Client/Server System Design & Implementation

Larry T. Vaughn

Shaku Atre,
Series Advisor

McGraw-Hill Series on Computer Communications

Client/Server System Design and Implementation

Larry T. Vaughn

McGraw-Hill, Inc.
New York San Francisco Washington, D.C. Auckland Bogotá
Caracas Lisbon London Madrid Mexico City Milan
Montreal New Delhi San Juan Singapore
Sydney Tokyo Toronto

Library of Congress Cataloging-in-Publication Data

Vaughn, Larry T.
Client/server system design and implementation / Larry T. Vaughn.
p. cm — (McGraw-Hill series on computer communications)
Includes index.
ISBN 0-07-067375-6
1. Client/server computing. 2. System design. I. Title.
II. Series.
QA76.9.C55V38 1994
004'.36—dc20 94-1539
CIP

2 3 4 5 6 7 8 9 0 DOC/DOC 9 0 9 8 7 6 5 4

ISBN 0-07-067375-6

The sponsoring editor for this book was Jerry Papke, the editing supervisor was Paul R. Sobel, and the production supervisor was Pamela A. Pelton. It was set in Century Schoolbook by McGraw-Hill's Professional Book Group composition unit.

Printed and bound by R. R. Donnelley & Sons Company.

Preface

Every industry and technical discipline has its own vocabulary. Although characterized by many as *techno-babble*, such technical vocabulary serves a valuable purpose in that it is often needed to introduce and describe truly new concepts. Relatively immature, rapidly evolving, and technology driven, the information processing industry's technical vocabulary has grown at an exponential rate. Unfortunately, much of this growth has come from hardware and software vendors seeking to gain competitive market advantage for their newest whiz-bang technology, industry publications seeking to boost their circulation, or technology analysts wishing to be the first to identify the latest breakthrough.

As a consultant, I should welcome this veritable blizzard of techno-babble. After all, the more confused the consumers of this technology are, the greater the need for so-called experts to sort the wheat from the chaff. But as a developer seeking to deliver system solutions I find that the "noise" ratio of hype to truth is becoming a serious obstacle in identifying truly innovative technologies that can be applied to real-world problems. Batch processing, database management, centralized processing, client/server processing, data modeling, distributed processing, hierarchical databases, local area networks, mainframe, minicomputer, multiprocessing, multithreading, multiuser, object orientation, peer-to-peer, personal computer, relational databases, server, structured analysis, transaction processing, virtual memory, wide-area networks, workstation, etc. What are these things? Which represent truly significant new solutions and tools, which represent the latest market hype, and which are just technologies looking for a problem to solve?

This book was written for the information systems professionals who are tasked with providing highly functional and cost-effective system solutions in support of their organization's business solutions and who are contemplating or are already in the process of delivering those solutions through client/server technologies. It provides both context and detail to address the needs of the following:

- The information systems executives tasked with integrating information systems into the basic fabric of the enterprise's operations and strategy and who need to understand the basic technological foundations, benefits, and obstacles that must be overcome before this new technology can be deployed effectively.
- The systems development line managers who have to face the move to the more open, flexible, and less stable client/server environments while continuing to support legacy systems based on closed proprietary development platforms.
- The project leaders tasked with installing the technological infrastructure, choosing the appropriate tools, and developing new systems structured and designed specifically to take advantage of the strengths of client/server technologies.
- The developers seeking to deepen their understanding of the new and totally alien environment within which they will be designing and building system solutions.

This book is divided into three parts. Part 1 provides a definition of client/server architecture, outlines the basic differences between this architectural approach and the traditional centralized mainframe-centric architecture, and discusses the impact this change can make to your enterprise. It identifies the specific benefits that are promised by the architecture and, more importantly, discusses the obstacles that must be surmounted and pitfalls to be avoided before these benefits can be realized.

Part 2 identifies and explains the technologies used by the client/server architecture. It breaks these technologies down into client, network, and server components and discusses how these technologies interact to provide a cost-effective, high-performance and stable foundation for delivering business system solutions to the enterprise. It discusses how these changes impact the application developer attempting the transition from

the traditional mainframe environment and provides guidelines and suggestions for selecting appropriate technologies to establish a stable foundation.

Part 3 offers a perspective that can help you decide if migrating to a client/server architecture is right for your enterprise and provides the framework for planning, preparing, and executing a strategy to develop and deliver client/server system solutions to your enterprise. It discusses planning and preparation and the activities necessary to establish an infrastructure within which development can take place. It identifies tools and techniques that have proven useful to capture business needs and develop applications that meet those needs. Finally, it proposes a new approach to application development that incorporates modern concepts and principles that are emerging from the research and experiments being conducted and nurtured by such organizations as the American Society for Manufacturing Excellence, the American Society for Quality Control, the American Institute of Research, and the American Quality Foundation, and implemented and proven by forward thinking enterprises throughout the industrial segments of our economy.

As an information processing professional over the past 18 years, I have studied, worked with, fought against, fought for, and tried to apply practically every new whiz-bang to come down the pike, and I have come to the conclusion that there are very few that represent truly significant advances in the way system solutions are developed and delivered. The impact of these few has been dramatic, each representing a significant and profound change in the way developers approach problems and deliver solutions. I believe the client/server approach to application design to be such a development, as significant in its impact as the shift from batch processing to transaction processing, tape data storage to multi-user databases, and the introduction of the personal computer.

But what is client/server processing? Where does it fit in today's environment? How is it different from current practice? How is it the same? What impact will it have on your organization? My goal in writing this book is to answer these questions and hopefully arm the information systems professional, technical managers, and MIS executives with the basic knowledge needed to apply this approach to real-world problems in their

organizations. Do not be misled by the objective and frank discussion of the good, the bad, and the ugly aspects of the client/server approach. I am an unabashed promoter of the client/server approach and believe it represents a genuine paradigm shift in the development of computer applications, and as a fellow information systems professional, I hope that you find this book to be a useful tool in your own application of this approach.

Acknowledgments

It is essential to remember that all successful systems development projects are a collaborative effort of individuals with varying backgrounds and skills who come together as a team to focus on a specific objective and come away from the process with greater personal and professional understanding. Ultimately, the only unsuccessful development effort is the one from which nothing is learned.

To celebrate one such special team I'd like to take this opportunity to acknowledge the contributions of Peter Thawley, Annette Runckel, Dave Webb, Katharine Hanson, and Bruce Randall to the experience which made this book possible.

Larry T. Vaughn

Contents

Preface xi

Part 1. Introduction

Chapter 1. Client/Server Architecture 3

Centralized Multiuser Architecture 4
Distributed Single-User Architecture 6
Client/Server Architecture as a Design Approach 9

Chapter 2. Benefits, Promises, and Obstacles 13

Benefits 13
Obstacles 17

Part 2. Technological Foundations

Chapter 3. The Foundations 25

Selecting Appropriate Technologies 27
The Foundation Technologies 33

Chapter 4. Client Technologies 35

Applications and Tools 36
Operating System 40
Hardware Platform 44
Database Access 46
Interprocess Communication Protocols 52

Chapter 5. Network Technologies 55

Local Area Network 56

The OSI Model 60
Reality 64

Chapter 6. Server Technologies 73

Server Operating System 75
Server Hardware 77
Database Access 78
Distributed Data Access 79
Database Engine 80

Chapter 7. Configurations 83

Sybase SQLServer Unix-to-Unix Configuration 84
MS-Windows to OS/2 SQLServer 85
Novell Network with Named Pipes 86
Novell Network without Named Pipes 87
MS-Windows to Oracle v7.0 Unix 88
Using Dynamic Data Exchange 90
Using Gateways to Establish Connectivity 93
Using ODBC and APPC 94
Summary 95

Chapter 8. The Database Engine 97

The Rational Model 98
Object-Oriented Models 113
Summary 114

Part 3. Building Client/Server Applications

Chapter 9. Planning 117

Planning Considerations 117
The Planning Framework 121
Planning for Change 123

Chapter 10. The Organizational Challenge 127

The Corporate Data Center 128
Redefining the MIS Organization 132
Skills Assessment and Training 134
Establishing Standards 135
Redefining Systems Development 137
Getting Started 144

Chapter 11. Methodology 149

Solution Definition 154

Solution Development 157
Solution Implementation 159
Continuous Improvement 161
Summary 162

Chapter 12. Reengineering Tools and Techniques 165

Data-Flow Analysis 165
Workflow Analysis 167
Cause-and-Effect Diagrams 169
Brainstorming 170
Rapid Prototyping 171

Chapter 13. Data Analysis 173

Information Modeling 174
The Conceptual Model 177
The Logical Model 182
The Physical Model 184
Summary 187

Chapter 14. Architecture Design 189

Structure Chart 189
Rapid Prototyping 192
Object-Oriented Analysis and Design 192
Benchmarking 195

Chapter 15. Client/Server-Specific Design Issues 199

Functional Distribution 199
Data Distribution 204

Chapter 16. Futures 211

Business Needs 211
Standards 212
Object-Oriented Programming 213
Fully Distributed Processing 213
Information Presentation 213
Conclusion 214

Bibliography 215
Index 217

Part

1

Introduction

Chapter

1

Client/Server Architecture

What is "client/server architecture"? Webster states that architecture is "the art and science of designing and erecting a building." Within the context of information systems we could then say that client/server architecture must be an approach to the design of an information system. As a general definition, this is fine as far as it goes but fails to describe the approach and what differentiates it from other approaches. There appears to be general agreement that client/server architecture is an approach to the design of a software application that decomposes the application into a small number of server functions that provide commonly used services and a larger number of client functions that perform more narrowly defined work in reliance on the common services provided by the server functions. Even this more descriptive definition does not go far enough to fully encompass the complex nature of the approach. Almost implicit in the approach is the assumption that the server and client functions of the resulting application will be executing on different interconnected hardware platforms. An expanded and more fully descriptive definition would therefore be

> Client/server architecture is an application design approach that results in the decomposition of an information system into a small number of server functions, executing on one or more hardware platforms, that provide commonly used services to a larger number of client functions, executing on one or more different

but interconnected hardware platforms, that perform more narrowly defined work in reliance on the common services provided by the server functions.

Even with a full definition and despite the large numbers of articles and seminars on this subject, there is still confusion in many information system professionals' minds regarding the application of this approach and its impact in organizations using more traditional architectures. Like art, where few can agree on a definition but everyone knows it when they see it, "client/server architecture" means different things to different people depending on their perceptions and backgrounds. Because of this any meaningful discussion of client/server architecture should begin with a review of the other architectural approaches most commonly used.

Centralized Multiuser Architecture

An architectural approach that designs an application so that all functional and data components of the application reside and execute upon one centralized computing platform (Fig. 1.1) used by multiple simultaneous users of that application. Examples of applications using this approach include order entry, accounting, manufacturing control, automated teller machines, and reservation systems.

This approach is the most common architecture used for business applications based on mainframe and minicomputer technologies, and its use is driven by a number of factors, the most important of which are

- The need to leverage the costly investment in hardware, software, and technical support staff represented by these cen-

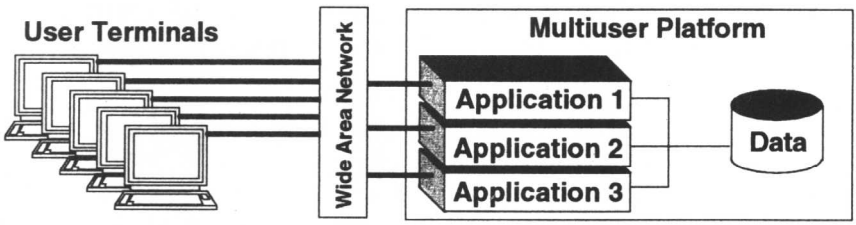

Figure 1.1 Centralized multiuser architecture.

tralized computers across as many applications and users as possible.

- The need to provide large numbers of simultaneous users (200 to 10,000+) with reliable and stable access to one or more special-purpose applications.
- The need to provide centralized storage for very large databases shared by many simultaneous users.
- The need to minimize the amount of data that flows across relatively slow (9.6 kbits/s to 1.5 Mbits/s) and expensive wide area networks.

Strengths

- This technology tends to be very stable, reliable, and well supported by responsible original equipment manufacturers (OEMs).
- It is capable of providing cost-effective application functionality and shared data access to thousands of users.
- A single OEM vendor can often provide all the system-level hardware, software, and networking components. This "one-stop shopping" capability significantly simplifies the administration and management of the environment.
- There is a large pool of highly skilled technical staff who are available to provide technical, operational, and developmental support.
- Business application software is commercially available from the OEMs and third-party vendors across a wide range of categories.

Weaknesses

- Technologies are proprietary and, with a few exceptions, generally incompatible across OEM vendors. In some cases this incompatibility even extends to different model lines from the same vendor.
- Technology within this category is expensive to acquire. Implementation costs can also be substantial as these platforms often require controlled environments with raised flooring, massive air- and liquid-cooling plants, sophisticated

power distribution, and special-purpose fire- and water-damage control systems.

- These technologies require large support staffs of personnel who are highly skilled in relatively narrow technical disciplines.
- Third-party business and system applications are commercially available from only a relatively limited number of vendors. License fees are generally based on hardware capacity and are expensive. For example, the mainframe version of a $1000 functionally equivalent personal computer database management product can cost between $100,000 and $400,000.
- The performance characteristics of multiuser systems often result in significant upgrade costs to support small incremental increases in demand as total system capacity is approached.

Distributed Single-User Architecture

This approach designs an application so that all functional and data components of the application reside on a single computing platform (Fig. 1.2) dedicated to the use of only one person at a time. The most common examples of applications using this approach are those which we generally categorize as personal productivity aids and include word processing, spreadsheets, graphics, and personal database applications.

A variation of this approach, made possible by the development of local area networking technology, allows single-user applications to provide a limited form of simultaneous shared

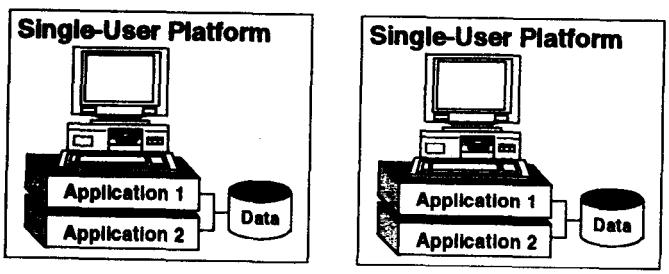

Figure 1.2 Isolated single-user architecture.

access to data across the platforms interconnected by the network. Implementations of this variation (Fig. 1.3) can take many forms, depending on the capabilities of the local area networking technology being used, but the application's architecture is still essentially single-user. The application "views" the data as if it were coresident on the same platform as the application and is still designed to be independently executable on a single platform by a single user.

In either its stand-alone or networked versions this approach is the most commonly used architecture for personal productivity applications. The primary factors driving its use are

- The purpose of the application itself as a "personal" tool, used to perform a limited function for a single user without needing to simultaneously provide access to a common shared repository of data.
- The ready availability of powerful, relatively inexpensive, and standardized hardware platforms and operating systems.
- The commercial availability of powerful and inexpensive applications.

The multiuser variation of this approach evolved in response to

- The availability of fast (≥10 Mbits/s) local area network technology, making it possible to inexpensively interconnect multiple single-user computers.

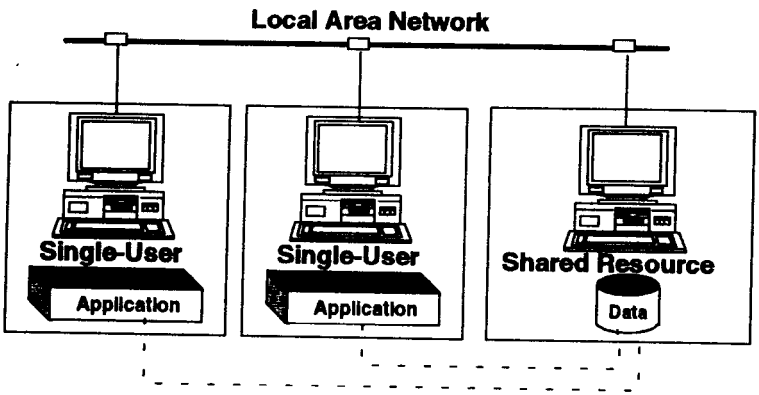

Figure 1.3 File server architecture.