NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CLIENT/SERVER COMPUTING AND APPLICATIONS

Graduation Project
COM- 400

Student: Mahmoud Abuhmaid (20012003)

Supervisor: Prof. Dr Dogan Ibrahim

Nicosia - 2004

ACKNOWLEDGMENT

"First I would like to thank my God for giving me the privilege strength and power. Secondly I would like to thank my supervisor Assoc. Prof. Dr Dogan Ibrahim for supervising my work. Under the guidance of him. I successfully overcome many difficulties and learned a lot about Client/Server computing and Applications. In each discussion, he used to explain and answer. While teaching, he always helped me a lot either in my study or my life, and I felt my quick progress from his advises.

I would like to give special thanks to my mother, she has supported me though out every step of my life for which I shall never forget it till the day I die and my father for without him I would have not been what I am today. On this note I would like to express my gratitude to Near East University for the knowledge that they supported me, and I would like to thank my teachers, I am forever in their gratitude for having the belief in me and not giving upon me. I am very grateful to all the people in my life, who have advised me, and who have always encouraged me to follow my dreams.

Special thanks for my friend Abdulmajeed Jaffer for supporting me and to complete my project and my roommate Mohammad Katabh.

Finally, I would like to thank all my friends who support me all the time and for spending a nice time with them in this university and in this country."

ACKNOWLEDGMENT i
ABSTRACT ii
INTRODUCTION iii
CHAPTER ONE - INTRODUCTION TO TCP/IP
1.1 An Overview of TCP/IP Components 2
1.1.1 Telnet 2
1.1.2 File Transfer Protocol 3
1.1.3 Simple Mail Transfer Protocol 3
1.1.4 Kerberos 3
1.1.5 Domain Name System 4
1.1.6 Simple Network Management Protocol 4
1.1.7 Network File System 4
1.1.8 Remote Procedure Call 4
1.1.9 Trivial File Transfer Protocol 5
1.1.10 Transmission Control Protocol 5
1.1.11 User Datagram Protocol 5
1.1.12 Internet Protocol 5
1.1.13 Internet Control Message Protocol 5
1.2 TCP/IP History 6
1.3 Berkeley UNIX Implementations and TCP/IP 8
1.4 OSI and TCP/IP 8
1.5 TCP/IP and Ethernet 10
1.6 IP Addresses 11
CHAPTER TWO - OVERVIEW OF CLIENT/SERVER COMPUTING
2.1 General 15
2.2 Evolution of Client-Server Computing 15
2.3 Configurations in Client-Server Computing 16
2.3.1 Client 16
2.3.2 Server 17
2.3.3 Middleware 18
2.3.4 Two-Tier Architecture 19
2.3.5 Three-Tier Architecture 19
2.3.6 The Cooperative Processing 20
2.3.7 The Distributed Processing 20
2.3.8 The Intranet 21
2.4 The Four Dominant Client/Server Application Models 21
2.4.1 SQL databases 22
2.4.2 Transaction Processing (TP) monitors 23
2.4.3 Groupware 23
2.4.4 Distributed objects 24

2.5 Characteristics and Features in Client-Server Computing 25
2.5.1 Consists of a networked webs of small and powerful machines 25
2.5.2 Open Systems 26
2.5.3 Modularity 26
2.5.4 Cost Reduction and Better Utilization of Resources 26
2.5.5 Complexity 27
2.6 Main Applications 27
2.6.1 Database Access 27
2.6.2 Transaction-Processing Applications 28
2.6.3 Office Systems 28
CHAPTER THREE - ADVANTAGES OF CLIENT/SERVER COMPUTING
3.1 General 29
3.2 The Advantages of Client/Server Computing 30
3.2.1 Enhanced Data Sharing 30
3.2.2 Integrated Services 30
3.2.3 Sharing Resources Among Diverse Platforms 31
3.2.4 Data Interchangeability and Interoperability 31
3.2.5 Masked Physical Data Access 32
3.2.6 Location Independence of Data and Processing 32
3.2.7 Centralized Management 32
3.3 Connectivity 33
3.4 User Productivity 34
3.5 Ways to Improve Performance 35
3.5.1 Offload Work to Server 35
3.5.2 Reduce Total Execution Time 35
3.5.3 Use a Multitasking Client 36
3.6 How to Reduce Network Traffic 37
3.6.1 Minimize Network Requests 37
3.6.2 Ease Strain on Network Resources 39
3.6.3 How to Reduce Costs 39
3.7 Vendor Independence 40
3.8 Faster Delivery of Systems 41
3.8.1 Smaller and Simpler Problems 42
3.8.2 Less Bureaucratic Inertia 42
CHAPTER FOUR - COMPONENTS OF CLIENT/SERVER COMPUTING
AND ITS APPLICATIONS-THE CLIENT
4.1 General 44
4.2 The Role of the Client 45
4.3 Client Services 46

4.4 Requests for Service 47
4.4.1 Remote Procedure Call (RPC) 48
4.4.2 Fax/Print Services 48
4.4.3 Window Services 49
4.4.4 Remote Boot Services 49
4.4.5 Utility Services 50
4.4.6 Message Services 50
4.4.7 Network Services 50
4.4.8 Application Services 50
4.4.9 Database Services 50
4.4.10 Network Management Services-Alerts 51
4.4.11 Dynamic Data Exchange (DDE) 51
4.4.12 Object Linking and Embedding (OLE) 51
4.4.13 Common Object Request Broker Architecture (CORBA) 52
4.5 Enterprise View 52
CHAPTER FIVE - COMPONENTS OF CLIENT/SERVER COMPUTING AND ITS APPLICATIONS-THE SERVER
5.1 General 53
5.2 The Role of the Server 55
5.3 Server Functionality in Detail 56
5.3.1 Request Processing 56
5.3.2 File Services 57
5.3.3 Fax/Print/Image Services 57
5.3.4 Database Services 58
5.3.5 Communications Services 60
5.3.6 Security Services 60
5.4 The Network Operating System 60
5.4.1 Novell NetWare 60
5.4.2 LAN Manager 62
5.4.3 IBM LAN Server 62
5.4.4 Banyan VINES 63
5.4.5 PC Network File Services (NFS) 63
5.5 The Server Operating System 64
5.5.1 NetWare 64
5.5.2 OS/2 64
5.5.3 Windows NT 64
5.5.4 MVS 65

5.5.5 OPENVMS 65
5.5.6 UNIX 66
5.6 System Application Architecture (SAA) 67
CONCLUSION 68
REFERANCE 69
LIST OF ABBREVIATIONS 70

ABSTRACT

Client/server is a computational architecture that involves client processes requesting service from server processes. Client/server computing is the logical extension of modular programming.

Client/server computing provides the capability to use the most cost-effective user interface, data storage, connectivity, and application services. Frequently, client/server products are deployed within the present organization but are not used effectively.

The client in the client/server model is the desktop workstation. Any workstation that is used by a single user is a client. The same workstation, when shared simultaneously by multiple users, is a server.

The server is a multiuser computer. There is no special hardware requirement that turns a computer into a server. The hardware platform should be selected based on application demands and economics.

INTRODUCTION

In the first chapter we talk about the introduction of TCP/IP and what protocols it consists of, the OSI Model we also talk on the history of TCP/IP. IP Addresses are also covered in this chapter. Using these concepts, we then move on to Client/Server Computing.

Here in the second chapter we talk on the evolution of Client/Server computing, what are the configurations needed in a Client/Server computing. We also discuss on the Client/Server Application Models. Finally we speak on the features of Client/Server computing.

Chapter three moves us to the advantages of Client/Server computing, how to enchance data sharing and what integrated services are available on the Client/Server computing and on how to improve the network preformance. It also tells us on how to reduce network traffic.

In chapter four we talk about the components of the client, the role of a client computer, what are the services in a client computer, how a client computer requests a service from a server.

In the last chapter which is chapter five, we talk about the components of a server, what the role of a server is, the types of services it provides such as file storeage, print/fax/imaging, database, communication and security services. It also talks about the Network Operating System. We also see what types of Operating Systems can be used in an server.

CHAPTER ONE

INTRODUCTION TO TCP/IP

Just what is TCP/IP? It is a software-based communications protocol used in networking. Although the name TCP/IP implies that the entire scope of the product is a combination of two protocols—Transmission Control Protocol and Internet Protocol—the term TCP/IP refers not to a single entity combining two protocols, but a larger set of software programs that provides network services such as remote logins, remote file transfers, and electronic mail. TCP/IP provides a method for transferring information from one machine to another. A communications protocol should handle errors in transmission, manage the routing and delivery of data, and control the actual transmission by the use of predetermined status signals. TCP/IP accomplishes all of this.

OSI Reference Model is composed of seven layers. TCP/IP was designed with layers as well, although they do not correspond one-to-one with the OSI-RM layers. You can overlay the TCP/IP programs on this model to give you a rough idea of where all the TCP/IP layers reside. Figure 1.1 shows the basic elements of the TCP/IP family of protocols. We can see that TCP/IP is not involved in the bottom two layers of the OSI model (data link and physical) but begins in the network layer, where the Internet Protocol (IP) resides. In the transport layer, the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are involved. Above this, the utilities and protocols that make up the rest of the TCP/IP suite are built using the TCP or UDP and IP layers for their communications system.

Figure 1.1 shows that some of the upper-layer protocols depend on TCP (such as Telnet and FTP), whereas some depend on UDP (such as TFTP and RPC). Most upper-layer TCP/IP protocols use only one of the two transport protocols (TCP or UDP), although a few, including DNS (Domain Name System) can use both. A note of caution about TCP/IP: Despite the fact that TCP/IP is an open protocol, many companies have modified it for their own networking system. There can be incompatibilities because of these modifications, which, even though they might adhere to the official standards, might have other aspects that cause problems. Luckily, these types of changes are not rampant, but you should be careful when choosing a TCP/IP product to ensure its compatibility with existing software and hardware.

Telnet - RemoteLogin
FTP - File Transfer Protocol
SMTP - Simple Mail Transfer Protocol
X - X Windows System
Kerberos - Security
DNS - Domain Name System
ASN - Abstract Syntax Notation
SNMP - Simple Network Management Protocol

NFS - Network File Server
RPC - Remote Procedure Calls
TFTP - Trivial File Transfer Protocol
TCP - Transmission Control Protocol
User Datagram Protocol
IP - Internet Protocol
ICMP - Internet Control Message Protocol

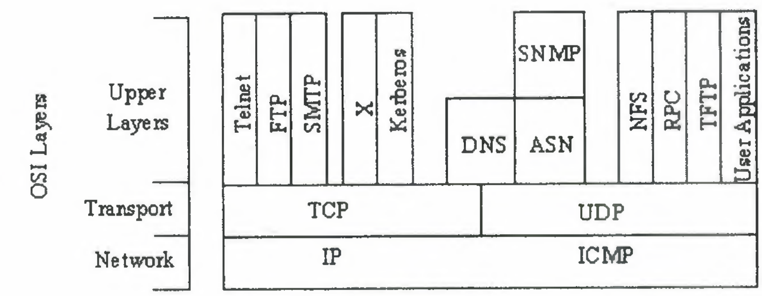

Figure 1.1. TCP/IP suite and OSI layers.

TCP/IP is dependent on the concept of clients and servers. This has nothing to do with a file server being accessed by a diskless workstation or PC. The term *client/server* has a simple meaning in TCP/IP: any device that initiates communications is the client, and the device that answers is the server. The server is responding to (serving) the client's requests.

1.1 An Overview of TCP/IP Components

To understand the roles of the many components of the TCP/IP protocol family, it is useful to know what you can do over a TCP/IP network. Then, once the applications are understood, the protocols that make it possible are a little easier to comprehend.

The following list is not exhaustive but mentions the primary user applications that TCP/IP provides.

1.1.1 Telnet

The Telnet program provides a remote login capability. This lets a user on one machine log onto another machine and act as though he or she were directly in front of the second machine. The connection can be anywhere on the local network or on another network anywhere in the world, as long as the user has permission to log onto the remote system.

We can use Telnet when we need to perform actions on a machine across the country. This isn't often done except in a LAN or WAN context, but a few systems accessible through the Internet allow Telnet sessions while users play around with a new application or operating system.

1.1.2 File Transfer Protocol

File Transfer Protocol (FTP) enables a file on one system to be copied to another system. The user doesn't actually log in as a full user to the machine he or she wants to access, as with Telnet, but instead uses the FTP program to enable access. Again, the correct permissions are necessary to provide access to the files.

Once the connection to a remote machine has been established, FTP enables us to copy one or more files to your machine. (The term *transfer* implies that the file is moved from one system to another but the original is not affected. Files are copied.) FTP is a widely used service on the Internet, as well as on many large LANs and WANs.

1.1.3 Simple Mail Transfer Protocol

Simple Mail Transfer Protocol (SMTP) is used for transferring electronic mail. SMTP is completely transparent to the user. Behind the scenes, SMTP connects to remote machines and transfers mail messages much like FTP transfers files. Users are almost never aware of SMTP working, and few system administrators have to bother with it. SMTP is a mostly trouble-free protocol and is in very wide use.

1.1.4 Kerberos

Kerberos is a widely supported security protocol. Kerberos uses a special application called an *authentication server* to validate passwords and encryption schemes. Kerberos is one of the more secure encryption systems used in communications and is quite common in UNIX.

1.1.5 Domain Name System

Domain Name System (DNS) enables a computer with a common name to be converted to a special network address. For example, a PC called Darkstar cannot be accessed by another machine on the same network (or any other connected network) unless some method of checking the local machine name and replacing the name with the machine's hardware address is available. DNS provides a conversion from the common local name to the unique physical address of the device's network connection.

1.1.6 Simple Network Management Protocol

Simple Network Management Protocol (SNMP) provides status messages and problem reports across a network to an administrator. SNMP uses User Datagram Protocol (UDP) as a transport mechanism. SNMP employs slightly different terms from TCP/IP, working with managers and agents instead of clients and servers (although they mean essentially the same thing). An agent provides information about a device, whereas a manager communicates across a network with agents.

1.1.7 Network File System

Network File System (NFS) is a set of protocols developed by Sun Microsystems to enable multiple machines to access each other's directories transparently. They accomplish this by using a distributed file system scheme. NFS systems are common in large corporate environments, especially those that use UNIX workstations.

1.1.8 Remote Procedure Call

The Remote Procedure Call (RPC) protocol is a set of functions that enable an application to communicate with another machine (the server). It provides for programming functions, return codes, and predefined variables to support distributed computing.

1.1.9 Trivial File Transfer Protocol

Trivial File Transfer Protocol (TFTP) is a very simple, unsophisticated file transfer protocol that lacks security. It uses UDP as a transport. TFTP performs the same task as FTP, but uses a different transport protocol.

1.1.10 Transmission Control Protocol

Transmission Control Protocol (the TCP part of TCP/IP) is a communications protocol that provides reliable transfer of data. It is responsible for assembling data passed from higher-layer applications into standard packets and ensuring that the data is transferred correctly.

1.1.11 User Datagram Protocol

User Datagram Protocol (UDP) is a connectionless-oriented protocol, meaning that it does not provide for the retransmission of datagrams (unlike TCP, which is connection-oriented). UDP is not very reliable, but it does have specialized purposes. If the applications that use UDP have reliability checking built into them, the shortcomings of UDP are overcome.

1.1.12 Internet Protocol

Internet Protocol (IP) is responsible for moving the packets of data assembled by either TCP or UDP across networks. It uses a set of unique addresses for every device on the network to determine routing and destinations.

1.1.13 Internet Control Message Protocol

Internet Control Message Protocol (ICMP) is responsible for checking and generating messages on the status of devices on a network. It can be used to inform other devices of a failure in one particular machine. ICMP and IP usually work together.

1.2 TCP/IP History

The architecture of TCP/IP is often called the Internet architecture because TCP/IP and the Internet as so closely interwoven. We have seen how the Internet standards were developed by the Defense Advanced Research Projects Agency (DARPA) and eventually passed on to the Internet Society.

The Internet was originally proposed by the precursor of DARPA, called the Advanced Research Projects Agency (ARPA), as a method of testing the viability of packet-switching networks. (When ARPA's focus became military in nature, the name was changed.) During its tenure with the project, ARPA foresaw a network of leased lines connected by switching nodes. The network was called ARPANET, and the switching nodes were called Internet Message Processors, or IMPs. The ARPANET was initially to be comprised of four IMPs located at the University of California at Los Angeles, the University of California at Santa Barbara, the Stanford Research Institute, and the University of Utah. The original IMPs were to be Honeywell 316 minicomputers.

The contract for the installation of the network was won by Bolt, Beranek, and Newman (BBN), a company that had a strong influence on the development of the network in the following years. The contract was awarded in late 1968, followed by testing and refinement over the next five years. In 1971, ARPANET entered into regular service. Machines used the ARPANET by connecting to an IMP using the "1822" protocol—so called because that was the number of the technical paper describing the system. During the early years, the purpose and utility of the network was widely (and sometimes heatedly) discussed, leading to refinements and modifications as users requested more functionality from the system.

A commonly recognized need was the capability to transfer files from one machine to another, as well as the capability to support remote logins. Remote logins would enable a user in Santa Barbara to connect to a machine in Los Angeles over the network and function as though he or she were in front of the UCLA machine. The protocol then in use on the network wasn't capable of handling these new functionality requests, so new protocols were continually developed, refined, and tested.

Remote login and remote file transfer were finally implemented in a protocol called the Network Control Program (NCP). Later, electronic mail was added through File Transfer

Protocol (FTP). Together with NCP's remote logins and file transfer, this formed the basic services for ARPANET. By 1973, it was clear that NCP was unable to handle the volume of traffic and proposed new functionality. A project was begun to develop a new protocol. The TCP/IP and gateway architectures were first proposed in 1974. The published article by Cerf and Kahn described a system that provided a standardized application protocol that also used end-to-end acknowledgments.

Neither of these concepts were really novel at the time, but more importantly (and with considerable vision), Cerf and Kahn suggested that the new protocol be independent of the underlying network and computer hardware. Also, they proposed universal connectivity throughout the network. These two ideas were radical in a world of proprietary hardware and software, because they would enable any kind of platform to participate in the network. The protocol was developed and became known as TCP/IP.

A series of RFCs (Requests for Comment, part of the process for adopting new Internet Standards) was issued in 1981, standardizing TCP/IP version 4 for the ARPANET. In 1982, TCP/IP supplanted NCP as the dominant protocol of the growing network, which was now connecting machines across the continent. It is estimated that a new computer was connected to ARPANET every 20 days during its first decade. (That might not seem like much compared to the current estimate of the Internet's size doubling every year, but in the early 1980s it was a phenomenal growth rate.)

During the development of ARPANET, it became obvious that nonmilitary researchers could use the network to their advantage, enabling faster communication of ideas as well as faster physical data transfer. A proposal to the National Science Foundation lead to funding for the Computer Science Network in 1981, joining the military with educational and research institutes to refine the network. This led to the splitting of the network into two different networks in 1984. MILNET was dedicated to unclassified military traffic, whereas ARPANET was left for research and other nonmilitary purposes. ARPANET's growth and subsequent demise came with the approval for the Office of Advanced Scientific Computing to develop wide access to supercomputers. They created NSFNET to connect six supercomputers spread across the country through T-1 lines (which operated at 1.544 Mbps). The Department of Defense finally declared ARPANET obsolete in 1990, when it was officially dismantled.

1.3 Berkeley UNIX Implementations and TCP/IP

TCP/IP became important when the Department of Defense started including the protocols as military standards, which were required for many contracts. TCP/IP became popular primarily because of the work done at UCB (Berkeley). UCB had been a center of UNIX development for years, but in 1983 they released a new version that incorporated TCP/IP as an integral element. That version—4.2BSD (Berkeley System Distribution)—was made available to the world as public domain software.

The popularity of 4.2BSD spurred the popularity of TCP/IP, especially as more sites connected to the growing ARPANET. Berkeley released an enhanced version (which included the so-called Berkeley Utilities) in 1986 as 4.3BSD. An optimized TCP implementation followed in 1988 (4.3BSD/Tahoe). Practically every version of TCP/IP available today has its roots (and much of its code) in the Berkeley versions.

1.4 OSI and TCP/IP

The adoption of TCP/IP didn't conflict with the OSI standards because the two developed concurrently. In some ways, TCP/IP contributed to OSI, and vice-versa. Several important differences do exist, though, which arise from the basic requirements of TCP/IP which are:

- A common set of applications
- Dynamic routing
- Connectionless protocols at the networking level
- Universal connectivity
- Packet-switching

The differences between the OSI architecture and that of TCP/IP relate to the layers above the transport level and those at the network level. OSI has both the session layer and the presentation layer, whereas TCP/IP combines both into an application layer. The requirement for a connectionless protocol also required TCP/IP to combine OSI's physical layer and data link layer into a network level. TCP/IP also includes the session and presentation layers of the OSI model into TCP/IP's application layer. A schematic view of TCP/IP's layered structure compared with OSI's seven-layer model is shown in Figure 1.2. TCP/IP calls the different network level elements *subnetworks*.

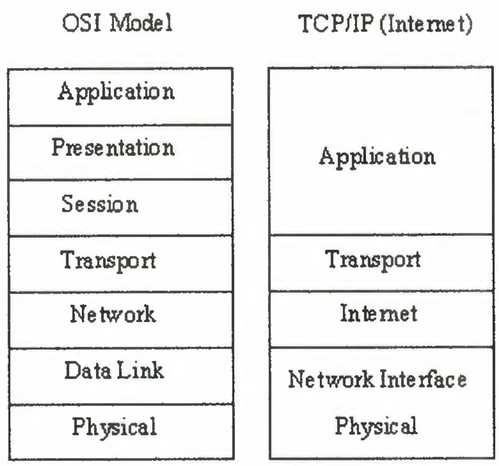

Figure 1.2. The OSI and TCP/IP layered structures.

Some fuss was made about the network level combination, although it soon became obvious that the argument was academic, as most implementations of the OSI model combined the physical and link levels on an intelligent controller (such as a network card). The combination of the two layers into a single layer had one major benefit: it enabled a subnetwork to be designed that was independent of any network protocols, because TCP/IP was oblivious to the details. This enabled proprietary, self-contained networks to implement the TCP/IP protocols for connectivity outside their closed systems.

The layered approach gave rise to the name TCP/IP. The transport layer uses the Transmission Control Protocol (TCP) or one of several variants, such as the User Datagram Protocol (UDP). (There are other protocols in use, but TCP and UDP are the most common.) There is, however, only one protocol for the network level—the Internet Protocol (IP). This is what assures the system of universal connectivity, one of the primary design goals.

There is a considerable amount of pressure from the user community to abandon the OSI model (and any future communications protocols developed that conform to it) in favor of TCP/IP. The argument hinges on some obvious reasons:

- TCP/IP is up and running and has a proven record.
- TCP/IP has an established, functioning management body.
- Thousands of applications currently use TCP/IP and its well-documented application programming interfaces.
- TCP/IP is the basis for most UNIX systems, which are gaining the largest share of the operating system market (other than desktop single-user machines such as the PC and Macintosh).
- TCP/IP is vendor-independent.

Arguing rather strenuously against TCP/IP, surprisingly enough, is the US government—the very body that sponsored it in the first place. Their primary argument is that TCP/IP is not an internationally adopted standard, whereas OSI has that recognition. The Department of Defense has even begun to move its systems away from the TCP/IP protocol set. A compromise will probably result, with some aspects of OSI adopted into the still-evolving TCP/IP protocol suite.

1.5 TCP/IP and Ethernet

For many people the terms TCP/IP and Ethernet go together almost automatically, primarily for historical reasons, as well as the simple fact that there are more Ethernet-based TCP/IP networks than any other type. Ethernet was originally developed at Xerox's Palo Alto Research Center as a step toward an electronic office communications system, and it has since grown in capability and popularity.

Ethernet is a hardware system providing for the data link and physical layers of the OSI model. As part of the Ethernet standards, issues such as cable type and broadcast speeds are established. There are several different versions of Ethernet, each with a different data transfer rate. The most common is Ethernet version 2, also called 10Base5, Thick Ethernet, and IEEE 802.3 (after the number of the standard that defines the system adopted by the Institute of Electrical and Electronic Engineers). This system has a 10 Mbps rate.

There are several commonly used variants of Ethernet, such as Thin Ethernet (called 10Base2), which can operate over thinner cable (such as the coaxial cable used in cable television systems), and Twisted-Pair Ethernet

(10BaseT), which uses simple twisted-pair wires similar to telephone cable. The latter variant is popular for small companies because it is inexpensive, easy to wire, and has no strict requirements for distance between machines.

Ethernet and TCP/IP work well together, with Ethernet providing the physical cabling (layers one and two) and TCP/IP the communications protocol (layers three and four) that is broadcast over the cable. The two have their own processes for packaging information: TCP/IP uses 32-bit addresses, whereas Ethernet uses a 48-bit scheme. The two work together, however, because of one component of TCP/IP called the Address Resolution Protocol (ARP), which converts between the two schemes. (I discuss ARP in more detail later, in the section titled "Address Resolution Protocol.")

Ethernet relies on a protocol called Carrier Sense Multiple Access with Collision Detect (CSMA/CD). To simplify the process, a device checks the network cable to see if anything is currently being sent. If it is clear, the device sends its data. If the cable is busy (carrier detect), the device waits for it to clear. If two devices transmit at the same time (a collision), the devices know because of their constant comparison of the cable traffic to the data in the sending buffer. If a collision occurs, the devices wait a random amount of time before trying again.

1.6 IP Addresses

TCP/IP uses a 32-bit address to identify a machine on a network and the network to which it is attached. IP addresses identify a machine's connection to the network, not the machine itself—an important distinction. Whenever a machine's location on the network changes, the IP address must be changed, too. The IP address is the set of numbers many people see on their workstations or terminals, such as 127.40.8.72, which uniquely identifies the device. IP (or Internet) addresses are assigned only by the Network Information Center (NIC), although if a network is not connected to the Internet, that network can determine its own numbering. For all Internet accesses, the IP address must be registered with the NIC.

There are four formats for the IP address, with each used depending on the size of the network. The four formats, called Class A through Class D, are shown in Figure 1.3. The class is identified by the first few bit sequences, shown in the figure as one bit for Class A and up to four bits for Class D. The class can be determined from the first three (high-order) bits. In fact, in most cases, the first two bits are enough, because there are few Class D networks.

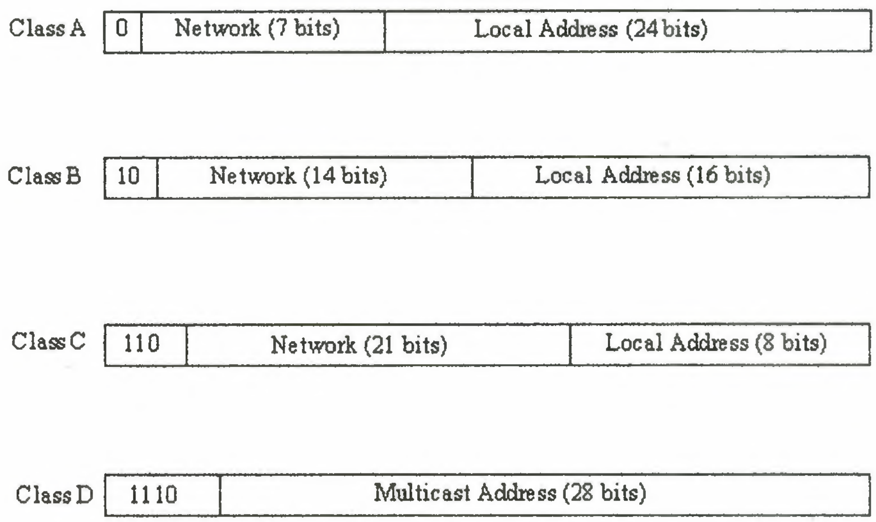

Figure 1.3. The four IP address class structures.

Class A addresses are for large networks that have many machines. The 24 bits for the local address (also frequently called the host address) are needed in these cases. The network address is kept to 7 bits, which limits the number of networks that can be identified. Class B addresses are for intermediate networks, with 16-bit local or host addresses and 14-bit network addresses. Class C networks have only 8 bits for the local or host address, limiting the number of devices to 256. There are 21 bits for the network address. Finally, Class D networks are used for multicasting purposes, when a general broadcast to more than one device is required. The lengths of each section of the IP address have been carefully chosen to provide maximum flexibility in assigning both network and local addresses.

IP addresses are four sets of 8 bits, for a total 32 bits. You often represent these bits as separated by a period for convenience, so the IP address format can be thought of as network.local.local.local for Class A or network.network.network.local for Class C. The IP addresses are usually written out in their decimal equivalents, instead of the long binary strings.

This is the familiar host address number that network users are used to seeing, such as 147.10.13.28, which would indicate that the network address is 147.10 and the local or host address is 13.28. Of course, the actual address is a set of 1s and 0s. The decimal notation used for IP addresses is properly called *dotted quad notation*—a bit of trivia for your next dinner party.

The IP addresses can be translated to common names and letters. This can pose a problem, though, because there must be some method of unambiguously relating the physical address, the network address, and a language-based name (such a tpci_ws_4 or bobs_machine). "The Domain Name System" looks at this aspect of address naming. From the IP address, a network can determine if the data is to be sent out through a gateway. If the network address is the same as the current address (routing to a local network device, called a *direct host*), the gateway is avoided, but all other network addresses are routed to a gateway to leave the local network (*indirect host*). The gateway receiving data to be transmitted to another network must then determine the routing from the data's IP address and an internal table that provides routing information.

As mentioned, if an address is set to all 1s, the address applies to all addresses on the network. (See the previous section titled "Physical Addresses.") The same rule applies to IP addresses, so that an IP address of 32 1s is considered a broadcast message to all networks and all devices. It is possible to broadcast to all machines in a network by altering the local or host address to all 1s, so that the address 147.10.255.255 for a Class B network (identified as network 147.10) would be received by all devices on that network (255.255 being the local addresses composed of all 1s), but the data would not leave the network. There are two contradictory ways to indicate broadcasts. The later versions of TCP/IP use 1s, but earlier BSD systems use 0s. This causes a lot of confusion. All the devices on a network must know which broadcast convention is used; otherwise, datagrams can be stuck on the network forever!

A slight twist is coding the network address as all 0s, which means the originating network or the local address being set to 0s, which refers to the originating device only (usually used only when a device is trying to determine its IP address). The all-zero network address format is used when the network IP address is not known but other devices on the network can still interpret the local address. If this were transmitted to another network, it could cause confusion! By convention, no local device is given a physical address of 0.

It is possible for a device to have more than one IP address if it is connected to more than one network, as is the case with gateways. These devices are called *multihomed,* because they have a unique address for each network they are connected to. In practice, it is best to have a dedicate machine for a multihomed gateway; otherwise, the applications on that machine can get confused as to which address they should use when building datagrams. Two networks can have the same network address if they are connected by a gateway. This can cause problems for addressing, because the gateway must be able to differentiate which network the physical address is on. This problem is looked at again in the next section, showing how it can be solved.

CHAPTER TWO

OVERVIEW OF CLIENT/SERVER COMPUTING

2.1 General

Client/server is a computational architecture that involves client processes requesting service from server processes. Client/server computing is the logical extension of modular programming. Modular programming has as its fundamental assumption that separation of a large piece of software into its constituent parts ("modules") creates the possibility for easier development and better maintainability.

Client/server computing takes this a step farther by recognizing that those modules need not all be executed within the same memory space. With this architecture, the calling module becomes the "client" (that which requests a service), and the called module becomes the "server" (that which provides the service). The logical extension of this is to have clients and servers running on the appropriate hardware and software platforms for their functions. For example, database management system servers running on platforms specially designed and configured to perform queries, or file servers running on platforms with special elements for managing files. It is this latter perspective that has created the widely-believed myth that client/server has something to do with PCs or Unix machines.

2.2 Evolution of Client-Server Computing

The evolution of Client-Server Computing has been driven by business needs, as well as the increasing costs for host (mainframe and midrange) machines and maintenance, the decreasing costs and increasing power of micro-computers and the increased reliability of LANs (Local Area Networks).

In the past twenty years, there are dramatic improvements in the hardware and software technologies for micro-computers. Micro-computers become affordable for small businesses and organizations. And at the same time their performances are becoming more and more reliable. On the other hand, the drop in price for mainframe is growing at a slower rate than the drop in its price. Little developments have achieved with mainframes.

The following are the improvements made by micro-computers:

- **Hardware:** The speed of desktop microprocessors has grown exponentially, from an 8MHz 386-based computers to 100Hz-based Pentium-based microprocessors. These mass-produced microprocessors are cheaper and more powerful than those used in mainframe and midrange computers. On the other hand, the capacity of main memory in micro-computers has been quadrupling every three years. Typically main memory size is 16 Megabytes nowadays. Besides, the amount of backup storage and memory such as hard disks and CD-ROMs that are able to support micro-computers has also puts an almost unlimited amount of data in reach for end-users.
- **Software:** The development and acceptance of GUIs (Graphical User Interfaces) such as Windows 3.1 and OS/2 has made the PC working environment more user-friendly. And the users are more efficient in learning new application softwares in a graphical environment. Besides GUIs, the use of multithreaded processing and relational databases has also contributed to the popularity of Client-Server Computing.

2.3 Configurations in Client-Server Computing

Client-Server Computing is divided into three components, a Client Process requesting service and a Server Process providing the requested service, with a Middleware in between them for their interaction.

2.3.1 Client

A Client Machine usually manage the user-interface portion of the application, validate data entered by the user, dispatch requests to server programs. It is the front-end of the application that the user sees and interacts with. Besides, the Client Process also manages the local resources that the user interacts with such as the monitor, keyboard, workstation, CPU and other peripherals.

The client is a process (program) that sends a message to a server process (program), requesting that the server perform a task (service). Client programs usually manage the user-interface portion of the application, validate data entered by the user, dispatch requests to server programs, and sometimes execute business logic. The client-based process is the front- end of the application that the user sees and interacts with.

The client process contains solution-specific logic and provides the interface between the user and the rest of the application system. The client process also manages the local resources that the user interacts with such as the monitor, keyboard, workstation CPU and peripherals. One of the key elements of a client workstation is the graphical user interface (GUI). Normally a part of operating system i.e. the window manager detects user actions, manages the windows on the display and displays the data in the windows.

2.3.2 Server

On the other hand, the Server Machine fulfills the client request by performing the service requested. After the server receives requests from clients, it executes database retrieval , updates and manages data integrity and dispatches responses to client requests. The server-based process may run on another machine on the network; the server is then provided both file system services and application services.Or in some cases, another desktop machine provides the application services. The server acts as software engine that manages shared resources such as databases, printers, communication links, or high powered-processors. The main aim of the Server Process is to perform the back-end tasks that are common to similar applications.

The simplest form of servers are disk servers and file servers. With a file server, the client passes requests for files or file records over a network to the file server. This form of data service requires large bandwidth and can slow a network with many users. The more advanced form of servers are Database servers, Transaction server and Application servers.

The following diagram illustrates the relationship between client and server computers. The client requests information; the server processes the request and sends a response back to the client.

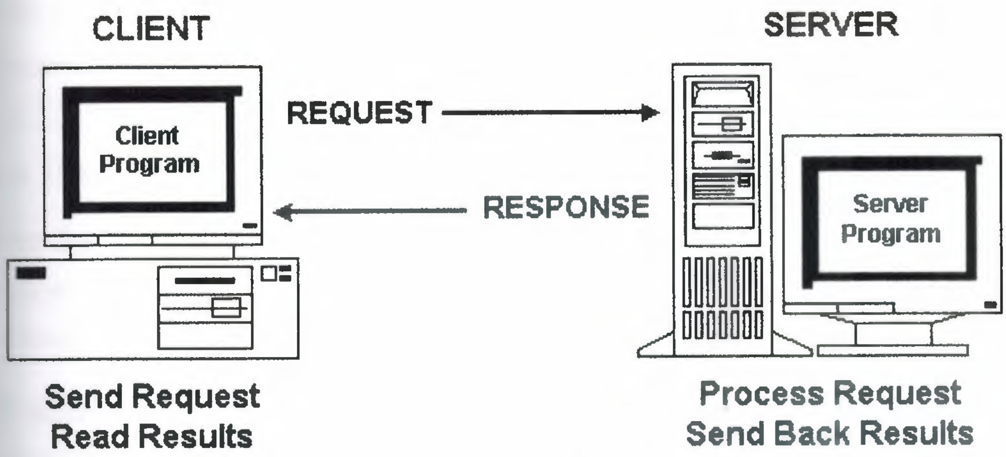

Figure 2.1. The relationship between client and server computers

2.3.3 Middleware

Middleware allows applications to transparently commnicate with other programs or processes regardless of location. The key element of Middleware is NOS (Network Operating System) that provides services such as routing, distribution, messaging and network management service. NOS rely on communiction protocols to provide specific services. Once the physical connection has been established and transport protocols chosen, a client-server protocol is required before the user can access the network services. A client-server protocol dictates the manner in which clients request information and services from a server and also how the server replies to that request.

The protocols are divided into three groups: media, transport and client-server protocols. Media protocols determine the type of physical connections used on a network (some examples of media protocols are Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), coaxial and twisted-pair). A transport protocol provides the mechanism to move packets of data from client to server (some examples of transport protocols are Novell's IPX/SPX, Apple's AppleTalk, Transmission Control Protocol/ Internet Protocol (TCP/IP), Open Systems Interconnection (OSI) and Government Open Systems Interconnection Profile(GOSIP)). Once the physical connection has been established and transport protocols chosen, a client-server protocol is required before the user can access the network services. A client-server protocol dictates the manner in which clients request information and services from a server and also how the server replies to that request (some examples of client- server protocols are NetBIOS, RPC, Advanced Program-to-

Program Communication (APPC), Named Pipes, Sockets, Transport Level Interface (TLI) and Sequenced Packet Exchange (SPX)).

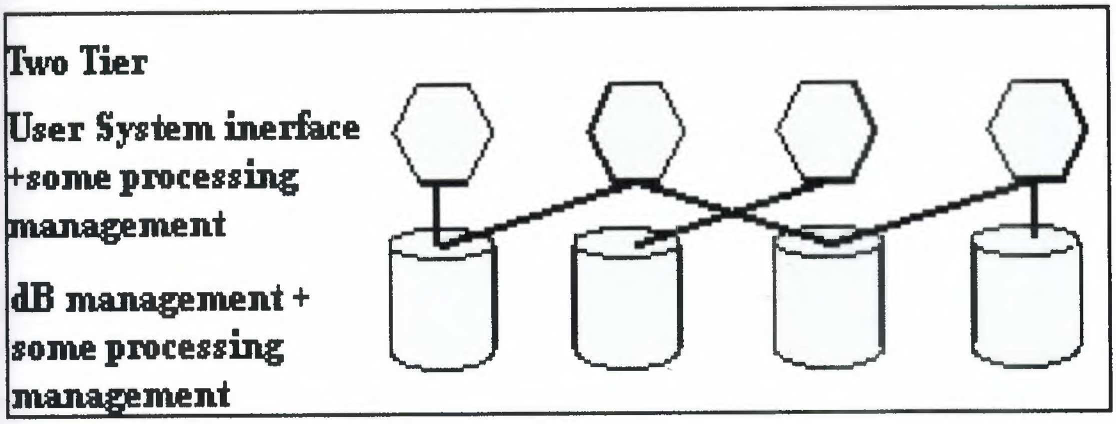

Figure 2.2. Two Tier Client Server Architecture Design

2.3.4 Two-Tier Architecture

A two-tier architecture is where a client talks directly to a server, with no intervening server(figure 2.2). It is typically used in small environments (less than 50 users). A common error in client/server development is to prototype an application in a small, two-tier environment, and then scale up by simply adding more users to the server. This approach will usually result in an ineffective system, as the server becomes overwhelmed. To properly scale to hundreds or thousands of users, it is usually necessary to move to a three-tier architecture.

2.3.5 Three-Tier Architecture

A three-tier architecture introduces a server (or an "agent") between the client and the server. The role of the agent is manyfold. It can provide translation services (as in adapting a legacy application on a mainframe to a client/server environment), metering services (as in acting as a transaction monitor to limit the number of simultaneous requests to a given server), or intellegent agent services (as in mapping a request to a number of different servers, collating the results, and returning a single response to the client.

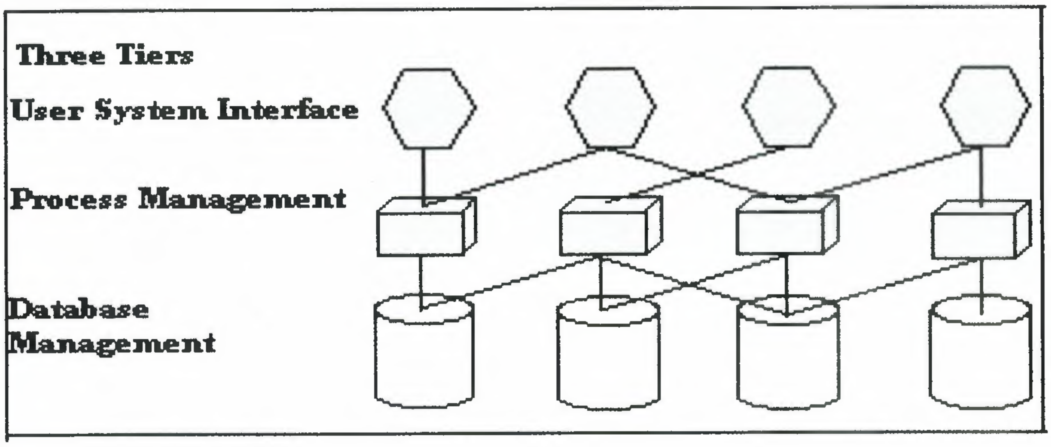

Figure 2.3. Three tier distributed client/server architecture depiction

2.3.6 The Cooperative Processing

Cooperative processing is computing which requires two or more distinct processors to complete a single transaction. Cooperative processing is related to both distributed and client/server processing. It is a form of distributed computing where two or more distinct processes are required to complete a single business transaction. Usually, these programs interact and execute concurrently on different processors. Cooperative processing can also be considered to be a style of client/server processing if communication between processors is performed through a message passing architecture.

2.3.7 The Distributed Processing

The distribution of applications and business logic across multiple processing platforms. Distributed processing implies that processing will occur on more than one processor in order for a transaction to be completed. In other words, processing is distributed across two or more machines and the processes are most likely not running at the same time, i.e. each process performs part of an application in a sequence. Often the data used in a distributed processing environment is also distributed across platforms.

2.3.8 The Intranet

The explosion of the World Wide Web is due to the world-wide acceptance of a common transport (TCP/IP), server standard (HTTP), and markup language (HTML). Many corporations have discovered that these same technologies can be used for internal client/server applications with the same ease that they are used on the Internet.

Thus was born the concept of the "Intranet": the use of Internet technologies for implementing internal client/server applications. One key advantage of Web-based intranets is that the problem of managing code on the client is greatly reduced. Assuming a standard browser on the desktop, all changes to user interface and functionality can be done by changing code on the HTTP server. Compare this with the cost of updating client code on 2,000 desktops. A second advantage is that if the corporation is already using the Internet, no additional code needs to be licensed or installed on client desktops. To the user, the internal and external information servers appear integrated. A rapidly-disappearing disadvantage is that there is limited ability to provide custom coding on the client. In the early days of the Web, there were limited ways of interacting with the client. The Web was essentially "read-only". With the release of code tools such as Java and JavaScript, this limitation is no longer a major issue.

2.4 The Four Dominant Client/Server Application Models

Having had a deeper look into the terms and architectures of client/server technology, let's consider the dominant application models available. Nowadays, there are four client/server application models that are widely used in the market. They are Structured Query Language (SQL) databases, Transaction Processing (TP) monitors, groupware and distributed objects. Each one of them is capabable of creating its own complete client/server applications with its own tools. Moreover, they also introduce their own favorable form of middleware (all this will be further discussed later). But first, what is the reason for having different models instead of having just one model, and what are the advantages/disadvantages of having just one particular model.

The reason why we need different models for different applications is because each one of them has their own advantages and disadvantages, and sometimes one model performs better than the others in one particular situation. Furthermore, standardizing the whole market with one particular model will not only discourage the vendors from developing other new (and better) models, but also put off other potential small companies from competing with those gigantic ones. Having said that, standardizing the market with one particular model does have the advantage of concentrating the development of that particular model-based softwares, and hence improvements can be achieved much faster and as a result, cost of running/implementing/services will reduce significantly.

Having realized the needs for different application models, the following will be dedicated to those models and at the end of this section, a comparison of the four models will be made and see which one of them will best suit our future needs in general.

2.4.1 SQL databases

SQL (Structured Query Language) has been the standard data description and access language for relational databases for almost a decade, making it the core technology for client/server computing and dominating the client/server landscape today. It began as a declarative language for manipulating data using a few simple commands; however, as SQL applications moved to more demanding client/server environments, it became clear that just managing data wasn't enough. There was also a need to manage the functions that manipulated the data. And stored procedures, sometimes called "TP-lite", met the needs.

A stored procedure is a named collection of SQL statements and procedural logic that is complied, verified and stored in a server database. ***Sybase*** pioneered the concept of stored procedures and now virtually all SQL vendors support stored procedure along with other SQL extensions. The extensions are used to enforce data integrity, perform system maintenance and implement the server side of an application's logic.

The problem with the SQL standard is that there are just too many of them. There are at least eight efforts underway trying to create a standard based on SQL. *(ANSI alone is responsible for three standards, either published or in progress).* And since SQL standards seem to lag vendor implementations by some years, almost everything that's interesting in client/server database technology is non-standard. This includes database administration, data replication, stored procedures, user-defined data types and the formats and protocols on networks. As a result, the lack of one widely accepted standard drives up the cost of databases and related tools and makes maintaining a client/server environment complex and difficult.

Although SQL suffers all this shortcomings, the fact that there are still so many people using it is because it is easy to create client/server applications in single-vendor/single-server environments. Many GUI tools make SQL applications easy to build, and most of all, it is familiar to the majority of programmers and users out there.

2.4.2 Transaction Processing (TP) monitors

In a simple client/server system, many clients issue requests and a server responses. This system may work for 50 or 100 clients, however, as the number of clients increases, the number of requests increases as well and eventually, it reaches the threshold limit of the system, i.e. the system crashes. Sadly, this is the case for most operating systems. Another technology, the TP monitors, is then developed to solve this problem.

Originally, TP monitor meant teleprocessing monitor - a program that multiplexed many terminals (clients) to a single central server. Over time, TP monitors took on more than just multiplexing and routing functions, and TP monitors came to mean transaction processing. TP monitors manage processes by breaking complex applications into pieces of code called *transactions*. A transaction can also be viewed as a set of actions that obeys the four so-called ***ACID*** properties - ***A***tomic, ***C***onsistent, ***I***solated, and ***D***urable. As for the middleware, TP monitors use some form of transactional RPC or peer-to-peer middleware.

TP monitors are probably overkilled in single-server/single-vendor departmental applications. That is probably one of the reasons why they have been so slow to take off. Moreover, vendors haven't come to grips yet with the realities of shrink wrapped software market, and they haven't been able to explain the advantages TP monitors offer. The modern client/server incarnations of TP monitors haven't dominant the Ethernet era, but they'll definitely be playing a major role in the intergalactic era. It won't be long to assume that every machine on the network will have a TP monitors to represent it in global transactions. The intergalactic era will make those TP monitors advantages increasingly self-evident.

2.4.3 Groupware

Groupware comprises five foundation technologies geared to support collaborative work: multimedia management, work flow, E-mail, conferencing and scheduling.

Groupware isn't just another downsized mainframe technology; it is indeed a new model of client/server computing. It helps users to collect unstructured data (e.g. text, images, and faxes) into a set of documents.

Groupware has the advantage of document database management and makes effective use of E-mail, which is its preferred form of middleware. E-mail is one of the easiest ways for electronic processes to communicate with humans. Asynchronous by nature, it's a good match for the way business really works. E-mail is ubiquitous, with over 50 million globally interconnected electronic mailboxes.

Using work-flow to manage business processes is another revolutionary aspect of groupware. In a work-flow, data passes from one program to another in structured or unstructured client/server environments. Modern work-flow software electronically simulates real world collaborative activity. Work can be routed in ways that correspond to interoffice communications. A good work-flow packages lets you specify acceptance criteria for moving work from one stage to another. So work-flow brings the information to people who can act on it such that the work gets done by the right people.

Groupware provides many of the components we need for creating intergalactic client/server applications. The technology is also starting to encroach on its competitors' turf.

2.4.4 Distributed objects

Distributed-object technology promises the most flexible client/server systems. This is because it encapsulates data and business logic in objects that can roam anywhere on networks, run on different platforms, talk to legacy applications by way of object wrappers, and manage themselves and the resources they control. They are designed to become the currency of the intergalactic client/server era.

When it comes to standards, distributed-object technology is way ahead of all other client/server approaches. Since 1989, OMG has been busy specifying the architecture for an open software bus on which object components written by different vendors can interoperate across networks and operating systems.

The secret to OMG's success is that it defined how to specify an interface between a component and the object bus. Specifications are written in IDL, independent of any programming language. IDL becomes the contract that binds client to server components. The beauty of IDL is that it can easily be used to encapsulate existing applications. This way, existing applications need not be rewritten in order to take the full advantages of distributed-object technology.

In addition to defining the object bus, OMG has specified an extensive set of ORB-related services for creating and deleting objects, accessing them by names, defining the complex relationships among them. Later, OMG also defined a comprehensive set of services for transactional objects such that ordinary object can be created and maintained (e.g. make it transactional, lockable and persistent by having it inherit the appropriate services) using simple IDL entries.

Being reckon as the currency for the future intergalactic client/server era, distributed objects have to be powerful enough to replace all other client/server models in many aspects. For instance, TP monitors now have the advantages of better handling in transactions, concurrency, and scalability. So what can distributed objects do in order to over run TP monitors? The OMG has anticipated these problems long ago. The CORBA 2.0 aims to solve them by defining key object services, including transactions, concurrency, and relationships etc. Microsoft, with the help from Digital Equipment, has a rival solution known as COM (Commom Object Model) to solve the same problems.

2.5 Characteristics and Features in Client-Server Computing

Although there are various different configurations, different hardware and software platforms and even different network protocols in Client-Server Architecture. Generally they all possess certain characteristics and features that distinguish them from traditional mainframe computing environment.

2.5.1 Consists of a networked webs of small and powerful machines (both servers and clients)

Client-Server Computing uses local processing power-the power of desktop platform. It changes the way enterprise accesses, distributes, and uses data. With this approach, data is no longer under the tight control of Seniors Managers and MIS (Management of Information Systems) staff; it is readily available to middle-rank personnel and staff. They can actively be involved in the decision-making and operation on behalf of the company. The company becomes more flexible and gives a faster response to the changing business environment outside. In addition, if one machine goes down, the company will still function properly.

2.5.2 Open Systems

Another feature of Client-Server Computing is open systems, which means you can configure your systems, both software and hardware from various vendors as long as they stick to a common standard. In this way, company can tailor their system for their particular situation and needs, pick up and choose the most cost-effective hardware and software components to suit their tasks. For example, you can grab the data and run it through a spreadsheet from your desktop using the brand of computer and software tools that you're most comfortable with and get the job done in your own way.

2.5.3 Modularity

Since we are mixing softwares and hardwares of different natures together as a whole. All the software and hardware components are actually modular in nature. This modularity allows the system to expand and modernizer to meet requirements and needs as the company grows. You can add or remove certain client or machine, implement some new application softwares and even add some hardware features without affecting the operation and functioning of the Client-Server System as a whole. Besides, as new computing platforms emerge, you can evaluate new environments and system components in a modular fashion.

2.5.4 Cost Reduction and Better Utilization of Resources

Potential cost savings prompt organizations to consider Client-Server Computing. The combined base price of hardware (machines and networks) and software for Client/Server systems is often a tenth of the cost to that of mainframe computing. Furthermore, another feature of Client-Server Computing is able to link existing hardware and software applications and utilize them in an efficient way.

2.5.5 Complexity

Since the environment is typically heterogeneous and multivendor. The hardware platform and operating system of client and server are not usually the same. The biggest challenge to successfully implementing the system is to put together this complex system of hardware and software from multiple vendors. Therefore we need expertise not just for software, hardware or networks but expertise of all these fields and understand the interdependencies and interconnection. Sometimes when the system is down, it is extremely difficult to identify a bug or mistake, but there are just several culprits that might cause the problems. Furthermore we have to pay extra efforts and times in training IS professionals to maintain this new environment in geographically dispersed location.

2.6 Main Applications

Client-Server Computing can also be categorized by their support function. The architecture of Client-Server Computing promotes group interaction, whether it is messages, mail, shared data, or shared applications. Users can be "closer" to one another. Users of an application can be anywhere on the network.

There are three main types of Client-Server Applications:

2.6.1 Database Access

This is the most important kind of application in Client-Server Computing. Applications in GUIs are written to access corporate data. These query- oriented applications provide a single window to the data of the organization. In some cases these applications are read-only, in others they are read-write. Benefits of these systems are also ease of use and increased worker productivity. Productivity with these systems is measured by how easily workers can access the data they need to do the job. As we can see, we have a networked web of workstations for workers to access at any node, their productivity will increase tremendously. This kind of database system also provide transparent and consistent access to data wherever it is located.

2.6.2 Transaction-Processing Applications

Typical OLTP (Online-Transaction Processing Applications), also known as mission-critical applications, include order entry, inventory, and point-of-sale systems.

This kind of mission-critical system must run continuously. If it is unavailable even for a brief moment, the organizations will experience severe repercussions.

Examples are stock exchange system, air traffic control networks and airline reservation systems. The transactions are generated at the client and sent to the server for processing. The server may in turn, send one or more operations to other servers.

For a transaction to be considered complete, all operations must be successfully performed. If any operation of transaction cannot be completed, the operations that have taken effect must be reversed using a process called commit and rollback.

2.6.3 Office Systems

Many organizations are employing Client-Server Computing to improve interpersonal communications, both internally and externally. Many organizations are using their linked LANs as a network for enterprise-wide mail systems and workgroup applications. In this way, personnel in organization can improve coordination and actively participate in strategy formulation and decision-making of the company.

CHAPTER THREE

ADVANTAGES OF CLIENT/SERVER COMPUTING

3.1 General

Organizations want to take advantage of the low-cost and user-friendly environment that existing desktop workstations provide. There is also a strong need and desire to capitalize on existing investment at the desktop and in the portfolio of business applications currently running in the host. Thus, corporate networks are typically put in place to connect user workstations to the host. Immediate benefits are possible by integrating these three technologies: workstations, connectivity, and hosts. Retraining and redevelopment costs are avoided by using the existing applications from an integrated desktop.

Client/server computing provides the capability to use the most cost-effective user interface, data storage, connectivity, and application services. Frequently, client/server products are deployed within the present organization but are not used effectively. The client/server model provides the technological means to use previous investments in concert with current technology options. There has been a dramatic decline in the cost of the technology components of client/server computing. Organizations see opportunities to use technology to provide business solutions. Service and quality competition in the marketplace further increase the need to take advantage of the benefits available from applications built on the client/server model.

Client/server computing in its best implementations moves the data-capture and information-processing functions directly to the knowledgeable worker—that is, the worker with the ability to respond to errors in the data, and the worker with the ability to use the information made available. Systems used in the front office, directly involved in the process of doing the business, are forced to show value. If they don't, they are discarded under the cost pressures of doing business. Systems that operate in the back room after the business process is complete are frequently designed and implemented to satisfy an administrative need, without regard to their impact on business operations. Client/server applications integrate the front and back office processes because data capture and usage become an integral part of the business rather than an after-the-fact administrative process. In this mode of operation, the processes are continuously evaluated for effectiveness. Client/server computing provides the technology platform to support the vital business practice of continuous improvement.

3.2 The Advantages of Client/Server Computing

The client/server computing model provides the means to integrate personal productivity applications for an individual employee or manager with specific business data processing needs to satisfy total information processing requirements for the entire enterprise.

3.2.1 Enhanced Data Sharing

Data that is collected as part of the normal business process and maintained on a server is immediately available to all authorized users. The use of Structured Query Language (SQL) to define and manipulate the data provides support for open access from all client processors and software. SQL grants all authorized users access to the information through a view that is consistent with their business need. Transparent network services ensure that the same data is available with the same currency to all designated users.

3.2.2 Integrated Services

In the client/server model, all information that the client (user) is entitled to use is available at the desktop. There is no need to change into terminal mode or log into another processor to access information. All authorized information and processes are directly available from the desktop interface. The desktop tools—e-mail, spreadsheet, presentation graphics, and word processing—are available and can be used to deal with information provided by application and database servers resident on the network. Desktop users can use their desktop tools in conjunction with information made available from the corporate systems to produce new and useful information.

3.2.3 Sharing Resources Among Diverse Platforms

The client/server computing model provides opportunities to achieve true open system computing. Applications may be created and implemented without regard to the hardware platforms or the technical characteristics of the software. Thus, users may obtain client services and transparent access to the services provided by database, communications, and applications servers. Operating systems software and platform hardware are independent of the application and masked by the development tools used to build the application.

In this approach, business applications are developed to deal with business processes invoked by the existence of a user-created "event." An event such as the push of a button, selection of a list element, entry in a dialog box, scan of a bar code, or flow of gasoline occurs without the application logic being sensitive to the physical platforms. Client/server applications operate in one of two ways. They can function as the front end to an existing application or they can provide data entry, storage, and reporting by using a distributed set of clients and servers. In either case, the use—or even the existence—of a mainframe host is totally masked from the workstation developer by the use of standard interfaces such as SQL.

3.2.4 Data Interchangeability and Interoperability

SQL is an industry-standard data definition and access language. This standard definition has enabled many vendors to develop production-class database engines to manage data as SQL tables. Almost all the development tools used for client/server development expect to reference a back-end database server accessed through SQL. Network services provide transparent connectivity between the client and local or remote servers. With some database products, such as Ingres Star, a user or application can define a consolidated view of data that is actually distributed between heterogeneous, multiple platforms.

3.2.5 Masked Physical Data Access

When SQL is used for data access, users can access information from databases anywhere in the network. From the local PC, local server, or wide area network (WAN) server, data access is supported with the developer and user using the same data request. The only noticeable difference may be performance degradation if the network bandwidth is inadequate. Data may be accessed from dynamic random-access memory (D-RAM), from magnetic disk, or from optical disk, with the same SQL statements. Logical tables can be accessed—without any knowledge of the ordering of columns or awareness of extraneous columns—by selecting a subset of the columns in a table. Several tables may be joined into a view that creates a new logical table for application program manipulation, without regard to its physical storage format.

The use of new data types, such as binary large objects (BLOBs), enables other types of information such as images, video, and audio to be stored and accessed using the same SQL statements for data access. RPCs frequently include data conversion facilities to translate the stored data of one processor into an acceptable format for another.

3.2.6 Location Independence of Data and Processing

We are moving from the machine-centered computing era of the 1970s and 1980s to a new era in which PC-familiar users demand systems that are user-centered. Previously, a user logged into a mainframe, mini-, or microapplication. The syntax of access was unique in each platform. Function keys, error messages, navigation methods, security, performance, and editing were all very visible. Today's users expect a standard "look and feel." Users log into an application from the desktop with no concern for the location or technology of the processors involved.

3.2.7 Centralized Management

As processing steers away from the central data center to the remote office and plant, workstation server, and local area network (LAN) reliability must approach that provided today by the centrally located mini- and mainframe computers. The most effective way to ensure this is through the provision of monitoring and support from these same central locations. A combination of technologies that can "see" the operation of hardware and software on the LAN—monitored by experienced support personnel—provides the best opportunity to achieve the level of reliability required.

The first step in effectively providing remote LAN management is to establish standards for hardware, software, networking, installation, development, and naming. These standards, used in concert with products such as IBM's Systemview, Hewlett-Packard's Openview, Elegant's ESRA, Digital's EMA, and AT&T's UNMA products, provide the remote view of the LAN. Other tools, such as PC Connect for remote connect, PCAssure from Centel for security, products for hardware and software inventory, and local monitoring tools such as Network General's Sniffer, are necessary for completing the management process.

3.3 Connectivity

The era of desktop workstations began in 1981 with the introduction of the IBM personal computer (PC). The PC provided early users with the capability to do spreadsheets, word processing, and basic database services for personal data. Within three years, it became clear that high-quality printers, backup tapes, high-capacity disk devices, and software products were too expensive to put on everyone's desktop. LAN technology evolved to solve this problem. Novell is and has been the most successful vendor in the LAN market.

Step 1—Workstations Emulate Corporate Systems

In most large organizations, desktop workstations provide personal productivity and some workgroup functions, but host services still provide most other business functions. The lack of desktop real estate encourages the addition of terminal emulation services to the workstation. This emulation capability connects the workstation directly to the corporate systems. The connection was and generally still is provided by a direct connection from the workstation to the host server or its controller.

Connectivity provides the opportunity to move beyond terminal emulation to use the full potential of the workstation. Often the first client/server applications in a large organization use existing mainframe applications. These are usually presentation services-only applications.

Step 2—Adding Servers for Database and Communications

The next step in connectivity is the implementation of specialized servers to provide database and communications services. These servers provide LAN users with a common database for shared applications and with a shared node to connect into the corporate network. The communications servers eliminate the need for extra cabling and workstation hardware to enable terminal emulation. The LAN cabling provides the necessary physical connection, and the communications server provides the necessary controller services.

Step 3—Full-Fledged Client/Server Applications

With its implementation of communications and database servers in place, an organization is ready for the next step up from presentation services-only client/server applications to full-fledged client/server applications. These new applications are built on the architecture defined as part of the system development environment (SDE).

3.4 User Productivity

Personal computer users are accustomed to being in control of their environment. Recently, users have been acclimated to the GUI provided by products such as Windows 3.x, OPEN LOOK, MacOS, and NeXtStep. Productivity is enhanced by the standard look and feel that most applications running in these environments provide. A user is trained both to get into applications and to move from function to function in a standard way. Users are accustomed to the availability of context-sensitive help, "friendly" error handling, rapid performance, and flexibility.

Compare the productivity achieved by a financial or budget analyst using a spreadsheet program such as Lotus 1-2-3 or Excel to that achieved when similar functionality is programmed in COBOL on a mainframe. Adding a new variable to an analysis or budget is a trivial task compared to the effort of making functions perform a similar change in the mainframe-based COBOL package. In the first instance, the change is made directly by the user who is familiar with the requirement into a visible model of the problem. In the instance of the mainframe, the change must be made by a programmer, who discusses the requirement with the analyst, attempts to understand the issues, and then tries to make the change using an abstraction of the problem.

The personal computer user makes the change and sees the result. The mainframe programmer must make the change, compile the program, invoke the program, and run the test. If the user understands the request, the implications, and the syntactical requirements, he or she may get it right the first time. Usually, it takes several iterations to actually get it right, often in concert with a frustrated user who tries to explain the real requirement.

We aren't suggesting that all applications can be developed by nonprogrammers using desktop-only tools. However, now that it has become rather easy to build these types of applications on the desktop, it is important for professional IS people to understand the expectations raised in the minds of the end-user community.

3.5 Ways to Improve Performance

Client/server-developed applications may achieve substantially greater performance when compared with traditional workstations or host-only applications.

3.5.1 Offload Work to Server

Database and communications processing are frequently offloaded to a faster server processor. Some applications processing also may be offloaded, particularly for a complex process, which is required by many users. The advantage of offloading is realized when the processing power of the server is significantly greater than that of the client workstation. Shared databases or specialized communications interfaces are best supported by separate processors. Thus, the client workstation is available to handle other client tasks. These advantages are best realized when the client workstation supports multitasking or at least easy and rapid task switching.

3.5.2 Reduce Total Execution Time

Database searches, extensive calculations, and stored procedure execution can be performed in parallel by the server while the client workstation deals directly with the current user needs. Several servers can be used together, each performing a specific function. Servers may be multiprocessors with shared memory, which enables programs to overlap the LAN functions and database search functions. In general, the increased power of the server enables it to perform its functions faster than the client workstation. In order for this approach to reduce the total elapsed time, the additional time required to transmit the request over the network to the server must be less than the saving. High-speed local area network topologies operating at 4, 10, 16, or 100Mbs (megabits per second) provide high-speed communications to manage the extra traffic in less time than the savings realized from the server. The time to transmit the request to the server, execute the request, and transmit the result to the requestor, must be less than the time to perform the entire transaction on the client workstation.

3.5.3 Use a Multitasking Client

As workstation users become more sophisticated, the capability to be simultaneously involved in multiple processes becomes attractive. Independent tasks can be activated to manage communications processes, such as electronic mail, electronic feeds from news media and the stock exchange, and remote data collection (downloading from remote servers). Personal productivity applications, such as word processors, spreadsheets, and presentation graphics, can be active. Several of these applications can be dynamically linked together to provide the desktop information processing environment. Functions such as Dynamic Data Exchange (DDE) and Object Linking and Embedding (OLE) permit including spreadsheets dynamically into word-processed documents. These links can be *hot* so that changes in the spreadsheet cause the word-processed document to be updated, or they can be *cut and paste* so that the current status of the spreadsheet is copied into the word-processed document.

Systems developers appreciate the capability to create, compile, link, and test programs in parallel. The complexity introduced by the integrated CASE environment requires multiple processes to be simultaneously active so the workstation need not be dedicated to a single long-running function. Effective use of modern CASE tools and workstation development products requires a client workstation that supports multitasking.

3.6 How to Reduce Network Traffic

Excessive network traffic is one of the most common causes of poor system performance. Designers must take special care to avoid this potential calamity .

3.6.1 Minimize Network Requests

In the centralized host model, network traffic is reduced to the input and output of presentation screens. In the client/server model, it is possible to introduce significantly more network traffic if detailed consideration is not given to the requestor-server interface.

In the file server model, as implemented by many database products, such as dBASE IV, FoxPro, Access, and Paradox, a search is processed in the client workstation. Record-level requests are transmitted to the server, and all filtering is performed on the workstation. This has the effect of causing all rows that cannot be explicitly filtered by primary key selection to be sent to the client workstation for rejection. In a large database, this action can be dramatic. Records that are owned by a client cannot be updated by another client without integrity conflicts. An in-flight transaction might lock records for hours if the client user leaves the workstation without completing the transaction. For this reason, the file server model breaks down when there are many users, or when the database is large and multikey access is required.

However, with the introduction of specific database server products in the client/server implementation, the search request is packaged and sent to the database server for execution. The SQL syntax is very powerful and—when combined with server trigger logic—enables all selection and rejection logic to execute on the server. This approach ensures that the answer set returns only the selected rows and has the effect of reducing the amount of traffic between the server and client on the LAN. (To support the client/server model, dBASE IV, FoxPro, and Paradox products have been retrofitted to be SQL development tools for database servers).

The performance advantages available from the client/server model of SQL services can be overcome. For example, if by using an unqualified SQL SELECT, all rows satisfying the request are returned to the client for further analysis. Minimally qualified requests that rely on the programmer's logic at the workstation for further selection can be exceedingly dangerous. Quite possibly, 1 million rows from the server can be returned to the client only to be reduced by the client to 10 useful rows. The JOIN function in SQL that causes multiple tables to be logically combined into a single table can be dangerous if users don't understand the operation of the database engine.

A classic problem with dynamic SQL is illustrated by a request to Oracle to JOIN a 10-row table at the client with a 1-million-row table at the server. Depending on the format of the request, either 10 useful rows may be transferred to the client or 1 million rows may be transferred so that the useless 999,990 can be discarded. You might argue that a competent programmer should know better; however, this argument breaks down when the requestor is a business analyst. Business analysts should not be expected to work out the intricacies of SQL syntax. Their tools must protect them from this complexity. (Some DBMSs are now making their optimizers more intelligent to deal with just these cases. So, it is important to look beyond transaction volumes when looking at DBMS engines.) If your business requirement necessitates using these types of dynamic SQL requests, it is important, when creating an SDE, that the architecture definition step selects products that have strong support for query optimization. Products such as Ingres are optimized for this type of request.

Online Transaction Processing (OLTP) in the client/server model requires products that use views, triggers, and stored procedures. Products such as Sybase, Ellipse, and Ingres use these facilities at the host server to perform the join, apply edit logic prior to updates, calculate virtual columns, or perform complex calculations. Wise use of OLTP can significantly reduce the traffic between client and server and use the powerful CPU capabilities of the server. Multiprocessor servers with shared memory are available from vendors such as Compaq, Hewlett Packard, and Sun. These enable execution to be divided between processors. CPU-intensive tasks such as query optimization and stored procedures can be separated from the database management processes.

3.6.2 Ease Strain on Network Resources

The use of application and database servers to produce the answer set required for client manipulation will dramatically reduce network traffic. There is no value in moving data to the client when it will be rejected there. The maximum reduction in network overhead is achieved when the only data returned to the client is that necessary to populate the presentation screen. Centralized operation, as implemented in minicomputer and mainframe environments, requires every computer interaction with a user to transfer screen images between the host and the workstation. When the minicomputer or mainframe is located geographically distant from the client workstation, WAN services are invoked to move the screen image. Client/server applications can reduce expensive WAN overhead by using the LAN to provide local communications services between the client workstation and the server. Many client/server applications use mixed LAN and WAN services: some information is managed on the LAN and some on the WAN. Application design must evaluate the requirements of each application to determine the most effective location for application and database servers.

3.6.3 How to Reduce Costs

Cost of operation is always a major design factor. Appropriate choice of technology and allocation of the work to be done can result in dramatic cost reduction.

Each mainframe user requires a certain amount of the expensive mainframe CPU to execute the client portion of the application. Each CICS user uses CPU cycles, disk queues, and D-RAM. These same resources are orders of magnitude cheaper on the workstation. If the same or better functionality can be provided by using the workstation as a client, significant savings can be realized. Frequently existing workstations currently used for personal productivity applications, such as terminal emulation, e-mail, word processing, and spreadsheet work may be used for mission-critical applications. The additional functionality of the client portion of a new application can thus be added without buying a new workstation. In this case, the cost savings of offloading mainframe processing can be substantial.

When you use a communications server on a LAN, each client workstation does not need to contain the hardware and software necessary to connect to the WAN. Communications servers can handle up to 128 clients for the cost of approximately six client communications cards and software. Despite the dramatic reductions in the price of D-RAM, companies will continue to need their existing client workstations. These devices may not be capable of further D-RAM upgrades, or it may not be feasible from a maintenance perspective to upgrade each device. The use of server technology to provide some of the functionality currently provided within a client workstation frees up valuable D-RAM for use by the client applications. This is particularly valuable for DOS-based clients.

The WAN communications functions and LAN services may each be offloaded in certain implementations. The use of WAN communications servers has the additional advantage of providing greater functionality from the dedicated communications server.

3.7 Vendor Independence

If client and server functionality is clearly split and standards-based access is used, there can be considerable vendor independence among application components. Most organizations use more expensive and more reliable workstations from a mainstream vendor such as Compaq, IBM, Apple, Sun, or Hewlett-Packard for their servers. Other organizations view client workstation technology as a commodity and select lower-priced and possibly less-reliable vendor equipment. The mainstream vendors have realized this trend and are providing competitively priced client workstations. Each of the mainstream vendors reduced its prices by at least 65 percent between 1991-93, primarily in response to an erosion of market share for client workstations.

The controversy over whether to move from offering a high-priced but best-quality product line to offering a more competitive commodity traumatized the industry in 1991, forcing Compaq to choose between retaining its founder as CEO or replacing him with a more fiscally aware upstart.

The resulting shakeout in the industry has significantly reduced the number of vendors and makes the use of traditionally low priced clones very risky. Hardware can generally be supported by third-party engineers, but software compatibility is a serious concern as organizations find they are unable to install and run new products.

The careful use of SQL and RPC requests enable database servers and application services to be used without regard to the vendor of the database engine or the application services platform. As noted previously, the operating system and hardware platform of the server can be kept totally independent of the client platform through the proper use of an SDE. However, use of these types of technologies can vastly complicate the development process.

An excellent example of this independence is the movement of products such as FoxPro and Paradox to use client services to invoke, through SQL, the server functions provided by Sybase SQL Server. A recent survey of client development products that support the Sybase SQL Server product identified 129 products. This is a result of the openness of the API provided by Sybase. Oracle also has provided access to its API, and several vendors—notably Concentric Data Systems, SQL Solutions, and DataEase—have developed front-end products for use with Oracle. ASK also has realized the importance of open access to buyers and is working with vendors such as Fox and PowerBuilder to port their front ends in support of the Ingres database engine.

An application developed to run in a single PC or file server mode can be migrated without modification to a client/server implementation using a database server. Sybase, Oracle, and Ingres execute transparently under Windows NT, OS/2, or UNIX on many hardware platforms. With some design care, the server platform identity can be transparent to the client user or developer. Despite this exciting opportunity, programmers or manufacturers often eliminate this transparency by incorporating UNIX-, Windows NT-, or OS/2-specific features into the implementation. Although FoxPro can work with SQL and Sybase, the default Xbase format for database access does not use SQL and therefore does not offer this independence. To take advantage of this platform transparency, organizations must institute standards into their development practices.

3.8 Faster Delivery of Systems

Some software development and systems integration vendors have had considerable success using client/server platforms for the development of systems targeted completely for mainframe execution. These developer workstations are often the first true client/server applications implemented by many organizations. The workstation environment, powerful multitasking CPU availability, single-user databases, and integrated testing tools all combine to provide the developer with considerable productivity improvements in a lower-cost environment. Our analysis shows that organizations that measure the "real" cost of mainframe computing will cost justify workstation development environments in 3 to 12 months.

Client/server application development shows considerable productivity improvement when the software is implemented within an SDE. As previously noted, organizational standards-based development provides the basis for object-oriented development techniques and considerable code reuse. This is particularly relevant in the client/server model, because some natural structuring takes place with the division of functionality between the client and server environments

3.8.1 Smaller and Simpler Problems

Client/server applications frequently are involved with data creation or data analysis. In such applications, the functionality is personal to a single user or a few users. These applications frequently can be created using standard desktop products with minimal functionality. For example, data may be captured directly into a form built with a forms development tool, edited by a word processor, and sent on through the e-mail system to a records management application. In the back end, data may be downloaded to a workstation for spreadsheet analysis.

3.8.2 Less Bureaucratic Inertia

Mainframes provide the stable, reliable environment that is desirable and necessary for production execution. This same stability is the bane of developers who require rapid changes to their test environments. The workstation environment is preferable because it is personal and responds to the user's priorities. Developers can make changes at their own pace and then deal with the mainframe bureaucracy if and when the application goes into production in the mainframe environment.

Many users typically run applications on the mainframe. Changes made to such applications affect all their users. In some instances, the entire mainframe may be unavailable during the implementation of a new application. Network reconfiguration, database utilities, application definition, and system software maintenance all can impact users beyond those specifically involved in a change. It is awkward to migrate only a portion of the users from the previous implementation to the new one. Typically, it is all or none of the users who must upgrade. This change process requires thorough and all-encompassing tests and careful control over the move to production.

The client/server environment provides more flexibility for phased implementation of the new production environment. The application is replicated at many different locations so the users may implement the new software individually rather than all at once. This environment adds the additional and significant complication of multiple updates. on.

Workgroup client/server applications frequently are used by fewer users. These users can be directly supported by the developer immediately after implementation. Corrections can be made and reimplemented more readily. This is not to suggest that in the client/server world change and production control procedures are not necessary, only that they can be less onerous for workgroup applications. Remote LAN management will be required for enterprise applications implemented throughout the corporation. Only in this way will support equivalent to that available today for host-based applications be available to remote client/server users.

CHAPTER FOUR

COMPONENTS OF CLIENT/SERVER COMPUTING AND ITS APPLICATIONS-THE CLIENT

4.1 General

The client in the client/server model is the desktop workstation. Any workstation that is used by a single user is a client. The same workstation, when shared simultaneously by multiple users, is a server. An Apple Macintosh SE, an IBM PS/2 Model 30, an ALR 386/220, a Compaq SystemPro, an NCD X-Terminal, a Sun Sparcstation, a DECstation 5000—all are used somewhere as a client workstation. There is no specific technological characteristic of a client.

During the past 15 years, workstation performance improved dramatically. For the same cost, workstation CPU performance increased by 50 times, main memory has increased by 25 times, and permanent disk storage has increased by 30 times. This growth in power allows much more sophisticated applications to be run from the desktop.

Communications and network speeds have improved equally in the last 15 years. In 1984, the performance and reliability of remote file, database, and print services were inadequate to support business applications. With the advent of high-speed local and wide area networks (LANs and WANs), networking protocols, digital switches, and fiber-optic cabling, both performance and reliability improved substantially. It is now practical to use these remote services as part of a critical business application.

The client workstation may use the DOS, Windows, Windows NT, OS/2, MacOS (also referred to as System 7), or UNIX operating system. The client workstation frequently provides personal productivity functions, such as word processing, which use only the hardware and software resident right on the workstation. When the client workstation is connected to a LAN, it has access to the services provided by the network operating system (NOS) in addition to those provided by the client workstation. The workstation may load software and save word-processed documents from a server and therefore use the file server functions provided through the NOS. It also can print to a remote printer through the NOS. The client workstation may be used as a terminal to access applications resident on a host minicomputer or mainframe processor. This enables the single workstation to replace the terminal, as well as provide client workstation functionality.

In a client/server application, functions are provided by a combination of resources using both the client workstation processor and the server processor. For example, a database server provides data in response to an SQL request issued by the client application. Local processing by the client might calculate the invoice amount and format the response to the workstation screen.

4.2 The Role of the Client

In the client/server model, the client is primarily a consumer of services provided by one or more server processors. The model provides a clear separation of functions based on the idea of servers acting as service providers responding to requests from clients. It is important to understand that a workstation can operate as a client in some instances while acting as a server in other instances. For example, in a LAN Manager environment, a workstation might act as a client for one user while simultaneously acting as a print server for many users. This chapter discusses the client functions.

The client almost always provides presentation services. User input and final output, if any, are presented at the client workstation. Current technology provides cost effective support for a graphical user interface (GUI). The windowing environment enables the client user to be involved in several simultaneous sessions. Such functions as word processing, spreadsheet, e-mail, and presentation graphics—in addition to the custom applications built by the organization—can be active simultaneously.

Facilities such as Dynamic Data Exchange (DDE), Object Level Embedding (OLE), and Communicating Object Request Broker Architecture (CORBA), which are discussed later in this chapter, provide support for cut-and-paste operations between word processors, databases, spreadsheets, and graphics in a windowing environment. Beyond this, a selectable set of tasks may be performed at the client. In fact, the client workstation can be both client and server when all information and logic pertinent to a request is resident and operates within the client workstation.

Software to support specific functions—for example, field edits, con-text-sensitive help, navigation, training, personal data storage, and manipulation—frequently executes on the client workstation. All these functions use the GUI and windowing functionality. Additional business logic for calculations, selection, and analysis can reside on the client workstation.

A client workstation uses a local operating system to host both basic services and the network operating system interfaces. This operating system may be the same or different from that of the server. Most personal computer users today use DOS or Windows 3.x as their client operating system, because current uses are primarily personal productivity applications—not ones requiring a client/server.

Those users running client/server applications from DOS or Windows typically run only a single business process at a time.

4.3 Client Services

The ideal client/server platform operates in an open systems environment using a requester-server discipline that is based on well-defined standards. This enables multiple hardware and software platforms to interact. When the standard requester-server discipline is adhered to, servers may grow and change their operating system and hardware platforms without changing the client applications. Clients can be entry-level Intel 386SX machines or very powerful RISC-based workstations, and run the same application issuing the same requests for service as long as the standard requester-server discipline is adhered to. Traditional host applications that use the client for presentation services operate only by sending and receiving a character data stream to and from a server. All application logic resides on the server. This is the manner in which many organizations use workstation technology today. The expensive mainframe CPU is being used to handle functions that are much more economically provided by the workstation.

First-generation client/server applications using software such as Easel enable the input and output data streams to be reformatted at the client without changes to the host applications. They use an API that defines the data stream format. Easel uses the IBM-defined Extended High Level Language Application Program Interface (EHLLAPI). GUI front ends may add additional functionality, such as the capability to select items for input from a list, selectively use color, or merge other data into the presentation without changing the host application.

An example of this form of client is an application developed for the emergency command and control services required by E911 dispatch applications. This computer application supports calls to the 911 emergency telephone number and dispatches fire, police, ambulance, or emergency vehicles to an incident.

4.4 Requests for Service

Client workstations request services from the attached server. Whether this server is in fact the same processor or a network processor, the application format of the request is the same. NOS software translates or adds the specifics required by the targeted requester to the application request.

Interprocess communication (IPC) is the generic term used to describe communication between running processes. In the client/server model, these processes might be on the same computer, across the LAN, or across the WAN.

The most basic service provided by the NOS is *redirection*. This service intercepts client workstation operating system calls and redirects them to the server operating system. In this way, requests for disk directories, disk files, printers, printer queues, serial devices, application programs, and named pipes are trapped by the redirection software and redirected (over the LAN) to the correct server location. It is still possible for some of these services to be provided by the client workstation. The local disk drives may be labeled A: and C: and the remote drives labeled D:, E:, and F:.
How does redirection work?

1. Any request for drive A: or C: is passed through to the local file system by the redirection software. Requests for other drives are passed to the server operating system. Printers are accessed through virtual serial and parallel ports defined by the NOS redirector software.

The NOS requester software constructs the remote procedure call (RPC) to include the API call to the NOS server.

2. The NOS server then processes the request as if it were executed locally and ships the response back to the application.

Novell commercialized this redirector concept for the Intel and MS-DOS platforms, and it has been adopted by all NOS and UNIX network file system (NFS) vendors. The simplicity of executing standard calls to a virtual network of services is its main advantage.

4.4.1 Remote Procedure Call (RPC)

Over the years, good programmers have developed modular code using structured techniques and subroutine logic. Today, developers want subroutines to be stored as a named objects "somewhere" and made available to everyone with the right to use them. Remote procedure calls (RPCs) provide this capability. RPCs standardize the way programmers must write calls, so that remote procedures can recognize and respond correctly.

If an application issues a functional request and this request is embedded in an RPC, the requested function can be located anywhere in the enterprise that the caller is authorized to access. The RPC facility provides for the invocation and execution of requests from processors running different operating systems and using hardware platforms different from that of the caller. Many RPCs also provide data translation services. The call causes dynamic translation of data between processors with different physical data storage formats. These standards are evolving and being adopted by the industry.

4.4.2 Fax/Print Services

The NOS enables the client to generate print requests even when the printer is busy. These are redirected by the NOS redirector software and managed by the print server queue manager. The client workstation can view the status of the print queues at any time. Many print servers notify the client workstation when the print request is completed. Fax services are made available in exactly the same manner as print servers, with the same requester server interface and notification made available.

4.4.3 Window Services

A client workstation may have several windows open on-screen at any time. The capability to activate, view, move, size, or hide a particular window is provided by the window services of the client operating system. These services are essential in a client/server implementation, because they interact with message services provided to notify the user of events that occur on a server. Application programs are written with no sensitivity to the windowing. Each application is written with the assumption that it has a virtual screen. This virtual screen can be an arbitrary size and can even be larger than the physical screen.

The application, using GUI software, places data into the virtual screen, and the windowing services handle placement and manipulation of the application window. This greatly simplifies application development, because there is no need for the developer to build or manage the windowing services. The client user is totally in control of his or her desktop and can give priority to the most important tasks at hand simply by positioning the window of interest to the "front and center." The NOS provides software on the client workstation to manage the creation of pop-up windows that display alerts generated from remote servers. E-mail receipt, print complete, Fax available, and application termination are examples of alerts that might generate a pop-up window to notify the client user.

4.4.4 Remote Boot Services

Some applications operate well on workstations without any local disk storage; X-terminals and workstations used in secure locations are examples. The client workstation must provide sufficient software burned into erasable programmable read-only memory (E-PROM) to start the initial program load (IPL)—that is, boot—process.

E-PROM is included in all workstations to hold the Basic Input/Output System (BIOS) services. This mini-operating system is powerful enough to load the remote software that provides the remaining services and applications functions to the client workstation or X-terminal.

4.4.5 Utility Services

The operating system provides local functions such as copy, move, edit, compare, and help that execute on the client workstation.

4.4.6 Message Services

Messages can be sent and received synchronously to or from the network. The message services provide the buffering, scheduling, and arbitration services to support this function.

4.4.7 Network Services

The client workstation communicates with the network through a set of services and APIs that create, send, receive, and format network messages. These services provide support for communications protocols, such as NetBIOS, IPX, TCP/IP, APPC, Ethernet, Token Ring, FDDI, X.25, and SNA.

4.4.8 Application Services

In addition to the remote execution services that the NOS provides, custom applications will use their own APIs embedded in an RPC to invoke specialized services from a remote server.

4.4.9 Database Services

Database requests are made using the SQL syntax. SQL is an industry standard language supported by many vendors. Because the language uses a standard form, the same application may be run on multiple platforms. There are syntactical differences and product extensions available from most vendors. These are provided to improve developer productivity and system performance and should be carefully evaluated to determine whether their uses are worth the incompatibility implied by using proprietary components. Using unique features may prevent the use of another vendor's products in a larger or smaller site. Certain extensions, such as stored procedures, are evolving into *de facto* standards.

The use of stored procedures is often a way of avoiding programmer use of proprietary extensions needed for performance. A clear understanding, by the technical architects on the project, of where the standards are going is an important component of the SDE standards for the project.

4.4.10 Network Management Services-Alerts

Most network interface cards (NICs) can generate alerts to signify detected errors and perhaps to signify messages sent and received. These alerts are valuable in remote LAN management to enable early detection of failures. Because many errors are transient at first, simple remote detection may allow problems to be resolved before they become critical. Applications may also generate alerts to signify real or potential problems. Certain error conditions indicate that important procedures are not being followed. Application program failure may occur because current versions of software are not being used.

Support for a remote client workstation may be greatly simplified if alerts are generated by the applications. This should be part of every standard SDE. Many alert situations can be generated automatically from standard code without the involvement of the application developer.

4.4.11 Dynamic Data Exchange (DDE)

DDE is a feature of Windows 3.x and OS/2 Presentation Manager that enables users to pass data between applications from different vendors through support for common APIs. For example, a charting package can be linked to a database to provide the latest chart data whenever the chart is referenced.

4.4.12 Object Linking and Embedding (OLE)

OLE is an extension to DDE that enables objects to be created with the object components software *aware*. Aware means that a reference to the object or one of its components automatically launches the appropriate software to manipulate the data. For example, a document created with a word processor may include an image created by a graphics package. The image can be converted to the internal graphics form of the word processor, such as WPG form for WordPerfect. With OLE, the image can be included in its original form within the document object; whenever the image is selected or highlighted, the graphics package will take control to manipulate the image. Activation of the software is totally transparent to the users as they navigate through the document.

Currently with OLE, one software package accesses data created from another through the use of a *viewer* or *launcher*. These viewers and launchers must be custom built for every application. With the viewer, users can see data from one software package while they are running another package. Launchers invoke the software package that created the data and thus provide the full functionality of the launched software.

Both these techniques require the user to be aware of the difference between data sources. DDE and OLE provide a substantial advantage: any DDE- or OLE-enabled application can use any software that supports these data interchange APIs. An e-mail application will be able to attach any number of components into the mail object without the need to provide custom viewers or launchers.

Not all Windows applications support OLE, but Microsoft has released its OLE 2.0 software development kit (SDK). The toolkit greatly simplifies OLE integration into third-party, developed applications. Organizations wanting to create a consistent desktop are beginning to use the OLE SDK as part of custom applications.

4.4.13 Common Object Request Broker Architecture (CORBA)

CORBA is a specification from the Object Management Group (OMG), a UNIX vendor consortium. OLE focuses on data sharing between applications on a single desktop, and CORBA addresses cross-platform data transfer and the process of moving objects over networks. CORBA support enables Windows and UNIX clients to share objects. A word processor operating on a Windows desktop can include graphics generated from a UNIX workstation.

4.5 Enterprise View

It is important for application designers and developers to understand and remember that the user view of the system is through the client workstation. Whatever technological miracles are performed at the server, a poor design or implementation at the client on the desktop still result in unfavorable user perception of the entire application!

CHAPTER FIVE

COMPONENTS OF CLIENT/SERVER COMPUTING AND ITS APPLICATIONS-THE SERVER

5.1 General

The server is a multiuser computer. There is no special hardware requirement that turns a computer into a server. The hardware platform should be selected based on application demands and economics. Servers for client/server applications work best when they are configured with an operating system that supports shared memory, application isolation, and preemptive multitasking. An operating system with preemptive multitasking enables a higher priority task to preempt or take control of the processor from a currently executing, lower priority task.

The server provides and controls shared access to server resources. Applications on a server must be isolated from each other so that an error in one cannot damage another. Preemptive multitasking ensures that no single task can take over all the resources of the server and prevent other tasks from providing service. There must be a means of defining the relative priority of the tasks on the server. These requirements are specific to the client/server implementation and not to the file server implementation. Because file servers execute only the single task of file service, they can operate in a more limited operating environment without the need for application isolation and preemptive multitasking.

Many organizations download data from legacy enterprise servers for local manipulation at workstations. In the client/server model, the definition of server will continue to include these functions, perhaps still implemented on the same or similar platforms. Moreover, the advent of open systems based servers is facilitating the placement of services on many different platforms.

Client/server computing is a phenomenon that developed from the ground up. Remote workgroups have needed to share expensive resources and have connected their desktop workstations into local area networks (LANs). LANs have grown until they are pervasive in the organization. However, frequently (similar to parking lots) they are isolated one from the other.

The Novell Network Operating System (NOS), NetWare, is the most widely installed LAN NOS. It provides the premier file and print server support. However, a limitation of NetWare for the needs of reliable client/server applications has been the requirement for an additional separate processor running as a database server. The availability of database server software—from companies such as Sybase and Oracle—to run on the NetWare server, is helping to diffuse this limitation. With the release of Novell 4.x, Netware supports an enterprise LAN (that is, a thousand internetworked devices) with better support for Directory Services and TCP/IP internetworking.

DEC demonstrated the Alpha AXP processor running Processor-Independent NetWare in native mode at the PC Expo exhibit in June 1993. HP, Sun, and other vendors developing NetWare on RISC-based systems announced shipment of developer kits for availability in early 1994. Native NetWare for RISC is scheduled for availability in late 1994. This will provide scalability for existing Netware users who run out of capacity on their Intel platforms.

Banyan VINES provides the competitive product to Novell 4.x for enterprise LANs. Directory services are provided in VINES through a feature called StreetTalk. VINES 5.5 provides excellent WAN connectivity and is very popular among customers with a heterogeneous mainframe and minicomputer enterprise. However, it suffers from a weak support for file and printer sharing and a general lack of application package support. Banyan's Enterprise Network Services (ENS) with StreetTalk provides the best Directory Services implementation today.

Microsoft's LAN Manager NOS and its several derivatives—including IBM Lan Server, HP LAN Manager/UX and DEC Pathworks—provide file and printer services but with less functionality, and more user complexity, than Novell's NetWare. The operating systems that support LAN Manager provide the necessary shared memory, protected memory, and preemptive multitasking services necessary for reliable client/server computing. They provide this support by operating natively with the OS/2, UNIX, VMS, and MVS operating systems. These operating systems all provide these services as part of their base functionality. The scalability of the platforms provides a real advantage for organizations building client/server, and not just file server, applications.

Network File System (NFS) is the standard UNIX support for shared files and printers. NFS provides another option for file and print services to client workstations with access to a UNIX server. PC NFS is the PC product that runs on the client and provides connectivity to the NFS file services under UNIX. NFS with TCP/IP provides the additional advantage of easy-to-use support for remote files and printers.

There is no preeminent hardware technology for the server. The primary characteristic of the server is its support for multiple simultaneous client requests for service. Therefore, the server must provide multitasking support and shared memory services.

5.2 The Role of the Server

Servers provide application, file, database, print, fax, image, communications, security, systems, and network management services. These are each described in some detail in the following sections.

It is important to understand that a server is an architectural concept, not a physical implementation description. Client and server functions can be provided by the same physical device. With the movement toward peer computing, every device will potentially operate as a client and server in response to requests for service.

Application servers provide business functionality to support the operation of the client workstation. In the client/server model these services can be provided for an entire or partial business function invoked through an InterProcess Communication (IPC) request for service.

Either message-based requests RPCs can be used. A collection of application servers may work in concert to provide an entire business function. For example, in a payroll system the employee information may be managed by one application server, earnings calculated by another application server, and deductions calculated by a third application server. These servers may run different operating systems on various hardware platforms and may use different database servers. The client application invokes these services without consideration of the technology or geographic location of the various servers. Object technology provides the technical basis for the application server, and widespread acceptance of the CORBA standards is ensuring the viability of this trend. File servers provide record level data services to nondatabase applications.

Space for storage is allocated, and free space is managed by the file server. Catalog functions are provided by the file server to support file naming and directory structure. Filename maximum length ranges from 8 to 256 characters, depending on the particular server operating system support. Stored programs are typically loaded from a file server for execution on a client or host server platform.

Database servers are managed by a database engine such as Sybase, IBM, Ingres, Informix, or Oracle. The file server provides the initial space, and the database engine allocates space for tables within the space provided by the file server. These host services are responsible for providing the specialized data services required of a database product—automatic backout and recovery after power, hardware, or software failure, space management within the file, database reorganization, record locking, deadlock detection, and management. Print servers provide support to receive client documents, queue them for printing, prioritize them, and execute the specific print driver logic required for the selected printer. The print server software must have the necessary logic to support the unique characteristics of each printer.

5.3 Server Functionality in Detail

The discussion in the following sections more specifically describes the functions provided by the server in a NOS environment.

5.3.1 Request Processing

Requests are issued by a client to the NOS services software resident on the client machine. These services format the request into an appropriate RPC and issue the request to the application layer of the client protocol stack. This request is received by the application layer of the protocol stack on the server.

5.3.2 File Services

File services handle access to the virtual directories and files located on the client workstation and to the server's permanent storage. These services are provided through the redirection software implemented as part of the client workstation operating environment. The file services provide this support at the remote server processor. In the typical implementation, software, shared data, databases, and backups are stored on disk, tape, and optical storage devices that are managed by the file server.

To minimize the effort and effect of installation and maintenance of software, software should be loaded from the server for execution on the client. New versions can be updated on the server and made immediately available to all users. In addition, installation in a central location reduces the effort required for each workstation user to handle the installation process. Because each client workstation user uses the same installation of the software, optional parameters are consistent, and remote help desk operators are aware of them. This simplifies the analysis that must occur to provide support. Sharing information, such as word processing documents, is easier when everyone is at the same release level and uses the same default setup within the software. Central productivity services such as style sheets and macros can be set up for general use. Most personal productivity products do permit local parameters such as colors, default printers, and so forth to be set locally as well.

Backups of the server can be scheduled and monitored by a trained support person. Backups of client workstations can be scheduled from the server, and data can be stored at the server to facilitate recovery. Tape or optical backup units are typically used for backup; these devices can readily provide support for many users. Placing the server and its backups in a secure location helps prevent theft or accidental destruction of backups. A central location is readily monitored by a support person who ensures that the backup functions are completed. With more organizations looking at multimedia and image technology, large optical storage devices are most appropriately implemented as shared servers.

5.3.3 Fax/Print/Image Services

High-quality printers, workstation-generated faxes, and plotters are natural candidates for support from a shared server. The server can accept input from many clients, queue it according to the priority of the request and handle it when the device is available. Many organizations realize substantial savings by enabling users to generate fax output from their workstations and queue it at a fax server for transmission when the communication costs are lower. Incoming faxes can be queued at the server and transmitted to the appropriate client either on receipt or on request. In concert with workflow management techniques, images can be captured and distributed to the appropriate client workstation from the image server. In the client/server model, work queues are maintained at the server by a supervisor in concert with default algorithms that determine how to distribute the queued work.

Incoming paper mail can be converted to image form in the mail room and sent to the appropriate client through the LAN rather than through interoffice mail. Centralized capture and distribution enable images to be centrally indexed. This index can be maintained by the database services for all authorized users to query. In this way, images are captured once and are available for distribution immediately to all authorized users. Well-defined standards for electronic document management will allow this technology to become fully integrated into the desktop work environment. There are dramatic opportunities for cost savings and improvements in efficiency if this technology is properly implemented and used.

5.3.4 Database Services

Early database servers were actually file servers with a different interface. Products such as dBASE, Clipper, FoxPro, and Paradox execute the database engine primarily on the client machine and use the file services provided by the file server for record access and free space management. These are new and more powerful implementations of the original flat-file models with extracted indexes for direct record access. Currency control is managed by the application program, which issues lock requests and lock checks, and by the database server, which creates a lock table that is interrogated whenever a record access lock check is generated. Because access is at the record level, all records satisfying the primary key must be returned to the client workstation for filtering. There are no facilities to execute procedural code at the server, to execute joins, or to filter rows prior to returning them to the workstation. This lack of capability dramatically increases the likelihood of records being locked when several clients are accessing the same database and increases network traffic when many unnecessary rows are returned to the workstation only to be rejected.

The lack of server execution logic prevents these products from providing automatic partial update backout and recovery after an application, system, or hardware failure. For this reason, systems that operate in this environment require an experienced system support programmer to assist in the recovery after a failure. When the applications are very straightforward and require only a single row to be updated in each interaction, this recovery issue does not arise. However, many client/server applications are required to update more than a single row as part of one logical unit of work.

Client/server database engines such as Sybase, IBM's Database Manager, Ingres, Oracle, and Informix provide support at the server to execute SQL requests issued from the client workstation. The file services are still used for space allocation and basic directory services, but all other services are provided directly by the database server. Relational database management systems are the current technology for data management. Figure 5.1 charts the evolution of database technology from the first computers in the late 1950s to the object-oriented database technologies that are becoming prevalent in the mid-1990s.

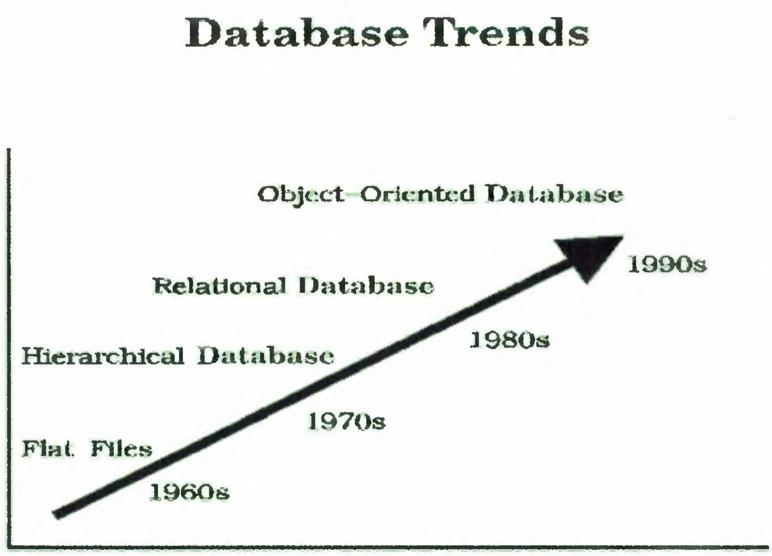

Figure 5.1. Database Trends

5.3.5 Communications Services

Client/server applications require LAN and WAN communication services. Basic LAN services are integral to the NOS. WAN services are provided by various communications server products. This chapter provides a complete discussion of connectivity issues in the client/server model.

5.3.6 Security Services

Client/server applications require similar security services to those provided by host environments. Every user should be required to log in with a user ID and password. If passwords might become visible to unauthorized users, the security server should insist that passwords be changed regularly. The enterprise on the desk implies that a single logon ID and logon sequence is used to gain the authority once to access all information and process for the user has a need and right of access.

New options, such as floppyless workstations with integrated data encryption standard (DES) coprocessors, are available from vendors such as Beaver Computer Company. These products automatically encrypt or decrypt data written or read to disk or a communication line. The encryption and decryption are done using the DES algorithm and the user password. This ensures that no unauthorized user can access stored data or communications data. This type of security is particularly useful for laptop computers participating in client/server applications, because laptops do not operate in surroundings with the same physical security of an office. To be able to access the system from a laptop without properly utilizing an ID number and password would be courting disaster.

5.4 The Network Operating System

The network operating system (NOS) provides the services not available from the client OS.

5.4.1 Novell NetWare

NetWare is a family of LAN products with support for IBM PC-compatible and Apple Macintosh clients, and IBM PC-compatible servers. NetWare is a proprietary NOS in the strict sense that it does not require another OS, such as DOS, Windows, Windows NT, OS/2, Mac System 7, or UNIX to run on a server. A separate Novell product—

Portable NetWare for UNIX—provides server support for leading RISC-based UNIX implementations, IBM PC-compatible systems running Windows NT, OS/2, high-end Apple Macs running Mac System 7, and Digital Equipment Corporation VAXs running VMS.

NetWare provides the premier LAN environment for file and printer resource sharing. It is widely installed as the standard product in many organizations. NetWare is the original LAN NOS for the PC world. As such, it incorporates many of the ease-of-use features required for sharing printers, data, software, and communications lines. Figure 4.2 shows the major components of the NetWare architecture, illustrating client and server functions.

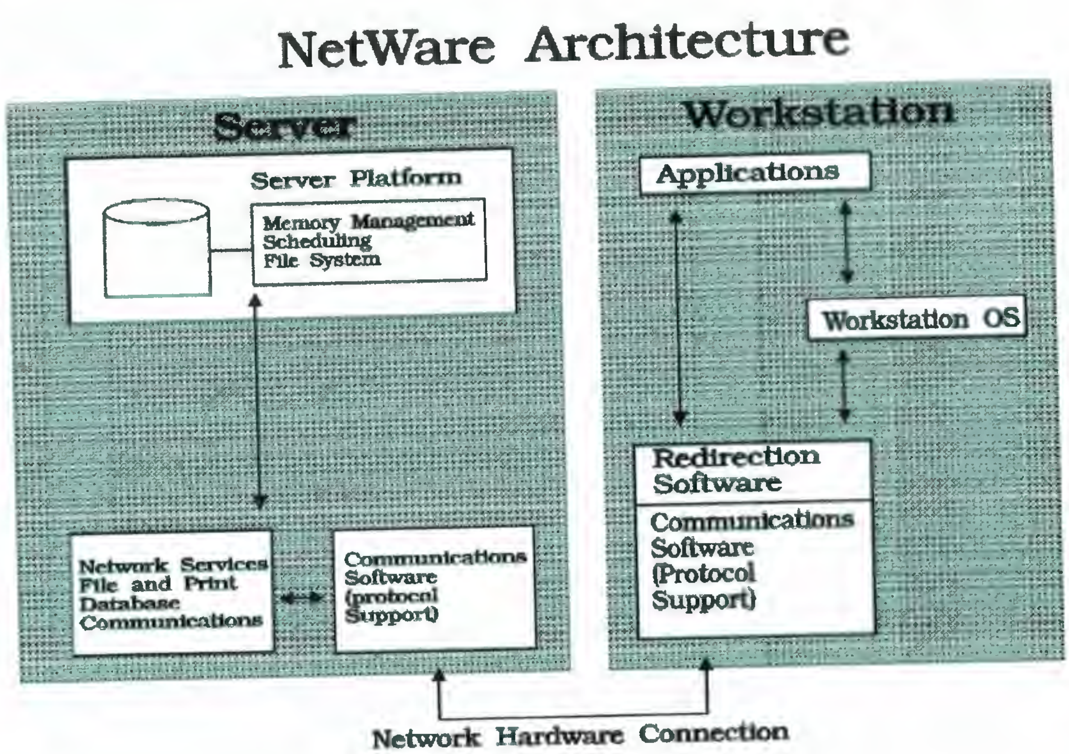

Figure 5.2. Netware Architicture

Novell has committed to move NetWare to an open architecture. Through the use of open protocol technology (OPT), Novell makes NetWare fully network protocol independent. Two standardized interfaces—open datalink interface (ODI) and NetWare Streams—enable other vendors to develop products for the NetWare environment. This facilitates its integration into other platforms

NetWare has benefitted from its high performance and low resource requirements as much as it has from its relative ease of use. This performance has been provided through the use of a proprietary operating system and network protocols. Even though this has given Novell an advantage in performance, it has caused difficulties in the implementation of application and database servers in the Novell LAN. Standard applications cannot run on the server processor, because NetWare does not provide compatible APIs. Instead, NetWare provides a high performance capability called a NetWare Loadable Module (NLM) that enables database servers such as Sybase and Oracle, and communications servers such as Gateway Communications provides, to be linked into the NetWare NOS. In addition, the tailored operating environment does not provide some system features, such as storage protection and multitasking, in the same fundamental way that OS/2 and UNIX do. However, Novell is committed to address these issues by supporting the use of UNIX, OPENVMS, OS/2, and Windows NT as native operating environments.

5.4.2 LAN Manager

LAN Manager and its IBM derivative, LAN Server, are the standard products for use in client/server implementations using OS/2 as the server operating system. LAN Manager/X is the standard product for client/server implementations using UNIX System V as the server operating system. Microsoft released its Advanced Server product with Windows NT in the third quarter of 1993. Then it will be enhanced with support for the Microsoft network management services, currently referred to as "Hermes," and Banyan's Enterprise Network Services (ENS).

Advanced Server is the natural migration path for existing Microsoft LAN Manager and IBM LAN Server customers. Existing LAN Manager/X customers probably did not find Advanced Server an answer to their dreams before 1995.

5.4.3 IBM LAN Server

IBM has entered into an agreement to resell and integrate the Novell NetWare product into environments where both IBM LAN Server and Novell NetWare are required. NetWare provides more functional, easier-to-use, and higher-performance file and print services. In environments where these are the only LAN functions, NetWare is preferable to LAN Manager derivatives. The capability to interconnect to the SNA world makes the IBM product LAN Server attractive to organizations that prefer to run both products. Most large organizations have department workgroups that require only the services that Novell provides well but may use LAN Server for client/server applications using SNA services such as APPN.

5.4.4 Banyan VINES

Banyan VINES provides basic file and print services similar to those of Novell and Lan Manager.

VINES incorporates a facility called StreetTalk that enables every resource in a Banyan enterprise LAN to be addressed by name. VINES also provides intelligent WAN routing within the communications server component. These two features are similar to the OSI Directory Services X.500 protocol.

StreetTalk enables resources to be uniquely identified on the network, making them easier to access and manage. All resources, including file services, users, and printers, are defined as objects. Each object has a StreetTalk name associated with it.

5.4.5 PC Network File Services (NFS)

NFS is the standard file system support for UNIX. PC NFS is available from SunSelect and FTP to provide file services support from a UNIX server to Windows, OS/2, Mac, and UNIX clients.

NFS lets a client mount an NFS host's filing system (or a part of it) as an extension of its own resources. NFS's resource-sharing mechanisms encompass interhost printing. The transactions among NFS systems traditionally ride across TCP/IP and Ethernet, but NFS works with any network that supports 802.3 frames.

SunSelect includes instructions for adding PC-NFS to an existing LAN Manager or Windows for Workgroups network using Network Driver Interface Specification (NDIS) drivers.

5.5 The Server Operating System

Servers provide the platform for application, database, and communication services. There are six operating system platforms that have the greatest potentional and/or are prevalent today: NetWare, OS/2, Windows NT, MVS, VMS, and UNIX.

5.5.1 NetWare

NetWare is used by many organizations, large and small, for the provision of file, printer, and network services. NetWare is a self-contained operating system. It does not require a separate OS (as do Windows NT, OS/2, and UNIX) to run. Novell is taking steps to allow NetWare to run on servers with UNIX. Novell purchased USL and will develop shrink-wrapped products to run under both NetWare and UNIX System V. The products will enable UNIX to simultaneously access information from both a NetWare and a UNIX server.

5.5.2 OS/2

OS/2 is the server platform for Intel products provided by IBM in the System Application Architecture (SAA) model. OS/2 provides the storage protection and preemptive multitasking services needed for the server platform. Several database and many application products have been ported to OS/2. The only network operating systems directly supported with OS/2 are LAN Manager and LAN Server. Novell supports the use of OS/2 servers running on separate processors from the NetWare server. The combination of Novell with an OS/2 database and application servers can provide the necessary environment for a production-quality client/server implementation.

5.5.3 Windows NT

With the release of Windows NT (New Technology) in September of 1993, Microsoft staked its unique position with a server operating system. Microsoft's previous development of OS/2 with IBM did not create the single standard UNIX alternative that was hoped for. NT provides the preemptive multitasking services required for a functional server. It provides excellent support for Windows clients and incorporates the necessary storage protection services required for a reliable server operating system. Its implementation of C2 level security goes well beyond that provided by OS/2 and most UNIX implementations. It took most of 1994 to get the applications and ruggedizing necessary to provide an industrial strength platform for business critical applications. With Microsoft's prestige and marketing muscle, NT installed by many organizations as their server of choice.

5.5.4 MVS

IBM provides MVS as a platform for large applications. Many of the existing application services that organizations have purchased operate on System 370-compatible hardware running MVS. The standard networking environment for many large organizations—SNA—is a component of MVS. IBM prefers to label proprietary systems today under the umbrella of SAA. The objective of SAA is to provide all services on all IBM platforms in a compatible way—the IBM version of the single-system image.

There is a commitment by IBM to provide support for the LAN Server running natively under MVS. This is an attractive option for organizations with large existing investments in MVS applications. The very large data storage capabilities provided by System 370-compatible platforms with MVS make the use of MVS for LAN services attractive to large organizations. MVS provides a powerful database server using DB2 and LU6.2. With broad industry support for LU6.2, requests that include DB2 databases as part of their view can be issued from a client/server application. Products such as Sybase provide high-performance static SQL support, making this implementation viable for high-performance production applications.

5.5.5 OPENVMS

Digital Equipment Corporation provides OPENVMS as its server platform of choice. VMS has a long history in the distributed computing arena and includes many of the features necessary to act as a server in the client/server model. DEC was slow to realize the importance of this technology, and only recently did the company enter the arena as a serious vendor. NetWare supports the use of OPENVMS servers for file services. DEC provides its own server interface using a LAN Manager derivative product called Pathworks.

Pathworks runs native on the VAX and RISC Alpha RXP. This is a particularly attractive configuration because it provides access on the same processor to the application, database, and file services provided by a combination of OPENVMS, NetWare, and LAN Manager. Digital and Microsoft have announced joint agreements to work together to provide a smooth integration of Windows, Windows NT, Pathworks, and OPENVMS. This will greatly facilitate the migration by OPENVMS customers to the client/server model.

5.5.6 UNIX

UNIX is a primary player as a server system in the client/server model. Certainly, the history of UNIX in the distributed computing arena and its open interfaces provide an excellent opportunity for it to be a server of choice. To understand what makes it an open operating system, look at the system's components. UNIX was conceived in the early 1970s by AT&T employees as an operating environment to provide services to software developers who were discouraged by the incompatibility of new computers and the lack of development tools for application development. The original intention of the UNIX architecture was to define a standard set of services to be provided by the UNIX kernel. These services are used by a shell that provides the command-line interface. Functionality is enhanced through the provision of a library of programs. Applications are built up from the program library and custom code. The power and appeal of UNIX lie in the common definition of the kernel and shell and in the large amount of software that has been built and is available. Applications built around these standards can be ported to many different hardware platforms.

The objectives of the original UNIX were very comprehensive and might have been achieved except that the original operating system was developed under the auspices of AT&T. Legal ramifications of the consent decree governing the breakup of the Regional Bell Operating Companies (RBOCs) prevented AT&T from getting into the computer business. As a result, the company had little motivation early on to promote UNIX as a product.

To overcome this, and in an attempt to achieve an implementation of UNIX better suited to the needs of developers, the University of California at Berkeley and other institutions developed better varieties of UNIX. As a result, the original objective of a portable platform was compromised. The new products were surely better, but they were not compatible with each other or the original implementation. Through the mid-1980s, many versions of UNIX that had increasing functionality were released. IBM, of course, entered the fray in 1986 with its own UNIX derivative, AIX. Finally, in 1989, an agreement was reached on the basic UNIX kernel, shell functions, and APIs.

5.6 System Application Architecture (SAA)

SAA is IBM's distributed environment. SAA was defined by IBM in 1986 as an architecture to integrate all IBM computers and operating systems, including MVS, VM/CMS, OS/400, and OS/2-EE. SAA defines standards for a common user access (CUA) method, common programming interfaces (CPI), and a common communication link (APPC).

To support the development of SAA-compliant applications, IBM described SAA frameworks (that somewhat resemble APIs). The first SAA framework is AD/Cycle, the SAA strategy for CASE application development. AD/Cycle is designed to use third-party tools within the IBM SAA hardware and mainframe Repository Manager/MVS data storage facility.

CONCLUSION

Client/server computing takes us a step farther by recognizing that those modules need not all be executed within the same memory space. With this architecture, the calling module becomes the "client" (that which requests a service), and the called module becomes the "server" (that which provides the service).

In the client/server model, the client is primarily a consumer of services provided by one or more server processors. The model provides a clear separation of functions based on the idea of servers acting as service providers responding to requests from clients.

Servers provide application, file, database, print, fax, image, communications, security, systems, and network management services.

Client/server computing provides the capability to use the most cost-effective user interface, data storage, connectivity, and application services. Frequently, client/server products are deployed within the present organization but are not used effectively. The client/server model provides the technological means to use previous investments in concert with current technology options.

REFERENCES

[1] James Chellis, Charles Perkings, Matthew Strebe, "Networking Essentials", SYBEX Publishers 1999.

[2] Charles W. "Understanding TCP/IP" BPB Publishers 1996

[3] Tanebaum Andrew S., Computer Networks, 1996

[4] Microsoft, Networking Essentials, Microsoft Corporation, 1996

[5] Sun Microsystems Inc, Network Models 1999

[6] Cisco Systems Inc. Network Designs 1999

[7] Microsoft Corp. TCP/IP Structures Second Edition 2001

[8] IDG, Network for Dummies 2000

[9] 3Com Corporation, Understanding IP Addressing 1996

[10] Michael Alexander, "The Underground Guide to Computer Networks", Fourth Edition, Addison-Wesley Press, November 1995.

[11] "FYI on Where to Start: A Bibliography of Inter-networking Information," Bowers, K.L.; LaQuey, T.L.; Reynolds, J.K.; Reubicek, K.; Stahl, M.K.; Yuan, A.; 1990

[12] TCP/IP Tutorial," Socolofsky, T.J.; Kale, C.J.; 1991

[13] "Standard for the Transmission of IP Datagrams over NetBIOS Networks," McLaughlin, L.J.; 1989

[14] "DNS Encoding of Network Names and Other Types," Mockapetris, P.V.; 1989

[15] "TFTP Protocol (Revision 2)," Sollins, K.R.; 1981

[16] "MIME (Multipurpose Internet Mail Extensions) Mechanisms for Specifying and Describing the Format of Internet Message Bodies," Borenstein, N.; Freed, N.; 1992

[17] "OSI Connectionless Transport Services on Top of UDP: Version 1," Shue, C.; Haggerty, W.; Dobbins, K.; 1991

LIST OF ABBREVIATIONS

IT	Information Technology
IS	Information Systems
OS	Operating Systems
GUI	Graphical User Interface
SDE	Systems Development Environment
SQL	Structured Query Language
WAN	Wide Area Network
BLOBs	Binary Large Objects
LAN	Local Area Network
PC	Personal Computer
OLE	Object Linking and Embedding
OLTP	Online Transaction Processing
NOS	Network Operating System
CORBA	Communicating Object Request Broker Architecture
EHLLAPI	Extended High Level Language Application Program Interface
IPC	Interprocess Communication
RPC	Remote Procedure Call
NFS	Network File System
BIOS	Basic Input/Output System
IPL	Initial Program Load
E-PROM	Erasable Programmable Read-Only Memory
NIC	Network Interface Card
OLE	Object Linking and Embedding
SDK	Software Development Kit
OMG	Object Management Group
ENS	Enterprise Network Services
SAA	System Application Architecture
OPT	Open Protocol Technology
ODI	Open Datalink Interface
NLM	NetWare Loadable Module

NT	New Technology
CUA	Common User Access
CPI	Common Programming Interfaces
ER	Entity Relationship
IEEE	Institute of Electrical and Electronic Engineers
UTP	Unshielded Twisted Pair
STP	shielded twisted pair
SNMP	Simple Network Management Protocol
CMIP	Common Management Information Protocol
OSI	Open Systems Interconnect
MAC	Media Access Control
LLC	Logical Link Control
ATM	Asynchronous Transfer Mode
ISDN	Integrated Services Digital Network
B-ISDN	Broadband Integrated Services Digital Network
TCP/IP	Transmission Control Protocol/Internet Protocol
DARPA	Defense Advanced Research Projects Agency
ARP	Address Resolution Protocols
FTP	File Transfer Protocol
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
APPC	Application Program-to-Program Communication
TQM	Total Quality Management
CASE	Computer-Aided Software Development
DBMS	Data-Base Management System
OODBMS	Object-Oriented Database Management System
ADDS	Application Driven Database Systems
BPR	Business Process Reengineering
UPS	Uninterruptible Power Supply
DME	Distributed Management Environment
DCE	Distributed Computing Environments
MIS	management information systems
SPA	Software Publishers' Association
OURS	Open User Recommended Solutions

LMF	License Management Facility
SRs	Service Requests
DCS	Distributed Computing System
ARPANET	Advanced Research Projects Agency Network
DOD	Department of Defense
OO	Object Orientation
OOUI	Object Oriented User Interfaces
OMG's	Object Management Group's
ORB	Object Request Broker
TOD	Time of Day
API	Application Programming Interface
OMG	Object Management Group's
OCSI	Object-Oriented Client/Server Internet
HTTP	Hyper Text Transfer Protocol
HTML	Hyper Text Markup Language
FDDI	Fiber Distributed Data Interface