# 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

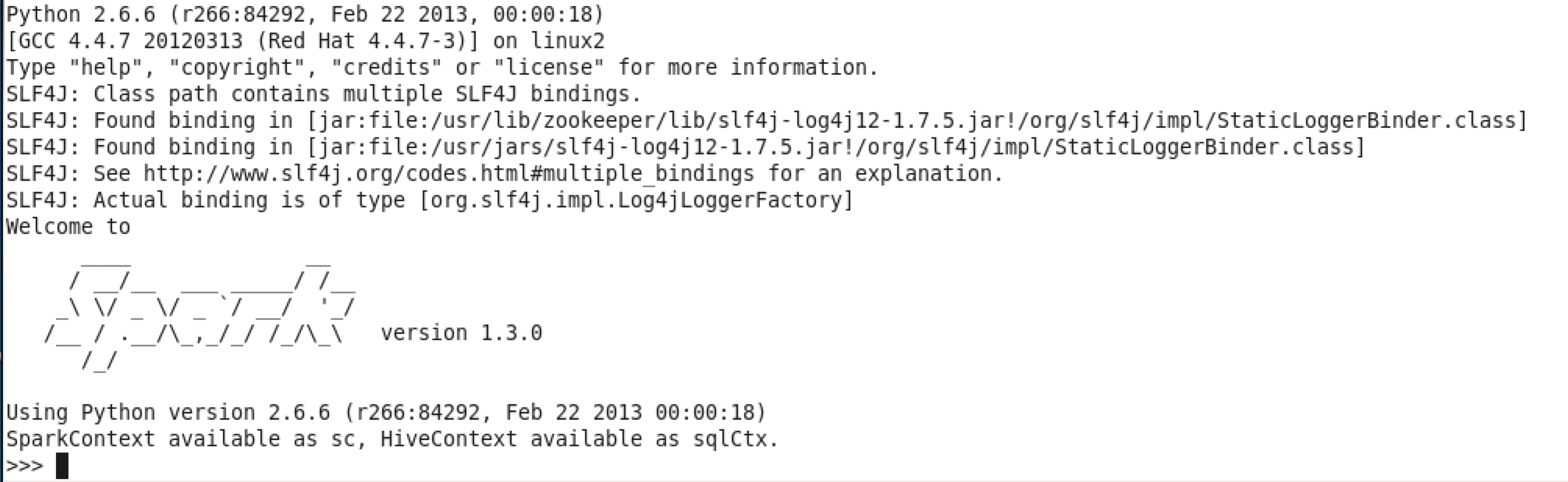
In the file log4j.properties, change:

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.

**log = sc.textFile("hdfs:///user/cloudera/access-log.txt")**

**f = log.filter(lambda x: "/assets/js/the-associates.js" in x)**

**f.count()**

**> 2456**

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

**ip = log.filter(lambda x: "10.99.99.186" in x)**

**ip.count()**

**> 6**

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.

Put the code and the answers in your portfolio.

**import re**

**log = sc.textFile('hdfs:///user/cloudera/access-log.txt')**

**f = log \**

**.map(lambda x: re.findall(r'\"\w+\ ([\w\/\.]+)\ HTTP\/[\d.]+\"', x, 0)) \**

**.filter(lambda x: x != []) \**

**.map(lambda x: (x[0], 1)) \**

**.reduceByKey(lambda acc, x: acc + x)**

**> [(u'/assets/css/combined.css', 117348), (u'/assets/js/javascript\_combined.js', 106818), (u'/', 99303), (u'/assets/css/printstyles.css', 93158), (u'/images/filmpics/0000/3695/Pelican\_Blood\_2D\_Pack.jpg', 91933), (u'/favicon.ico', 66831), (u'/robots.txt', 51975), (u'/images/filmpics/0000/3139/SBX476\_Vanquisher\_2d.jpg', 39591), (u'/assets/img/play\_icon.png', 34151), (u'/images/filmmediablock/290/Harpoon\_2d.JPG', 32533), (u'/assets/img/x.gif', 29377), (u'/images/filmpics/0000/1421/RagingPhoenix\_2DSleeve.jpeg', 29243), (u'/search/', 23055), (u'/images/filmmediablock/293/NewsMakers\_2DBluRay.jpeg', 13466), (u'/images/clientlogos/0000/0010/Manga.jpg', 11503), (u'/images/filmmediablock/39/bluray\_pontypool2d\_new.jpg', 11318), (u'/images/frontpagepics/0000/0077/Risenfront.jpg', 11197), (u'/database/index.php', 9513), (u'/images/clientlogos/0000/0019/Momentum\_logo.jpg', 8411), (u'/assets/css/ie.css', 8279)]**

## 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.

**import re, string**

**text\_file = sc.textFile('hdfs:///user/cloudera/1013.txt')**

**text\_file.take(5)**

**punc = '!"#$%&\'()\*+,./:;<=>?@[\\]^\_`{|}~'**

**def uni\_to\_clean\_str(x):**

**converted = x.encode('utf-8')**

**lowercased\_str = converted.lower()**

**# for more difficult cases use re.split(' A|B')**

**lowercased\_str = lowercased\_str.replace('--',' ')**

**clean\_str = lowercased\_str.translate(None, punc) #Change 1**

**return clean\_str**

**r = text\_file.flatMap(lambda x: uni\_to\_clean\_str(x).split()) \**

**.map(lambda x: (x, 1)) \**

**.reduceByKey(lambda x,y: x + y) \**

**.sortBy(lambda x: x[1], False)**

**r.take(10)**

**> [('the', 3943), ('and', 2748), ('of', 2510), ('i', 2133), ('a', 1852), ('to', 1676), ('that', 1085), ('in', 1053), ('it', 1045), ('was', 931)]**

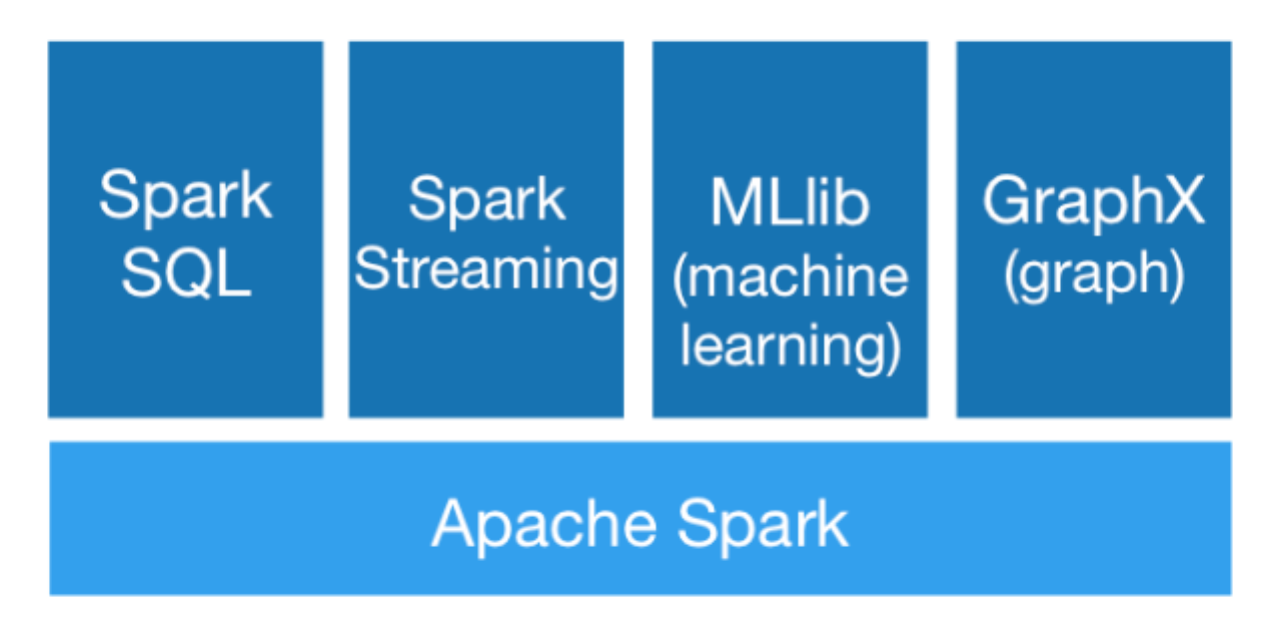
## Activity 3: Explore the Spark basics

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

* RDD
  + A Resilient Distributed Dataset (RDD), the basic abstraction in Spark. Represents an immutable, partitioned collection of elements that can be operated on in parallel. Basically, it is an immutable collection to which you can apply transformations which in turn returns a new RDD. Bascially the same as in Haskell by default.
* Transformations
  + Transformations are functions you can apply on an RDD to generate a new dataset.
* Actions
  + An action is a function you can apply on an RDD to return a single value or a subset of the original data.

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.