Universität Heidelberg
Wintersemester 18/19

Group Parallel and Distributed Systems (PVS)
Artur Andrzejak, Diego Costa

Problem Set 3 for lecture Distributed Systems I (IVS1)

Due: 13.11.2018, 14:00 Uhr

Exercise 1 (3 Points)

Read the introductory material on MapReduce framework[1] and understand how map and reduce operations work. Solve the following problems in an efficient pseudocode implementation using only map and reduce operations (note that operations in map-reduce may differ from Spark map-reduce operations). In problems 1 to 4, the input is a huge set of integer numbers. You may use more than one map-reduce job when necessary.

1. Output the biggest of the input numbers.
2. Output the geometric mean of the input numbers.
3. Output the input set of numbers, but without duplicates.
4. Output the size of the input set (ignoring multiplicity)
5. (1 Point) Output the frequencies of a word given from a huge text file (Word Count).

Exercise 2 (3 Points)

Read the paper that introduced the concept of Resilient Distributed Datasets (RDD)[2] and answer the following questions:

1. What was the state-of-the-art of distributed computing frameworks? What particular application feature was not properly handled by existing computing frameworks?
2. What are the main advantages and limitations of RDDs compared against Distributed Shared Memory?
3. How are RDDs and its dependencies represented in the Spark system?

Exercise 3 (4 Points)

Familiarize yourself with Spark. You can use either Python or Java.

1. Describe each of the following transformations: `join()`, `sort()`, `groupby()`. In particular, what is in each case the type of input and type of output? Give for each one an example ("on paper") with a simple input and output.
2. Give three own programming examples of your choice for transformations (but not for `map()` or `filter()`) and three examples for actions (again, of your choice). Write executable code and test its correctness (either single program or several ones). To generate initial RDDs you can use code from the Spark documentation. Submit as solution the source code and results of program runs.

[1] https://www.tutorialspoint.com/map_reduce/map_reduce_introduction.htm
[2] https://cs.stanford.edu/~matei/papers/2012/nsdi_spark.pdf

Exercise 4 (6 Points)

Follow the tutorial of Machine-Learning with PySpark[3] from Data Camp. This tutorial covers several topics relevant to PySpark programming, from PySpark installation and configuration to the use of simple Machine Learning methods in Spark. For this exercise we will be focusing on the following topics:

- Installing Apache Spark. Extra steps are required to use Spark in Windows systems[4].
- The Data: Loading and Exploring your Data
- Data Exploration

Follow each step of the tutorial until Data Exploration (inclusive), and submit all your code into Moodle. You might use Jupyter Notebook (as stated in the tutorial) or plain Python code. After concluding the targetted topics in the tutorial, answer the following questions:

1. Compare the Spark Shell environment provided by Spark against the conventional python Spark Application development. What tailored features Spark-shell provides and when would you recommend the usage of such interface?
2. Explain why `collect` action should be avoided on large datasets. What other actions can be used to inspect the contents of your RDDs?
3. In the context of Spark programming define Dataframes. What are the advantages of using DataFrames over conventional RDDs?
4. Using only SQL operations (select, groupby) calculate:
   - The latitude of the northernmost household from California
   - The most common households size in the dataset (given by „households“ column)
   - The highest ratio of bedroom per population in the dataset.

[3] https://www.datacamp.com/community/tutorials/apache-spark-tutorial-machine-learning
[4] https://medium.com/@dvainrub/how-to-install-apache-spark-2-x-in-your-pc-e2047246ffc3