Dynamic Filtering Points Based Skyline Query Processing In Distributed Data Sites

M.Amuthavalli, Prof. P.S.Balamurugan

Abstract— The Skyline query used in a large number of unstructured distributed databases located at different data sites, it extracts the data from, where relevant data are distributed among geographically scattered sites. Constrained skyline query is attached with constraints on specific dimensions. A constraint on a dimension is a range specifying the user's interest. Given a skyline query with constraints, all relevant sites are partitioned into incomparable groups, among which the query can be executed in parallel based on partition algorithm. We then give a parallel distributed skyline algorithm to estimate the overall query response time. We have used some filtering points in distributed query processing such that the amount of data to be transmitted via the network connection is reduced. In this paper we propose dynamic filtering points to reduce the large number of filtering points used in a same data sites and get a good accurate result. In previous we used skyline only in centralized approach, now we are used skyline in distributed environment at larger network. The results of an extensive experimental study demonstrate that our proposals are effective and efficient.

Index Terms— Dynamic filtering points, partition algorithm, parallel distributed skyline algorithm

1 INTRODUCTION

Skyline is used in distributed database, because the database will not be in one system. It will be stored in multiple systems at different locations, if it is connected using internet. A Query is called as Skyline which query works or execute based on data points. Skyline query returns many multidimensional points. It extracts information from different places of distributed database at different sites. Skyline query returns all interesting points that are not dominated by any other points. Skyline queries play an important role in multi criteria decision making and user preference applications. For instance, a tourist can issue a skyline query on a hotel relation to get those hotels with high stars and cheap prices.

Most works on skyline queries so far have assumed a centralized data storage, and been focused on providing efficient skyline computation algorithms on a sole database. This assumption, however, fails to reflect the distributed computing environments consisting of different computers, which are located at geographically scattered sites and connected via Internet. Given a distributed environment without any overlay structures, our objective is efficient query processing strategies that shorten the overall query response time.

We first speed up the overall query processing by achieving parallelism of distributed query execution. Given a skyline query with constraints, all relevant sites are partitioned into incomparable groups among which the query can be executed in parallel. The parallel execution also makes it possible to report skyline points progressively, which is usually desirable to users. Within each group, specific plans are proposed to further improve the query processing involving all intra group sites.

The filtering points are used to communicate between the query and distributed sites. On a processing site, dynamic filtering points are calculatingly picked based on their overall dominating potential from the local skyline. They are then sent to other sites with the query request, where they help identify more unqualified points that would otherwise be reported as false positives, and thus, reducing the communication between data sites. We propose a specific partition algorithm that helps to extract information quickly by partition then it divides all relevant sites into groups such that a given query can be executed in parallel among all those site groups. Then it is making interrelated sites as a block. We elaborate on the intra group query execution strategies. We then give a parallel distributed skyline algorithm to estimate the overall query response time and it work parallel at different sites at a particular time. We detail heuristics for selecting a dynamic filtering point in distributed query processing such that the amount of data to be transmitted via the network is reduced. .

The filtering points which is created to remove unwanted data from skyline query database. In previous paper they propose a cost-efficient model for dynamically determining the number of filtering points to be sent to a particular site such that the benefit of using filtering points is maximized. In this paper we propose dynamic filtering points to use generate an efficient query in short time. We conduct an extensive experimental study on real data sets, and the results demonstrate the effectiveness, efficiency, and robustness of our proposals.

2 PROJECT OVERVIEW

In this project we use five modules to get a accurate result from distributed data sites and reduce the query response time.

2.1 SKYLINE QUERY WITH CONSTRAINTS

This is the first module of project which is concern with the process of forming skyline query. The module first need the dataset as input this input is processed and various rules are taken it consideration and these rules are identified. Based on condition of these rules the constraints are identified.

2.2 CONSTRAINTS FETCH DATASITES

The distributed database systems in each database are located as different places each place will be distributed. This distributed database is called as a dataset. From this data sites the skyline query would be executed. Each dataset identified by based on the database located at different distributed data places where a data sites are virtually found in finding the results for skyline query.

2.3 FORMATION OF SITES AND CONSTRAINTS

After identifying the constraints and sites the next important thing is to fit the constraints to the dataset and fetch the details in which a result will be skyline query. A formation of sites is based on different location and form of constraints is based on condition.

2.4 RESULT

After formation of conditions and data sites the is generated the results are based on the various points that is calculated point in order to filtering points for different skyline. Each data sites have multiple filtering points which based upon the skyline query. The filtering point review is the result is taken as group of results.

2.5 SELECT THE DATA

Once the result is got the next step is to identify and fetch the results of the query. The query results are got from the execution of the skyline query. The execution of the skyline query based on the data sites and the constraints give set of data that uses the select skyline query and fetches the result.

3 ABOUT THE DATASET

Skyline query processing is with respect to the process of distributed. A distributed database has data set interact which has fields of students that is taken from the fields of student name, college, place, percentage, sex. These fields play a important role in finding the result of queries. A dataset fields has relevant to a process of skyline query.

4 BACKGROUND STUDY AND RELATED WORK

Data Mining is a multidisciplinary field, drawing work from areas including database management systems, artificial intelligence, machine learning, neural networks, statistics, pattern recognition, knowledge-based systems, knowledge acquisition, information retrieval, high-performance computing, and data visualization.

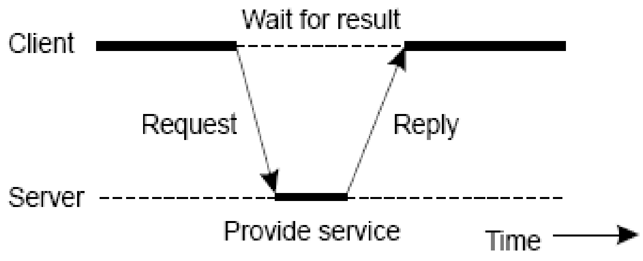

Fig.1 Basic Client–Server Model in distributed systems

Data mining extracts the information's from large distributed data bases it increase the query response time we introduce the skyline to distributed systems to reduce the overall query response time.

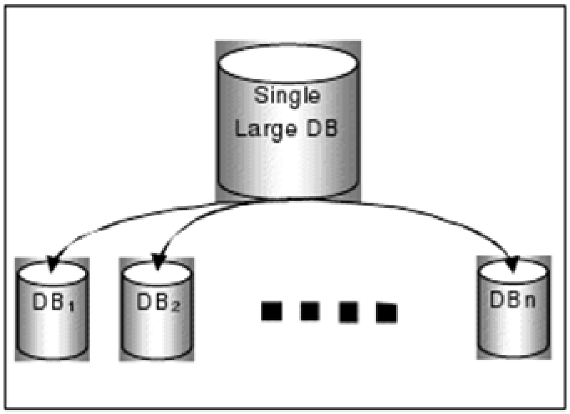

Fig.2 Partitioning of Database

5 PARALLEL DISTRIBUTED QUERY EXECUTION

Query execution order among different sites, the query with condition first asks each site Si for its MBRi, then-dimensional minimum bounding box of the local relation Ri. If a site's MBR disjoins with the constraint set C specified in the query, it will not be considered in the following query processing. For each site whose MBRi overlaps with C, we only need to consider the intersection MBRi \ C. We call this intersection reduced minimum bounding box and use rMBRi to represent it. We proceed to partition all those sites left into several groups according to their rMBRs such that the skyline computation in any one group does not depend on or affect the computation in any other group. Therefore, the given skyline query can be executed in parallel among those site groups. While within any individual site group, the query is executed according to some local plans, which will be Partition algorithm.A distributed database is not stored in its entirety at a single physical location. Instead, it is spread across a network of computers that are geographically dispersed and connected via communications links.

6 PROBLEM DEFINITION

Given a set of N sites S1; S2; . . . ; Sn distributed at different geographic locations, each Si has a local relation Ri. Every tuple in any Ri is an n-dimensional point, represented as ðp1, p2 . . . pnÞ. Different Ris may overlap; it is possible that Ri \ Rj Without loss of generality, we assume that smaller values are preferred in the skyline operator. We use pt1 _ pt2 to represent point pt1 dominates point pt2. In addition, we suppose any site Sorg, able to directly communicate with any other site Si 2 S through wired end-to-end connections, may initiate against all R is a skyline query with a set of n constraints C1, C2,. . . ; Cng. Each Ci is either a range [li; ui], or a indicating no constraint in that dimension. Our goal is to get the result for the constrained skyline query efficiently, i.e., with short response time. We define the query response time as the time period from the moment a query is issued by a site Sorg to the moment Sorg receives the complete and correct answers after contacting other sites. To shorten the response time of a query, we mainly endeavor on two aspects. On one hand, we propose effective ways to guide deciding the query forwarding and execution order between different sites, so as to obtain the query execution parallelism, and boost the result reporting progressiveness. On the other hand, we generalize the single filtering point idea to use dynamic filtering points, and thus, enhancing the filtering power and reducing the amount of data transmitted between remote sites. In contrast to previous work, their problem definition does not assume the availability of an overlay network, where different nodes hold disjoint data partitions. Instead, different relations on different sites may overlap. Therefore, their previous methods are not applicable to our problem.

7 QUERY PROCESSING IN DISTRIBUTED DATABASE

Query processing in a distributed database system requires the transmission f data between computers in a network. The arrangement of data transmissions and local data processing is known as a distribution strategy for a query. Two cost measures, response time and total time are used to judge the quality of a distribution strategy. Simple algorithms are presented that derive distribution strategies which have minimal response time and minimal total time, for a special class of queries.

Today's data is rarely stored in centralized location due to the enormous amount of information that needs to be stored and also to increase reliability, availability and performance of the system. Same data is stored in different format into different company's database as well as they may be partitioned or replicated. We consider various scenarios of distributed database such as horizontal, vertical fragmentation and attribute overlapping. Allowing access to integrated information from these multiple datasets can provide accurate and wholesome information to the end-user. We research on efficient querying to these distributed databases to get top k elements matching the ranking order provided by the user. We also discuss hierarchical way of using the top k algorithm and their limitations to our problem. In today's world of universal dependence on information systems, all sorts of people need access to companies' databases. In addition to a company's own employees, these include the company's customers, potential customers, suppliers, and vendors of all types. It is possible for a company to have all of its databases concentrated at one mainframe computer site with worldwide access to this site provided by telecommunications networks, including the Internet. Although the management of such a centralized system and its databases can be controlled in a well-contained manner and this can be advantageous, it poses some problems as well.

- ➢ Parallelized search
- ➢ Irrelevant nodes pruning
- ➢ Reduction of duplicate query forwarding

8 SKYLINE QUERY PROCESSING WITH RESPECT TO THE DATASET

Skyline query processing executed based on the data set. These data sets are stored in relevant distributed databases in different sites. First we give the skyline query with constraints all relevant sites are partitioned into incomparable groups among which the query can be executed in parallel.

The parallel execution also makes it possible to report skyline points progressively, which is usually desirable to users. Within each group, specific plans are proposed to further improve the query processing involving all intra group sites.

The filtering points are used to communicate between the query and distributed sites. On a processing site, dynamic filtering points are calculatingly picked based on their overall dominating potential from the local skyline. They are then sent to other sites with the query request, where they help identify more unqualified points that would otherwise be reported as false positives, and thus, reducing the communication between data sites.

We propose a specific partition algorithm that helps to extract information quickly by partition then it divides all relevant sites into groups such that a given query can be executed in parallel among all those site groups. Then it was making interrelated sites as a block. We elaborate on the intra group query execution strategies.

We then give a parallel distributed skyline algorithm to estimate the overall query response time and it work parallel at different sites at a particular time. We detail heuristics for selecting dynamic filtering points in distributed query processing such that the amount of data to be transmitted via the network is reduced. The filtering points which is created to remove unwanted data from skyline query database. In this paper we propose a dynamic filtering point to use generate a efficient query in short time. We conduct an extensive experimental study on both synthetic and real data sets, and the results demonstrate the effectiveness, efficiency, and robustness of our proposals.

9 CONCLUSION

In previous paper they used feed-back-based skyline algorithm for horizontal processing and multiple filtering points in distributed skyline query processing in different distributed data sites. It increases the query response time and transmission of network bandwidth size.

In this paper, we have addressed the problem of constrained skyline query processing against distributed data sites. To accelerate the query processing, we partition all relevant sites into incomparable groups and parallelize the query processing among all groups.

We select local skyline points and send them as dynamic filtering points together with the query to relevant data sites, in order to prevent more data from being transmitted through the network. Furthermore, a dynamic filtering point's selection strategy is proposed based on a PaDSkyline (parallel distributed skyline algorithm) to reduce the query response time and get effective query.

Extensive experimental results demonstrate the efficiency and effectiveness of our proposals in a distributed network environment.

REFERENCES

[1] W.-T. Balke, U. Guntzer, and J.X. Zheng, "Efficient Distributed Sky lining for Web Information Systems," Proc. Int'l Conf. Extending Database Technology (EDBT), pp. 256-273, 2004.

[2] P. Wu, C. Zhang, Y. Feng, B.Y. Zhao, D. Agrawal, and A.E. Abbadi, "Parallelizing Skyline Queries for Scalable Distribution,"Proc. Int'l Conf. Extending Database Technology (EDBT), pp. 112-130,2006.

[3] S. Wang, B.C. Ooi, A.K.H. Tung, and L. Xu, "Efficient Skyline Query Processing on Peer-to-Peer Networks," Proc. IEEE Int'l Conf. Data Eng. (ICDE), pp. 1126-1135, 2007.

[4] Y. Lu, J. Zhao, L. Chen, B. Cui, and D. Yang, "Effective Skyline Cardinality Estimation on Data Streams," Proc. Int'l Conf. Database and Expert Systems Applications (DEXA), pp. 241-254, 2008

[5] Y. Tao, L. Ding, X. Lin, and J. Pei, "Distance-Based Representative Skyline," Proc. Int'l Conf. Data Eng. (ICDE), pp. 892-903, 2009.

[6] L. Zhu, Y. Tao, and S. Zhou, "Distributed Skyline Retrieval with Low Bandwidth Consumption," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 3, pp. 384-400, Mar. 2009.

M.Amuthavalli, Received BE degree in 2010 from Anna University of technology Coimbatore and currently pursuing ME Degree at Karpagam University, Coimbatore

Prof. P.S.Balamurugan received BE degree in 2003 from Bharathiyar University Coimbatore and ME in 2005 from Anna University. Currently he is pursuing his Doctoral degree at Anna University, Coimbatore.