4-04-35 Client/Server Challenges and Benefits

Brenda Castiel

Payoff

Debunking the persisting myths surrounding client/server technology, including the idea that client/server development is cheaper and easier than mainframe development, is the first step in a realistic assessment of the technology's true potential. Reviewing case studies of early experiences with client/server applications development in a variety of industries helps IS managers better understand the challenges and benefits of this computing environment.

Problems Addressed

Despite growing organizational experience with client/server technology, numerous fallacies and myths surrounding the technology persist. Debunking these myths and reviewing a broad spectrum of case studies of early experiences with client/server systems form the basis for an accurate assessment of the technology's potential. The resulting distillation of lessons learned will help IS managers more effectively plan for the challenges encountered in client/server projects and increase the chances of achieving realistic benefits.

Myths of Client/Server Computing

These are some of the commonly held myths surrounding client/server technology:

- Client/server development is cheaper and easier than mainframe development.
- Everyone is doing it.
- Everyone is succeeding.
- The mainframe is dead.

Several studies show that when all relevant costs are included, client/server computing is actually far more costly than continuing to build on the mainframe. Although numerous companies are exploring client/server technology, not many are completely committed to it. Many organizations concerned about the perceived lack of systems management tools and relatively lax security facilities are reluctant to make major commitments to what they consider immature technology.

In the area of success, it is important to consider that it is much more pleasant to publicize success than failure, and many failed projects go unreported. Finally, there is strong evidence that the mainframe will continue to play a role in the future as a repository for enterprise information resources. In fact, mainframe sales rose 25% in 1995 and showed continued growth in 1996.

The tendency to view client/server technology as a magic bullet is based on a belief that applications development will become quick and easy, mainframe-related costs will disappear, and mainframe developers will learn client/server concepts easily. People who
espouse this belief think that the new tools they buy will integrate seamlessly and work perfectly together, even with beta code.

Certainly, client/server is not the panacea for the entire range of applications development problems. Some easy-to-use tools are available, but these may address only the code generation portion of the application effort. Off-loading work to another platform may delay a mainframe upgrade, but only in rare cases is the mainframe actually displaced.

Learning new tools and design techniques is not easy, and not everyone can make the transition to the client/server environment. One study indicates that only 40% of mainframe development staff will effectively do so.

Client/server technology presents the additional challenge of integrating and testing new software and hardware components. New software may not have reached the stability that IS professionals have come to expect from well-established mainframe products. Because not all vendors are equally committed to fixing problems, schedules are affected by unanticipated software problems.

**The Reality of Client/Server Computing**

Client/server computing is not magically easy. It is more expensive than mainframe development, largely because of infrastructure setup and learning curve issues. Ongoing support costs also tend to be higher. A study by IBM's consulting division found that as much as 75% of customers' client/server costs, including such items as salaries for network experts, were not attributed to client/server projects. These are the so-called hidden costs of client/server computing, and they often go unreported.

In addition, although saving money is cited as the second most popular reason why organizations are moving work off the mainframe (applications reengineering was the first reason among larger organizations), savings are not automatically realized. Not every organization that strives to reduce its dependency on mainframes actually succeeds. In fact, a study conducted by the Gartner Group shows that only 3% to 8% of mainframe work is off-loaded. This is hardly a significant amount. The mainframe most frequently remains in place, not only to run legacy systems, but also to act as data base server.

It is interesting, however, that even though the Gartner Group study showed that projects to move work off the mainframe overran their budgets and schedules (reported by 73% and 82% of respondents, respectively), the projects were judged beneficial by the organizations involved. Despite their high costs, these projects must therefore have achieved some of their business objectives, such as improving business processes, reducing cycle time, and increasing competitiveness.

Exhibit 1 illustrates how the relationship between labor and capital varies by computing platform and the increased cost of client/server computing when all costs are considered. Soft costs, such as support, training, and standards setting, are higher in a LAN environment than in a mainframe environment.

**Client/Server Costs**

Despite this reality, client/server technology is still the right choice for many organizations. It is the right choice not because it is cheaper, but because it allows businesses to provide new products, serve their customers better, or transform the way they do business.
The Mainframe Outlook

A survey reported by InformationWeek back in June 1994 indicates that the mainframe will be around for quite a while. A full 90% of technology managers believe that the mainframe has long-term practicality, because the speed, power, reliability, and overall efficiency of this hardware environment have yet to be matched. The difficulty of converting legacy systems and the higher costs per user associated with other platforms are also important factors in the mainframe's longevity.

The endurance of the mainframe is supported by recent trends in mainframe usage. Although there has been a slowdown in new mainframe installations over the past seven years (measured in units shipped), the decrease is not nearly as steep as might be expected. In fact, total installed capacity rose by 16% in 1994, and IBM's mainframe business increased 25% in 1995.

Newer parallel processing technologies are making mainframes more cost-effective as they provide platforms for surviving legacy systems and new data server roles in client/server environments. In many of the case studies this article presents, the central data base on a mainframe remains intact, but the data is accessed and used in innovative new ways, providing increased value. The data warehouse has, in may cases, given new life to the traditional data repository role.

It is also important to mention the impact of the even newer concept of network computing. In this concept, a no-frills desktop PC obtains both its software and data from the network (i.e., an internal corporate network or the Internet). The thin client concept, in which little data or function resides on the desktop, yields ease of management and support, not to mention reduced hardware and software costs. It marks a return to the centralized computing model with much greater control in the hands of IS management. Although there is some resistance to this approach in the form of criticism that so-called anorexic desktop devises are nothing but a throwback to the days of dumb terminals, larger organizations have begun evaluating this computing model.

Making the Transition to Enterprise Systems

Making the transition to client/server computing involves a completely new environment, analogous somewhat to the transition from batch to on-line computing that occurred 20 years ago. As such, there is a huge learning curve for most developers, and it is measured in months, not weeks.

Until recently, many client/server applications were departmental or tactical in nature or added a GUI front end to an existing application. These systems were relatively easy to implement and often provided quick benefits.

The goal, however, is to avoid piecemeal solutions with the redundancy and lack of integration that often characterized older systems built in the 1970s (i.e., legacy systems). The next wave of client/server computing must address enterprise systems, involving greater integration, interfaces to legacy systems, more middleware, and increased data sharing. High transaction rates, complex processing, and three-tier (or more) architectures present the challenges for IS. Although these systems are much more complex, they are worth pursuing, not because they present a cost-cutting solution, but because of their business benefits.
Case Studies

The following case studies present lessons learned from client/server projects in the finance, leasing, health care, distribution, retail, banking, utilities, and manufacturing industries.

Finance

One of the top five firms on Wall Street, which handles mutual funds and various other investments for both private and institutional clients, wanted to provide a single point of service for its clients. To do this, the firm needed to integrate several mainframe applications.

The resulting system, which serves a total of 600 users, lets the firm provide more effective service but requires fewer skills among sales support staff. Routine requests are now easier to handle, and research, service, and sales activities can be combined in a single enriched job function. The new system provides remote access to records held by the transfer agent and displays a seamless view of the customer's various accounts. System scripts allow use of less specialized staff, by leveraging intelligence built into the application.

The system is based on Gupta SQL Windows, RS/6000 hardware with UNIX, and Sybase. The back-end mainframe systems use VSAM, IDMS, and DB2; the legacy systems are viewed as the data repository. To position itself for future implementation of imaging technology, the firm installed 486 PCs with 17-inch monitors as workstations. A fairly rigorous software selection process was used; each product was evaluated for one week. Prototypes were built to determine the strengths and weaknesses of each tool.

The system is used, for example, to respond to telephone-based customer requests. The user enters the customer name or account; data is pulled from the mainframe to the server and stays on the server just for the length of the phone call. Data is uploaded only when updating is necessary. Approximately 90% of calls are inquiry-only; exceptions include notes about the call or customer instructions. Some local data resides on the PC or server (e.g., mutual fund information, help information, and commonly referenced data), but client data resides on the mainframe.

The system is now fully rolled out. Although there are few objective measurements of its benefits, the firm believes that sales people are able to make more calls, routine requests are handled more efficiently, and communication with customers has improved. The system also helps the firm maintain its reputation for technology leadership.

What has this organization learned from the experience? Following is a list of their lessons:

- Pay attention to infrastructure setup early on.
- Think of client/server not as an application, but as an environment.
- Take time to develop standards.
- Be prepared for testing to take much more time than it did with traditional development.
- Build an adequate network installation plan, if upgrading workstations.
Obtain solid support from business process owners.

Create integrated development/business teams: this group had 15 business users full-time.

Plan for early education for users, including team-building courses. The users of this system view it positively, because it enables them to be more productive while improving the level of customer service.

Equipment Leasing

The next example involves a computer equipment leasing company. This organization wanted its sales people to be able to finalize deals on site at the client's office but also to ensure the deals' profitability.

This project began in 1989, using a cooperative, peer-to-peer design, Btrieve on PCs, and InFront, a screen-scaper tool that downloads mainframe data to the PC. An intelligent rules-based program ensures that the negotiated contracts are financially sound. In addition, the system creates legal documents ready for signature. Data is subsequently sent to a mainframe system for reconciliation and further back-end processing, such as calculating sales commissions.

With the limited technology then available, the system was deemed a success. It has been in production for many years, serving 100 to 200 users.

Having succeeded with its first effort, the company next wanted to downsize all its mainframe applications. This project was initiated in 1991, using the now-popular combination of PowerBuilder and Sybase. Programmers received training in each of these tools, but not in client/server architecture and GUI design. The project had no defined architecture for distribution of data or functions. It had a rather large scope and lacked a clearly identified business objective. For these and other reasons, this project did not go forward.

The technology group was not discouraged by this setback. It tried a smaller project using the same technology. Once again, the goal was to allow the sales force to take a proactive role with customers, analyzing lease renewal options (i.e., renew lease, cancel, or purchase the equipment) in the customer's office and finalizing deals on site.

The system captures details of the lease using the InFront tool and sends the data to the mainframe. The financial information is used to derive the quotes. Information from the mainframe is then converted to a Pyramid server, where the Sybase data base resides. The client PCs send data to the server, adding information obtained from the customer's location.

The company found PowerBuilder easy to use but slow for complex transactions. Although the company was not satisfied with the system's performance, the project went into production. SQL access to Sybase proved to be too slow for its needs.

Among the lessons learned, the company stresses: acquire the right skills, set a reasonable scope, evaluate tools rigorously, apply strong project management skills, and pay attention to performance early in the project design cycle. If GUI design is not well understood, the result is likely to be a menu-based interface that resembles a traditional 3270 screen. Finally, the company learned that some applications are better left on the mainframe.
Health Care

This next group, the Joint Commission on Accreditation of Health Care Organizations, is responsible for certifying health care groups, such as hospitals, across the US. It handles a great deal of complex data used to plan inspections. The commission wanted to migrate off their Prime computer, simplify the data entry process, and improve data accuracy and integrity. It selected PowerBuilder and Sybase and employed object-based development (i.e., development that is not truly object-oriented but is moving in that direction) to obtain an event-driven system.

Complex data had to be entered into the system and validated. Program shells were used to reduce redundant code, and Sybase triggers and inheritance features were used to further optimize the programming effort. More than 300 stored procedures were used on a data base of 120 tables. Skilled designers were brought in to assist with the shells and to optimize the SQL calls. Additional edits enhanced data integrity.

System hardware consists of HP 9000s with UNIX. PowerBuilder 3.0 runs on the network, with little software loaded on the PCs, which run Windows. The commission has 400 users on one LAN. Because there was little to maintain on the workstations, software distribution was not an issue.

The commission found that the four levels of inheritance and the high degree of common code (not all of it actually used) contributed to a poor response time, as high as five seconds. Although it also discovered that traditional charting methods, such as data flow diagrams, were not meaningful in this object-based environment, nothing has as yet been developed to replace them. The commission also noted the requirement for standards and procedures to handle shells, navigation, libraries, and version control. The entire area of maintenance and migration proved to be problematic.

The project was a success and has been in operation for some time. The commission shared the following important lessons from its experience:

- Expect the first client/server project to be painful—the organization will experience a steep learning curve.
- Identify needs in business terms, not technical terms.
- Take time to set up standards and procedures at the beginning of the project.
- Use experienced staff or find needed skills by hiring consultants, especially for architecture and screen design.
- Generate metrics throughout the project.
- A first project is a time for learning, so start small.
- Build a live prototype that uses actual data and provides some skeleton function, not just screens.
- Start with a simple GUI screen design and enhance it later.
- Ensure that the pilot delivers core function quickly, then it can evolve.
Benefits compound as more applications are built within the same environment and with the same technology.

The commission believes that its first client/server system pays the costs of the learning curve and infrastructure setup, whereas subsequent applications will reap the benefits of the investment. This is one of the reasons it stresses the need for ongoing measurements. The group is moving ahead with additional client/server applications.

**Distribution**

The next case involves a successful PC distributor. In 1993, this group was in difficulty. An overloaded AS/400, serving 1,000 users, was so slow during peak daytime hours that users waited until evening to enter their transactions. Order lead times of 30 to 90 days were becoming common. The distributor needed a new system quickly to meet its business peak in the fourth quarter. In addition, it wanted to implement a single point of service to resolve customer problems, so that sales, technical support, and order entry could all be handled by a single service representative.

The distributor chose UNIX on HP 9000, C++ from Borland, and Sybase, using object-oriented design. Performance was found to be important: although PowerBuilder was easy to use, it did not provide the needed performance.

The distributor surmounted the challenges of new client/server technology and achieved its goal of getting the system implemented before the hectic fourth quarter. It learned several lessons from its experience, including:

- Take the time to understand current system functions.
- Obtain commitment for full-time user involvement.
- Pay attention to performance.
- Acquire the necessary skills for object-oriented design and new tools such as C++.
- Version control, a comprehensive testing methodology, and a strong architecture are all important success factors.

**Retail**

This next case study involves Grossman's, a chain of building supply stores. Grossman's was seeking an enhanced customer interface at the point of sale, an improved in-stock position, and flexibility to allow for future changes. Because the retailer wanted to leverage its skills base, it chose familiar tools: DB2 for OS/2 on the workstations and DB2 for MVS on the mainframe. These tools let the retailer achieve its goals.

Exhibit 2 depicts the system configuration. Using DDCS/2, the retailer distributes data transparently among its network of 138 stores. Through automated replenishment, customer sales information is transmitted to the host, automatically initiating orders for new stock. This relieves purchasing staff of most of the routine administrative work and ensures that customers walking into the store (usually busy contractors) find all the items they need. Radio frequency scanners are used at the loading dock to scan bar codes on incoming shipments, match them to purchase orders, and, as a result, improve accuracy in the warehouse receiving function.
Configuration of a Client/Server System in the Retail Sector

Because Grossman's managers believe that the cashier's interaction with the customer is paramount, they invested in full GUI interface functionality on the cash registers. All job functions are performed there: checking inventory at local or distant warehouse locations, arranging delivery, processing payment, and checking credit. This design reduces training time for new staff from days to hours. The technology includes customized PCs with 14-inch color screens, cash drawers and printers, scanners, servers in each store, and satellite-based credit check services to provide fast response.

Grossman's is satisfied with the results of its new system, which is now fully rolled out and meets all objectives. The cost of inventory management has been reduced, and additional revenue is being generated through shelves that are fully stocked at all times. Among management's conclusions:

- Choose familiar technology to minimize the learning curve.
- Design for maximum flexibility. Grossman's management is satisfied that the technology base is flexible enough so that they can build on it. In the future, for example, mobile sales representatives may travel to construction sites with portable devices to take orders on site.
- Be prepared for a change in business practices.
- Allow for redefinition of job roles. The retailer noted that not all purchasing staff felt comfortable with the change from a solely administrative job to a more creative and proactive role.

Banking

A large international bank has implemented several client/server applications among its more than 1,800 branches. Early on, the bank began with OS/2 Database Manager; now it is using DB2 for OS/2. The LAN servers are PS/2 Model 95s. The bank chose the IBM DB2 for OS/2 software to capitalize on its host DB2 skills, relying on the convergence of the products.

One application, called relationship management, supports account managers by helping them remain informed and in touch with many clients. This is accomplished by combining clients' financial data with other more personal information and downloading it to the PC. For example, the account manager can track the birthdays, anniversaries, hobbies, and personal preferences of key clients. Reminders of important dates are transferred to the manager's E-mail during nightly jobs. The system lets staff provide more personal service to more clients and more easily fill in for a manager who is absent.

Other applications on this platform include 24-hour TeleBanking (i.e., banking through touch-dial responses and interactive voice recognition) using DirectTalk/2 on DB2/2; a financial analysis system using knowledge-based software to assist analysts; and reconciliation between this bank and other banks on interest rates and fluctuating currency exchange rates.

The bank's biggest challenge is remote management of its network: tools are immature and communications lines costly. The bank is currently using IBM's Netview and has
entered into an agreement with another software company to develop a more powerful and extensive network object distribution system. It has not distributed a great deal of data, preferring the security of the mainframe and DB2 for MVS. The bank’s philosophy is: “Where functions are distributed, distribute the data where risk is minimal and security is not an issue.”

Utilities

This example involves a gas company that wanted a new customer information system and to migrate from a CICS/VSAM system on the mainframe to a client/server base. The company entered into a joint development project with two other gas companies that had similar goals, thereby sharing costs. The consortium selected Application Development Workbench (ADW) from Knowledgeware for analysis, and Netron/CAP for design and construction.

CICS for OS/2 was chosen to communicate with the LAN server; DB2 for OS/2 ran on the server, mainly to hold look-up data; and CICS/DB2 resided on the MVS host. The tools generated CICS/COBOL code, and the project team attained 80% code reusability. This project sounded promising; it used innovative techniques and interesting new technologies. Before long, however, problems arose.

With development spread among three locations, communication was a challenge. The degree of similarity of requirements was overestimated; the three groups were in constant conflict over specifications. After three years of work, the project was canceled.

Examining what went wrong yields valuable lessons. First, the planned costs and schedule were too optimistic for the size and scope of this major project. Client/server technology was new to the staff, who needed extra time to learn. There was a large database with a highly normalized database design (i.e., more than 1,500 tables, all linked with referential integrity rules), and no capability to split the data among the servers. The scope of the project was large and included virtually all the utility’s basic business functions—billing, credit, collections, and service. Furthermore, it was designed using a big-bang approach, with no provision for phasing in functions over time. Adding to the complexity and increasing the scope was the need for many bridges to interface with older legacy systems.

The project plan covered five years but did not provide for changes in direction or technology reviews along the way. It did not include a design for distribution of data or function. In addition, although the utility found Netron/CAP to be a powerful tool, it also had a high learning curve. It is fair to say that project management probably contributed as much to this rather spectacular failure as did any technical issues.

What was the end result? Certainly the reputation of the IS group was greatly damaged. For now, they are simply maintaining current systems and evaluating technology alternatives for the future, such as CMOS Parallel technology and UNIX servers.

Manufacturing

This case study concerns Andersen Windows, a large manufacturer of windows and doors with sales in excess of one billion dollars per year. Andersen Windows's methods for handling standard sizes were fairly efficient, but custom orders were processed manually, with handwritten work orders and a resulting high error rate. The manufacturer wanted to manage custom orders as effectively as standard size products. It also wanted many functions integrated, with a single point of entry.
The resulting system handles the entire life cycle: design, order, build, and deliver. Bar codes are used to identify each product, and the products are linked to a customer order. Handheld radio-frequency devices scan the bar codes in the warehouse. The radio frequency devices also collect inventory data and transmit build orders to the manufacturing line to produce any products in low supply. Another innovation is store kiosks that allow customers to design their own windows, using Macintosh, laser disc, and video. The system produces immediate responses to the customers' designs, in the form of specifications, pricing information, and delivery lead times. It has proven to be an effective sales tool.

The hardware configuration comprises Stratus (R310), Sequent 2000/450, PCs (286, 386, 486, and Macintosh), and radio frequency micros from Telxon. Stratus (fault-tolerant) machines control the transaction collection and validation, receipts, and shipments. COBOL programs and flat files are used here. The Sequent processors handle history and batch processing, operating an Oracle database.

Although each component was prototyped and analyzed for performance issues, the heterogenous environment necessitated tuning for every aspect of the system to achieve subsecond response time.

The order fulfillment portion used Forte, an object-oriented tool that was in beta test at the time. Having the tool designers available on site was an important factor in the project's success.

The Andersen people discovered that developers skilled in object-oriented techniques were difficult to find. The Navigator System Series from Ernst & Young was used as the basis for a customized methodology.

As is seen in Exhibit 3, the scanners collect the data; Stratus machines run COBOL code with flat files and act as requester and server. Sequent machines use Oracle and post the data. The host is a Unisys processor.

**Configuration of a Client/Server System in the Manufacturing Sector**

The new system serves more than 600 users and has achieved Andersen's objectives of integrating major processes and improving efficiency. Andersen was able to streamline its order processing while reducing errors to less than 1%.

The team at Andersen learned the following lessons from its client/server implementation:

- Use fewer people for more effective communication.
- Prototype business processes (some processes do not need improvement).
- Test support procedures; these can make or break a system's acceptance.
- Plan on a phased implementation; in this case, a new function was delivered every six months.
- High performance requires a high level of expertise.
Recommended Course of Action

The case studies present a panorama of what several companies in different industries accomplished with their first client/server applications. Despite the disparate business areas, varying sizes of the organizations, and variety of technologies selected, a few common threads appear. The following list represents a distillation of thousands of work-months of effort and experience among these and other organizations:

- Set realistic scope for a first client/server effort.
- Prototype designs in advance.
- Build an infrastructure including methods, standards, and procedures.
- Plan for a steep learning curve.
- Work closely with the business users—add them to the project team.
- Build on existing skills.
- Obtain other specialized skills from outside if necessary.
- Conduct a rigorous tool selection process.
- Improve key business processes.
- Plan on adequate and early training.
- Ensure strong project management.
- Pay attention to performance early in the design.
- Plan project phases to begin delivering benefits quickly.

Many of these lessons do not differ from sound mainframe practices. In fact, it is vital that IS managers not forget the methods and skills that have served well in traditional development. Continuing to deploy these skills and practices helps ensure a successful client/server development project.

Author Biographies

Brenda Castiel

Brenda Castiel is a senior manager with Ernst & Young in Los Angeles. She has more than 20 years of data processing experience, many of which she spent as senior systems engineer and data base specialist with IBM Canada. A frequent speaker at technical conferences and local user groups and author of several articles, she can be reached at Brenda.Castiel@ey.com or 102131.1764@compuserve.com.
Source: Gartner Group, 1994
30 Full-Screen (like 3270 terminals) Requester

Also have 50 RF (handheld) terminals with a separate requester process.