Lab 2 – Word Count On-disk (External Sorting)

CC5212-1

March 18, 2015

- Download `http://aidanhogan.com/teaching/cc5212-1/code/mdp-lab2.zip`.
- Download `http://aidanhogan.com/teaching/cc5212-1/data/lab1/es-abstracts.txt.gz` (if needed; same file as last week) and put it in the same folder as above. Do NOT unzip it. It contains abstracts from all of Spanish Wikipedia.
- Unzip the first file (`mdp-lap2.zip`) and open it as a project in Eclipse.
- Now you want to count the number of occurrences of Spanish bi-grams (pairs of consecutive words) appearing in the abstracts and print the top k most popular bi-grams.
- The first step is to extract the bi-grams from the abstracts file. Run `org.mdp.cli.ExtractBigrams` with `-i [dir]/es-abstracts.txt.gz -igz -o [dir]/es-bigrams.txt.gz -ogz`. It will write all the bi-grams to a file with one bi-gram per line.
- Now you need to code the external sorting method:
  - First you need to implement the `writeSortedBatches` method. Load (max) `batchSize` lines from `in` into memory and sort them. Make sure to keep duplicates. For the moment, you can ignore `reverseOrder`. Once you have a sorted collection of lines of size `batchSize` (or file is empty), call `writeBatch(lines, tmpFolder, batchId)` where `batchId` is the number of batches done so far. Add the file name returned to `batchNames`. When finished, return `batchNames`. (Be careful with the last batch ... try to avoid duplicating code if you can figure out a way!)
  - Second you need to implement the `mergeSortedBatches` method. The batches have been opened for you. Each batch is sorted. You need to read from the file whose next line is lowest, write the line, and then move that file onto the next line. This is a bit tricky ...
- Try calling your sorting method over the bigrams file: run `org.mdp.cli.ExternalMergeSort -i [dir]/bigrams.txt.gz -igz -o [dir]/bigrams-s.txt.gz -ogz -b 500000` (`-b` sets the batch size!). Also set `-Xmx500M` in the VM arguments (bigger heap). How long did it take?
- Next open `org.mdp.cli.CountDuplicates`. Here we're going to open the sorted bi-grams and count the consecutive duplicates. Opening files and such is done for you. Your task is to implement the `countDuplicates` method.
  - You need to read the sorted file from `in` and keep a count `c` of consecutive lines that are the same.
  - When the line becomes different, you want to output a line to `out` with `c+"\t"+line`.
  - Make sure to catch the count of the last line!
- Now we want to sort this count-file using `ExternalMergeSort` to get top-k. This leaves us two small tasks:
  - We need to print the output of `org.mdp.cli.CountDuplicates` so that the lines will sort naturally by occurrence. How can we do this?

  - In `org.mdp.cli.ExternalMergeSort`, we need to implement the `reverseOrder` flag in both the `writeSortedBatches` and `mergeSortedBatches` methods.

- Run `org.mdp.cli.CountDuplicates` over the sorted bi-gram file from earlier: `-i [dir]/bigrams-s.txt.gz -igz -o [dir]/bigrams-c.txt.gz -ogz`.

- Sort the output `[dir]/bigrams-c.txt.gz` using `ExternalMergeSort`: `-i [dir]/bigrams-c.txt.gz -igz -o [dir]/bigrams-cs.txt.gz -ogz -b 1000000 -r`. The `-r` flag says to run the sort in reverse order. Also set `-Xmx500M` in the VM arguments (bigger heap).

- Did you make it this far in class? Try find the best batch size to run the fastest sort over the original `[dir]/bigrams.txt.gz` file. ☺

- Submit the `RunWordCountLocally` class to ucursos before the Monday lecture.