# GCD Assignments week 6 – Apache Spark

In this workshop we will explore Apache Spark. For examples to illustrate the activities below see <http://spark.apache.org/examples.html>.

## Preparation

1. Change your Spark properties file, to remove unnecessary information during executing Spark jobs.

Type:

cd /usr/lib/spark/conf

sudo cp log4j.properties.template log4j.properties

sudo gedit log4j.properties

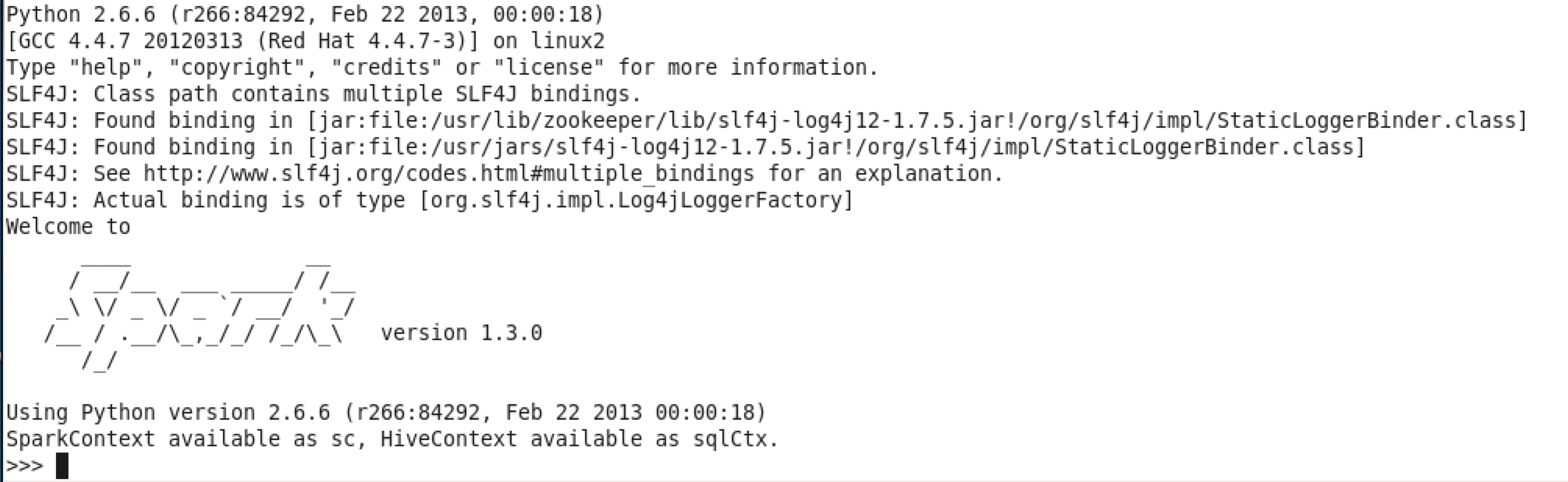
In the file log4j.properties, change:

log4j.rootCategory=INFO, console

to log4j.rootCategory=ERROR, console

2. Open PySpark (pyspark) in the folder /usr/lib/spark/bin

Result:



## Activity 1: Analyse a weblog

This exercise is derived from Project 3 from Intro to Hadoop and MapReduce from Udacity. In this exercise you will load the log file to HDFS and use Spark to make small overviews. Use Python or Scala.

Activity 2.0: download access\_log.txt (500MB): <http://www.fhict.nl/docent/downloads/BGDT/GCD-Week-6-access_log.txt>

Activity 2.1: how many hits are there to page ‘/assets/js/the-associates.js’?

Write Spark Python statements, or Scala statements.

>>> reg = re.compile('(?P<ipaddress>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}) - - \[(?P<dateandtime>\d{2}\/[a-zA-Z]{3}\/\d{4}:\d{2}:\d{2}:\d{2} (\+|\-)\d{4})\] ((\"(GET|POST) )(?P<url>.+) (HTTP\/1\.1\")) (?P<statuscode>\d{3}) (?P<bytessent>\d+)')

>>> counts = fl.map(lambda line: reg.match(line)).map(lambda group: group.group('url') if group else None).map(lambda url: (url, 1)).reduceByKey(lambda a,b:a+b)

>>> c = dict(counts.collect())

>>> c2[url\_to\_match]

217

Activity 2.2: how many hits are there from ip address 10.99.99.186?

>>> reg = re.compile('(?P<ipaddress>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}) - - \[(?P<dateandtime>\d{2}\/[a-zA-Z]{3}\/\d{4}:\d{2}:\d{2}:\d{2} (\+|\-)\d{4})\] ((\"(GET|POST) )(?P<url>.+) (HTTP\/1\.1\")) (?P<statuscode>\d{3}) (?P<bytessent>\d+)')

>>> counts\_ip = fl.map(lambda line: reg.match(line)).map(lambda group: group.group('ipaddress') if group else None).map(lambda ip: (ip, 1)).reduceByKey(lambda a,b:a+b)

>>> c = dict(counts\_ip.collect())

>>> c['10.99.99.186']

4

Activity 2.3: 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.

>>> counts = fl.map(lambda line: reg.match(line)).map(lambda group: group.group('url') if group else None).map(lambda url: (url, 1)).reduceByKey(lambda a,b:a+b).sortBy(lambda pair: -pair[1])

>>> c = counts.collect()

>>> c[:10]

[(None, 637124), (u'/assets/css/combined.css', 103708), (u'/assets/js/javascript\_combined.js', 99176), (u'/', 92369), (u'/assets/img/home-logo.png', 84122), (u'/assets/css/printstyles.css', 83268), (u'/images/filmpics/0000/3695/Pelican\_Blood\_2D\_Pack.jpg', 83105), (u'/favicon.ico', 63228), (u'/robots.txt', 38732), (u'/images/filmpics/0000/3139/SBX476\_Vanquisher\_2d.jpg', 35633)]

None is because of the wrong regex. It should be tweaked to better fit the log format

Put the code and the answers in your portfolio.

## Activity 2: Spark Wordcount on Gutenberg

1. Put the Gutenberg files from the previous week in an appropriate folder in HDFS

2. Open pyspark

3. Program Wordcount for the Gutenberg set.

Tips:

* Start with one file, then try to expand the wordcount to all files.
* Google how to use wordcount in Spark.

4. Show an output of the result in your portfolio.

The output (as well as the input) can be found in HDFS.

>>> def rem\_punc(inp):

... return inp.encode('utf-8').translate(None, string.punctuation).decode('utf-8')

...

>>> count = tf.flatMap(lambda line: [word for word in rem\_punc(line).lower().split(' ') if word and word not in stopwords]).map(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b).sortBy(lambda pair: -pair[1])

>>> c = count.collect()

>>> c[:10]

[(u'a', 344566), (u'one', 49502), (u'said', 49272), (u'now', 31991), (u'will', 28301), (u'us', 28247), (u'time', 26309), (u'like', 26102), (u'project', 25751), (u'can', 24689)]

>>>

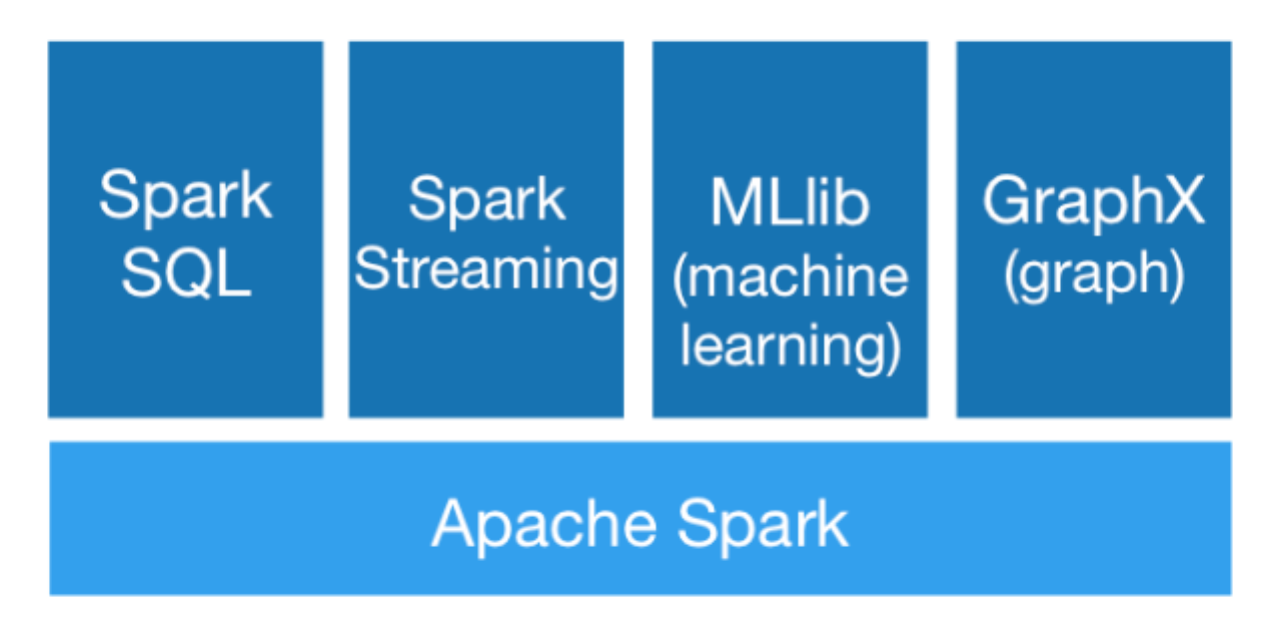
## Activity 3: Explore the Spark basics

Make your own statements to show useful usage of the following topics.

* RDD
* Transformations
* Actions

Put the results in your portfolio.

## Activity 4: Explore the Spark framework

Study the documentation and examples of the Spark framework (SQL, Streaming, MLib, GraphX).

This does not have to be documented in your portfolio.