Inverted Index and Search

Final cloud computing project, a.y. 2024/2025

Context

The inverted index is the foundational data structure used in information retrieval systems like Google Search. Given a large collection of text files (e.g., articles, books, web pages), the inverted index maps each detected word to the files in which it appears. This kind of data structure is fundamental to enable quick search of files containing a specific term.

Goal of the project

In this project, you will build a basic search engine backend, constructing an inverted index from a collection of files. Specifically, in order to get **up to 4 points**, you need to:

- Install Hadoop in **fully distributed** mode (as described in the installation tutorial that I shared on Teams)
- Implement two solutions, one in **Hadoop** (Java) and one in **Spark** (Python), which produce an inverted index given a collection of files. For each detected word, the inverted index should report: i) any filename where that word is found; ii) the number of times the word appears in that file. The produced inverted index may **for example** be in the following form, where filename and number of occurrences are separated by a ":":

```
cloud    doc1.txt:1    doc2.txt:1
computing    doc1.txt:1    doc2.txt:1
is    doc1.txt:1
important    doc2.txt:1
```

- Implement a **combiner** logic
- Use **`setup()`** and **`cleanup()`** methods, only if appropriate
- Implement a very simple **search-query system** as a non-parallel Python script, which returns all the filenames containing the query term. The output must contain only the filenames, without any indication of the number of occurrences in each file. If the query is composed of multiple terms, the system must return the files that contain all the terms in the search query. For example, assume the search query is "*cloud computing*". `doc1.txt` contains only "*cloud*", whereas `doc2.txt` contains both "*cloud*" and "*computing*" (even not close to one another). In this example, the search output is only `doc2.txt`
- Choose **your own file collection** (e.g., articles, books, web pages). Besides, consider **different sizes** of file collection, spanning from **few KBs to some GBs**
- Perform a **comprehensive performance evaluation,** collecting statistics on execution aspects such as execution time, memory usage, etc., comparing the Hadoop and Spark solutions
- Write a brief (maximum 4-5 pages) **report** to present the MapReduce pseudocode of the Hadoop solution, description of datasets, and experimental results.

In order to get up to further 3 points (for a total of 7 points):

- Implement **in-mapper combining**
- Develop (and evaluate over the different datasets and file sizes) an equivalent **Python non-parallel** solution to build the inverted index
- Increase the **number of reducers** and collect execution statistics

FAQs on the Cloud computing project and final exam

1. **When and how to upload the final project?**
   - The final project (code + report) must be uploaded through the **Google form** at: https://forms.gle/ycfMLuy5qaMsgG1q8
   - Each group has to submit their project as a .zip file named "Group{Name}_Session{DD-MM-YYYY}.zip"
   - Please, do not include the input datasets in the .zip file
   - Project upload must be performed **three working days before** the official session day, at the latest. For example, if the official session is scheduled for the 05/06/2025 (Thursday), students have time to upload their project by 01/06/2025 included (Sunday), hence allowing for three working days before the official session day (Monday, Tuesday, Wednesday). **Projects that are uploaded after the deadline will be considered for the next session**
   - Once the project is uploaded, **no further modifications** are allowed to it

2. **What about the project discussion?**
   - Project discussion will take place on the official session day or a few days before, depending on the number of projects submitted in that session. In the second case, I will inform you of the exact date and time
   - **All group members** must be present that day and discuss the project as part of their exam
   - Project discussion will consist of **two parts**:
     - For the first part (duration 15 minutes – hard time), students have to prepare a presentation (either through a separate PowerPoint or directly commenting the project report) of the MapReduce pseudocode and related design choices, comment the datasets and obtained results, and show an execution demo of their work on the cluster
     - The second part (no duration pre-established) will be a detailed discussion of code
   - All the group members must show **complete knowledge** of each aspect of the project. If a student does not demonstrate active involvement in the different parts of the project, that student will have to develop the project assigned to single students and discuss it in another exam session

- Discussion of the project is in English if at least one group member is a non-Italian speaker. Otherwise, the group can choose to discuss their project either in English or in Italian

3. **What about the oral exam?**
   - The overall exam consists of Hadoop project discussion and the oral exams for my part and Prof. Vallati's one
   - Students can take the different parts of the exam in **different sessions and in any order**
   - Each part of the exam will have a **validity of one year**. After that deadline, that part must be repeated by the student. For example, if you discuss my oral part on 05/06/2025, this will be valid until 04/06/2026
   - Oral exams will take place on official **session days** only
   - When you enroll for a session on the system, please **specify in the notes** which part(s) of the exam you are going to take
   - **Groups do not stand** for oral exams: members of the same group can take oral exams in different sessions
   - Students can choose to take their oral exam either in English or in Italian