Universität Hamburg
Department Informatik
Wissenschaftliches Rechnen
Dr. J. Kunkel

Exercise Sheet 11
Big Data Analytics
WiSe 2017/2018
Due Date: 2018-01-26, 420 P

Please document your code, without sufficent documentation you won't recive any points.

1 Basics with Spark (Python) (60 P)

This exercise will start with a few basics using Spark:

- Create an RDD using `/user/bigdata/enwiki-10k.csv`
- Use RDD operations to compute the number of articles (absolute and relative to the total number) which contain the word computer.
- Save the articles that contain the word into a file on HDFS.
- Compute the number of articles containing the word computer again using only accumulators.
- Now use the methods for data frames for computing the number of articles containing the word computer.

Perform this task first using Sparks local mode. Then use spark-submit and YARN to distribute the tasks across multiple nodes. Document the arguments to start Spark.

1.1 Hints

Further documentation about the operations from RDDs are provided at: `http://spark.apache.org/docs/latest/`. You may also use $help(object)$ in Python to retrieve the manual for any object.

Submission:

`1-spark.py`	Your Python program performing all the tasks mentioned above.
`1-spark-call.txt`	The calls to start Spark and the output of your script.

2 Wikipedia Article Distance with Spark (Python) (180 P)

In this task, we compute the distance between any two Wikipedia articles. Therefore, design and implement a distance metric d based on the word frequencies of the articles. You may also extend this metric. The distance $d(A, B)$ between two articles A and B should be between 0 and 1.
Then we want to filter similar articles: A user can define a threshold t (between 0 and 1); for each article, all articles that have a distance below this threshold shall be returned ordered by the distance. That means, the program returns a CSV file with:

```
[wiki article title], [list of similar articles as tuples (title, score)]
```

Document the runtime of your application on a subset of the Wikipedia data set `/user/bigdata/enwiki-10k.csv`. Include a subset of the resulting CSV file.

2.1 Hints

A code skeleton can be found in `/home/bigdata/11/wikipedia-distanz.py`.
For debugging purpose it may be necessary to document the expected format of the tupels and inspect intermediate results.

Submission:

`2-wikipedia-distanz.py`	The Python program.
`2-wikipedia-distanz.txt`	Documentation of the invocation of Spark, the runtime measurements and a few example outputs.

3 Clustering with Spark (Python) (120 P)

Similar to the previous exercise, we use Spark to identify similar articles in the Wikipedia. But now we will use the KMeans algorithm of Sparks MLlib.
Use the data file `/user/bigdata/enwiki-10k.csv`.
Evaluate several numbers of clusters and use the mean distance of an article to the cluster center as error metric. Analyze the different cluster centers and sort the articles of a cluster based on the distance to the center. Document the runtime.

3.1 Hints

Test your script using the interactive pyspark shell. Scripts started via `spark-submit` may be difficult to debug. Investigate the job details on the Spark history server by forwarding the port.
Further information about k-means: `https://spark.apache.org/docs/1.6.1/mllib-clustering.html#k-means`.

Submission:

`3-kmeans.py`	The Spark program.
`3-analyse.pdf`	Runtime and example results.

4 Performance Analysis (Theory) (60 P)

In this task, we analyze the programs of two previously conducted exercises that operate on larger data sets that perform similar operations. You may also rewrite the task to be very simple (like subsetting only some data). If possible, use two different technologies such as Spark and Pig, or Hive and HBase.
You may have to rerun the original application measuring the runtime.
Measure the runtime for the input with only one input tuple, then use an input data volume of, e.g., 100 MiB and 1000 MiB. Delete your input files afterwards to free space.
Compute the latency (startup time) of the program and measure the throughput in (Tupels/s)

Estimate/compute the actually used CPU, network and E/A throughput. Which performance did you expect on one/all available nodes?

4.1 Hints

The Abu nodes utilize Gigabit Ethernet and are equipped with one HDD per node providing roughly 100 MiB/s. A node hosts 4 sockets each equipped with a 12 core AMD CPU @2.6 GHz.

Submission:

`4-analysis.pdf`	Your performance analysis.