Unit 7 - Class Notes

Client Server Computing

Client server computing uses distributed computing paradigm. If we look back at the history of modern computing and look at the computing being done using mainframes or mini computers we see some advantages of using them. One of the big advantages was the sharing of resources and data. Data stored on the hard disk of a mainframe or minicomputer was available to any program running on the machine. In the same way hardware resources (like printer, punch card reader, etc.) connected to the machine were available to any program running on the machine. The main problem with the mainframes and minicomputers was that they were expensive and hard to use. Personal Computers (or PC) were invented in 1970s. PC is easier to use (especially since the development of graphical user interface (GUI) and point and click devices like mouse). GUIs and mouse provides natural point and click user interface to computer applications. Each PC has its own resources (like hard disk, printer, etc.), which makes it hard to share resources like data and printer. Networking of PCs resulted in sharing of resources like printers but it was still not possible to share the data in real time.

Client server computing combines best features of main frame computing and PC computing. In client server computing we can share hardware as well as software. In client server computing, application is decomposed into parts or components. Some components run on the shared server machine and some components run on the individual clients. Server provides data processing and functionality to persist data. Client provides user interface using natural point and click graphical interface to application. User interface can be a custom application or browser based (web application). Clients can be divided into two categories. A thick client runs some or all of the application's logic and provides user interfaces. A desktop PC, laptop, or a hand held computing device can be used as a thick client. A thin client provides user interface only. All processing is done on the server. In many cases screens for user interfaces are generated on the server and bit map images of the UI is transmitted to the thin client which in turns displays it on the monitor. Examples of devices used for thin clients include WYSE thin client terminal and Pano logic zero client (see below for links to their web sites).

A WYSE (now owned by Dell) thin client terminal is shown in the above picture. It contains RAM and ports for connecting keyboard, mouse, monitor, speakers, microphone, USB devices, and network. It can be configured to connect with a server. It can communicate with server using Remote Desktop Connection. All applications run on the server. Client is used for user interface only.

A Pano Logic Zero client is shown in the above picture. It contains ports for connecting keyboard, mouse, monitor, speakers, microphone, USB devices, and network. It does not contain any RAM. When it is booted it sends its device ID on the net which can be intercepted by a management widget. Management widget in turn returns server IP to the client so it can connect with the server. All applications run on the server and client is used for user interface only. Unfortunately Pano Logic went out of business on 10/23/2012. Teradici is another company that offers Zero Client (PCoIP Zero Client). Watch the following video about their zero client:

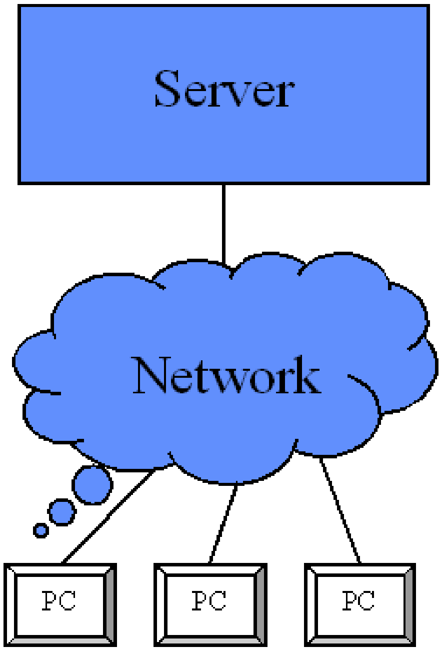
http://www.youtube.com/embed/DpJ_A7B_TWk?autoplay=1

Advantages of client server computing include:

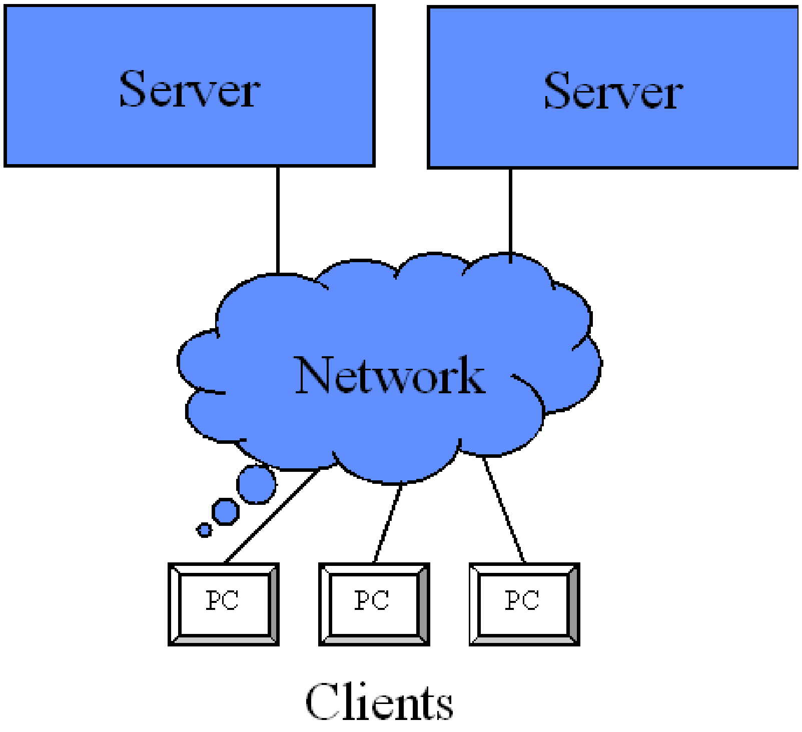
- Centralization – access, resources, and data security are controlled through server.
- Upgradeable – any element can be upgraded when needed.
- Flexibility – new technology can be easily integrated into the system.
- Interoperability – all components (clients, network, servers) work together to accomplish the given task.
- Accessibility – servers (and all the resources on it) can be accessed remotely and across multiple platforms.
- Ease of application development – server and client applications can be developed independently.
- User friendly – familiar point and click interface.

Disadvantages of client server computing include:

- Dependability – when server goes down, all operations will cease.
- Lack of scalability – it is not always possible to add more users without downgrading the response time of the server.
- Can cause network congestion which can again result in an increase in response time.
- Security – as server can be accessed remotely, it is more vulnerable to attack.

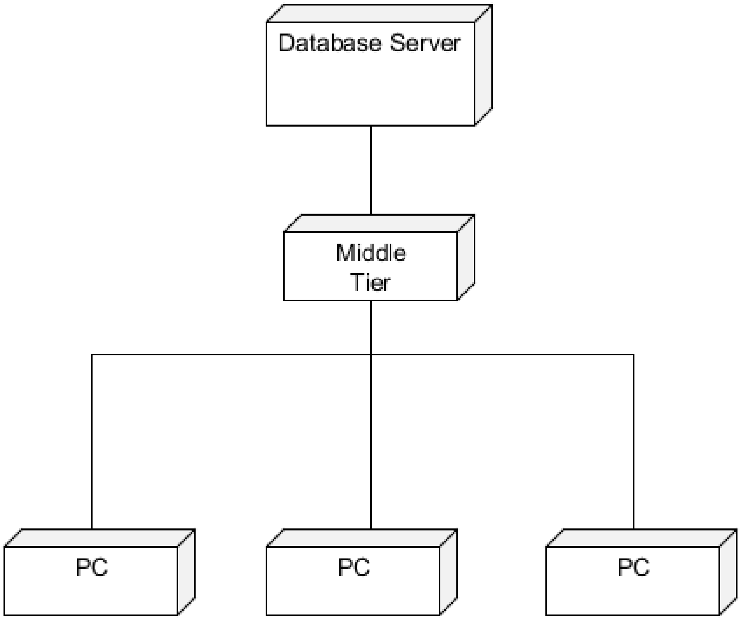


There are basically two types of client server architecture: two tier and three tier. In a two tier architecture (shown in the above picture), the application components are distributed between the server and the clients. The client assumes the bulk of the responsibility for executing application logic. Business rules related to the applications are implemented in the client components. The presentation to the user is also handled by the client software. In addition to running parts of the application software, the server is responsible for data persistence. All data access is through the server. The server assumes the bulk of the responsibility for data integrity check, query capabilities, data extraction and most of the data intensive tasks, including sending the required data to the appropriate client. Data sent to the client is manipulated by the client software for reporting, business analysis, and "what if" analysis.



Three tier architecture (shown in the above figure) separates the presentation, processing, and data persistence into three distinct entities. The client handles the presentation to the user. When data or processing is required by the presentation client, a call is made to the middle-tier functionality (or application) server. This tier performs all the calculations, generates reports, and makes any needed calls to other servers (like data base server). Middle tiers are usually coded in highly portable, non-proprietary language such as C or Java. Middle tier are multi-threaded (runs in multiple process) and can be accessed by multiple clients at the same time. Data layer (or database server) is responsible for data persistence, data integrity check, query capabilities, data extraction and most of the data intensive tasks, including sending the required data to the middle tier. Middle tier and database tier can run on either same physical machine or on separate physical machines.

Note: The above figure shows the physical interconnection of the three layers. All components are interconnected through the same LAN but the clients can talk to middle tier only. Clients don't have direct access to database server. The following figure shows the logical connection between three tiers:



Please read the following articles to learn more about client server computing:

Client-Server Computing in Mobile Environment by Jing, Helal, and Elmagarmid (See online class notes for link to download the paper). This is paper was published in June 1999 but it is amazing that they were visualizing mobile computing even before smart phones and tablets.

Client/Server Fundamentals: http://gbengasesan.com/fyp/7/ch25.htm

WYSE Technology: http://www.dell.com/en-us/work/learn/virtual-client

One example of the client server computing is the web application. Web applications are more vulnerable to security related issues. Please read the following articles about network and web application security:

OWASP (Open Web Application Security Project)
http://www.owasp.org/index.php/Main_Page

Five Common Web Application Vulnerabilities
http://www.securityfocus.com/infocus/1864

The following site lists tools to check the vulnerability of web applications:

Top 10 Web Vulnerability Scanners http://sectools.org/web-scanners.html