Class 1: Homework

New York University

Spring 2019

Homework

Class 1

Homework

1. You will require access to a Hadoop/Spark cluster in order to complete the homework assignments for this class.
   For homework assignments you can use Dumbo (recommended) or the Cloudera Quickstart VM.

   a. Easiest: Apply for a Dumbo account (this is NYU's Hadoop cluster, it has Spark and the Spark shells installed).
   I recommend using Dumbo, especially if your host machine has too little memory (<= 4GB is too little memory to use the VM).
   There is no cost for using Dumbo.

      1. Request an account on this site (you can select Suzanne McIntosh for sponsor):
      https://wikis.nyu.edu/display/NYUHPC/Getting+or+renewing+an+HPC+account
      You will use putty to log into Dumbo - instructions are on the wiki.

      2. You can read about Dumbo here:
      https://wikis.nyu.edu/display/NYUHPC/Clusters+-+Dumbo

      3. Once you have an account, instructions for logging in are here:
      https://wikis.nyu.edu/display/NYUHPC/Clusters+-+Dumbo#Clusters-Dumbo-LOGGING_INLoggingIn

      4. If you run into trouble, please use the Forum and/or send a note to the NYU HPC IT group at: hpc@nyu.edu

   b. Download the Cloudera Quickstart VM (recommend using VirtualBox because the trial license is long):
   https://www.cloudera.com/downloads/quickstart_vms/5-10.html
   You will also need to install VirtualBox if you don't already have it.
   Everything you need in order to complete the Spark homework assignments is already installed in the VM
   (Hadoop-HDFS, Spark, Spark shells for Scala and Python, etc.).

   ** Please make sure your host machine meets the memory requirements posted on the VM website.*

Homework (continued)

2. Verify that HDFS is working on Dumbo (if you're using the Quickstart VM, verify HDFS works there too).

Try out the Hadoop HDFS commands in your Hadoop environment:

A great reference: *http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-common/FileSystemShell.html*

Open a terminal window and type these commands at the Linux command line - you shouldn't see any errors:

```
hdfs dfs -ls /              -- View the contents of the top-level directory in HDFS

hdfs dfs -ls                -- View the contents of your user directory (likely empty)

hdfs dfs -mkdir myNewDir    -- Create a new directory named 'myNewDir' in your user directory

hdfs dfs -ls                -- Verify that you now have a directory called 'myNewDir'

hdfs dfs -rm -r myNewDir    -- Remove directory 'myNewDir'

hdfs dfs -ls                -- Verify that you successfully removed 'myNewDir'
```

Homework (continued)

3. Create file **cs_fun.txt** use your favorite editor (e.g. vi, emacs) and cut and paste the following content into the file (from James Iry's "A Brief, Incomplete, and Mostly Wrong History of Programming Languages"):

```
1940s - Various "computers" are "programmed" using direct wiring and switches. Engineers
        do this in order to avoid the tabs vs spaces debate.
```

Open a terminal window and type these commands - you shouldn't see any errors;

```
hdfs dfs -mkdir funny_input                    -- Create a new directory
hdfs dfs -put cs_fun.txt funny_input           -- Put your file into HDFS
hdfs dfs -cat funny_input/cs_fun.txt           -- Output the contents of your HDFS file
cat cs_fun.txt                                 -- Output the contents of your local file

-- Get the file from HDFS and store it locally into new_copy_from_hdfs.txt:
hdfs dfs -get funny_input/cs_fun.txt new_copy_from_hdfs.txt

cat new_copy_from_hdfs.txt                     -- View the new local version of the file

diff cs_fun.txt new_copy_from_hdfs.txt         -- The two files should be the same
```

Homework (continued)

4. Verify that the Spark REPL (shell) is working in your environment (Dumbo or the Quickstart VM).
   In the already open terminal window type the following command to start the Spark shell - you shouldn't see any errors (warnings can be ignored):

```
$ spark-shell                          — Start the Scala version of the Spark REPL
```

… After some output from the shell, you should see a scala> prompt …

Type the following commands and take a screenshot to show that your Spark environment is working. Upload the screenshot to NYU Classes:

```
scala> :help                           — In the Spark shell, try the help command

scala> sc[TAB]                         — View the commands available in the Spark Context (sc)

scala> sc.version                      — View the version of Spark that is running in the shell

scala> val myConstant: Int = 2016

scala> myConstant

scala> my[TAB]

scala> myConstant.[TAB]

scala> myConstant.to[TAB]

scala> myConstant.toFloat

scala> myConstant                        — Note that myConstant has not changed; it's still an Int

scala> myConstant.toFloat.toInt

scala> val myString = myConstant         — Note the type inferred for myString

scala> :type val myString2 = myConstant — Use the :type command to view the type that is inferred for myString2
```

Homework (continued)

5. Please read the following papers if you have not already done so for a previous class.
   - "MapReduce: Simplified Data Processing on Large Clusters", Dean and Ghemawat, OSDI 2004.
     http://static.usenix.org/event/osdi04/tech/full_papers/dean/dean.pdf
     Describes how the MapReduce paradigm was implemented by Google. This work was the inspiration for Hadoop MapReduce (Hadoop's open source implementation of the MapReduce paradigm).

   - "The Google File System", by Ghemawat, Gobioff, and Leung.
     http://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf
     Read sections 1 and 2 at least. You will notice a difference in terminology when compared with HDFS; GFS was the inspiration for the open source Hadoop Distributed File System, HDFS.