

Exploration and Presentation - Assignment 3

April 11, 2021

benjamin kongshaug `cph-bk131@cphbusiness.dk`

Amanda Hansen `cph-ah433@cphbusiness.dk`

Amalie Landt `cph-sl307@cphbusiness.dk`

Contents

1	Task 1	3
1.1	Find a point in your program that can be optimized (for speed), for example by using a profiler	3
1.2	A hypothesis of what causes the problem	3
1.3	A changed program with better performance	4
1.4	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)	4
1.5	make the program 50 procent more efficient	5

1 Task 1

1.1 Find a point in your program that can be optimized (for speed), for example by using a profiler

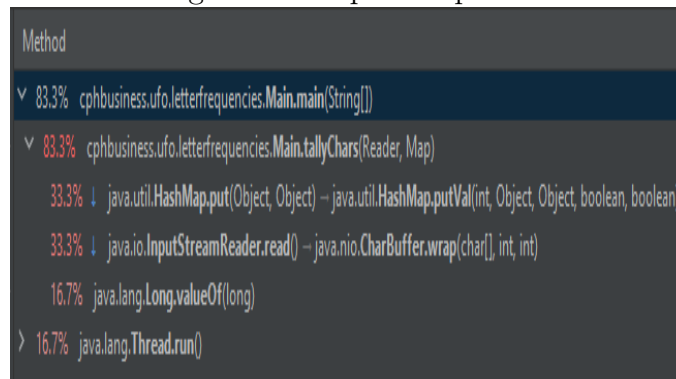
Før optimering tager programmet: Time elapsed (millisec): 389

Først og fremmest kan vi se 83 procent af tiden befinder programmet sig i metoden tallychars og at det er her vi skal optimere

Sekundært ser vi en ligelig fordeling på 33,3 procent af tiden brugt på at lægge input i hashMap og på at læse fra input reader filen

Vi kan optimere ved at finde en alternativ måde at indlæse fra filen og ved at gemme mængden af hver bogstav på en mere effektiv måde

Figure 1: Output fra profiler



1.2 A hypothesis of what causes the problem

Reader reader = new FileReader(fileName); En reader læser data fra en fil, en linje af gangen når man bruger reader.read() hvilket ikke er særlig effektivt når man har af gøre med en stor fil

1.3 A changed program with better performance

ved at anvende en `BufferedInputStream` istedet for en reader kan vi gøre indlæsningen af dataen hurtigere, da vores `.read` metode tager mod data fra bufferen istedet for filen og det er hurtigere at indlæse data fra en buffer.

Vi forbedre programmet ved at anvende en Buffered input stream som illustreret nedenfor:

Figure 2: Output fra profiler

```
String fileName = "/Users/benja/Documents/Downloads/letterfrequencies/src/main/resources/FoundationSeries.txt";  
// Reader reader = new FileReader(fileName);  
  
File f = new File(fileName);  
FileInputStream fis = new FileInputStream(f);  
BufferedInputStream reader = new BufferedInputStream(fis);
```

1.4 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)

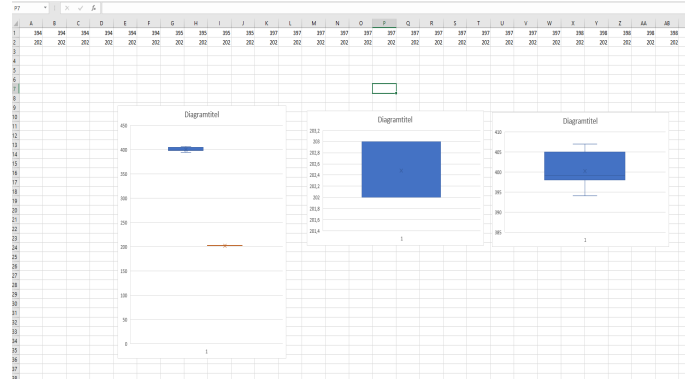
Tid for metoden at køre før optimering i millisec over 100 kørsler :

[illegible]

Tid for metoden at køre efter optimering i millisec over 100 kørsler:

(2)

Figure 3: Output fra profiler



Vi kan se ud fra vores box plot at vores mean og standard diviacion på ingen måde overlapper og at der tydeligt er et proformance boost i vores program.

1.5 make the program 50 procent more efficient

Vi kan se at vores avrage proformance før optimering ligger omkring 400 milisec og at den ligger omkring 200 efter. Vi har derfor opnået en 50 procent forbedring.