CS5406 Performance Engineering of Computer Systems

Lab 3 – Search Engine

index No: 198011G

Name: K. T. D. S. De Silva

---

step 2

The inverted index is stored in a text file.

Each keyword is given a separate line in the file

ex:

continuing {https://calomel.org/network_performance.html, https://graphics.stanford.edu/~seander/bithacks.html}

contractual {http://en.wikipedia.org/wiki/Computer_performance}

contraints {http://www.javaperformancetuning.com/}

contrast {https://calomel.org/network_performance.html}

the keywords and matching URLs are separated by a “\t”, and URLs are separated by a comma

Keywords are stored in alphabetical order, so it is easier to search keywords

Step 3

Index generator is developed as “index_generator.cpp”, using C++ programming language. The program has the following steps

1. read the URL’s saved in the folder “search_space”, one by one.

2. read each line of the file, convert it into lower case, remove the white spaces, and unnecessary spacing characters from the text line

3. separate the words in the file, and store them in a local map data structure

4. In the local map, the key is the keyword, and the value is the number of occurrences of the keywords inside the URL. (Keeping track of the number of occurrences will help finding the best search result in the next phase)

5. At the end of reading one URL, store all the keywords in the “Keyword_map” map data structure.

In this global map, the keywords are stored as, keyword, and the set of URLs carrying each keyword. Set URLs are concatenated into one *string* separated by commas. This can save space consumed by a string array.

6. Since the map data structure stores keys in alphabetical order, it is written directly into a text file in the format as mentioned in step 2

step 4

the search engine is developed as “search_engine.cpp”.

1. Read the text file *inv_index.txt* line by line.

2. Load the data into a std::map data structure as;

key – keyword, value – set of URLs containing the keyword as a single string, separated by commas.

The URLs are splitted into separate strings only when the keyword appear in a search result.

3. search for the string word by word: by using an iterator in map data structure, which is in alphabetical order.

Results:

total elapsed time: 59.9 ms

average elapsed time (per search): 0.0599 ms

Machine details

CPU	Intel Core i7-4510U
Number of Cores	4
Clock speed	2.00 GHz
RAM amount	5985 MB
OS Version	Ubuntu 16.04.6 LTS, Linux 4.15.0-48-generic (x86_64)
Compiler version	g++ with no optimization