Crawler based Search Engine

Problem Description

Human-powered directory:

A human-powered directory, such as the Open Directory, depends on humans for its listings. When directories have enough websites for a category and it is “full” more categories and subcategories are created. Even characterizing and ranking are also done by humans only.

Ex: Yahoo open directories

Problem with this system:

In this case the open directory system is centralized and is more of human intervention and the rapid growth of web makes it very difficult to review and rank all the pages which were entered into the directories.

Crawler based Search Engine:

In the crawler based search engine crawler crawls the web periodically to find the updates and incorporates that in its database, Here there is no intervention of the human which is the case with the directory based search engine.

In this system, the web crawling (downloading of web pages) is done by crawler. There is a URLserver that sends lists of URLs to be fetched to the crawler. The web pages that are fetched are then sent to the storeserver. The storeserver then compresses and stores the web pages into a repository. Every web page has an associated ID number called a docID which is assigned whenever a new URL is parsed out of a web page. The indexing function is performed by the indexer and the sorter. The indexer performs a number of functions. It reads the repository, uncompresses the documents, and parses them. Each document is converted into a set of word occurrences called hits. The hits record the word, position in document, an approximation of font size, and capitalization. The indexer performs another important function. It parses out all the links in every web page and stores important information about them in an anchors table. This table contains enough information to determine where each link points from and to, and the text of the link.

The URLresolver reads the anchors file and converts relative URLs into absolute URLs and in turn into docIDs. It puts the anchor text into the forward index, associated with the docID that the anchor points to. It also generates a database of
links which are pairs of docIDs. The links database is used to compute PageRanks for all the documents.

The crawler will periodically return to the sites to check for any information that has changed. The frequency with which this happens is determined by the administrators of the search engine.

**Requirement of Distributedness:**

If the whole above system it is centralized as the number of sites increasing day by day (thousands of new sites are added everyday) In order to crawl entire web the crawler takes a lot of time, so it may not cache the recent data in turn it may not produce the recent result in the search result. If that is the case then the whole purpose of web crawler is lost. For this reason we go for the distributed database systems in which The main crawler is subdivided into smaller crawlers and different sub crawlers do the crawling activity in isolation with one other, resulting much improvement in the crawling operation.

When a user enters a query into a search engine (typically by using key words), the engine examines its index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text. The index is built from the information stored with the data and the method by which the information is indexed.

**Queries:**

1) The search engine should provide the possibility of giving the type of the content we are searching for (ex: searching for images, videos, doc etc.,)

2) This should allow the customization of the location search (ex: pages from only US) site or domain

3) Exact words mentioned in the specified query

4) Find the pages similar to the page

5) Find the pages that link to that page

6) Search for the query based on the position of the words stored in a web page
Entities:
- URL Server
- Crawler
- Store Server
- URL Resolver
- Anchor Info
- Search Info
- Indexer

Frequency of Queries:

<table>
<thead>
<tr>
<th>Site</th>
<th>Web</th>
<th>India</th>
<th>US</th>
<th>China</th>
<th>Russia</th>
<th>Japan</th>
<th>England</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1:</td>
<td>293</td>
<td>46</td>
<td>68</td>
<td>60</td>
<td>38</td>
<td>34</td>
<td>47</td>
</tr>
<tr>
<td>Q2:</td>
<td>-</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Q3:</td>
<td>20</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Q4:</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Q5:</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Q6:</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>