© 2016 IJSRST | Volume 2 | Issue 2 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Engineering and Technology

Efficient Sort Search on Massive Data

Nandhini A, Noureen P. T, Kanimozhi. R

Department of Computer Science and Engineering, Dhanalakshmi College of Engineering, Chennai, Tamil Nadu, India

ABSTRACT

Efficient top-N retrieval of records from a database has been an active research field for many years. The problem from a real world application point of view has the order of records according to some similarity function on an attribute is not unique. Many researchers have same values in several attributes and thus their ranking in those attributes is arbitrary (based on random choice).For instance, in large person databases many individuals have the same first name, the same date of birth, or live in the same city. Existing algorithms are ill-equipped to handle such cases efficiently. We introduce a Dynamic TMS searcher, which retrieves larger chunks of records from the sorted lists using fixed limits, and which focuses its efforts on records that are ranked high in more than one ordering and thus are more promising candidates. We experimentally show that our method outperforms Dynamic Sorting Algorithm (DSA) for top-k retrieval in those very common cases where we used with dynamically scheduling the resources based on the data which are provided with , this efficient short search algorithm along with the massive data retrieval on a very fine tuple data's can be of a different dataset. Here in this project we are going to use these logics for the need of solution in the field of medical research, where there are many manageable databases that are been used in a common path for the end of healthy need and the retrieval of solution for the cause of illness to a human being.

Keywords : Massive data, Indexing, Top-k retrieval, Dataset, Attribute, Sorted list

I. INTRODUCTION

Health care system is the organization of people institutions and resources that deliver health care services to meet the health needs of target populations. This project dea that are currently being formulated for the development of universal healthcare system in is about the plans.

India, which would provide universal health coverage throughout India. The scope of this project is to detect and identify group of moving objects travelling together for a certain time period. Here, we propose a series of techniques which address the searching and updating issues respectively in large databases.

II. METHODS AND MATERIAL

Related Works

A. Rank Aggregation

In Rank Aggregation, we have an approach to perform efficient similarity search and classification in high dimensional data. The database elements are considered as vectors in a Euclidean space and our goal is to find the database elements which are similar to the query. A small number of independent "voters'' rank the database elements similar to the query and these rankings are then combined by a highly efficient aggregation algorithm. This methodology has two techniques: To compute approximate nearest neighbours and to serve as a conceptually rich alternative to nearest neighbours. Nearest neighbours finds the best k matches % and also finds the 10 most similar images.

B. TOP-K Queries

We study the problem of answering top-k queries using views. The view in this context is a materialized version of a previously posed query, requesting a number of highest ranked values according to some monotone combining function defined on subset of the attributes of a relation. Several problems of interest arise in the presence of such views. We start by presenting a new algorithm capable of combining the information from a number of views to answer ad-hoc top-k queries.

C. Optimized Index Access

Top-k queries operate on index lists for a query's elementary conditions and aggregate scores for result candidates. One of the best implementation methods in this setting is the family of threshold algorithms, which aim to terminate the index scans as early as possible based on lower and upper bounds for the final scores of result candidates. This procedure performs sequential disk accesses for sorted index scans, but also has the option of performing random accesses to resolve score uncertainty. This entails scheduling for the two kinds of accesses: 1) The prioritization of different index lists in the sequential accesses, and 2) The decision on when to perform random accesses and for which candidates.

D. Optimal Aggregation

Assume that each object in a database has m grades, or scores, one for each of m attributes. For example, an object can have a color grade that tells how red it is and a shape grade that tells how round it is. For each attribute, there is a sorted list, which lists each object and its grade under that attribute, sorted by grade (highest grade first). Each object is assigned an overall grade that is obtained by combining the attribute grades using a fixed monotone aggregation function, or combining rule, such as min or average. To determine the top k objects, that is, k objects with the highest overall grades, the naive algorithm must access every object in the database, to find its grade under each attribute. Fagin has given an algorithm (''Fagin's Algorithm'', or FA) that is much more efficient. For some monotone aggregation functions, FA is optimal with high probability in the worst case. We analyze an elegant and remarkably simple algorithm (''the threshold algorithm'', or TA) that is optimal in a much stronger sense than FA.

E. Best Position Algorithm

The general problem of answering top-k queries can be modeled using lists of data items sorted by their local scores. The most efficient algorithm proposed so far for answering top-k queries over sorted lists is the Threshold Algorithm (TA). However, TA may still incur a lot of useless accesses to the lists. In this paper, we propose two new algorithms which stop much sooner. First, we propose the best position algorithm (BPA) which executes top-k queries more efficiently than TA. For any database instance (i.e. set of sorted lists), we prove that BPA stops as early as TA, and that its execution cost is never higher than TA. We show that the position at which BPA stops can be (m-1) times lower than that of TA, where m is the number of lists. We also show that the execution cost of our algorithm can be (m-1) times lower than that of TA. Second, we propose the BPA2 algorithm which is much more efficient than BPA. We show that the number of accesses to the lists done by BPA2 can be about (m-1) times lower than that of BPA. Our performance evaluation shows that over our test databases, BPA and BPA2 achieve significant performance gains in comparison with TA.

F. Existing System

The existing application provides query based table scan to retrieve the results. Algorithm used here is T2S-Table scan. It can maintain only a fixed number of tuples to compute results. They also use indexes with specific attributes to build the performance on view. The cost of pre- computing the data structures and update process is a major cost. It is difficult for a user to retrieve data by scanning each and every row from a table. This leads to poor efficiency in terms of time and performance. It is tedious task for the Government to maintain the entire database of the hospitals.

III. RESULT AND DISCUSSION

1. Proposed System

To overcome the disadvantages in the existing system, we proposed a system which will handle huge databases

in efficient manner and it uses "Dynamic sort search algorithm" to search and retrieve the efficient data very easily from databases. For example, the database will help to identify the patient's medical record. In addition, we have introduced a feature which is user-friendly as it depicts the concept of location based search. Here, users are allowed to prioritize his/herown choice of medical centres. This algorithm helps the Government to maintain the database of every patient in a nominal way.

2. Experimental Work

A. Authentication Management

Generally authentication phase involves registration of both users and administrators of medical centres. If an user wishes to register they are supposed to create an account which includes details such as username, password, residential address, contact number, alternative mail id and a special question must be answered for security purpose. The same procedure is applicable for the administrators of the medical centres but it requires additional information such as Registration number and treatments provided in the medical centres. The administrator must login using the registered mail id and password. If it is incorrect, authentication will be failed.

Once the authentication is successful, the admin has the rights to add details about their own medical centres. The user is allowed to access the details based on their need. For example, if the user is in need of any particular blood group, they can be able to check out the availability of blood and use it. The admin have the access to update the information whenever it is necessary.

B. Hospitality and Maintenance

The registration of a new hospital is mandatory to create account. Only after the registration, the admin of the hospital has able to access the system. Every hospital has to login and update the patient details in order to maintain proper database for every patient. Treatment handling process have to be maintain separately, that is every hospital has some certain treatments and that must be applied to facilitate the process. Every hospital has to maintain the record of patient primarily by date of admission and then the details have been feed based upon the type of treatment given to the patients.

At first, list of available blood groups and the available units on each blood group is been listed. Now, any hospital that requires the rare blood from another hospital can get the available unit of blood by a request procedure. Each and every hospital has to update the specialist (Doctor) available in their hospital and also specify the treatment they are handling.

C. Doctor's Schedule

There is no need for each and every specialist (Doctor) to set the limit of patients. The online users are allowed to set their appointment based on their treatment. Each hospital has its own list of specialist based on the treatment. Here, the specialist (doctors) schedules their available date, time for the treatment. This reduces the man power by not going to hospitals miles away. Patients can get their appointment in online. If the patient is in a critical situation then it is mandatory for the doctors to consider them the most and it is obligatory to change their fixed appointments.

D. User Appointment

The user who refers to a particular treatment has to schedule an appointment with the respective specialist (doctor). By doing so a confirmation for the scheduled appointment is also been generated. The User Report has to be collected by the user in physical after the course of the treatment. The user need not go miles away to make appointment. He/She can make it from their own place. They can make their appointments based on their comfort zone. The user should maintain his report for future purpose. The User must produce the report to their doctor periodically. This report will help the doctor to analyse the patient's medical history.

E. Report Generation

The administrator of this application will be having tie-up with government to provide all kind of medicines for registered hospital on demand.

Administrator will be able view and monitor and record of patients on daily basis for providing information for government. The analysed report by the admin must be produced to the Government on time. Future enhancement of medical centres is based on the reports of the admin.

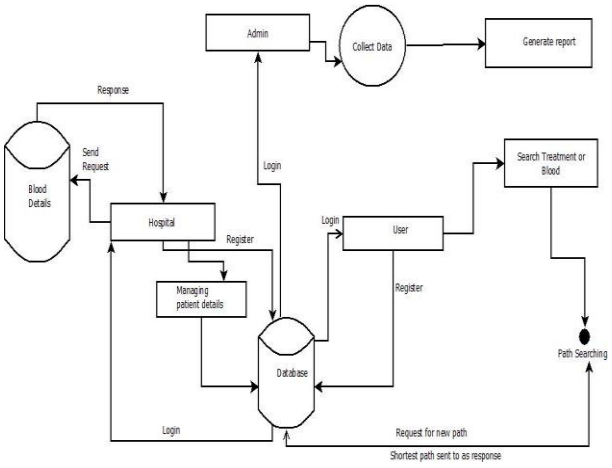
Improved medication can be provided to the patients. This report is substantial for the Government.

3. Algorithm Study

The nature of the search problem can be big in terms of the massive data search to make the efficient results provided for the end user. Each time when the auxiliary index becomes too large, we merge it into the main index. The cost of this merging operation depends on the resource needed and sometimes there may be resource engaged with another process this makes the result to be delay.

Indexing is based on two keys : primary key and secondary key. Primary key has single attribute, no duplicates whereas secondary key has one or more attributes % duplicates are allowed % indexing in M-dimensional feature spaces. Query may be match on exact match, partial match and range queries. In exact match, all attribute values are specified % with example name = "smith" and salary=30,000. Partial match are specified with example % name= "smith" and salary. Here not all attribute values are specified. Range queries have range of attribute values that are not specified as % with example name= "smith" and (20,000<=salary<=30,000) % find images within distance T%. Nearest neighbour finds the K best matches % and finds the 10 most similar images.

4. Architecture Diagram



IV. CONCLUSION

This paper proposes a novel DYNAMIC SORT SEARCH ALGORITHM for top-k retrieval of efficient data from huge databases. It produces rank based results which can be more efficient and effective to the user. In very common cases, we used the concept of dynamically scheduling the resources based on the data which are provided with this efficient sort search algorithm along with the massive data retrieval on a very fine tuple data's can be of a different dataset. Here in this project we used these logics for the need of solution in the field of medical research where there are many manageable databases that are been used in a common path for the end of healthy need and the retrieval of solution for the cause of illness to a human being.

V. REFERENCES

[1] R. Fagin, R. Kumar, and D. Sivakumar, "Efficient similarity search and classification via rank aggregation," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2003, pp. 301–312.

[2] G. Das, D. Gunopulos, N. Koudas, and D. Tsirogiannis, "Answering top-k queries using views." in Proc. 32nd Int. Conf. Very Large Data Bases., pp. 451–462, 2006.

[3] H. Bast, D. Majumdar, R. Schenkel, M. Theobald, and G. Weikum, "Io-top-k: Index-access optimized top-k query processing," in Proc. 32nd Int. Conf. Very Large Data Bases, 2006, pp. 475–486.

[4] R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," in Proc. 20th ACM SIGMOD-SIGACTSIGART Symp. Principles Database Syst., 2001, pp. 102–113.

[5] R. Akbarinia, E. Pacitti, and P. Valduriez, "Best position algorithms for top-k queries," in Proc. 33rd Int. Conf. Very Large Databases, 2007, pp. 495–506.