COMP4321 Project Phase 1 Report

RocksDB database

For the rocksDB database, we have put all the database files under the “rocksDBFiles” folder in the project root directory. We only use one single database in one directory but we used different “ColumnFamilies”[1] to separate indices of different types.

The followings ColumnFamilies are used:

1. “UrlIdToMetaData”: table used to store metadata (page title, url, last modification date, size, top 5 keywords and all the child links) for the site that is going to appear in the search result
2. “ParentUrlIdToChildUrlIdData”: table used to map parent site url id to child site url id(s), the ids will be stored in a string with a space character as separator if there is more than one child site to the parent site
3. "ChildUrlIdToParentUrlIdData": table used to map child site url id to parent site url id(s), the ids will be stored in a string with a space character as separator if there is more than one parent site to the child site
4. "UrlToUrlIdData": table used to map from url to url id, using url as key and url id as value
5. "UrlIdToUrlData": table used to map from url id to url, using url id as key and url as value
6. "WordToWordIdData": table used to map from word to word id, using word as key and word id as value
7. “WordIdToWordData”: table used to map from word id to word, using word id as key and word as value
8. "UrlIdToKeywordFrequencyData": table used to map from url id to a json string that store the keyword (key) frequency (value)
9. "UrlIdToTop5Keyword": table used to store the top 5 keywords for each site using url id as key

The ColumnFamilies that are going to be built:

1. “InvertedIndexData”: table that store inverted index, using word id as key and the value will be a list of documents using url id and the location of the word
2. “VectorSpaceIndexData”: table that store the vector space model index data, with url id as key and the keyword vector as the value

We created mapping table like word ⇔ word id and url ⇔ url id as conversion table, we mainly used their id for finding relevant information because that would save some space on the indexing part in the database (assuming url string size > integer of the index). We then have a metadata table to store the metadata of the site to enable quick retrieval of the site information once we confirmed to include that site into our search results. The parent link could be ever changing in the indexing part, so we will compute the parent links on the fly during query. We also keep track of the site network graph in two tables, one table only stores links from parent to child and another mainly stores from child to parent to know all the links inflow and outflow.

[1] https://github.com/facebook/rocksdb/wiki/Column-Families

We also have two tables that store keyword frequency and top 5 keywords for now. As it is still in development, it would be useful for us to query that after indexing. But we think that at the end of the program, we will be able to do all the indexing we need to generate inverted index and vector space index data and store them in the corresponding tables. These 2 tables that store keyword frequencies might be deprecated in future and replaced by inverted index data, vector space index data and other tables that we used to implement the extra bonus features.