Distributed DBMS decisions

Will you go with a client/server DBMS or a 'true' distributed DBMS? Find out what makes each distributed computing approach tick

BY GEORGE SCHUSSEL

Thanks to a distributed database management system, Citicorp's securities traders can execute trades, perform analyses and monitor groups for profitability and risk from their desktops worldwide. This New York institution has been using distributed computing during the last few years to gain a technology edge in the cutthroat field of financial services.

Citicorp is doing what many organizations today are only contemplating: distributing data to smaller, cheaper platforms in hopes of achieving distributed computing's much-touted benefits of downsized costs and improved productivity.

What about your company? If you're deciding to distribute, read on. Understanding the two distributed DBMS approaches used today and the products that put them into practice is a key part of your decision-making process.

Distributed DBMSs

Distributed DBMSs are software products that support distributed computing over a network (see functionality list page 83). At least two separate remote processors split the work for a transaction, with one of the processors supporting DBMS processing.

With the emergence of SQL as a de facto standard, DBMS vendors have begun to add distributed or client/server computing functions to their products as well as support for object approaches, database semantics and relational functionality.

The distribution of relational processing to multiple lower priced DBMS servers, furthermore, enables relational DBMSs to compete effectively in transaction processing. For users, this lowers the cost of computer cycles used.

The distributed DBMS arena primarily consists of "true" distributed DBMS and client/server DBMS approaches. As the downsizing trend progresses in the 1990s, information systems shops will turn to true distributed DBMSs and client/server options to provide high-level IS services on "downpriced" personal computer platforms.

The difference between true distributed DBMSs and client/server DBMSs is in the concept of location transparency. With location transparency, a program running at any node need not know the physical location of the computer in which the requested data resides.

True distributed DBMSs support location transparency, with each separate physical node in the network running a copy of the DBMS and associated data dictionary. It is the true distributed DBMS' responsibility to determine an access strategy to that data.

In a client/server DBMS, a limited number of designated nodes run the DBMS. Normally, there will not be a full physical copy of the DBMS at nodes that run the bulk of the application logic. Client/server DBMSs do not support location transparency, so the application must contain logic that knows where data is located.

In client/server computing, the bulk of the application logic and control of the application rest on the client; the DBMS and data operate on the server.

Although both client/server and true distributed DBMS products support network-based DBMS computing, the approach a company chooses depends on its goals. Client/server DBMSs are based on database machine concepts of the 1980s and are best for high-performance, high-transaction-rate computing. Client/server DBMSs can form the cornerstone of a cooperative processing setup and could help cut the costs of the large systems hardware/software environment for building industrial-strength applications by as much as 90%.

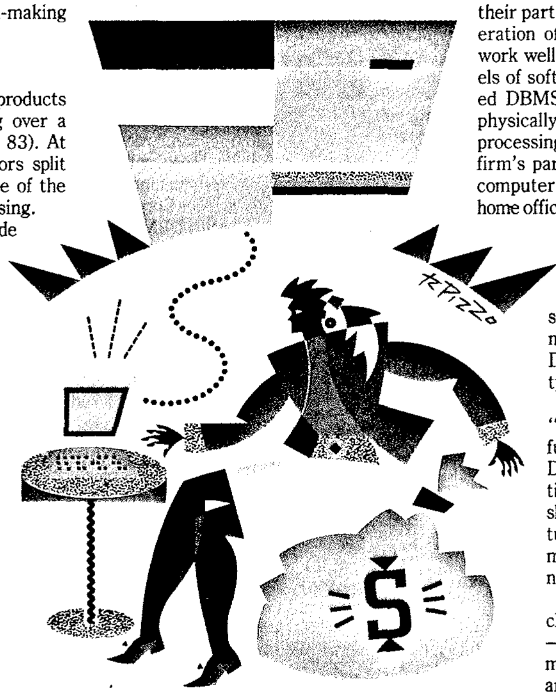
True distributed DBMS products, for their part, can be thought of as the next generation of relational SQL processing. They work well for companies needing higher levels of software functionality. True distributed DBMSs are a good fit in implementing physically separate but logically integrated processing, such as when a manufacturing firm's parts data is located on a warehouse computer while customer data is on the home office computer.

True distributed DBMSs

True distributed DBMS products occupy the Mercedes-Benz segment of the distributed DBMS marketplace, supporting a local DBMS and data dictionary capability at every network node.

Industry analysts have published "rules" or lists of requirements a fully functional true distributed DBMS should meet. These functions are listed as follows, but they should not be taken as a product feature checklist. Today's products meet some of the requirements, but no product meets them all fully.

For example, many products claim to have software optimizers — intelligent software that determines the best navigation path for an SQL query — but only a few *Continued on page 82*



Robert Pizzo

Schussel is president of Digital Consulting, Inc. in Andover, Mass., and chairman of the National Database and 4th/5th Generation Language Symposium.

• One has location transparency, one doesn't

• Product breakdown

• The two approaches may merge by 1993

Continued from page 81

have good functionality in this area.

• **Location transparency.** Programs and queries access a single logical database view; this logical view is physically distributed over a number of different sites and nodes. Queries access distributed objects for both reading and writing without knowing the location of those objects.

A change in the physical location of objects without a change in the logical view does not require a change of application programs. To meet this requirement, a full local DBMS and data dictionary must reside at each node.

• **Performance transparency.** It is essential to have a cost-based software optimizer to determine the best use of computing facilities in accessing data to satisfy a query.

In doing its job, the optimizer should understand where the data is located, how to access it efficiently, the speed and availability of computing resources and the cost and availability of communications facilities.

With a software optimizer, a query should cost the same amount to run, regardless of whether it originated from point A or point B. Software optimization technology in existing products is primitive.

• **Copy transparency.** As an option, a true distributed DBMS can support multiple physical copies of the same logical data. Advantages to this capability include superior performance gained from having local rather than remote access to data and nonstop operation in the event of one site going down. If a site does go down, the software must be smart enough to reroute a query to another source where data exists.

• **Transaction transparency.** The system supports transactions that update data at multiple sites. Those transactions behave exactly as local ones do; that is, they commit or abort. Distributed commit capabilities are made possible through the two-phase commit technical protocol.

A key concern for users is that two-phase commit uses twice as much communications capability as a more common update process running entirely within one machine. Updating data in a single logical record implemented or duplicated at several physical sites is much more complex than an update process occurring within one computer.

The question for companies, then, becomes whether faster response time, reduced processing costs and improved programming productivity compensate for the increased costs of communications and management.

• **Fragmentation transparency.** A true distributed DBMS allows a user to cut relations into pieces horizontally or vertically and place those pieces at multiple physical sites. The software is able to recombine those tables into units to answer queries as necessary.

• **Advanced applications development.** The software should support an advanced applications development environment. It should allow the easy creation of business rules that execute on the server.

These stored procedures execute from within the DBMS operation. The software should support triggers — rules executed when data equals certain values. Event alerter notice from the database to programs is available.

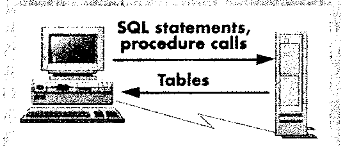
All of those functions go beyond the basic relational functions of domains, entity and referential integrity. Applications in client machines may be built with fourth-generation languages (4GL).

• **Schema change transparency.** Changes to database object design need be made only once in the distributed data dictionary. The dictionary and DBMS automatically populate other physical catalogs.

• **Local DBMS transparency.** Distributed DBMS services are provided regardless of the brand of the local DBMS. This means support for remote data access and gateways into heterogeneous DBMS products is necessary.

What does what

A breakdown of client/server functions



CLIENT

Application program
Screen forms
SQL generation
Application control

NETWORK

Hardware/wire
Communications software
Multiple clients and servers

SERVER

Optimize and execute SQL
Manage transactions
Stored procedures and triggers
Security
Concurrency management
Logging and recovery
Database creation and definition
Data dictionary

Source: Digital Consulting, Inc. CW Chart: Doreen St. John

Taking sides

Distinguishing true distributed and client/server DBMS products

CLIENT/SERVER DBMSs
Sybase Corp., Microsoft Corp.: SQL Server
Gupta Technologies, Inc.: SQLbase
Oracle Corp.: Oracle Server
Novell, Inc.: Netware SQL
XDB Systems, Inc.: XDB-Server

TRUE DISTRIBUTED DBMSs
Ask Computer Systems, Inc.'s Ingres Division: Ingres Star
Oracle: Oracle
Computer Associates International, Inc.: Datacom
Ashton-Tate Corp.: Interbase
IBM: Information Warehouse
Digital Equipment Corp.: RDB, Information Network

Source: Digital Consulting Inc. CW Chart: Doreen St. John

There are no products now on the market that support transaction processing standards. The concept of remote procedure calls will eventually enable implementation across heterogeneous DBMS products.

How the vendors stack up

Among true distributed DBMS products, the technology high ground is shared by Star from Ask Computer Systems, Inc.'s Ingres Products Division and Interbase from Ashton-Tate Corp.

Ingres' Star comes closest to fully implementing the requirements in the list above. Its software optimization capability is especially noteworthy. Even as the functionality leader, however, Ingres does not provide full distributed functionality. For example, it does not support event alerters or distributed transactions over heterogeneous DBMSs.

Interbase provides a high level of true distributed DBMS functionality and is a good seller in the engineering workstation/Unix world. The software extends the relational model with additional functions such as extendable field types and event alerters. Ashton-Tate recently acquired vendor Interbase Software Corp. and its namesake product; therefore, it is likely there will be future Dbase/Interbase product integration.

Oracle Corp., the largest independent software DBMS vendor, has pursued high performance in its distributed strategy to date rather than adding relational functionality such as referential integrity or stored procedures.

In its distributed products, Oracle functionality is limited to reading, not updating, distributed data. As a server, Oracle supports only a rudimentary software optimizer.

Computer Associates International, Inc. also has a high-performance distributed implementation built on top of its proprietary Datacom product.

This distributed implementation isn't currently based on SQL, but as SQL becomes more fully implemented in the Datacom kernel, support for distributed Datacom SQL will become available.

Limited use

Because vendors have been taking the better part of a decade to deliver all of the pieces that constitute a true distributed DBMS, companies' use of true distributed DBMSs has been hindered.

In fact, of the 630,000 database systems in use at banks, financial institutions and insurance companies, 8,350 are considered client/server systems, while only 25 are true distributed DBMSs, according to a 1988 study by Business Research Group in Newton, Mass.

By 1993, those numbers are expected to grow to 63,000 and 1,600, respectively.

If true distributed DBMS products are the Mercedes-Benzes, then client/server DBMS products are the Mazda Miatas — trim, nice looking and low-priced (savings come from computer cycles used and the ease with which they are installed and managed). By accepting some reduction in location transparency function, a user is able to use client/server technology to build a distributed computing environment that runs well with today's hardware and networks.

Client/server computing provides the industrial-strength security, integrity and database capabilities of minicomputer or mainframe architectures while allowing companies to build and run their applications on PCs, Apple Computer, Inc. Macintoshes or minicomputer networks. Unlike file-based network computing, client/server software provides secure multiuser concurrent access to shared databases. Furthermore, client/server DBMSs enable one application to connect to multiple database servers; however, not all client/server systems support this.

Three components

A client/server computing environment consists of three principal components: the client, the server and the network that connects the two (see chart at left for individual functions).

The application program runs on the client and may be written in a 3GL or 4GL. An emerging group of "Windows 4GLs" allows painting of applications under windows-based operating systems. Windows 4GLs support windows-oriented applications development and execution.

Powersoft's Powerbuilder, Ingres' Windows 4GL and Gupta Technologies, Inc.'s SQLwindows fall into this category. Using any of these application-building approaches results in a runtime configuration in which the I/O and application control come from the client, while the database and associated semantics run on the server.

The transaction capabilities of client/server software working with lower end PC servers or "superservers" — minicomputer-style cabinets built with microprocessors such as Intel Corp.'s I486 or Mips Computer Systems, Inc.'s R4000 — are astounding.

For example, on the low end of the hardware scale, Gupta's SQLbase and Microsoft Corp.'s SQL Server can both run on Intel 80386-based PCs processing approximately 10 TPC-A transactions per second. (A rate of 10 transactions per second is adequate to support 250 automated teller machines on a single server.)

PC hardware can support disks with 16-msec access times and 2M- to 3M-byte transfer rates. Such a machine can be configured with 300M bytes of memory for less than $10,000.

However, companies need not relegate client/server DBMSs to the low end of the transaction processing spectrum, especially if they combine the DBMS with high-end superservers from Solbourne Computer, Inc., Pyramid Techology Corp., Concurrent Computer Corp., Compaq Computer Corp., IBM or Digital Equipment Corp.

This high-end superserver hardware is typically going to be built with parallel Intel 80386, I486 and/or reduced instruction set computing chips from Mips Computer Systems or Sun Microsystems, Inc.

More for less

Configurations consisting of a server with a multiprocessor design and an "open" operating system such as Unix, Banyan Systems, Inc.'s Virtual Networking Software, Microsoft's OS/2 or IBM's LAN Manager give users a machine with hundreds of millions of instructions per second of processing power and 250G bytes of disk data storage at a cost of less than $500,000.

Combining this technology with small computer systems interface or other high-speed channels and a client/server DBMS offers a configuration that can replace a $14 million IBM System 3090 running DB2. That's a potential savings of up to 95%.

Companies must weigh IBM's DB2 and DEC's RDB sophisticated operating functions and utilities against the cost savings the combination of distributed DBMS, open systems and superserver technology can provide.

For high-performance computing and transaction processing in the Unix and DEC VAX markets, Sybase, Inc. is the current client/server DBMS leader. Although Sybase doesn't support location transparency, it is at the forefront of DBMS embedded stored procedures and open gateway technology — an important piece in heterogeneous DBMS computing.

Sybase and Microsoft have teamed up to provide the OS/2 market with a version of Sybase's SQL Server. The offering's functionality, however, is less than what Sybase provides on Unix and VAX platforms.

The functional gap between client/server and true distributed DBMS products is closing. Increasingly, client/server DBMSs will be able to support multiple data servers and perform functions across those servers.

In fact, by the time true distributed DBMSs mature — about 1993 — the technical differences between client/server and true distributed DBMSs will likely have disappeared. This means users can have it their way, with either technology being a good choice for downsizing and distributing applications. ●

Form and function

Distributed DBMS software, of which client/server DBMSs and true distributed DBMSs are a part, has to provide all the functionality of multiuser mainframe database software and allow the data in the database to exist on a number of different but physically connected computers.

Distributed DBMSs should have the following functions:

- Data integrity through automatically locking records and rolling back partially complete transactions.
- The ability to attack deadlocks, automatically recovering completed transactions in the event of system failure.
- The ability to optimize data access for a wide variety of application demands.
- Specialized I/O handling and space management techniques to ensure fast and stable transaction throughput.
- Full database security and administration utilities.