GCD Week 6 – Spark

Kevin Jetten

i318843

Inhoud

[Preparation 3](#_Toc496638007)

[Activity 1: Analyze a weblog 3](#_Toc496638008)

[Activity 1.1: how many hits are there to page ‘/assets/js/the-associates.js’? 3](#_Toc496638009)

[Activity 1.2: how many hits are there from ip address 10.99.99.186? 4](#_Toc496638010)

[Activity 1.3: Find the most popular file on the website, that is, whose path occurs most often in access\_log. The output should be: the file’s path and the number of times it occurs in the log. 4](#_Toc496638011)

[The code 4](#_Toc496638012)

[The result 5](#_Toc496638013)

[Activity 2: Spark Wordcount on Gutenberg 6](#_Toc496638014)

[Program Wordcount for the Gutenberg set. 6](#_Toc496638015)

[Show an output of the results 6](#_Toc496638016)

[Activity 3: Explore the Spark basics 7](#_Toc496638017)

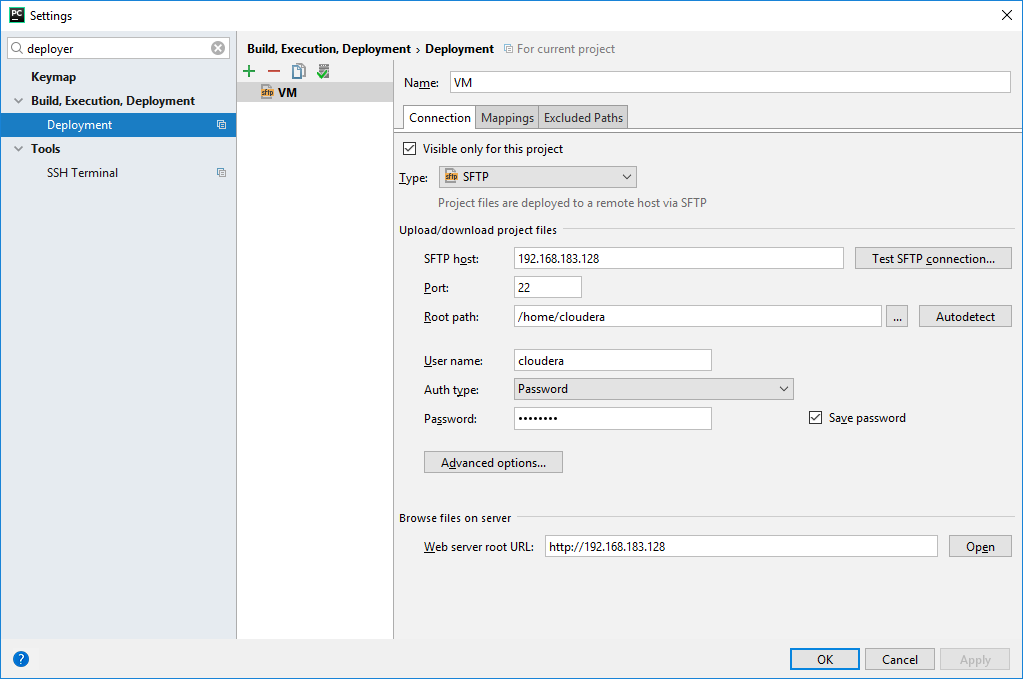
[RDD 7](#_Toc496638018)

[Transformations 7](#_Toc496638019)

[Actions 7](#_Toc496638020)

# Preparation

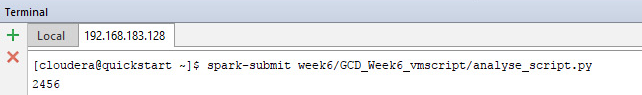
To avoid having to develop python scripts on the VM, I added the VM and spark as an interpreter to my Pycharm installed on my normal system.

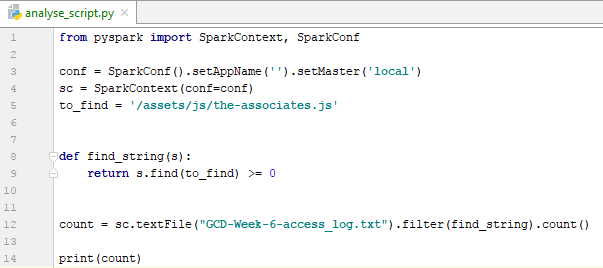


# Activity 1: Analyze a weblog

## Activity 1.1: how many hits are there to page ‘/assets/js/the-associates.js’?

**The answer is: 2456**

****

****

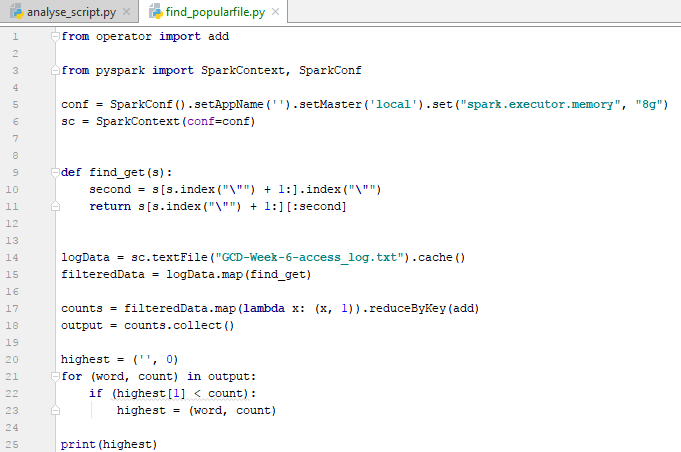
## Activity 1.2: how many hits are there from ip address 10.99.99.186?

**The answer is: 6**

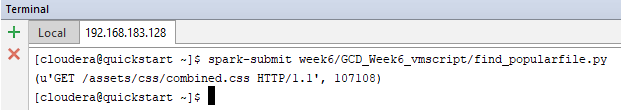
To avoid adding nearly the same screenshot again I won’t post this code because it is the same as the one above except I changed the value of ‘to\_find’ to ‘10.99.99.186’

## Activity 1.3: Find the most popular file on the website, that is, whose path occurs most often in access\_log. The output should be: the file’s path and the number of times it occurs in the log.

## The code



## The result

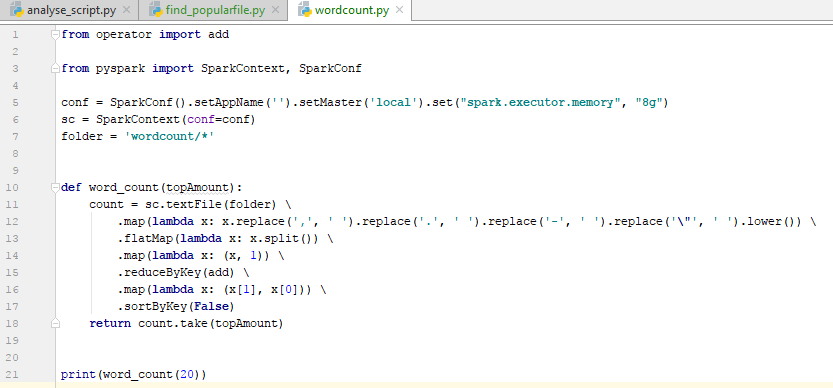


As it might be not clear. The following sentence had the most occurrences:

**/assets/css/combined.css – 107108 occurrences.**

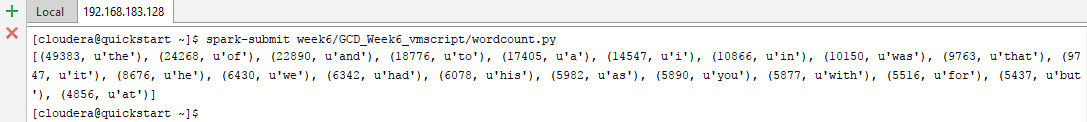
# Activity 2: Spark Wordcount on Gutenberg

## Program Wordcount for the Gutenberg set.

****

I replaced some punctuation so they aren’t counted as single-character words. The sort by key disable is done so it will sort the list by the values, by which we want it to do.

## Show an output of the results



The most used word is “**the**” with **49383 occurrences**

# Activity 3: Explore the Spark basics

## RDD

One of the most interesting parts about RDD is that the data is only created when the data is referenced. This is great for your performance because if you have a large dataset but you only use a small portion of it, not everything will be loaded in.

What I found very enlightening about RDD’s when creating the exercises above (especially 1.3) is that when you call .collect it will bring all the data to the driver program which for me resulted in a Java heap overflow. The usage of .collect should be avoided when working with very large datasets.

## Transformations

With transformations it is possible to create a new RDD with the filtered data from an existing one. These transformations are computed lazily which means the values the values are not evaluated until they are needed. I found this a little bit confusing in the beginning, because at sometimes I expected an error but it didn’t throw one because I did not do anything with the value yet.

Although in the beginning a bit confusing I understand that they can cause performance increase.

## Actions

Actions are a sort of trigger for the RDD to create its final computation. While working with spark I found that I used the following actions most often: Count, First and Collect. I don’t think I have to explain what these do because they kind of speak for themselves.

I like the flow that Spark created and it has a very logical flow once you understand the workings of each part.