TP_Spark_Core

Axel Jacquin

January 17, 2017

One of the commons uses for Spark is doing data Extract/Transform/Load operations (ETL). Sometimes data is stored in line-oriented records, but sometimes the data is in a multi-line format that must be processed as a whole file. In this exercise you will practice working with line-oriented records.

1 Prepare your VM

1. Launch the VM if you haven't already done so, and then run the following command to boost performance by disabling services that are not needed for this class:

```
$ ~/scripts/sparkdev/training_setup_sparkdev.sh
```

2 Use the Spark Shell

2.1 Exploring the Spark Shell

2. In a terminal window, start the Scala Spark Shell:

```
$ spark-shell
```

You may get several INFO and WARNING messages, which you can disregard. If you don't see the scala> prompt after a few seconds, hit Enter a few times to clear the screen output.

3. Spark creates a SparkContext object for you called sc. Make sure the object exists:

```
scala>sc
```

4. Using command completion, you can see all the available SparkContext methods: type sc. (sc followed by a dot) and then the [TAB] key.

5. You can exit the shell by hitting Ctrl-D or typing exit

3 Use RDDs to Transform a Dataset

6. Review the simple text file we will be using by viewing (without editing) the file in a text editor. The file is located at:

/home/training/training_materials/sparkdev/data/frostroad.txt

7. Start a Spark Shell

8. Use sc.textFile to define an RDD to be created by reading in a simple test file.

9. Note that Spark has not yet read the file. **Why ?**.

10. Try counting the number of lines in the dataset.

10. Try executing the collect operation to display the data in the RDD. Note that this returns and displays the entire dataset. This is convenient for very small RDDs like this one, but be careful using collect for more typical large datasets.

12. Using command completion, you can see all the available transformations and operations you can perform on an RDD.

3.1 Explore the Loudacre web log files

You are going to use this directory:

```
$ cd ~/training_materials/sparkdev/data/weblogs.
```

13. Review one of the .log files in the directory. Note the format of the lines

14. In the previous example you used a local datafile. In the real world, you will almost always be working with data on the HDFS cluster instead. Create an HDFS directory /loudacre for the course, then copy the dataset to this HDFS directory.

Initially you will work with the log file from a single day. Later you will work with the full data set consisting of many days worth of logs.

15. In your Spark Shell, set a variable for the data file so you do not have to retype it each time.

```
scala> val logfile="/loudacre/weblogs/2013-09-15.log"
```

16. Create a RDD from the file.

17. Create an RDD containing only those lines that are requests for JPG files.

18. View the first 10 lines of the data.

18. Sometimes you do not need to store intermediate data in a variable, in which case you can combine the steps into a single line of code. Count the number of JPG requests using 1 line command.

19. Now try using the map function to define a new RDD. Start with a very simple map that returns the length of each line in the log file.

20. That's not very useful. Instead, try mapping to an array of words for each line.

21. Now that you know how map works, define a new RDD *ips* containing just the IP addresses from each line in the log file.

22. Although take and collect are useful ways to look at data in an RDD, their output is sometimes not very readable. Fortunately, though, they return arrays, which you can iterate through. Use *foreach* to print each IP address on one line.

23. Finally, save the list of IP addresses as a text file then take a look at this file.

24. As you did in the previous step, save a list of IP addresses, but this time, use the whole web log data set (weblogs/*) instead of a single day's log.

25. Use RDD transformations to create a dataset consisting of the IP address and corresponding user ID for each request for an HTML file. (Disregard requests for other file types). The user ID is the third field in each log file line. Display the data in the form ipaddress/userid