# CS542000 Cloud Programming
# Homework 1: Inverted Index

Due: April 25, 2016

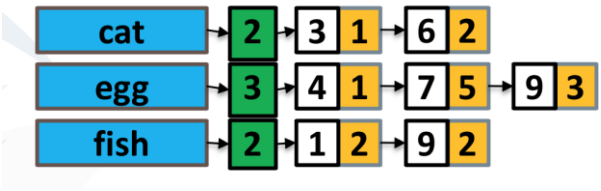
## 1 GOAL

This assignment helps you get familiar with Hadoop Map Reduce Framework on distributed cluster by implementing rank-based search engine which includes inverted index table and retrieval procedure. **Both Inverted Index and Retrieval should be done by using Hadoop Map Reduce API**.

## 2 ASSIGNMENT

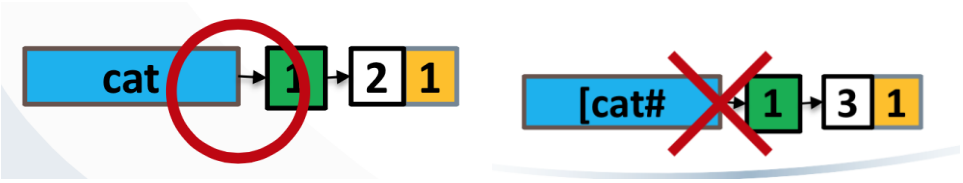
### 1. Inverted Index

Make an inverted index table for retrieval search. Your inverted index table should include **term frequency (tf)** and **document frequency (df)** of each word.




Every consecutive alphabets characters ([a-zA-Z]) in the inputs should be viewed as a word. Also, words should not contain any useless notation (e.g. [, #, $ …)

For example, `cloud-programming` should be divided into `cloud` and `programming`. `I'm` should be divided into `I` and `m`.




Please label fileID from 1~N and order them by filename.

Finally, your inverted index should not fix # of input files, since TA may demo with other test case.

## 2. Retrieval

Use MapReduce API to search words based on your inverted index table, and output their rank.

Use the **TF. IDF Term Weighting** to rank words

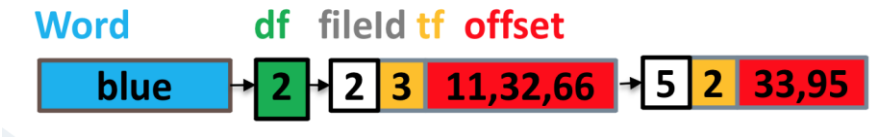
$$w_{i,j} = tf_{i,j} * \log_{10}(\frac{N}{df_i})$$

Your retrieval procedure must be capable of:

1. Retrieve multiple key words (OR operation) for each query.
2. Output 10 highest score files. Order files by filename and give them same rank if they have same score.

## 3. Extend to full inverted index

Enhance your inverted index table with field offsets. Offset denotes the **byte position of** a word in the file, which is used to quickly locate the word in the file.




Enhance retrieval procedure and 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**.

## 4. Implement one extension

Please also implement at least one of the following advanced feature in your rank-based search engine:

1. Retrieval can support "AND/NOT" search
2. Retrieval can support "Ignore uppercase or lowercase"

3. Any other interesting extension you can think of!

**5. Report**

Instruction: *how to compile and execute your program*

Design: *explain your algorithm*

Question: *choose two of them to answer*

1. How many #phases you used to run mapreduce in Inverted Index

   Is there any other way to do it?

   What's the pros and cons?

2. What's your extension?

   What's the most difficult part in your implementation?

   How do you filter those useless notation?

3. If we need to search these special notations, how to modify your filter?

# 3 INPUT / OUTPUT FORMAT

**1. Input format**

Inverted index should take several files as input, and build/output an inverted index table.

Retrieval should take inverted index table as input, and output the ranking of files.

Sample input files for debugging are Shakespeare's book splitting into 44 files. You can find them at /home/cp2016/shared/hw1/input.

**2. Output format**

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

**Inverted Index Table:**

```
{Word1}   df;file1 tf1 [offset1,offset2,...];file2 tf2 ...
{Word2}   df;file1 tf1 [offset1,offset2,...];file2 tf2 ...
...
```

**Retrieval:**

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

# 4 GRADING

1. Inverted Index (45%)
2. Retrieval (20%)
3. Extend to full inverted index (10%)
4. Implement one extension (5%)
5. Report & Demo (20%)

# 5 REMINDER

1. Please package your codes and report in a file named **HW1_{student-ID}.zip** which contains:

   **i、 HW1_{student-ID}_code.tar.gz**

   **ii、 HW1_{student-ID}_report.pdf**

   And upload to iLMS before **4/25/2016 (Mon) 23:59**

2. 0 will be given to cheaters. Do not copy & paste!
3. Since we have limited resources for you guys to use, please start your work ASAP. Do not leave it until the last day!
4. Late submission penalty policy please refer to the course syllabus.
5. Asking questions through iLMS or email are welcomed!