

A

Seminar report

On

Client Server Architecture

Submitted in partial fulfillment of the requirement for the award of degree
of Bachelor of Technology in Computer Science

SUBMITTED TO:
www.studymafia.org

SUBMITTED BY:
www.studymafia.org

Acknowledgement

I would like to thank respected Mr..... and Mr.for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a seminar report. It helped me a lot to realize of what we study for.

Secondly, I would like to thank my parents who patiently helped me as i went through my work and helped to modify and eliminate some of the irrelevant or un-necessary stuffs.

Thirdly, I would like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Next, I would thank Microsoft for developing such a wonderful tool like MS Word. It helped my work a lot to remain error-free.

Last but clearly not the least, I would thank The Almighty for giving me strength to complete my report on time.

Preface

I have made this report file on the topic **Client Server Architecture**; I have tried my best to elucidate all the relevant detail to the topic to be included in the report. While in the beginning I have tried to give a general view about this topic.

My efforts and wholehearted co-corporation of each and everyone has ended on a successful note. I express my sincere gratitude towho assisting me throughout the preparation of this topic. I thank him for providing me the reinforcement, confidence and most importantly the track for the topic whenever I needed it.

www.studymafia.org

Introduction to Client/Server Architecture

Client-server architecture, architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them.

Ideally, a server provides a standardized transparent interface to clients so that clients need not be aware of the specifics of the system (i.e., the hardware and software) that is providing the service. Clients are often situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines. This computing model is especially effective when clients and the server each have distinct tasks that they routinely perform.

In hospital data processing, for example, a client computer can be running an application program for entering patient information while the server computer is running another program that manages the database in which the information is permanently stored. Many clients can access the server's information simultaneously, and, at the same time, a client computer can perform other tasks, such as sending e-mail.

Because both client and server computers are considered intelligent devices, the client-server model is completely different from the old "mainframe" model, in which a centralized mainframe computer performed all the tasks for its associated "dumb" terminals.

The Purpose of Client/Server Architecture

We are in an era where information technology plays a critical role in business applications, considered as an area an organization would highly invest in order to widen the opportunities available to compete the global market. “*A competitive global economy will ensure obsolescence and obscurity to those who cannot or are unwilling to compete*”(Client/Server Architecture,2011), according to this statement it's necessary for organizations sustain its market position by re-engineering prevailing organizational structures and business practices to achieve their business goals. In short it's a basic need to evolve with the change of technological aspects.

Therefore organizations should undergo a mechanism to retrieve and process its corporate data to make business procedures more efficient to excel or to survive in the global market. The client/server model brings out a logical perspective of distributed corporative processing where a server handles and processes all client requests. This can be also viewed as a revolutionary milestone to the data processing industry.

“Client/server computing is the most effective source for the tools that empower employees with authority and responsibility.”(Client/Server Architecture,2011)

“Workstation power, workgroup empowerment, preservation of existing investments, remote network management, and market-driven business are the forces creating the need for client/server computing”. (Client/Server Architecture,2011)

Client/server computing has a vast progression in the computer industry leaving any area or corner untouched. Often hybrid skills are required for the development of client/server applications including database design, transaction processing, communication skills, graphical user interface design and development etc. Advanced applications require expertise of distributed objects and component infrastructures.

Most commonly found client/server strategy today is PC LAN implementation optimized for the usage of group/batch. This has basically given threshold to many new distributed enterprises as it eliminates host-centric computing.

What Is A Client/Server?



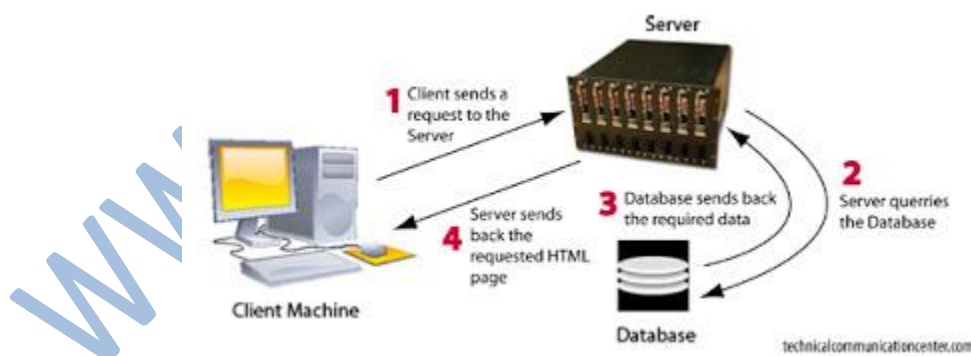
Client

A client is a single-user workstation that provides presentation services, database services and connectivity along with an interface for user interaction to acquire business needs.

Server

A server is one or more multi-user processors with a higher capacity of shared memory which provides connectivity and the database services along with interfaces relevant to the business procedures.

Client/Server computing provides an environment that enhances business procedures by appropriately synchronizing the application processing between the client and the server.



Client/Server Architecture

The protocol relies on the basis of request serving; request for services by clients; response of processed results by the server. The communication between the two aspects is gained through InterProcess communication (IPC) which facilitates the distributed placement of the client and server procedures.

The Client/Server model is basically platform independent and blends with “cooperating processing” or “peer-to-peer” model. The platform provides the opportunity for users to access the business functionality thereby exposing into risky situations since its transparent to the underlying technology as well as to the user.

www.studymafia.org

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.

- **Service:** The client/server is basically a relationship between processes running on distributed devices, server process considered as a supplier of services where the client process is a consumer of services. Briefly, this methodology provides a separation of functionalities subject to offered services.
- **Resource sharing:** A server is eligible of handling clients simultaneously, controlling the service access for the resources.
- **Asymmetrical protocols:** Client/server is considered as a many-to-one relationship which is initiated by clients through request of service while the server passively awaits. Sometimes a client may pass a reference to a callback object when it requests a service. This enforces the server call back the client, making the server a client itself.
- **Transparency of location:** The server process can reside in a client or any machine across a network .In such situations Client/Server software is responsible for implying the server location by redirecting service calls. Therefore a program can be a client/server/or both.
- **Inter- Communication via messages:** Interaction between clients and servers is obtained through a message-passing mechanism mainly to deliver service requests and responses.

- **Encapsulation of services:** A server is specialized in satisfying client requests varyingly and can be upgraded without affecting its external environment (clients, shared resources) as long as the message broadcasting interface remains the same.
- **Scalability:** Client/Server systems can be scaled horizontally or vertically. Horizontal scaling implies the addition or removal of client workstations with a minor impact in performance. Migration to more efficient servers or dividing the work load over numerous servers is considered as vertical scaling.
- **Integrity:** Since the server code and server data is managed centrally, maintenance cost is less and results in shared data consistency and undependability of clients.

Advantages

Organizations often seek opportunities to maintain service and quality competition to sustain its market position with the help of technology where the client/server model makes an effective impact. Deployment of client/server computing in an organization will positively increase productivity through the usage of cost-effective user interfaces, enhanced data storage, vast connectivity and reliable application services.

If properly implemented its capable of improving organizational behavior with the help of the knowledgeable worker-who can manipulate data and respond to the errors appropriately.

- ***Improved Data Sharing***

Data is retained by usual business processes and manipulated on a server is available for designated users (clients) over an authorized access. The use of Structured Query Language (SQL) supports open access from all client aspects and also transparency in network services depict that similar data is being shared among users.

- ***Integration of Services***

Every client is given the opportunity to access corporate information via the desktop interface eliminating the necessity to log into a terminal mode or another processor. Desktop tools like spreadsheet, power point presentations etc can be

used to deal with corporate data with the help of database and application servers resident on the network to produce meaningful information.

- ***Shared Resources amongst Different Platforms***

Applications used for client/server model is built regardless of the hardware platform or technical background of the entitled software (Operating System S/W) providing an open computing environment, enforcing users to obtain the services of clients and servers (database, application, communication servers).

- ***Inter-Operation of Data***

All development tools used for client/server applications access the back-end database server through SQL, an industry-standard data definition and access language, helpful for consistent management of corporate data. Advanced database products enable user/application to gain a merged view of corporate data dispersed over several platforms. Rather than a single target platform this ensures database integrity with the ability to perform updates on multiple locations enforcing quality recital and recovery.

- ***Data Processing capability despite the location***

We are in an era which undergoes a transformation of machine-centered systems to user-centered systems. Machine-centered systems like mainframe, mini-micro applications had unique access platforms and functionality keys, navigation options, performance and security were all visible. Through client/server users can directly log into a system despite of the location or technology of the processors.

- ***Easy maintenance***

Since client/server architecture is a distributed model representing dispersed responsibilities among independent computers integrated across a network, it's an advantage in terms of maintenance. It's easy to replace, repair, upgrade and relocate a server while clients remain unaffected. This unawareness of change is called as encapsulation.

- *Security*

Servers have better control access and resources to ensure that only authorized clients can access or manipulate data and server-updates are administered effectively.

www.studymafia.org

Disadvantages-compared to peer-peer networks

- ***Overloaded servers***

When there are frequent simultaneous client requests, servers severely get overloaded, forming traffic congestion. But in a P2P network adding more nodes will increase its bandwidth since it's calculated as the sum of bandwidths of each node in the network.(slideshare 2011)

- ***Impact of centralized architecture***

Since its centralized if a critical server fails, client requests are not accomplished. Therefore client/server lacks robustness of a good P2P network (resources are distributed among many nodes).

Implementation methods and their applications

The client/server architecture functionality divides into categories as 2-tier, 3-tier and N-tier models and the serviceable units consist of user interface, business logic and the shared data. Since the content of this report is towards a network perspective, I have not elaborated on the 1-tier architecture as they are non-distributed, end-user applications

2-tier Client/Server Architecture

Two tier software architectures were built in 1980's based on the file server software architectural behavior projected mainly to improve usability through the support of form based user-friendly interfaces. It also provides increased scalability via supporting upto 100 users though the file server model supports only dozen of users. Often corporate data can be shared over homogenous environment enhancing flexible usage. This model necessitates minimal operator intervention mostly used in non-complex, non-time critical information processing systems. Eg: File Servers, Database Servers with Stored Procedure (Client/Server Architectures,2011).

Two tier paradigms consist of three components dispersed in two layers: client (suppllicant of services) and server (supplier of services) mentioned as below.

- **User System Interface** (session, text and dialog input, display management services)

- • **Processing Management** (process development and performance, process monitoring, process resource services)
- • **Database Management** (data and file services)

In this architecture User System Interface is entirely allocated to the client and the Database Management is placed on the server whereas the Processing Management is functionally dispersed on both client/server resulting two layers (Client/Server Architectures, 2011).

The application logic can reside on client-side, within a user interface or server-side within a database independently or on both sides. These applications can be simply built using *visual builder tools* helpful in developing applications for small-scale groupware entitled for decision support systems and in the creation of web publishing applications (Client/Server Architectures, 2011).

Components

- **Clients**
- **Servers**
- **Communication Networks**

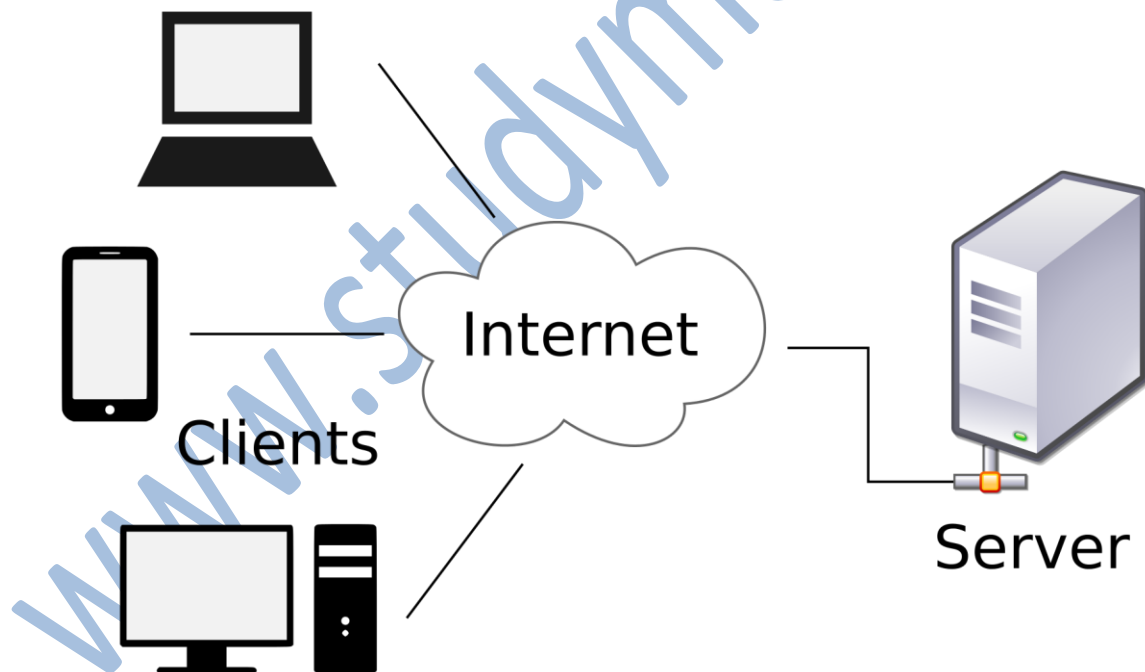
Clients

- Applications that run on computers
- Rely on servers for
 - Files
 - Devices
 - Processing power
- Example: E-mail client
 - An application that enables you to send and receive e-mail

Servers

- Computers or processes that manage network resources
 - Disk drives (file servers)
 - Printers (print servers)
 - Network traffic (network servers)
- Example: Database Server
 - A computer system that processes database queries

Communication Networks



Client Server Architecture Types

Two main forms of networking exist: peer-to-peer and client/server. Peer-to-peer users connect with other desktops on their network and allows its users to share printers, files and other resources via their network. Chances are if you have a home network, you and the other computers in your house interact through a peer-to-peer networking system.

The client/server model is designed for larger networks where users aren't localized and connect from a variety of outside places. Most of the websites you visit are based on the this setup. The client/server model consists of two major architecture types--the two-tier and the three-tier.

Two-Tier

- Two-tier is the simplest of the architecture types, consisting of only the server and the client application. The server, also known as the database, houses the information of a network, while the client requests to access the information. In a two-tier model, this request is direct, thus easy to develop and maintain. Though this architecture type is more common, it is usually only implemented in smaller, less demanding conditions.

Three-Tier

- The three-tier model adds another level to the two-tier setup. Instead of the client connecting directly to the server as it does with the two-tier type, the client application must make requests to the server through a server application. For example, if you were trying to access a website's database application. The database application would then send a request, on your behalf, to the main server. The main server would process your request and forward it back to the database application, which would then present the information to you.

Reference

- www.google.com
- www.wikipedia.com
- www.studymafia.org