3-02-10 Realizing the Benefits of Client/Server Computing
Peter M. Spenser

Payoff
Whether an organization is planning for or attempting to manage a cooperative processing environment, IS managers have many questions about the client/server model. This article explains the basic characteristics and benefits of client/server computing and, most important, addresses the main questions IS managers have about how client/server computing fits with their organizations' existing systems, development process, and infrastructure.

Introduction
The two issues for corporate management to consider when evaluating client/server technology are: how does client/server computing fit within the overall information architecture, and what is required to deploy and manage it?

Client/server computing is the result of several important developments in both hardware and software over the past decade. This article describes the technical foundations of client/server computing, placing it against the background of the business's changing demand and expectations for information technology. The benefits of the technology are identified, followed by an examination of the key issues senior IS managers must resolve to realize the benefits of client/server computing in their organizations.

Basics and Definitions
Many business applications are characterized by three distinct layers of processing: presentation and display, application logic, and data management. Without distribution of processing, all the programming logic resides in one computer, which is typical of traditional mainframe and midrange systems as well as standalone microcomputers and workstations. By connecting machines together, cooperative processing is made possible. Client/server computing is a form of cooperative processing.

The Conceptual Model
The basic client/server model, shown in Exhibit 1, has the following characteristics:

- A well-defined interface between the client and server processes.
- Transparent configurations (i.e., client and server may be on the same or separate machines).
- Support for distribution of an application.
Client/Server Model

The decoupling of the processes—a simple but fundamental element of client/server computing—through a well-defined, high-level interface leads to a clean separation of processing and a more efficient and powerful implementation of function. This model will increasingly provide the basis for data base servers, communications servers, image servers, and specialized applications servers. In all cases, standardized, high-level interfaces will be key in the building of distributed systems.

Technical Developments

Client/server technology is possible because of the following advances in technology:

- Increasingly powerful processors available at ever-decreasing cost.
- The Graphical User Interface. Different graphical user interface (GUI) have been developed, each layered on top of a base operating system.
- Inexpensive disk storage. In parallel with the decrease in magnetic disk storage cost are significant improvements in disk access speeds and throughput rates.
- Local area networks (LANs), primarily Ethernet and Token Ring. LANs are becoming widespread because of standardization of differing wiring schemes (in addition to the original coaxial standard), the availability of less expensive and more powerful network interface boards, and the development of Network Operating System.
- Wide area networks (WANs) that provide high-bandwidth digital transmission at affordable costs. T1 lines, together with intelligent bridges and routers, allow the linking of multiple LANs into true, enterprisewide networks.
- Relational data bases with improved performance and capability. Structured query language (SQL) has become established as a standard language for data base access.

Client/Server Computing: An Example

Client/server computing is the combination of the client/server model with these advances in technology. Exhibit 2 is an example of the processing characteristics of a client/server architecture. In this example, a financial analyst at a PC running Windows uses a customer analysis program. This program is a client to two data base servers (customer information and product cost data bases). The analysis program also acts as a server to a client spreadsheet using the Dynamic Data Exchange (DDE) mechanism and updates a spreadsheet.

Client/Server Computing

At a UNIX workstation, an administrator sends electronic mail messages to a customer on a foreign E-mail system using the X.400 gateway. X.400 is a standardized message-handling protocol consisting of user agents (i.e., the clients) and message transfer agent
(i.e., the servers). The same workstation accesses an IBM System Network Architecture network through the communications server. The System Network Architecture communications protocol is not resident in the workstation. Only the high-level interface to the communications server is needed; it, in turn, handles all the System Network Architecture protocol logic. The customer information data base server runs on a midrange computer, such as VAX/VMS, while the product cost data base is resident on a large Reduced Instruction Set Computing-based computer running the UNIX operating system.

Many types of processing services fall within the framework of client/server computing. However, the area of most immediate interest to many IS managers is that of client/server data bases, which is often used synonymously with client/server computing. In the previous example, each data base server supports a number of clients by controlling multiuser access to the stored information and performing integrity checks and business rule logic. Integrity checks for relational tables would include logic such as “do not delete a customer entry while orders remain for that customer”; business rules could include “do not enter new customer records unless entered by authorized user.” Exhibit 3 shows the client/server data base model and how it maps to the three layers of processing.

### Client/Server Data Base

#### Business Requirements of Information Technology

In addition to technological developments, changes in the business environment favor the client/server model. Corporations in all industries want to become more responsive to the customer. This has resulted in a push toward flatter organizational structures. Such an organization demands more responsive, flexible systems with online access to corporate data and information. Corporate users want better access to and control of the presentation and manipulation of corporate data. Often, users are not the owners of the base data and thus either download data or rekey it into a personal computer (PC) application. Information technology management must contend with a more sophisticated and demanding end user who is often less willing to accept a systems environment that is perceived as unnecessarily centralized and overcontrolled.

#### Benefits of Client/Server Computing

Client/server computing meets the challenges of the business environment by providing a middle ground between the centrally controlled mainframe computing environment and the completely distributed environment in which data management is lax and resources underused (e.g., multiple standalone PCs with separate Lotus 1-2-3 models containing the same data). In the client/server architecture, the key assets—corporate data and associated business rules—can be controlled and managed centrally and users can have access to, manipulate, and present the data locally. Additional benefits are associated with client/server computing.

#### Optimization

By dividing and distributing functions, systems resources can be selected and tuned for specific processing requirements. In Exhibit 2, users work with either a PC or a workstation. In either case, the resource is optimized for a single user performing calculations with a highly graphical display. The data base server hardware is optimized for
high-performance, multiuser data access with no display or presentation requirements. In each case, the machine is configured and optimized for the functions it performs.

**Scalability**
If the size of the data base grows or the number of applications accessing it increases significantly, the data base server machine can be replaced by a larger one without any impact on other servers in the overall configuration.

**Reduced Network Traffic**
By maintaining a high-level interface between the client and server processes, the traffic on the network is often minimized. Usually, a PC performing a data base retrieval with a LAN-based common file system would result in a lot of network traffic back to the PC for the appropriate records to be selected. With a data base server using the high-level Structured Query Language interface, record selection is performed at the server itself. The result can be a significant reduction of network traffic for well-specified queries, because only the appropriate records are returned to the PC client.

**Flexibility**
One user may access a data base through a forms-oriented front-end program developed by IS; another user may access the same information concurrently using a Lotus 1-2-3 application with a SQL interface. The data base server may be set up so that the Lotus user has only read-only access to the data base.

**Leverage of IS.**
The demands on IS and the applications backlog can be reduced by providing users with workstation-based tools and a controlled environment. The IS applications development staff can benefit from working with users who have business knowledge and sophisticated tools. For example, through the use of spreadsheet-type clients for report reformatting, demands on IS have been significantly reduced.

**Cost**
Hardware costs compare favorably with older midrange and mainframe technologies. Client hardware can be scaled to individual needs, and growth of the overall configuration is incremental. Other, less obvious costs are discussed in a subsequent section of this article.

**Key Management Issues**
By comparison with traditional, homogeneous, centralized architectures, client/server computing is more complex, especially when it uses a multivendor configuration. IS management needs not only new approaches for dealing with such issues as backup and security but new techniques that have no counterpart in the traditional mainframe world (such as programming for a Graphical User Interface environment). To realize the benefits of the technology, IS managers must take into consideration their organizations':

- Architecture and existing systems.
- Development process.
- Infrastructure.
Architecture and Existing Systems
The central question here is how client/server fits within an overall application architecture that is currently mainframe based. In the client/server data base model, processing is distributed, with generic application processing (e.g., common business rule logic) and data management processing residing in one or more data base servers and with specific application processing and presentation logic residing in the client workstations or PCs. However, there are other ways to distribute the processing.

Role of the Mainframe
Some IS organizations have decided that a Graphical User Interface alone can improve user access to mature mainframe applications. This can be accomplished through front-end tools that use a high-level language Application Program Interface (HLLAPI) to apply a graphical veneer to 3270-based mainframe application screens. The front-end program runs on a PC and intercepts and processes the 3270 screens using a predefined script to partially automate the dialogue with the mainframe. The user sees a graphical interface and often does not see the underlying 3270 screens; to the host computer, nothing has changed. From an architecture standpoint, the presentation logic has been moved to the PC, but the data base management and application logic remain on the host. This approach is sometimes used as a variant of client/server computing, although the distribution of processing away from the mainframe is limited and the approach is tactical rather than strategic. As part of an orderly migration of applications from the mainframe, the graphical user interface (GUI) front end can, however, be one step in a process that may lead either to a permanent three-tiered client/server architecture, in which the mainframe remains as the enterprise data server with second-tier departmental servers (possibly UNIX based), or to a mainframeless architecture.

Downsized Environments
Downsizing and cost cutting have been cited by some people as the key benefit of client/server computing. However, the dynamics of labor and capital ratios change significantly between the different technology platforms. IS management must analyze downsizing projects to determine the increase in proportion of labor (which is ongoing and rising in cost) to technology (which is written off over time and decreasing in cost). Other important factors for IS to consider include the large one-time cost associated with converting existing data and the risk of business disruption as the new systems are brought online.

Justifying downsizing on a cost-reduction basis alone may prove difficult in some cases. The real benefits, such as increased systems flexibility and the responsiveness that helps a company’s competitiveness, may be less tangible. Among companies that have done away with their mainframes entirely, the savings reported are significant, ranging from $100,000 to $4.5 million with an average first-year payback of $1.6 million.

Data Base Servers and Relational DBMS Packages
Data base servers are a key component of client/server architectures. The standard data base server is Structured Query Language-based and uses the relational model (e.g., Oracle, Sybase SQL Server, and IBM data base 2). Several key characteristics should be carefully considered when comparing different relational data base products, namely:
· Query optimization strategy and efficiency.
· Performance (e.g., process per user).
· Support for features such as triggers and stored procedures.
· Logging and recovery features.
· Support for distribution of data and two-phase commit.
· Availability over a range of platforms.

Some vendors have distinguished their products by providing useful, unique features. For example, Sybase, Inc., and Ingres Corp. pioneered stored procedures and triggers used to enforce integrity checks and business rules. Sybase SQL Server has some built-in operating system services, such as multithreading and memory management, which offer higher performance. Oracle provides declarative referential integrity capability. Over the next few years, these features will become standard for all relational data bases.

In addition to true relational data bases, some vendors are cloaking older data base models (such as IMS/hierarchical and IDMS/network) with SQL interfaces so that they may be incorporated into client/server architectures. In the mainframe world, there is considerably more corporate information stored in nonrelational data bases and flat file than in relational data bases. EDA/SQL Server (Information Builders, Inc.) includes data drivers for various data sources together with an artificial intelligence-based engine. SQL-based requests are translated by the EDA/SQL engine (resident on the mainframe) using a data drive appropriate for the data required. Using a Windows-based spreadsheet with the Q+E Structured Query Language interface, a user can access Virtual Storage Access Method-based data files without any special programming needs.

Adapting the Development Process

The management issue here is how a company with a traditional IS department adapts to client/server computing. One of the biggest challenges for traditional IS departments in the client/server environment centers on skill sets. In the client/server environment, a blend of traditional IS skills and PC programming skills is needed.

Acquiring Necessary Skill Sets

To move staff into the new environment, IS organizations adopt two distinct approaches:

· Large-scale retraining of programmers.
· Wrapping the technology in a more familiar package.

To realize the full benefits of client/server computing, including speedier and more responsive applications development, a significant retraining effort is required. Classes are available to train Common Business Oriented Language programmers in the new tools and techniques. However, classes are ineffective unless reinforced with some real development effort.
One approach that many organizations have found successful is to select a small team consisting of staff who relate well to other members of IS and use the team to pioneer a small pilot application. This project is then followed by a larger applications development project, using the core team alongside newly trained staff. Retraining is facilitated as easier and more powerful development tools become available that do not require Common Business Oriented Language programmers to learn C or a pure object-oriented language such as Smalltalk.

Some IS organizations have decided that retraining mainframe programmers incurs unacceptable expense or long learning periods that they cannot afford. To realize some of the benefits of the new technology, they package the technology to make it look more familiar. For example, UNUM Life Insurance Company (Portland ME) is converting its mainframe environment to client/server computing. The company’s 160 developers continue to write applications in COBOL, but rather than running on the mainframe, the programs are to be encapsulated within callable, reusable C modules that run on a LAN-based OS/2 server or OS/2 client. The tools that enable this approach are contained within a set of programmer workbenches that have been developed using a combination of commercial products and custom code.

**Reevaluating the Development Methodology Currently in Place**

Client/server computing challenges most of the systems development methodologies, which range from the waterfall approach to structured techniques to the data-centric information engineering (IE) methodologies to Object-Oriented Analysis and design. There is a fundamental shift in moving from system-centric mainframe architectures, in which hardware is the critical resource, to the user-centric client/server world, in which hardware costs are minimal. The system must respond to the user, and the client side is typically event-driven and therefore not well modeled by process decomposition.

Many companies are pursuing a mix-and-match approach to applications development. Upper-CASE tools are frequently used for the analysis and design phases. Then, instead of following this phase with lower-CASE tools, a separate set of intelligent workstation tools performs subsequent prototyping and code generation. In this approach, the interfaces between the upper-CASE tools and the code-generation tools are usually manual and must be carefully coordinated. Exhibit 4 shows an approach based on a methodology outlined by Microsoft Corp. The dual-threaded architecture reflects the split between the event-driven client and the data-driven server.

**Client/Server Applications Development**
### PC-Based Tools for Developing Client Applications

Many applications developed in the client/server environment employ a PC or workstation for the client application program and require a graphical interface rather than a traditional character-based presentation. Applications development for the Graphical User Interface client using a third-Generation Language, such as C, is time-consuming and laborious; the creation of a simple window with a fixed one-line message can take many lines of code.

An improved technique is to build graphical images from distinct objects. Many PC/workstation applications development tools include some form of object-orientation, and most are structured to build event-driven user interfaces. A typical development tool has three parts:

- A painter that builds the interface by placing objects (e.g., push buttons, field input, and list boxes) on the screen.

- A script/fourth-generation language (4GL) to process events for each object (e.g., clicking a button, entering data into a field, and scrolling through a list).

- A high-level interface to the data base server.

Currently, third-generation language (3GL) may be needed in the case of truly time-critical applications, where the overhead of object-orientation and higher-level languages is not acceptable.

Tools exist in different categories for either systems developers or end users. Exhibit 5 lists some of the tool types. Many of the tools for PC-based applications can be linked using the Dynamic Data Exchange (DDE) mechanism introduced by Microsoft. Most graphical word processors and spreadsheets have DDE, which allows the integration of
these packages into PC client-based applications. Using one of these tools to build and test each major screen for an online application, each mainframe programmer may experience a learning period of two to four weeks, followed by an average productivity gain of three to five days.

### Applications Development Tools

<table>
<thead>
<tr>
<th>USER</th>
<th>TOOL TYPE</th>
<th>TYPICAL USES</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>Programmer</td>
<td>Online complex Processing (OLCP)</td>
<td>-Complex logic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Diverse transaction mix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Low volume</td>
</tr>
<tr>
<td></td>
<td>Online transaction Processing (OLTP)</td>
<td>-Simple logic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Uniform transaction mix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-High Volume</td>
</tr>
<tr>
<td>Reporting tools</td>
<td></td>
<td>-Complex requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Canned reports</td>
</tr>
<tr>
<td>GUI code generator</td>
<td></td>
<td>-Embed in third-generation language</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Portable code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Vendor independent code</td>
</tr>
<tr>
<td>Front End</td>
<td></td>
<td>-GUI veneer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-HLLAPI-based</td>
</tr>
<tr>
<td>End User</td>
<td>Established PC tools</td>
<td>-Data bases</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Spreadsheets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Graphics</td>
</tr>
<tr>
<td>Ad-hoc query and report</td>
<td></td>
<td>-Executive information systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Online access</td>
</tr>
<tr>
<td>Hypercard</td>
<td></td>
<td>-Executive information systems</td>
</tr>
</tbody>
</table>

### Infrastructure Issues

The issue here is whether the existing communications network is appropriate for client/server systems and how the network should be managed. The network (both LAN and Wide Area Network) is the backbone of the distributed processing environment. Its role is critical in the success of any large client/server architecture.

There are many devices to be managed, including routers, bridges, gateways, servers, and workstations. A network management system must provide configuration information, performance data, and tools for fault isolation, as well as help with change management, network planning, and design. Given these requirements and the multivendor configurations and multiple protocols available, the need for a sophisticated network management architecture becomes clear.

In the heterogeneous vendor environment, a simple but extensible network management architecture—simple network management protocol (SNMP)—has been adopted by many vendors for multivendor TCP/IP network management. Simple Network Management Protocol is built on a request/response model with each SNMP agent.
maintaining a data base (the Management Information Base or management information base) that describes a managed object. Network monitoring is performed through reading the values of selected management information base objects. The network management system can actively manage the network by setting specific management information base object values to trigger the agent software. Long term, network management is expected to be incorporated fully into an overall systems management architecture.

**Enterprise Systems Management**

In many organizations, all operations and systems management are well established and centralized on the mainframe. What is the client/server counterpart? This is one of the most frequently voiced concerns of IS managers when they consider a move to the client/server world. In the new environment, systems management is in an immature state. However, some partial solutions are being adopted, and attention is being focused on comprehensive, strategic solutions.

As LAN administrators have learned, software distribution, checking and enforcement of software licenses, and managing system capacities and other system functions require significant time and effort. Different vendors have different approaches to distributed systems management.

The objective of the Distributed Management Environment (DME), proposed by the Open Software Foundation, will be to create an environment that would provide an extensible framework to unify systems (both UNIX and non-UNIX), networks, and application management. The DME model is layered and has well-defined interfaces between each layer that allow different vendors to supply different modules.

Companies can adopt products, such as OpenView from Hewlett-Packard Co. and mccDirector from Digital Equipment Corp., which allow migration to DME. Increasing network bandwidth makes it possible to physically locate key production servers at a central location at any distance from their clients. For some companies, a pragmatic solution is to place servers in one building in a glass room with UPS sources, air-conditioning, raised floors, and security.

**Performance Issues**

To ensure the required performance for applications in the new environment, IS needs to understand how some of the ground rules for applications tuning and optimization have changed. In traditional, centralized mainframe environments with dumb terminals, user response was often dictated by data base/file access considerations and system loading characteristics. In the client/server world, network loading and the front-end workstations themselves may become significant factors. For example, Graphical User Interface applications place heavy demands on PC hardware, particularly for graphics processing. At minimum, a 33 MHz 386 PC is required for Windows-based (or OS/2-based) clients. Although these performance issues are not a concern for new PCs, they are often a problem when there is a large installed base of older PCs. A similar situation can occur for companies with existing LANs that are based on slower technologies (e.g., Arcnet) and older wiring schemes. To fully realize the benefits of client/server technology and to accommodate distributed systems management (and possibly other applications, such as imaging), companies must look carefully at their existing and planned network architecture and ensure that it is consistent with their systems and applications strategy.
Conclusion

Client/server computing will become the dominant architecture for all new applications in the coming years. IS management's current concerns center on how client/server technology fits with their organizations' existing systems, development process, and infrastructure. These issues must be examined and resolved before the benefits of client/server technology—power, flexibility, and savings—can be fully realized in the organization.

Many existing large business systems will continue to reside on the mainframe, though in some cases IS may choose to cloak these systems either with a Graphical User Interface veneer or with access paths added that use products with a Structured Query Language interface. Other IS managers may opt for selectively moving applications to the client/server environment while retaining the mainframe for core applications (i.e., rightsizing). These are examples of strategies that must be adopted in a new style IS model to provide not only traditional services but tools, training, and the support that enables users to develop their own customized client applications.

Author Biographies

Peter M. Spenser

Peter M. Spenser is a manager in the Deloitte & Touche Management Consulting Practice in New York, specializing in information technology. In addition to writing and speaking about client/server computing, document imaging, and open systems, he has 15 years of experience in strategic planning, emerging technology evaluation, systems design and implementation, and project and divisional management. He has worked with clients in the financial services, insurance, legal, and manufacturing industries. He has an MA in theoretical physics from Cambridge University and a PhD in astrophysics from the University of London. (A portion of this article was published previously in Data Management Review magazine.)
- Client initiates a request.
- Client accepts results from server.
- Client may access many servers.

- Server responds to a request.
- Server manages synchronization of services.
- Server may service many clients.