Client/Server Computing

In this report:

Client/Server Definitions .... -102

Client/Server Pros and Cons .... -105

The Goal: Interoperability .... -106

Conclusion .... -106

Synopsis

Editor's Note
"Client/server computing" means different things to different people within the computer industry. Computer and LAN hardware vendors, applications software vendors, and market research firms all define the term differently; but it is used freely throughout the information processing industry. Understanding what people mean when discussing "client/server computing" is very important. Information professionals are planning tomorrow's distributed computer networks based on their perceptions of this concept.

The client/server computing concept is intimately tied to PCs and LAN environments, and could profoundly affect PC networks in the near future. Although defining client/server computing is problematic, Datapro believed this report for *Datapro Reports on PC Communications* must be a top priority. For more information on this and related topics, see "Database Servers," Report 714-101.

Report Highlights
Although they use the term "client/server computing" freely, most networking, PC, database management, and LAN vendors do not have well-defined client/server strategies or are unable to discuss them. According to IBM, client/server computing involves the physical separation of applications and data between or among programmable systems. IBM then distinguishes between at least two major types of client/server applications. In Datapro's basic definition, client/server architectures divide an application into separate processes operating on separate CPUs connected over a network.

One intended client/server computing goal is linking different applications across multivendor platforms. Vendors now use at least four methods to link networked applications, some of which might be standardized in the foreseeable future. These are applications programming interfaces (APIs), database servers, remote windows, and remote procedure call (RPC) software. Without industry standards, however, little progress will be achieved in the related areas of client/server computing and multivendor interoperability. This report presents different client/server computing definitions, including Datapro's; explains its pros and cons; and lists client/server software platform vendors.

Client/Server Computing

"Client/server computing" means different things to different people within the computer industry. The International Standards Organization's (ISO's) Vocabulary Committee has not established a standard definition for the term. Computer and LAN hardware vendors, applications software vendors, and market research firms all define it differently. Yet, the term "client/server computing" is used freely and indiscriminately by vendors, users, and the trade press.

In a recent survey of *Fortune 500* MIS managers conducted by the Sierra Group, most confessed they did not understand the term at all. Of the 50-member Sierra Group/First Boston MIS Executive Council, 41 percent did not understand how client/server computing would change their present environments. Statements such as "The client/server issue is too far out . . . we will be forced to change, but we're not sure how were typical."

Nevertheless, we believe that LAN managers must at least be generally familiar with the concept of client/server computing and its different definitions. It is an important issue: future information networks are planned around the concept.

Client/Server Definitions

Although they use the term "client/server computing" freely, most networking, PC, database management, and LAN vendors do not have well-defined client/server strategies or are unable to discuss them. Among industry leaders and watchers Datapro contacted, only IBM, software vendor Oracle Corp., and market research firm Forrester Research, Inc. were prepared to discuss their positions. Users need a consistent definition of client/server computing before they can progress; however, none exists. At the very least, then, users must know how vendors define the term.

The Industry Perspective

IBM knows that client/server computing is not defined consistently. IBM's definition, according to IBM's Art Olbert, Director of Client/Server Computing, describes the physical separation of applications and data between or among programmable systems. It is not an end product, like an application or a solution, but a *technology.*

To IBM, client/server computing consists of multiple DOS-based workstation clients attached over a LAN and sharing files through a server box. In its simplest form, it is called the *redirector.* Inside DOS, Unix, or another operating system, an application requests data from a file. Instead of accessing a file located on a local hard disk or diskette drive, the request is intercepted and redirected over the LAN to a server. The server then feeds back the requested data.

In an IBM LAN, everything from an IBM Personal System/2 to a System/370 mainframe can be a server. Host computers, from AS/400s to System/370s, can perform as servers because they can offer services to LAN systems. A single host, such as a System/370 mainframe or AS/400 minicomputer, might contain both sets of services—performing as both a host and a server.

IBM draws a distinction, however, between a server and a host. According to Olbert, a *server* device must be able to:

- Deal with a set of intelligent clients.
- Provide relational database services.
- Provide print spooling services.
- Provide communications services.

By contrast, the key *host* capability is handling highly interactive, integrated applications for a large number of users against a centralized data repository. Traditional host applications include transaction processing, billing, and inventory control.

IBM cites two major uses for client/server technology. The first is in traditional departmental computing where customers link workstations together on LANs equipped with powerful servers. The second is in organizations using workstation technology to provide better end-user interfaces and more efficient data processing. The workstations work with host applications, which IBM labels *cooperative processing.*

IBM believes that office automation is better suited to departmental client/server computing than are other applications. For office automation, LAN systems are more appropriate for structured knowledge workers—those in "workgroups" performing tasks such as filling in screen fields and running predefined applications. Typically, these applications involve IBM's IMS and CICS, Tandem Nonstop applications, and the Digital transaction processing system. What IBM calls cooperative processing, on the other hand, is more appropriate for unstructured knowledge workers who deal with ad hoc computing for tasks such as decision support.

IBM's key operating system environments for client/server computing are AIX (IBM's version of Unix), Systems Application Architecture (SAA), and the relations among them—meaning VM and MVS operating systems on the System/370; OS/400 on the AS/400; OS/2 and DOS on the PS/2; and AIX on the PS/2, RT workstation, and System/370. The key role for 370 systems, however, is in managing the "enterprise," the job designated for MVS. Olbert claims that IBM will therefore get AIX servers to talk to MVS.

Software-based database management vendors view client/server differently. According to Oracle Corp., a vendor of relational database management systems, a client-server architecture allows tools and applications (clients) residing on one computer to access a database server residing on a different computer across a network. This is a much narrower definition than that used by IBM; vendor definitions of client/server computing are based on, and limited by, their particular product sets.

Forrester Research, Inc., a market research firm, provides a more useful, nonvendor definition of client/server computing. Forrester's president, George Colony, says "It is an architecture in which clients—PCs or workstations—cooperate with servers to do a job. The idea is that you're actually taking a process and splitting it into two pieces." One piece sits on a desktop, the other in a shared facility called a server.

Servers are networked computers, which describes several machine classes:

- Dedicated servers for PC networks, such as those marketed by NetFrame (Sunnyvale, CA) and Tricord (Minneapolis, MN).
- Dedicated Unix servers, such as those marketed by Auspex (Santa Clara, CA).
- Unix workstations as servers.
- PCs as servers.
- Minicomputers as servers.
- Mainframes as servers.

According to Colony, configuring mainframes and minicomputers as servers is a migratory strategy. By 1995 or 1996, a different type of machine will perform as a server. The difference between new and old machines will be the number of network I/Os per second, or NIPS, supported. NIPS describes the ability to move large blocks of information quickly through a network, as opposed to MIPS, which essentially measures the power of an individual machine.

Datapro's Definition

The preceding client/server definitions demonstrate the different views and levels of complexity perceived by the industry. Datapro's definition of client/server computing is more general.

A client/server architecture divides an application into separate processes operating on separate CPUs connected over a network. Client/server computing is inherently one-way, unlike peer-to-peer communications in which both processors can send instructions to one another. The client CPU, generally an intelligent workstation, manages or controls the user interface and communicates commands to drive the activity of a server across a network via remote procedure calls (RPCs). RPCs, the heart of client/server computing, are software capabilities used in distributed applications. Programs operating on a local system can "call" procedures or processes operating on remote systems by ordering messages, translating different data formats, and maintaining the integrity of transferred data. (RPCs are discussed further later in this report.)

In the client/server environment, the portion of the application shared among all users is installed on the server. The user interface is installed on the client machine. Some users define client-server architecture as one where the client is fully dependent on the server and cannot operate without it. Others hold that the server's purpose is to enhance the client's capabilities without any implied dependency. Attitudes about departmental autonomy seem to correlate with attitudes about distributed computing.

Client/server systems are *loosely coupled.* This means that, although processes and systems are integrated, individual CPUs run under their own operating systems. In a tightly coupled system, a number of processors are integrated under a single operating system. In fact, most distributed systems today are considered loosely coupled. A loosely coupled network containing a server and workstations shares data and other resources. Most integrators aim to provide capabilities in a loosely coupled network that exceed those of the individual workstation, allowing clients to access more memory or off-load processing.

Client/Server Application Links

Of the four leading methods of linking applications, some may become standardized in the foreseeable future. The methods are APIs, database servers, remote windows, and RPC software.

Application Program Interfaces (APIs)

A well-established way to share information between different applications is through network Application Program Interfaces (network APIs). A network transport API is a vendor-provided tool allowing application programmers to access a proprietary client/server environment, or to connect different applications running under the same operating environment. Without an API, a programmer must be an expert in the LAN environment he or she wishes to access. Working with an API does still require specialized networking experience. A disadvantage is that a universal API model or definition does not exist. Consequently, APIs are not easily portable among different operating environments. Examples of APIs and their operating environments include Named Pipes (Microsoft LAN Manager), SPX (Novell NetWare), and Sockets (Berkeley UNIX).

Database Servers

To some observers, the database-server concept has erroneously become synonymous with client/server computing. Although database management is a viable client/server application, it is not the only one. In a database server network, an application server is dedicated to providing clients with distributed access to a database management system, usually employing the structured query language (SQL) relational database model to communicate between client and server. Complex database manipulation tasks are handled by the specialized database server in a central location, off-loading client workstations and reducing network traffic. By contrast, an entire database is usually downloaded from a file server to a client in most simpler, PC file-server environments. By using the standardized SQL language, database servers can extend user applications across different network operating environments; i.e., multivendor interoperability among database programs. Database servers are limited, however, to data-intensive applications such as transaction processing where programs share only raw data.

The most common client/server application is the client in a SQL database. A database server accepts instructions from a user, changes them into SQL commands, and tells the server what records to select or update via remote procedure calls or RPCs. Many dissimilar computers can control a server by using the RPC as a common ground for distributed applications. Database server vendors include Ashton Tate/Microsoft, IBM, Ingres, Gupta Technologies, Novell, and Oracle.

Remote Windows

Remote window systems are extensions of the "window" concept popular on PCs and workstations, wherein different application environments are brought together at the user interface. The concept uses the Universal Terminal standard, ensuring multivendor connectivity. Remote windows allow a user to see two or more applications on one screen, collected from far-flung locations; however, they cannot actually be merged in a common application (such as a document composed of word processing text, spreadsheets, and business graphics). Examples include X Window System for UNIX, developed by the Massachusetts Institute of Technology and adopted by Digital Equipment Corporation and the Open Software Foundation (OSF); NeWs, developed by Sun and adopted by AT&T; and Hewlett-Packard's VPlus Windows. The X Window System is the most widely accepted.

RPCs

Remote procedure calls (RPCs) are perhaps the most promising new technology supported under client/server computing. RPCs are based on computer automated software engineering (CASE) programming tools allowing conventional procedure calls to be extended across a network. An RPC allows application programmers to develop applications for multiple (usually LAN) environments without understanding the underlying network operating systems and communications protocols, which are handled by the RPC. To do this, RPCs must function across several OSI upper levels. Programmers are insulated from the networking environment, allowing them to concentrate on developing applications.

Other advantages include a drastic reduction in the time required to develop networking applications and a high degree of portability among LAN operating system environments. A disadvantage is that, similar to APIs, no single standard exists for would-be RPC users. (The ISO originally devised a standard for RPC-like functions, but failed; it is in the initial stages of formulating a new standard.)

RPCs are relative newcomers to the communications industry. These tools are being acquired and used primarily by vendors and large users for developing applications software. The fruit of this development work—portable application programs for multivendor LAN environments—is only now reaching the market. There are two primary RPC software vendors: Netwise, Inc., Boulder, CO; and Apollo Computer Hewlett-Packard's division, Chelmsford, MA.

RPCs may become even more versatile in the months ahead, especially since an industry-wide standardization effort is under way. One advocate for an RPC standard is the Open Software Foundation (OSF), a vendor-sponsored group promoting open (nonproprietary) software platforms. The OSF is studying two proposals from different teams of vendors. One team consists of Apollo, IBM, Locus Computing Corp., Transarc Corp., Digital Equipment, and Microsoft, all supporting an RPC based partly on Apollo's Network Computing System (NCS). The other team is composed of Netwise and Sun Microsystems supporting an RPC originally developed by Netwise. NCR, Data General, Wang, and Unify support the Netwise/Sun proposal; in early 1990, it was also endorsed by Banyan, 3Com, Lotus, and 26 other integrated systems vendors.

Client/Server Pros and Cons

The Pros

Corporations constantly seek ways to save money and increase productivity without sacrificing existing computer hardware and software investments. Client/server computing, when properly implemented, promises to fulfill both goals. As desktop systems become cheaper and more powerful, users find they can off-load tasks onto smaller systems, saving mainframe access expenses (or eliminating mainframes altogether). In a distributed processing environment, the aim is to use the workstation (or client) to the limit of its power before using the server. Users obtain host system functionality from a distributed application for about half the cost. In some cases, the savings are even greater.

Additionally, client/server users must often create applications in fourth generation languages (4GLs) to access information contained in a large host system, often a mainframe DBMS. If properly planned and programmed, these new adjunct applications avoid the need to upgrade the expensive host mainframe software itself, and add computer power where it is available at reduced cost—at the workstation.

Other advantages include the following:

- Because the client machines handle the user interface, each server can handle more users.
- Shared information is locked for a shorter time.
- More users can access data concurrently.
- Sophisticated locking mechanisms are available.
- Users can share data without significantly degrading performance.
- Developers can choose a familiar client environment while accessing data on a server with a different configuration, without learning the server environment.

The Cons

While the hardware required for implementing client/server computing is less expensive than a mainframe, off-the-shelf distributed applications are not yet available. There are few standards for workgroup computing, and no clearly defined lines between technologies. Many users with genuinely distributed software must write their own programs, which is not an option for most users.

There are also political implications to decentralized computing. Turf issues, centralized control and management strategies, and corporate procurement policies may conflict with a decentralized computing philosophy. Because most of today's information systems were not designed for this level of cooperation, there are problems with performance and integration as well. To succeed, client/server computing demands a coordinated effort between several technological areas—applications software, computer operating systems, and networking software and hardware.

As defined by Datapro, client/server computing attempts to maximize processing tools in a distributed environment. Impediments to this model include existing time sharing systems and dumb terminals that operate in a centralized environment with a host CPU. Users in timesharing environments may not be able to justify a switch to client/server computing, which requires CPUs performing as clients. A desktop CPU—PC, workstation, or other intelligent device—can cost as much as $7,000 each, compared to $200 per dumb terminal.

Network performance is also taxed by client/server computing because data transmission is file oriented. Client/server programs swap huge chunks of data between the client and server. Compared to terminal/host timesharing systems, network bandwidth requirements increase greatly.

Client/server computing requires distributed network access to both files and devices. A single client-run application can perform operations on several files, which may not all reside on the same server. Applications software must maintain the files' integrity as changes pass across the network from server to server. The graphical user interface, in particular, is a critical component of client/server architecture.

The Goal: Interoperability

One goal of client/server computing is linking applications—mixing and matching different components across networked environments. Linking people with applications requires more than simply connecting hardware (connectivity). Today, users search for ways to connect applications from different vendors (multivendor interoperability).

Before interoperability can become reality, a standard interface is needed to broaden client/server computing's acceptance. Such a standard would simplify porting software packages, client portion across desktop computers' many hardware and operating systems platforms. With a standard interface, users need not adapt to a new look and feel when changing from one computer to another.

Client/server computing is an important linchpin for interoperability. Intuitively, it is a logical progression from sharing files and processing among different users to sharing information among different applications, such as spreadsheets, databases, and word processing programs. Application sharing is now possible within certain vendors' product lines, e.g., Lotus' Symphony; the goal is to link applications from different vendors. This capability is not unrealistic: new networking tools developed by LAN vendors for client/server applications are bringing users closer to true multivendor interoperability.

Conclusion

Client/server computing will require the close coordination of several technologies—from the network level to the applications level. Few vendors will admit that their current product lines limit the progress of client/server computing. Database developers point to networks bandwidth constraints, while computer hardware manufacturers believe that better applications would enhance client/server's appeal. Network software vendors focus on limitations inherent to industry standards.

What standards are in store for the future? To become widely accepted, client/server computing requires a standard network protocol stack, a standard RPC, and standards for the way applications communicate. Therefore, it will take years to realistically define the term "client/server computing."

Vendors

Listed here, for your convenience, are the addresses and telephone numbers of the vendors whose products are listed in Table 1.

Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014 (408) 996-1010

Fox Software
118 W.S. Boundary
Perrysburg, OH 43551 (419) 874-0162

Table 1. Database Servers

Vendor Database Server	Server Operating System	Client Operating System	Network Protocol*	Other Features & Extensions
Apple CL/1 Server**	VMS, MVS & VM	Mac	AppleTalk-2	depends on server database***
Fox Software Fox Server	NetWare 386	DOS	SPX-IPX	—
Gupta Technologies SQL Base	DOS & OS/2	DOS, DOS-Windows & OS/2	NETBIOS	Stored Procedures & DB2 Interface
IBM OS/2 EE Database Manager	OS/2 EE	DOS & OS/2 EE	NETBIOS & APPC	—
Ingres Ingres	DOS, OS/2 & UNIX	OS/2, UNIX, VMS & MVS	Async, NETBIOS, TCP/IP & SPX-IX	Data Encryption, Triggers & Stored Procedures
Informix Informix-Net & Online	UNIX-Netware 386	DOS & UNIX	TCP/IP & SPX-IX	—
Microrim Atlas	OS/2, UNIX, VMS, MVS & VM	DOS, DOS-Windows, OS/2, Mac, UNIX, & VMS	—	—
mdbs MDBS IV	DOS, OS/2, UNIX & VMS	DOS, OS/2, UNIX & VMS	—	Data Encryption
Microsoft-Sybase SQL Server	OS/2	DOS, DOS-Windows & OS/2	Named Pipes	Triggers & Stored Procedures
Neuron Data Nexpert Object	OS/2, UNIX, VMS & MVS	DOS, OS/2, Mac, UNIX & VMS	—	—
Novell NetWare SQL	NetWare-DOS & NetWare-OS/2	DOS, OS/2, Mac, UNIX, others	SPX-IPX	Data Encryption
Oracle Oracle Server	OS/2, UNIX, NetWare & VINES	DOS, OS/2, Mac & UNIX	NETBIOS, Named Pipes, APPC, TCP/IP & SPX-IX	—
Progress Software LAN Progress	DOS & UNIX	DOS & UNIX	NETBIOS	Data Encryption & Stored Procedures
Ratliff Software Emerald Bay***	DOS	DOS	NETBIOS & SPX-IX	Data Encryption
Via Information Systems VIA/DRE	DOS, OS/2 & UNIX	DOS, OS/2 & UNIX	NETBIOS	Triggers & Stored Procedures
XDB Systems XDB Server	DOS, OS/2 & Netware 386	DOS & OS/2	NETBIOS, APPC, Named Pipes & TCP/IP	Data Encryption, Triggers & Stored Procedures

**Protocols depend on the specific client-server operating systems and networks employed.*
***Employs CL/1 versions of Oracle, Ingress, Informix, Sybase, DB2, SQL/DS or Rdb databases.*
****Emerald Bay is a nonrelational database with optional SQL links.*

Gupta Technologies, Inc.
1020 Marsh Road, Suite 210
Menlo Park, CA 94025 (415) 321-9500

Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025 (415) 322-4100

Ingres Corp. (formerly Relational Technology)
1080 Marina Village Parkway
Alameda, CA 94501 (415) 769-1400

International Business Machines Corp. (IBM)
Old Orchard Road
Armonk, NY 10504
Contact your local IBM representative.

Micro Data Base Systems, Inc. (mdbs)
P.O. Box 248, Two Executive Drive
Lafayette, IN 47902 (317) 463-2581

Microrim, Inc.
3925 159th Avenue NE
Redmond, WA 98052 (206) 885-2000

Microsoft Corp.
16011 NE 36th Way, P.O. Box 97017
Redmond, WA 98073 (206) 882-8080

Neuron Data, Inc.
444 High Street
Palo Alto, CA 94301 (415) 321-4488

Novell, Inc.
122 E. 1700 S.
Provo, UT 84601 (801) 379-5900

Oracle Inc.
20 Davis Drive
Belmont, CA 94002 (415) 598-8000

Progress Software Corp.
5 Oak Park
Bedford, MA 01730 (617) 275-4500

Ratliff Software
2155 Verdungo Boulevard, Suite 20
Montrose, CA 91020 (818) 546-3850

Sybase, Inc.
6475 Christie Avenue
Emeryville, CA 94608 (415) 596-3500

Via Information Systems
101 Carnegie Center, Suite 209
Princeton, NJ 08540 (609) 243-0433

XDB Systems
7309 Baltimore Avenue, Suite 220
College Park, MD 20740 (301) 779-5486 ■