Payoff

Few companies can afford the continuously escalating costs and implementation delays associated with first-generation client/server strategies. Second-generation or enterprise strategies that combine centralized and distributed computing hold promise for more rapid and cost-effective delivery of organizationwide benefits.

Introduction

The client/server model, which originated as a basic, fundamentally simple approach to the organization of IS resources in a heterogeneous computing environment, has become confused by a plethora of popular interpretations. These interpretations share two characteristics:

- They increase technical complexities.
- They increase IS costs.

Increasing complexities and costs, and the concomitant reduction in productivity, are the main reasons first-generation client/server computing has not achieved expected gains. They are also the reasons why new kinds of second-generation or enterprise client/server strategies are being defined to more rapidly and cost-effectively deliver increased benefits throughout the organization. These strategies create a dynamic synthesis of centralized and distributed computing that leverages the latent competitive power of size and scale in new ways.

Overview of Enterprise Client/Server Computing

Second-generation client/server strategies are characterized by:

- **Consolidation of key IS resources.** Client/server implementations no longer need to be built around small-scale local area networks (LANs) but are integrated into enterprise networks using intelligent routers and hubs.

- **Efficient IS infrastructures.** Rationalization of hardware resources, software portfolios, and network infrastructures increases the efficiency of IS resource use.

- **Focused application deployment.** Second-generation client/server strategies target fewer but higher-priority projects with more direct and substantial impact on business competitiveness and overall profitability.

As a reaction to the disadvantages of organization size, first-generation client/server computing strategies were geared to small-scale, decentralized solutions. Today, a growing number of business leaders recognize that the aggressive decentralization popular in the 1980s is not the key to competitiveness.

Reevaluating the role of centralization inevitably means reevaluating the role of mainframe-based computing. First-generation client/server thinking erroneously assumed
that mainframes were the least important platform in planning for client/server implementation. Most organizations now recognize that moving large-scale OnLine Transaction Processing and batch applications off mainframes is, at best, a distant goal. Legacy systems run the core business processes of most large organizations and will continue to do so for the foreseeable future. Mainframe data bases maintain the vast majority of data supporting the interactive, query-intensive applications that are the backbone of first-generation client/server computing.

Although mainframes can never become the exclusive, or even predominant, platform in client/server strategies for large heterogeneous organizations, it is a comparatively fast and simple process to fuse the embedded strengths of mainframe architecture with new functional capabilities of the mainframe as an enterprise server. Achieving mainframe-client/server synergy, which is depicted in Exhibit 1, is thus key to the next generation of client/server computing.

Basic Principle of Mainframe and Client/Server Synergy

The first step to moving to second-generation, enterprise client/server strategies is practical. It is to remove the complications and verbiage surrounding the client/server model and revert to a simple, workable set of definitions and techniques. This article clarifies these definitions and then discusses the key principles of an enterprise strategy.

Client/Server Model

The client/server model defines a relationship between multiple intelligent systems in which one (i.e., a server) performs a service or services for others (i.e., clients). It is:

- **Platform-independent.** It does not specify, or require, the use of any particular platform, and any intelligent system may be used in a client or server role. There is some logic to associating the client level of the model with PCs and other forms of intelligent workstations. But servers, application logic, data, networks and other IS resources may be organized in many different ways. A server can be any type of platform located anywhere within an organization.

- **Technology-independent.** The client/server model defines a relationship between systems but does not specify their underlying hardware and software technologies. Relational Data Base Management System(DBMSs), for example, play an important role in many client/server applications. But an application that lets end users query hierarchical data bases or sequential file systems from a workstation conforms just as well to the client/server model.

Similarly, Graphical User Interface(GUIs) are useful and are not restricted to PCs. A Graphical User Interface provides the same value if accessed on a PC or on a dumb or intelligent terminal. In addition, many organizations have implemented client/server computing using simple character-based interfaces.

It is technically inaccurate to include LANs in the definition of client/server computing. Network protocols, of whatever type, form part of the infrastructure that links clients and servers. The client/server model does not specify any particular technical content for this infrastructure. The primary logical relationship remains between the client and the server.
Previous screen
Many organizations are integrating LANs into enterprisewide networks and consolidating servers.

Architectures

Client/server computing is often confused with distributed computing. The client/server model deals with the logical organization of IS resources and requires only that client workstations be physically distributed. Distributed computing is an overlay to the client/server model that proposes the physical distribution of both servers and workstations.

In practice, the client/server model is consistent with at least three different types of architectures:

- **Two-tier architectures.** This most common form of client/server computing involves simple binary relationships between client workstations and servers. Most data base query applications, electronic mail systems, and end-user computing environments are built around links between client workstations and single servers. This type of architecture has the lowest technical complexity and cost.

- **Three-tier architectures.** Divide all logical and physical resources between centralized servers, distributed servers, and workstations. They have the greatest technical complexity and highest cost.

- **Hybrid or two-and-a-half-tier architectures.** Use routers, hubs, and other types of intelligent network controllers to substitute for an intermediate layer of server architecture. Logically, hybrids are two-tier architectures, because application and data resources are divided predominantly between centralized servers and workstations. Physically, the architectures are three-tier. A local server that does not constitute a separate logical level is used to provide peripheral support and store-and-forward functions. Communications with central servers typically use medium- to high-speed network bandwidths to prevent response-time degradation. The complexity and cost of hybrid architectures fall between that of the two- and three-tier models.

An organization's choice of architecture depends on several factors, including business model, applications portfolio, data structures, and geographic dispersion. The overall trend, however, is toward hybrid architectures.

Generally, network bandwidth is a relatively inexpensive commodity that can be used as an alternative to the physical decentralization of servers, data bases, and support staff. Latest-generation LAN technologies allow speeds of more than 100M bps, and frame-relay wide area networks (WANs) support line rates of up to 2M bps. New backbone capabilities are provided by T1/T3 links. asynchronous transfer mode (ATM) at speeds ranging from 25M bps to more than 2G bps, combined with fiber optic technology, offers major new options for both local and wide area interconnection.

Modes of Computing

One early misinterpretation of client/server computing focused on the 1980s paradigm of end-user applications involving spreadsheets and word-processing applications combined with data base query tools. In practice, this broadly applied misconception has generated unnecessary organizational and technical complexities. In addition, it overlooks the sheer
diversity of the tasks performed on workstations and processed by servers in the typical business organization.

A more useful approach is to distinguish between five modes of computing:

- End-user computing.
- Query-intensive computing.
- Online Transaction Processing (OLTP).
- Mixed-mode computing.
- Batch processing.

**End-User Computing.**
End-user computing involves a basic form of client/server relationship. PCs or workstations are used for local computation, text-processing, graphics generation, time management, project control, peer-to-peer communications, and various other personal productivity applications. Applications are predominantly single-user, and servers simply provide access to common applications and shared peripherals. Electronic mail also falls into this category. As a natural extension of PC-based text processing, it requires little interaction with other systems in the organization.

**Query-Intensive Computing.**
The query-intensive computing mode involves periodic queries initiated from workstations to server data bases. Data is extracted, downloaded, and processed on workstations. Query-intensive applications include general-purpose decision support and specialized tools used by various categories of analysts and professionals. Graphical User Interface are common, and relational data bases, which are easier to query than hierarchical data structures, are the norm at the server level.

Simple, ad hoc queries are relatively easy to handle. Larger, more complex queries that require sequential access to multiple files and data bases place significantly greater demands on both the DBMS and the server that supports it. More powerful workstations, more functional workstation-based tools, and higher network bandwidth are also typically required.

**Online Transaction Processing (OLTP).**
OLTP involves continuous but relatively short interactions between workstations and hosts. Relatively simple data structures are used to make additions, deletions, and changes to host data bases. Processing is initiated by, but not necessarily performed on, workstations. OLTP applications include point-of-sale and front-office systems.

The characteristics of OLTP thus differ significantly from those of end-user and query-intensive computing. In most organizations, features commonly associated with first-generation client/server computing have been slow to penetrate OLTP environments. In many cases, they are functionally unnecessary. For example, relational data structure can actually reduce performance in high-volume OLTP applications, and attempting to channel both transactions and queries into the same data base can cause problems with data integrity. Many organizations thus continue to use hierarchical DBMSs and sequential file. Access to production data is provided by a relational DBMS acting as a query server, as shown in Exhibit 2.
Adding Client/Server Capability to Legacy OLTP Applications

Similarly, although graphical user interfaces can increase productivity, the extent to which they do so depends greatly on the nature of tasks performed at the workstation interface. Latest-generation graphical user interfaces can actually reduce user productivity by complicating otherwise simple work processes.

Many OLTP applications, particularly business-critical applications, require close synchronization of application and data processes and high levels of data and transaction integrity. Unnecessary technical complexity should be avoided.

Mixed-Mode Computing.

Mixed-mode computing involves combinations of transactions and queries. This relatively new type of application is generally built around a relational DBMS, an integrated application resident on a server, and workstation-based interfaces and tools. Typically, mixed-mode systems involve placing queries from a workstation (e.g., to find out the status of a customer order or to obtain product or service information) to a server database, then processing orders, or initiating other actions. Applications may include end-user computing features, such as spreadsheets to calculate purchase options or electronic mail to follow up on customer requests. This type of application is used in such areas as securities trading, telemarketing, and customer service systems.

Batch Computing.

Batch computing involves the execution of processing tasks uniquely on a host system. Workstations are used to input data, manage operations, or review computation results. Otherwise, however, no end-user interaction occurs or is required. Batch computing has traditionally been associated with accounting reconciliation and other periodic data consolidations.

In many organizations, however, batch workloads are increasing as a result of more frequent consolidations (e.g., production data is required for analysis daily rather than weekly or monthly), direct mail marketing (e.g., high-volume addressing and print output), image-processing applications (which involve a large batch content), as well as centralized backup of distributed data and host compilation of workstation-developed applications.

New informational applications handling large data volumes also commonly run in batch mode. For example, securities firms run large, computer-intensive portfolio analysis workloads on hosts. Other new applications have surfaced in such areas as econometric modeling, market research, and financial analysis. In addition, emerging large-scale commercial data mining applications are batch-oriented.

Application Styles

Because of the wide variety of different applications within a typical organization, a highly granular approach toward applications is required. Client/server applications should be designed and implemented in a way that recognizes variations in size, complexity, and functional characteristics stemming from the different business processes and tasks supported. From that perspective, most client/server application requirements can be met with several styles, which are described in the following sections.

Large-Scale Application with GUI.
This style is appropriate for systems that support large or organizationwide business processes and operations in all modes. The applications generally support hundreds or thousands of users, handle large workloads, and employ large data bases. For large-scale legacy applications, if the business logic is correct and the application remains sufficiently functional, a simple Graphical User Interface front end may be added. New large-scale applications are typically developed using up-to-date Fourth-Generation Language, Computer-Aided Software Engineering, or Rapid Application Development tools; incorporating graphical user interfaces; and using relational or hierarchical DBMSs (i.e., in the case of OLTP systems).

**Single-User Application with Host Query/Extract.**
This style is appropriate for most PC-based host query and decision support applications. It is simple to implement through software packages, ad hoc structured query language (SQL) tools, or user-oriented 4GL tools with built-in query functions.

Applications typically use a graphical user interfaces and run on a PC or workstation. Data bases are located on a host server or servers. Data is downloaded to a PC-based DBMS and worked on locally.

**Departmental Application with Host Data Base Replication.**
This style is appropriate for systems that support relatively small, discrete business operations that draw data from and contribute data to organizational data bases. They include Decision Support System built around small-scale, multiuser relational DBMSs and most types of departmental or workgroup applications.

Interactions between host and local data base servers are occasional and realized through relatively simple replication techniques. Implementations are thus usually less complex than those using two-phase commit. Applications run on the departmental data base server and on the workstations attached to it.

**Distributed DBMS with Two-Phase Commit.**
This is the most complex client/server style. System, application, data, and management processes on clients and servers need to be closely synchronized across three-tier logical and physical architectures with high levels of data and transaction integrity. This style is expected to be used in distributed OLTP applications.

**Principles of An Enterprise Client/Server Strategy**

**Logical Separation of Resources**

One of the main attractions of the client/server model is that it allows different IS resources to be located on different platforms. It is thus possible to logically distinguish between four key categories of resources:

- Network infrastructures.
- Data.
- Applications.
- Resource management tools.
Addressing these four categories separately provides the key to determining which resources should be consolidated and which should be decentralized.

**Network Infrastructures.**

The key resource variable is the network infrastructure, the primary basis for planning. Bandwidth, reliability, and intelligence (i.e., in the form of routers, hubs, and equivalents) determine the degree to which consolidation is possible. Decisions as to whether consolidation is desirable in any particular instance are then based on other criteria.

In this context, the network infrastructure includes communications facilities; all types of local and wide area network protocols; middleware, such as data base access tools; and shared organizationwide services such as electronic mail networks (which can be linked to existing departmental and divisional systems by gateways) and electronic data interchange (EDI) systems.

**Data Strategy.**

The cost/benefit equations for distributed processing and distributed data differ significantly. Distributed processing is comparatively easy to implement. Distributed data is not.

A second-generation strategy manages and protects distributed data by creating a corporate data repository, often referred to as a data warehouse. Data from all major production systems is streamed into the repository, and secure master copies of distributed data are created by replicating the contents of distributed DBMSs into the central repository at regular intervals. Similarly, LAN servers are backed up automatically and remotely.

The benefits of consolidation are thus obtained while retaining end-user control over, and access to, local data resources. Exhibit 3 depicts a representative corporate data repository in which all key corporate data are more easily managed and protected at the central location. Corporate IS staff manage business data policies; define and institute common data access standards; manage the logical organization of shared data resources housed in the repository; ensure that secure, easily recoverable master copies of all data are maintained for disaster recovery purposes; handle routine data base administration tasks; and implement access control, security administration, and audit procedures.

**Corporate Data Repository**

**Applications Strategy.**

In this area, a second-generation client/server strategy starts by segmenting the organization's applications portfolio. Applications specific to the business processes and operations of individual divisions, business units, and departments are candidates for decentralization.

End-user computing applications, such as query tools, spreadsheets, word processing, various office software packages, and small electronic mail systems, are usually devolved to individual or departmental users. However, standardization should be considered to reduce implementation, training, support, and management overhead, and to facilitate procurement economies.
The third category of applications, large-scale systems, are defined as applications that are inherently large, because the business processes and operations they support are inherently large or organizationwide. Large-scale systems typically include:

- Business control systems.
- Core business systems.
- Large-scale informational systems.

**Business Control Systems.**

Business control systems handle such functions as accounting, finance, payroll, asset management, and personnel administration. By definition, they are organizationwide, with all divisions, business units, and departments using the same standardized services. Because such applications are generally considered to have a relatively minor impact on competitive performance, most organizations give them low priority. Decentralized applications, however, offer an important consolidation and reengineering opportunity for reducing IS costs (typically in the 50% to 75% range), improving applications quality, and increasing access to key information for business analysis, resource planning, and organizationwide cost control.

**Core Business Systems.**

Core business systems support large, core business operations usually built around transaction-intensive processes. Examples include systems for bank branch automation, Electronic Funds Transfer, retail point-of-sale, reservations, manufacturing management, inventory control, claims processing, customer information, as well as various front-office and logistics applications.

These systems are usually business-critical because companies could not function without them. Where such systems are inherently hierarchical in nature (i.e., transactions and data flow to a central point), both data and application logic usually remain centralized. If their business logic and application functionality remain adequate, core business systems are usually maintained. Failure to do so easily creates a ripple effect throughout the organization of reduced competitiveness, inefficiencies, and missed business opportunities.

Significant consolidation, reengineering, or new large-applications development may be required when multiple, functionally similar systems result from such factors as acquisitions, mergers, and previous decentralization policies. For example, business reengineering exercises may involve replacing previously separate departmental applications and data bases with single, unified systems.

**Large-Scale Informational Systems.**

These systems are a relatively new and increasingly important type of application. Complex calculations involving the analysis of large data volumes are processed using specialized systems such as supercomputers or parallel query servers rather than departmental servers or workstations.

Although this type of application differs qualitatively from conventional OLTP and batch systems, it is inherently large-scale in nature, and its implementation is usually centralized.
Resource Management

Failure to address IS resource management is one of the major weaknesses of first-generation client/server strategies. Because organizations often select platforms lacking both reliable internal management functionality and add-on tools, they experience cost overruns, application delays, and major problems involving data availability, integrity, and security in distributed environments.

Resource management functions can be categorized into the following broad groups, each of which comprises myriad tasks: system management/administration, performance management, storage management, configuration/event management, data management, availability, network management, operations, and administration.

A second-generation client/server strategy aims to implement infrastructures that automate routine management tasks that were previously performed manually, often in a highly fragmented fashion, for distributed servers and LANs. This approach typically significantly reduces staffing levels and increases utilization levels of all server-based resources, thereby lowering hardware and software costs.

The remaining functions are then concentrated in control centers, where professional staff remotely monitor, manage, and maintain large bases of networked clients and servers. Management tools dealing with underlying resources with which users do not interact directly are thus centralized without any obvious impairment of user control.

Rethinking the Mainframe as Enterprise Server

In a large organization, the roles of the enterprise server tend logically to consolidation and can easily be defined for four functions:

- Enterprise data base server.
- Enterprise query server.
- Enterprise application server.
- Enterprise management server.

These functions are largely consistent, regardless of the platform or platforms used. Whether a mainframe can perform these functions depends on the technologies being used. Hierarchical data bases, sequential file, obsolete hardware, and manually maintained legacy applications do not make for a good enterprise server. But latest-generation mainframe capabilities change this picture in all four key functional areas.

Enterprise Data Base Server.

The enterprise data base server requires relational DBMSs and supporting hardware along with software facilities that can efficiently handle large volumes of data. Platforms such as data base 2 meet this requirement. They are typically larger and more robust than the popular distributed relational DBMSs but use the same basic architecture as these systems and have major compatibilities with them. Multilevel approaches are thus employed that use mainframe DBMSs to act as repositories for organizational data and distributed DBMSs for local data exploitation.
There have been a few popular attempts to exclude mainframe relational DBMSs from the definition of client/server computing. There is no architectural, technological, or functional basis for such views and most may charitably be described as biased.

**Enterprise Query Server.**

The enterprise query server functions are supplied by direct SQL access to mainframe data for light-duty applications and by dedicated parallel query servers for applications requiring higher performance levels. Dedicated servers, using MVS- or UNIX-based parallel architectures, are among the most powerful query processors in existence. Such servers are widely used in large-scale data mining, one of the most valuable types of application that can be implemented in a client/server environment.

**Enterprise Application Server.**

The enterprise application server hosts both conventional OLTP and batch systems, as well as newer types of large-scale mixed-mode and informational applications. Latest-generation mainframe 4GL, CASE, RAD, and visual programming tools provide client/server functionality, including object-oriented and Graphical User Interface capabilities, along with support for PCs. Equivalent or even superior functionality, such as flexibility and user-friendliness, popularly associated with smaller platforms can also be realized by using up-to-date tools and techniques for mainframe-based applications.

Although software packages may be appropriate for small-scale or relatively standardized applications, their value for large-scale systems is less clear. CASE and RAD tools allow applications to be closely customized to the business models, competitive strategies, and operational requirements of individual organizations.

If software packages incorporate implicit business models, subsequent modifications to accommodate changes in business structure and practices may be difficult. Most packages require some level of customization for large organization requirements. If customization is substantial, hybrids of vendor-supplied and internally developed code may prove to be inelegant and difficult to maintain. This is particularly likely if packages come bundled with proprietary development tools and data structures. Moreover, because software packages are inevitably available to all users, by definition they provide little or no direct competitive advantage to individual organizations.

**Enterprise Management Server.**

Enterprise management server functionality is provided by a wide range of mainframe-based management tools for distributed servers and LANs, remote backup, and related services. The mainframe role is significantly enhanced by two factors:

- As client/server computing moves to the enterprise level, management of large numbers of distributed clients and servers becomes a business-critical application. Thus, the management server must have high levels of availability, data integrity, and managed resource capability if adequate service levels are to be maintained.

- In most large organizations, mainframes are already the primary platform for management of WAN backbones. As LANs are progressively interconnected to backbones using intelligent routers, hubs, and gateways, the mainframe role is naturally extended.
Client/Server Priorities

Enterprise server functionality provides major new value. But it will still be necessary to maintain the conventional core business and business control systems that, in most organizations, continue to run predominantly on mainframes. These systems are the source of the fundamental synergy for client/server initiatives. Mainframes manage and maintain the vast majority of data that is exploited by new interactive, query-intensive client/server applications.

An organization attempting to phase out mainframes would thus find it necessary not merely to rehost applications but to rebuild an entire corporate data base infrastructure around dozens, or even hundreds, of logically separate, physically dispersed DBMSs. Providing cross-access to data for users in multiple business units at large numbers of sites is a major task. Implementing directories, data integrity, and security mechanisms, along with replication and recovery facilities, is also extremely complex and expensive in this type of environment.

Similarly, an enterprise server must possess many of the key historical strengths of the mainframe environment. These strengths include handling large data bases and workloads; maintaining high levels of availability, data integrity, and security; managing system, data, storage, and network resources effectively; and doing all of this simultaneously, 24 hours a day, 365 days a year.

That the mainframe remains in one form or another is a central fact that must be taken into account in the IS strategy of any large organization. This fact raises several new questions about client/server priorities. For example, has the avoidance of an increased mainframe role ceased to represent common sense and good business practice and become an end in itself? Is a complex three-tier approach functionally necessary, or is it being adopted to compensate for the weaknesses of alternative platforms? How much of the client/server strategy is still driven by an implicit assumption that the mainframe will eventually be replaced? And how high a price is the business prepared to pay for a mere question of nomenclature?

It is becoming clear that it is a great deal easier, faster, and less expensive to add enterprise server functionality to a mainframe than it is to add mainframe capabilities to alternative platforms. The issue is not difficult to resolve and as such should occupy no more money, personnel, and management attention than necessary. The IS strategy of any organization has more pressing and ultimately more rewarding challenges to address.

Conclusion

IS trends are moving in several directions at the same time. More computing power is moving to local servers and desktops with increased functionality to the user interface. Equally, both legacy and new types of large-scale applications, data bases, and resource management functions are moving toward consolidation. More powerful and more intelligent enterprise network s mean that the physical location of hardware no longer dictates IS strategy.

By allowing different categories of IS resources to be logically separated, the real potential of client/server computing is in enabling users to take advantage of all of these trends.

In the final analysis, client/server computing is about organizing IS resources in a heterogeneous environment. Its essential characteristic is diversity of design and implementation. The client/server model allows for literally thousands of different
permutations of platforms, software technologies, modes of computing, application styles, and IS structures.

New opportunities have been created for customizing IS strategies to the unique business strengths and competitive goals of individual organizations. An enterprisewide, second-generation client/server strategy thus allows choices to be made in new and innovative ways.

**Author Biographies**

**Brian Jeffery**

Brian Jeffery is managing director of the International Technology Group of Mountain View CA, an independent consulting firm specializing in analyses of IS technologies and markets.
