Optimization of Letter Frequencies - Take Two

Anders Jacobsen, Dima Karaush

April 20, 2021

Contents

1 Test Enviroment **1**

2 Introduction **1**

3 Tools for Benchmarking **1**

4 Benchmark for original **2**

5 Optimization **2**

6 Benchmark for Optimized **2**

7 Benchmark Comparison **2**

8 Conclusion **2**

1 Test Enviroment

System information Java information IDE info?

2 Introduction

what are we working with what approach do we take on the task what results are we expecting?

3 Tools for Benchmarking

Timer klassen Benchmark timeren Lavet metoder statiske og kalder dem ved benchmarking

4 Benchmark for original

What is going on in the program As it is visible on Listing 1, the original letter frequencies program uses a FileReader and a HashMap<Integer, Long> to read the file and safe the letter frequencies. It manipulates the Hashmap through the static tallyChars method. Then the program uses the other static method, print_tally, to show the letters alligned with their frequency in the file.

```
public static void main(String[] args) throws FileNotFoundException
    , IOException {

    String filePath = "src/main/resources/FoundationSeries.txt";

    Reader reader = new FileReader(filePath);
    Map<Integer, Long> freq = new HashMap<>();

    tallyChars(reader, freq);

    print_tally(freq);
}
```

Listing 1: The main method of the original Letter Frequncies program without optimizations

code snippets of benchmark points How did we make the benchmark method What can be optimized

5 Optimization

What changes do we make to the program Show snippets of changes

6 Benchmark for Optimized

show the results of the benchmark

7 Benchmark Comparison

Compare the original vs the optimized times show tons of graphs and charts

8 Conclusion

Sum it up talk about most remarkable changes.