Hadoop Architecture Labs

Lab 3: Loading Data Into HDFS

Begin by connecting to the command line on your cluster using the `vagrant ssh` command.

```
    ssh -i ~/sparkcourse.pem hadoop@$MY_IP_ADDRESS
```

You should see output that looks like the following:

```
Last login: Sun Aug 20 18:06:02 2017 from 10.0.2.2
[hadoop@ip-172-31-57-176 mnt]$
```

Now change directories to /vagrant and examine the contents of that directory:

```
    cd /mnt/sparkclass
    ls
```

You should see the following contents that live in the edge node's local file system. Your formatting should be prettier than what's shown below due to this document's line length limits.

```
data  exercises  handouts  images  slides
[hadoop@ip-172-31-57-176 sparkclass]$
```

Now look at the contents of the HDFS file system current directory using the `hdfs dfs -ls` command. You'll see different results:

```
[hadoop@ip-172-31-57-176 sparkclass]$ hdfs dfs -ls
Found 1 items
drwxr-xr-x   - vagrant supergroup          0 2017-08-20 17:29 .sparkStaging
[hadoop@ip-172-31-57-176 sparkclass]$
```

Let's examine the root directory of HDFS. From this point on, we'll simply show the commands we type (as indicated by the `[vagrant@edge vagrant]$` prompt) and the result in the same code block.

```
[hadoop@ip-172-31-57-176 sparkclass]$ hdfs dfs -ls /
Found 4 items
drwxr-xr-x   - hdfs hadoop          0 2017-08-24 18:36 /apps
drwxrwxrwt   - hdfs hadoop          0 2017-08-24 18:39 /tmp
drwxr-xr-x   - hdfs hadoop          0 2017-08-24 18:36 /user
drwxr-xr-x   - hdfs hadoop          0 2017-08-24 18:36 /var
[hadoop@ip-172-31-57-176 sparkclass]$
```

We're now going to add the contents of the local data directory to the HDFS directory /data. We do this using the `hdfs dfs -put` command and then examine the /data directory with `hdfs dfs -ls /data` command. The first command will take a minute or two to copy roughly 1.5GB of data.

```
[hadoop@ip-172-31-57-176 sparkclass]$ hdfs dfs -put data /
[hadoop@ip-172-31-57-176 sparkclass]$ hdfs dfs -ls /data
Found 22 items
-rw-r--r--   1 hadoop hadoop      10244 2017-08-29 11:15 /data/.DS_Store
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:15 /data/bag-o-words
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:15 /data/big-r-data
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:15 /data/data
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:15 /data/dividends
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:15 /data/dividends-flat
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:15 /data/employees
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:15 /data/employees-pig
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/ge
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/logs
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/ml-100k
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/movies
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/orders-pig
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/python
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/shakespeare
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/shakespeare_wc
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/spark-resources-data
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/sparkml-data
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/stocks
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/stocks-flat
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/twitter
drwxr-xr-x   - hadoop hadoop          0 2017-08-29 11:16 /data/twitter-pig
[hadoop@ip-172-31-57-176 sparkclass]$
```

These directories contain source datasets that we can use throughout this Hadoop course. However, not many of these datasets are large ones. To allow us to give our Hadoop cluster a bit more exercise, we'll load some multi-million line datasets regarding US airline flights that we've downloaded from the FAA. These datasets are in the local file directory `flightdata`. We'll add that dataset to the `/data` directory on HDFS using another `put` command.

One way you could do that is as follows.

```
[hadoop@ip-172-31-57-176 sparkclass]$ hdfs dfs -mkdir /data/flightdata
[hadoop@ip-172-31-57-176 sparkclass]$ ls flightdata
aircraft-registrations  parquet-trimmed  tickets
[hadoop@ip-172-31-57-176 sparkclass]$ hdfs dfs -put flightdata/* /data/flightdata
```

Just do demonstrate how we would remove a dataset, we'll delete that directory and its contents from HDFS and load the data in a simpler way.

```
[hadoop@ip-172-31-57-176 sparkclass]$ hdfs dfs -rm -R /data/flightdata
17/08/23 20:40:03 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval
Deleted /data/flightdata
[hadoop@ip-172-31-57-176 sparkclass]$ hdfs dfs -put flightdata /data
```

This step concludes the lab.