IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

The Research and Application of Improved Search Method Based on Big Data

To cite this article: Xiaozhao Liu 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **452** 042021

View the article online for updates and enhancements.

The Research and Application of Improved Search Method Based on Big Data

Xiaozhao Liu *

Suzhou Institute of Trade and Commerce, College of Mechatronics and Information,
Suzhou, Jiangsu; 215009, China

*Corresponding author e-mail: zzz@zzz.com

Abstract. This paper improved an efficient search method to the problem of low efficiency for large data questions. Using shared history query results as a set of intermediate results, when a new query request arrives, analyze query request, provide keywords for user to choose and pick maser word and secondary word up. Master word match for historical inquiry is directly added to the matching portion of the historical results for directly as part of the new query result of the request if achieving matching. It can reduce the large number of double counting query history, save search time and improve query efficiency. By experimental comparison and analysis shows that big data based on improved query methods on can improve query efficiency.

Keywords: Big data; Algorithm; Search; Cloud database.

1. Introduction

Facing the huge amount of information and the need of accurate, efficient and personalized search information, researchers at home and abroad have put forward many search methods, platforms and models. Previous literatures mainly put forward the following algorithms: a set of intelligent text search technology including information retrieval, information extraction and information filtering; by analyzing the existing mainstream Chinese word segmentation algorithm and Lucene correlation sorting algorithm, an improved word segmentation algorithm and an improved correlation sorting algorithm are proposed; This paper presents a solution based on hardware search platform TCAM, which can realize real-time search of large text sets and enhance the ability of information processing. Although there are many new technologies or methods to deal with large data, it still takes a long time to extract records or web pages from large data to meet users' query needs. Interactive query is still a huge challenge. An Effective Search Method Based on Large Data Aiming at the inefficiency of large data query, this paper proposes that the shared historical query results should be used as the intermediate result set. When a new query request arrives, it first matches the historical query. If it can be matched, the historical query results of the matched part will be directly matched. As part of the new query request directly, a large amount of repeated computation of historical queries is reduced.

This paper improves the search method of large data. When a new query request arrives, it first analyzes the user's query request intention, provides keywords for the user to choose, determines the final keywords used by the user, then partitions the keywords, extracts the subject words and auxiliary words. Subject words are classified and matched with historical queries, and then the shared historical query results and the query results of new dates are combined. If there are auxiliary words, the historical query results are regarded as the intermediate result set, and the query is continued on this basis. If there are no auxiliary words, the query results are directly combined as the query results. The improved search method can save search time and improve query efficiency.

2. Improved search idea for big data

According to the new query request from the user, the user's query request intent is analyzed, the keywords are provided for the user to choose, the final keywords are determined, and the keywords are segmented, and the subject words and auxiliary words are extracted. Then, according to the subject words in the keywords, whether there is a precedent for the query or part of the query in the history record can be judged. If there is, the results of the history query can be shared, so as to reduce the time consumption of re-querying the whole collection of large data.

The concept includes the following sections:

Step 1 the user first asks for a query;

Step 2 analyzes the query intent of the query request, provides the query keywords with similar query intent for the user to choose, and obtains the final query keywords. The key words are segmented, and the key words and auxiliary words are extracted;

Step 3 match the query subject headings with the history query network. There are 3 cases in the matching result:

(1) Perfect match. If it is a perfect match, it shows that the new query request only has subject words, and the subject words have appeared before, so that the query results obtained by the previous query can be directly used by this query, so that the query results of the same query history can be shared. At the same time, since the history query is only a query for the data before a certain period of time, it is possible to generate new data records after the history query, so the new data still needs to be queried and new query results are obtained. The historical query results and the new query results are merged to get the final result that the user needs.

(2) Partial matching. If it is partially matched, the keywords of the new query request are extracted. The subject words section has appeared before, so that the same query results can be used directly for this query, which can share the query results of the same part of the query history. At the same time, because the history query is only a query for the data before a certain period of time, it is possible to generate new data records after the history query, so the new data still need to be queried to obtain new query results. The results of historical query and new query results are combined to get the result of query matching part. Then in the results of the query matching part, continue to execute the query of auxiliary words to get the final results required by the user.

(3) Mismatch. If the query is completely mismatched, it means that the new query request has no history query records to share, and all queries need to be re-executed to get the desired results.

Step 4 feed back the results of user queries to users;

Step 5 to update the history query network.

3. Key technologies of big data search after improvement

Under the improved concept of large data search, the following focuses on the concept of query intention of the query keywords to provide methods, query keywords and historical query network matching technology, new data query method and update method of historical query network and other key technologies.

3.1. Extract keyword query method based on query intention

User query requests are essentially a series of independent short texts. After receiving the user's query request, the system first analyzes the query intent of the query request, provides the query keywords with similar query intent for the user to choose, and obtains the final query keywords. Then the word segmentation is done. After word segmentation, a word vector is obtained, in which each word carries part-of-speech markers, such as nouns, verbs, adjectives, location words and other types. Words with different parts of speech have different contributions to the expression of the subject. Verbs and nouns play the most important roles in the expression and identification of the subject. Therefore, these two parts of speech can be considered in word frequency statistics, and other parts of speech can be ignored. Finally, the key words and auxiliary words are extracted from users' query requests.

3.2. New query request matching with historical query network

Figure 1 shows a schematic diagram of a new query request matching with the history query network.

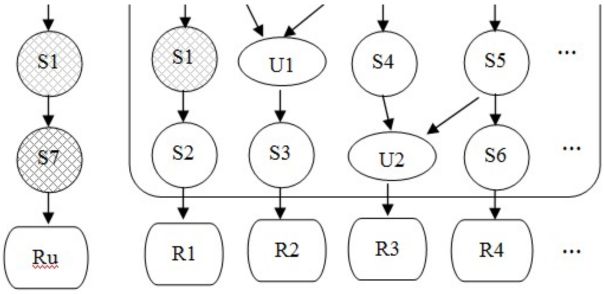

Fig. 1 matching diagram of new query request and history query network

Big data applications typically use cloud databases to store data records, such as Google's BigTable, Apache's Base cloud database, and so on.

The S1, S2, S3, S4, S5, S6, U1 and U2 in Figure 1 are all query conditions in the history query network. At the same time, the historical query network keeps all the historical query records in the nodesS1,S2,S3, S4, S5, S6, U1and U2 corresponding to the query conditions. Among them, S1,S2,S3,S4,S5 and S6 are single query result nodes, U1 and U2 are joint query result nodes, R1,R2,R3,R4 and Ruare final query result nodes. The key word extracted from the query request is S1, and the auxiliary word is S7, which belongs to the partial matching in the matching result. The key word S1 can match the results in the history query network. Therefore, matching analysis shows that S1 query nodes can share the historical query data result set. If new data records are generated after the historical query, it is necessary to query the newly added data to obtain new query results. The results of historical query and new query results are combined to get the result of query matching part. Then in the result of the query matching part, continue to execute the query of the auxiliary word S7, get the query result of Ru. The query utilizes the collection of historical query results, which reduces query time and improves query efficiency.

The matching algorithm of the improved query request and the history query network is as follows:

Input: New query requests (such as S1, S7), History Query Network; Output: New query requests and matching results of History Query Network. According to the query request, the user's query intent is analyzed, and the query keywords related to the query intent are provided to get the final user's query keywords. Matching with historical query network in cloud database;

Do case query keywords (e.g. S1; S1 and S7; S7);

Case is completely matched (e.g. S1);

In cloud database, S1 nodes get the historical query results matching S1;

Query the new data after the expiration date of the historical query results and get the new query results;

The final result is obtained by merging the historical query results with the new query results, returning to the user, and updating the content of the S1 node;

Case partial matching (such as S1 and S7)

In the cloud database, the S1 query results are partly matched with the S1 nodes;

Query the new data after the expiration date of the historical query results and get the new query results;

Merge historical query results and new query results to get the final result and update the contents of S1 nodes;

Continue query S7 in Ru, get the result Ru, return to the user, and record the content to the Ru node;

If there are still keywords, continue to match, get the result, return to the user, and record the content to the corresponding node;

Case does not match (for example, S7).

Query S7 in the cloud database, get the result, return to the user, and record the content to the S7 node.

End Case

3.3. New data query implementation method

Suppose that the cloud database table is used to store all records. An important feature of cloud databases is that when new records come in, all records are appended, so all new records are appended in the appended chronological order. This makes it easy to get additional records, that is, just find the last record in the history query at that time. Therefore, the number of newly added records can be easily obtained.

Suppose a user needs to look up all the automobile records in a site's microblog record this year, and someone has done the same in the past, but that person is looking up all the automobile records before May this year. As new microblog records continue to be appended, the history query can only be shared before May all the historical records, and after May new data needs to be re-queried.

Assuming that there are 60,000 microblog entries containing "cars" in all of the hundreds of millions of microblog entries in the first five months, all the time spent searching for 60,000 microblog entries containing "cars" from all the hundreds of millions of microblog entries in the previous five months can be saved by just searching for those published by users from May to today. Finding the number of microblog records that meet the criteria of "car" in tens of millions of records, assuming that 10,000 records meet the criteria, greatly reduces the search scope and can share the results of previous historical queries. Finally, 70 thousand conditions that satisfy the requirements are fed back to the user.

If you want to inquire about the number of microblog records of "automobile", then continue to inquire about the "automobile" condition in 70,000 items that meet the "automobile" condition, and get the final result.

3.4. Updating method of historical query network

In Figure 1, a relationship between the new query request and the history query network is analyzed. Analysis shows that some queries can share the query results, but there are still new query conditions S7 is not in the history query network, the updated query network as shown in Figure 2.

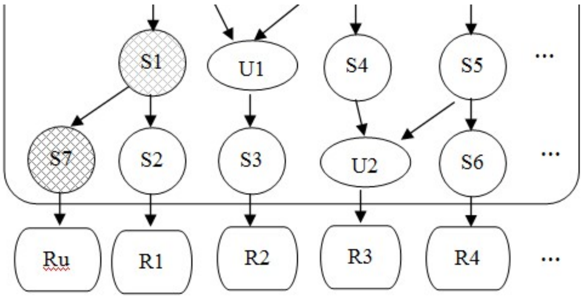

Fig. 2 updated query network

4. Simulation experiment

4.1. Building of experimental environment

Using Hadoop as the experimental environment, one Master node, one shadow node and two Slave nodes were used in four machines. The 4 machines install the Linux operating system and various components related to Hadoop.

4.2. Analysis of simulation results

1) Comparison of simple query search simulation experiments

Simulation Experiment 1 mainly verifies the time comparison between search single keyword (e.g. automobile) and multi-keyword (e.g. automobile) before and after improvement. The improved search method needs to segment the word first, the subject word is "automobile" and the auxiliary word is "automatic block". Matching the subject word "car" with the history query network, since there have been previous searches for "car" in history, only a new search is needed for the records in the new section directly. On the basis of the merged records of the historical query results and the newly added query parts, the "automatic block" search is carried out and the results are obtained. It can be seen that the improved search method only needs to search fewer search records. Therefore, the search time is greatly reduced and the query efficiency is improved.

2) Comparison of complex query search simulation experiments

Simulation Experiment 2 is mainly to verify the Join connection of two data sets. This simulation directly selects two tables from the data set for Join connection and complex query computation. It can be seen that the improved search method and the improved search method in a time-consuming comparison, from the comparison can be seen that our search method only needs to search for fewer records, reduce the search time, improve the query efficiency.

5. Conclusion

In view of the low efficiency of big data query, an effective search method is improved. The main improvement is that when the new query request arrives, it analyzes the user's query request intention, provides keywords for the user to choose, determines the final keywords used by the user, and then partitions the keywords, extracts the subject words and auxiliary words. By segmenting, the probability of matching success is increased, and the query results can be better utilized to quicken queries. Through the comparison and analysis of simulation experiments, the improved search method based on large data can better use the historical query results, reduce a lot of repeated calculation of historical query, save the search time and improve the query efficiency.

Acknowledgments

The Natural Science Foundation of the Jiangsu Higher Education Institutions of China(18KJB510042)

References

[1] Yuan J, Ding S. A method for detecting buffer overflow vulnerabilities:Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, 2011[C]. Xi'an: [s.n.], 2011:188-192.

[2] Ekanayake J, Li Hui, Zhang Bing-jing, et al. Twister: A Runtime for Iterative MapReduce[C].The First International Workshop on MapReduce and its Applications (MAPREDUCE' 10). 2010:110-119

[3] Bu Y Y, Howe B, Balazinska M, et al. HaLoop: Efficient itera-tive data processing on large clusters[J]. PVLDB.2010, 2010, 3(1/2):285-296