GCD Week 5 – MapReduce

Kevin Jetten

i318843

Inhoud

[Part 1: Installation and introduction 3](#_Toc495442376)

[Part 2: Wordcount 4](#_Toc495442377)

[Fixing the data 4](#_Toc495442378)

[Activity 2.2: Optimization 4](#_Toc495442379)

[Activity 2.3: Performance 6](#_Toc495442380)

# Part 1: Installation and introduction

There was a slight issue I had when I tried to run the examples. The console could not recognize the command “python”. Apparently, on W10 with powershell, you have to use the command “py” instead.

In the image below I added the results I got.

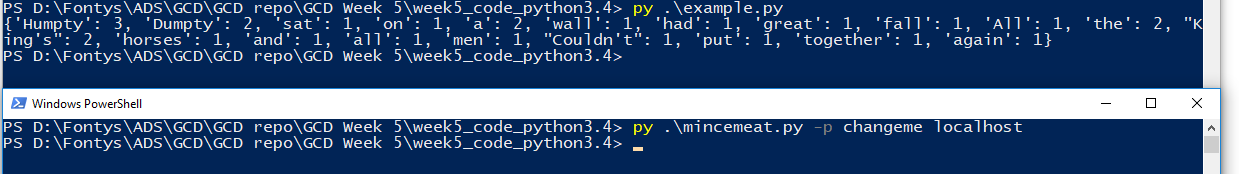


Image Example resultPart 2: Wordcount

## Fixing the data

When reading the books provided in the Gutenberg Small folder you get Unicode errors. Because there are so little errors, I decided to fix them by hand instead of writing a complicated piece of code to automatically fix the invalid text.

To find the correct text, I googled the previous lines to find the correct word without a special character.

**The following text is edited:**

vrctr10.txt - position 309292: aperε -> aperusic



## Activity 2.2: Optimization

Because the optimization points are summed up in the assignment I will talk about them in the same way.

**Stopwords**

I imported the python file and put the stop words in a dictionary with all the words or letters that have to be filtered.

**One-letter words, like ‘a’**

To avoid forgetting common used one letter words I just took all letters from the strings package and added them to the dictionary with filter words.

**Hyphens, periods, commas, other punctuation**

I choose to filter these out before adding the datasource to the server. This way the word count will be correct and you won’t delete words that have these in them or words at the end of the sentence. I added a more detailed example in the code.

**Top 10 words:**



**Code:**

## Activity 2.3: Performance

To test the execution time as correctly as possible I had to automatically start the clients when the server is running. Even with a bat script starting the clients it’s still not accurate because you have to click the bat file manually.

When I tried to start the bat file from the python file I had the problem that if it’s launched before the server is started it won’t connect and automatically close the cmd and if the server was started the python script would not launch the next line of code because its waiting for a server response. I fixed this with some basic multi-threading.

These were the results:

|  |  |
| --- | --- |
| Number of processes | Time spend in seconds |
| 1 | 33.999 |
| 2 | 13.295 |
| 3 | 14.364 |
| 4 | 12.75 |
| 5 | 12.113 |

It is very obvious that there is a real difference between one and multiple processes but after 2 processes there is little to no difference anymore.