Unit-3

Network Operating System and Distributed Operating System

Loosely Coupled System:

It is a type of system in which, there is distributed memory instead of shared memory. In loosely coupled system, data rate is low rather than tightly coupled system. In loosely coupled multiprocessor system, modules are connected through MTS (Message transfer system) network.

In a loosely coupled system, hardware and software may interact but they are not dependent on each other.

In another meaning loosely coupled architecture or system means changes in one module / section that affect the other components and every module is somewhat independent of each other.

Tightly Coupled System:

It is a type of system in which, there is shared memory. In tightly coupled system, data rate is high rather than loosely coupled system.

It is a concept of system design and computing where every hardware and software components that are linked together in such manner that each component is dependent upon each other.

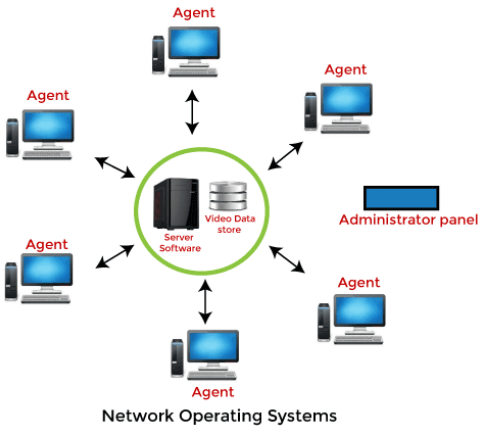
Tightly coupled architecture is fragile as the minor issue in one segment can bring the whole system down.

Network operating system (NOS)

A network operating system (NOS) is a computer operating system (OS) that is designed primarily to support workstations, personal computers and, in some instances, older terminals that are connected on a local area network (LAN). The software behind a NOS allows multiple devices within a network to communicate and share resources with each other.

Some popular network operating systems are Novell Netware, Windows NT/2000, Linux, Sun Solaris, UNIX, and IBM OS/2. The network operating system which was first developed is Novell Netware. It was developed in 1983.

The composition of hardware that typically uses a NOS includes a number of personal computers, a printer, a server and file server with a local network that connects them together. The role of the NOS is to then provide basic network services and features that support multiple input requests simultaneously in a multiuser environment.



Network Operating Systems

Some of the features of Network Operating System are to:

- It allows multiple computers to connect so that they can share data, files and hardware devices.
- Provide basic operating system features such as support for processors, protocols, automatic hardware detection and support multi-processing of applications.
- Provide security features such as authentication, logon restrictions and access control.
- Provide name and directory services.
- Provide file, print, web services and back-up services.
- Support Internetworking such as routing and WAN ports.
- User management and support for logon and logoff, remote access; system management, administration and auditing tools with graphical interfaces.
- It has clustering capabilities.

The **advantages** of network operating systems are as follows −

- Centralized servers are highly stable.
- Security is server managed.
- Upgrades to new technologies and hardware can be easily integrated into the system.
- Remote access to servers is possible from different locations and types of systems.

The **disadvantages** of network operating systems are as follows −

- High cost of buying and running a server.
- Dependency on a central location for most operations.
- Regular maintenance and updates are required.

Examples of network operating systems

True network operating systems are categorized as software that enhances the functionality of operating systems by providing added network features. A few examples of these network operating systems and their service providers are:

- Artisoft's LANtastic- This is a simple, user-friendly NOS that supports most PC operating systems.
- Banyan's VINES- This uses a client-server architecture to request specific functions and services.
- Novell's NetWare- This was the first network operating system to be released and is designed based on XNS protocol architecture.
- Microsoft's LAN Manager- This operates as a server application and was developed to run under the Microsoft OS. Now, most of the functionality of LAN Manager is included in the Windows OS itself.

In addition, some multi-purpose operating systems, such as Windows NT and Digital's OpenVMS come with capabilities that enable them to be described as a network operating system. Further, the most popular operating systems like Windows, Unix, Linux and Mac include built-in networking functions that may not require additional network services.

Services of Network Operating System(NOS)

A network operating system provides an environment in which users, who are aware of the multiplicity of machines, can access remote resources by either logging in to the appropriate remote machine or transferring data from the remote machine to their own machines. Currently, all general-purpose operating systems, and even embedded operating systems such as Android and iOS, are network operating systems.

Remote Login

An important function of a network operating system is to allow users to log in remotely. The Internet provides the ssh facility for this purpose. To illustrate, let's suppose that a user at Westminster College wishes to compute oncs.yale.edu, a computer that is located at Yale University. To do so, the user must have a valid account on that machine. To log in remotely, the user issues the command

ssh cs.yale.edu

This command results in the formation of an encrypted socket connection between the local machine at Westminster College and the ―cs.yale.edu‖ computer. After this connection has been established, the networking software creates a transparent, bidirectional link so that all characters entered by the user are sent to a process on ―cs.yale.edu‖ and all the output from that process is sent back to the user. The process on the remote machine asks the user for a login name and a password. Once the correct information has been received, the process acts as a proxy for the user, who can compute on the remote machine just as any local user can.

Remote File Transfer

Another major function of a network operating system is to provide a mechanism for remote file transfer from one machine to another. In such an environment, each computer maintains its own local file system. If a user at one site (say, cs.uvm.edu) wants to access a file located on another computer (say, cs.yale.edu), then the file must be copied explicitly from the computer at Yale to the computer at the University of Vermont.

The Internet provides a mechanism for such a transfer with the file transfer protocol (FTP) program and the more private secure file transfer protocol (SFTP) program. Suppose that a user on ―cs.uvm.edu‖ wants to copy a Java program Server.java that resides on ―cs.yale.edu.‖ The user must first invoke the sftp program by executing

sftp cs.yale.edu

The program then asks the user for a login name and a password. Once the correct information has been received, the user must connect to the subdirectory where the file Server.java resides and then copy the file by executing

get Server.java

In this scheme, the file location is not transparent to the user; users must know exactly where each file is. Moreover, there is no real file sharing, because a user can only copy a file from one site to another. Thus, several copies of the same file may exist, resulting in a waste of space. In addition, if these copies are modified, the various copies will be inconsistent.

FTP also provides a way to allow a user who does not have an account on the Yale computer to copy files remotely. This remote copying is accomplished through the ―anonymous FTP‖ method, which works as follows. The file to be copied (that is, Server.java) must be placed in a special subdirectory (say, ftp) with

the protection set to allow the public to read the file. A user who wishes to copy the file uses the ftp command. When the user is asked for the login name, the user supplies the name ―anonymous‖ and an arbitrary password.

Implementation of the FTP mechanism is similar to ssh implementation. A daemon on the remote site watches for requests to connect to the system's FTP port. Login authentication is accomplished, and the user is allowed to execute transfer commands remotely.

These include the following:

• **get** — Transfer a file from the remote machine to the local machine.

• **put** — Transfer from the local machine to the remote machine.

• **ls or dir** — List files in the current directory on the remote machine.

• **cd**— Change the current directory on the remote machine.

There are also various commands to change transfer modes (for binary or ASCII files) and to determine connection status.

Network operating system is that in which two or more computers are connected to each other to share resources with each other. **There are two types(implementation) of network operating system**.

- **Peer to peer network**
- **Client and server network**

Peer-To—Peer Network (P2P)

This operating systems allow users to share network resources saved in a common, accessible network location. In this architecture, all devices are treated equally in terms of functionality. Peer-to-peer usually works best for small to medium LANs and is cheaper to set up.

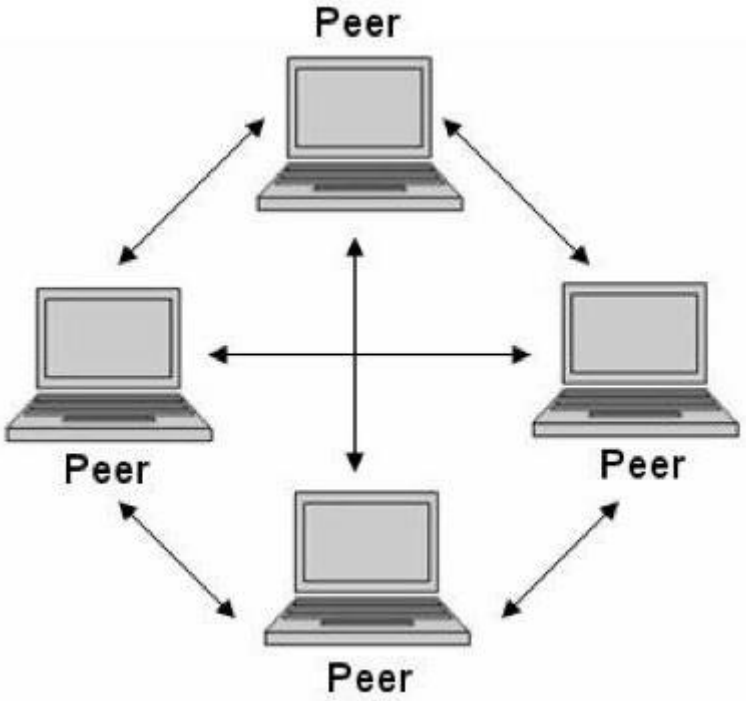


Fig Peer to peer network operating system

This is a type of network in which all computers are connected to each other. It is inexpensive to setup. Files are placed on any computer and can be accessed by any other computer in the network. This type of network is best for small or medium size organization.

or

In its simplest form, a peer-to-peer (P2P) network is created when two or more PCs are connected and share resources without going through a separate server computer. A P2P network can be an ad hoc connection—a couple of computers connected via a Universal Serial Bus to transfer files.

Computers (peers) can share peripheral devices like printer, hard drive, or scanner with each other. Peer to peer network is made between computers through modem, switch or hub. The connection between peers can be wired or wireless. In wired connection coaxial cables, copper cable, and fibre optics can be used. Note that the central server is not involved in making this type of network and peers communicate with each other independently.

Characteristics of Peer to Peer Computing

- Peer to peer networks are usually formed by groups of a dozen or less computers. These computers all store their data using individual security but also share data with all the other nodes.
- The nodes in peer to peer networks both use resources and provide resources. So, if the nodes increase, then the resource sharing capacity of the peer to peer network increases. This is different than client server networks where the server gets overwhelmed if the nodes increase.
- Since nodes in peer to peer networks act as both clients and servers, it is difficult to provide adequate security for the nodes. This can lead to denial of service attacks.
- Most modern operating systems such as Windows and Mac OS contain software to implement peer to peer networks.

Advantages of Peer to Peer Computing

Some advantages of peer to peer computing are as follows:

- Each computer in the peer to peer network manages itself. So, the network is quite easy to set up and maintain.
- In the client server network, the server handles all the requests of the clients. This provision is not required in peer to peer computing and the cost of the server is saved.
- It is easy to scale the peer to peer network and add more nodes. This only increases the data sharing capacity of the system.
- None of the nodes in the peer to peer network are dependent on the others for their functioning.

Disadvantages of Peer to Peer Computing

Some disadvantages of peer to peer computing are as follows:

- It is difficult to backup the data as it is stored in different computer systems and there is no central server.

- It is difficult to provide overall security in the peer to peer network as each system is independent and contains its own data.

Examples of peer to peer network:-

Torrent: Torrent is a big example of a P2P network. In torrent, all the computers are connected to each other on the internet. One computer can upload any file in the network and other computers start downloading the files. Also, every computer can upload parts of the file if that computer has already downloaded some chunks of the file.

A computer attached to the LAN: In home or in a small office, computers can make peer to peer network with each other and share data and resources with each other.

Client-Server Systems

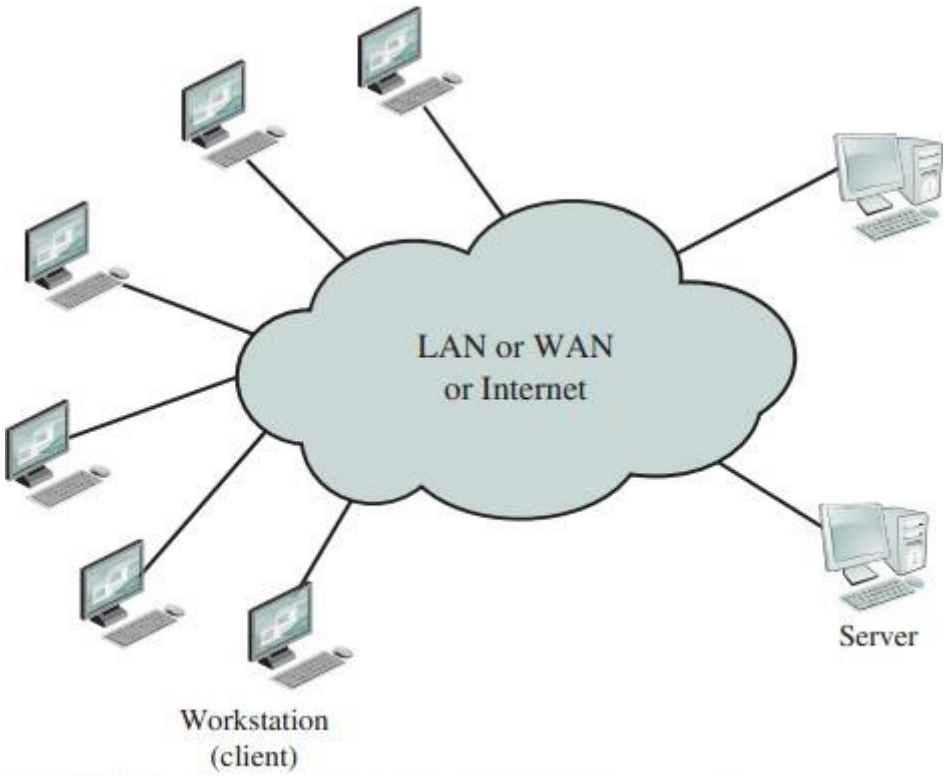


Figure 16.1 Generic Client/Server Environment

Client/server network operating systems provide users with access to resources through a server. In this architecture, all functions and applications are unified under one file server that can be used to

execute individual client actions regardless of physical location. Client/server tends to be most expensive to implement and requires a large amount of technical maintenance. An advantage to the client/server model is that the network is controlled centrally, makes changes or additions to technology easier to incorporate.

This is type of network in which there is a server that is attached to client computer. So one computer is behaving as a center server controlling and managing other computers.

In client server computing, the clients request a resource and the server provides that resource. A server may serve multiple clients at the same time while a client is in contact with only one server. Both the client and server usually communicate via a computer network but sometimes they may reside in the same system.

Characteristics of Client Server Computing

The salient points for client server computing are as follows:

- The client server computing works with a system of request and response. The client sends a request to the server and the server responds with the desired information.
- The client and server should follow a common communication protocol so they can easily interact with each other. All the communication protocols are available at the application layer.
- A server can only accommodate a limited number of client requests at a time. So it uses a system based to priority to respond to the requests.
- Denial of Service attacks hindera servers ability to respond to authentic client requests by inundating it with false requests.
- An example of a client server computing system is a web server. It returns the web pages to the clients that requested them.

Difference between Client Server Computing and Peer to Peer Computing

The major differences between client server computing and peer to peer computing are as follows:

- In client server computing, a server is a central node that services many client nodes. On the other hand, in a peer to peer system, the nodes collectively use their resources and communicate with each other.
- In client server computing the server is the one that communicates with the other nodes. In peer to peer to computing, all the nodes are equal and share data with each other directly.
- Client Server computing is believed to be a subcategory of the peer to peer computing.

Advantages of Client Server Computing

The different advantages of client server computing are:

- All the required data is concentrated in a single place i.e. the server. So it is easy to protect the data and provide authorisation and authentication.
- The server need not be located physically close to the clients. Yet the data can be accessed efficiently.
- It is easy to replace, upgrade or relocate the nodes in the client server model because all the nodes are independent and request data only from the server.
- All the nodes i.e clients and server may not be build on similar platforms yet they can easily facilitate the transfer of data.

Disadvantages of Client Server Computing

The different disadvantages of client server computing are:

- If all the clients simultaneously request data from the server, it may get overloaded. This may lead to congestion in the network.
- If the server fails for any reason, then none of the requests of the clients can be fulfilled. This leads of failure of the client server network.
- The cost of setting and maintaining a client server model are quite high.

Example NOS: UNIX OS

- UNIX is a computer operating system.
- An operating system is the program that controls all the other parts of a computer system, both the hardware and the software. It allocates the computer's resources and schedules tasks. It allows you to make use of the facilities provided by the system. Every computer requires an operating system.
- UNIX is a multi-user, multi-tasking operating system. Multiple users may have multiple tasks running simultaneously. This is very different from PC operating systems such as MS-DOS or MS-Windows (which allows multiple tasks to be carried out simultaneously but not multiple users).
- UNIX is a machine independent operating system. Not specific to just one type of computer hardware. Designed from the beginning to be independent of the computer hardware.

- UNIX is a software development environment. Was born in and designed to function within this type of environment.
- The "UNIX" trademark, previously owned by AT&T and then deeded to UNIX Systems Laboratories (USL), an AT&T subsidiary, passed to Novell when it acquired USL. After a brief period of negotiations with rival Unix vendors, namely, Sun Microsystems, Santa Cruz Operation, International Business Machines, and Hewlett-Packard, Novell granted exclusive licensing rights of the UNIX trademark to X/Open Co. Ltd., an Open Systems industry standards branding agent based in the United Kingdom.

UNIX Components

- Kernel
  - The core of the UNIX system. Loaded at system start up (boot). Memory-resident control program.
  - Manages the entire resources of the system, presenting them to you and every other user as a coherent system. Provides service to user applications such as device management, process scheduling, etc.
  - Example functions performed by the kernel are:
    - managing the machine's memory and allocating it to each process.
    - scheduling the work done by the CPU so that the work of each user is carried out as efficiently as is possible.
    - accomplishing the transfer of data from one part of the machine to another
    - interpreting and executing instructions from the shell
    - enforcing file access permissions
  - You do not need to know anything about the kernel in order to use a UNIX system. These details are provided for your information only.
- Shell
  - Whenever you login to a Unix system you are placed in a shell program. The shell's prompt is usually visible at the cursor's position on your screen. To get your work done, you enter commands at this prompt.
  - The shell is a command interpreter; it takes each command and passes it to the operating system kernel to be acted upon. It then displays the results of this operation on your screen.
  - Several shells are usually available on any UNIX system, each with its own strengths and weaknesses.
  - Different users may use different shells. Initially, your system adminstrator will supply a default shell, which can be overridden or changed. The most commonly available shells are:
    - Bourne shell (sh)
    - C shell (csh)
    - Korn shell (ksh)

        - TC Shell (tcsh)
        - Bourne Again Shell (bash)
    - Each shell also includes its own programming language. Command files, called "shell scripts" are used to accomplish a series of tasks.
- Utilities
    - UNIX provides several hundred utility programs, often referred to as commands.
    - Accomplish universal functions
        - editing
        - file maintenance
        - printing
        - sorting
        - programming support
        - online info
        - etc.
    - Modular: single functions can be grouped to perform more complex tasks

Features of UNIX

High reliability, scalability and powerful features make UNIX a popular operating system, according to Intel. Now beyond its 40th year as of 2010, UNIX is the backbone of many data centers including the Internet. Big players using UNIX include Sun Microsystems, Apple Inc., Hewlett-Packard and AT&T, which is the original parent company of UNIX. The Open Group owns all UNIX specifications and the trademark, which are freely accessible and available over the Internet.

Multitasking and Portability

The main features of UNIX include multiuser, multitasking and portability capabilities. Multiple users access the system by connecting to points known as terminals. Several users can run multiple programs or processes simultaneously on one system. UNIX uses a high-level language that is easy to comprehend, modify and transfer to other machines, which means you can change language codes according to the requirements of new hardware on your computer. You, therefore, have the flexibility to choose any hardware, modify the UNIX codes accordingly and use UNIX across multiple architectures.

The Kernel and the Shell

The hub of a UNIX operating system, the kernel manages the applications and peripherals on a system. Together, the kernel and the shell carry out your requests and commands. You communicate with your system through the UNIX shell, which translates to the kernel.

When you turn on your terminal, a system process starts that overlooks your inputs. When you enter your password, the system associates the shell program with your terminal. The shell allows you to customize options even if you are not technically savvy. For example, if you partially type a command, the shell anticipates the command for which you are aiming and displays the command for you. The UNIX shell is a program that gives and displays your prompts and, in conjunction with the kernel, executes your commands. The shell even maintains a history of the commands you enter, allowing you to reuse a command by scrolling through your history of commands.

Files and Processes

All the functions in UNIX involve either a file or a process. Processes are executions of programs, while files are collections of data created by you. Files may include a document, programming instructions for the system or a directory. UNIX uses a hierarchical file structure in its design that starts with a root directory--signified by the forward slash (/). The root is followed by its subdirectories, as in an inverted tree, and ends with the file. In the example "/Demand/Articles/UNIX.doc," the main directory "Demand" has a subdirectory "Articles," which has a file "UNIX.doc."

Some other features of UNIX are:

- **Portability:** The system is written in high-level language making it easier to read, understand, change and, therefore move to other machines. The code can be changed and complied on a new machine. Customers can then choose from a wide variety of hardware vendors without being locked in with a particular vendor.
- **Machine-independence:** The System hides the machine architecture from the user, making it easier to write applications that can run on micros, mins and mainframes.
- **Multi-Tasking:** Unix is a powerful multi-tasking operating system; it means when a active task in in process, there can be a simultaneous background process working too. Unix handles these active and background threads efficiently and manages the system resources in a fair-share manner.
- **Multi-User Operations:** UNIX is a multi-user system designed to support a group of users simultaneously. The system allows for the sharing of processing power and peripheral resources, white at the same time providing excellent security features.

- **Hierarchical File System:** UNIX uses a hierarchical file structure to store information. This structure has the maximum flexibility in grouping information in a way that reflects its natural state. It allows for easy maintenance and efficient implementation.
- **UNIX shell:** UNIX has a simple user interface called the shell that has the power to provide the services that the user wants. It protects the user from having to know the intricate hardware details.
- **Pipes and Filters:** UNIX has facilities called Pipes and Filters which permit the user to create complex programs from simple programs.
- **Utilities:** UNIX has over 200 utility programs for various functions. New utilities can be built effortlessly by combining existing utilities.

Distributed Operating System

A distributed system is a collection of processors that do not share memory or a clock. Instead, each node has its own local memory. The nodes communicate with one another through various networks, such as high-speed buses and the Internet.

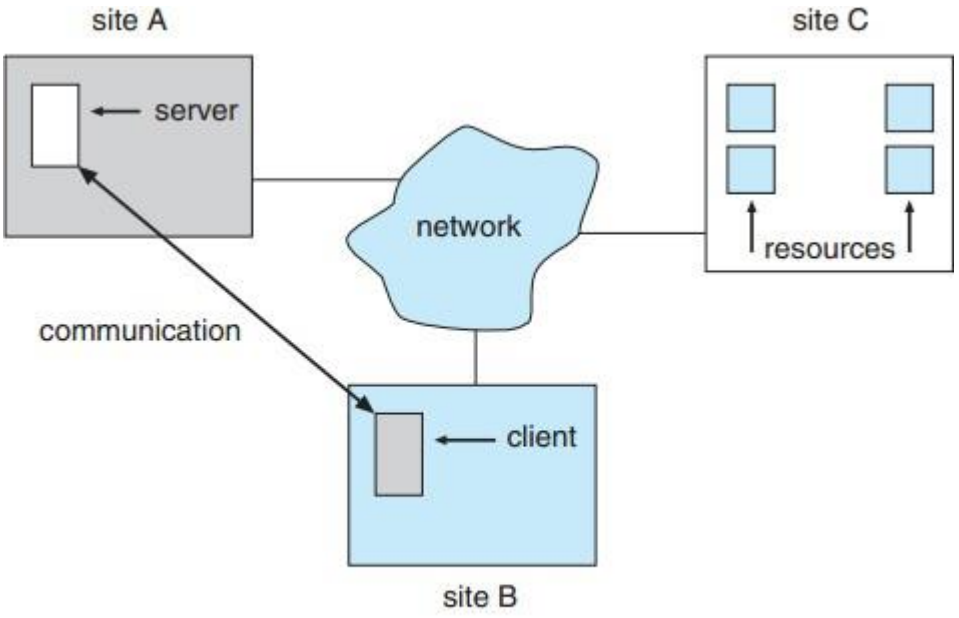


Figure 17.1 A distributed system.

Advantages of Distributed Systems

1. **Resource Sharing**

If a number of different sites (with different capabilities) are connected to one another, then a user at one site may be able to use the resources available at another. For example, a user at site A may be using a laser printer located at site B. Meanwhile, a user at B may access a file that resides at A. In general, resource sharing in a distributed system provides mechanisms for sharing files at remote sites, processing information in a distributed database, printing files at remote sites, using remote specialized hardware devices (such as a supercomputer), and performing other operations.

2. **Computation Speedup**

If a particular computation can be partitioned into sub-computations that can run concurrently, then a distributed system allows us to distribute the sub-computations among the various sites. The sub-computations can be run concurrently and thus provide computation speedup. In addition, if a particular site is currently overloaded with jobs, some of them can be moved to other, lightly loaded sites. This

movement of jobs is called load sharing or job migration . Automated load sharing, in which the distributed operating system automatically moves jobs, is not yet common in commercial systems.

3. **Reliability**

If one site fails in a distributed system, the remaining sites can continue operating, giving the system better reliability. If the system is composed of multiple large autonomous installations(that is, general-purpose computers), the failure of one of them should not affect the rest. If, however, the system is composed of small machines, each of which is responsible for some crucial system function (such as the web server or the file system), then a single failure may halt the operation of the whole system.

The failure of a site must be detected by the system, and appropriate action may be needed to recover from the failure.

4. **Communication**

When several sites are connected to one another by a communication network, users at the various sites have the opportunity to exchange information. At a low level, messages are passed between systems, much as messages are passed between processes in the single-computer message system

Disadvantages of distributed operating systems

1) Security problem due to sharing
2) Some messages can be lost in the network system
3) Bandwidth is another problem if there is large data then all network wires to be replaced which tends to become expensive
4) Overloading is another problem in distributed operating systems
5) If there is a database connected on local system and many users accessing that database through remote or distributed way then performance become slow
6) The databases in distributed operating is difficult to administrate then single user system

Below given are some of the examples of distributed operating systems:

l. IRIX operating system; is the implementation of UNIX System V, Release 3 for Silicon Graphics multiprocessor workstations.

2. DYNIX operating system running on Sequent Symmetry multiprocessor computers.

3. AIX operating system for IBM RS/6000 computers.
4. Solaris operating system for SUN multiprocessor workstations.
5. Mach/OS is a multithreading and multitasking UNIX compatible operating system.
6. OSF/1 operating system developed by Open Foundation Software: UNIX compatible.

Difference between Network Operating System and Distributed Operating System

Basis For Comparison	Network Operating System	Distributed Operating System
Objective	Network Operating System's main objective is to provide the local services to remote client	Distributed Operating System's main objective is to manage the hardware resources.
Architecture	2-tier client/server architecture.	N-tier client/server architecture.
Level of Transparency	Low	High
Communication	In Network Operating System, Communication takes place on the basis of files.	In Distributed Operating System, Communication takes place on the basis of messages and share memory.
Scalability	Network Operating System is more scalable than Distributed Operating System.	Distributed Operating System is less scalable than Network Operating System.
Fault Tolerance	In Network Operating System, fault tolerance is less.	While in Distributed Operating System, fault tolerance is high.
Type of Operating System	In Network Operating System, All nodes can have different operating system.	While in Distributed Operating System, All nodes have same operating system.
Resource Management	Handled at each node.	Global central or distributed management.
Ease of implementation	High	Low
Rate of autonomy	Less	High

Distributed Denial of Service Attack (DDos)

A distributed denial-of-service (DDoS) attack is an attack in which multiple compromised computer systems attack a target, such as a server, website or other network resource, and cause a denial of service for users of the targeted resource. The flood of incoming messages, connection requests or malformed packets to the target system forces it to slow down or even crash and shut down, thereby denying service to legitimate users or systems.

DDoS attacks have been carried out by diverse threat actors, ranging from individual criminal hackers to organized crime rings and government agencies. In certain situations, often ones related to poor coding, missing patches or generally unstable systems, even legitimate requests to target systems can result in DDoS-like results.

How DDoS attacks work

In a typical DDoS attack, the assailant begins by exploiting a vulnerability in one computer system and making it the DDoS master. The attack master system identifies other vulnerable systems and gains control over them by either infecting the systems with malware or through bypassing the authentication controls (i.e., guessing the default password on a widely used system or device).

A computer or networked device under the control of an intruder is known as a zombie, or bot. The attacker creates what is called a command-and-control server to command the network of bots, also called a botnet. The person in control of a botnet is sometimes referred to as the botmaster (that term has also historically been used to refer to the first system "recruited" into a botnet because it is used to control the spread and activity of other systems in the botnet).

Botnets can be comprised of almost any number of bots; botnets with tens or hundreds of thousands of nodes have become increasingly common, and there may not be an upper limit to their size. Once the botnet is assembled, the attacker can use the traffic generated by the compromised devices to flood the target domain and knock it offline.

Types of DDoS Attacks

There are many types of DDoS attacks. Common attacks include the following:

1. **Traffic attacks**: Traffic flooding attacks send a huge volume of TCP, UDP and ICPM packets to the target. Legitimate requests get lost and these attacks may be accompanied by malware exploitation.
2. **Bandwidth attacks:** This DDos attack overloads the target with massive amounts of junk data. This results in a loss of network bandwidth and equipment resources and can lead to a complete denial of service.

3. **Application attacks:** Application-layer data messages can deplete resources in the application layer, leaving the target's system services unavailable.

How to stop a DDoS attack

1. Identify the DDoS attack early

If you run your own servers, then you need to be able to identify when you are under attack. That's because the sooner you can establish that problems with your website are due to a DDoS attack, the sooner you can stop the DDoS attack.

To be in a position to do this, it's a good idea to familiarize yourself with your typical inbound traffic profile; the more you know about what your normal traffic looks like, the easier it is to spot when its profile changes.

2. Overprovision bandwidth

It generally makes sense to have more bandwidth available to your Web server than you ever think you are likely to need. That way, you can accommodate sudden and unexpected surges in traffic that could be a result of an advertising campaign, a special offer or even a mention of your company in the media.

3. Defend at the network perimeter (if you run your own web server)

There are a few technical measures that can be taken to partially mitigate the effect of an attack -- especially in the first minutes -- and some of these are quite simple. For example, you can:

- rate limit your router to prevent your Web server from being overwhelmed
- add filters to tell your router to drop packets from obvious sources of attack
- timeout half-open connections more aggressively
- drop spoofed or malformed packages
- set lower SYN, ICMP, and UDP flood drop thresholds

4. Call your ISP or hosting provider

The next step is to call your ISP (or hosting provider if you do not host your own Web server), tell them you are under attack, and ask for help. Keep emergency contacts for your ISP or hosting provider readily available so you can do this quickly. Depending on the strength of the attack, the ISP or hoster may already have detected it – or they may themselves start to be overwhelmed by the attack.

5. Call a DDoS mitigation specialist

For very large attacks, it's likely that your best chance of staying online is to use a specialist DDoS mitigation company. These organizations have large-scale infrastructure and use a variety of technologies, including data scrubbing, to help keep your website online. You may need to contact a DDoS mitigation company directly, or your hosting company or service provider may have a partnership agreement with one to handle large attacks.