CS542000 Cloud Programming

HW2: Inverted Index

資工碩一 張維元 103062590

May 17, 2015

Instruction

1. Part1
   - javac -classpath hadoop-core-1.0.3.jar -d class1/ part1/*
   - jar -cvf InvertIndex.jar -C class1 .
   - hadoop jar InvertIndex.jar hw2_part1.InvertIndex Hw2-Part1-input Hw2-Part1-output
   - Input: hdfs/input/*
   - Output: hdfs/part-00000

2. part2
   - javac -classpath hadoop-core-1.0.3.jar -d class2/ part2/*
   - jar -cvf Retrieval.jar -C class2/ .
   - hadoop jar Retrieval.jar hw2_part2.Retrieval
   - Input: hdfs/part-00000
   - Output: local/SampleOutput-retrieval.txt

Design

1. Part1
   - In the mapper, each word in the file and its filename as input, produce the key-value pair ((file, filename), fileWordcount), where value is Wordcount in this file.
   - In the combiner, calculate the TF and transform to the key: file – value: (filename, TF) pair, and pass it to reducer.
   - In the reducer: calculate the IDF and output this form:
     - (**word**\t**IDF**, **TF1**\t**filename1**, **TF2**\t**filename2**, …)
2. part2
   - In Retrieval, we will split the query by blank, and then search the hdfs inverted index table individually.
   - Calculate and merge(or) the results to the SampleOutput file.

Questions

1. How many #phases you used to run map reduce in part1? Is there any other way to do it? What's the pros and cons?
    - I used one pass mapReduce in part1.
    - Many other ways:
        - TF can be calculated in mapper.
            - Pro: No combiner.
            - Con: The mapper needs to keep the counts and offsets of all kinds of words. It wastes more memory.
        - TF can be calculated in the reducer.
            - Con: That needs one more map reduce pass for calculating document frequency.

2. What's your extension? What's the most difficult part in your implementation?
    - Query can ignore case
        - In Part1, all words are converted to lowercase first, and the query in Part2 are converted to lowercase.
    - Filter the stop words
        - I created the StopWords list to filter those useless notations. Omit the word in StopWords when reading words in Part1.

3. How do you filter those useless notations? If we need to search these special notations, how to modify your filter?
    - Omit the notations using Regular Expression
        - replaceAll("\\W", " ") to filter the useless notations