



# Server Classifications

**Shu Zhang**

*Computer Science Corporation, Los Angeles*

**Ming Wang**

*California State University, Los Angeles*

## I. SERVERS

### II. NETWORK OPERATING SYSTEMS FOR SERVERS

## GLOSSARY

**centralized system** Computers are hosted in data centers and users access the host computers through terminals.

**distributed system** Client/server system consisting of many autonomous servers.

**network operating systems** Operating systems designed for and dedicated to network servers.

**servers** Computers on the network that serve multiple users.

A **SERVER** is a networked computer that serves the requests from multiple computers. Servers play very important roles in information systems today, especially in distributed client/server computing environments. Network operating systems have crucial impacts for server performances. In today's competitive market place, all aspects of servers evolve rapidly. Functionalities of servers are reviewed. Servers are classified based on functionality.

## I. SERVERS

A server is a networked computer that serves the requests from multiple computers. Servers play very important roles in information systems, especially in distributed computing environments and client/server architectures. A common example would be a database server serving many users over a network simultaneously. In this case, users might use a desktop PC (personal computer) with a graphical display to com-

## III. SERVER CLASSIFICATION BY FUNCTIONALITY

pose and send a request, and receive and display the result of the request from the server. Since a server might serve hundreds or even thousands of users at the same time, it needs a more powerful CPU (central processing unit), possibly multiple CPUs, and faster and redundant data storage devices, like a RAID (redundant array of inexpensive disks) device. Because a server might store shared critical data, usually it has a tape drive, or it can connect to a networked tape drive device to archive data for ensuring data safety and integrity.

## A. From Hosts to Servers

### 1. Centralized System

The concept of network computer architecture is evolutionary. The origin of the centralized computer dates back to the 1940s. During that time host computers were very large and expensive machines, like the famous MARK I, ENIAC, and EDVAC, etc. Even after computers were commercialized around the 1950s, most computers were "hosted" in highly secured data processing centers, and users accessed the host computers through "dumb" terminals. By the late 1960s, IBM became a dominant vendor of large-scale computers called a "mainframe" host. In the mid-1970s, minicomputers started challenging mainframe computers. In many cases, minicomputers could host applications and perform the same functions as mainframes, but with less cost. Since the host is the center of this system architecture, it is called a centralized system. In the early 1980s, most computers,

no matter whether they were large-scale mainframes or smaller minicomputers, were operated as application hosts, while terminal users had limited access to their hosts; in fact, most terminal users never have any physical access to host computers. Generally speaking, a host is a computer designed for massive parallel processing of large quantities of information connected with terminals utilized by end users. All network services, application executions, and database requests are hosted in this computer, and all data are stored in this host.

Basically, minicomputers and mainframe computers were the de facto standard of enterprise centralized computing systems before PCs entered the professional computing area. Figure 1 shows a typical centralized system.

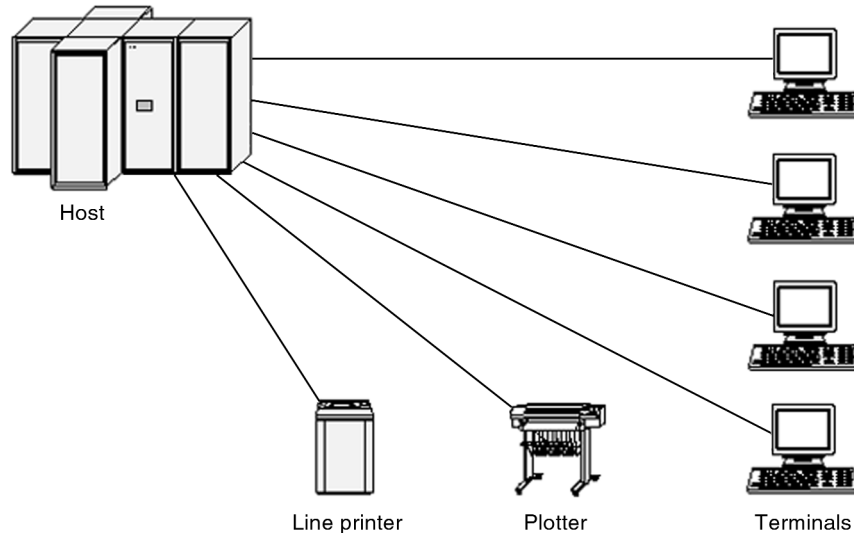
## 2. Distributed System

A distributed system consists of a collection of microcomputers connected to one or more computer servers by a computer network and equipped with network operating system software. Network operating system software enables computers to coordinate their activities and to share the resources of the system—hardware, software, and data. It also can coordinate activities among servers to achieve better over all performance for network tasks. A well-designed distributed system could provide users with a single, integrated computing environment even though the computers are located in geographically separated facilities.

The development of distributed systems followed the emergence of high-speed LAN (local area computer

networks) and WAN (wide area networks) in the early 1980s. Ever since IBM introduced IBM PC into the computing market, the enterprise computing system has changed dramatically, as computers have become more and more affordable to users. The availability of high performance microcomputers, work stations, and server computers has resulted in a major shift towards distributed systems and away from centralized computing systems. People are no longer being tied to high-end and expensive centralized computing environments. The trend has been accelerated by the development of distributed system software such as Oracle Server and SQL server packages, designed to support the development of distributed applications. It is very common now to see a distributed application running collaboratively among some servers. For a well-designed distributed application, any task could be executed by more than one server, so a single faulty server won't bring down the applications. For example, a state wide hospital system's IS (information system) department might have to support tens of medical centers scattered state wide and have tens of departmental information systems (like a UNIX based RIS for Radiology and a Windows NT based Dietary system, etc.) running on Windows 2000/NT servers and different UNIX platforms, while the HIS (Hospital Information System) and CIS (Clinical Information System) are on mainframe computers. Only a distributed computing environment could bring so many autonomous departmental systems together and make them work collaboratively.

Enterprise computing systems differ in significant ways from centralized and distributed systems. Since the late 1980s, a move has occurred from mainframe



**Figure 1** Centralized system.

systems to networked personal computing systems, with network software providing such functionality as shared data storage and electronic mail. Despite the interconnectivity of distributed systems, they remain largely independent: each user runs his or her applications on their own microcomputers, and any interactions between systems are through shared files and mails. Client-server architecture has become common in such a distributed system. A number of client computers are configured as a sort of ring around the server, which provides database functionality and file management. Again, the client computers interact indirectly through shared servers. Figure 2 shows a typical distributed system.

**B. Servers for Client/Server Computing**

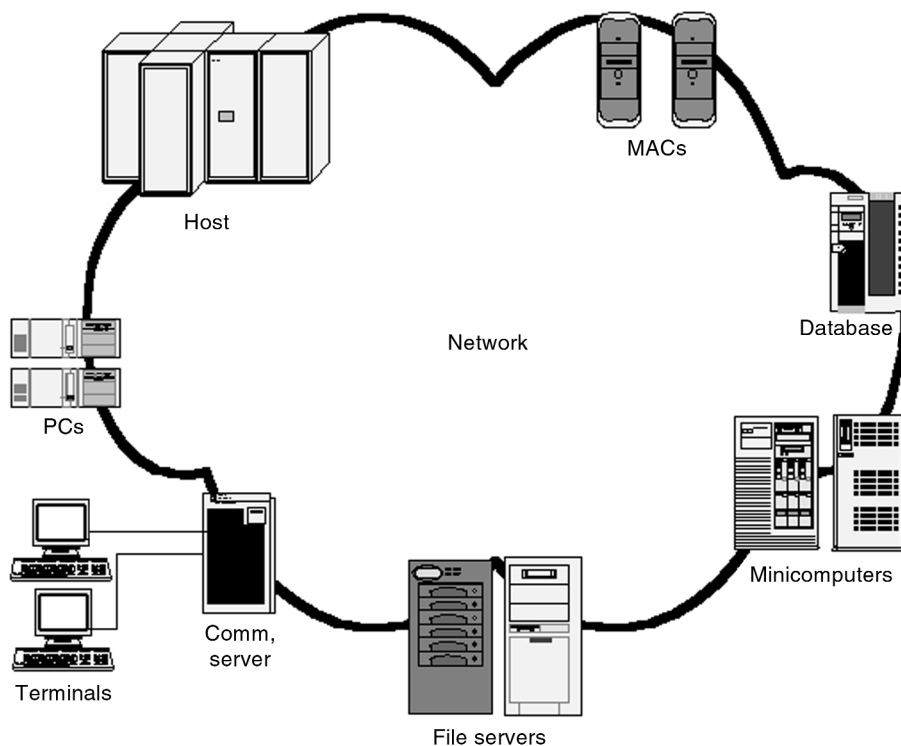
Client/server computing is a phenomenon that has developed in the past decade. The inexpensive and powerful PC took over previously “dumb” terminal-oriented enterprise desktops as quickly as people could think. To use the excessive computing capacity of desktop PCs or workstations, many organizations began downloading data from those enterprise host computers for local manipulation at the user’s fin-

gertips. In this client/server model, the definition of the server will continue to include what those traditional hosts and servers have, but people can envision the placement of network and application services on many different operating system platforms.

**1. Servers for “Thin” Clients**

**a. “FAT” CLIENTS VERSUS “THIN” CLIENTS**

Client-server computing architecture refers to the way in which software components interact to form a system. As the name suggests there is a client process, which requires some resource, and a server, which provides the resource. There is no requirement that the client and server must reside on the same machine. In practice, it is quite common to place a server at one site in a local area network and the clients at the other sites. Clients can be categorized into two types: fat clients and thin clients. A “fat” client requires considerable resources on the client’s computer to run effectively. This includes disk space, RAM, and CPU power. It has significant client-side administration overhead. A “thin” client requires fewer resources on the client’s computer and is responsible for only simple logic processing, such as input validation. It has less expensive hardware because the client is thin.



**Figure 2** Distributed system.

### b. THIN CLIENTS

In the thin-client/server computing model, applications execute 100% on the server. The client computers are just ordinary desktop PCs running one or more terminal programs to access servers over LAN or WAN. The thin-client/server involves connecting thin-client software or a thin-client hardware device with the server side using a highly efficient network protocol. The thin-client/server architecture enables 100% server-based processing, management, deployment, and support for mission-critical productivity, Web-based, or other custom applications across any type of connection to any type of client hardware, regardless of platforms. The client hardware can include desktop PCs, network computers, handheld computers, wireless PDA, and Windows-CE devices.

### c. ADVANTAGES OF “THIN” CLIENTS

Though it appears to be a very primitive approach for client/server computing since it simply replaces one or more dumb terminals with a desktop PC, the thin-client/server model has regained some ground recently because of the TCO (total cost of ownership) consideration for IS operation and the appearance of lower-powered client devices like a PDA (personal data assistant) for palmtop computing. For the thin-client/server computing model, there is no need to purchase or upgrade client hardware—just run the latest software on servers instead. The client can let it comfortably evolve, leveraging existing hardware, operating systems, software, networks, and standards. Thin-client/server computing extends the life of the organization's existing computing infrastructure considerably and might reduce TCO if it is planned and implemented carefully with well-scaled servers.

## 2. Servers for Multiple Tiers of Client/Server

In a typical client/server based application, the client process and server process can be on the same computer, or distributed in two or more computers. A single-tier client/server application consists of a single layer that supports the user interface, the business rules, and the data manipulation processes all on one computer. This kind of application is rarely used today because it will not take advantage of the distributed computing environment.

### a. TWO-TIER CLIENT/SERVER ARCHITECTURE

The two-tier client/server structure is the simplest client/server structure that is still in use for many ap-

plications today. In a two-tier application, the business rules and user interface remain as part of the client application on the client's computers. The traditional two-tier client/server architecture provides a basic separation of tasks. The client (tier 1) is primarily responsible for the presentation of data to the user, and the server (tier 2) is primarily responsible for supplying data services to the client. The client handles user interface actions and the main business application logic. The server provides server side validation, data retrieval, and data manipulation. This separate application could be a RDBMS (relational database management system), which is functioning as a data storage/retrieval system for the application.

### b. THREE-TIER CLIENT/SERVER ARCHITECTURE

The need for enterprise scalability challenged the traditional two-tier client/server architecture. In the mid-1990s, as applications became more complex and potentially could be deployed to hundreds or thousands of end-users, the client side presented the problems that prevented true scalability. Because two-tier client/server applications are not optimized for WAN connections, response time is often unacceptable for remote users. Application upgrades require software and often hardware upgrades on all client PCs, resulting in potential version control problems.

By 1995, three new layers of client/server architecture were proposed, each running on a different platform:

1. Tier one is the user interface layer, which runs on the end-user's computer.
2. Tier two is the business logic and data processing layer. This middle tier runs on a server and is often called the application server. This added middle layer is called an application server.
3. Tier three is the data storage system, which stores the data required by the middle tier. This tier may run on a separate server called the database server. This third layer is called the back-end server.

In a three-tier application, the user interface processes remain on the client's computers, but the business rules processes are resided and executed on the application middle layer between the client's computer and the computer which hosts the data storage/retrieval system. One application server is designed to serve multiple clients. In this type of application, the client would never access the data storage system directly.

**c. ADVANTAGES OF THREE-TIER CLIENT/SERVER ARCHITECTURE**

Since there are three physically separated layers for the application, the added modularity makes it easier to modify or replace one tier without affecting the other tiers. Application maintenance is centralized with the transfer of the business logic for many end-users to a single application server. This eliminates the concerns of software distribution that are problematic in the traditional two-tier client/server architecture. An additional advantage is that the three-tier architecture maps quite naturally to the Web environment, with a Web browser acting as the thin client, a Web server acting as the application server, and a data/database system server as the back-end.

**d. MULTITIER CLIENT/SERVER ARCHITECTURE**

The three-tier architecture can be extended to *n*-tiers, with additional tiers added to provide more flexibility and scalability. Some distributed computing systems have more than three layers, but the basic rules are the same as those for three-tier applications. For example, the middle tier of the three-tier architecture could be split into two, with one tier for the Web server and another for the application server. More than one server used in the second and third layers will usually increase overall application effi-

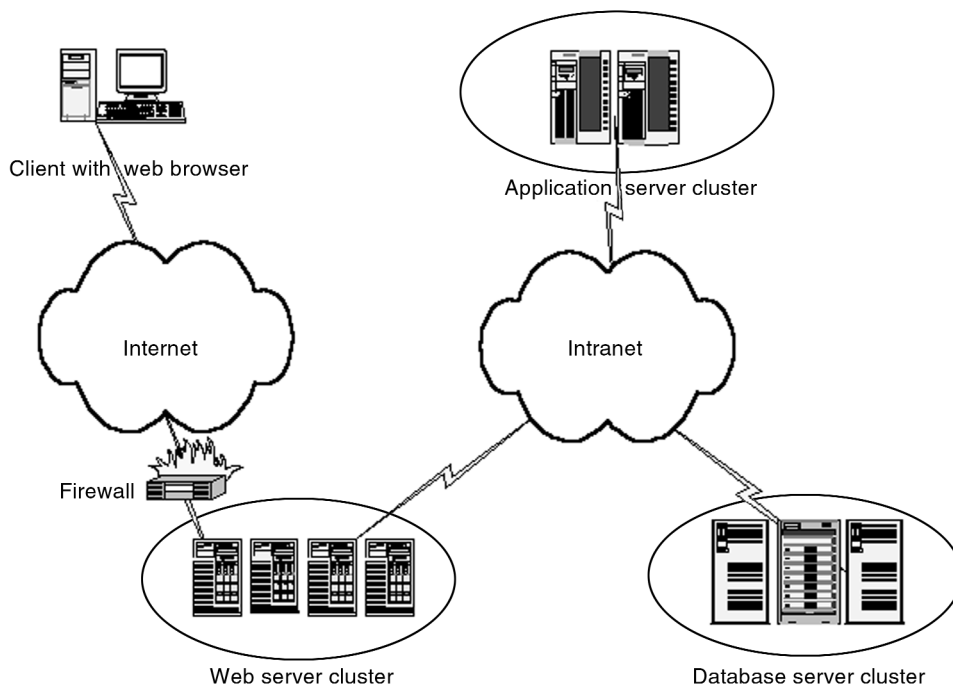
ciency as needed. Figure 3 shows a typical multiple tier client/server architecture.

**II. NETWORK OPERATING SYSTEMS FOR SERVERS**

Servers works best when they are configured with an NOS (network operating system) that supports shared memory, preemptive multitasking, and application isolation. To handle massive data processing load and multiple applications efficiently, servers need to use an OS that provides a multithread environment and scalability for multiple CPUs. Since servers are tied to networks to provide a variety of network services, some operating systems are dedicated to the servers and are called an NOS (network operating system). They are introduced in the following sections.

**A. Novell NetWare**

For a number of years, NetWare was synonymous with networking. NetWare, an exceptionally fast file-and-print server, allows desktop clients to share files, data, and printers. From its inception, Novell's redirector implementation strategy reflected the need for speed.



**Figure 3** A multitier client/server diagram.

The NetWare redirector was designed to intercept and redirect system calls for disk access. However, Windows 2000/NT operating systems are giving NetWare some stiff competition because of their native API functionalities with desktop operating systems such as Windows 2000 Professional and Windows XP/ME/98/95 systems.

## B. Microsoft Windows 2000/NT

The Microsoft Windows 2000/NT Server is a very commonly used 32-bit operating system. The Windows 2000 Server kernel has evolved directly from its predecessor Windows NT Server. The Windows 2000/NT Server leverages the Windows GUI into the world of networking. The Windows 2000/NT Server uses the same user interfaces as Windows XP/ME/98/95 with true multithreading, built-in networking, security, and memory protection. The Windows 2000 Server has three editions to address different market requirements, namely Standard Server, Advanced Server, and Data Center Server. The Windows 2000 Server provides exceptional portability, scalability, and distributed computing capacities with built-in DCOM components, as well as POSIX compliance and C2 security.

## C. UNIX Families

UNIX is an interactive time-sharing operating system that was invented in the 1960s. UNIX has powerful multiuser capabilities because of its architecture and features. Among the many features of UNIX are 32-bit and 64-bit architectures, preemptive multitasking, robustness, and networking capability. There are many UNIX vendors such as Solaris from Sun Microsystems, AIX from IBM, and UX from HP. In comparison to other network operating systems such as NetWare and Windows 2000/NT, UNIX is quite expensive to maintain and support with the exception of Linux. Though UNIX was the most widely used multiuser general-purpose operating system in the enterprise level computing world in the 1990s, it has yielded more and more to the Windows 2000/NT operating system.

## D. Some Proprietary Systems

### 1. IBM's MVS

Though IBM mainframe computers have acted as a traditional application host for many years, there is a

commitment made by IBM to provide support for networking services running under its MVS (multiple virtual storage) environment. This is an attractive option for organizations with a large investment in MVS applications under IBM System 370-compatible mainframe computers. MVS provides a powerful database server using DB2. With MVS networking services, users can issue SQL requests from a client/server application to the target databases as part of the view and make this implementation viable for high-performance tasks. IBM also makes a similar commitment to its middle range computers, the AS/400 systems under the OS/400 operating system.

### 2. DEC's VMS

DEC (Digital Equipment Corp.), now a division of Compaq, provides VMS as its server OS platform. VMS has a long history in the distributed computing environment arena and includes many of the features necessary to act as a server in client/server models. VMS support for DBMS products such as Oracle, SyBase, RDB, and Ingres allow this platform to execute effectively as database servers for client/server applications. DEC has made its VMS, Digital-UNIX, and PathWorks products well integrated to provide a networking environment that covers its own middle range systems, RISC (reduced instruction set computing) based minicomputers, and CISC (complex instruction set computing) based Intel CPU servers. However, after Compaq Computers acquired DEC, that commitment may have some changes to be in alliance with the Compaq strategies.

There are many other proprietary network operating systems, and most of them have their own domains for either historical reasons or very special needs. However, people will see more and more realignment and strategy changes among those proprietary network operating system vendors in the rapid advancement of high-tech industries and very competitive market place of today.

## III. SERVER CLASSIFICATION BY FUNCTIONALITY

### A. File Servers

File servers are network computers that store programs and data files for shared access and that may control access to remote network printers, providing spooling spaces for printer queues. File servers also provide record level data services to nondatabase ap-

plications, acting like a remote disk drive. File servers manage allocation of shared storage space, and either deny or accept a user request for a file according to the user's privilege. File servers have file catalog functions to support file operation and directory structure. The maximum length of a file name usually ranges up to 256 characters, depending on the network operating system. A stored program file is typically loaded from a file server for execution on a client or an application server. The difference between a file server and an application server is that the file server stores the programs and data, while the application server runs the programs and processes the data.

A file server supports one or more file sharing protocols. If a client wants to have access to multiple file servers running different protocols, either the client supports the protocol of each server or the server supports the protocol of each client. Client software that adds this capability is very common and allows interoperability between Windows, Macintosh, NetWare, UNIX, and other platforms. The most commonly used protocols for file servers are TCP/IP, SPS/IPX, and NETBEUI. Usually file servers use a hierarchical file system that stores data in a top-to-bottom organization structure. All internal access to the data starts from the top to the bottom.

## 1. Novell NetWare Family

The NetWare family of network operating systems supports DOS, Windows, OS/2, and Macintosh clients. UNIX client support is available from third parties. In the early 1990s, NetWare was the most popular LAN operating system. Until NetWare 5, which natively supports TCP/IP and Java, NetWare always used its own proprietary protocols (IPX/SPX/NCP). Its hard disks are in the NetWare format. Although DOS and Windows applications reside on the server, they cannot be run on the server unless they have been compiled into NetWare Loadable Modules (NLM) using Novell libraries.

Novell Directory Services (NDS) is Novell's flagship directory service that is included in NetWare beginning with Version 4. It is also available for Windows NT and Solaris. NDS maintains a hierarchical database of information about the network resources within a global enterprise, including networks, users, subgroups, servers, volumes, and printers. Unlike the bindery, which was the directory service in NetWare 3.x, NDS users log onto the network as a whole, not a specific server, and NDS determines their access rights. NDS is based on the X.500 directory standard and is LDAP compliant. Novell provides the NDS

source code free of charge to developers that wish to integrate it into their products. In NDS, every network resource is called an "object," and each object contains properties (fields). For example, a user object would contain login ID, password, name, address, telephone, and node address.

NetWare 4 (1993) was the first NetWare version to use the much-acclaimed Novell Directory Services (NDS), which provides directory services for a global enterprise. NetWare 5 (1998) fully supported TCP/IP and Java and includes a kernel that natively supports symmetric multiprocessing (SMP). Introduced in 2001, NetWare 6 adds disk pooling and Novell Internet Printing (NIP), which enables documents to be printed over the Internet.

## 2. Windows 2000/NT File Servers

Windows 2000 is known as Microsoft "WIN2K" or "W2K." It is a major upgrade to Windows NT 4. Launched in February 2000, the Windows 2000 server came in one client version and three server versions. It added support for Plug and Play, which made adding peripherals considerably easier than in NT 4. Windows 2000 uses Active Directory, which replaces NT's domain system and makes network administration simpler. This is a major redesign of the directory structure for companies. Windows 2000 is more stable than Windows NT and is designed to eliminate erroneous replacement of DLLs when applications are installed.

Windows 2000 Advanced Server is a replacement for the Windows NT 4.0 Server Enterprise Edition. It supports clustering and automatic fail-over in the event of a system failure. Windows 2000 Data Center Server supports more advanced clustering and is targeting enterprise data service offering. Windows 2000 supports multiprocessing systems (SMP), adds extensive security and administrative features, and offers a dual boot capability. Designed for enterprise use, each application can access 2GB of virtual memory. With terminal service and IIS service options, Windows 2000 Server makes application deployment much easier and straightforward.

## 3. NFS File Servers

### a. NFS

Network File System (NFS), developed by Sun Microsystems, is a file sharing protocol and a de facto UNIX network standard. It is widely known as a "distributed file system." Almost all NIX vendors implement NFS as part of their offerings, as well as Network

Information Service (NIS) developed by Sun. The advantages of such centralization include allowing unified user access to network resources and making the distributed file system more secure and manageable.

#### b. WEBNFS

WebNFS is derived from NFS. As a Web version of the NFS distributed file system from SunSoft, WebNFS enabled Web server browsers to access Web pages as much as 10 times faster than the standard HTTP protocol. Unlike HTTP, which drops the connection after each tiny file is downloaded, WebNFS downloads multiple files with a single connection.

### 4. Network Attached Storage (NAS) Systems

Network Attached Storage (NAS) systems are another file service device. The NAS is connected to the LAN just like a file server. Rather than containing a full-blown OS, it typically uses a slim microkernel specialized for handling only I/O requests such as NFS (UNIX), CIFS/SMB (Windows 2000/NT), and NCP (NetWare). Adding or removing a NAS system is like adding or removing any network node. For example, it doesn't get much simpler than the Meridian Data Snap server—containing only an on/off switch and an ethernet port. It provides an instant storage boost by simply plugging it into the network switch or hub port. However, the NAS is subject to the variable behavior and overhead of the network, which makes NAS less desirable in many cases.

## B. Database Servers

### 1. Description

Database servers are networked computers on a network dedicated to database storage and data retrieval from the database. The database server is a key component in a client/server computing environment. It holds the database management system (DBMS) and the databases. In the database context, the client manages the user interface and application logic, acting as a sophisticated workstation on which to run database applications. The client takes the user's request, checks the syntax, and generates database requests in SQL or another database language. It then transmits the message to the server, waits for a response, and formats the response to the end-user. The server accepts and processes the database requests, then transmits the results back to the client. The process in-

volves checking authorization, ensuring integrity maintaining the system catalog, and performing query and update process.

### 2. Advantages of Using Database Servers

- (i) It enables wider access to existing databases
- (ii) It increases performance. If the clients and server reside on different computers then different CPUs can process applications in parallel.
- (iii) Hardware cost may be reduced. It is only the server that requires storage for the database and processing power sufficient to store and manage the database.
- (iv) It increases database consistency. The server can handle integrity checks, so that constraints need to be defined and validated only by the server.

### 3. Oracle Server

The Oracle database was the first DBMS to be ported to a wide variety of platforms. Oracle offers a variety of application development tools and is a major promoter of the network computer. Its Network Computer subsidiary defines the specifications for a compliant platform. ORACLE 9 is the newest relational database management system (RDBMS) with extended object-relational database features. The current version of Oracle's flagship product includes such features as replication and high availability. Oracle 9 runs on more than 80 platforms.

The new ORACLE enterprise server package includes many Internet enhancements. Noteworthy features are increased performance and support for XML and JDBC applications. JVM (Java Virtual Machine) is built into the RDBMS so that triggers and stored procedures can be written and executed in Java directly. It enables Internet developers to write applications and database procedures in the Java language. In addition, the JVM can also execute Enterprise JavaBeans (EJBs), turning the DBMS into an application server.

### 4. Microsoft SQL Server

SQL Server is a relational DBMS from Microsoft that runs on Windows 2000/NT servers. It is Microsoft's high-end client/server database and a key component in its BackOffice suite of server products. SQL Server was originally developed by Sybase. After Microsoft bought it in 1992, the company began developing its own version. Today, Microsoft SQL Server and Sybase

SQL Server are independent products with some compatibility. SQL Server 2000 is the newest version release in 2000. Along with Microsoft's other BackOffice products, like Microsoft Transaction Server (MTS), SQL Server is targeting the enterprise computing arena with almost all the features that its competitors have, plus native Windows interface. However, Microsoft SQL Server only operates under Windows 2000/NT systems.

## 5. DB2 Servers

DB2 is a RDBMS from IBM that was originally developed for its mainframes. It is IBM's major database product with the fully featured SQL language. Known for its industrial strength reliability, IBM has made DB2 available for all of its own platforms, including OS/2, OS/400, AIX-RS/6000, and OS/390, as well as for Solaris on Sun systems and HP-UX on HP 9000 workstations and servers.

DB2 Universal DataBase (UDB) is an enhanced and very popular version of DB2 that combines relational and object-relational database technology as well as various query optimization techniques for parallel processing. Also geared for electronic commerce, DB2 UDB provides graphical administration, Java, and JDBC support. DB2 UDB runs on mainframes, Windows NT, and various versions of UNIX.

## 6. Informix Servers

Informix is a relational database management system originally from Informix Software. It was acquired by IBM in 2001. Informix runs on most UNIX platforms, including SCO UNIX for x86 machines and NetWare. Development tools from Informix include INFORMIX-4GL, a fourth-generation language, and INFORMIX-New Era, a client/server development system for Windows clients that supports INFORMIX and non-INFORMIX databases. It is expected that either Informix will be tightly integrated with the IBM DB2 production line, or migrated into DB2 in the near future.

## 7. Sybase SQL Server

Sybase SQL Server is a relational database management system from Sybase that runs on OS/2, Windows NT, NetWare, VAX, and UNIX servers. It was designed for the client/server environment and is accessed by applications using SQL or via Sybase's own QBE and decision-support utilities. After Sybase acquired Powersoft, a maker of the PowerBuilder ap-

plication development software in 1995, and NEON Software, an Enterprise System Integration service vendor in 2001, it offered a variety of application development tools, compilers, middleware, and database and data warehousing products for its Sybase SQL Server, as well as for other RDBMS.

## C. Application Servers

Application servers are network computers that store and run an application for client computers. Application servers, whatever their function, occupy a large chunk of computing territory between database servers and the end user. Most broadly, this is called "middleware" which tells us something about what application servers do. First and foremost, application servers connect database information (usually coming from a database server) and the end-user or client program (often running in a Web browser). There are many reasons for having an intermediate player in this connection, including a desire to decrease the size and complexity of client programs, the need to cache and control the data flow for better performance, and a requirement to provide security for both data and user traffic.

In the early days of application servers, it was realized that applications themselves, the programs people were using to get work done, were becoming bigger and more complex, both to write and maintain. At the same time, pressure was increasing for applications to share more of their data and sometimes functionality. More applications were either located on a network or used networks extensively. It seemed logical to have some kind of program residing on the network that would help share application capabilities in an organized and efficient way, and make it easier to write, manage, and maintain the applications.

These designated application servers first appeared in client/server computing and on LANs. At first, they were often associated with "tiered" applications, when people described the functionality of applications as two-tiered (database and client program), three-tiered (database, client program, and application service), or *n*-tiered (all of the above plus whatever). This was (and still is) a complex model of application development, and it resisted wide-scale implementation. Then along came the Internet application. The Internet application is automatically three-tiered (usually consisting of a database, client program, and Web servers). Managing data along with application functionality suddenly became not only an esoteric exercise in better program design, but also a downright

necessity. This vaulted the application server from obscurity to the top of a pedestal, and literally scores of companies jumped in to develop products. Not surprisingly, people do not consider or think of the role of the application server in the same way, so application servers have different roles and functionalities as different companies build from their requirements and understanding. Scalability is a good example. Some companies might want an application server that simply helps them organize their applications for the Web, give them better control over the business logic they contain, and make it easier to monitor and secure the data. They do not need thousands of servers. Other companies, especially big ones, do need to manage thousands of servers. For them, the scalability of an application server is crucial. So some application servers feature scalability, others feature other things, and some try to do everything. Also, application server products belong to a variety of programming domains: some are Java based, while others are written by C++; one might support CORBA, and another could be implemented through Microsoft DCOM. It is relatively important to consider these servers in light of an organization's programming preferences.

## D. Web Servers

### 1. Description

A Web server is a network server that manages access to files, folders, and other resources over the Internet or a local Intranet via the HTTP (hypertext transfer protocol). HTTP is a client/server protocol that defines how clients and servers communicate. It transmits information between servers and clients. In addition, Web servers possess unique Web networking characteristics. They handle permissions, execute programs, keep track of directories and files, and communicate with the client computer. These client computers make requests for files and actions from server computers using HTTP. Web servers serve content over the Internet using HTTP. The Web server accepts HTTP requests from browsers like Internet Explorer and Netscape Communicator and returns the appropriate hypertext markup language (HTML) documents, images, and Java Applets. A number of server-sided technologies can be used to increase the power of the server beyond its ability to deliver standard HTML pages. These include CGI scripts, Active Server Page (ASP), Java Server Page (JSP), and Java Servlet.

### 2. Development

Initially, Web servers served static content to a Web browser at a very basic level, and many new Web servers are still starting in the same way. This means that the Web server receives a request for a Web page like `http://www.anyname.com/` and maps that uniform resource locator (URL) to a local default starting page file on the host server. In most cases, the file is "index.html" or "default.html" as configured by Web masters. The server then loads this file from server storage and serves it out across the network to the user's Web browser. The browser and server use HTTP for this entire exchange. Users could access any file by name under that URL directly if the file on the server is accessible for the users, for example, `http://www.anyname.com/anydir/anypage.html`. Perhaps the most important expansion of this was the concept of dynamic content, in which Web pages generated by CGI, ASP, JSP, and Java Servlet could respond to a user's input, whether directly or indirectly. A Web server could run those programs locally and transmit their output through the Web server to the user's Web browser that is requesting the dynamic content.

The second important advance, and the one that makes e-commerce possible, was the introduction of hypertext transmission protocol, secure (HTTPS). This protocol allows secure communication to go on between the browser and Web server. This means that it is safe for the user and server to transmit sensitive data to each other across what might be considered an insecure network using HTTPS.

### 3. Examples

Among the hundreds of Web servers, Microsoft Personal Server (MPS), Microsoft Internet Information Server (IIS), Apache, and Jigsaw remain popular so far. Microsoft Internet Information Server (IIS) is the one most commonly used by commercial Web sites, because of its higher manageability and flexibility. However, it is also very common to see some small or nonprofit sites running a variety of free distributed Web servers.

## E. Mail Servers

Mail servers are network servers that provide electronic mail services for Internet users. Web servers may be the most important and ubiquitous servers on

the Internet. Mail servers rank a close second because e-mail is considered the most important service provided by the Internet. For Internet mail servers, a very important factor is the support of standards. The two major protocols are SMTP (simple mail transfer protocol) for outgoing mail and POP3 (post office protocol) for incoming mail. A more recent protocol is IMAP4 (Internet messaging access protocol). IMAP offers a number of important features, including user management of mail on the server. Other Internet protocols include ESMTP (extended simple mail transfer protocol), APOP (authenticated post office protocol), MIME (multipurpose Internet mail extensions), and Ph (directory access protocol). Many mail servers are also adding S/MIME, SSL, or RSA (Rivest, Shamir, Adleman Algorithm) support for message encryption, and LDAP (lightweight directory access protocol) support to access operating system directory information about mail users. In general, the more standards a server supports, the better mail server it is.

The highly publicized viruses that attack through e-mail clients have put the spotlight on e-mail as a vulnerable point in an enterprise's firewall. In response, mail server vendors, along with major client vendors, have begun producing add-ons and built-in features that will help to scan mail, segregate questionable messages, and deal with viruses. As might be expected, this is a rapidly expanding element of e-mail servers and should be considered important when comparing products.

E-mail servers rank high in difficulty to install and manage. One reason is that they are tied to the inherently variable source-Internet connection and mail traffic. Another reason is that they require constant attention to user lists, user rights, and message storage. Thus, it is important to look for servers that provide ease of administration and ease of use. A GUI (graphical user interface) is a nice feature for user-friendly e-mail clients, but it is not necessarily the route to easy administration. This type of interface is often related to a platform (operating system), and it is important to keep in mind that many products are intended to run on a single platform (e.g., Unix or Windows).

## F. Proxy Servers

Proxy servers provide a gateway for applications and filter traffic between an organization's network and the Internet for security purposes. Although proxy servers have been around for a long time (since the

early days of the WAN), the Internet has transformed them. They used to be found in only the large corporations. They are now found as a critical component in small organizations with an Internet connection. They will be found running in some private homes in the future. As a key part of a network firewall system, proxy servers keep unwanted intruders, hackers, and viruses away from internal networks, while they allow approved users to access the Internet resources. Schematically, a proxy server sits between a client program (typically a Web browser) and some external server (typically another server on the Web). The proxy server can monitor and intercept any and all requests being sent to the external server or requests that come in from the Internet connection. This positioning gives the proxy server three key capabilities: filtering requests, improving performance, and sharing connections. Filtering requests is the security function and the original reason for having a proxy server. Proxy servers can inspect all traffic (in and out) over an Internet connection and determine if there is anything that should be denied transmission, reception, or access. Since this filtering cuts both ways, a proxy server can be used to keep users out of particular Web sites (by monitoring for specific URLs) or restrict unauthorized access to the internal network by authenticating users. Before a connection is made, the server can ask the user to log in. To a Web user this makes every site look like it requires a log in. Because proxy servers are handling all communications, they can log everything the user does. For HTTP (Web) proxies this includes logging every URL. For FTP proxies this includes every downloaded file. A proxy can also examine the content of transmissions for "inappropriate" words or scan for viruses, although this may impose serious overhead on performance.

It should be obvious that part and parcel of any proxy server system is the need to create policies for using it to filter Internet traffic. Few decisions can be more politically charged within an enterprise than who is allowed to do what on the Internet, and many privacy-related issues go with such decisions. It is important that the proxy server provide adequate ways not only to incorporate the rules for filtering, but also to help organize and document those rules.

The other aspect of proxy servers, improving performance, is far less controversial. This capability is usually called proxy server caching. In simplest terms, the proxy server analyzes user requests and determines which, if any, should have the content stored temporarily for immediate access. A typical corporate example would be a company's home page located on

a remote server. Many employees may visit this page several times a day. Since this page is requested repeatedly, the proxy server would cache it for immediate delivery to the Web browser. Cache management is a big part of many proxy servers. It is important to consider how easily the cache can be tuned and for whom it provides the most benefit.

Some proxy servers, particularly those targeted at small business, provide a means for sharing a single Internet connection among a number of workstations. While this has practical limits in performance, it can still be a very effective and inexpensive way to provide Internet services, such as e-mail, throughout an office. There are many proxy server vendors, especially OS vendors like Microsoft, Sun Microsystems, and IBM, etc. Some proxy server vendors are targeting small businesses or the home computing market.

## G. Fax Servers

A fax server is a network computer that provides a bank of fax/modems to manage the receipt and delivery of faxes. In fact, many of the functions of sophisticated fax servers involve managing limited incoming and outgoing telephone resources, which would be completely replaced by e-mail for document exchanges. In a sense, fax servers are a bridge between the old way of doing business and the new. But as long as documents continue to stampede across this bridge, the fax server market continues to breathe life vigorously.

In many respects, a fax server is similar to an e-mail server. Both types of servers are bridges between outgoing and incoming messages. Both must route incoming messages to a destination. In the case of e-mail servers, this destination is always an inbox for a particular user. Fax servers for small, single-user environments often assume that the receiving computer itself is the sole destination, so the "inbox model" does not apply. On the other hand, fax servers designed for corporate environments indeed parallel the e-mail server model, delivering incoming faxes to particular destinations assigned to individual users.

A well-designed fax server may offer extra conveniences for handling incoming faxes, such as direct-to-printer output. It may also provide outgoing specialties, such as scheduled broadcasts of a document to many recipients, and automated outgoing faxes triggered by incoming requests. Corporate fax servers must also juggle numerous outgoing faxes, possibly queued up by a number of different users. How well fax server software can effectively manage a limited

number of phone lines, so as to schedule both outgoing and incoming faxes without conflict, is a major selling point for enterprise level fax servers.

Sophisticated fax servers also feature strong integration with electronic messaging systems, like Microsoft Exchange and Lotus Notes. Such features enable the fax server to become a seamless bridge between electronic documents and the anachronistic world of fax documents. Fax servers range widely in capabilities, scaled to different environments, from the home or small office needs addressed by WinFAX and RelayFax, to enterprise-level products, like RightFAX, FAXport, and Faxination, etc.

## H. Management Servers

Management servers are network servers that run some dedicated network management software to monitor and manage server farms for enterprises. Almost all network management software supports the Simple Network Management Protocol (SNMP). Network management software manages computer systems in an enterprise, which may include any or all of the following functions: software distribution and upgrading, user profile management, version control, backup and recovery, printer spooling, job scheduling, virus protection, and performance and capacity planning. For example, ZENworks is a family of directory enabled system management products from Novell. ZENworks supports Windows clients, NetWare, and Windows 2000/NT servers. With ZENworks, system administration features allow users to log in from any PC and obtain their custom desktop configuration under Novell's popular NDS directory service. OpenView is Network management software from HP. Some third-party products that run under OpenView support SNA and DECnet network management protocols. OpenView supports almost all platforms either in server mode or in client agent mode and has been widely used as an enterprise-wide network management solution.

NetView is a IBM SNA network management software that provides centralized monitoring and control for SNA, non-SNA, and non-IBM devices. NetView/PC interconnects NetView with Token Ring LANs, Rolm CBXs, and non-IBM modems, while maintaining control in the host.

Systems Management Server is from Microsoft for Windows 2000/NT Server. It requires a Microsoft SQL Server database and is used to distribute software, monitor and analyze network usage, and perform various Windows network administration tasks.

Other network management server products are Sun Microsystems's SunNet Manager and Novell's NMS, which work best for their native platforms, and also have extended features to other platforms.

## I. Communication Servers

Communication servers are network servers that provide network communications for remote access users either through direct dial up or through an ISP (Internet service provider) through authentication protocols. In some cases, communication servers are just regular file servers with a modem, while a dedicated server has a modem pool consisting of internal or external modems.

Windows Terminal Service is an option for Windows 2000 Advanced Edition and Data Center Edition and can be used as a communication server. Windows Terminal Service enables an application to be run simultaneously by multiple users at different remote Windows PCs. Windows Terminal Service turns a Windows 2000 server into a centralized, time-shared computer like the good old days of mainframes and dumb terminals.

Shiva's LANRover is another widely used proprietary communication software for servers, which provides secured connectivity for remote users through

either analog modems or ISDN connections. Also, some remote control software packages provide server versions, which also could be used as a communication server for remote users to access geographically separated resources.

## SEE ALSO THE FOLLOWING ARTICLES

Electronic Data Interchange • End-User Computing Tools • Operating Systems

## BIBLIOGRAPHY

- Connolly, T., *et al.* (2002). *Database systems: A practice approach to design, implementation and management*, 3rd ed. Reading, MA: Addison-Wesley.
- Coulouris, G., *et al.* (1994). *Distributed systems concept and design*, 2nd ed. Reading, MA: Addison-Wesley.
- Deitel, H. M., *et al.* (2000). *Internet & World Wide Web: How to program*. Englewood Cliffs, NJ: Prentice Hall.
- DeRoest, J. W. (1997). *AIX version 4: System administration guide*. New York: McGraw-Hill.
- Peterson, L. L., *et al.* (2000). *Computer networks*, 2nd ed. San Mateo, CA: Morgan Kaufmann.
- Russel, C., *et al.* (2000). *Windows 2000 server administrator's companion*. Microsoft Press.
- Sobell, M. G. (1997). *A practical guide to SOLARIS*. Reading, MA: Addison-Wesley.