Bigtop Integration tests starting with the Hadoop Real World Solutions Cookbook

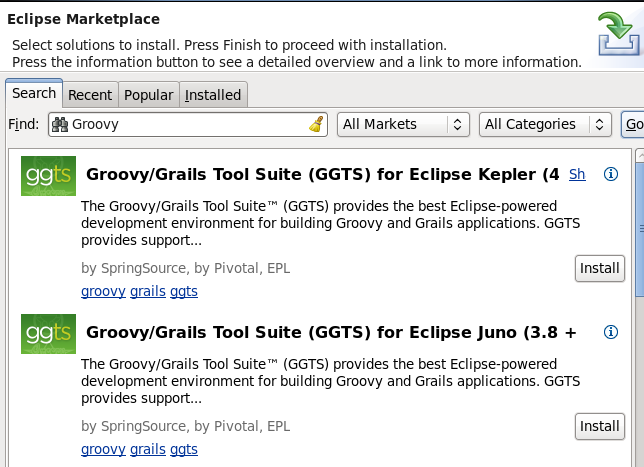
OK this is an experiment. Feel free to follow your own framework for writing integration tests. We covered how to run mvn verify and mvn install in a separate document.

Install bigtop-0.6.0 onto your development machine. Follow the wiki and make sure you can both run map reduce jobs and access hdfs.

If I had to start writing integration tests now my code would be so crappy and I would be too slow. So this exercise is a practice session to both learn how Groovy and Java work and to get some basic debug skills before jumping into more complexity.

**Install the Groovy plugin into Eclipse.**

I used the Eclipse>>Help>>Marketplace option and searched for Groovy. Then I installed the Juno plugin listed in the search result.

****

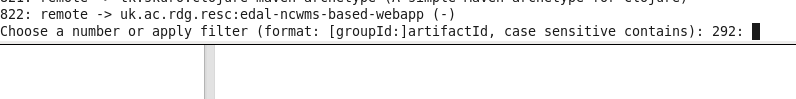
Using Eclipse, create a test groovy project and write a simple test helloworld program to make sure the installation is successful. Use the File>>New Project>>Groovy Project and get something to print out in Eclipse.

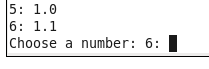
Create a Maven test project using the standard Java archetype.

>mvn archetype:generate

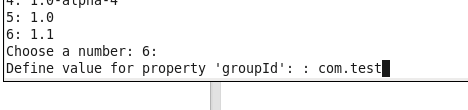
There are questions mvn comes back with, you can answer any way which you are used to. I just hit enter for as many questions as possible.

Hit enter, 292 is close enough for our Groovy/Java project.

Maven asks for a version number. Hit enter. 



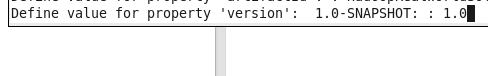
Make up a groupId, I use com.test. You can use something closer to apache like org.apache if you like but we have a long way to go with parent poms and directory structures before we get there.



The artifactId is our folder name and our Eclipse project name. I am choosing HadoopRealWorldSolutions. I should see a folder with this name.



Hit Enter



Maven creates the subdirectory , cd into it and you should see a src directory and pom.xml file.



Your pom file should look something like this:

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.test</groupId>

<artifactId>HadoopRealWorldSolutions</artifactId>

<version>1.0</version>

<packaging>jar</packaging>

<name>HadoopRealWorldSolutions</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

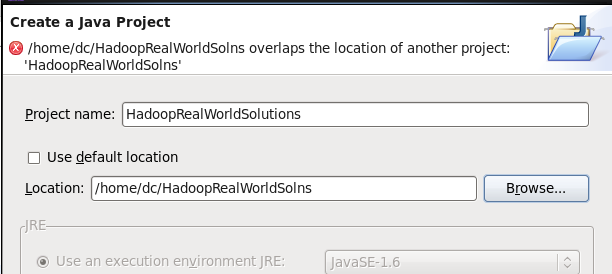
</dependencies>

</project>

Lets turn this subdirectory into an Eclipse project. The command below creates .classpath and .project direectories which you can't see unless you do a >ls -al

>mvn eclipse:eclipse

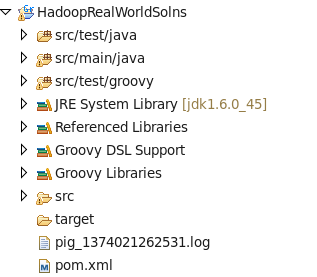
Start Eclipse and create a new Java project. Enter the same name as you used for the artifactId in the Maven project:



Uncheck the Use default location and type in the path of the Location in the text box below. I have a red circle and x above because I already have this project. You shouldn't see this error message.

Once the Eclipse project is created you should see a src/test/java and src/main/java directory.

Right click on your project file and create a new source directory, I used /src/test/groovy. Then create a new package, I used com.test. After this you should see both a src/main/java, src/test/java and /src/test/groovy test directory.



Using the book as a reference I am going to create 2 types of test files, ones using Java and ones using Groovy. This is **DIFFERENT** than the Bigtop convemtion where all the tests are under main/groovy as seen under the /bigtop-0.6.0/bigtop-tests/test-artifacts/hbase/src/main/groovy/org/apache/bigtop/itest/hbase/smoke directory where both Java and Groovy files are in the same directory.

This handout uses a different structure because it was easier to debug using the Groovy plugin and Java IDE settings at the same time. Feel free to pursue your own structure. For example the Java dependencies don't show up when using the Groovy plugin.

First create a test Java program which can access the HDFS file system. This is equivalent to doing a >hadoop fs -ls -R /user/$USER but you should be able to do this in code.

Here is my Java program for doing a ls:

**package** com.test;

**import** org.apache.hadoop.conf.Configuration;

**import** org.apache.hadoop.fs.FileSystem;

**import** org.apache.hadoop.fs.LocatedFileStatus;

**import** org.apache.hadoop.fs.Path;

**import** org.apache.hadoop.fs.RemoteIterator;

**public** **class** AccessHDFSFromJava {

**public** **static** **void** main(String []args){

**try**{

Configuration conf = **new** Configuration();

conf.addResource(**new** Path("/etc/hadoop/conf/core-site.xml"));

FileSystem fs = FileSystem.*get*(conf);

RemoteIterator<LocatedFileStatus> it = fs.listFiles(**new** Path("/user/dc"), **false**);

**while**(it.hasNext()){

LocatedFileStatus locatedFile = it.next();

System.*out*.println(locatedFile.getPath());

}

}**catch**(Exception e){

e.printStackTrace();

}

}

}

Verify you can access HDFS using >hadoop fs -ls -R /user/$USER showing HDFS is up and running before running your Java program. The Java program doesn't show the timeout messages which you get from the command line.

Run as Java Application:

SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.

hdfs://storm1:8020/user/dc/apache\_clf.txt

hdfs://storm1:8020/user/dc/apache\_nobots\_tsv.txt

hdfs://storm1:8020/user/dc/data

Your output will be different if you have nothing in your HDFS file system.

Feel free to create a file and import it into HDFS using the -copyFromLocal flag;

create tmp.txt, add some random text to it.

>hadoop fs -copyFromLocal tmp.txt /user/$USER/

Test to see if your Groovy installation works inside your project. I created a Groovy class and a simple print statement:

**package** com.test

**class** TestGroovy {

**public** **static** **void** *main*(String []args){

*println*("I hate groovy cause it confuses me");

}

}

You can run this as a Groovy Script or Java Application:

I am going to put the cookbook recipes into Groovy code as practice for developing integration flows later.

Pig: Install pig using

>sudo yum install pig

Create a new Groovy class using New>>Other>>Groovy Class

I call mine TestPig

**package** com.test

**class** TestPig {

**static** *main*(args) {

}

}

Groovy is faster to work in for many Bigtop test cases. We saw how much code it took us to access hdfs, we can do the same thing in grovy using:

*println* "hadoop fs -ls -R /".*execute*().*text*

Add this to the above and run.

**package** com.test

**class** TestPig {

**static** *main*(args) {

*println* "hadoop fs -ls -R /".*execute*().*text*

}

}

Here is my output in Eclipse:

drwxrwxrwt - hdfs supergroup 0 2013-07-18 02:13 /tmp

drwxr-xr-x - hdfs supergroup 0 2013-07-16 14:32 /tmp/hadoop-yarn

drwxrwxrwt - mapred mapred 0 2013-07-16 16:43 /tmp/hadoop-yarn/staging

drwx------ - dc mapred 0 2013-07-16 16:43 /tmp/hadoop-yarn/staging/dc

drwx------ - dc mapred 0 2013-07-18 02:13 /tmp/hadoop-yarn/staging/dc/.staging

drwx------ - dc mapred 0 2013-07-16 16:52 /tmp/hadoop-yarn/staging/dc/.staging/job\_1373922240656\_0005

-rw-r--r-- 1 dc mapred 7 2013-07-16 16:52 /tmp/hadoop-yarn/staging/dc/.staging/job\_1373922240656\_0005/appTokens

-rw-r--r-- 10 dc mapred 6892542 2013-07-16 16:52 /tmp/hadoop-yarn/staging/dc/.staging/job\_1373922240656\_0005/job.jar

-rw-r--r-- 10 dc mapred 357 2013-07-16 16:52 /tmp/hadoop-yarn/staging/dc/.staging/job\_1373922240656\_0005/job.split

-rw-r--r-- 1 dc mapred 35 2013-07-16 16:52 /tmp/hadoop-yarn/staging/dc/.staging/job\_1373922240656\_0005/job.splitmetainfo

-rw-r--r-- 1 dc mapred 129004 2013-07-16 16:52 /tmp/hadoop-yarn/staging/dc/.staging/job\_1373922240656\_0005/job.xml

drwxr-xr-x - mapred mapred 0 2013-07-17 20:09 /tmp/hadoop-yarn/staging/history

drwxrwx--- - mapred mapred 0 2013-07-17 20:11 /tmp/hadoop-yarn/staging/history/done

drwxrwxrwt - mapred mapred 0 2013-07-16 16:56 /tmp/hadoop-yarn/staging/history/done\_intermediate

drwxrwx--- - dc mapred 0 2013-07-18 02:15 /tmp/hadoop-yarn/staging/history/done\_intermediate/dc

drwxr-xr-x - dc supergroup 0 2013-07-16 17:02 /tmp/temp-1386153182

drwxr-xr-x - dc supergroup 0 2013-07-16 17:37 /tmp/temp-1681962723

drwxr-xr-x - dc supergroup 0 2013-07-16 17:37 /tmp/temp-1681962723/tmp-2135663820

-rw-r--r-- 1 dc supergroup 0 2013-07-16 17:37 /tmp/temp-1681962723/tmp-2135663820/\_SUCCESS

-rw-r--r-- 1 dc supergroup 84 2013-07-16 17:37 /tmp/temp-1681962723/tmp-2135663820/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-16 16:59 /tmp/temp-1748535856

drwxr-xr-x - dc supergroup 0 2013-07-16 16:59 /tmp/temp-1748535856/tmp-980356956

-rw-r--r-- 1 dc supergroup 0 2013-07-16 16:59 /tmp/temp-1748535856/tmp-980356956/\_SUCCESS

-rw-r--r-- 1 dc supergroup 61 2013-07-16 16:59 /tmp/temp-1748535856/tmp-980356956/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:05 /tmp/temp-279396683

drwxr-xr-x - dc supergroup 0 2013-07-17 20:05 /tmp/temp-279396683/tmp-1094845523

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:05 /tmp/temp-279396683/tmp-1094845523/\_SUCCESS

-rw-r--r-- 1 dc supergroup 59017 2013-07-17 20:05 /tmp/temp-279396683/tmp-1094845523/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:04 /tmp/temp-279396683/tmp-263261164

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:04 /tmp/temp-279396683/tmp-263261164/\_SUCCESS

-rw-r--r-- 1 dc supergroup 59017 2013-07-17 20:04 /tmp/temp-279396683/tmp-263261164/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:41 /tmp/temp-395544683

drwxr-xr-x - dc supergroup 0 2013-07-17 20:11 /tmp/temp-395544683/tmp-1363644292

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:11 /tmp/temp-395544683/tmp-1363644292/\_SUCCESS

-rw-r--r-- 1 dc supergroup 59017 2013-07-17 20:11 /tmp/temp-395544683/tmp-1363644292/part-r-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:10 /tmp/temp-395544683/tmp-1405803927

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:10 /tmp/temp-395544683/tmp-1405803927/\_SUCCESS

-rw-r--r-- 1 dc supergroup 59017 2013-07-17 20:10 /tmp/temp-395544683/tmp-1405803927/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:41 /tmp/temp-395544683/tmp-1868552207

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:41 /tmp/temp-395544683/tmp-1868552207/\_SUCCESS

-rw-r--r-- 1 dc supergroup 59017 2013-07-17 20:41 /tmp/temp-395544683/tmp-1868552207/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:10 /tmp/temp-395544683/tmp2011917606

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:10 /tmp/temp-395544683/tmp2011917606/\_SUCCESS

-rw-r--r-- 1 dc supergroup 57 2013-07-17 20:10 /tmp/temp-395544683/tmp2011917606/part-r-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:41 /tmp/temp-395544683/tmp839255380

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:41 /tmp/temp-395544683/tmp839255380/\_SUCCESS

-rw-r--r-- 1 dc supergroup 57 2013-07-17 20:41 /tmp/temp-395544683/tmp839255380/part-r-00000

drwxr-xr-x - dc supergroup 0 2013-07-16 17:18 /tmp/temp-644938549

drwxr-xr-x - dc supergroup 0 2013-07-16 17:18 /tmp/temp-644938549/tmp-787223133

-rw-r--r-- 1 dc supergroup 0 2013-07-16 17:18 /tmp/temp-644938549/tmp-787223133/\_SUCCESS

-rw-r--r-- 1 dc supergroup 33 2013-07-16 17:18 /tmp/temp-644938549/tmp-787223133/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:07 /tmp/temp-735782161

drwxr-xr-x - dc supergroup 0 2013-07-17 20:06 /tmp/temp-735782161/tmp-1669051001

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:06 /tmp/temp-735782161/tmp-1669051001/\_SUCCESS

-rw-r--r-- 1 dc supergroup 59017 2013-07-17 20:06 /tmp/temp-735782161/tmp-1669051001/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:06 /tmp/temp-735782161/tmp-944436899

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:06 /tmp/temp-735782161/tmp-944436899/\_SUCCESS

-rw-r--r-- 1 dc supergroup 57 2013-07-17 20:06 /tmp/temp-735782161/tmp-944436899/part-r-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:07 /tmp/temp-735782161/tmp1512402075

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:07 /tmp/temp-735782161/tmp1512402075/\_SUCCESS

-rw-r--r-- 1 dc supergroup 59017 2013-07-17 20:07 /tmp/temp-735782161/tmp1512402075/part-r-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:03 /tmp/temp1273863272

drwxr-xr-x - dc supergroup 0 2013-07-17 20:03 /tmp/temp1273863272/tmp-1165931250

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:03 /tmp/temp1273863272/tmp-1165931250/\_SUCCESS

-rw-r--r-- 1 dc supergroup 59017 2013-07-17 20:03 /tmp/temp1273863272/tmp-1165931250/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:01 /tmp/temp1273863272/tmp-2124012818

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:01 /tmp/temp1273863272/tmp-2124012818/\_SUCCESS

-rw-r--r-- 1 dc supergroup 25027 2013-07-17 20:01 /tmp/temp1273863272/tmp-2124012818/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:00 /tmp/temp1918843880

drwxr-xr-x - dc supergroup 0 2013-07-17 19:59 /tmp/temp1918843880/tmp-1614558338

-rw-r--r-- 1 dc supergroup 0 2013-07-17 19:59 /tmp/temp1918843880/tmp-1614558338/\_SUCCESS

-rw-r--r-- 1 dc supergroup 23482 2013-07-17 19:59 /tmp/temp1918843880/tmp-1614558338/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 20:00 /tmp/temp1918843880/tmp-593277599

-rw-r--r-- 1 dc supergroup 0 2013-07-17 20:00 /tmp/temp1918843880/tmp-593277599/\_SUCCESS

-rw-r--r-- 1 dc supergroup 25027 2013-07-17 20:00 /tmp/temp1918843880/tmp-593277599/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 19:58 /tmp/temp1918843880/tmp1947095926

-rw-r--r-- 1 dc supergroup 0 2013-07-17 19:58 /tmp/temp1918843880/tmp1947095926/\_SUCCESS

-rw-r--r-- 1 dc supergroup 10168 2013-07-17 19:58 /tmp/temp1918843880/tmp1947095926/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 19:57 /tmp/temp1918843880/tmp223819854

-rw-r--r-- 1 dc supergroup 0 2013-07-17 19:57 /tmp/temp1918843880/tmp223819854/\_SUCCESS

-rw-r--r-- 1 dc supergroup 62622 2013-07-17 19:57 /tmp/temp1918843880/tmp223819854/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-17 19:58 /tmp/temp1918843880/tmp636385655

-rw-r--r-- 1 dc supergroup 0 2013-07-17 19:58 /tmp/temp1918843880/tmp636385655/\_SUCCESS

-rw-r--r-- 1 dc supergroup 21937 2013-07-17 19:58 /tmp/temp1918843880/tmp636385655/part-m-00000

drwxr-xr-x - dc supergroup 0 2013-07-16 16:46 /tmp/temp242143763

drwxr-xr-x - dc supergroup 0 2013-07-16 16:44 /tmp/temp247206300

drwxr-xr-x - dc supergroup 0 2013-07-16 16:43 /tmp/temp816464825

drwxr-xr-x - hdfs supergroup 0 2013-07-16 14:31 /user

drwxrwx--- - dc dc 0 2013-07-18 02:12 /user/dc

-rw-r--r-- 1 dc dc 172032 2013-07-17 15:08 /user/dc/apache\_clf.txt

-rw-r--r-- 1 dc dc 57472 2013-07-17 16:34 /user/dc/apache\_nobots\_tsv.txt

-rw-r--r-- 1 dc dc 17 2013-07-16 16:52 /user/dc/data

drwxr-xr-x - dc dc 0 2013-07-17 20:49 /user/dc/orderednw

-rw-r--r-- 1 dc dc 0 2013-07-17 20:49 /user/dc/orderednw/\_SUCCESS

-rw-r--r-- 1 dc dc 57472 2013-07-17 20:49 /user/dc/orderednw/part-r-00000

drwxr-xr-x - dc dc 0 2013-07-17 15:24 /user/dc/out1.txt

-rw-r--r-- 1 dc dc 0 2013-07-17 15:24 /user/dc/out1.txt/\_SUCCESS

-rw-r--r-- 1 dc dc 109842 2013-07-17 15:24 /user/dc/out1.txt/part-m-00000

drwxr-xr-x - dc dc 0 2013-07-16 17:34 /user/dc/pigdata

-rw-r--r-- 1 dc dc 9 2013-07-16 17:01 /user/dc/pigdata/data

-rw-r--r-- 1 dc dc 24 2013-07-16 17:34 /user/dc/pigdata/groovydata

drwxr-xr-x - dc dc 0 2013-07-18 02:12 /user/dc/sessions

-rw-r--r-- 1 dc dc 0 2013-07-18 02:12 /user/dc/sessions/\_SUCCESS

-rw-r--r-- 1 dc dc 57023 2013-07-18 02:12 /user/dc/sessions/part-r-00000

drwxr-xr-x - dc dc 0 2013-07-17 20:42 /user/dc/testme

-rw-r--r-- 1 dc dc 0 2013-07-17 20:42 /user/dc/testme/\_SUCCESS

-rw-r--r-- 1 dc dc 57472 2013-07-17 20:42 /user/dc/testme/part-r-00000

drwxrwx--- - mapred mapred 0 2013-07-16 14:31 /user/history

drwxr-xr-x - hdfs supergroup 0 2013-07-16 14:31 /var

drwxr-xr-x - hdfs supergroup 0 2013-07-16 14:31 /var/log

drwxr-xr-x - yarn mapred 0 2013-07-16 16:55 /var/log/hadoop-yarn

drwxrwxrwt - yarn mapred 0 2013-07-16 16:56 /var/log/hadoop-yarn/apps

drwxrwx--- - dc mapred 0 2013-07-16 16:56 /var/log/hadoop-yarn/apps/dc

drwxrwx--- - dc mapred 0 2013-07-18 02:12 /var/log/hadoop-yarn/apps/dc/logs

drwxrwx--- - dc mapred 0 2013-07-16 16:56 /var/log/hadoop-yarn/apps/dc/logs/application\_1374018911887\_0001

-rw-r----- 1 dc mapred 162686 2013-07-16 16:56 /var/log/hadoop-yarn/apps/dc/logs/application\_1374018911887\_0001/localhost.localdomain\_56868

drwxrwx--- - dc mapred 0 2013-07-16 17:00 /var/log/hadoop-yarn/apps/dc/logs/application\_1374018911887\_0002

-rw-r----- 1 dc mapred 61015 2013-07-16 17:00 /var/log/hadoop-yarn/apps/dc/logs/application\_1374018911887\_0002/localhost.localdomain\_56868

drwxrwx--- - dc mapred 0 2013-07-16 17:18 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0001

-rw-r----- 1 dc mapred 162648 2013-07-16 17:18 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0001/storm1\_57240

drwxrwx--- - dc mapred 0 2013-07-16 17:18 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0002

-rw-r----- 1 dc mapred 61342 2013-07-16 17:18 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0002/storm1\_57240

drwxrwx--- - dc mapred 0 2013-07-16 17:36 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0003

-rw-r----- 1 dc mapred 61332 2013-07-16 17:36 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0003/storm1\_57240

drwxrwx--- - dc mapred 0 2013-07-16 17:37 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0004

-rw-r----- 1 dc mapred 60939 2013-07-16 17:37 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0004/storm1\_57240

drwxrwx--- - dc mapred 0 2013-07-16 17:42 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0005

-rw-r----- 1 dc mapred 60962 2013-07-16 17:42 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0005/storm1\_57240

drwxrwx--- - dc mapred 0 2013-07-16 17:42 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0006

-rw-r----- 1 dc mapred 61331 2013-07-16 17:42 /var/log/hadoop-yarn/apps/dc/logs/application\_1374020165932\_0006/storm1\_57240

drwxrwx--- - dc mapred 0 2013-07-17 12:17 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0001

-rw-r----- 1 dc mapred 166389 2013-07-17 12:17 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0001/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 15:09 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0002

-rw-r----- 1 dc mapred 87623 2013-07-17 15:09 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0002/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 15:13 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0003

-rw-r----- 1 dc mapred 87621 2013-07-17 15:13 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0003/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 15:14 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0004

-rw-r----- 1 dc mapred 86883 2013-07-17 15:14 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0004/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 15:16 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0005

-rw-r----- 1 dc mapred 36999 2013-07-17 15:16 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0005/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 15:25 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0006

-rw-r----- 1 dc mapred 36999 2013-07-17 15:25 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0006/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 19:57 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0007

-rw-r----- 1 dc mapred 61538 2013-07-17 19:57 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0007/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 19:58 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0008

-rw-r----- 1 dc mapred 61552 2013-07-17 19:58 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0008/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 19:58 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0009

-rw-r----- 1 dc mapred 61552 2013-07-17 19:58 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0009/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 20:00 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0010

-rw-r----- 1 dc mapred 61553 2013-07-17 20:00 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0010/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 20:00 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0011

-rw-r----- 1 dc mapred 60762 2013-07-17 20:00 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0011/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 20:01 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0012

-rw-r----- 1 dc mapred 61553 2013-07-17 20:01 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0012/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 20:03 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0013

-rw-r----- 1 dc mapred 60394 2013-07-17 20:03 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0013/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 20:04 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0014

-rw-r----- 1 dc mapred 61552 2013-07-17 20:04 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0014/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 20:05 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0015

-rw-r----- 1 dc mapred 60394 2013-07-17 20:05 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0015/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 20:06 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0016

-rw-r----- 1 dc mapred 61158 2013-07-17 20:06 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0016/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 20:07 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0017

-rw-r----- 1 dc mapred 103277 2013-07-17 20:07 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0017/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 20:07 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0018

-rw-r----- 1 dc mapred 105910 2013-07-17 20:07 /var/log/hadoop-yarn/apps/dc/logs/application\_1374088559842\_0018/localhost.localdomain\_52022

drwxrwx--- - dc mapred 0 2013-07-17 20:10 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0001

-rw-r----- 1 dc mapred 60612 2013-07-17 20:10 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0001/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:10 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0002

-rw-r----- 1 dc mapred 103026 2013-07-17 20:10 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0002/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:11 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0003

-rw-r----- 1 dc mapred 103627 2013-07-17 20:11 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0003/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:41 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0004

-rw-r----- 1 dc mapred 60978 2013-07-17 20:41 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0004/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:41 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0005

-rw-r----- 1 dc mapred 102652 2013-07-17 20:41 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0005/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:42 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0006

-rw-r----- 1 dc mapred 104336 2013-07-17 20:42 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0006/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:45 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0007

-rw-r----- 1 dc mapred 60955 2013-07-17 20:45 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0007/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:45 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0008

-rw-r----- 1 dc mapred 102631 2013-07-17 20:45 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0008/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:46 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0009

-rw-r----- 1 dc mapred 104697 2013-07-17 20:46 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0009/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:48 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0010

-rw-r----- 1 dc mapred 60588 2013-07-17 20:48 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0010/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:49 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0011

-rw-r----- 1 dc mapred 102632 2013-07-17 20:49 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0011/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 20:49 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0012

-rw-r----- 1 dc mapred 103924 2013-07-17 20:49 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0012/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-17 22:09 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0013

-rw-r----- 1 dc mapred 35812 2013-07-17 22:09 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0013/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-18 02:01 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0014

-rw-r----- 1 dc mapred 35813 2013-07-18 02:01 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0014/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-18 02:05 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0015

-rw-r----- 1 dc mapred 61399 2013-07-18 02:05 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0015/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-18 02:08 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0016

-rw-r----- 1 dc mapred 35824 2013-07-18 02:08 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0016/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-18 02:08 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0017

-rw-r----- 1 dc mapred 105491 2013-07-18 02:08 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0017/storm1\_51459

drwxrwx--- - dc mapred 0 2013-07-18 02:13 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0018

-rw-r----- 1 dc mapred 105479 2013-07-18 02:13 /var/log/hadoop-yarn/apps/dc/logs/application\_1374116915752\_0018/storm1\_51459

This is a lot easier than using the Java API. Roman is a genius. Your output will be different.

**Real World Solutions Chapter 3:**

Lets work through the Ch3 examples

Under /src/main/java or /src/test/java create a new Java class. I used a different name for my class.

Cut and paste the Java code from the book.

There is an error in the book code example.

package com.test;

import java.io.IOException;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import org.apache.hadoop.conf.\*;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

//import org.apache.hadoop.mapred.jobcontrol.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

import org.apache.hadoop.util.Tool;

import org.apache.hadoop.util.ToolRunner;

public class TestPigTSV extends Configured implements Tool{

//book is wrong, need static else compiler creates a static ctor from the parent class

//and there is no default ctor w/no args b/c the parent doesnt have one

public static class CLFMapper extends Mapper<Object, Text, Text, Text>{

private SimpleDateFormat dateFormatter =

new SimpleDateFormat("dd/MMM/yyyy:HH:mm:ss Z");

private Pattern p =

Pattern.compile("^([\\d.]+) (\\S+) (\\S+)"

+ " \\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(\\w+) (.+?) (.+?)\" "

+ "(\\d+) (\\d+) \"([^\"]+|(.+?))\" \"([^\"]+|(.+?))\"",

Pattern.DOTALL);

private Text outputKey = new Text();

private Text outputValue = new Text();

@Override

protected void map(Object key, Text value, Context

context) throws IOException, InterruptedException {

String entry = value.toString();

Matcher m = p.matcher(entry);

if (!m.matches()) {

return;

}

Date date = null;

try {

date = dateFormatter.parse(m.group(4));

} catch (ParseException ex) {

return;

}

outputKey.set(m.group(1)); //ip

StringBuilder b = new StringBuilder();

b.append(date.getTime()); //timestamp

b.append('\t');

b.append(m.group(6)); //page

b.append('\t');

b.append(m.group(8)); //http status

b.append('\t');

b.append(m.group(9)); //bytes

b.append('\t');

b.append(m.group(12)); //useragent

outputValue.set(b.toString());

context.write(outputKey, outputValue);

}

}

public int run(String[] args) throws Exception {

Path inputPath = new Path(args[0]);

Path outputPath = new Path(args[1]);

Configuration conf = getConf();

Job weblogJob = new Job(conf);

weblogJob.setJobName("Weblog Transformer");

weblogJob.setJarByClass(getClass());

weblogJob.setNumReduceTasks(0);

weblogJob.setMapperClass(CLFMapper.class);

weblogJob.setMapOutputKeyClass(Text.class);

weblogJob.setMapOutputValueClass(Text.class);

weblogJob.setOutputKeyClass(Text.class);

weblogJob.setOutputValueClass(Text.class);

weblogJob.setInputFormatClass(TextInputFormat.class);

weblogJob.setOutputFormatClass(TextOutputFormat.class);

FileInputFormat.setInputPaths(weblogJob, inputPath);

FileOutputFormat.setOutputPath(weblogJob, outputPath);

if(weblogJob.waitForCompletion(true)) {

return 0;

}

return 1;

}

public static void main( String[] args ) throws Exception {

int returnCode = ToolRunner.run(new TestPigTSV(), args);

System.exit(returnCode);

}

}

Right click on the HadoopRealWorldSolns project and Export>>Jar. This creates a Jar named HadoopRealWorldSolns.jar with the class com.test.TestPigTSV in it.

Go back to your groovy test directory and add groovy code to run the class. You can do this in the command line like the book says but the objective is to get more experience with Groovy and how it works in the Hadoop environment as practice.

Make sure to change the paths in the book examples to your development machine path, I had to change /user/hadoop to /user/dc

Before writing code make sure your environment is setup correctly. Run the

hadoop jar /home/dc/workspace/HadoopRealWorldSolns.jar com.test.TestPigTSV /user/dc/apache\_clf.txt /user/dc/apache\_clf\_tsv

from the command line first and make sure you get correct output.

[dc@localhost ~]$ hadoop jar /home/dc/workspace/HadoopRealWorldSolns.jar com.test.TestPigTSV '/user/dc/apache\_clf.txt' '/user/dc/apacheout.tsv'

13/07/18 16:18:05 INFO service.AbstractService: Service:org.apache.hadoop.yarn.client.YarnClientImpl is inited.

13/07/18 16:18:05 INFO service.AbstractService: Service:org.apache.hadoop.yarn.client.YarnClientImpl is started.

13/07/18 16:18:06 INFO input.FileInputFormat: Total input paths to process : 1

13/07/18 16:18:06 INFO mapreduce.JobSubmitter: number of splits:1

13/07/18 16:18:06 WARN conf.Configuration: mapred.jar is deprecated. Instead, use mapreduce.job.jar

13/07/18 16:18:06 WARN conf.Configuration: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces

13/07/18 16:18:06 WARN conf.Configuration: mapred.output.value.class is deprecated. Instead, use mapreduce.job.output.value.class

13/07/18 16:18:06 WARN conf.Configuration: mapred.mapoutput.value.class is deprecated. Instead, use mapreduce.map.output.value.class

13/07/18 16:18:06 WARN conf.Configuration: mapreduce.map.class is deprecated. Instead, use mapreduce.job.map.class

13/07/18 16:18:06 WARN conf.Configuration: mapred.job.name is deprecated. Instead, use mapreduce.job.name

13/07/18 16:18:06 WARN conf.Configuration: mapreduce.inputformat.class is deprecated. Instead, use mapreduce.job.inputformat.class

13/07/18 16:18:06 WARN conf.Configuration: mapred.input.dir is deprecated. Instead, use mapreduce.input.fileinputformat.inputdir

13/07/18 16:18:06 WARN conf.Configuration: mapred.output.dir is deprecated. Instead, use mapreduce.output.fileoutputformat.outputdir

13/07/18 16:18:06 WARN conf.Configuration: mapreduce.outputformat.class is deprecated. Instead, use mapreduce.job.outputformat.class

13/07/18 16:18:06 WARN conf.Configuration: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps

13/07/18 16:18:06 WARN conf.Configuration: mapred.output.key.class is deprecated. Instead, use mapreduce.job.output.key.class

13/07/18 16:18:06 WARN conf.Configuration: mapred.mapoutput.key.class is deprecated. Instead, use mapreduce.map.output.key.class

13/07/18 16:18:06 WARN conf.Configuration: mapred.working.dir is deprecated. Instead, use mapreduce.job.working.dir

13/07/18 16:18:06 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1374189323400\_0001

13/07/18 16:18:07 INFO client.YarnClientImpl: Submitted application application\_1374189323400\_0001 to ResourceManager at /0.0.0.0:8032

13/07/18 16:18:07 INFO mapreduce.Job: The url to track the job: http://localhost.localdomain:8088/proxy/application\_1374189323400\_0001/

13/07/18 16:18:07 INFO mapreduce.Job: Running job: job\_1374189323400\_0001

13/07/18 16:18:13 INFO mapreduce.Job: Job job\_1374189323400\_0001 running in uber mode : false

13/07/18 16:18:13 INFO mapreduce.Job: map 0% reduce 0%

13/07/18 16:18:17 INFO mapreduce.Job: map 100% reduce 0%

13/07/18 16:18:17 INFO mapreduce.Job: Job job\_1374189323400\_0001 completed successfully

13/07/18 16:18:17 INFO mapreduce.Job: Counters: 27

File System Counters

FILE: Number of bytes read=0

FILE: Number of bytes written=73696

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=172138

HDFS: Number of bytes written=109842

HDFS: Number of read operations=5

HDFS: Number of large read operations=0

HDFS: Number of write operations=2

Job Counters

Launched map tasks=1

Data-local map tasks=1

Total time spent by all maps in occupied slots (ms)=3180

Total time spent by all reduces in occupied slots (ms)=0

Map-Reduce Framework

Map input records=943

Map output records=942

Input split bytes=106

Spilled Records=0

Failed Shuffles=0

Merged Map outputs=0

GC time elapsed (ms)=15

CPU time spent (ms)=960

Physical memory (bytes) snapshot=123514880

Virtual memory (bytes) snapshot=626745344

Total committed heap usage (bytes)=120520704

File Input Format Counters

Bytes Read=172032

File Output Format Counters

OK once the environment is setup lets try adding the same command line we just tested to our Groovy code. We should only need one line of Groovy code if we do things correctly.

**package** com.test

**class** TestPig {

**static** *main*(args) {

*println* "hadoop jar /home/dc/workspace/HadoopRealWorldSolns.jar com.test.TestPigTSV /user/dc/apache\_clf.txt /user/dc/apache\_clf\_tsv".*execute*().*text*

}

}

Verify you get the same result after running this in Eclipse. The command line output doesn't show up but you can delete the /user/$USER/apache\_clf.tsv and verify it was recreated after the Groovy code is run.

[dc@localhost ~]$ hadoop fs -rm -R /user/dc/apache\_clf\_tsv

Deleted /user/dc/apache\_clf\_tsv

Run Eclipse Groovy code.

Verify

[dc@localhost ~]$ hadoop fs -ls -R /user/dc

-rw-r--r-- 1 dc dc 172032 2013-07-17 15:08 /user/dc/apache\_clf.txt

drwxr-xr-x - dc dc 0 2013-07-18 16:26 /user/dc/apache\_clf\_tsv

-rw-r--r-- 1 dc dc 0 2013-07-18 16:26 /user/dc/apache\_clf\_tsv/\_SUCCESS

OK this is sort of a hassle having to mix command line hadoop fs commands with Groovy.

Here is one example of using Groovy code to test if the file is there, if not import it into HDFS.

**package** com.test

**class** TestPig {

**static** *main*(args) {

String result = *println* "hadoop fs -stat /user/dc/apache\_clf.txt }| grep No".*execute*().*text*

**if** (result == **null**){

*println* "hadoop fs -copyFromLocal /home/dc/tmppig/apache.clf.txt /user/$USER".*execute*().*text*

}

*println* "hadoop jar /home/dc/workspace/HadoopRealWorldSolns.jar com.test.TestPigTSV /user/dc/apache\_clf.txt /user/dc/apache\_clf\_tsv".*execute*().*text*

}

}

There are many ways to test if a file exists in HDFS. The above example can also use hadoop fs -test which returns 0/1

Another way to do this is to embed Java code using the HDFS Client API we used earlier.

2) **Using Apache Pig to filter bot traffic from web server logs**

This example creates a simple Pig UDF Java class called IsUseragentBot and creates a pig script to run several pig commands at once. Let's try something different. Since we have a separate Java namespace setup in /src/java/main/ and /src/java/tests/ lets try creating the class in there and importing it into our Groovy test sequence instead of exporting a Jar and running a Java commmand line.

Cut and paste the class into a Java namespace. Normally we would write a unit tests for this but let's ignore this for now since we are provided working code.

There are errors in the code printout in the book. Blacklist in the FileReader should be blacklist.txt

**package** com.test;

**import** java.util.\*;

**import** java.io.\*;

**import** org.apache.hadoop.\*;

**import** org.apache.pig.\*;

**import** org.apache.pig.data.\*;

**public** **class** IsUseragentBot **extends** FilterFunc {

**private** Set<String> blacklist = **null**;

**private** **void** loadBlacklist() **throws** IOException {

blacklist = **new** HashSet<String>();

BufferedReader in = **new** BufferedReader(**new**

FileReader("blacklist.txt"));

String userAgent = **null**;

**while** ((userAgent = in.readLine()) != **null**) {

blacklist.add(userAgent);

}

}

@Override

**public** Boolean exec(Tuple tuple) **throws** IOException {

**if** (blacklist == **null**) {

loadBlacklist();

}

**if** (tuple == **null** || tuple.size() == 0) {

**return** **null**;

}

String ua = (String) tuple.get(0);

**if** (blacklist.contains(ua)) {

**return** **true**;

}

**return** **false**;

}

}

Export this jar, place it in your workspace directory, and run the Groovy program to create the pig script. You can run the generated pigscript file independently of Groovy using >pig pigscript. You should see something like this if your pig script is correct:

2013-07-18 12:10:26,401 [JobControl] INFO org.apache.hadoop.mapreduce.Job - The url to track the job: http://storm1:8088/proxy/application\_1374173954037\_0005/

2013-07-18 12:10:26,464 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 0% complete

2013-07-18 12:10:37,586 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 50% complete

2013-07-18 12:10:42,116 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete

2013-07-18 12:10:42,118 [main] INFO org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:

HadoopVersion PigVersion UserId StartedAt FinishedAt Features

2.0.5-alpha 0.11.1 dc 2013-07-18 12:10:21 2013-07-18 12:10:42 FILTER

Success!

Job Stats (time in seconds):

JobId Maps Reduces MaxMapTime MinMapTIme AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime Alias Feature Outputs

job\_1374173954037\_0005 1 0 3 3 3 3 0 00 0 all\_weblogs,nobots\_weblogs MAP\_ONLY /user/dc/nobots\_weblogs,

Input(s):

Successfully read 515 records (57839 bytes) from: "/user/dc/apache\_nobots\_tsv.txt"

Output(s):

Successfully stored 515 records (57472 bytes) in: "/user/dc/nobots\_weblogs"

Counters:

Total records written : 515

Total bytes written : 57472

Spillable Memory Manager spill count : 0

Total bags proactively spilled: 0

Total records proactively spilled: 0

Job DAG:

job\_1374173954037\_0005

2013-07-18 12:10:42,217 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!

[dc@storm1 tmppig]$

Running the Groovy program:

**package** com.test;

**import** com.test.IsUseragentBot;

**class** TestPig {

**static** *testCh03TSV*(){

String result = *println* "hadoop fs -stat /user/dc/apache\_clf.txt | grep No".*execute*().*text*

**if** (result == **null**){

*println* "hadoop fs -copyFromLocal '/home/dc/tmppig/apache.clf.txt' '/user/dc'".*execute*().*text*

}

"hadoop jar /home/dc/workspace/HadoopRealWorldSolns.jar com.test.TestPigTSV /user/dc/apache\_clf.txt /user/dc/apache\_clf\_tsv".*execute*().*text*

//verify /user/dc/apache\_clf\_tsv exists

String correct = *println* "hadoop fs -stat /user/dc/apache\_clf\_tsv }| grep No".*execute*().*text*

**if**(!correct){

*println*("TSV test correct")

}**else**{

*println*("TSV test failed")

}

}

**static** *testCh03Filter*(){

String isBlackListInHDFS = *println* "hadoop fs -stat /user/dc/blacklist.txt | grep No".*execute*().*text*

**if**(!isBlackListInHDFS){

*println* "hadoop fs -copyFromLocal /home/dc/bigtop-0.6.0/dl/tmppig/blacklist.txt /user/dc".*execute*().*text*

}

File pigScript = **new** File("/home/dc/bigtop-0.6.0/dl/tmppig/pigscript")

**if**(pigScript.exists()){

pigScript.delete()

}

pigScript<<"set mapred.cache.files '/user/dc/blacklist.txt';\n"

pigScript<<"set mapred.create.symlink 'yes';\n"

pigScript<<"register HadoopRealWorldSolns.jar;\n"

pigScript<<"\n"

pigScript<<"\n"

pigScript<<"all\_weblogs = LOAD '/user/dc/apache\_nobots\_tsv.txt' AS (ip: chararray, timestamp:long, page:chararray, http\_status:int, payload\_size:int, useragent:chararray);\n"

pigScript<<"nobots\_weblogs = FILTER all\_weblogs BY NOT com.test.IsUseragentBot(useragent);\n"

pigScript<<"\n"

pigScript<<"\n"

pigScript<<"STORE nobots\_weblogs INTO '/user/dc/nobots\_weblogs';"

*println* "pig /home/dc/bigtop-0.6.0/dl/tmppig/pigscript".*execute*().*text*

//verify

**boolean** success = *println* "hadoop fs -stat /user/dc/nobots\_weblogs | grep No".*execute*().*text*

*println* success

//clean up

**if**(success){

pigScript.delete();

}

}

**static** *main*(args) {

//testCh03TSV()

*testCh03Filter*()

}

}

3) **Using Apache Pig to sort web server log data by timestamp**

Create a Groovy class, this time for variation use the Java Main instead of the Groovy main().

Write to a pigscript and run in Groovy.

package com.test;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.LocatedFileStatus;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.fs.RemoteIterator;

public class TestPigNOBots {

public static void main(String []args){

try{

//import into hdfs if not there

Configuration conf = new Configuration();

conf.addResource(new Path("/etc/conf/hadoop/core-site.xml"));

FileSystem fs = FileSystem.get(conf);

boolean isThere = fs.exists(new Path("/user/dc/apache\_nobots\_tsv.txt"));

if(!isThere){

println "hadoop fs -copyFromLocal /home/dc/bigtop-0.6.0/dl/tmppig/apache\_nobots\_tsv.txt /user/dc".execute().text

}

//create pigscript and run the pig script

File pigScript = new File("/home/dc/bigtop-0.6.0/dl/tmppig/pigscript");

if(pigScript.exists()){

pigScript.delete();

}

//if /user/dc/orderednw exists delete it

if(fs.exists(new Path("/user/dc/orderednw"))){

println "hadoop fs -rm -r /user/dc/orderednw".execute().text

}

pigScript<<"nw = LOAD '/user/dc/apache\_nobots\_tsv.txt' as (ip:chararray, timestamp:long, page:chararray, http\_status:int, payload\_size:int, useragent:chararray);\n"

pigScript<<"orderednw = ORDER nw BY timestamp;\n"

pigScript<<"STORE orderednw INTO '/user/dc/orderednw';\n";

println "pig /home/dc/bigtop-0.6.0/dl/tmppig/pigscript".execute().text

}catch(Exception e){

e.printStackTrace();

}

}

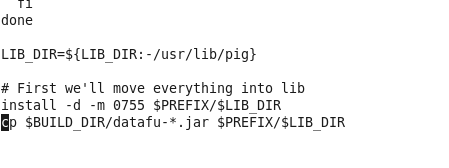
}

**4) Using Apache Pig to sessionize web server log data**

**Digression into Pig UDFs.**

Bigtop includes Pig UDFs open sourced by LinkedIn known as DataFu. You cannot install datafu using sudo yum install datafu. How do you install it? Where is datafu located? It isn't under the standard location /usr/lib/datafu.

The location for datafu is in same subdirectory do-component-build is in.



Datafu is under the /usr/lib/pig directory, copy the jar to your local directory so the pig script can find it or run the pigsript from /usr/lib/pig

The integration tests for datafu are under bigtop-tests/test-artifacts/datafu

Let's modify the book example to use the datafu Sessionize UDF instead of what is in the book.

First let's take a look at a smaller data sample of apache\_nobots\_tsv.txt to see how pig syntax works. Restart the PIG CLI, the ClI timeouts on my machine after inactivity.

Create a smaller version of the apache logs and import it into HDFS:

[dc@localhost ~]$ cat smalllog.txt

221.220.8.0 1341369803000 /index.html 200 140 Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201

166.36.182.90 1341365250000 /index.html 200 140 Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201

221.220.8.0 1341382663000 /about.html 200 140 Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201

166.36.182.90 1341365861000 /index.html 200 140 Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201

213.59.105.58 1341372015000 /about.html 200 140 Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201

221.220.8.0 1341398627000 /cart.html 200 140 Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201

166.36.182.90 1341397968000 /about.html 200 140 Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201

213.59.105.58 1341399229000 /index.html 200 140 Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201

221.220.8.0 1341394762000 /index.html 200 140 Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201

166.36.182.90 1341379096000 /index.html 200 140 Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201

[dc@localhost ~]$

grunt> small = LOAD '/user/dc/smalllog.txt' as (ip:chararray, timestamp:long, page:chararray, http\_status:int, payload\_size:int, useragent:chararray);

grunt> ip\_groups = GROUP small by ip;

grunt> dump ip\_groups

(debug output removed)

(221.220.8.0,{(221.220.8.0,1341394762000,/index.html,200,140,Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201),(221.220.8.0,1341398627000,/cart.html,200,140,Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201),(221.220.8.0,1341382663000,/about.html,200,140,Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201),(221.220.8.0,1341369803000,/index.html,200,140,Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201)})

(166.36.182.90,{(166.36.182.90,1341379096000,/index.html,200,140,Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201),(166.36.182.90,1341397968000,/about.html,200,140,Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201),(166.36.182.90,1341365861000,/index.html,200,140,Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201),(166.36.182.90,1341365250000,/index.html,200,140,Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201)})

(213.59.105.58,{(213.59.105.58,1341399229000,/index.html,200,140,Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201),(213.59.105.58,1341372015000,/about.html,200,140,Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201)})

Notice there are 3 IP addresses: 221.220.8.0, 166.36.182.90, 213.59.105.58 describing bags containing the records from smalllog.txt which have the same IP address.

We have to use a FLATTEN command to change the bags back into tuples to do counting on the bags.

Try this in 2 steps, first loop through the ip addresses printing each one to make sure we have the FOREACH syntax correct.

Testforeach = FOREACH ip\_groups GENERATE ip

Now test the flatten to extract common attributes between all entries in the bag:

[dc@localhost tmppig]$ cat pigscriptsessionize

register datafu-0.0.6.jar;

%declare TIME\_WINDOW 30m

define Sessionize datafu.pig.sessions.Sessionize('$TIME\_WINDOW');

nobots\_weblogs = LOAD '/user/dc/apache\_nobots\_tsv.txt' AS (ip:chararray, timestamp:long, page:chararray, http\_status:int, payload\_size:int, useragent:chararray);

ip\_groups = GROUP nobots\_weblogs by ip;

sessions = FOREACH ip\_groups{

ordered\_by\_timestamp = ORDER nobots\_weblogs by timestamp;

GENERATE FLATTEN(Sessionize(ordered\_by\_timestamp));

}

DUMP sessions;

**Using Python to extend Apache Pig functionality:**

**SKIP**

**Chapter 4:**

If you installed Hue, the Hive metastore\_db may not be accessible to the you with your user permissions.

>chmod -R dc:dc



