NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]
ISSN: 2394-3696 Website: ijiert.org
VOLUME 7, ISSUE 11, Nov.-2020

RESEARCH INDEXING ALGORITHMS

MENGATOVA XURSHIDA TOSHMUXAMATOVNA
Teacher, Termez branch of Tashkent State Technical University named after Islam Karimov,
xurshidamengatova7288@gmail.com

SAYMANOV SHAROFIDDIN SIROJIDDIN OGLI
student, Termez State University,
saymatovsh@mail.ru

ABSTRACT
This article analyzes the indexing algorithms used in advanced search engines, which are currently being developed and have a wide range of features. B-tree indexing method, reverse indexing method, data (proper) indexing methods in direct indexing methods. The results of the analyzes can be used to index the database of electronic documents when creating an advanced search engine.

KEYWORDS: B-tree, GIN (reverse indexing), GIST (data (proper) indexing), HASH, indexing, Trigger, PostgreSQL

INTRODUCTION

The growing volume of electronic documents causes problems with finding the necessary information quickly and accurately. As a solution to this problem, many large companies working in the field of software development, presented their products. Work is also underway to further improve them [1].
This article discusses data indexing methods in the B-tree indexing method, reverse indexing method, proper indexing methods, which are currently used in developing and advanced search engines.

MAIN PART

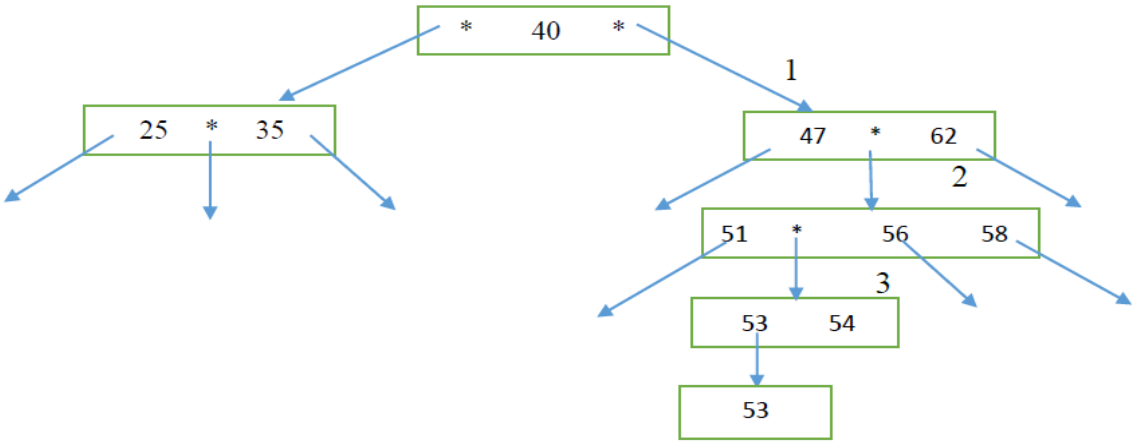
The indexing methods are analyzed in the example of PostgreSQL software. PostgreSQL supports several types of indexes: B-tree, GIN (inverse index), GIST (proper index) and HASH indexes. In practice, only three of these, B-tree, GIN and GIST, are used.

1. B-tree indexing method:

B-tree - Performs a tree-like search for structured data. This method was first developed by R. Beyer (R. Bayer) and E. This method was proposed by McCrate in 1970 and developed in 1972. [2]. The B-tree structure is used to organize indexes in many modern database management systems. The structure of the B-tree is used for the organization of indexes in many modern systems of database management. Performs indexing, breaking data in memory into blocks.
By default, PostgreSQL creates a B-tree index.
This index is created using the following command:

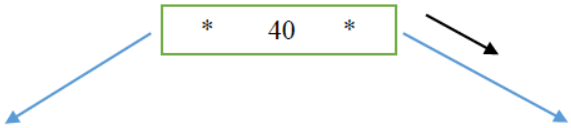
```
CREATE INDEX   index_name
ON  table_name
USING  btree   (column_name  COLLATE pg_catalog."default")
```

Like indexing algorithms, this algorithm also allows you to increase your search speed. In the following example, we can take a deeper look at how this algorithm works.
For example, let's look at the value 53. We start the search with a block that stores the value 40.



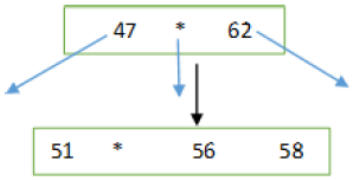
First, we compare the desired number with a value of 40. Since the value you are looking for in the expression is large, we continue the search on the right [3].

40 < 53



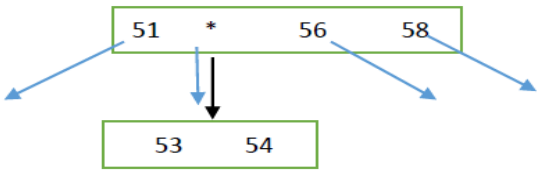
In the following steps, similar comparison expressions are performed, and it is determined in which block the search will continue.

47 < 53 < 62



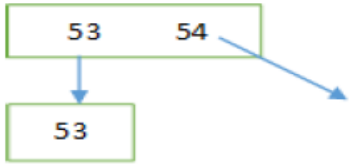
The values in all blocks are in ascending order. Therefore, as can be seen from the comparison result, we continue the search from the block of sorted values located in the central part.

51 < 53 < 56



Since the expression 51 <53 <56 is less than the second value, comparison with the next value is not performed, and we are looking for a block to the second value. At the end, you will find the value you are looking for.

53 = 53



The search is searched in the range corresponding to the numerical value until the key is found [3]. If the key you are looking for exists in the root consisting of blocks, the key will be found. To balance the algorithm, keys in blocks are distributed in equal amounts. This in turn minimizes the number of transitions to the next block.

2. Reverse indexing method

Reverse indexes are a clear data structure that stores words stored in files and information about their location. In the reverse indexing method, when indexing words stored in files, it separates the words without their suffixes and connecting words and saves these values as a list in its index database. The words in the list are in alphabetical order and in the location of the files. Typically, search engines use reverse indexing to compile a list of files, store words in search queries, and sort files in a list by participating in search queries [4].

There are two approaches to using inverse indexes. The first is to index the data area by function, and the second is to save the search vector at an additional level of the table, which will be maintained throughout the subsequent indexation [5]. As an example, let’s show the construction sequence of this index in PostgreSQL software.:

```
CREATE INDEX index_name
ON table_file
USING GIN
(to_tsvector('russian'::regconfig, (text || ' '::text) || title))
```

Here:
index_name - index name;
table_file - table name;
title - a column in the table;
to_tsvector() – indexing function.

In this figure 1, an index is created for our column called title, and if we type mant into this column, the tvs column will automatically index it and save the vector values.

title text	tsv tsvector
O‘zbekiston Respublikasi Bosh vaziri	'98':393 'abduhakimov':168 'abdulaziz':194 'abdulla':5 'abdusamatov':370 'adham':118,187,252 'ad
1.Архитектура серверных операционных	'1':1 '10':100 '11':113 '12':126 '13':134 '14':143 '15':150 '16':160 '17':175 '18':184 '19':197
сетях	'сетях':1
Замонавий шифрлаш алгоритмлари икки	'ажраладиган':6 'алгоритмлари':3,13 'боғлиқ':11 'бутунлилиги':8 'замонавий':1 'икки':4 'калитлар':1

Figure 1.

Alternatively, triggers are created to automatically update values stored in the tsvector format. If the text in the created trigger text format column is changed, deleted, or new text added, the values in the tsvector column are automatically updated. The command to create a trigger is as follows:

```
CREATE TRIGGER tsvectorupdate
BEFORE INSERT OR UPDATE
ON table_test
FOR EACH ROW
EXECUTE PROCEDURE tsvector_update_trigger('tsv', 'pg_catalog.russian', 'title');
```

Here:
tsvectorupdate – trigger name;
tvs – a column that stores values in tsvector format.

The reverse indexing algorithm is used in all systems. Because it allows you to speed up the search process, but the data will be blown away when converting the document to an index file. A tricky method is almost always used to save back indexed files, and the files are compressed. The mathematical model is used to perform a file search using reverse indexes (at the user's request) and to simplify the process of determining

the amount of participation in a request in all found request files. The more fully he answers this request, the higher the likelihood that search results will be available.
The main goal of the mathematical model is to search for the files necessary for the search query in the reverse index database, and then sort them in descending order of the availability of the query word in the files in accordance with the given query. The advantage of this indexing method is that there is no problem storing and updating indexes in a large database.

3. Proper indexing method

The difference between this method and the reverse indexing method is that it saves only those words that belong to the specified files in the index database. Reverse indexes store their location and the word for each word in the index database list. Proper indexes, on the other hand, store only words with a vector value. Proper indexing method really creates indexes quickly. It stores the obtained values in an array of type tsvector and uses these values to search for files in tsquery. For a small database, these indexes work well. However, such indexes are much slower due to the large amount of energy needed to re-read the data (or indexes) for a large database[5]. The command to build these indexes in PostgreSQL is as follows.

```
CREATE INDEX index_name
ON table_file
USING GIST
(to_tsvector(' russian '':: regconfig, (text || ' '::text) || title));
```

Here:
index_name – name of index to create;
table_file – table name;
GIST (proper index) – index type;
title, text – name of the columns in the table;
to_tsvector() - indexing function.
However, the order in which files are indexed is similar to reverse indexing, that is, compares a word with words in dictionaries, separates the part without suffixes, and combines the words. Inside the PostgreSQL database, the stop file contains the link words that must be removed from the text content during indexing. In addition to this file, we can include Uzbek connective words. This leads to an increase in search speed and a sufficient decrease in the index base. Figure 2 shows the file in which the connecting words in Russian are stored. During the indexing process, the words stored in this file are deleted from the text.

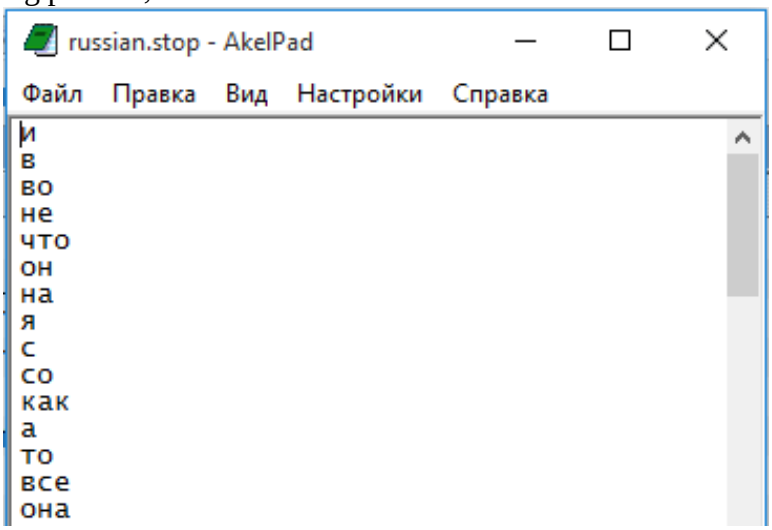


Figure 2.
In addition to this file, we will add connective words in the Uzbek language and draw the result. The result is shown in Figure 3.

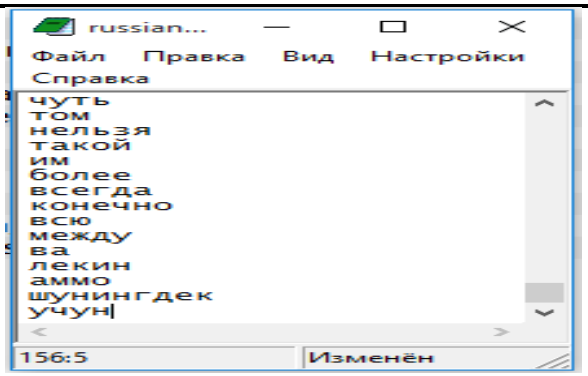


Figure 3.

After adding Uzbek connective words to the database of Russian connective words, we present the indexation result. Figure 4 below shows the result.

files text	tsvector_file tsvector
На этих примерах видно, что	'видн':4 'кажд':7 'массив':13 'област':10 'операц
ва мен лекин сен	'ва':1 'лекин':3 'мен':2 'сен':4
ва мен лекин сен	'ва':1 'лекин':3 'мен':2 'сен':4
в школу	'школ':2
ва мен лекин сен	'ва':1 'лекин':3 'мен':2 'сен':4
в школах и колледжах	'колледж':4 'школ':2
ва мен лекин сен	'мен':2 'сен':4

Figure 4.

In addition, triggers are created for these indices to automatically update values stored in the tsvector format. The command to create a trigger is as follows:

```
CREATE TRIGGER doc_trigger
BEFORE INSERT OR UPDATE
ON table_file
FOR EACH ROW
EXECUTE PROCEDURE tsvector_update_trigger ('ftvs', 'pg_catalog.russian', 'texts', 'title');
```

Here:

doc_trigger – trigger name;

ftvs – a column that stores values in tsvector format.

CONCLUSION

When building and using indexes, it is important to pay attention to the size of the database. In text search, it's convenient to use the proper indexes. Since the indices of this method store words and vector values in a file, we can get the desired result when performing a text search.

The article discussed the available advanced indexing algorithms. The results of the study can be used to index the database of electronic documents when creating an advanced search engine.

REFERENCES

1) N. A. Gaydamakin. Automated information systems, databases and banks. M.: Gelios ARV, 2002. 259-260p.
2) B-tree https://ru.wikipedia.org/wiki/B-дерево
3) Development of "Under the hood" indexes Postgres. https://habrahabr. ru/company /mailru /blog/261871/
4) Snippet, search back algorithm, page indexing and working features of Yandex http://joomla-s.ru/interesnye-stati/51-seo/uchimsya-nravitsya-yandeksu-i-google/158-cnippet-algoritm-obratnogo-poiska-indeksatsiya-stranits-i-osobennosti-raboty-yandeksa
5) Full text search in PostgreSQL. http://postgresql.ru.net/docs/fullsearch.html