Multi-computer distributed systems

Mariam A. Salih
Introduction.

COMPUTER NETWORKS BASICS

Multicomputer distributed system

CLIENT/SERVER SYSTEMS

IPC (Inter-process Communication)

Sockets

Remote Procedure Call (RPC)

Middleware

A Client Server Framework
In previous we studied computing systems consisting of multiple processing units connected via some interconnection network.

There are two major factors that differentiate such systems: the processing units and the interconnection network that ties them together.

We learned that the processing units could communicate and interact with each other using either shared memory or message passing methods.

In this chapter we discuss network computing, in which the nodes are stand-alone computers that could be connected via a switch, local area network, or the Internet. The main idea is to divide the application into semi-independent parts according to the kind of processing needed.
**COMPUTER NETWORKS BASICS**

- **Wide area network (WAN);** A WAN connects a large number of computers that are spread over large geographic distances. It can span sites in multiple cities, countries, and continents.

- **Metropolitan area network (MAN);** The MAN is an intermediate level between the LAN and WAN and can perhaps span a single city.

- **Local area network (LAN);** A LAN connects a small number of computers in a small area within a building or campus.

- **System or storage area network (SAN).** A SAN connects computers or storage devices to make a single system.
Latency is an element that contributes to network speed. The term latency refers to any kind of delay typically incurred in processing of network data. A low latency connection is one that generally experiences small delay times, while a high latency connection generally suffers from long delays.
SPECIAL NETWORK TECHNOLOGIES
(Fast Ethernet and Gigabit Ethernet)

- **Fast Ethernet (100Base-T)**: is a high speed LAN that allows a computer to transmit or receive data at 100 Megabits per second (100 Mbps).

- **Gigabit Ethernet Fast Ethernet (1000Base-T)**: has become an attractive choice for corporate backbone networks and high-performance clusters of workstations.
The Fiber Distributed Data Interface (FDDI)

The FDDI specifies a 100 Mbps token-passing, dual-ring LAN using fiber-optic cable.

The FDDI is frequently used as high-speed backbone technology because of its support for high bandwidth and greater distances than copper.
SPECIAL NETWORK TECHNOLOGIES
(High-Performance Parallel Interface (HiPPI))

- **High-Performance Parallel Interface (HiPPI)**: The HiPPI is a point-to-point communication channel and it does not support multi-drop configurations.
- HiPPI is capable of transferring data at 800 Mbps using 32 parallel line or 1.6 Gbps over 64 parallel lines.
Asynchronous Transfer Mode (ATM) The ATM is a connection-oriented scheme that is suitable for both LANs and WANs.

- It transfers data in small fixed size packets called cells.
- It can handle multimedia in an integrated way.
- Cells are allowed to transfer using several different media such as both copper and fiber-optic cables.
- It is designed to permit high-speed data. The fastest ATM hardware can switch data at a gigabit rate.
SPECIAL NETWORK TECHNOLOGIES
(Scalable Coherent Interface (SCI))

- **Scalable Coherent Interface (SCI)** The SCI is an IEEE standard that is quite popular for PC clusters.
- It represents a point-to-point architecture with directory based cache coherence.
- It provides a cluster-wide shared memory system.
- A remote communication in SCI takes place as just part of a simple load or store process in a processor.
Multicomputer distributed system

- A Multicomputer distributed system is a network of autonomous computers that communicate with each other in order to achieve a goal.

- The computers in a distributed system are independent and do not physically share memory or processors. They communicate with each other using messages (pieces of information) transferred from one computer to another over a network.

- Messages can communicate many things: computers can tell other computers to execute a procedure with particular arguments, they can send and receive packets of data, or send signals that tell other computers to behave a certain way.
CLIENT/SERVER SYSTEMS

- In Client/Server distributed system where the application is divided into at least two parts: one or more servers perform one part and the other part is performed by one or more clients.

- Furthermore, the clients are connected to the servers by some kind of network.

- A client computer may do very little more than simply display data accessed from the server, or a more sophisticated client may run a full application, which uses data provided by the server.
Client/Server systems categorized as two-tier or three-tier.

- A two-tier system separates clients from servers: all clients are on one tier, and all servers are on the second tier. For example, client PCs may access a database on one or more servers.

- A three-tier system separates the clients from the servers as does a two-tier system, but in addition, servers are divided into two more tiers. The application servers fit into a middle level, called the second tier, and the database servers fit into a third level called tier 3. For example, client PCs might be connected to a web server (tier 2) that in turn accesses a database server (tier 3) to handle storage.
two-tier

three-tier
IPC (Inter-process Communication)

IPC (Inter-process Communication) Inter-process communication (IPC) is a set of techniques for the exchange of data among multiple threads in one or more processes. Processes may be running on one or more computers connected by a network.

- The method of IPC used may vary based on the bandwidth and latency of communication between the threads, and the type of data being communicated.
IPC
(Inter-process Communication)
diagram
Inter-process communication Protocol

- In a distributed application, two processes perform inter-process communication must agreed upon protocol.

- The specification of a protocol should include:
  1. the sequence of data exchange, which can be described using a time event diagram.
  2. the format of the data exchange at each step.
Sockets

- **Sockets** are used to provide the capability of making connections from one application running on one machine to another running on a different machine.

- A *socket abstraction consists of* (1) the data structure that holds the information needed for communication, and (2) the system calls that manipulate the socket structure.

- Once a socket is created, it can be used to wait for an incoming connection (**passive socket**), or can be used to initiate connection (**active socket**).
A client can establish an active connection to a remote server by creating an instance of a socket.

To establish a server connection and bind it to a particular port number, we should create an instance of a server socket.

A server socket listens on a TCP port for a connection from a client (passive socket). When a client connects to that port, the server accepts the connection.
Sockets

- Once the connection is established, the client and server can read from and write to the socket using input and output streams.
- Streams are ordered sequences of data that have a source (input stream), or destination (output stream).
- Once the client or server finishes using the socket, the socket structure is de-allocated.
Sockets

**CLIENT STEPS**
1. The client tries to establish a connection with the server.
2. When the connection is established, the client receives a data from the server.
3. The client process this data.

**SERVER STEPS**
1. The server waits for a connection from a client.
2. When the connection is established, the server sends a data to the client.
3. The server closes the connection.
Remote Procedure Call (RPC)

- Remote procedure call (RPC) is the basis of most client/server systems.

- Think of RPC as a procedure call where the procedure is located on a different computer than the caller. Thus, when the procedure is called, its parameters are passed (sent) via the network to the remote computer, and then the remote computer executes the procedure, returns the result(s), and continues on its way.
Steps in executing a remote procedure call

- The client calls a local procedure, called the client stub.
- Network messages are sent by the client stub to the remote system.
- Network messages are transferred by the kernel to the remote system via some protocol.
- A server stub, sometimes called the skeleton, receives the messages on the server.
- The server stub calls the server function (which, to the client, is the remote procedure)
Steps in executing a remote procedure call

- When the server function is finished, it returns to the server stub with its return values.
- The server stub converts the return values, if necessary, and marshals them into one or more network messages to send to the client stub.
- Messages get sent back across the network to the client stub.
- The client stub reads the messages from the local kernel.
- The client stub then returns the results to the client function, converting them from the network representation to a local one if necessary.
Remote Procedure Call (RPC)
Remote Procedure Call (RPC)

- The RPC can be constructed on top of sockets. That is, the socket mechanism can be used to pass parameters and the name of the procedure to be activated on the remote computer, and so on.
- The RPC can be **blocking** or **nonblocking**.
Remote Procedure Call (RPC)

- A **blocking RPC** means that the program that places the call is stopped in its tracks while waiting for a reply.

- The **nonblocking RPC**, allows the calling program to continue without waiting for a reply. In this case, the caller must explicitly ask for the reply at some later time, or else the return value will never get back to the caller.
Middleware

Middleware is an important part of client/server systems because it is an important intermediate layer of software, for the following reasons:

- It makes it possible for new systems to coexist with old systems, and cooperate different systems;
- It solves a number of interoperability problems because it can simultaneously convert formats and gain access without code rewriting;
- It isolates system components so that changes in one component have little effect on other components.
Middleware (MW)
A Client Server Framework

- Parallel applications can be designed using the client/server model. Where the **client acts as the master (supervisor)** while the **servers act as the slaves (workers)**.

- A client may divide a big application into several smaller problems that can be processed by multiple servers simultaneously and send their results to the client.

- The client assembles the results from each server and outputs the final result to the user.
Simple Quiz..