K Most Popular Words

Programming Assignment -1

Due Date: 04/23/2023 11:59 PM Points 15

Goal

Multiple factors affect the performance of an application. The assignment aims to analyze such factors -- input size (or load), algorithm efficiency, data structures, system resource utilization (e.g., CPU, Memory), and others. The assignment will allow you to understand the impact of the above factors on performance. And we'll learn how the size of the input dataset impacts performance and what techniques we should use to alleviate the problems introduced by increased input dataset size (the crux of Big Data).

Total points awarded: 15 points, i.e.15% of your final grade.

Input Datasets

Initially, try testing the code with a text file of a smaller size provider below.

https://drive.google.com/file/d/1dX8sgA2dX-0Pr2NFePopKMtFGDYnmVBo/view?usp=sharing

Then move onto three large datasets of different sizes. The link for the dataset is as follows:

https://drive.google.com/file/d/1kaVM15rD9_O9HsvzrUKkIZ4R6ETUAdo4/view?usp=share_link

The dataset is a zip file that consists of three files with only English words.

1. data_300MB.txt – A text file of size 300MB.
2. data_2.5GB.txt – A text file of size 2.5GB.
3. data_16GB.txt – A text file of size 16GB.

Another file contains the stop words (**https://gist.github.com/sebleier/554280**) These words should be skipped.

Programming Language: Any language you prefer.

The Problem Statement

Design and Implement an efficient java/python/scala code (or any language you prefer) to determine Top K most frequent/repeated words in a given dataset (example: K = 10). The objective here is to obtain the result with the least possible execution Time (or with the best performance on your computer). You can use the same k value across all the files. Please note that you need to skip the "stop" words such as 'the' – please see the list of stop words here: **https://gist.github.com/sebleier/554280**

Execute your code on each of the three input data files separately. It is a good practice to execute your code initially on a 10 MB dataset (or any small text file) and then repeat on larger datasets.

Analyze the performance through different metrics such as running time, speedup, CPU utilization, memory usage, etc. Or You can come up with any metrics that make sense for this application/program.

Present a detailed analysis of why you use a particular algorithm or a particular data structure to solve this problem.

NOTE:

- The result should preserve case sensitivity i.e. words "Title" and "title" are considered as two different words.
- The input dataset contains only English alphabets, white spaces and hyphen separated words i.e. "a-z","A-Z","\s".

Deliverables

NOTE: All these files should be in a SINGLE tar/zip file uploaded on Camino

1. Source Code (Java or Python or Scala files, optional scripts for preprocessing if needed) (10 points)

   Note: Code should be documented and commented on. Please provide a README file to run your code.

2. PDF report file containing the analysis of the results. The maximum number of pages you should use is 6 pages (max) only including everything, graphs, references, etc. (5 points)
3. A Demo of code execution.

The report should contain the following analysis:

- Algorithm & Data Structures & Design: Justify the algorithm and overall design used in your code. Data Structures: Reason for choosing the data structures and their advantages w.r.t the problem size/complexity.
- Presenting Performance Data: This is the part you can show your skills in data visualization and data analysis. You can use a wide variety of ways to present data. For example, you can plot graphs of speed up vs execution parameters for each data set. Explain your observations with proper reasoning.
- Good Analysis Report: (2 Bonus Points)

Please describe your system specification: List out your laptop configuration – CPU cores, memory, etc.

Note: **The extra 3 bonus points** will be awarded for detailed and holistic analysis.

Note: **One submission per group is enough**

Note: For JVM tuning references: Guide to the Most Important JVM Parameters | Baeldung

Typical criteria for grading:

1. How well the basic implementation is analyzed using different experiments and presenting different insights?
2. How well the data is presented and overall how well the analysis is conveyed?
3. What are different optimization techniques applied & How well analyzed the impact of these techniques.