Class 3: Homework

New York University

Spring 2018

Homework - Part 1

1. Become familiar with online Spark documentation by checking out: http://spark.apache.org/docs/1.6.1/
    - From the Programming Guides menu at the top of the page, select the Spark Programming Guide and scan through it
    - From the API Docs menu at the top of the page, select Scala and scan the material

2. Please read in the class text: Chapter 2 and Chapter 3

3. Please read: "Spark: Cluster Computing with Working Sets"
    By Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica

Homework - Part 2

4. Use the REPL to explore Spark RDDs. (Use the Answer Sheet document to record the answers to the blue prompts below.)

    a. In a terminal window, start the Scala Spark Shell: `$ spark-shell`

      Note: For Dumbo, use this command instead: `spark-shell --master local`

        On the VM, `spark-shell` is all you need, but for a true cluster like Dumbo, you need to issue: `spark-shell --master local`

        HPC is defaulting the running of commands to Dumbo's worker nodes, but our local file is not necessarily on the worker node Hadoop selected so we get a File-Not-Found exception.

        To make this work, start the shell like this for Dumbo when you want to use a local file: `spark-shell --master local`
        You will also need to specify the file as: `"file:///home/yourNetID/frostroad.txt"`

    b. Spark creates a SparkContext object for you called sc. Make sure the object exists: `scala> sc`

    c. Using command completion, you can see all the available SparkContext methods: type: `sc.[TAB]`

    d. Copy the input file, `frostroad.txt`, into the local filesystem of your VM (not in HDFS).

    e. Define an RDD named `mydata` to be created from `frostroad.txt` that exists in your local file system (not HDFS).
    (Reference the file as: `"file:///path/to/your/file/frostroad.txt"`)

        1) Provide the command you used to create your RDD.

    f. Once the above command is issued, remember that Spark has not yet read the file. It will not do so until you perform an action on the RDD. Count the number of lines in the dataset.

        2) Provide the command you used to count the elements (lines) in your RDD.
        3) Provide the number of elements.

    g. Use collect to display the data in the RDD. This is convenient for very small RDDs like this one, but be careful using collect for very large datasets.
        4) Provide the collect command you used.

    h. Using command completion, view the available transformations and actions you can perform on an RDD. Type: `mydata.[TAB]`

Homework - Part 2 (continued)

5. Transform a small dataset using RDDs. (Use the Answer Sheet document to record the answers to the blue prompts below.)

a. Copy the weblog file, **2014-03-15.log**, into the VM. Create a directory in HDFS called **loudacre/weblog** and put the file into the **weblog** directory.

5) Provide the command you used to create the HDFS directory.
6) Provide the command you used to put the file into HDFS.

b. View the HDFS version of the file.

7) Provide the command you used to view the file.

The format of the file is:
IP Address: **116.180.70.237**

                    -
User ID :   **128** *[15/Sep/2013:23:59:53 +0100]*
Request: **"GET /KBDOC-00031.html HTTP/1.0"**

                    ...

Homework - Part 2 (continued)

5. Continued: Transform a small dataset using RDDs. (Use the Answer Sheet document to record the answers to the blue prompts below.)

c. Store the full file path to a variable named `logfile`, then process the lines in `logfile` as follows:

Provide the commands you used for all of the following steps:

8) Initialize `logfile`.

9) Create an RDD from the file.

10) View 10 lines of the data.

11) Create an RDD containing only lines that are requests for `jpg` files.

12) View 10 lines of the data.

13) Chain the previous commands into a single command that counts the number of JPG requests.

14) Create an RDD using the `map` function to return the length of each line of the log file.

15) Create an RDD using the `map` and `split` functions to map an array of words for each line.

16) Create an RDD containing only the IP addresses from each line.

17) Use `foreach(println)` to output IP addresses.

18) Save the list of IP addresses to an HDFS directory named `loudacre/iplist` using `saveAsTextFile`.

19) Provide a screenshot of the contents of the `loudacre/iplist` folder.

Homework - Part 2 (continued)

6. Transform a large dataset using RDDs. (Use the Answer Sheet document to record the answers to the blue prompts below.)

    e. Copy the `weblogs.zip` file to the VM, unzip it, and store it to the `loudacre` directory.

        Provide the commands you used for all of the following steps (these will be similar to steps you completed in part 5.):

        20) Initialize `logfile`.

        21) Create an RDD from the file.

        22) View 10 lines of the data.

        23) Create an RDD containing only lines that are requests for `jpg` files.

        24) View 10 lines of the data.

        25) Chain the previous commands into a single command that counts the number of JPG requests.

        26) Create an RDD using the `map` function to return the length of each line of the log file.

        27) Create an RDD using the `map` and `split` functions to map an array of words for each line.

        28) Create an RDD containing only the IP addresses from each line.

        29) Use `foreach(println)` to output IP addresses.

        30) Save the list of IP addresses to a file in an HDFS directory named `loudacre/bigiplist` - use `saveAsTextFile`.

        31) Provide a screenshot of the contents of the `loudacre/bigiplist` folder.

            Note: You may see multiple files, including several `part-xxxxx` files, which are the files containing the output data. "Part" files are numbered because there may be results from multiple tasks running in the cluster (the tasks are part of your Spark job). Review the contents of one of the files to confirm that they were created correctly.