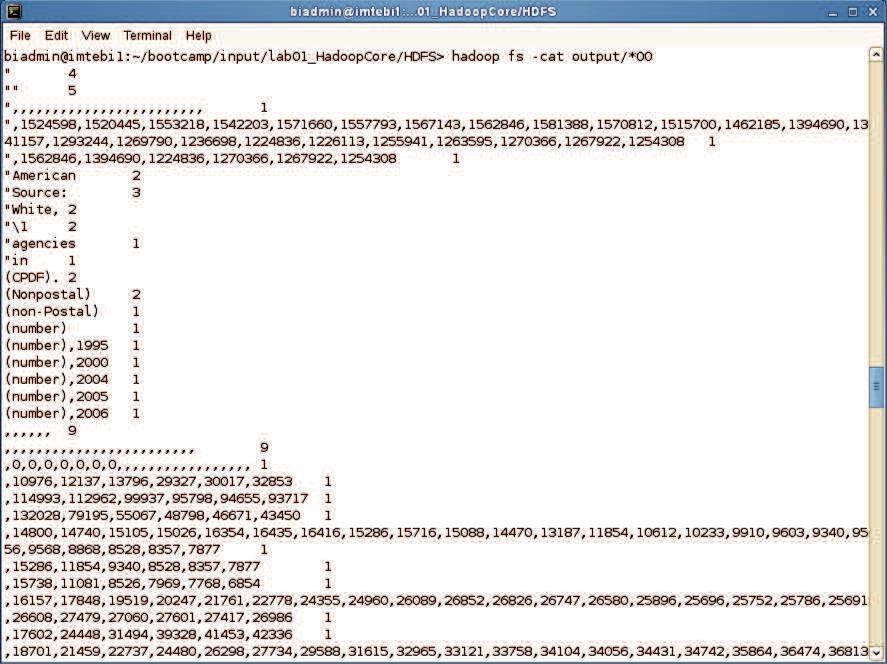
****

**MapReduce result files**

Step 5: To view the contents of the part-r-0000 file issue the command below:



**hadoop fs -cat output/\*00**

****

**MapReduce output**