8-16-1996

ClientServer Multimedia Database Systems Development

JaiLang Seng
National ChengChi University, jljan@cc.nccu.edu.tw

ClientServer Multimedia Database Systems Development

JiaLang Seng,
National ChengChi University
Taipei, Taiwan
jljan @ cc.nccu.edu.tw

1. Client/Server System

A client/server (C/S) system is a distributed system which consists of a collection of sites, connected together via communications network, in which some sites are client sites and others are server sites [2]. The logical division of responsibilities in the C/S system is that the client is the application or called the frontend, and the server is the DBMS or called the backend. Figure 1 shows a complex C/S environment where each site play the dual role of client and server. And, every application can access different remote databases on the network. The location independence or called transparency, however, is not supported, that is, applications need to specify the location path.

Client/Server architecture reflects the distributed organization in an enterprise. The architecture gives the advantages of efficient data processing and better data accessibility. Basic functions of a distributed system include (1) data presentation which provides user interfaces to give data requests, (2) data processing which analyzes the requests and produces machine-readable code, and (3) data access which finds the disk location of the required data, performs the operations, and returns the results, and (4) data communication which is responsible to send and receive the requests and results as messages on the network.

Client/Server systems are usually cross-platforms, that is, the client sites can have various applications, the server sites can have heterogenous database systems. The wide and various software and hardware platforms give the problems of compatibility, communication, and conversion. Current practice is to use the middleware services to connect and convert heterogenous systems. Figure 2 shows an example of the middleware connection in the client/server setting. Each client site can have its own application platform. Through the gateway manager, the client accesses the backend database servers on different platforms. The gateway or router translates the data requests and passes them to the driver which in turn calls the services of the server to process the requests. When the job is finished, the server returns the results to the client through the gateway manager.

2. Multimedia Database

A multimedia database is a database which consists of arbitrary data types and data sources. Arbitrary data types mean numbers, text, memos, graphics, images, sounds, and movies, that is, textual, audio, and video data types. Database technology faces new problems to deal with the representation and storage of large data objects [3].

Arbitrary data sources mean audio and video devices which were excluded from the conventional input database devices. Database technology needs to accept and analyze these data inputs from the multimedia devices. The protocols and standards have to be set to allow the arbitrary presentations to be processable by the database system.

Further, conventional data definition and manipulation language are inadequate to define and specify the schemas and operations of the multimedia databases. Relational database system (RDBS) research has considered to incorporate object-oriented modeling to accommodate the arbitrary data types and sources. By modeling the multimedia objects and methods in relations, we allow the object relational database system (ORDBS) to accept large data objects in a uniform representation, and to access multimedia data in use of the object query language, and to take advantage of the security and control features of RDBS.

3. Client/Server Multimedia Database System Architecture

In this paper, we propose a conceptual architecture of a distributed multimedia database system. The architecture is designed to include the features desired to build a client/server system and a multimedia database system as described above. For each client/server site, there are four layers of system components in the architecture as shown in Figure 3.

The first layer consists of the data presentation and data language service components. The second layer has the data processing component. The third layer comprises the data compression and data storage service components. The fourth layer includes the communication and transaction management components.

The first component is called the multimedia graphical user interface (MGUI) which consists of the multimedia editor and browser. The second component is the relational and object query language (ROQL) which allows users to define multimedia schemas and issue relational queries. The third component is the multimedia data processor (MDP) to perform the language compilation and query optimization.

The fourth component is the multimedia data compressor/decompressor (MDCD) which manages the compression/decompression of large data objects. The fifth component is the disk manager (DM) which handles the physical data storage and retrieval.

The fifth component is called the distributed transaction manager (DTM) which is responsible for the backup, recovery, concurrency, and security of distributed data processing. The sixth component is called the distributed gateway manager (DGM) which manages the connections and protocol conversions on heterogenous systems. These two components represent the distributed modules on the architecture.

References

[1] Bernstein, P. A., "Middleware: A Model for Distributed System Services," CACM, vol. 39, no. 2, February 1996, pp. 86-98.

[2] Date, C. J., An Introduction to Database Systems, 6th Ed., Addison-Wesley, 1995.

[3] Gemmell, D. J. et al. "Multimedia Storage Servers: A Tutorial," IEEE Computer, vol. 28, no. 5, May 1995, pp. 40-52.

[4] Khoshafian, S., A. et al. A Guide to Developing Client/Server SQL Applications,

Morgan Kaufmann, 1993.

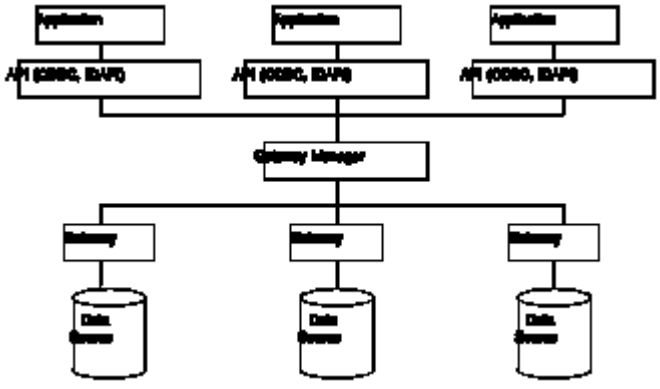

Figure 2: An Example of Client Server Middleware Connection

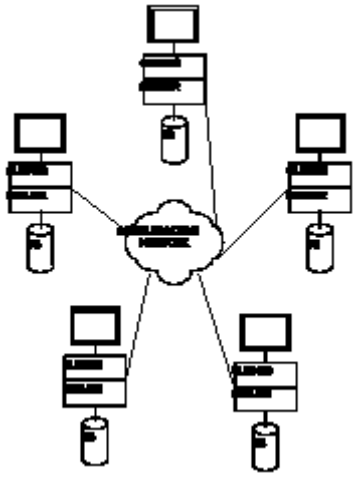

Figure 1: An Example of Client Server Environment

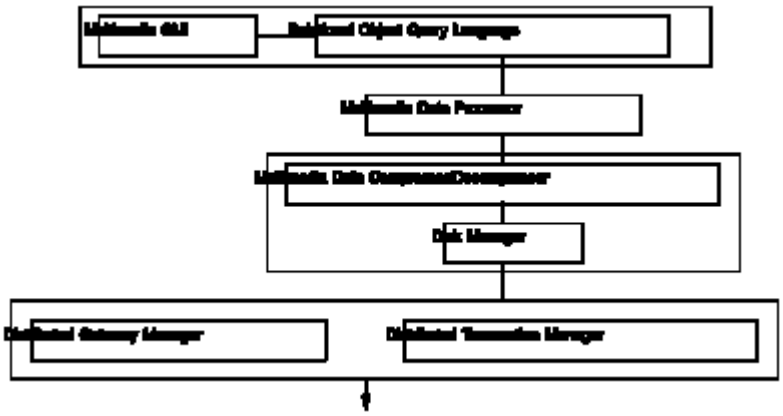

Figure 3: A Client/Server Multimedia Database System Architecture