Xi'an Jiaotong-Liverpool University
西交利物浦大學

DTS303TC Big Data Security and Analytics

Laboratory Session 1 – Setting Up Cloudera and Hadoop – Word Count and Map Reduce

(Duration: 2 hours)

Overview:

According to the requirements of the syllabus, through the installation of big data tools and the use of cases, a preliminary understanding is obtained.

In this activity, you will:

- Launch the Cloudera VM.
- Execute the Word Count application.
- Copy the results from Word Count out of HDFS.

Hardware Requirements: (A) Quad Core Processor, 64-bit; (B) 8 GB RAM; (C) 20 GB disk free.

PART 1: Downloading and Installing the Cloudera VM Instructions (Windows)

Task Instructions

Please use the following instructions to download and install the Cloudera Quickstart VM with VirtualBox before proceeding to the Getting Started with the Cloudera VM Environment video. The screenshots are from a Mac but the instructions should be the same for Windows. Please see the discussion boards if you have any issues.

Step1: Install VirtualBox. Go to https://www.virtualbox.org/wiki/Downloads to download and install VirtualBox for your computer. The course uses VirtualBox 6.1.X. For Windows, select the link "VirtualBox 6.1.X for Windows hosts x86/amd64" where 'X' is the latest version(6.1.36).

Step2: Download the Cloudera VM. Download the Cloudera VM from https://downloads.cloudera.com/demo_vm/virtualbox/cloudera-quickstart-vm-5.5.0-0-virtualbox.zip. The VM is over 4GB, so will take some time to download.

Step 3: Unzip the Cloudera VM:

Right-click cloudera-quickstart-vm-5.5.0-0-virtualbox.zip and select "Extract All…"

Step 4: Start VirtualBox.

Step 5: Begin importing. Import the VM by going to File -> Import Appliance

Step 6: Click the Folder icon.

Step 7: Select the cloudera-quickstart-vm-5.5.0-0-virtualbox.ovf from the Folder where you unzipped the VirtualBox VM and click Open.

Step 8: Click Next to proceed.

Step 9: Click Import.

Step 10: The virtual machine image will be imported. This can take several minutes.

Step 11: Launch Cloudera VM. When the importing is finished, the quickstart-vm-5.5.0-0 VM will appear on the left in the VirtualBox window. Select it and click the Start button to launch the VM.

Step 12: Cloudera VM booting. It will take several minutes for the Virtual Machine to start. The booting process takes a long time since many Hadoop tools are started.

Step 13: The Cloudera VM desktop. Once the booting process is complete, the desktop will appear with a browser.

PART 2: Running the Word Count Map Reduce Example

Task Instructions

Please use the following WordCount program Instructions on Cloudera VM in VirtualBox before proceeding to the Getting Started with the Word Count Map Reduce Example video.

```
[cloudera@quickstart ~]$ cd Downloads/
[cloudera@quickstart Downloads]$ ls
words.txt
[cloudera@quickstart Downloads]$ hadoop fs -copyFromLocal words.txt
[cloudera@quickstart Downloads]$ hadoop fs -ls
Found 1 items
-rw-r--r--   1 cloudera cloudera    5458199 2022-08-21 05:20 words.txt
```

Step 1: Open a terminal shell. Start the Cloudera VM in VirtualBox, if not already running, and open a terminal shell. Detailed instructions for these steps can be found in the previous Readings.

Step 2: See example MapReduce programs. Hadoop comes with several example MapReduce applications. You can see a list of them by running *hadoop jar /usr/jars/hadoop-examples.jar*. We are interested in running WordCount.

```
[cloudera@quickstart Downloads]$ hadoop jar /usr/jars/hadoop-examples.jar
An example program must be given as the first argument.
Valid program names are:
  aggregatewordcount: An Aggregate based map/reduce program that counts the words in the input files.
  aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of the words in the input file
s.
  bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
  dbcount: An example job that count the pageview counts from a database.
  distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
  grep: A map/reduce program that counts the matches of a regex in the input.
  join: A job that effects a join over sorted, equally partitioned datasets
  multifilewc: A job that counts words from several files.
  pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
  pi: A map/reduce program that estimates Pi using a quasi-Monte Carlo method.
  randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.
  randomwriter: A map/reduce program that writes 10GB of random data per node.
  secondarysort: An example defining a secondary sort to the reduce.
  sort: A map/reduce program that sorts the data written by the random writer.
  sudoku: A sudoku solver.
  teragen: Generate data for the terasort
  terasort: Run the terasort
  teravalidate: Checking results of terasort
  wordcount: A map/reduce program that counts the words in the input files.
  wordmean: A map/reduce program that counts the average length of the words in the input files.
  wordmedian: A map/reduce program that counts the median length of the words in the input files.
  wordstandarddeviation: A map/reduce program that counts the standard deviation of the length of the words in the in
put files.
```

Step 3: Verify input file exists. In the previous Reading, we downloaded the complete works of Shakespeare and copied them into HDFS. Let's make sure this file is still in HDFS so we can run WordCount on it. Run *hadoop fs -ls*

```
[cloudera@quickstart Downloads]$ hadoop fs -ls
Found 1 items
-rw-r--r--   1 cloudera cloudera    5458199 2022-08-21 05:20 words.txt
```

Step 4: See WordCount command line arguments. We can learn how to run WordCount by examining its command-line arguments. Run *hadoop jar /usr/jars/hadoop-examples.jar wordcount*

```
[cloudera@quickstart Downloads]$ hadoop jar /usr/jars/hadoop-examples.jar wordcount
Usage: wordcount <in> [<in>...] <out>
```

Step 5: Run WordCount. Run WordCount for words.txt: *hadoop jar /usr/jars/hadoop-examples.jar wordcount words.txt out*

```
[cloudera@quickstart Downloads]$ hadoop jar /usr/jars/hadoop-examples.jar wordcount words.txt out
22/08/21 05:22:36 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032

22/08/21 05:22:36 INFO input.FileInputFormat: Total input paths to process : 1
22/08/21 05:22:36 INFO mapreduce.JobSubmitter: number of splits:1
22/08/21 05:22:37 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1661084039644_0001
22/08/21 05:22:37 INFO impl.YarnClientImpl: Submitted application application_1661084039644_0001
22/08/21 05:22:37 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_166
1084039644_0001/
22/08/21 05:22:37 INFO mapreduce.Job: Running job: job_1661084039644_0001
22/08/21 05:22:43 INFO mapreduce.Job: Job job_1661084039644_0001 running in uber mode : false
22/08/21 05:22:43 INFO mapreduce.Job:  map 0% reduce 0%
22/08/21 05:22:49 INFO mapreduce.Job:  map 100% reduce 0%
22/08/21 05:22:54 INFO mapreduce.Job:  map 100% reduce 100%
22/08/21 05:22:54 INFO mapreduce.Job: Job job_1661084039644_0001 completed successfully
22/08/21 05:22:54 INFO mapreduce.Job: Counters: 49
```

Step 6: See WordCount output directory. Once WordCount is finished, let's verify the output was created. First, let's see that the output directory, out, was created in HDFS by running *hadoop fs –ls*

```
[cloudera@quickstart Downloads]$ hadoop fs -ls
Found 2 items
drwxr-xr-x   - cloudera cloudera          0 2022-08-21 05:22 out
-rw-r--r--   1 cloudera cloudera    5458199 2022-08-21 05:20 words.txt
```

Step 7: Look inside output directory. The directory created by WordCount contains several files. Look inside the directory by running *hadoop –fs ls out*

```
[cloudera@quickstart Downloads]$ hadoop fs -ls out
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2022-08-21 05:22 out/_SUCCESS
-rw-r--r--   1 cloudera cloudera     717768 2022-08-21 05:22 out/part-r-00000
```

Step 8: Copy WordCount results to local file system. Copy part-r-00000 to the local file system by running *hadoop fs –copyToLocal out/part-r-00000 local.txt*

```
[cloudera@quickstart Downloads]$ hadoop fs -copyToLocal out/part-r-00000 local.txt
```

Step 9: View the WordCount results. View the contents of the results: *more local.txt*

```
[cloudera@quickstart Downloads]$ more local.txt
"       241
"'Tis   1
"A      4
"AS-IS".        1
"Air,"  1
"Alas,  1
"Amen"  2
"Amen"? 1
"Amen," 1
"And    1
"Aroint 1
"B      1
"Black  1
"Break  1
"Brutus"        1
"Brutus,        2
"C      1
"Caesar"?       1
"Caesar,        1
"Caesar."       2
```