ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 10, October 2016

Optimal Solution for Fragment Allocation in Distributed Database

[1] M.Kavitha [2] Y.Pavithra [3] P.Gayathri [4] Chavala P V S Prudhvi
[1] Associate Professor [2][3][4] Student
[1][2][3][4]Dept of CSE, Sri Venkateswara College of Engineering, Nellore

Abstract: **-- The distributed data processing is an upstanding way to improve reliability, availability and pursuance of a database system. In this paper we will concentrate on data allocation problem with the aim to persuade an optimal distribution of data in the process of the distributed database architecture in interconnection with data fragmentation. Efficient allocation of fragments requires a proportion between costs, performance and data distribution restrictions. The allocation of fragments is closely related to the replication of data from distributed databases. In addition, we analyzed the cost of fragmentation and replication.**

***Key words and phrase*: distributed databases, fragmentation architecture, allocation architecture, strategies, methods, cost analysis.**

I. INTRODUCTION

Definition: A distributed database system consists of a collection of sites, connected together via some kind of communications network, in which:

1) Each site is a full database system site in its own right, but
2) The sites have agreed to work together so that a user at any site can access data anywhere in the network exactly as if the data were all stored at the user's own site.

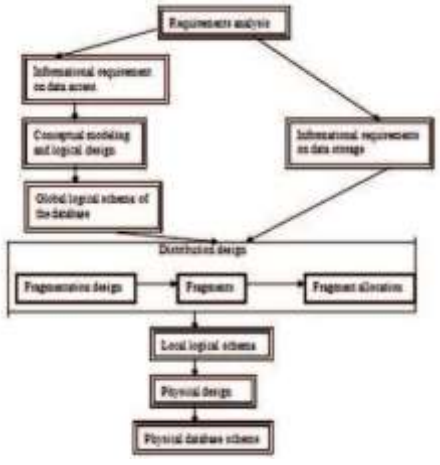
Definition: It follows that a distributed database (DDB) is really a kind of virtual database, whose component parts are physically stored in a number of distinct "real" databases at a number of distinct sites (in erect. it is the logical union of those database) [4].

Definition: A distributed related database management system (DDBMS) is a software system that manages a distributed database while making the distribution transparent to the user [1]. Distribution is normally discussed solely in terms of the fragmentation and replication of data. A data Definition: A distributed database system consists of a collection of sites, connected together via some kind of communications network, in which:

1) Each site is a full database system site in its own right, but
2) The sites have agreed to work together so that a user at any site can access data anywhere in the network exactly as if the data were all stored at the user's own site.

Definition: It follows that a distributed database (DDB) is really a kind of virtual database, whose component parts are physically stored in a number of distinct "real" databases at a number of distinct sites (in erect. it is the logical union of those database) [4].

Definition: A distributed related database management system (DDBMS) is a software system that manages a distributed database while making the distribution transparent to the user [1]. Distribution is normally discussed solely in terms of the fragmentation and replication of data. A data



2.1. Fragmentation design

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 10, October 2016

Definition 2.1. Fragmentation. The system partitions the relation into several fragments, and stores each fragment at a different site. If a distributed database B is partitioned this means that the parties P1; P2; :::; Pj, i 2 I, I = 1; 2; :::; n which compose her form disjoint subsets:

B = fP1 [P2 [::: [Pjg,with fP1 \ ::: \ Pjg = ;:

The fragmentation is the partitioning of a global relation R into fragments R1; R2; :::; Ri, containing enough information to reconstruct the original relation R.

There are three basic rules that should be looked at during the fragmentation, which ensure that the database does not have semantic changes during fragmentation, i.e. ensure consistency of the database Completeness. If relation R is decomposed into fragments R1; R2; :::; Rn, each data item that can be found in R must appear in at least one fragment. Reconstruction. It must be possible to define a relational operation that will reconstruct R from the fragments. Reconstruction for horizontal fragmentation is Union operation and Join for vertical. Disjointness. If data item di appears in fragment Ri, then it should not appear in any other fragment.

Exception: vertical fragmentation, where primary key attributes must be repeated to allow reconstruction. In the case of horizontal fragmentation, data item is a tuple; for vertical fragmentation, data item is an attribute.

2.1.1. Fragmentation strategies. Consider a relation with scheme R. The fragmentation of R consists of determining the number of fragments (subschema) Ri obtained by applying an algebraic relation on R (as operations on relations which show the logical properties of data). In this context, the fragmentation of data collection can be done in two ways:

a) Horizontal. The horizontal fragmentation of a relation R is the subdivision of its tuples into subsets called fragments, the fragmentation is correct if each tuple of R is mapped into at least one tuple of the fragments (completeness condition). An additional disjointness condition, requiring that each tuple of R be mapped into exactly one tuple of one of the fragments, is often introduced in distributed database systems in order to control the existence of duplication explicitly at the fragment level (by having multiple copies of the same fragment). The resulted fragments Ri have the same scheme structure as well as collection R, but differ by the data they contain and are resulted by applying a selection on R.

Selection op(R) - denies a relation that Qcontains only those tuples of R that satisfy the specified condition (predicate p): op(a1; :::; an(R)). A horizontal fragment can be obtained by applying a restriction: Ri = gcondi(R). So we can rebuild the original relation by union as follows: R = R1 [R2 [::: [Rk.

For example:

R1=3/4type=House (Property For Sale)

R2=3/4type=Flat (Property For Sale)

There are two versions of horizontal partitioning: primary horizontal fragmentation of a relation is ac ieved through the use of predicates denied on that relation which restricts the tuples of the relation. derived horizontal fragmentations is realized by using predicates that are denied on other relations.

b) Vertical. It divides the relation vertically by columns. The resulted fragments Ri contain only part from the collection structure R. It keeps only certain attributes at certain site and they contain the primary key of the relation R to ensure that the restore is possible and are resulted from the application of a projection operation of relational algebra: Q a1; :::; an(R), where a1; :::; an are attributes of the relation R.

The fragmentation is correct if each attribute of the relation is mapped into at to be in the least one of the attribute of the fragments; moreover, it must be possible to reconstruct the original relation by joining the fragments together R = R1 N R2 N ::: N Rn; in other words, the fragments must be a lossless join decomposition of the relation.

For example:

R1 = Q staff No, position, salary (Staff)

R2 = staff No, ¯first Name, last Name, branch No (Staff)

c) Sometimes, only vertical or horizontal

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 10, October 2016

fragmentation of a database scheme is in-sufficient to distribute adequately data for some applications. Instead it can be useful to be used to be implement mixed or hybrid fragmentation. A mixed fragment from a relation consists of a horizontal fragment that is vertically fragmented, or a vertical fragment that is horizontally fragmented. A mixed fragmentation is denied using selection and projection operations of relational algebra: ¾p (Q a1; :::; an(R)) or Q a1; :::; an(¾p(R)).

The comparison of the related fragmentation strategies is to be showed in the table

1. for each fragment of a relation R:
Condition C = True (all tuples are selected).
List (L = ATTRS(R)) = True
Security - data not required by local applications is not restored, and consequently not available to unauthorized users;
Performance of global applications that require data from several fragments located at different sites many be slower.

2.2. Allocation design.

A database is named distributed if any of its tables are stored at dire rent sites; one or more of its tables are replicated and their copies are stored at dire rent sites; one or more of its tables are fragmented and the fragments are stored at deferent sites; and so on. In general, a database is distributed if not all of its data is localized at a single site [6].

Definition 2.2. Replication

The system maintains several identical replicas (copies) of the relation, and stores each replica at a different site. The alternative to replication is to store only one copy of relation r.

3. Cost analysis

Assume the set of sites is S = fS1; S2; :::; Spg. Let p be the total number of sites, NRel be the total number of relations, m be the total number of fragments, Nfrag be the cardinality of the fragmented relation, n be the number of fragments of the fragmented relation, Nrep be the cardinality of each other (NRel ¡ 1) replicated relations, Njoin be the cardinality of the joined relations, k be the number of attributes in both fragmented and replicated relations, Kjoin be the number of attributes after joining the relations from any site, Kp be the number of attributes to be projected,

CT comp be cost per tuple comparison, CTconc be the cost per tuple concatenation, Tcost¡attr be transmission cost per attribute, TCR be the replication transmission costs, CPp¡attr be the cost per projected attribute.

4. Conclusion

The objective of a data allocation algorithm is to determine an assignment of fragments at different sites in order to minimize the total data transfer cost involved in executing a set of queries. This is equivalent to minimization of the average query execution time, which has a primary importance in a wide area of distributed applications. The fragmentation in a distributed database management system increases the level of concurrency and therefore system throughput for query processing. Distributed databases have appeared as a necessity, because they improve availability and reliability of data and assure high performance in data processing by allowing parallel processing of queries, but also reduce processing costs.

REFERENCES

[1] R. Elmasri and S. Navathe, Fundamentals of database systems (4th ed.), Boston: Addison-Wesley, 2004

[2] N.M. Iacob (Ciobanu), The use of distributed databases in e-learning systems, Procedia Social and Behavioral Sciences Journal 15 (2011), 3rd World Conference on Educational Sciences - WCES 2011, Bahcesehir University, Istanbul - Turkey, 03-07 February 2011, 2673{2677w York: Palgrave-Macmillan, 2004.

[3] P. Beynon-Davies, Database systems (3rd ed.), New York: Palgrave-Macmillan, 2004.

[4] C.J. Date, An introduction to Database Systems (8th ed.), Addison Wesley, 2003

[5] S. Rahimi and F.S. Haug, Distributed database management systems: A Practical Approach, IEEE,

Computer Society, Hoboken, N. J: Wiley, 2010.

[6] M.T. Ozsu and P. Valduriez, Principles of Distributed Database Systems (3th ed.), New York: Springer, 2011.

[7]A. Silberschatz, H.F. Korth and S. Sudarshan, Database System Concepts (6th ed.), McGraw-Hill, 2010.