1. Summary

With the goal of improving search quality and supporting for academic search, the authors (Brin and Page) present Google as a prototype of a scalable and high-quality search engine in this paper. In order to keep up with the dramatic growth of the web, the techniques of search engine also need scaling up.

The major data structures that Google utilizes include BigFiles (allocation), repository (full HTML of web pages), document index (information of each document: docID, URL, title), hit list or barrel(occurrence of a word in a document), lexicon (word list and hash table of word -> barrel pointers), forward index (docID -> list of wordID's) and inverted index (wordID -> docID's list).

With innovative features such as PageRank (can prioritize important results) and Anchor Text (brings accurate descriptions of web pages and facilitates crawling non-text contents), Google has a system architecture designed for implementing subtasks including (1) crawling, (2) indexing and (3) searching. Generally, the search engine inputs "uncontrolled heterogenous documents" from the web and outputs search results with quality and efficiency to users.

For each subtask,
(1) crawling is processed on a distributed system with URL servers implemented in Python. DNS lookup and asynchronous IO makes crawling even more complex.
(2) indexing could be further divided into 1) parsing, 2) indexing documents into barrels and 3) sorting to obtain inverted indices from forward indices. Note that parsing not only convert each document into hits/barrels but also convert each link into an anchors file. Moreover, indexing and sorting applies different approaches (e.g., log extra words then merge, subdivide to fit into memory) to achieve parallelization.
(3) searching takes into account a document ranking function of many factors including hit lists, hits from anchor text and PageRank. For a multi-word search, word hit proximity also plays an important role in ranking. Additionally, weights for different types in ranking could be better determined with user feedback.

Thanks to PageRank, anchor text and proximity, Google returns search results with even higher qualities compared to its peers. Also, with efficient storage (about 100 GB), it is able to answer most search queries with 10 seconds.

2. Comments

The prototype of Google is very attractive on the way how it makes the system scalable and search qualitied. It is wise to take seriously the issue scalability of search engines given the background of dramatically expanding web. Out of the same reason of having too many candidate search results, improving the search quality to return top-relevant ones is necessary and significant. On one hand, Well-designed data structures help saving time and space cost. For example, a relative wordID is stored for forward indices instead of absolute wordID so that less bits are needed. Also, parallelization is implemented during crawling and indexing (sorting) to reduce time cost. On the other hand, link structure, anchor text and PageRank boost the quality of search results. It is worth mentioning that anchor text not only materializes the crawling of non-text web contents but also provides abundant valuable information for rangking, and that PageRank is one of the most ranking algorithms in the world though no longer the only method Google company uses. Additionally, proximity measure also makes search results more relevant for multiple-word queries.

Furthermore, the authors also mention in the future work section many more additional functions to be added into Google, which contribute to today's world-leading platform Internet service and products (e.g., Google Maps, Google Scholar). What is amazing is the self-positioning of not only a search engine but also a research tool. Indeed, it provides people with valuable information, and itself is also being a wonderful "lab" of exploring new ideas in the research field of search engine.

3. Questions

I am curious about some questions such as how the crawler deals with unexcepted/incorrect response from the web pages in Section 4.3, how users' feedback is included in the determination of values of weight coefficients of the ranking function in Section 4.5.2, and how the crawler update downloaded webpages (in what time interval and how to do the work efficiently) in Section 6.1. The answers to these questions might be included in references or other online resources. And I believe that there are no fixed answers as time goes by and techniques develops.