Client-Server Computing

5.1 Introduction

The client–server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. A client-server network consists of two or more clients, or users, computers and at least one server computer. In a true client-server situation, the client computers do not use each other's resources, only those of the server. The server computer controls all files, folders, printers, and other resources on the network. Let us now define a client and a server.

Client - A client is a single-user workstation that provides presentation services and the appropriate computing, connectivity and the database services and the interfaces relevant to the business need.

Server- A server is one or more multi-user processors with share memory providing computing, connectivity and the database services and the interfaces relevant to the business need.

Figure 5-1 illustrates a simple client-server network. The server is a powerful computer that authenticates users and supplies files, folders, and resources to the users. The server also controls the network printer. Client computers must send a request to print, access a file, and so on, to the server first. Note that the hub is the central connection for all cables.

Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server machine is a host that is running one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

A small network might use one server for 2 or 100 clients. As networks expand and the need for services increases, you can add multiple servers to a network.

Very large networks might use 10 or 20 servers, for example, that serve hundreds or thousands of clients.

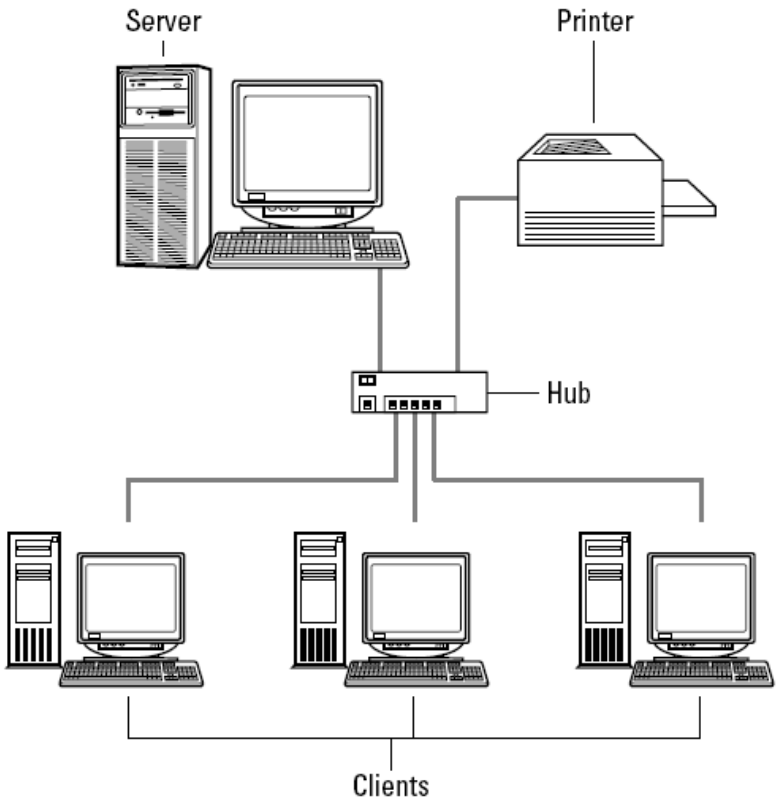

Figure 5-1: A Simple Client/Server Network Using Traditional Cabling Methods

Servers generally use faster processors and have more memory (RAM) and disk space to perform the management of the network. Servers perform multiple duties, including the following:

- Authenticating users
- Allowing access to resources—files, folders, printers, and so on
- Providing Web access
- Tracking resource usage
- Logging security breaches
- Distributing e-mail
- Providing application access and data

The client–server model has become one of the central ideas of network computing. Many business applications being written today use the client–

server model. So do the Internet's main application protocols, such as HTTP, SMTP, Telnet, and DNS.

Specific types of clients include web browsers, email clients, and online chat clients. Specific types of servers include web servers, ftp servers, application servers, database servers, name servers, mail servers, file servers, print servers, and terminal servers. Most web services are also types of servers.

5.1.1 Considering Network Requirements

Usually, a client/server network uses two or more client computers attached to a server computer. Although having only one client and one server in a client/server network is possible, you may not need a server in this situation. For each computer on the network, you need a network adapter card. Compatible cabling is also necessary, as well as any other networking hardware, such as a hub, switch, or phone jacks.

You can add printers and other resources to your network as well. When you add resources, you choose whether to attach them to a local computer or to the server computer. For example, you might have a laser printer you attach to the server and let everyone share. On the other hand, you also might have a color inkjet or an old dot-matrix printer that you attach to just one machine for use by that one person.

Figure 5-2 illustrates a client-server network in which the users share the laser printer attached to the network. The inkjet printer and the scanner, however, are attached to individual computers and can be used only locally. A network operating system doesn’t share resources attached to the clients, only resources attached to the network or to the server. If you want to set up the network in Figure 5-2 to share the inkjet printer and the scanner, you can, depending on the network operating system you use and how you configure your network.

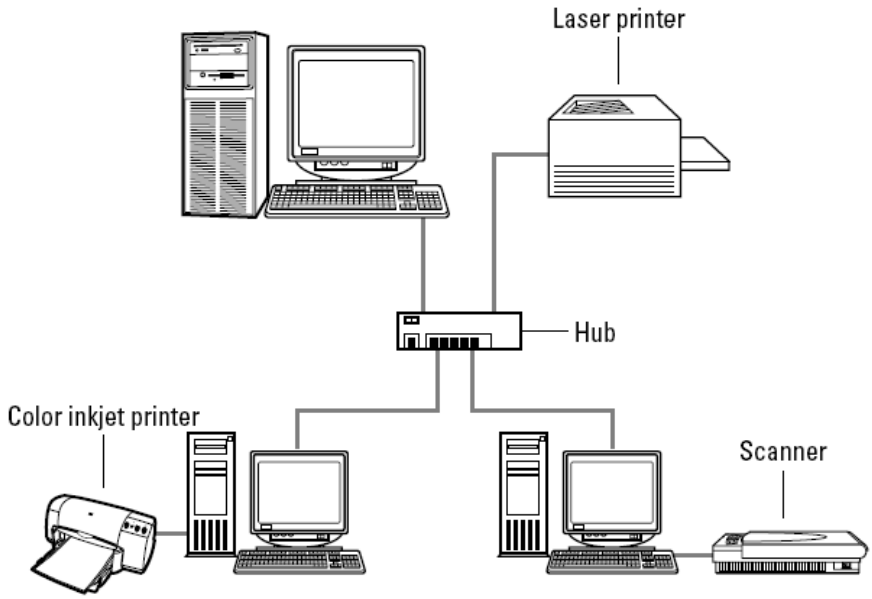

Figure 5-2: A Client-Server Network Uses Only Network Resources, Not Local Resources

5.1.2 Considering Server Requirements

A server computer must have sufficient processor speed, RAM, and disk space to provide various services to the clients. The server's hardware configuration depends on the type of services it will offer and the number of clients on the network.

In addition to hardware, the server needs a compatible operating system. You can use a network operating system, or you can use Windows XP as a server of sorts. (Microsoft has fazed out Windows 2000 Professional—a client computer—but they still have a specific network operating system called Windows 2000 Server.) A network operating system supplies more management, security, and other features and tools that make operating the network efficient. Windows XP as a server operating system limits the services, but might work for your network.

The hardware you choose for your server, first and foremost, must be compatible with the operating system. You first should choose the OS you will use and then purchase the server computer. Each OS requires specific amounts of memory and disk space and perhaps certain types of drives—such as Small Computer System Interface (SCSI) or Integrated Drive Electronics (IDE)—and other such requirements.

You can purchase server computers, complete with operating system, that are built specifically for the job. These computers have all the hardware compatible with the chosen operating system. You might be able to afford a supercomputer, which is a computer that has massive amounts of RAM, caching (temporary memory for speeding up computer processing), and disk space. However, you might need only a computer with extra memory and disk space for storing files and accessing the Internet. The choice you make depends on your budget, the number and type of services you want to offer, and the number of client computers.

5.1.3 Considering Client Requirements

Client computers can be almost any brand, any operating system, and have few or many bells and whistles. As long as a client computer is relatively new (2 to 3 years old) and you install a network interface card and networking software, you can use that computer as a client. Naturally, the client computer will work more efficiently on the network if it's a newer and more powerful computer; but even older computers and operating systems will likely work.

The next thing you need to check is the hardware. Does the computer already have a network card? If not, does it have a slot for a network card? After you confirm your operating system and get a network card, you're ready to connect the computer as a client.

5.1.4 Considering Network Operating Systems

A network operating system is one designed specifically for a server. A NOS offers many features and tools that help you manage the network, clients, applications, security, and other facets of the network. A network operating system is also difficult to install, configure, and maintain.

You can purchase a server computer, complete with a NOS installed. Nevertheless, you still need to configure the users, shares (permissions for others to use your files, folders, and so on), and other elements of the system in order to reap the full benefit of the server.

You already may be familiar with a network operating system from your job or some other experience. You may want to use a NOS for any of a variety of reasons. Perhaps you have specific networking needs—security problems, home business, Web business, and so on.

Another reason to choose a particular network operating system is that you're using an application that requires it. For example, Internet Information Server (a Web server) works best with the Windows 2000 network operating system. Some vertical (specially built) applications, such as a program for selling and listing real estate or managing an insurance business, might require a specific NOS.

Finally, you may have a server computer with an operating system already installed. Perhaps you bought a used server from work, or someone gave you equipment compatible with a specific network operating system.

If you have no preferences for a NOS, you should research the alternatives and find one that suits your needs. If, for example, you like Windows and you're comfortable with Windows, you might want to try Windows 2000 Server, or try using Windows XP as a server. If you're a big Linux fan, consider using a Linux server.

Realize, however, that a network operating system is not as easy to install, configure, and maintain as Linux or Windows on a standalone machine. A NOS takes a lot of time and effort to run successfully.

When looking for a NOS, decide what is important to you in your network. Obviously, easy maintenance should be a priority in a home network. You also must consider your budget. Any network operating system should include certain essentials, however, and some features you simply might want in a NOS. Consider the following:

- All network operating systems include a tool for naming the users of the network and limiting their access to certain resources. Through user accounts, you can choose which files and folders a user may access, as well as which resources he or she can use, and you can limit access to other computers or servers as well.
- A good network operating system should include some sort of printer management tool. This tool helps you direct print jobs to the appropriate printer, cancel and delete jobs, and otherwise control printing on the network.
- Most network operating systems include diagnostic tools for examining the network components, such as protocols and connections. When something goes wrong with a connection, these tools make it easier to find the problem.

- Tools and utilities for gathering and analyzing network data might be important to you. Some NOSs log application errors and security breaches, for example. Others use optimization utilities to help you figure out where the network connections are slow.
- Some NOSs include Web utilities and support for browsers. You might want to create your own Web server, for example, for displaying home pages over the Internet.

Network operating systems are complicated to install and operate. If you plan to use a NOS, you'll be investing considerable time and energy in learning the program and managing the network. Before you purchase a network operating system, find out about the features, requirements, and compatibilities of the system first. Make sure that the system meets your networking needs before you go in for it.

Following is a list of the most popular network operating systems:

- Microsoft Windows 2000 Server, 2003 Server, 2008 Server
- Novell NetWare/IntranetWare
- Mac OS eSoft Server
- Linux

5.2 Client-side Vs Server-side programs

Web pages are displayed in your browser on your local machine. Just like a customer or client for a restaurant, you and your browser are a client using a Web site. The Web site is displayed by your browser, which interprets code that was sent to it. In general this code will be primarily HTML, but may also contain anything supported by your browser such as javascript, flash movies, and more.

The machine where the Web site actually resides is called a Web server. When you send a request for a Web page by entering a Web site address, this request is sent to a Web Server. The Web server then sends the Web page to your browser. The interesting thing about a Web server is that it can manipulate the code within a Web page before sending it to your browser. For most Web pages, no manipulation is done. Rather, the server simply sends a copy to the browser and the browser does the work of displaying the code.

A Web Server can manipulate what is included in a Web page before sending it. When the request is made to a server to send a Web page, the server can

actually execute a program instead. This can be a program written in a variety of languages. Some of the programming languages and technologies that can be used are Active Server Pages (ASP), PHP, C/C++, Java Server Pages (JSP), and more. Of course, in order to use any of these languages, the server must support them.

An important issue when working with Web applications is to remember that the Web server is separate from the Web browsers that will use a Web site's pages. The Web server can be located anywhere in the world -- or even off the world! The browser is generally on a machine you are using. Stated a different way, you can act as a customer, or client, using a browser to access a Web site that is located on a Web server.

The programs running on the Web Server are server side programs because they are on the side of the internet that the Web server is on. The browser being used to access the Web site is on the same side of the Web as you, the client side. If code is executed on the Web server, it is considered server side code. If code is executed on your browser, it is considered client-side. Because the internet is vast, the client side and server side programs are not constantly in contact with each other. Because of this, there is code that you can use on the server and there is code that you can use on the client.

There are a number of types of programming things that can be done on the server. One of the primary functions you can do with code is to prepare the code that is to be sent to the Web browser. This includes such tasks as building pages customized for the type of browser that requested a page. It could also include doing tapping into a database to create information for a Web page. A very popular server side program is a Visitor Counter that keeps track of the number of people who have accessed a Web Site. The counter program would keep track of people who have come and store the information. It would also provide the actual number for any Web pages that are sent back to a browser.

HTML, javascript, Flash files, ActiveX controls, Java applets, and a number of other technologies can be executed on the client side. You can execute any technology supported by your browser. Client side programming is used because the browser is separate from the server. By including code within a web page, a number of features can be added to a Web page without the need to send information to the Web Server which takes time. Tasks done on the client side include data validation, special formatting features that go beyond HTML, controls that take care of page navigation and ad presentation, and more.

5.3 Client-Side Scripts

"Scripts" is a general programming term for short, text-based software programs. Client-side scripts are embedded in HTML web pages, which execute within the viewer's web browser software (i.e. they execute on the "client" side of the client-server web architecture, rather than on the web server). These scripts can be used to do simple computations, verify data, display calculated information on a page, respond to the user's mouse motion and mouse clicks, put new browser windows on the screen, show and hide text and images, change text styles, and create drop-down menus and similar effects.

Client-side scripting languages do not have the complete functionality that would be available in a full-fledged programming language (such as C or Java), and for security reasons, are limited to doing calculations and actions based on the content, markup, and embedded data from the HTML page they are in, along with information entered by the user in forms on that page, and reactions to things the user does in the browser window (such as moving the mouse).

The most commonly-used client-side scripting language is JavaScript, which is supported on most relatively recent browsers; unfortunately, each browser has its own variation of JavaScript, as there are not any real standards for the language. Other client-side scripting languages, which are supported only by a limited number of browsers, include VBScript and JScript. Typically, a software developer is needed to write client-side scripts, but many pre-written scripts can be purchased or downloaded, for use and modification by less technical web designers.

5.4 Browser Plug-Ins

Browser plug-ins are programs a user can download, which run inside a web browser window and add functionality to the browser -- that is, they allow the browser to display non-HTML content, and interact with the user in ways that go beyond client-side scripting. In some cases, the plug-in enables content to be displayed inside an HTML page; in other cases, the plug-in takes up the whole browser window. Generally, once the user has downloaded a plug-in for their browser, whenever the browser encounters content that corresponds to the plug-in, it automatically displays it in the current browser window.

There are many plug-ins available for web browsers; some of the most commonly used are the Adobe Acrobat Reader, for displaying PDF files; Sun,

Microsoft, and IBM Java Runtimes, for running Java applets embedded in HTML pages; Windows Media Player, RealPlayer, and Apple QuickTime, for playing sounds and movies in various formats, usually embedded in HTML pages; and Macromedia Flash, for displaying Flash animation embedded in HTML pages.

Creating a plug-in program requires significant software development expertise and time, and each type of browser, on each operating system, requires a separate plug-in to be developed. For that reason, most web designers and developers will not develop their own plug-in software, but instead may develop content that can be displayed using existing plug-ins (such as Flash Action Scripts, MPG movies, and PDF files). Software development expertise is also necessary to create complex Flash scripts and Java applets.

The table below lists a number of effects that are commonly used on web sites, and shows which technologies can be used to achieve them.

Effect	Description	Technology
Backgrounds	Background colors and images for pages and parts of pages	HTML and CSS
Appearing and disappearing text and images	When the user clicks or moves the mouse in some region on the web page, text and/or image(s) appear or disappear in that region or elsewhere on the page	Client-side scripts or Flash plug-in
Drop-Down menus	The user can choose from a menu, or a cascading series of menus, to navigate to different pages in the web site	Client-side scripts or Flash plug-in
Sounds	The user hears music or other sounds when viewing a web page	Flash or other media plug-in
Animation	A movie or other animation is displayed in a web page	Animated GIF image embedded in mark-up, for simple animations; plug-ins for more complex movies and animation
Surveys and forms	The user can enter information in a form, and transmit it to the site owner	HTML markup to display the page, and server-side scripts and databases (or a content management system) for processing and storing the information. Client-side scripts can also be useful for validating the information before the user transmits it.
Shopping	The user can choose items from an on-line catalog, put them in a shopping cart, and purchase them	HTML markup to display the pages; server-side scripts (or a content management system) to process the orders. Most web designers will use a pre-built shopping cart system, rather than developing their own.
Interactive games and learning	and learning The user can play a game, use a learning tool, or do something else interactive, where their mouse motions and/or keyboard entries are immediately responded to	Flash and Java applet plug-ins
Searching and sorting	The user can search the site for products or information, and receives a sorted list of answers	Server-side scripts and databases (or a content management system) can be used to create a web-based catalog (or other list), or a searchable information index. Simple site search of static HTML pages can be done using commercial search engines (i.e. making use of their database of indexed web pages and their server-side scripts for searching the database).
Guest book	The user can enter comments, or perhaps upload photos or other documents, for viewing by other users	Server-side script and database, or content management system
Time-dependent content	Content that changes based on the date or time of day	Server-side scripts and database, or content management system
Password protection	User must log in to visit parts of the site	Simple password protection can be done using server set-up preferences. Some login-enabled sites are also built using server-side scripts and databases; content management systems generally have advanced user management capabilities.
Custom error pages	If user requests a non-existent page, or tries to access a page they are not authorized to view, they get a prettier and more comprehensible error page, rather than the default "404 not found" error page	Server set-up

5.5 Common Gateway Interface (CGI) Programs

The Common Gateway Interface (CGI) program is used to provide interactivity to web applications so that they become responsive and dynamic to the requests made by a web browser or a client application. A CGI program contains a set of conventions that enable the client programs and the web servers to communicate. The web server usually acts as an intermediary between the CGI program and the client browser. When a client browser sends a request to the web server, it executes the CGI program based on the input received and sends the output back to the web browser for processing.

CGI programs are the most common way for Web servers to interact dynamically with users. Many HTML pages that contain forms, for example, use a CGI program to process the form's data once it's submitted. Another increasingly common way to provide dynamic feedback for Web users is to include scripts or programs that run on the user's machine rather than the Web server. These programs can be Java applets, Java scripts, or ActiveX controls. These technologies are known collectively as *client-side* solutions, while the use of CGI is a *server-side* solution because the processing occurs on the Web server.

CGI programs can be written in any programming language and they can also be used to interact with other programs or CGI programs during their execution.

A CGI program would not do the following.

- it does not interact with a user directly.
- It does not interact directly with a web browser or a graphical interface. In other words it does not display or retrieve information from menus, commands and other interactive features of a client browser.
- It does not create graphics or windows by itself.

Following are some of the advantages and disadvantages of CGI programs.

5.5.1 Advantages

- CGI programs are portable and work on a wide variety of web servers and platforms.
- They are language independent. You can write then in any language and make them work in a wide variety of environments. Some of the programming languages good enough to write your CGI scripts are Perl, UNIX Shell, C language, Visual Basic, Python, C# and Java.

- They provide simple interfaces for the clients to interact with the web servers.
- They are scalable programs and you can use them to perform simple tasks in the application layer as well as more complex tasks such as interacting with databases and shopping carts.
- They provide interactivity to a web application and enhance user experience.
- CGI programs are cost effective. By using CGI programs, businesses can lower their development and maintenance costs.

5.5.2 Disadvantages

- The CGI programs are memory intensive programs. Every time a request is made to a server, it has to launch the CGI program. If they are written in a scripting language, the interpreter for the scripting language has to evaluate the entire script to execute the CGI program each time the program is initiated. If the server has busy traffic, repeated client side requests would consume a great deal of server resources and impact server performance. This problem is minimized to some extent by CGI scripts written in a compiled or interpreted language like C where there is no need for an interpreter to read the CGI program code separately.
- CGI programs are not easy to write. They require complex programming and designing skills on the part of the web developers since a lot depends upon how they are implemented in the server environment.
- If proper care is not taken, CGI programs may compromise server security.
- Most of the CGI programs are well known, free and easily available. Their strengths and vulnerabilities are known to most web developers. This often results in their exploitation and misuse.

5.6 The Applet Concept

An applet is any small application that performs one specific task, sometimes running within the context of a larger program, perhaps as a plug-in.. Prior to the World Wide Web, the built-in writing and drawing programs that came with Windows were sometimes called "applets." On the Web, using Java, the object-oriented programming language, an applet is a small program that can be sent along with a Web page to a user. Java applets can perform interactive animations, immediate calculations, or other simple tasks without having to send a user request back to the server.

Some applets are able to function as any other normal software application (provided they are hosted by an operating system), but are small in size and perform only a small set of tasks. Examples of applications often classified as applets are all of the accessories bundled in Microsoft Windows (such as Windows Notepad or Microsoft Paint). Applets are not full-featured application programs. In some cases, an applet does not run independently. Such applets must run in a container, which is provided by a host program, through a plugin, or a variety of other applications including mobile devices that support the applet programming model

HTML pages may embed parameters that are passed to the applet. Hence the same applet may appear differently depending on the parameters that were passed. Examples of Web-based Applets include:

- QuickTime movies
- Flash movies.
- Windows Media Player applets - used to display embedded video files in Internet Explorer (and other browsers that support the plugin).
- 3D-modelling display applets allowing a view of a model to be rotated and zoomed.
- Browser games can be applet-based, though some may develop into fully functional applications that require installation.

5.7 Characteristics of the Client and the Server

The clients and the servers are the logical entities that work together over a network to accomplish a task. The distinguishing characteristics of the Client-Server systems are:

Service: The client/server is primarily a relationship between processes running on separate machines. The server process is a provider of services. The client is a consumer of services. In essence, client-server provides a clean separation of function based on the idea of service.

Shared Resources: A server can service many clients at the same time and regulate their access to shared resources.

Asymmetrical protocols: There is a many-to-one relationship between the clients and the server. Clients always initiate the dialog by requesting a service. Servers are passively awaiting request from the clients. In some cases a client may pass a reference to a callback object when it invokes a service. This lets the server call back the client. So the client becomes a server.

Transparency of location: The server is a process that can reside on the same machine as the client or on a different machine across a network. Client-Server software usually masks the location of the server from the clients by the redirecting the service calls when needed. A program can be a client, a server, or both.

Mix-and-match: The ideal client-server software is independent of hardware or operating system software platforms. You should be able to mix-and-match client and server platforms.

Message-based exchanges: Clients and servers are loosely coupled systems that interact through a message-passing mechanism. The message is the delivery mechanism for the service request and replies.

Encapsulation of services: The server is a specialist. A message tells a server that it is requested; it is then up to the server to determine how to get the job done. Servers can be upgraded without affecting the clients as long as the published message interface is not changed.

Scalability: Client-Server systems can be scaled horizontally or vertically. Horizontal scaling means adding or removing client workstations with only a slight performance impact. Vertical scaling means either migrating to a larger and faster server machine or distributing the processing load across multiple servers.

Integrity: The server code and server data is centrally managed, which results in cheaper maintenance and the guarding of shared data integrity. At the same time, the clients remain personal and independent.

5.8 Types of Servers

5.8.1 File Server

File Servers shown in Figure 5-3 are useful for sharing information across the network. The client passes a request for file records over a network to the file server. This is the most primitive type of data service used for exchanging messages over the network to find the requested data. The file servers provide access to the remote server processors. In the typical implementation the software, shared data, databases and backups are stored on disk, tape and optical storage devices that are managed by the file server.

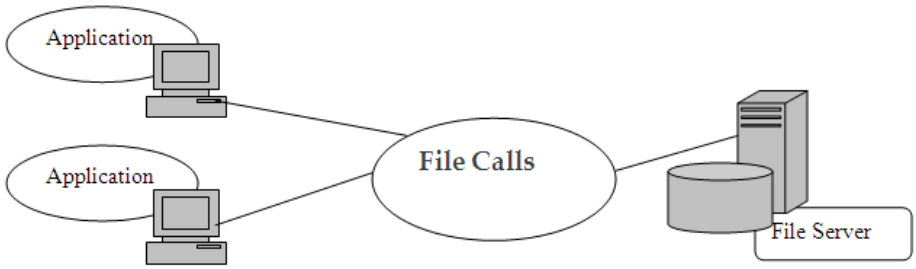

Figure 5-3: A Client-Server with a File Server

5.8.2 Database Server

The client passes the SQL requests as messages to the database server; the result of each SQL command is returned over the network. The code, which processes the SQL request and the data, reside in the same machine, the server uses its own processing power to find the requested data back to the client, instead of passing all the records back to the client. This results in a much more efficient use of the distributed processing power. The database servers shown in Figure 5-4 provide the foundation for decision-support systems and also provide key role in data warehousing.

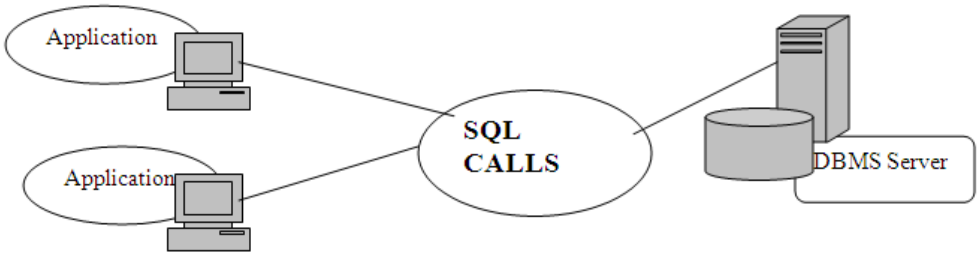

Figure 5-4: A Client-Server with a Database Server

5.8.3 Transaction Servers

The client can invoke remote procedure or services that reside on the server with an SQL database engine using the transaction server. The network exchange consists of a single request/ reply. The SQL statements either all succeed or fail as a unit. These grouped SQL statements are called transactions. With a transaction server shown in Figure 5-5 you create the client-server application by writing the code for both the client and the server components. The client component usually includes a Graphical User Interface (GUI). The server component consists of SQL transaction against a database. These applications are called Online Transaction Processing or OLTP. The OLTP are mission-critical applications that require a very less response time (1-3 sec). The

OLTP applications also require tight controls over the security and integrity of the database.

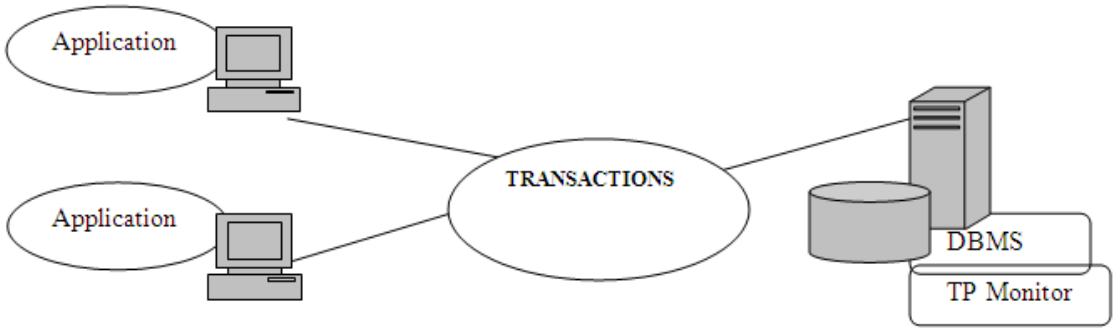

Figure 5-5: A Client-Server with a Transaction Server

5.8.4 Groupware Servers

It involves the management of semi-structured information such as text, image, mail, bulletin boards and the flow of work. This client-server system places people in direct contact with other people. Best examples are webmail servers, etc. Specialized groupware software can be built on top of a vendor's canned set of client-server API's. In many cases, applications are created using a scripting language and form-based interfaces provided by the vendor. Now many groupware products use e-mail as their standard messaging middleware. Also, Internet is quickly becoming the middleware platform of choice for groupware.

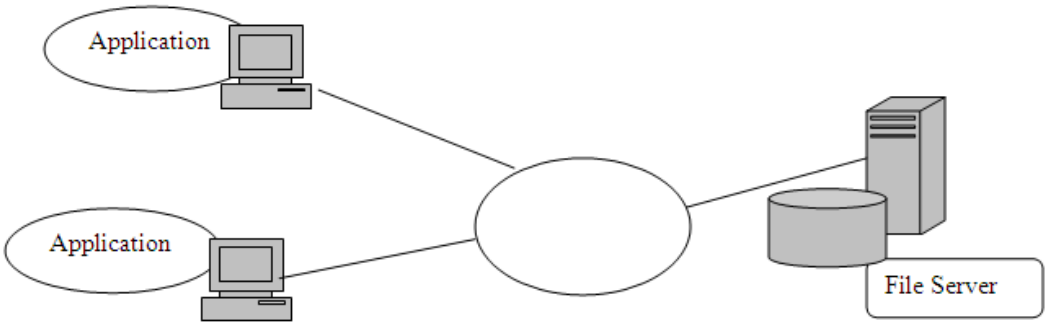

Figure 5-6: A Client-Server with a Groupware Server

5.9 The Architectures

The gurus of the client-server architecture refer to them as 2-tier, 3-tier and N-tier client/server architecture. This is the means by which they are functionally split. The functional units comprise of user interface, business logic and the shared data.

5.9.1 2-tier Client/Server Architecture

Two tier software architectures were developed in the 1980s from the file server software architecture design. The two-tier architecture shown in Figure 5-7 is intended to improve usability by supporting a forms-based, user-friendly interface. The two-tier architecture improves scalability by accommodating up to 100 users (file server architectures only accommodate a dozen users). It also proves flexibility by allowing data to be shared, usually within a homogeneous environment. The two-tier architecture requires minimal operator intervention, and is frequently used in non-complex, non-time critical information processing systems.

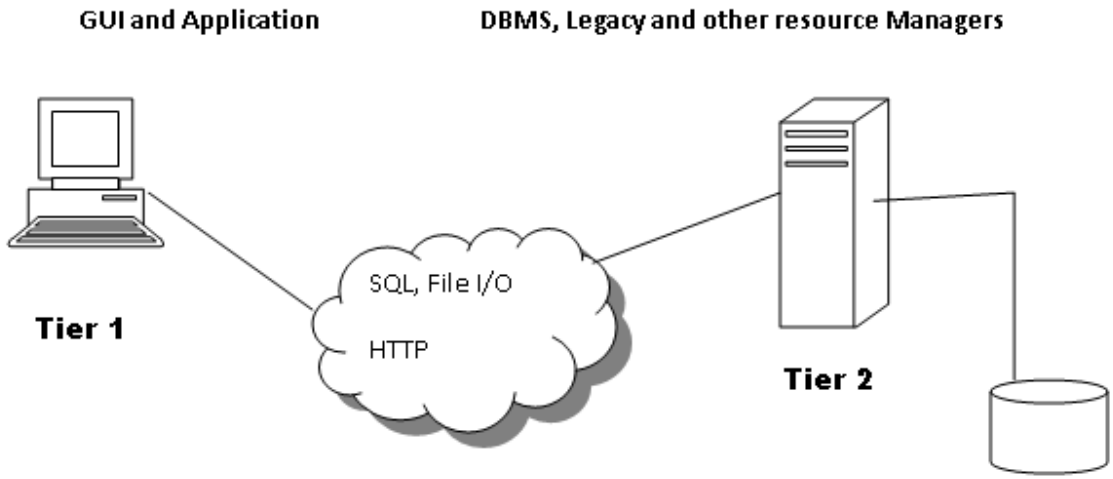

Figure 5-7: A 2-tier Client- Server Architecture

Two tier architectures consist of three components distributed in two layers: client (requester of services) and server (provider of services).

The three components are

1. User System Interface (such as session, text input, dialog, and display management services)
2. Processing Management (such as process development, process enactment, process monitoring, and process resource services)
3. Database Management (such as data and file services)

The two-tier design allocates the user system interface exclusively to the client. It places database management on the server and splits the processing management between client and server, creating two layers.

The application logic may be present at the client side within a user interface or it may be present within the database on the server or on the both. It is most popular because of its simplicity. These applications can be quickly built by using and visual builder tools; which can be used for developing applications for decision support system of small-scale groupware or you may build a simple web publishing applications.

But the real problem arises only when you deploy them beyond the departmental LAN. Typically the applications that worked perfectly well in prototypes and small installations failed for large-scale productions. It actually went through a transition phase, where it grew beyond the departmental LAN's. Thus the complex world is now faced by the 3-tier and the N-tier applications.

5.9.2 3-tier Client-Server Architecture

The three-tier software architecture shown in Figure 5-8 emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging.

The three tier architecture is used when an effective distributed client-server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. They are also easy to manage and deploy the network and most of the code runs on the server.

The middle tier provides process management services (such as process development, process enactment, process monitoring, and process resourcing) that are shared by multiple applications. The third tier provides database management functionality and is dedicated to data and file services that can be optimized without using any proprietary database management system languages.

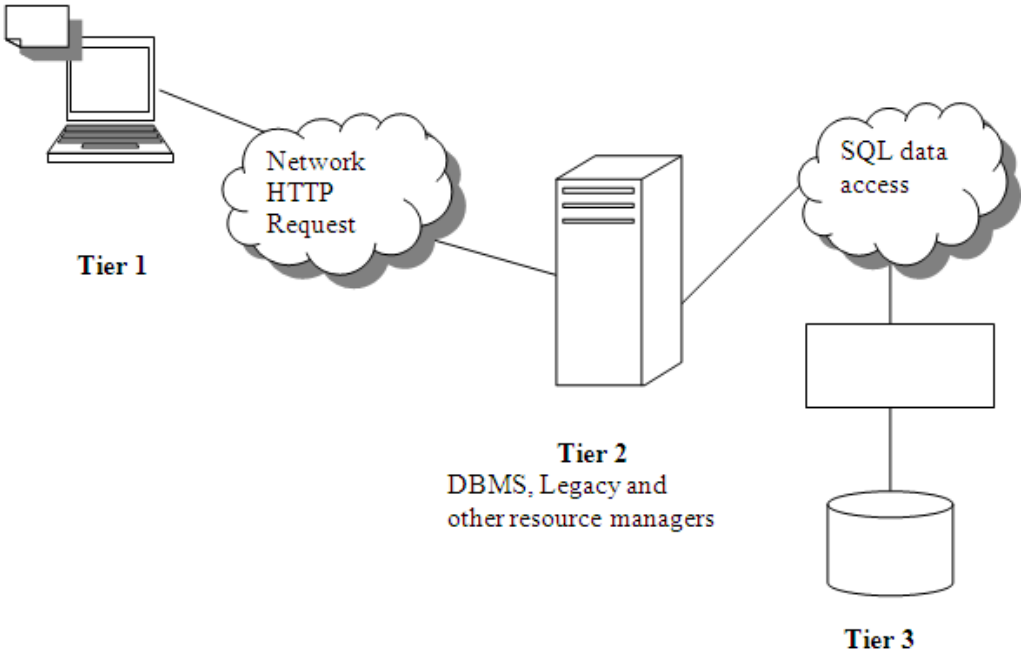

Figure 5-8: A 3-tier Client- Server Architecture