Assignment 2 UFO

Andreas Guldborg Heick, Mohammad Hariri, Rasmus Jarnborg Friis

November 20, 2020

0.1 Documentation of the current performance

Original times in milliseconds:
[202, 130, 90, 123, 102, 104, 113, 97, 116, 113, 113, 100, 111, 110, 84, 90, 84, 82, 83, 84]

0.2 An explanation of the bottlenecks in the program

Reader is slower than BufferedReader. Reader is a class that helps to read data from a file. BufferReader is a class that helps to read text from a character based input stream. Thus, this is the main difference between FileReader and BufferedReader.

0.3 A hypothesis of what is causing the problem

Our hypothesis as to why we have performance issues, is that the Reader that is used is not the most efficient way of reading from a file, the way we need to in this program.

0.4 A changed program which is improved to solve the problem

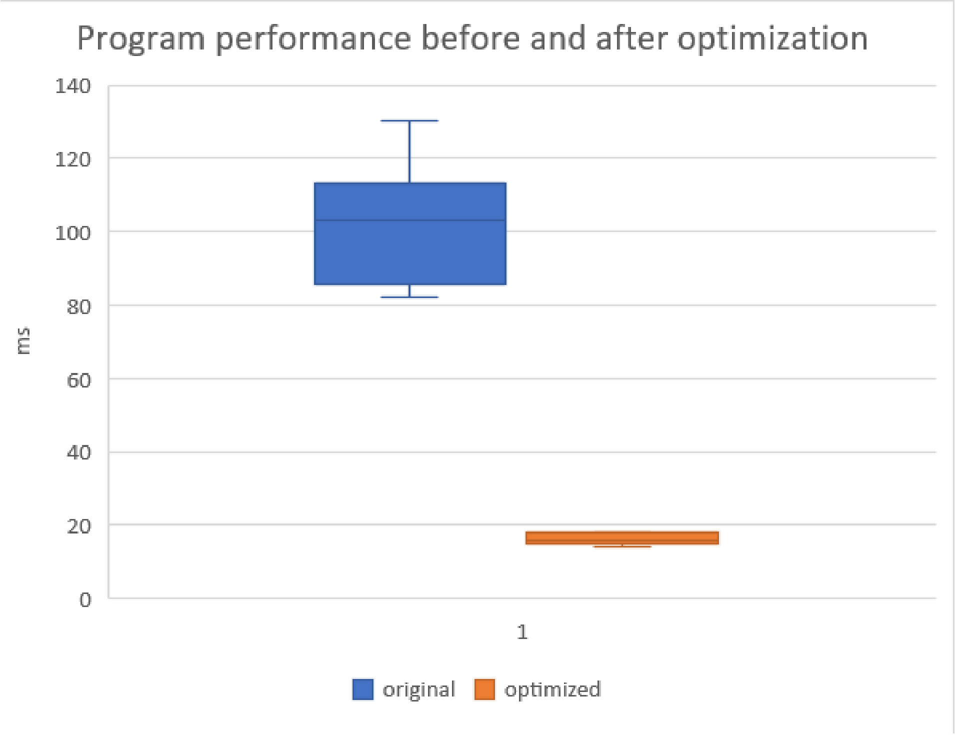
We implemented the BufferedReader like this:

```
private static void tallyCharsOptimizedWithBufferedReader
(Reader reader, Map<Integer, Long> freq){
        int[] charCodeArray = new int[128];
        try (BufferedReader buffer = new BufferedReader(reader)) {
            int b;
            while ((b = buffer.read()) != -1) {
                charCodeArray[b]++;
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        for (int i = 0; i < charCodeArray.length; i++) {
            freq.put(i, (long) charCodeArray[i]);
        }
    }
```

0.5 Documentation of the new performance

Optimized times in milliseconds:
[98, 28, 17, 16, 14, 14, 14, 15, 18, 18, 15, 15, 16, 16, 16, 16, 27, 17, 14, 18]

0.6 Box diagram over performance gain from optimization


Program performance before and after optimization
ms
original
optimized