coreservlets.com – Hadoop Course

Running MapReduce Jobs

In this exercise, you will have a chance to practice running MapReduce jobs. You will exercise various options on passing properties as well as configuring both client's and tasks' CLASSPATH. If time allows, extra credit section explores utilizing `mapred` tool to get job's status as well as to kill currently running job(s). Extra credit also offers practice in implementing JUnit test for a MapReduce job.

Approx. Time: 60 minutes

Perform

1. Run `Tool` implementation **`mapRed.runningJobs.ExpectProperty`** `which` expects property ***training.prop*** to be set, and if the property is not set it emits `java.lang.IllegalArgumentException`:

```
$ yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.ExpectProperty
Exception in thread "main" java.lang.IllegalArgumentException: Expected property [training.prop] to be provided
        at mapRed.runningJobs.ExpectProperty.run(ExpectProperty.java:14)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70)
        ...
        at org.apache.hadoop.util.RunJar.main(RunJar.java:208)
```

2. Set the property ***training.prop*** by adding extra parameters to the command line
3. Set the property ***training.prop*** by providing external configuration file. You will need to create a brand new configuration file and then specify your file via command line.
4. Run `Tool` implementation **`mapRed.runningJobs.ExpectClassOnClient`** which expects class common.PropPrinter to on CLASSPATH on the client's classpath; this means that the class is used within Tool's implementation; you will get java.lang.ClassNotFoundException:

```
$ yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.ExpectClassOnClient
Exception in thread "main" java.lang.NoClassDefFoundError: common/PropPrinter
      at mapRed.runningJobs.ExpectClass.run(ExpectClass.java:14)
      ...
      at org.apache.hadoop.util.RunJar.main(RunJar.java:208)
Caused by: java.lang.ClassNotFoundException: common.PropPrinter
      at java.net.URLClassLoader$1.run(URLClassLoader.java:202)
      at java.security.AccessController.doPrivileged(Native Method)
      at java.net.URLClassLoader.findClass(URLClassLoader.java:190)
      at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
      at java.lang.ClassLoader.loadClass(ClassLoader.java:247)
      ... 9 more
```

5. The required class, **`common.PropPrinter`**, can be found in $PLAY_AREA/HadoopSample.jar; add the required jar to the CLASSPATH to run the tool without an exception:

```
yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.ExpectClassOnClient
```

6. Run `Tool` implementation **`mapRed.runningJobs.ExpectClassOnTask`** which expects class common.PropPrinter to be on the Mapper Task's CLASSPATH; you will get java.lang.ClassNotFoundException:

```
$ yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.ExpectClassOnTask
/training/data/hamlet.txt /training/playArea/ExpectClassOnTask
...
2012-09-17 14:05:38,562 INFO  mapreduce.Job (Job.java:monitorAndPrintJob(1275)) - Running job:
job_1347904219060_0005
...
2012-09-17 14:05:56,215 INFO  mapreduce.Job (Job.java:printTaskEvents(1391)) - Task Id :
attempt_1347904219060_0005_m_000000_2, Status : FAILED
Error: java.lang.ClassNotFoundException: common.PropPrinter
        at java.net.URLClassLoader$1.run(URLClassLoader.java:202)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(URLClassLoader.java:190)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:301)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:247)
        at
mapRed.runningJobs.ExpectClassOnTask$ExpectClassOnTaskMapper.map(ExpectClassOnTask.java:40)
        at
mapRed.runningJobs.ExpectClassOnTask$ExpectClassOnTaskMapper.map(ExpectClassOnTask.java:36)
...
        Job Counters
                Failed map tasks=4
...
```

7. The required class, **`common.PropPrinter`**, can be found in $PLAY_AREA/HadoopSample.jar; add the required jar to the Mapper Task's CLASSPATH to run the job without an exception.

Perform – Extra Credit

1. Execute `mapRed.runningJobs.NeverEndingJob` whose Mapper implementation is an infinite loop that logs a message then sleeps for 2 seconds. This job will never end by itself.

```
$ yarn jar $PLAY_AREA/Exercises.jar \
     mapRed.runningJobs.NeverEndingJob \
     /training/data/hamlet.txt \
     /training/playArea/NeverEndingJob
2012-09-17 17:13:07,411 INFO  mapreduce.Job (Job.java:monitorAndPrintJob(1275)) -
Running job: job_1347904219060_0010
2012-09-17 17:13:13,832 INFO  mapreduce.Job (Job.java:monitorAndPrintJob(1296)) -
Job job_1347904219060_0010 running in uber mode : false
2012-09-17 17:13:13,834 INFO  mapreduce.Job (Job.java:monitorAndPrintJob(1303)) -
map 0% reduce 0%
```

   Locate a Map task via YARN Management UI by going to http://localhost:8088/cluster and

   1. navigating to this specific YARN application
   2. to the ApplicationMaster
   3. select your job
   4. Select a Map task

5. View Map's logs
6. Select syslog

you should see something like this in the log:

...

2012-09-17 17:19:54,887 INFO [main] mapRed.runningJobs.NeverEndingJob: Ha ha! I will never end! Enjoy this UUID: 88a2abd4-10ac-49b7-9a93-2cdac07636b2

2012-09-17 17:19:56,887 INFO [main] mapRed.runningJobs.NeverEndingJob: Ha ha! I will never end! Enjoy this UUID: 36fb875e-009d-466a-a377-da63d683c9b7

2012-09-17 17:19:58,888 INFO [main] mapRed.runningJobs.NeverEndingJob: Ha ha! I will never end! Enjoy this UUID: 8ff725f0-b16c-40aa-94ac-d9599e6cb35a

2. Use command line to display status of NeverEndingJob
3. Kill NeverEndingJob
4. `mapRed.runningJobs.CountTokens` class is a MapReduce job that calculates the total number of tokens in the provided file. Your task is to implement a JUnit test. Run the MapReduce job locally within unit tests. A JUnit class was set up for you in the Exercises project – `mapRed.runningJobs.CountTokensTests`; The unit test sets up a file with nine tokens therefore the job should produce the output of *"count 9"*

Solution

1. N/A
2. $ `yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.ExpectProperty -Dtraining.prop=hi`

   ```
   Property [training.prop] is set to [hi]
   ```

3. Follow these steps:

   ```
   $ vi conf.xml
   ```

   ```
   <?xml version="1.0"?>
   <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
   <configuration>
     <property>
       <name>training.prop</name>
       <value>fileProp</value>
     </property>
   </configuration>
   ```

   ```
   $ yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.ExpectProperty -conf conf.xml

   Property [training.prop] is set to [fileProp]
   ```

4. N/A
5. Follow these steps:
   a. *First Add HaddopSample.jar to the client's CLASSPATH by editing hadoop-env.sh*

      ```
      $ vi $HADOOP_CONF_DIR/hadoop-env.sh
      ```

      export HADOOP_CLASSPATH=$HBASE_HOME/*:$HBASE_HOME/conf:$HADOOP_CLASSPATH:**$PLAY_AREA/HadoopSamples.jar**

   b. verify that the jar is on the client's classpath

      ```
      $ yarn classpath | grep HadoopSamples
      ```

   c. run the Tool again:

      ```
      $ yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.ExpectClassOnClient
      Class [class common.PropPrinter] was on CLASSPATH
      ```

   d. *Clean up by removing HadoopSample.jar from the CLASSPATH*

      ```
      $ vi $HADOOP_CONF_DIR/hadoop-env.sh
      export HADOOP_CLASSPATH=$HBASE_HOME/*:$HBASE_HOME/conf:$HADOOP_CLASSPATH
      ```

6. N/A
7. $ yarn jar $PLAY_AREA/Exercises.jar mapRed.runningJobs.ExpectClassOnTask \

   -libjars $PLAY_AREA/HadoopSamples.jar \

   /training/data/hamlet.txt /training/playArea/ExpectClassOnTask

Extra Credit Solution

1. N/A
2. `$ mapred job -status job_1347904219060_0010`
3. `$ mapred job -kill job_1347904219060_0010`

   ```
   Killed job job_1347904219060_0010
   ```

4. The code can be found in the Solution's project:
   `/src/test/java/mapRed.runningJobs.CountTokensTests.java`