Fun with Spark Lab

Prepare Sample Data

For this assignment we are going to use two sample data sets. One is a text file version of the classic book “War and Peace”, the other is the same housing sales data from Week 2.

If you have not already done so, please put home_data.csv onto your VM and then also war_and_peace.txt. To get the data onto your VM you can either download to your local machine and then use the SCP command to copy the files e.g: “scp home_data.csv username@ip: (the colon is necessary)
Or use the wget command and pass in the link directly, similar to how you downloaded the HDP sandbox. Note that the “stylized” single quote marks in this doc might cause problems if you copy/paste directly into command line.

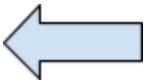
```
wget
‘https://drive.google.com/a/uw.edu/uc?authuser=2&id=0B0Ntj7VtxrluZG9xRkc0NmZ4Q0E&export=download’ -O war_and_peace.txt
```

```
wget
‘https://drive.google.com/a/uw.edu/uc?authuser=2&id=0B0Ntj7VtxrluN1dFWlRiY0pHNHM&export=download’ -O home_data.csv
```

If you would like to, you can instead add to HDFS and access from there. To copy home_data.csv to HDFS can either scp into sandbox and then use HDFS commands to copy to HDFS:

```
Scp home_data.csv root@localhost:
```
Then SSH into sandbox and run
```
Hadoop fs -put home_data.csv /tmp/home_data.csv
```

Or use the Ambari Web UI “Files View” (located same place you find the “Hive2 View” and can upload directly from local machine via web UI

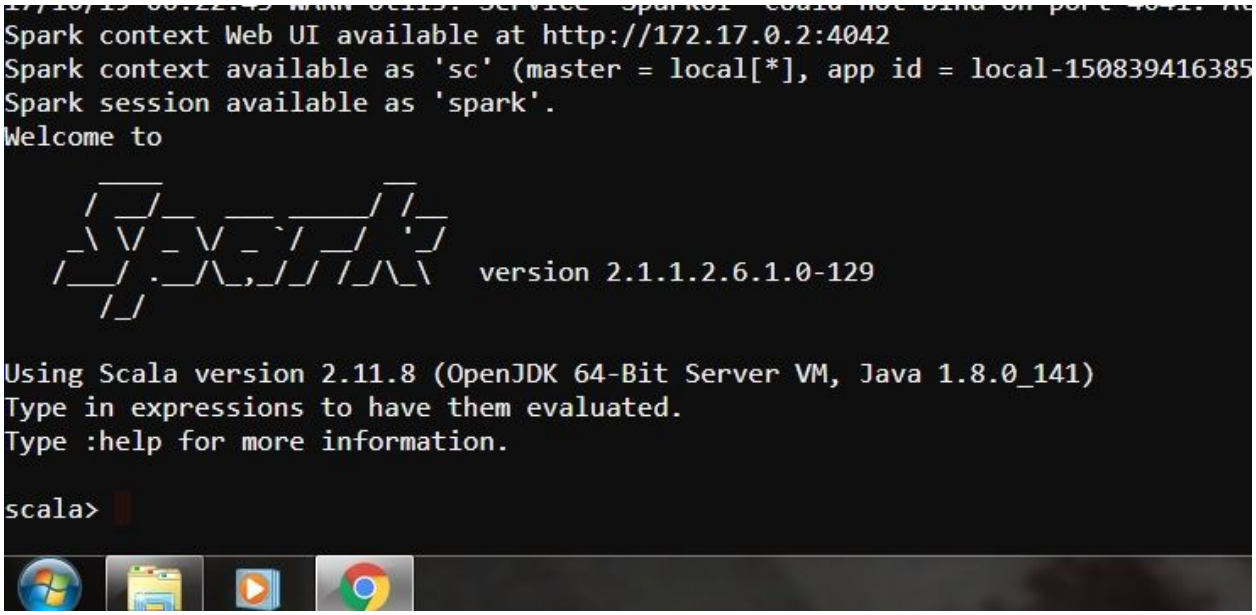


Run Spark Shell

From within the sandbox (after you ssh -p root@localhost) run the command "`spark-shell`" (or "`pyspark`" for a Python shell)

You can safely ignore warning about "SparkUI can't bind to port…"

If all goes well you should see a screen with some excellent ASCII art like this:



Congratulations, now you are ready to Spark!

To get some familiarity and make sure Spark is working properly, I'd recommend running through a few of the examples from the slides this week.

Assignment 3

Please answer the following questions using the sample data provided, providing code that you used. Either Scala or Python is acceptable. If you are unable to come up with code to answer a question please describe how you think you would solve the problem in a "sparkified" way based on what we have learned so far.

You will likely find the String method "split" in both Scala and Python, and "contains" in Scala ("in" in Python) useful in many of these exercises. This week's material shows some sample usage.

1) Which lines from the book war_and_peace.txt contain both words "war" and "peace"?
2) **Approximately** how many sentences are in the book war_and_peace.txt? HINT: Let's define a sentence as a sequence of text that ends with a period,and note that an approximation is ok.
3) Save a random sampling of 500 lines from war_and_peace.txt either to local or HDFS storage as text
4) From home_data.csv, how many houses sold were built prior to 1979?
5) From home_data.csv, how many houses sold had a lot size (in sq. ft) that was greater than triple the living space (in sq. ft)? Save this output to file either locally or on HDFS. Example: lot size 500 living size 100 would count (500 > 300) but 500/200 would not (500 < 600).

Bonus Exercise

Remember, bonus exercise is meant for those either with prior knowledge on the topic or the interest (and time) to do some research beyond what we covered in class and is optional.

Exercise 1:

Let's again use the zipcode list from Assignment 2, you can access via

```
wget
'https://drive.google.com/a/uw.edu/uc?authuser=0&id=0B0Ntj7VtxrluSXhiakdXLWx0N3c&export=download' -O wa_zipcodes.csv
```

How many homes were sold with a zipcode being defined as in "Seattle"?

Hint: how do you go about doing this in SQL?

Exercise 2:

Using home_data.csv create an RDD of key/value pairs where the key is the “id” of a row and the value is everything else.

For Example:
"7129300520","20141013T000000",221900,"3","1",1180,5650,"1",0,0,3,7,1180,0,1955,0,"98178",47.5112,-122.257,1340,5650

The key would be "7129300520" and the value would be "20141013T000000",221900,"3","1",1180,5650,"1",0,0,3,7,1180,0,1955,0,"98178",47.5112,-122.257,1340,5650

Hint: Most of the transformations we have done so far are simple one liners; But remember that you can use any arbitrary function that you define in your transformation, like we did with the square root example in the functional programming demo.