BINFOR-105: Cloud Computing (Winter Term 2019/20)
Dr. Vinu Venugopal
**Exercise Sheet #1 – PART-1 demos on Oct. 15, 2019 and PART-2 demos on Oct. 22, 2019**

*Send your solutions before the due date per email to `cloudcomputing@uni.lu` in a single zip file (containing just the java files) using your group id as the file name. Include some brief instructions on how to run your solution to each problem in a file called* `problem_X.txt`. *All solutions may be submitted in groups of up to 3 students.*

## I. MapReduce & Apache Hadoop

## PART-1

### Wordcount on AWS Cloud — **10 Points**

**Problem 1.** Study the impact of the number of *reduce tasks* on the runtime in a MapReduce job. (The number of reduce tasks for a job is set via `Job.setNumReduceTasks(int)`.)

- Consider the `customwordcount.java` given on `moodle.uni.lu` as your application and `Wikipedia-EN-20120601_ARTICLES.tar.gz` as the input document collection.
- Deploy your jar file(s) and dataset on an AWS EC2 instance and perform your experiments there.
- Prepare a plot with X-axis as the count of reduce tasks (say, from 1 to 6) and Y-axis as the runtime.

**Note.** You may either create a new EC2 instance for each member of your student group individually or share an instance among all members of your group. Please refer to our "GettingStarted-AWS" guide on Moodle for instructions on how to access your AWS account and create an EC2 instance. Note that you should always develop your code locally before uploading it to the AWS (refer also to our "GettingStarted" guide for instructions on setting up your local Hadoop environment).

Upload dataset file from your local directory to your EC2 instance by issuing the following commands:

- `scp -i <path-to-your/awskey.pem> </path-to-your/Wikipedia-EN-20120601_ARTICLES.tar.gz> <your-EC2-instance-DNS-name-or-IP-address>:~/`

### Wordcount using YAGO dataset — **12 Points**

**Problem 2.** Modify the `wordcount.java` example given on `moodle.uni.lu` to find the number of occurrences of each *predicate* in the entire yago dataset.

- Identify the top three frequently occurring predicates from the output file of your MapReduce job.

**Note.** A downloadable version of the entire Yago dataset is given on the moodle page. The Yago dataset contains facts in the form of triples and each triple has three parts: *subject*, *predicate* and *object*. Note that in this exercise you are asked to find only the occurrences of *predicates.*

**Problem 3.** Implement *distributed grep operation.*

- Modify the `wordcount2.java` example given on the moodle to perform a *distributed grep operation* where the patterns to be filtered should be read from an external file (say, `patterns.txt`).
- Consider the entire Yago dataset as input and filter all those triples containing a predicate, whose frequency in the overall dataset is among the top three.
- Follow the output from Problem 1 to find out the predicates with the highest three frequencies and list them in the pattern.txt file for giving as input to your MapReduce job.

## PART-2

### Processing Joins via Hadoop **18 Points**

**Problem 4.** Implement *memory-backed join.*

- Utilising the distributed grep application that you developed in Problem 3, filter those triples with predicate-name `livesIn` into a file. You may rename the output file to `set1.txt`.
- Perform a *memory-backed join* by sharing `set1.txt` with all the Map Workers to find out all the triples in the entire Yago dataset which could make a subject-to-subject join with at least one triple in `set1.txt`.

**Problem 5.** Implement a *reduce-side join* to find out all the subjects (`x`) and objects (`y` and `z`) matching the following pattern from the Yago dataset.

```
?x <hasGivenName> ?y.  ?x <livesIn> ?z.
```