**Client-Server Architecture**

**Centralized processing**

A host computer (often a mainframe) handles all processing, including input, output, data storage and retrieval. Predominant in late 1970s.

**Distributed processing**

A number of computers (minis, workstations, PCs, ...) handle all processing, They are distributed physically and connected thru a communications network. Use presently

**Cooperative processing**

A number of computers (minis, workstations, PCs, ...) handle all processing, They are distributed physically and connected thru a communications network.

Processing thru sharing of resources, transparently to the users

Use presently and in future

**Clients and Servers**

**Basic Definition**

* Server: provides services
* Client: requests for services
* Service: any resource
* (e.g., data, type definition, file, control, object, CPU time, display device, etc.)

**Typical Properties**

* It is up to the server to determine how to get the job done. A service request is about "what" is needed, and it often made abstractly (cf. MILs, ADTs --- Larch, Z)
* The ideal client/server software is independent of hardware or OS platform. The location of clients and servers are usually transparent to the user.
* A client may become a server; a server may become a client.
* A client/server system can be scaled with only a slight performance impact
* **Horizontally,** i.e., by adding/removing client workstations

**Vertically**, i.e., by migrating to a larger and faster server machines or multi-servers.

**Client/Server with File Servers**

**Centralized**

* Clients can reside in the same machine or separate machines (typically PCs)
* The client passes requests to the file server (software) for file records
* Requests can be either local or over a network.
* Indispensable for documents, images, drawings, and other large data objects

**As per drawing in class lecture both for centralized and distributed file server**

**Client/Server with Database Servers**

At present the majority of existing client/server-based software is to be found in the area of databases, and it is here that the greatest challenge to any corporation currently lies.

Events (violation of integrity constraints, temporal conditions, errors) trigger event handlers

**-> Implicit invocation, blackboards, events**

A DBMS also offers features for recovery and concurrency control

**As per drawing in class lecture both for centralized and distributed file server**

**Client/Server Communication**

**Sockets** provides interface over the networks to established connection, send request, response with reply and terminate connection.

**RPCs (Remote Procedure Calls)**

A transparent mechanism to give the client procedure the illusion that transports entity

10 steps to execute a RPC it is making a direct call on the distant server procedure as follows

1. *stub call*
2. *collect & pack the parameters(marshalling parameter into a msg);*
3. *send the call msg*
4. *pass the msg*
5. *parameter unmarshalling (unpack the msg nto parameters);*
6. *server proc returns*
7. *result marshalling( pack the result into a msg a sys call)*
8. *send the return msg*
9. *pass the msg*
10. *unmarshalling result marshalling (unpack the msg into result)*

***MOM (Message-Oriented Middleware)***

*Every DAD (Distributed Application Development) needs MOM*

*Many-to-many messaging via queues*

**Client/Server with Transaction Processing**

Transactions are a way to make ACID operations a general commodity

***Atomicity***

* *a transaction is an indivisible unit of work*
* *an all-or-nothing proposition*
* *all updates to a database, displays on the clients’ screens, message queues*

e.g., salary increase for all 1 million employees or none

**Consistency**

* a transaction is an indivisible unit of work
* integrity constraints (e.g., mgr.salaray > salary)

***Isolation***

* a transaction’s behavior not affected by other transactions running concurrently
* e.g., reserve a seat
* serialization technique

***Durability***

* persistence
* a transaction’s effects are permanent after it commits.

**Client/Server Groupware**

Support for business reengineering: - maximize profit

* helps manage (and track) the product thru its various phases collaborative/workgroup computing

**Five foundation technologies**

change the way people communicate with each other

1. Multimedia document management

from electronic imaging (scanning, digitization, display, storage and retrieval)

to document (component types: text, image, graphics, faxes, mail, voice clips, BBs)

1. Work flow: The "workflow river" carries the flow of work from port to port,

value being added along the way

1. E-mail
2. Conferencing
3. Scheduling

**Web Client/Server**

1. *The Web browser collects the data within the form, assembles it into a string of name/value pairs, specifies a POST method, the URL of the target program in the "cgi-bin" directory*
2. *The HTTP server receives the method invocation via a socket connection*
3. *The server parses the message to discover that it’s a POST for the "cgi-bin" program*
4. *The Common Gateway Interface (CGI ) program reads the environment variables*
5. *The HTTP server starts a CGI program*
6. *The CGI program receives the message body via the input pipe*
7. *The CGI program does some work, typically by interacting with a DBMS, TP Monitor*
8. *The CGI program returns the results via the output pipe*
9. *The HTTP server returns the results to the Web brower*