Techniques for large-scale data (DIT871/DAT345)

CSE, Gothenburg University | Chalmers, Period 4, 2018
Graham Kemp, Alexander Schliep (alexander.schliep@cse.gu.se)

Problem set 2 from April 25, 2018 · Due on May 3, 2018

Problem 1 (7pt)**.** Implement the computation of summary statistics and a histogram using MRJob as a parallel Map-Reduce program.

Input: The input is a file containing records `<id>\t<group>\t<value>` on each line. Here `<id>` and `<group>` are integer keys, and `<value>` is a scalar real-valued variable.

Output: mean and standard deviation of the values, their minimal and maximal value, as well as the counts necessary for producing a histogram; i.e. for how many lines does the `value` fall into a specific bin. (Note: you do not need to produce a graphic displaying the histogram).

Specifically address the following tasks:

a. Implement the Map-Reduce program for the summary statistics and histogram taking input and producing output as described above. Use `mrjob-summary-statistics-data.py`, which you find on Canvas, to generate data.

b. Benchmark your solution for the previous subproblem with a variable number of cores (e.g., $1, 2, 4$) and various input sizes (e.g., $n, 2n, 4n, \ldots$...; you get to choose n) to estimate your implementations empirical computational complexity (the function relating running time to input size). Produce a plot including a fit and an argument what type of function (linear, log-linear, polynomial,...) you choose to describe the relation.

c. Describe and explain problems which can arise if a small value of n is chosen in the previous subproblem.

d. Modify the code from the first subproblem to restrict the analysis to records for which `<group>` is an element of a set specified on the command line.

e. (Optional; extra credit) Add computation of the median.

Problem 2 (3pt)**.** Your work for a manufacturing company which has extensive sensor network for all the machinery used in the plant. There is about 1,300 GB of data generated every day. The analysis workflow, which is started daily at 20:00 after production stops for the day, keeps increasing in complexity, and now runs over 20 hours. In the near future, it will be inevitable that the daily analysis will not be able to keep up with the daily influx of data.

You propose scenario (a), buying new compute servers, requiring a non-trivial capital investment. Your boss heard scenario (b), "The Cloud", mentioned on the news and thinks it would be a great idea to assure that a previous' day analysis can be reviewed at 8:00 the next working data.

What are the *most* relevant factors in analyzing your total running times for the analysis workflow in the two scenarios? Under which conditions are scenarios (a) respectively (b) *feasible* with respect to hitting the 8:00 deadline?

Note: Please keep your analysis concise and use mathematical formulas where possible and appropriate. A solution to the problem should not take more than half a pager.

Submission: Please submit a PDF with your answers to the questions, and, additionally, Python code. Jupyter notebooks are only accepted in addition to a PDF. We will not accept files in other formats.