International Journal of Modern Trends in Engineering and Research
www.ijmter.com
e-ISSN: 2349-9745
p-ISSN: 2393-8161

Fragmentation as a Part of Security in Distributed Database: A Survey

Vaidik Ochurinda[1]
[1]*External Student, MCA, IGNOU.*

Abstract*:* The query optimization is a way of optimizing use of resources. In a way if query optimization is taken as a part to provide a security. Key Goal of this paper is to take fragmentation as part of security. This is a survey on techniques which can used to provide security through fragmentation.

Keywords –Query Optimization, Distributed Database System, Query Processing, Fragmentation

I. INTRODUCTION

Query optimization is a function of much relational database management system. Generally Query optimization is taken as how to divide a query from top and how data can be merged. Fragmentation is when a relation is broken up into smaller relations and stored in different sites. Relations can be either fragmented horizontally or vertically. In horizontal fragmentation, subsets of the relation are split up, and in vertical fragmentation columns of the relations are split up and stored at different sites. Fragmentation is preferable if locality is to be exploited. Fragmentation is a way of query optimization. When network is taken into consideration at that point we need to have some specific protocols which can protect these data, but fragmentation can itself provide security because this data is not a merged data rather it is scattered data which will be send from different sites and they will be merged at some intermediate place[1,2]

The whole scenario can work like data is located at different sites. When query is fired it will be divided and it will be send at different sites. These data will be merged at different sites. These sites can be intermediate sites. But merging of these data can me done as late as possible. So when data is on network at once it is not on the complete data rather a part of data. It will be merged at last site where data is to be used until that time all of the merging is differed.

II. FRAGMENTATION

Primary concern of distributed database system design is to making fragmentation of the relations in case of relational database or classes in case of object oriented databases, allocation and replication of the fragments in different sites of the distributed system, and local optimization in each site. Fragmentation is a design technique to divide a single relation or class of a database into two or more partitions such that the combination of the partitions provides the original database without any loss of information. This reduces the amount of irrelevant data

@IJMTER-2014, All rights Reserved

e-ISSN: 2349-9745
p-ISSN: 2393-8161

accessed by the applications of the database, thus reducing the number of disk accesses. Fragmentation can be horizontal, vertical or mixed/hybrid.

2.1. Horizontal fragmentation

Horizontal fragmentation (HF) allows a relation or class to be partitioned into disjoint tuples or instances. Intuition behind horizontal fragmentation is that Every site should hold all information that is used to query at the site and the information at the site should be fragmented so the queries of the site run faster[1-4].

2.2. Vertical Fragmentation

Vertical fragmentation (VF) allows a relation or class to be partitioned into disjoint sets of columns or attributes except the primary key. Each partition must include the primary key attribute(s) of the table. This arrangement can make sense when different sites are responsible for processing different functions involving an entity[5-9]. Objective of vertical fragmentation is to partition a relation into a set of smaller relations so that many of the applications will run on only one fragment.

Any Data located on different sites can be considered as fragmented data and these data can be considered for query optimization.

III. USE OF QUERY OPTIOMIZATION FOR SECURITY

Problem of passing data from relation is explained below in figure 3.

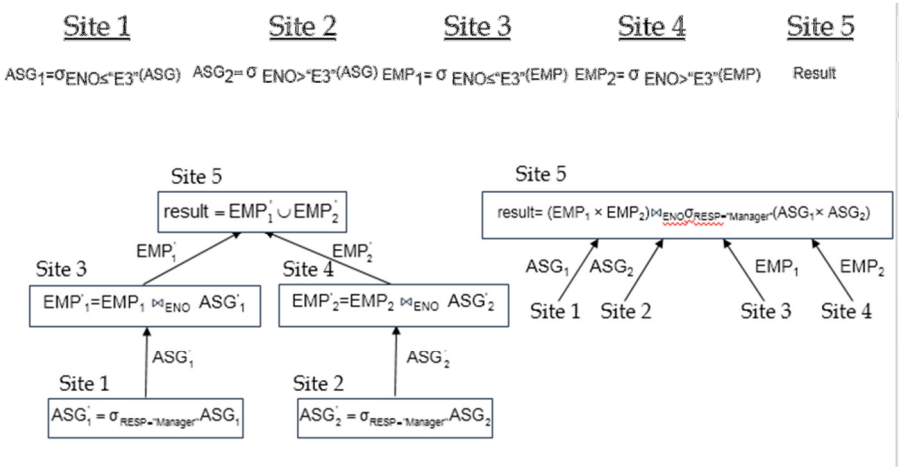

Figure 1: Process of execution in two different methodology

As mention in figure 3, there are five sites and result is to be sent at site 5. There can be two approaches for that. Approach 1(shown on left side), Calculate intermediate relations in each and every site and results of relations are sent to next site on hierarchy. In this manner, results are combined and propagated upward. These results finally reach to resultant site. This process incur execution time, communication cost, data transmission cost for only resultant data of each site.

@IJMTER-2014, All rights Reserved

e-ISSN: 2349-9745
p-ISSN: 2393-8161

Approach 2(shown on right side), send all the relations on resultant site and execution is done on resultant site. This process incur execution time, communication cost, data transmission cost for all relational data of each site [10,11].

If cost of both the approaches is compared then we come to know that approach 2 incurs more cost than approach 1[12].

IV. CONCLUSION

As seen above that fragmentation can be used as a base for data located on different sites. Now characteristics of these data can be used to use it as security purpose. This data can if inturpted than it needs other data to have actual meaning of it unless this partial data is useless. If this possibility is shorten and it is considered that only two sites are having data of interest then also half of the data is missing so this is a secure method.

References

[1]M. T. Ozsu and P. Valduriez, Principles of Distributed Database Systems, 2nd ed., New Jersey: Prentice-Hall, 1999.

[2]S. Ceri and G. Pelagatti, Distributed Databases Principles and System, 1st ed., New York: McGraw-Hill, 1984.

[3]S. Navathe, K. Karlapalem, and M. Ra, "A mixed fragmentationmethodology for initial distributed database design," Journal of Computer and Software Engineering Vol. 3, No. 4 pp 395–426, 1995.

[4]F. Bai˜ao, M. Mattoso, and G. Zaverucha, "A distribution design methodology for object DBMS," Distributed and Parallel Databases, Springer, Vol. 16, No. 1, pp. 45–90, 2004.

[5]S. Ceri, M. Negri, and G. Pelagatti, "Horizontal data partitioning in database design," in Proc. ACM SIGMOD, 1982, pp. 128–136.

[6]S. B. Navathe, S. Ceri, G. Wiederhold, and J. Dour, "Vertical partitioning algorithms for database design," ACM Transactions on Database Systems (TODS), Vol. 9, No. 4, pp. 680–710, 1984.

[7]D. G. Shin, and K. B. Irani, "Fragmenting relations horizontally using a knowledge based approach," IEEE Transactions on Software Engineering (TSE), Vol. 17, No. 9, pp. 872–883, 1991.

[8]M. Ra, "Horizontal partitioning for distributed database design," In Advances in Database Research, World Scientific Publishing, pp. 101–120, 1993.

[9]C. H. Cheng, W. K. Lee, and K. F. Wong, "A genetic algorithm-based clustering approach for database partitioning," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 32, No. 3, pp. 215–230, 2002.

[10]Cyrus Shahabi, Latifur Khan, Dennis Mcleod. A probe based technique to optimize join queries in distributed internet bases, Knowledge and Information Systems.2000, 2.

[11]TSAIPSM, CHENALP, Optimizing queries with foreign function in a distributed environment, IEEE Trans on Knowledge and Data Eng, 2002, 14(2):809-824.

[12]J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 1986, pp. 679-698.

@IJMTER-2014, All rights Reserved