Hadoop Fundamentals

Unit 2: Hadoop Architecture

Contents

Lab 2 Hadoop Architecture

The overwhelming trend towards digital services, combined with cheap storage, has generated massive amounts of data that enterprises need to effectively gather, process, and analyze. Data analysis techniques from the data warehouse and high-performance computing communities are invaluable for many enterprises, however often times their cost or complexity of scale-up discourages the accumulation of data without an immediate need. As valuable knowledge may nevertheless be buried in this data, related scaled-up technologies have been developed. Examples include Google's MapReduce, and the open-source implementation, Apache Hadoop.

Hadoop is an open-source project administered by the Apache Software Foundation. Hadoop's contributors work for some of the world's biggest technology companies. That diverse, motivated community has produced a collaborative platform for consolidating, combining and understanding data. After completing this hands-on lab, you'll be able to:

- Use Hadoop commands to explore HDFS on the Hadoop system

Allow 10 minutes to 20 minutes to complete this lab.

This version of the lab was designed using the IBM Analytics Cloud Sandbox. Throughout the lab, use the same username and password that was used in the Lab Setup.

2.1 Getting Started

1. Log into your account on https://my.imdemocloud.com.
2. Startup Ambari and PuTTY and log into both of them.

2.2 Basic HDFS interactions using the Command Line

The Hadoop Distributed File System (HDFS) allows user data to be organized in the form of files and directories. It provides a command line interface called FS shell that lets a user interact with the data in HDFS accessible to Hadoop MapReduce programs.

You can interact with HDFS at the command line:

1. **`hdfs dfs`** ***`command options`***

Where command is the particular command (**`ls`**, **`rm`**, **`mkdir`**, …) and options are variations on the particular command and may be followed by a list of files or a list of directories. The command is preceded by a single dash ("-") and the options may be preceded by a single dash.

Start with the **`ls`** command to list files and directories. In your VM, type the following command and hit **Enter** . Pause after each to review your results.

```
hadoop fs -ls
hadoop fs -ls .
hadoop fs -ls /
```

```
[trand@iop-bi-master ~]$ hadoop fs -ls
[trand@iop-bi-master ~]$ hadoop fs -ls .
[trand@iop-bi-master ~]$ hadoop fs -ls /
Found 10 items
drwxrwxrwx   - yarn   hadoop          0 2016-01-15 01:49 /app-logs
drwxr-xr-x   - hdfs   hdfs            0 2015-10-13 16:46 /apps
drwxrwxr-x   - hdfs   hadoop          0 2015-12-23 11:31 /biginsights
drwxr-xr-x   - hdfs   hdfs            0 2015-12-22 19:39 /ibmpacks
drwxr-xr-x   - hdfs   hdfs            0 2015-10-13 16:43 /iop
drwxr-xr-x   - mapred hdfs            0 2015-10-13 16:43 /mapred
drwxr-xr-x   - hdfs   hdfs            0 2015-10-13 16:43 /mr-history
drwxrwxr-x   - hdfs   hdfs            0 2015-10-27 13:27 /sample_data
drwxrwxrwx   - hdfs   hdfs            0 2016-01-13 11:41 /tmp
drwxrwx--x+  - hdfs   hdfs            0 2016-01-18 17:02 /user
[trand@iop-bi-master ~]$
```

The first of these lists the files in the current directory — there are none. The second is a little more explicit since it asks for files in dot (“.”), a synonym for “here” (again the current directory). The third lists files at the root level within the HDFS (and there are ten directories).

Look at the directory, **/iop/apps/4.1.0.0** — this is an example of other files and directories that are stored in HDFS. Some directories listed here are pig, spark, sqoop and more, which are names of Big Data programming! If you are curious about any of these, Big Data University has courses on all of these programs!

```
hadoop fs -ls /iop/apps/4.1.0.0
```

```
[trand@iop-bi-master ~]$ hadoop fs -ls /iop/apps/4.1.0.0
Found 5 items
dr-xr-xr-x   - hdfs hdfs          0 2015-10-13 16:45 /iop/apps/4.1.0.0/hive
dr-xr-xr-x   - hdfs hdfs          0 2015-10-13 16:45 /iop/apps/4.1.0.0/mapreduce
dr-xr-xr-x   - hdfs hdfs          0 2015-10-13 16:45 /iop/apps/4.1.0.0/pig
drwxr-xr-x   - hdfs hdfs          0 2016-01-26 09:19 /iop/apps/4.1.0.0/spark
dr-xr-xr-x   - hdfs hdfs          0 2015-10-13 16:45 /iop/apps/4.1.0.0/sqoop
[trand@iop-bi-master ~]$
```

Create a directory called *test* in the /user/spark/ directory

```
hadoop fs -mkdir test
```

Check the contents of your home directory before and after the command to see that is created.

```
hadoop fs -ls
```

```
[trand@iop-bi-master ~]$ hadoop fs -ls
[trand@iop-bi-master ~]$ hadoop fs -mkdir test
[trand@iop-bi-master ~]$ hadoop fs -ls
Found 1 items
drwxrwx---+  - trand trand          0 2016-01-19 16:16 test
[trand@iop-bi-master ~]$
```

Create a file in your Linux home directory. Execute the following commands: (Ctrl-c means to press-and-hold the **Ctrl** key and then the **c** key.)

```
cd ~
cat > myfile.txt
this is some data
in my file
Ctrl-c
```

Next upload this newly created file to the *test* directory that you just created. In this example, <username> is trand. The text file is located in /mnt/home/<username>, where /mnt/home/ is a shared directory that contains all users "accounts".

```
hadoop fs -put /mnt/home/<username>/*.txt test
```

```
[trand@iop-bi-master ~]$ cat > myfile.txt
this is some data
in my file
^C
[trand@iop-bi-master ~]$ hadoop fs -put /mnt/home/trand/*.txt test
[trand@iop-bi-master ~]$
```

Now list your *test* directory in HDFS. Note that –R recursively list the contents of directories:

```
hadoop fs –ls –R .
```

```
[trand@iop-bi-master ~]$ hadoop fs -ls -R .
drwxrwx---+  - trand trand          0 2016-01-19 16:24 test
-rw-rw----+  3 trand trand         29 2016-01-19 16:24 test/myfile.txt
[trand@iop-bi-master ~]$
```

Note the number 3 that follows the permissions. This is the replication factor for that data file. Normally, in a cluster this is 3, but sometimes in a single-node cluster such as the one that you are running with, there might be only one copy of each block ("split") of the this file.

The value 3 (or 1, or something else) is the result of a configuration setting of HDFS that sets the number of replicants by default.

To view the contents of the uploaded file, execute

```
hadoop fs -cat test/myfile.txt
```

```
[trand@iop-bi-master ~]$ hadoop fs -cat test/myfile.txt
this is some data
in my file
[trand@iop-bi-master ~]$
```

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

```
hadoop fs -cat test/myfile.txt | grep my
```

```
[trand@iop-bi-master ~]$ hadoop fs -cat test/myfile.txt | grep my
in my file
[trand@iop-bi-master ~]$
```

Or,

```
hadoop fs -ls -R . | grep test
```

```
[trand@iop-bi-master ~]$ hadoop fs -ls -R . | grep test
drwxrwx---+  - trand trand          0 2016-01-19 16:24 test
-rw-rw----+  3 trand trand         29 2016-01-19 16:24 test/myfile.txt
[trand@iop-bi-master ~]$
```

To find the size of a particular file, like *myfile.txt,* execute the following:

```
hadoop fs -du test/myfile.txt
```

Or, to get the size of all files in a directory by using a directory name rather than a file name.

```
hadoop fs -du test
```

Or get a total file size value for all files in a directory:

```
hadoop fs -du -s test
```

Note: I added in a second file called myfile2.txt to show the difference between hadoop fs –du test/myfile.txt and hadoop fs –du test

```
[trand@iop-bi-master ~]$ hadoop fs -du test/myfile.txt
29  test/myfile.txt
[trand@iop-bi-master ~]$ hadoop fs -du test
29  test/myfile.txt
18  test/myfile2.txt
[trand@iop-bi-master ~]$ hadoop fs -du -s test
47  test
[trand@iop-bi-master ~]$
```

Remember that you can always use the **-help** parameter to get more help:

```
hadoop fs -help
hadoop fs -help du
```

```
[trand@iop-bi-master ~]$ hadoop fs -help du
-du [-s] [-h] <path> ... :
  Show the amount of space, in bytes, used by the files that match the specified
  file pattern. The following flags are optional:

  -s  Rather than showing the size of each individual file that matches the
      pattern, shows the total (summary) size.
  -h  Formats the sizes of files in a human-readable fashion rather than a number
      of bytes.

  Note that, even without the -s option, this only shows size summaries one level
  deep into a directory.

  The output is in the form
        size    name(full path)
[trand@iop-bi-master ~]$
```

You can close the command line window.

2.3 Summary

Congratulations! You are now familiar with the Hadoop Distributed File System (HDFS). You now know how to manipulate files within HDFS by using the command line.

You may move on to the next unit.

NOTES

NOTES

Please Recycle

IBM Software