Chapter 26
Trends in Improving Performances in Distributed Database Management Systems

Ismail Omar Hababeh
United Arab Emirates University, UAE

Muthu Ramachandran
Leeds Metropolitan University, UK

ABSTRACT

Database technology has been a significant field to work in for developing real life applications in network information systems. An enterprise’s reliance on its network and database applications in Distributed Database Management systems (DDBMS) environment is likely to continue growing exponentially. In such a system the estimation and prediction of Quality of Service (QoS) performance improvements are crucial since it increases understanding the issues that affect the distributed database networking system behaviour; like database fragmentation, clustering database network sites, and data allocation and replication that would reduce the amount of irrelevant data and speed up the transactions response time. This chapter introduces the trends of database management systems DBMS and presents an integrated method for designing Distributed Relational networking Database Management System DRDBMS that efficiently and effectively achieves the objectives of database fragmentation, clustering database network sites, and fragments allocation and replication. It is based on high speed partitioning, clustering, and data allocation techniques that minimize the data fragments accessed and data transferred through the network sites, maximize the overall system throughput by increasing the degree of concurrent transactions processing of multiple fragments located in different sites, and result in better QoS design and decision support.

1. INTRODUCTION

The database management systems (DBMS) are now such an integral part of the organizations daily life that have major roles in designing, analysing and developing real world applications, and have major effects in data control to achieve best system performance.

This chapter aims to introduce the literature review in the design issues of the distributed database
management systems (DDBMS) that were seen to be relevant to particular areas of database fragmentation, sites clustering, data allocation, query processing, metrics and QoS. New methods in this context are proposed and investigated against the available methods in the literature.

There have been a lot of models, programs, and tools in the literature used to describe, execute, and implement the operations in the databases. Therefore, setting up the definition for each field may help to distinguish between different types of DBMS and to categorize them accordingly.

In this context, the database, database management system (DBMS), database application, and database system are defined by Connolly (2004):

- **Database**: collection of related data
- **Database Management System (DBMS)**: software that manages and controls access to the database.
- **Database Application**: program that interacts with the database at some point in its execution.
- **Database System**: collection of application programs that interact with the database along with the DBMS and database itself.

Too many enterprises consider the distributed databases as a best platform for their services that spread over a wide geographical area. Most recent database researches have focused on DDBMS because of the high cost of the network connectivity and impracticality of the centralized database systems.

In addition, Oszu and Valduriez (1991) have defined the distributed database as a collection of multiple, logically interrelated databases distributed over a computer network. On the other hand, Hoffer, Prescott, and McFadden (2004) define the distributed database as single **logical database** that is spread physically across computers in multiple locations that are connected by a data communications link, where it is different from decentralized databases which are defined as a collection of independent databases on non-networked computers.

Traditionally, DBMS have been classified according to the database environment, data model, and user-computer systems basis. Hoffer, Prescott, and McFadden (2004) have distinguished two main types of DBMS in a distributed database environment:

- **Homogeneous**: The same DBMS is used at each node. This type is easy to manage but difficult to enforce. It can be divided into two subtypes; **Autonomous** in which independent DBMS are used and **Non-autonomous** where central and coordinating DBMS are considered together.

- **Heterogeneous**: The different database management systems are used at different nodes. This type is difficult to manage and preferred by independent organizations. Two subtypes can be distinguished in this database environment; Systems which can be divided into full DBMS functionality or **partial-multidatabase** and Gateways where simple paths are created to other databases without the benefits of one logical database. Partial-multidatabase consists of **federated** type which is categorized by loose and tight integrations, and **unfederated** type that allows remote access.

Alternatively, (Graham 2005) has categorized DBMS based on their data model into the following groups:

- **Hierarchical**: In this model data are stored as a series of records (e.g. IBM’s IMS). Although the hierarchical model is no longer used as the basis for current commercially produced systems, there are still legacy systems in existence based on this model.
Related Content

**Extraction of an Architectural Model for Least Privilege Analysis**
[www.igi-global.com/article/extraction-architectural-model-least-privilege/74843?camid=4v1a](www.igi-global.com/article/extraction-architectural-model-least-privilege/74843?camid=4v1a)

**A Method to Support Fault Tolerance Design in Service Oriented Computing Systems**
[www.igi-global.com/chapter/method-support-fault-tolerance-design/66808?camid=4v1a](www.igi-global.com/chapter/method-support-fault-tolerance-design/66808?camid=4v1a)

**Designing User-Defined Modeling Languages with SIMTHESys**
Mauro Iacono (2014). *Theory and Application of Multi-Formalism Modeling* (pp. 196-237).
[www.igi-global.com/chapter/designing-user-defined-modeling-languages-with-simthesys/91948?camid=4v1a](www.igi-global.com/chapter/designing-user-defined-modeling-languages-with-simthesys/91948?camid=4v1a)

**Steps Towards Fuzz Testing in Agile Test Automation**
[www.igi-global.com/article/steps-towards-fuzz-testing-in-agile-test-automation/144789?camid=4v1a](www.igi-global.com/article/steps-towards-fuzz-testing-in-agile-test-automation/144789?camid=4v1a)