T–79.5308 **Autumn 2013**

Scalable Cloud Computing P

Home Assignment 1 - Deadline: 5th of November 2013 at 10:15 (strict deadline!)

Return your answer via email to `T-79.5308@cs.hut.fi` with "Home Assignment 1, [student number]" as the subject. Attach to the email a zip or tar file, packaging all your answer files inside the directory "`assignment-1`". A template .tar file for the home assignment 1 can be found from:
`https://noppa.aalto.fi/noppa/kurssi/t-79.5308/harjoitustyot`
under Assignment 1. Download it, unpack the files, and modify them to contain your answers. When you are done, pack the files to a tar or zip file. On unix workstations this can be done with the command "`tar cvf assignment-1-answer.tar assignment-1`", when you have an existing directory "`assignment-1`" with your answer files in it. Then send the package to the course email address with the subject mentioned above. Submissions that arrive late are not graded! Be sure to send your answer in time, and remember that it may take some time for email messages to arrive. The course assistant will send you a confirmation once he receives your submission.
The home exercises are personal, no group work allowed! There are two rounds of home exercises of 10 points each. To pass the home exercises ≥ 10 points are needed and ≥ 16 points gives a $+1$ to the exam grade (no effect to exam grades 0 or 5).
The home assignments require you to have a working Apache Hadoop installation, please see the following Noppa page on how to have Hadoop set up:
`https://noppa.aalto.fi/noppa/kurssi/t-79.5308/practical_matters`

1. a) The directory "`question-1`" contains the "`WordCount.java`" word count example also used in the Tutorials and Lectures. Please run the WordCount example on the file "`vanrikki-stool.txt`" placing the log output generated by Hadoop to the console to the file "`question-1-a-log.txt`" (we are expecting similar output as in Lecture 3, slides 20-26). Also copy the output directory generated by the Hadoop WordCount job from HDFS to subdirectory "`output-stool-a`" of the "`question-1`" directory. (3p)

   b) Do a small modification of the "`WordCount.java`" word count example into a new returned Java file "`TopCount.java`". The only modification is that the Reducer should only output the words which appear at least 100 times in the input text. Please run your TopCount job on the file "`vanrikki-stool.txt`" placing the log output generated by Hadoop to the console to the file "`question-1-b-log.txt`" (Again we are expecting similar output as in Lec-

ture 3, slides 20-26). Also copy the output directory generated by the Hadoop TopCount job from HDFS to subdirectory "`output-stool-b`" of the "`question-1`" directory. (3p)

2. In this example use the subdirectory "`question-2`". It contains an input file "`helmet-most-wanted-2011-05-23.txt`", where each row is a library booking record with several optional fields. The file in question contains the most (over) booked items from the Libraries in the greater Helsinki area as of 2011-05-23. Your task is to create a Hadoop program in "`AuthorUrl.java`" returned in the "`question-2`" directory. The program should do the following: The Mapper should parse each one of the Library booking records (the format should be self-explanatory for author and url fields) and should output `(author,url)` pairs for all booking records that contain both `author` and `url` fields. (Hint: Not all records will contain both fields!) The Reducer should just output the `author` as the key and as a value a string consisting of urls of the entries affiliated with that author in a comma separated format:
`"http://foo.bar.com","http://abc.def.fi"`

   In addition to the "`AuthorUrl.java`" file, also return a run log similar to the ones created in question 1 to the file "`question-2-log.txt`". Also copy the output directory generated by the Hadoop AuthorUrl job from HDFS to subdirectory "`output-author-url-index`" of the "`question-2`" directory.

   (4p)

P.S. For licence of the data file of question 2, and more open data from Helmet Library, see: `http://data.kirjastot.fi/`.