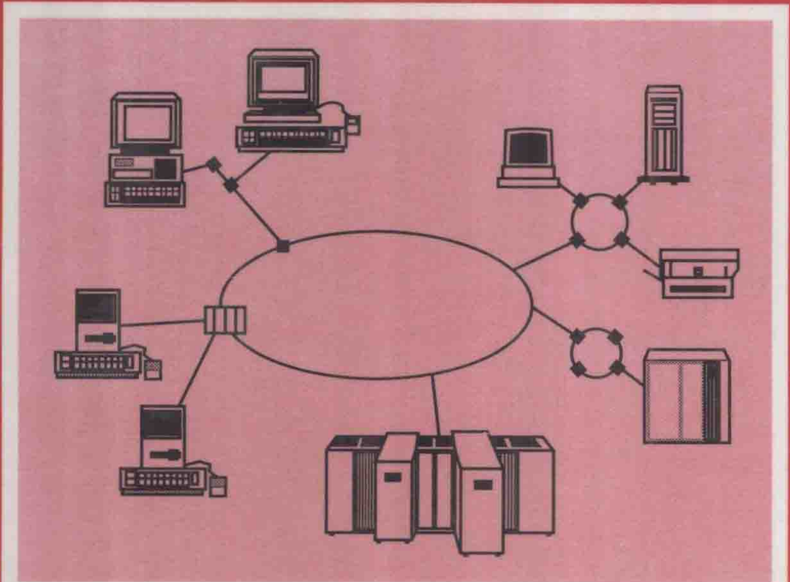
Client/Server Computing

Architecture, Applications, and Distributed Systems Management



Bruce Elbert • Bobby Martyna

Client/Server Computing

Architecture, Applications, and Distributed Systems Management

Bruce Elbert and Bobby Martyna

Artech House
Boston • London

Library of Congress Cataloging-in-Publication Data

Elbert, Bruce R.
Client/server computing: architecture, applications, and distributed systems management/
Bruce Elbert and Bobby Martyna.
Includes bibliographical references and index.
ISBN 0-89006-691-4
1. Client/server computing. I. Martyna, Bobby. II. Title.
QA76.9.C55E53 1994 94-2605
004'.36—dc20 CIP

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library.

685 Canton Street
Norwood, MA 02062

International Standard Book Number: 0-89006-691-4
Library of Congress Catalog Card Number: 94-2605

10 9 8 7 6 5 4 3 2

Preface

No term has invaded the thoughts and expectations of information professionals like *client/server*. It is a multifaceted concept, covering everything from the inner workings of an advanced computer operating system, to a means of communication over a data network, to an architecture for developing and running business applications. While we elaborate on each of these areas, our focus is clearly on the networking and application development aspects.

Client/server computing is relatively immature, as represented by the wide range and make of hardware devices and software products available to develop applications. Attempts to categorize software development tools, for example, have produced terms like GUI builders, frontware, 4GLs, screen scrapers, upper and lower CASE tools, and automatic code generators. The definitions of these terms are often imprecise and there is considerable overlap, duplication, and even misrepresentation of capability. Current and would-be developers of client/server applications for business are concerned about this confusion of purpose and capability of software tools because they must make a considerable commitment to learning how to apply any system and language. The learning curve for a given technology or product is usually measured in months before one can really validate that a particular approach will work in the selected environment. Another consideration is that a given software system will change over time as the vendor adds capability to meet marketplace demand for greater functionality and ease of use. One of the biggest "paradigm shifts" in the area of client/server application development was the introduction of object-oriented programming and rapid application development by leaders like Powersoft, Microsoft, Borland, and Uniface.

First, we propose a general working definition of client/serving computing from a business application development standpoint:

Client/server computing is a process and architecture allowing applications to be developed as efficiently as possible, exploiting the capabilities of personal workstation "clients" and intelligent "server" computers. A client typically uses a multiprocessing operating system with a graphical user interface "front end." The server is dedicated to one or more "back end" functions, particularly the creation, management, and manipulation of databases. Clients and servers can be connected locally or through a wide area network. Application-specific code is usually split between client and server.

This definition addresses a number of major issues and desired outcomes in the field of modern business computing. The notion that client/server application development is efficient and therefore inexpensive continues to be debated. Depending on the size of the application and the using organization, client/server could come out ahead or behind of traditional host-based computing. It is extremely important that the right needs and projects be aimed at this relatively immature process. Another aspect is the well-accepted principle that GUIs are better for people to use because they are more intuitive. Visible objects or *widgets* on the screen (which allow users to point and click with a mouse) behave more like the physical world we live in than do the scripted dialogs of MS DOS and older operating systems like IBM's TSO. The concept of objects is driven further and further into the application and the system through the use of graphical application development tools based on object-oriented technology. The database itself through the relational model represents objects in the form of business entities like customers, employees, engineering designs, and manufactured items.

The server hardware provides the appropriate degree of centralization, whether there is one server on a LAN or several servers providing what is termed a distributed database. Performance is important because any given server must provide application and database support transparently to multiple clients, ranging from a few dozen to a hundred or more. Client/server computing offers the ability to grow the client base, a property called scalability, either by adding servers or by upgrading the computing and memory power of the existing server. The latter can be accomplished by replacing the processor with one that possesses more raw computing power, by adding more RAM and disk storage capacity, or by doing both. Improvements in operating systems, discussed in the following section, can add functionality and improve performance in an environment of growing client demand. Throughput requirements on the interconnecting links can also be improved with higher transmission speeds and better message routing services. These networking topics are discussed in Chapters 3 and 6.

The previous application-oriented definition of client/server computing considers an architecture where the user gains access to the application through a client workstation. Display processing and data checking code would likely be stored on the client while much of the business logic and data would be deployed to the servers, with the network providing the means of connecting the two together. Probably the most common function of the application is to provide the user with efficient access to a database of pertinent information. This is, of course, the typical role of an information system in the corporate environment. The database stores highly structured data concerning many aspects of the business, such as customer records, inventory, financial accounting data, manufacturing records, and control information. The degree to which these data are integrated and available on line depends on many factors. From a technical standpoint, the client/server computing paradigm is capable of supporting any of these functions and can allow whatever degree of integration is appropriate (and, of course, cost effective).

Client/server computing also has a very technical meaning that relates to the underlying interaction between different processes. A specific definition is offered in [Stevens, 1990]:

> The server process is started on some computer system. It initializes itself, then goes to sleep waiting for a client process to contact it requesting some service. A client process is started, either on the same system or on another system that is connected to the server's system with a network. Client processes are often initiated by an interactive user entering a command to a timesharing system. The client process sends a request across the network to the server requesting service of some form. Some examples of the type of service that a server can provide are: return the time of day; print a file on a printer for the client; read or write a file on a server's system for the client; allow the client to log on to the server's system; and execute a command for the client on the server's system.

In this process model, the client issues a request and the server provides a response or series of actions. The server can either act immediately upon the request or add it to a queue and deal with it concurrently with requests from other clients. A process in this context is typically a program running within a partition of the computer's main memory. The client process takes the first step by sending a message to awaken the server process. The server does not initiate communication—it effectively listens for client requests. This can be considered to be event driven, because the client starts the action (that is, it creates the event) while the server stands idle. In the real world of client/server architecture, the client machine could actually provide server process functionality to the physical server machine, which in turn is actively engaged through its own client process. The best example of this is the X Window System, discussed later in this chapter, wherein the client display terminal provides graphical display X service to an external server computer that acts as an X client by requesting display services.

Our goal is to find the right environment and applications for the potential client/server architecture. This considers the various classes of information systems employed by organizations to meet business needs. A hierarchy of information systems is suggested in Table P.1, indicating the predominant three levels of (1) on-line transaction processing (OLTP), (2) management information systems (MIS), and (3) decision support systems (DSS). The lower an application is on this stack, the more crucial it is to the day-to-day operation of the business.

OLTP systems are tightly coupled to value chain activities of the business since they provide the means to identify and serve customers. A "transaction" in this environment is a single unit of work that must be completed in its entirety through the use of an integrated data processing environment. An example of a transaction is a deposit in a bank account, which actually consists of several steps: the teller transmits the customer name, account type and number, and the amount to be deposited (or withdrawn, transferred, etc.); the application makes the change in the bank database and returns the new balance to the teller.

Table P.1
Classes of Applications in the Hierarchy of Information Systems

Application Class	*Value*	*Examples*
Decision support systems (DSS)	Growing in importance	• Sales trend forecasting • 5-year budgeting • Profit planning
Management information systems (MIS)	Important	• Sales management • Inventory control • Capital investment analysis
On-line transaction processing systems (OLTP)	Critical to the business	• Order processing and tracking control • Material movement, • Accounting systems

Such a transaction involves several steps that could cause changes to more than one database and multiple messages across the network. Failure of any one step would require that the entire transaction be nullified, or rolled-back in OLTP parlance.

OLTP systems provide operational computing services—that is, they must be running at all times so that business transactions can be processed in a timely manner. Failure of an OLTP system may have serious effect on the business unless service is restored within an acceptable time frame. OLTP systems are used by banks, airlines, package delivery services, and telemarketing organizations live or die by their OLTP systems. The associated data is maintained as current records within a very dynamic database and processing environment. If this is a distributed system, then the management of it is a complex task since users at different locations may need immediate access to the same business records more or less simultaneously.

Moving up the hierarchy we encounter MIS systems that are important to the business but that are not nearly as time critical as OLTP. The need here is for accuracy and ease of access; the time frame can be somewhat delayed. At the MIS layer we find critical services that allow managers to control their respective operations. Most companies use an MIS to generate and print the reports that are distributed in paper form. Examples include records of daily sales by product category, financial summaries of each business unit, and manufacturing output summarized on a daily or weekly basis. Managers at all levels require information that is accurate and properly categorized. Various types of reports, which are generated according to previously defined formats, are usually appropriate. Whether the information is available on line, such as from an executive information system (EIS), depends on the nature of the business.

Organizations typically use their information bases to provide summary data concerning the overall performance of the business. The DSS allows analysts to derive information from the MIS and OLTP systems, as appropriate, to do statistical analysis leading to decisions about possible changes in marketing or business strategy. Long-term financial analyses are used to assess where the company is going and what results might be expected under a variety of assumptions about markets, demographics, economic and political situations, and so on. Analysts who retrieve the data and massage it need a variety of statistical and report writing tools. Most often, large quantities of accounting and marketing data are introduced and summarized. The OLTP and MIS systems may have been the source of the data, but it is generally accepted that DSS imposes much different requirements for access to records, computation, and display of results.

The three levels of information systems have different demands on a potential client/server architecture. An optimum OLTP system would not necessarily be a good platform for DSS (in fact, experience has shown that it is not). Vendors of DBMSs and host computing systems argue that this is possible—a possibility worth achieving. The consequence of forcing such a combination might be severe performance degradation of OLTP service during peak requirements (when customer activity is maximum). As a general rule, every situation is different and the demands of each of the three levels are no exception.

Client/server computing can address many of the information system needs of an organization. The basis for this is the benefits that the architecture offers. The client side is embodied in an intelligent workstation running a modern stand-alone operating system. The associated raw computing power and memory capacity (random access and disk) give us many options as to how to implement the architecture. The GUI makes using the system as easy and enjoyable as possible. Exploiting local computing power to provide part of the application software is possible, but must be considered carefully as it does complicate software maintenance. The user interface is built right on the client and can be demonstrated to users early in the development process. No longer does the host have to generate the user display, even for prototyping purposes. Some user-based aspects can also be attached to the client, such as formatting of queries to the system, generation of specialized reports, and even creation of part of the application by the user.

The application will require some centralization of the data and part of the application code. Server hardware may not be involved in the development stage but certainly is key to data management and network access. The server hardware and relational database platforms must have the capacity and performance to support all users in whatever role they play. This brings in the greatest complexity to such a project and operating environment. An unstructured approach will not work as the number of users and locations increase. For this reason, managers should expect that major projects will be complex and lengthy, as much as (or possibly more than) their mainframe counterparts.

Up until 1992, success stories involved relatively small projects in the DSS area and were not what are termed "industrial-strength" applications. As we move toward the year 2000, our customers are demanding full functionality and mainframelike reliability for OLTP and MIS applications. Business unit leaders leap at client/server computing in a wish to maintain their independence and achieve more responsiveness than they receive from traditional MIS departments. The good news is that our new computing and application development paradigm can meet many expectations for a quick, effective, and attractive information system. On the other hand, not every project is a good candidate for client/server application development.

Mainframe experts see that their expertise with centralized data management does not directly transfer to distributing computing. Their counterparts on the network side are also having trouble keeping up with new protocol stacks and LAN and WAN hardware and services. That you can teach such "old dogs" new tricks is not the question. The technologies employing GUI builders and database management systems are learnable and usable within a reasonable time frame. The key is to match the right hardware, development tools, and people with the application problems.

Conversion of obsolete application systems and databases is a massive task, amounting to man years of effort for many organizations. This is why many employ the strategy of leaving the legacy applications on the mainframe and using client/server computing to satisfy new requirements. Gradually, the legacy applications become obsolete to the business and can be turned down. These residual needs typically are no longer strategic to the business and often can be outsourced to a service bureau with the appropriate equipment and expertise.

The challenge of client/server computing is to tie together the business need and the technology platform, coupling the unique environment of the organization. Some companies are comfortable doing their own development—for them, the full range of the technology is available. They can create applications using the simplest tools to save time, but are able to dive as deeply as needed into programming using the most sophisticated development environments, including 3GLs. For a majority of companies, the best approach will be to limit the search to the easiest and most intuitive 4GLs, some of which eliminate the need for coding. These exploit the ease of use of the GUI development platform along with the best that object-oriented technology has to offer.

The trade-off is between ease of use and flexibility of GUI 4GL tools on the one hand versus the power and diversity of the full range of development tools (whatever GL) on the other. Another aspect of this tradeoff is the role of open systems and standards (see Chapters 8 and 9). Industry leaders IBM, DEC, and HP have banded together to employ portable operating systems and network architectures. Alternatively, there are nonpartisan groups and international standards bodies pursuing open systems and networking. Selecting a proprietary client/server architecture may meet today's need but can pose a problem if the vendor behind the technology withdraws from the market. If you can satisfy your requirements in a cost-effective way with a very specialized system, then is makes sense to proceed. But you will need to keep one eye open for changes in your environment lest the solution erode in value too quickly.

Client/Server Computing is organized into 12 chapters, each corresponding to a major building block. Chapter 1 sets the stage by defining industry terminology and providing some concrete examples of commercial products that can be used to structure a high-performance client/server environment. Some of the most pressing issues that confront the creation and use of client/server systems are identified and resolved in Chapter 2, which covers strategic applications. The data networking architecture is further refined in Chapter 3, where protocols like OSI, TCP/IP, SNA, and DECnet are analyzed from the standpoint of their relevance to modern networks of computers.

Strategies and technologies for building the client/server architecture are covered in detail in Chapter 4. The chapter reviews client and server hardware platforms and operating systems, relational databases, and techniques for dividing up the data in a distributed computing environment. Chapter 5 takes us through the programming methodologies and tools that make client/server applications possible and effective. None of this would be possible, however, without facilitating software and development environments that simplify the job of the programmer and systems architect. These aspects are covered in Chapters 7 and 8. Here, we get into the subjects of middleware, APIs, and open environments like OSF DCE and the Object Management Group. Open network standards, particularly OSI, as introduced in Chapter 9 as a means to resolve the incompatibilities among different vendors communications services and products.

One of the most complex issues in client/server computing management of the infrastructure, which is no long centralized. The topic of distributed systems management is evaluated in detail in Chapter 10, wherein many contemporary products and solutions are offered. IBM and HP are expending considerable financial and technical resources to dominate the field. However, smaller firms like Tivoli Systems and Teknekron Communications Systems are also creating software that can turn clients into effective management tools. Network performance measurement and prediction, covered in some detail here, provide the basis for forecasting network demand and troubleshooting problems nearly in real time. Associated with management is the question of system and network security, which is made more complex by the wide array of devices and networks. Chapter 10 introduces such topics as the Kerberos distributed security system and the highly regarded encryption products of RSA Data Security.

Groupware, covered in Chapter 11, is an attractive application of client/server computing that addresses many requirements of widespread organizations that need to stay coordinated. Current products like Lotus Notes offer less technical sophistication that an advanced 4GL but allow users to tailor the environment to their particular needs. This is an evolving field, so what is presented here can only be regarded as a snapshot in time. Finally, Chapter 12 projects what client/server computing systems of the future might be capable of doing, based on research underway in the U.S. and Europe with high-speed network architectures, particularly ATM. The information superhighways of the future will build upon this foundation, and client/server computing will no doubt go through an metamorphosis as a result.

The idea and much of the material for this book came out of a short engineering course the authors presented at UCLA Extension. Newer material was added as we continued to conduct seminars in client/server applications through the facilities of Technology Training Corporation. Anyone interested in creating a course around the book is invited to contact Bruce Elbert to obtain teaching materials we have used over the past two years.

We wish to express our appreciation to Bill Goodin and Don Hausknecht of UCLA and Mark Mitchell of TTC for the opportunity to develop this topic and bring it to audiences on three continents. The creation of this book particularly benefits from the help offered by William M. Bazzy and Mark Walsh of Artech House, both of whom supported us throughout this project.

The Artech House Telecommunications Library

Vinton G. Cerf, Series Editor

Advances in Computer Communications and Networking, Wesley W. Chu, ed.

Advances in Computer Systems Security, Volume III, Rein Turn, ed.

Asynchronous Transfer Mode Networks: Performance Issues, Raif O. Onvural

A Bibliography of Telecommunications and Socio-Economic Development, Heather E. Hudson

Broadband: Business Services, Technologies, and Strategic Impact, David Wright

Broadband Network Analysis and Design, Daniel Minoli

Broadband Telecommunications Technology, Byeong Gi Lee, Minho Kang, and Jonghee Lee

Cellular Radio Systems, D. M. Balston and R. C. V. Macario, eds.

Client/Server Computing: Architecture, Applications, and Distributed Systems Management, Bruce Elbert and Bobby Martyna

Communication Satellites in the Geostationary Orbit, Donald M. Jansky and Michel C. Jeruchim

Computer Telephone Integration, Rob Walters

Computerized Document Imaging Systems: Technology and Applications, Nathan J. Muller

The Corporate Cabling Guide, Mark W. McElroy

Corporate Networks: The Strategic Use of Telecommunications, Thomas Valovic

Design and Prospects for the ISDN, G. Dicenet

Digital Cellular Radio, George Calhoun

Digital Switching Control Architectures, Giuseppe Fantauzzi

Digital Transmission Design and Jitter Analysis, Yoshitaka Takasaki

Distributed Processing Systems, Volume I, Wesley W. Chu, ed.

Disaster Recovery Planning for Telecommunications, Leo A. Wrobel

EDI Security, Control, and Audit, Albert J. Marcella and Sally Chan

Enterprise Networking: Fractional T1 to SONET, Frame Relay to BISDN, Daniel Minoli

Expert Systems Applications in Integrated Network Management, E. C. Ericson, L. T. Ericson, and D. Minoli, eds.

FAX: Digital Facsimile Technology and Applications, Second Edition, Dennis Bodson, Kenneth McConnell, and Richard Schaphorst

Fiber Network Service Survivability, Tsong-Ho Wu

A Guide to Fractional T1, J. E. Trulove

A Guide to the TCP/IP Protocol Suite, Floyd Wilder

Handbook of Satellite Telecommunications and Broadcasting, L.Ya. Kantor, ed.

Implementing EDI, Mike Hendry

Implementing X.400 and X.500: The PP and QUIPU Systems, Steve Kille

Inbound Call Centers: Design, Implementation, and Management, Robert A. Gable

Information Superhighways: The Economics of Advanced Public Communication Networks, Bruce Egan

Integrated Broadband Networks, Amit Bhargava

Integrated Services Digital Networks, Anthony M. Rutkowski

Intelligent Hubs, Nathan J. Muller

International Telecommunications Management, Bruce R. Elbert

International Telecommunications Standards Organizations, Andrew Macpherson

Internetworking LANs: Operation, Design, and Management, Robert Davidson and Nathan Muller

Introduction to Satellite Communication, Burce R. Elbert

Introduction to T1/T3 Networking, Regis J. (Bud) Bates

Introduction to Telecommunication Electronics, A. Michael Noll

Introduction to Telephones and Telephone Systems, Second Edition, A. Michael Noll

Introduction to X.400, Cemil Betanov

The ITU in a Changing World, George A. Codding, Jr. and Anthony M. Rutkowski

Jitter in Digital Transmission Systems, Patrick R. Trischitta and Eve L. Varma

Land Mobile Radio System Engineering, Garry C. Hess

LAN/WAN Optimization Techniques, Harrell Van Norman

LANs to WANs: Network Management in the 1990s, Nathan J. Muller and Robert P. Davidson

The Law and Regulation of International Space Communication, Harold M. White, Jr. and Rita Lauria White

Long Distance Services: A Buyer's Guide, Daniel D. Briere

Mathematical Methods of Information Transmission, K. Arbenz and J. C. Martin

Meteor Burst Communication, Jacob Z. Schanker

Minimum Risk Strategy for Acquiring Communications Equipment and Services, Nathan J. Muller

Mobil Communications in the U. S. and Europe: Regulation, Technology, and Markets, Michael Paetsch

Mobile Information Systems, John Walker

Narrowband Land-Mobile Radio Networks, Jean-Paul Linnartz

Networking Strategies for Information Technology, Bruce Elbert

Numerical Analysis of Linear Networks and Systems, Hermann Kremer et al.

Optimization of Digital Transmission Systems, K. Trondle and Gunter Soder

Packet Switching Evolution from Narrowband to Broadband ISDN, M. Smouts

Packet Video: Modeling and Signal Processing, Naohisa Ohta

Principles of Secure Communication Systems, Second Edition, Don J. Torrieri

Private Telecommunication Networks, Bruce Elbert

Radiodetermination Satellite Services and Standards, Martin Rothblatt

Residential Fiber Optic Networks: An Engineering and Economic Analysis, David Reed

Secure Data Networking, Michael Purser

Setting Global Telecommunication Standards: The Stakes, the Players, and the Process, Gerd Wallenstein

Signal Processing with Lapped Transforms, Henrique S. Malvar

The Telecommunications Deregulation Sourcebook, Stuart N. Brotman, ed.

Television Technology: Fundamentals and Future Prospects, A. Michael Noll

Telecommunications Technology Handbook, Daniel Minoli

Telephone Company and Cable Television Competition, Stuart N. Brotman

Terrestrial Digital Microwave Communciations, Ferdo Ivanek, ed.

Transmission Networking: SONET and the SDH, Mike Sexton and Andy Reid

Transmission Performance of Evolving Telecommunications Networks, John Gruber and Godfrey Williams

Troposcatter Radio Links, G. Roda

UNIX Internetworking, Uday O. Pabrai

Virtual Networks: A Buyer's Guide, Daniel D. Briere

Voice Processing, Second Edition, Walt Tetschner

Voice Teletraffic System Engineering, James R. Boucher

Wireless Access and the Local Telephone Network, George Calhoun

Wireless LAN Systems, A. Santamaria and F. J. López-Hernández, eds.

Writing Disaster Recovery Plans for Telecommunications and LANs, Leo A. Wrobel

X Window System User's Guide, Uday O. Pabrai and Hemant T. Shah

For further information on these and other Artech House titles, contact:

Artech House
685 Canton Street
Norwood, MA 01602
(617) 769-9750
Fax:(617) 762-9230
Telex: 951-659
email: artech@world.std.com

Artech House
Portland House
Stag Place
London SW1E5X4 England
+44(0)71 973-8077
Fax:+44(0)71 630-0166
Telex: 951-659
email: artech@world.std.com

Contents