Lab: Running a MapReduce Job

|  |  |
| --- | --- |
| **Objective:** | To learn tips on how to troubleshoot issues with a MapReduce job. |
| **Successful Outcome:** | You will have run the IndexInverter MapReduce job several times and viewed its log files. |
| **Before You Begin:** | Go into terminal on hadoop-master. |

1. Put the File into HDFS
   1. Change directories to the materials folder:

# cd /root/materials

* 1. Notice a file named urls.txt. View its contents:

# more urls.txt

This file contains URLS, along with keywords found on the webpages of each URL.

**NOTE**: The MapReduce job in this lab computes an inverted index, one of the earliest use cases of Hadoop and MapReduce. A Web crawler scans the Internet and retrieves URLs along with keywords on each page. The index inverter job flips this information around, outputting the keywords along with each web page that contains the keyword.

1. Run the IndexInverterJob
   1. Enter the following command to run the IndexInverterJob:

# hadoop jar invertedindex.jar inverted.IndexInverterJob urls.txt index\_output

* 1. The job fails. What is the issue? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Run the Job Again
   1. Put the file /root/materials/urls.txt into HDFS:

# hadoop fs -put /root/materials/urls.txt urls.txt

*NOTE: you may need to enter a* ***&> output.txt*** *to see results.*

* 1. Run the job again, using the same command as the previous step.
  2. The job should run successfully this time. How many map tasks were needed for this job? \_\_\_\_\_\_\_\_

How many reduce tasks? \_\_\_\_\_\_\_\_\_\_\_

* 1. How long (in ms) did it take for all the mappers to run? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  2. How long (in ms) did it take for all the reducers to run? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  3. How many bytes did the mappers of this job process? \_\_\_\_\_\_\_\_\_\_\_\_
  4. How many bytes did the reducers output? \_\_\_\_\_\_\_\_\_\_\_\_

1. View the Output
   1. Verify the index\_output folder was created in HDFS:

# hadoop fs -ls index\_output

You should see a single output file named part-r-00000:

* 1. View the contents of part-r-00000:

# hadoop fs -cat index\_output/part-r-00000

* 1. What did the reducer use as the key for its output? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  2. What did the reducer use as the values for its output? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Run the Job Again
   1. Run the IndexInverterJob again with the exact same command.
   2. The job failed. Why? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. Delete the index\_output folder in HDFS.
   4. Run the job again, and it should run successfully this time.
2. View the Ambari UI
   1. Point your web browser to Ambari at http://localhost:8080.
   2. Point your web browser to the Job tab to the right of dashboard. You can see the job details there.
3. [OPTIONAL] Troubleshoot a Problem
   1. From a previous lab, you may have a JAR file in /user/root in HDFS:

# hadoop fs -ls shakespeare.jar

Found 1 items

-rw-r--r-- 3 root root 4708852 hbase.jar

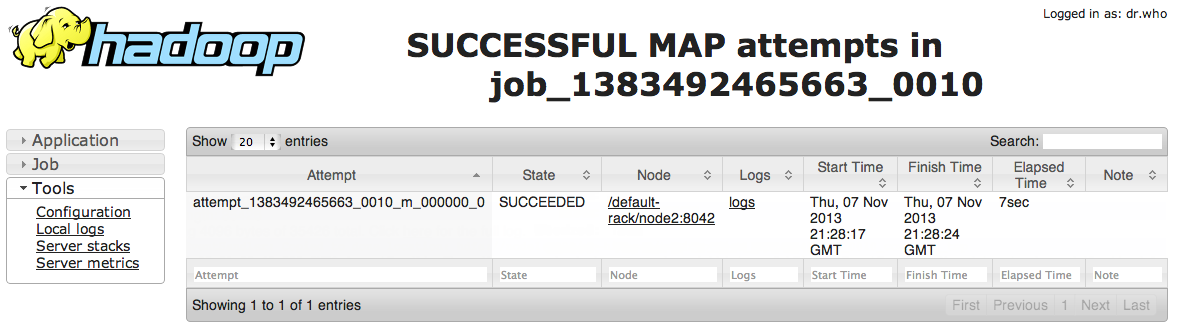
If you do not have this file in HDFS, put it there. The file should be found in your /root/materials folder.

* 1. Run the IndexInverterJob using the following command (entered on a single line):

# hadoop jar invertedindex.jar inverted.IndexInverterJob shakespeare.jar index\_output2

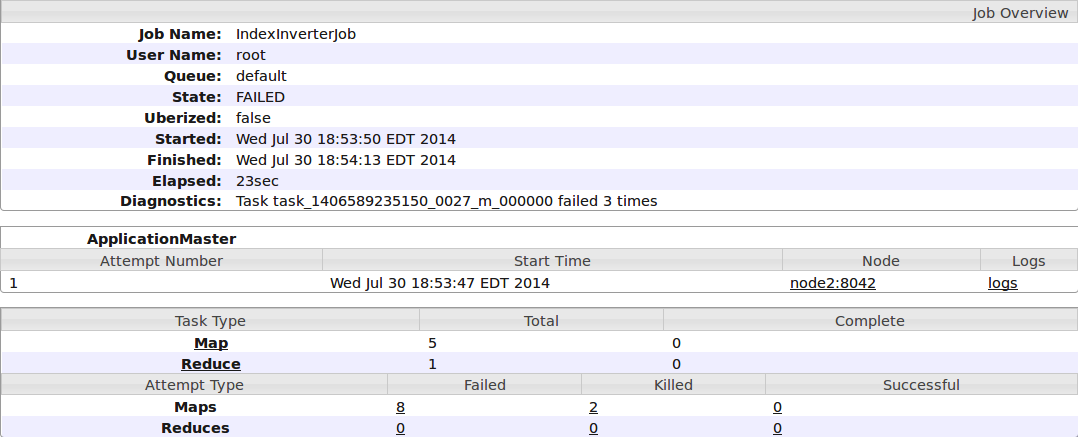
* 1. Notice exceptions are thrown, and eventually the job will fail. From the output of the job, how many map tasks were launched? \_\_\_\_\_\_\_\_\_ How many map tasks failed? \_\_\_\_\_\_\_\_ How many were killed? \_\_\_\_\_\_\_\_
  2. The input file, hbase.jar, is split into 5 blocks in HDFS. Why did this MapReduce job launch 10 map tasks? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. [Optional] View the Log Files – depending on your release of Hadoop
   1. Let’s figure out what happened to the IndexInverterJob. View the Job History page of node2 by pointing your browser to http://localhost:19888:

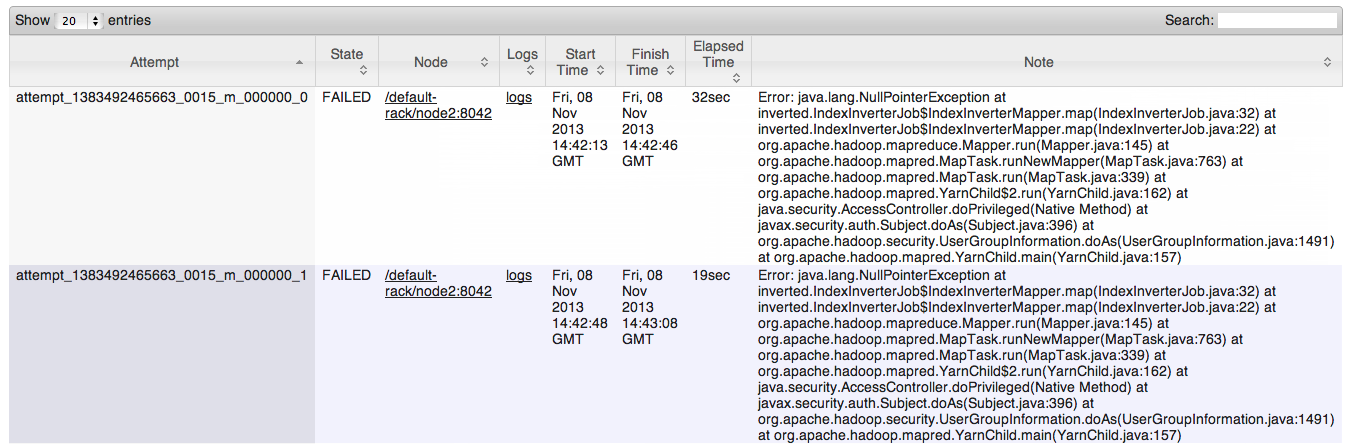


Notice the most recent job at the top of the list has a status of FAILED.

* 1. Click on Job ID of the failed IndexInverterJob. You should see the details page for this job:



* 1. Notice this page contains useful details about the job, including the average map and reduce time, and how long it took to execute the entire job.
  2. Also notice that 5 mappers and 1 reducer were started for this job. In the screen shot above, 8 map tasks failed, 2 were killed, and 0 were successful. Notice these numbers are links - click on your number of failed map tasks:



* 1. In the Logs column, click on the logs link of one of the failed map tasks to view the corresponding log file:



* 1. What happened in this job? Why did the mapper fail? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**RESULT**: You have executed a MapReduce job that failed for several different reasons. Being able to troubleshoot these types of issues is an important and handy skill for any Hadoop administrator.

**ANSWERS**:

2.2: The job fails because the input file for the job does not exist in HDFS.

3.3: 1 mapper and 1 reducer, as found in the Job Counters section of the output.

3.4: Look for the counter “Total time spent by all maps in occupied slots (ms)”

3.5: Similarly, look for “Total time spent by all reduces in occupied slots (ms)”

3.6: Bytes Read=1126, as found in the File Input Format Counters section.

3.7: Bytes Written=2997, as found in the File Output Format Counters section.

5.2: The output folder of a MapReduce job cannot exist. You should have gotten the following error message: “FileAlreadyExistsException: Output directory hdfs://node1:8020/user/root/index\_output already exists”

7.3: You should see 10 launched map tasks. The number of failed and killed tasks will vary, but expect about 8 failed and 2 killed.

7.4: When a map task fails, the MapReduce framework launches the map task again. A map task has to fail 2 times (by default) before the entire job fails. The input file was split into 5 blocks, and each block generated a map task that failed 2 times, so 5x2=10.

8.6: A NullPointerException was thrown on line 32 of IndexInverterJob.java. Useful information for the Java developer!