Exploration and Presentation - Assignment 3

Anders Jacobsen, Dima Karaush

April 14, 2021

Contents

Intro

We are using the project Letter Frequencies downloaded from `https://github.com/CPHBusinessSoftUFO/letterfrequencies` in this paper.

1 Task 1

1. Find a point in your program that can be optimized (for speed), for example by using a profiler
2. Make a measurement of the point to optimize, for example by running a number of times, and calculating the mean and standard deviation (see the paper from Sestoft)
3. If you work on the letter frequencies program, make it at least 50% faster

1.1 Find a point in your program that can be optimized

We did not use a profiler. The reason is that we could not get the profiler to run in Visual Studio Code. We have however located multiple points in the program that can be optimized. Since we are reading a relatively big text-file, we believe it is here we can optimize the most. We took multiple steps to make the program faster as shown in the list below:

1. Upgrading the FileReader to a BufferedReader.
2. Replacing the "HashMap<Integer, Long>" with a "int[]"
3. Limiting the while loop to only saving letters A-Z

We've decided to measure the two methods in the program "tallyChars()" and "print_tally()". We measure them together as a sequence to see the difference execution time on all our optimizations.

1.2 Make a measurement of the point to optimize

To make measurements we first had to create a timer that supports our measurements. The timer class can be found in "letterfrequencies/src/main/java/cphbusiness/ufo/letterfrequencies/Timer.java".

Procedure

We've decided to make multiple measurements in order to create a realistic image of the run-times of both the optimized and non-optimized classes. The way we do this is by creating a loop and then run the two methods for a specified amount of iterations and writing each iterations run-time to a CSV file for the analysis.

Results

1.3 Make it at least 50% faster