COMP 4321 - Search Engines for Web and Enterprise Data

Project – Phase 1

Group member:

MOK Chi Wing 20213055

CHOY Ting Wai 20211980

CHU Kwok Ning 20212001

# Jdbm Database scheme design:

We use hash table in our jdbm database.

The format of each table is as Table\_Name (Key, Value)

1. **pageId (URL, PageId)**

This is a mapping table for URL to reduce the overhead for other index tables. Each unique URL will be assigned a unique ID, e.g.: 0000000, 00000001 etc.

1. **page (PageId, URL)**

This is a mapping table for retrieve page URL. Our system can make use of this table to retrieve the specific page URL by giving the specific page ID.

1. **wordId (Word, WordId)**

This is a mapping table for word to reduce the overhead for other index tables.

Each unique word will be assigned a unique ID, e.g.: 0000000, 00000001 etc.

1. **word (WordId, word)**

This is a mapping table for retrieve word. Our system can make use of this table to retrieve the specific word by giving the specific word ID.

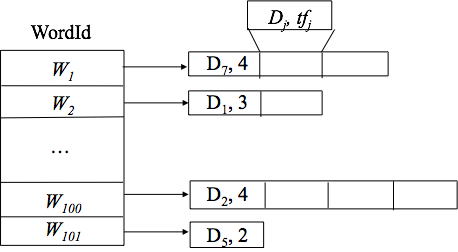
1. **pageInfo (pageId, (title, last modification, size))**

This is a table that stores page information, including title, last modification time and size, for each page by page ID.

1. **BodyWordPosting (WordId, List<(PageId, term frequency, List<word position>)>)**
2. **TitleWordPosting (WordId, List<(PageId, term frequency, List<word position>)>)**

BodyWordPosting and TitleWordPosting store the posting list for each word by word ID. BodyWordPosting will handle all the word which extracted from the pages, while TitleWordPosting only handle the title’s word extracted from the pages. Our search engine will derive and implement a mechanism to favor matches in title. For example, a match in the title would significantly boost the rank of a page. The posting list will store the Page ID, term frequency in the document and the word position. This will be used to calculate cosine similarity.

The word position will be used to support phase searching in our search engine.



1. **InvertedBodyWordPosting (PageId, List<WordId, term frequency>)**
2. **InvertedTitleWordPosting (PageId, List<WordId, term frequency>)**



InvertedBodyWordPosting and InvertedTitleWordPosting are the inverted posting list which is used to support deletion of a webpage entry. It will store the list of word ID and term frequency in the page by page ID. Upon deletion, we will retrieve the forwarding list for this particular webpage and update the inverted posting list accordingly.

1. **ParentLink (PageId, List<Parent link>)**
2. **ChildLink (PageId, List<Child link> )**

ParentLink and ChildLink store the parent link and child link for each page by page ID. These tables are used to implement Google PageRank in final phase.

The following table will be added in final phase:

1. **hubWeight (PageId, Weight)**
2. **authWeight (PageId, Weight)**

The hubWeight and authWeight store the hub weight and authority weight for each page. Our system will use these tables to implement Google PageRank.

# Work contribution and tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Mok Chi Wing | CHOY Ting Wai | CHU Kwok Ning |
| Database design | ✔ | ✔ | ✔ |
| Spider |  | ✔ |  |
| Parent and child link table |  |  | ✔ |
| Word and Page table | ✔ |  |  |
| Test program | ✔ | ✔ | ✔ |
| Debug | ✔ | ✔ | ✔ |

|  |  |  |  |
| --- | --- | --- | --- |
| Work contribution % | Mok Chi Wing | CHOY Ting Wai | CHU Kwok Ning |
| Overall | 33.3% | 33.3% | 33.3% |