ISSN: 2393-8528

IJICSE
Open Access

Contents lists available at www.ijicse.in

International Journal of Innovative Computer Science & Engineering

Volume 3 Issue 2; March-April-2016; Page No. 47-50

Fragmentation Techniques for Distribution Database: A Review

Sunil Kumar Verma

Himalayan University, Itanagar, Arunachal Pradesh (India)

iqrasoftware@gmail.com

ARTICLE INFO

Received: 25 April. 2016
Accepted 10 May 2016

Corresponding Author:

Sunil Kumar Verma

Himalayan University, Itanagar, Arunachal Pradesh (India)

Key words: Fragmentation, Centralized and Distributed Database System, Hybrid fragmentation.

ABSTRACT

The wide acceptance of the relational approach in data-processing applications and the continuous improvement of commercially existing relational databases has increased the interest for using database in non-conventional applications, such as computer-aided design (CAD), geographic information systems, image, and graphic database systems. The Centralized and distributed database systems are developed for balancing the load and scattering the data over different sites on an organization. So in order to distribute the database on different sites of an organization, fragmentation methods are used. There is several fragmentation methods reviewed in this paper.

INTRODUCTION

Information storage has been a challenging endeavor throughout human history and existed even before modern computer systems [1]. The last three decades are marked by rapid growth of computer technology. This has raised the needs to evolve new techniques to manage huge amounts of data. Today, mostly centralized databases are used to store and manage data [2]. They carry the advantages of high degree of security, concurrency and backup and recovery control. However, they also have disadvantages of high communication costs (when the client is far away and communication is very frequent), unavailability in case of system failure and a single source bottleneck [3]. These issues raise the need of distribution of databases over various systems or locations. But the main motivation behind the concept of distributed databases is the efficient management of huge amount of data with increased availability and reduced communication cost. Research conducted in 1991 for distributed databases predicted a huge shift from traditional databases to distributed databases in the coming arena primarily due to organizational needs to manage huge amounts of data [2]. According to Ilker, many applications in the future will be distributed due to the development in technology, and therefore the databases will also be distributed [3]. The telecommunication sector also wants to embrace this technology of data distribution. But before distribution, fragmentation is a very important and critical task that needs to be done. The fragmentation strategy is chosen carefully according to the database model. Fragments are made and then they are distributed to the sites where needed. Recent developments in network technologies have increased the intensity of data in telecommunication services. So the telecommunication sector is now at a point where they should move to fault-tolerant, more reliable and less expensive means of distributed data management [4]. But this emerging field has very critical issues to be dealt with before implementation in the telecom industry. Primarily these issues are: concurrency control, availability, transparency, recovery and throughput etc. Compared with the use of distributed databases in other traditional industries, telecommunication databases need to fulfill some additional and tough requirements on availability, recovery, throughput, and reliability generally and response time specifically [2]. In the given context response time is of special importance before implementing distributed database at Ericsson. A lot of research has been done on distributed databases in general; little is, however, done with special consideration to the telecom environment [5]. In our thesis we present a state of the art report about the response time and issues related to it including effect of increase in data and impact on the response time when data is not locally present.

II. BACKGROUND THEORY

The purpose of this paper is to present some ground knowledge to databases, their types and related

issues.

A. DATABASE SYSTEM:

The 20th century witnessed a lot of development in databases to meet the needs of information storage and retrieval. It started with simple file systems, stored individually at separate places. However, it was not very fruitful for huge amounts of data due to problems of redundancy, separation and isolation of data, data dependence and incompatible file formats [6]. Later these files were merged into one unit and in 1964 the term "database" was coined for this unit. Advantages associated with the database were faster and shared access, data integrity and data consistency etc. Later different database architectures were devised according to the need of different applications. IBM"s IMS (Information Management System) in 1966 was the first commercial hierarchical database system [7]. It provided the features of easy data updating, fast retrieval, multiple association and simplicity of structure. However, pointer path restrict access, data replication and large computer storage made the scope of hierarchical databases very limited. Object oriented database came out to support graph-structured objects. They incorporated many useful features of object oriented paradigms as encapsulation, inheritance and object identity. However their complexity of model and lack of standard limited their use. One main problem with these databases was the intermingling of conceptual relationships with the physical storage. Though it provided efficient access it reduced flexibility. The solution to this problem emerged in the inception of relational databases by E.F.Codd in 1970 [7]. This database model is based on the mathematical principles of set theory and predicate logic. It provided faster access by joining multiple tables using relations. Relational databases are now present on almost every type of computers.

B. CENTRALIZED DATABASES:

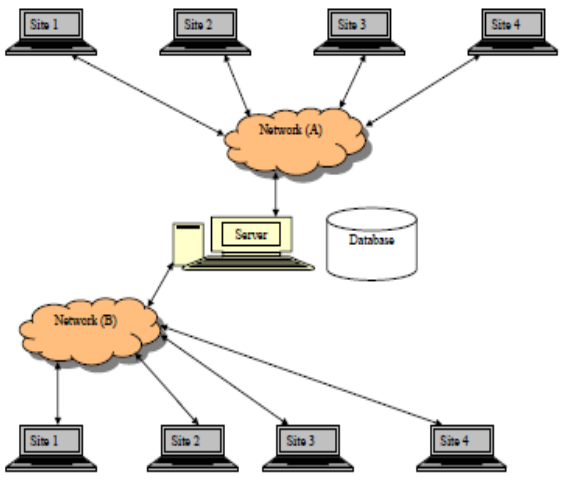

Figure 1: Centralized Database system Architecture

A centralized database system is a system where all the data is stored at a single location. Users retrieve the data from that database, perform some actions and store it again at that location. Making your database centralized carry many advantages including easy data management, high degree of security, concurrency, and backup and recovery etc. The figure 1 depicts the general architecture of centralized database system for two countries (A, B); all sites can access the data from the central database server via their network. The obvious advantages of this system include data consistency, security, and easy management.

C. DISTRIBUTED DATABASES:

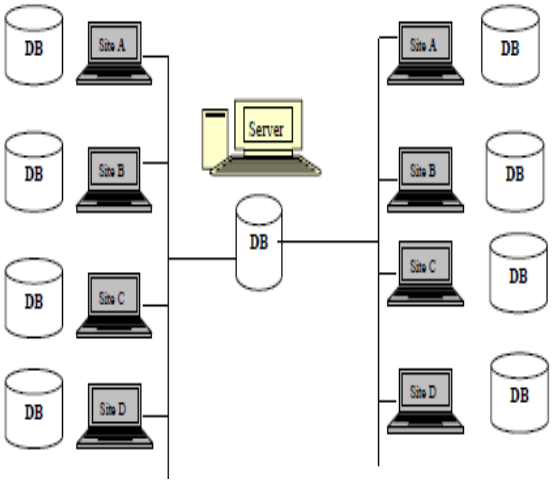

Figure 2: Distributed Database system Architecture

"*A distributed database is a collection of multiple, logically interrelated databases distributed over a computer network*" [2]. It may also be a single database divided into chunks and distributed over several locations. The database is scattered over various locations which provides local access to data and thus reduces communication costs and increases availability. Most of today's business applications have shifted from traditional processing to online processing.

Generally, distributed database is the collection of databases distributed across different locations or sites over a network as illustrated in Figure 2. Similarly, it may also be a single database, divided into chunks and distributed over various sites. Each site has a certain amount of data that it needs frequently and it can get the rest from some other site. Distributed databases are very useful when availability and fast response time is needed. They increase performance and reduce communication costs. However, many serious issues are also related with distributed databases which must be investigated. Among those issues, consistency, concurrency control and query optimization.

III. FRAGMENTATION

It is the decomposition of a relation into fragments each being treated as a unit [2]. Fragmentation is done according to the data selection patterns of applications running on the database. It permits to divide a single query into a set of multiple sub-queries that can execute parallel on fragments. Fragmentation can be of any type: horizontal, vertical and hybrid/mixed.

A. VERTICAL FRAGMENTATION:

It divides a relation into fragments which contain a subset of attributes of a relation along with the primary key attribute of the relation. The purpose of vertical fragmentation is to partition a relation into a set of smaller relations to enable user applications to run on only one fragment [2].

Name	Reg.No.	Course	Dept
Fragmentation1	Fragmentation2	Fragmentation3	

Figure 3: Vertical Fragmentation

B. HORIZONTAL FRAGMENTATION

It divides a relation into fragments along its tuples. Each fragment is a subset of tuples of a relation. It identifies some specific rows based on some criteria and marks it as a fragment.

Name	Reg.No.	Course	Dept
Fragmentation1			
Fragmentation2			
Fragmentation3			
Fragmentation4			

Figure4: Horizontal Fragmentation

Horizontal fragmentation is further divided into two types.

Primary horizontal fragmentation

This type of fragmentation is done where the tables in a database are neither joined nor have dependencies. So, no relationship exists among the tables.

Derived horizontal fragmentation

Derived horizontal fragmentation is used for parent relation. It is used where tables are interlinked with the help of foreign keys. It ensures that the fragments which are joined together are put on the same site. In this thesis, derived horizontal fragmentation is used.

C. HYBRID/MIXED FRAGMENTATION

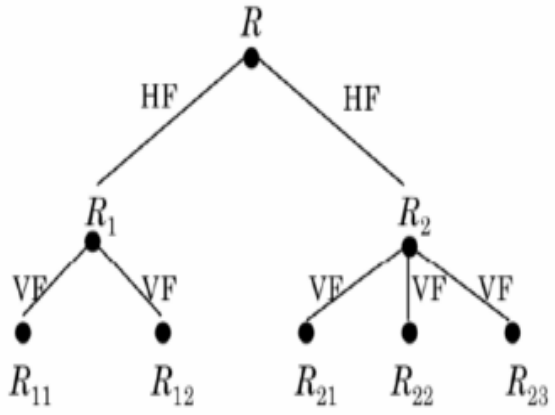

Figure 5: Hybrid/Mixed Fragmentation

The Mixed/Hybrid fragmentation is Combination of horizontal and vertical fragmentations. This type is most complex one, because both types are used in which horizontal and vertical fragmentation of the DB application. The original relation is obtained back by join or union operations.

The Mixed/Hybrid fragmentation is Combination of horizontal and vertical fragmentations. This type is most complex one, because both types are used in which horizontal and vertical fragmentation of the DB application. The original relation is obtained back by join or union operations.

Name	Reg.No.	Course	Dept
Fragmentation 1		Fragmentation 2	
Fragmentation 3	Fragmentation 4		

Figure 6: Mixed Fragmentation Chart

In this report, the data mining fragmentation technique is used to improve the performance of the distributed DW system and reduces the execution time of queries. Here, allocation process and queries process are also involved. The allocation process allocates the data on the sites in network and maintains the replication of data. Queries are used to increase the accessing speed of data from the tables. Fragmentation, Allocation and queries improve the efficiency and performance of the system.

IV. CONCLUSION

It is important to manage an appropriate methodology for data fragmentation in order to utilize the resources and thus it is must to select an accurate and efficient fragmentation methodology to enrich the power of distributed database system.

REFERENCES

1. Ebrary Inc. 1999. *Funding a revolution [Elektronisk resurs] government support for computing research*. National Academy Press, Washington, D.C.
2. Özsu, M.T. And Valduriez, P. 1999. *Principles of distributed database systems*. Prentice Hall, Upper Saddle River, N.J.
3. Köse , I. Spring , 2002. Distributed Database Security.
4. Jonker, W. 2000. *Databases in telecommunications: international workshop co- located with VLDB-99, Edinburgh, Scotland, UK, September 6th 1999 : proceedings*. Springer, Berlin.
5. Cooper, B.F., Ramakrichnan, R., Srivastava, U., Silberstein, A., Bohannon, P., Jacobsen, H.-A., Puz, N.,Weaver, D. And Yerneni, R. August 2008. PNUTS: Yahoo!'s hosted data serving platform.
6. Connolly, T.M. And Begg, C.E. 1998. *Database systems: a practical approach to design, implementation and management*. Addison-Wesley, Harlow.
7. Vaughn 14 Nov, 2003. CSPC 343: *A Sketch of Database History*.