CS542000 - Cloud Programming

HW1-Inverted Index

National Tsing Hua University

2016, Spring Semester

Outline

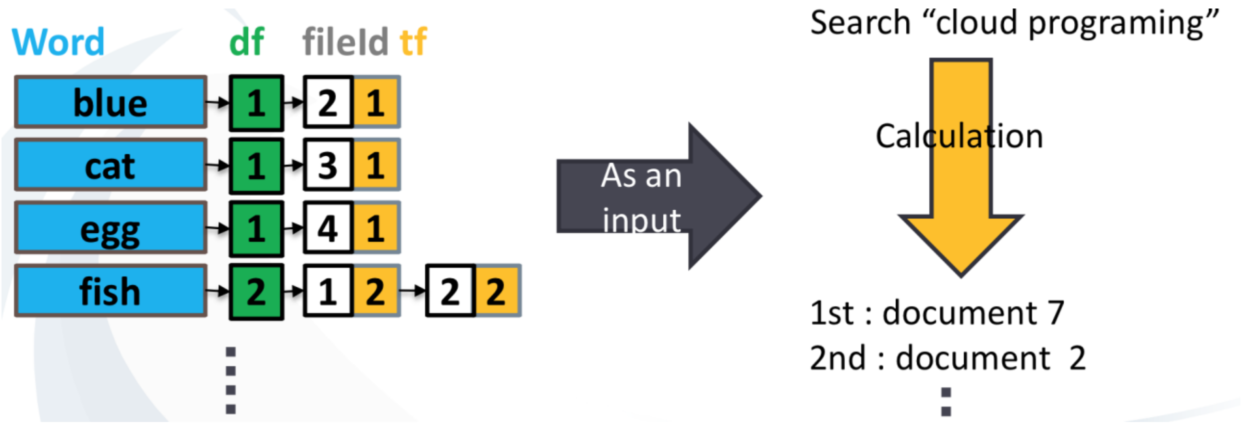
- Problem Description
- Input/Output Formats
- Grading
- Reminder

Outline

- Problem Description
- Input/Output Formats
- Grading
- Reminder

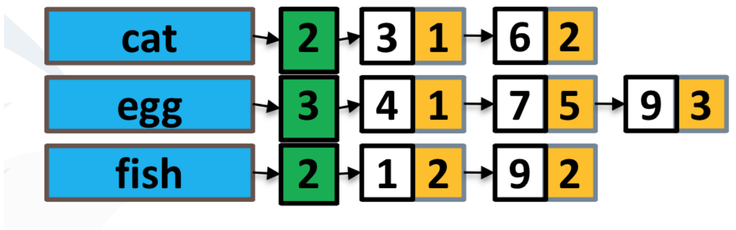
Problem Description

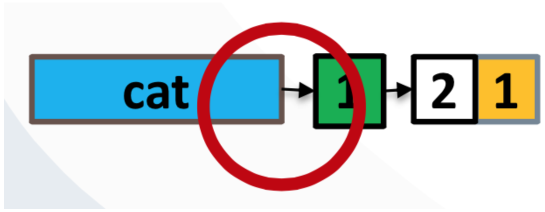
- Write a ranked-based search engine, which includes
  - Part 1: Inverted Index
  - Part 2: Retrieval
- Your inverted index table should include **term frequency(tf)** and **document frequency(df)** of each word. Thus, you can search by this table in part 2.

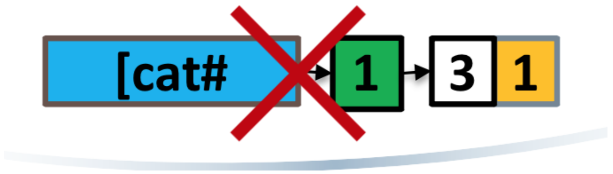


Problem Description

- Part 1 - Inverted Index
- Write mapreduce code to output inverted index table
  - Your table should include **document frequency** and **term frequency** for each word
  - File name should be sorted.
  - Words in your table should not contain useless notation







Problem Description

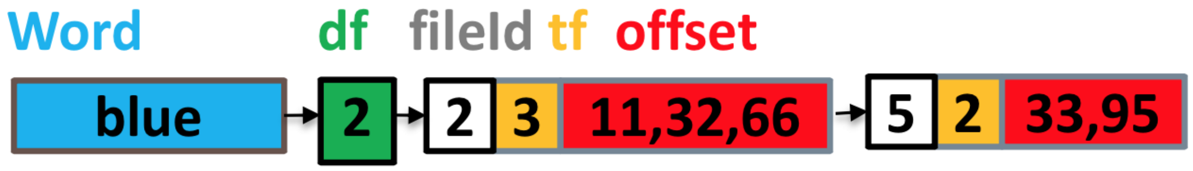
- Part 2 - Retrieval
- Use **MapReduce API** to search words based on your inverted index table, and output their rank
  - Use **TF.IDF Term Weighting** to rank words

$$w_{i,j} = tf_{i,j} \ * \log(\frac{N}{df_i})$$

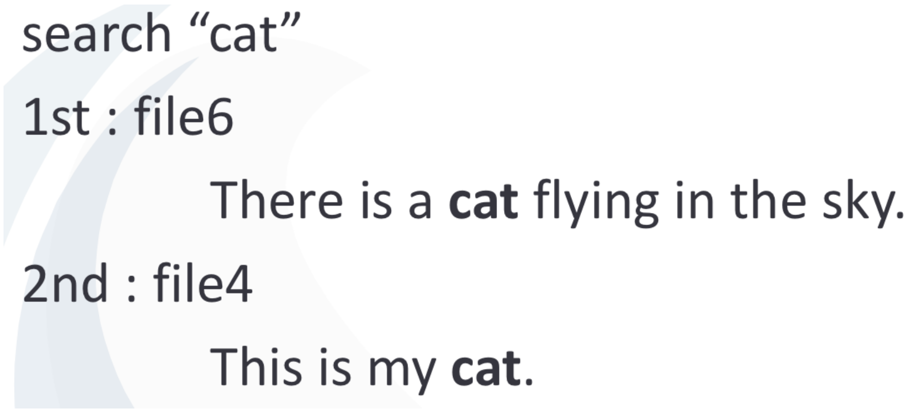
  - Be able to retrieve **multiple** key words for each query
  - Output the 10 highest files
  - You should not fix #files. (Demo with other testcase)

Problem Description

- Extend to full inverted index
  - Add field offset for each file



  - Output some fragments of file which contain at least one of keywords



search "cat"

1st : file6

There is a **cat** flying in the sky.

2nd : file4

This is my **cat**.

Problem Description

- Implement **at least one** advanced function
  - Retrieval can support “AND/NOT”
  - Retrieval can support “Ignore uppercase or lowercase”
  - Any other interesting extension you can think of!

Problem Description

- Report
  - Instruction : how to compile and execute your program
  - Design : explain your algorithm
  - Questions : choose two of them to answer
    - How many #phases you used to run mapreduce in part1?
      Is there any other way to do it?
      What's the pros and cons?
    - What's your extension?
      What's the most difficult part in your implementation?
    - How do you filter those useless notation?
      If we need to search these special notations, how to modify your filter?

Outline

- Problem Description
- Input/Output Formats
- Grading
- Reminder

Input

- Input files are Shakespeare’s book splitting into 44 files
- Input files are at /home/cp2016/shared/hw1/input

Output

- **Inverted Index Table** (We would checkout content in the table)

```
Word  df;file1 tf1 [offset1,offset2,...];file2 tf2...
```

- **Retrieval**

```
Rank {RANK}:      {FILENAME1} score = {SCORE}
************************
offset1     {FILE_FRAGMENT1}
offset2     {FILE_FRAGMENT2}
************************
```

Output

- You need not strictly follow the format as long as information of **df**, **tf**, **etc.** can be clearly distinguished.
- For Inverted Index, you do not have to merge all outputs into one files if you are using more than one reducer.
- Sample output format for implementation
  - output_invertedindex.txt
  - output_retrieval.txt

Outline

- Problem Description
- Input/Output Formats
- Grading
- Reminder

Grading

- [45%] Inverted Index
- [20%] Retrieval
- [10%] Extend to full inverted index
- [5%] Implement one extension
- [20%] Report + Demo

Outline

- Problem Description
- Input/Output Formats
- Grading
- Reminder

Reminder

- Upload HW1_{Student-ID}.zip to iLMS before **4/25 23:59:59**
  - HW2_{Student-ID}_code.tar.gz
  - HW2_{Student-ID}_report.pdf
- 0 will be given to cheaters. Do not copy & paste!
- Please start your work ASAP and do not leave it until the last day!
- Please refer to syllabus for late submission penalty.
- Feel free to ask question on iLMS or through e-mail.

Hint

- To get file name
    - Use Reporter and FileSplit class in mapper

Reference

- Hadoop
  - http://hadoop.apache.org/
- Hadoop 2.7.2 API
  - https://hadoop.apache.org/docs/stable/api/