UNIT 1: DISTRIBUTED OPERATING SYSTEM

Unit Structure

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- About object-based distributed object systems
- Characteristics models of Content Delivery Network
- About Processor Pool Model

1.1 Introduction

Distributed Operating System as shown in fig 1.1 is a model where distributed applications are running on multiple computers linked by communications. Such type of operating system is an advancement of network operating system which is basically designed for higher communication and integration levels for certain machines which are on network. It will appear as simple centralized operating system to its users where it can handle and perform multiple independent central processing units operations.

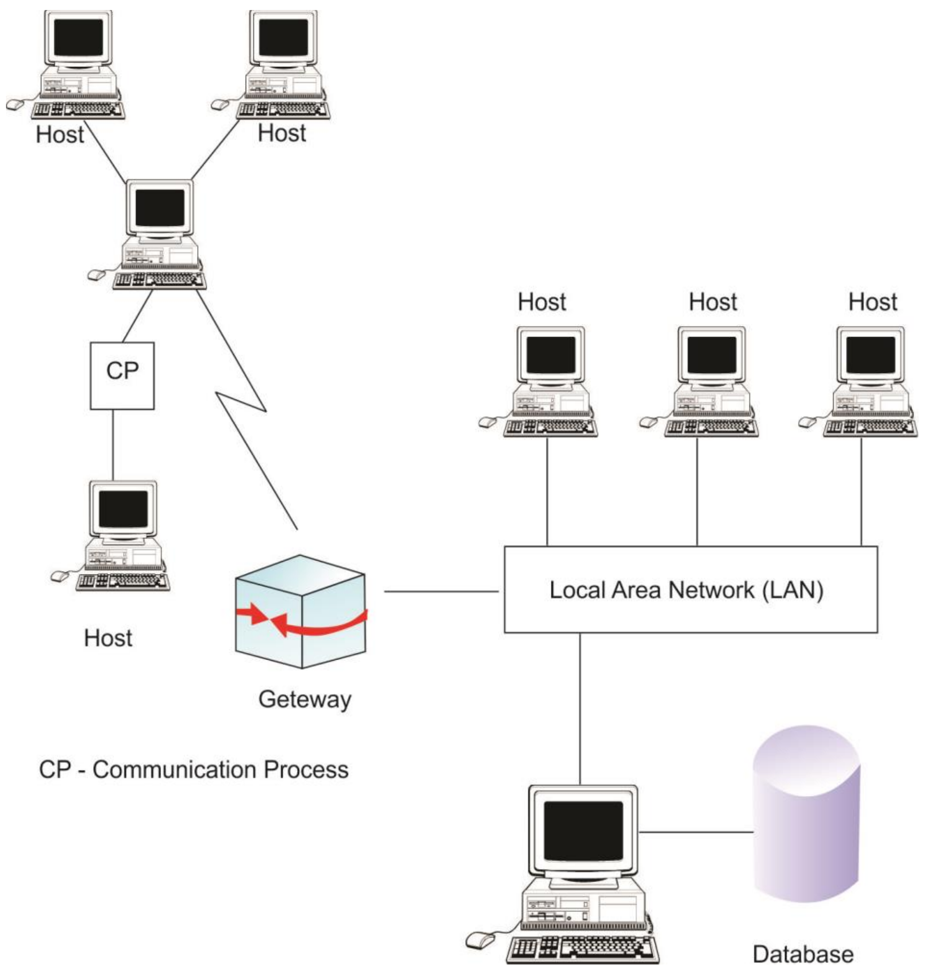

Fig 1.1 Distributed Operating System

These systems are referred as loosely coupled systems where each processor has its own local memory and processors communicate with one another through various communication lines, such as high speed buses or telephone lines. By loosely coupled systems, we mean that such computers possess no hardware connections at the CPU - memory bus level, but are connected by external interfaces that run under the control of software. The Distributed O/S involves a collection of independent computer systems, competent of communicating and cooperating with each other from side to side by LAN / WAN. A Distributed OS provides a virtual machine abstraction to its users and wide sharing of resources like as computational capacity, I/O and files etc.

1.2 Need for Distributed OS

The ODP standards, and this text, assume a model where distributed applications are running in multiple processes in multiple computers linked by communications. The application programmer will be supported by a programming environment and run-time system that will make many aspects of distribution in the system transparent. For instance the programmer may not have to worry about where the parts of the application are running this can all be taken care of, if required; this is called location transparency.

There is another approach to supporting applications in a distributed system that is by using a distributed operating system. On every computer system with an operating system the O/S provides an interface which the programs use to obtain services, such as input and output.

Distributed systems are potentially more reliable than a central system because if a system has only one instance of some critical component, such as a CPU, disk, or network interface, and that component fails, the system will go down. When there are multiple instances, the system may be able to continue in spite of occasional failures. In addition to hardware failures, one can also consider software failures. Distributed systems allow both hardware and software errors to be dealt with.

A distributed system is a set of computers that communicate and collaborate each other using software and hardware interconnecting components. Multiprocessors (MIMD computers using shared memory architecture), multicomputer connected through static or dynamic interconnection networks (MIMD computers using message passing architecture) and workstations connected through local area network are examples of such distributed systems.

A distributed system is managed by a distributed operating system. A distributed operating system manages the system shared resources used by multiple processes, the process scheduling activity (how processes are allocating on available processors), the communication and synchronization between running processes and so on. The software for parallel computers could be also tightly coupled or loosely coupled. The loosely coupled software allows computers and users of a distributed system to be independent each other but having a limited possibility to cooperate. An example of such a system is a group of computers connected through a local network. Every computer has its own memory, hard disk. There are some shared resources such files and printers. If the interconnection network broke down, individual computers could be used but without some features like printing to a non-local printer.

Check your progress 1

1. Distributed Operating System is also called as?

a. loose coupled systems	c. fat coupled systems
b. tight coupled systems	d. thin coupled systems

2. In Distributed Operating System, the processor has its own__________.

a. data	c. memory
b. information	d. all

3. The distributed systems comprises of critical component like______.

a. hard disk	c. memory
b. CPU	d. modem

1.3 Architecture of Distributed OS

The architecture of distributed O/S comprises off our styles such as:-

- Layered architecture
- Object-based architecture
- Data-centered architecture
- Event-based architecture

The architecture is organized into logically different components, and distributed components over various machines.

The layered architecture is an arrangement of client system model as shown in fig 1.2.

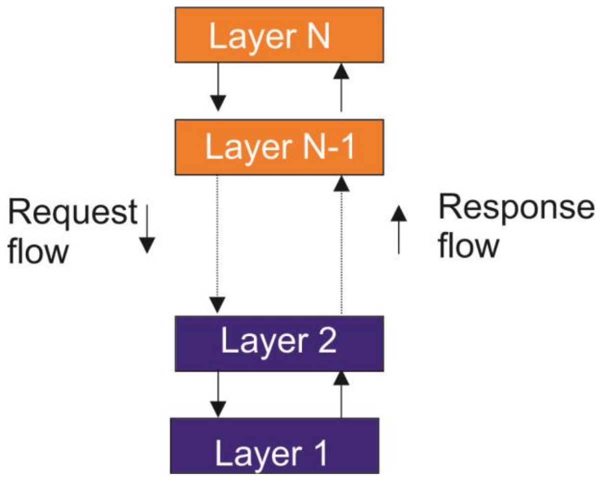

Fig 1.2 layered architecture arrangement

Fig 1.3 shows an object-based style for distributed object systems which is less structured having component similar as object and containing connector as RPC or RMI.

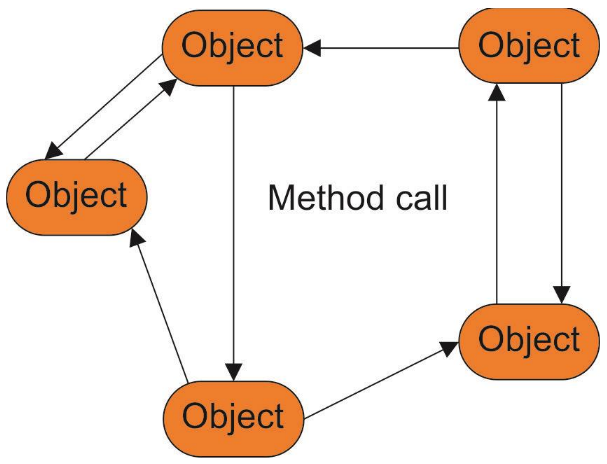

Fig 1.3 object-based distributed object systems

In this, the decoupling processes in space and also time led to various alternative styles as shown in fig 1.4.

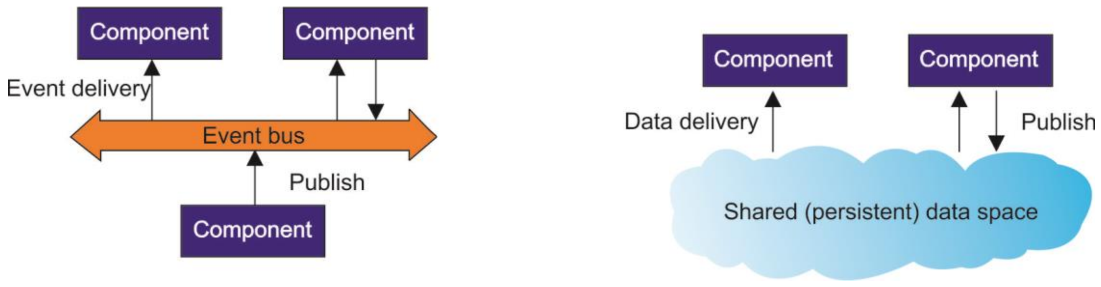

Fig 1.4 Styles in object-based distributed object systems

In case of publish/subscribe event-based architecture as shown in fig 1.5 carries-

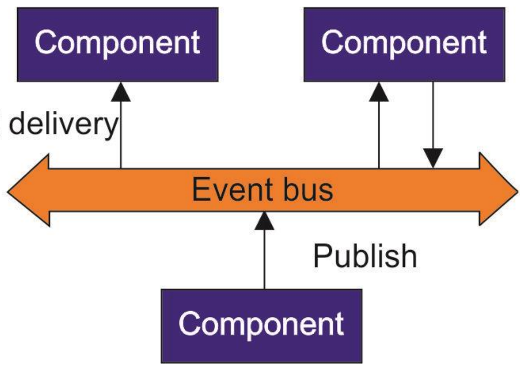

Fig 1.5 publish/subscribe event-based architecture

- Publish-subscribe
- Broadcast
- Point-to-point

Here the decouples sender & receiver acts as asynchronous communication. In the event driven architecture (EDA), the activities such as production, detection, consumption of reaction results in occurrence of various events.

Here we explain an event as an important change in state. The benefit of such type of architecture is that they are loosely coupled since they require no explicitly that belongs to each other.

In case of shared data space which a data cantered + event-based architecture as shown in fig 1.6, contains-

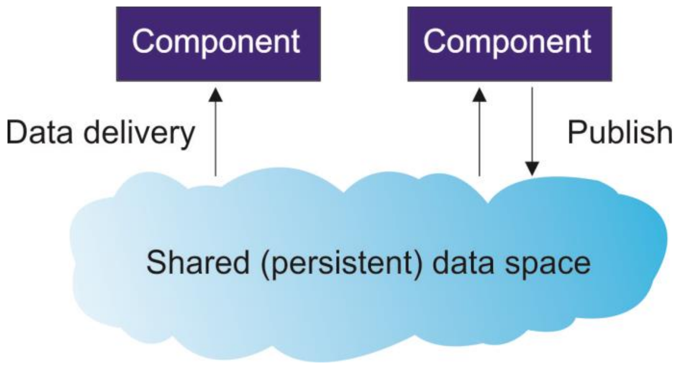

Fig 1.6 data cantered

Such type of data centred architecture access and update data that are kept for data-centred system. In this, the processes communicate or exchange the data or information mainly by reading and modifying data in some shared repository. It is seen that a many web-based distributed systems communicate among themselves through the use of shared Web based data services.

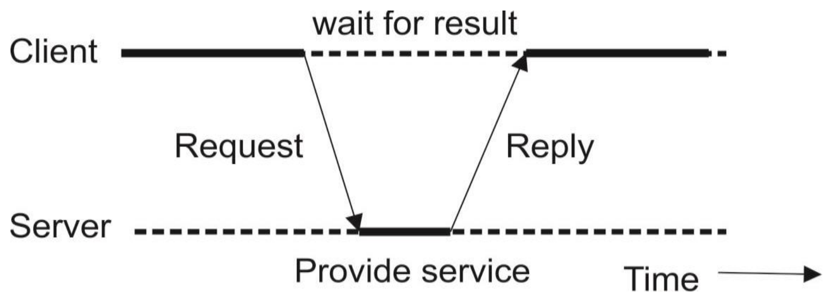

Fig 1.7 Client Server Model

Further it is seen that in basic Client Server Model as shown in fig 1.7, there exists certain main features that-

- There are certain process that offers the server services.
- There are certain process which uses client services.
- There seems that the clients and servers can be on different machines.
- There follows that clients request/reply model be used for services.
- There exists certain synchronous communication where request or reply protocol takes place.
- In case of LANs, it works with connectionless protocol as they are unreliable.
- In case of WANs, the communication exists as connection oriented TCP/IP as they are reliable.

Since many years it is seen that there exists high level development growth in peer-to-peer systems. These can be:-

- Structured P2P: where the nodes are arranged having a particular distributed data structure.
- Unstructured P2P: where the nodes have arbitrarily selected other close nodes.
- Hybrid P2P: where some nodes are presented as special functions in a good organized manner.

In case of structured P2P systems, we see that:-

Nodes are arranged in a structure overlay network like logical ring as shown in fig 1.8 and develop a specific node that are responsible for services based only on their ID.

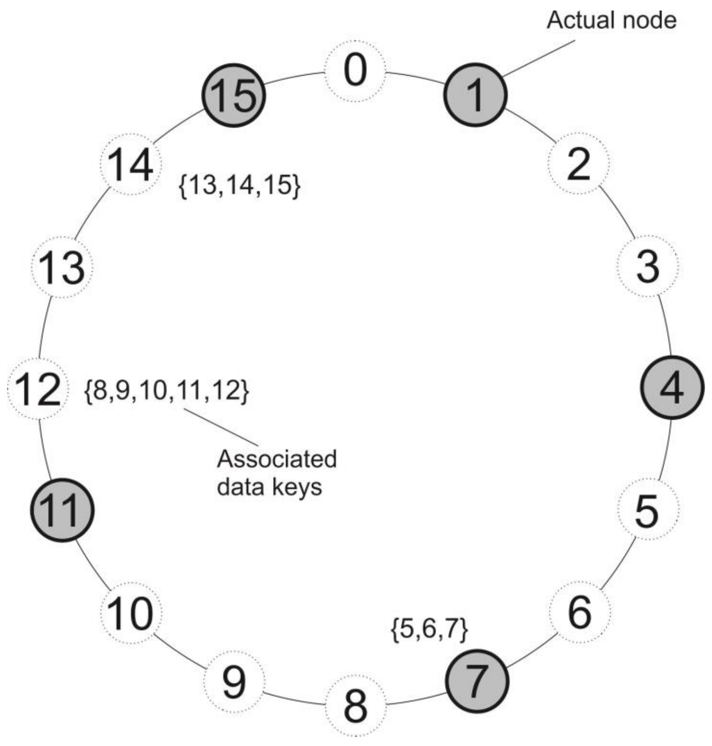

Fig 1.8 Nodes arranged logical ring network

Here the idea is to use a distributed hash table (DHT) to arrange the nodes. The earlier hash functions transform a key to hash value that can be used as an index in hash table:

- Keys are unique –each key shows an object to keep in the table;
- Hash function value is there to insert an object in hash table and can get anytime.

It is seen that in DHT, the data objects and nodes are both given a key which hashes to a random number from a very big identifier space. In this, mapping function gives objects to nodes that are based on hash function value. In such case a lookup based on hash function value will go back to the network address of the node which keeps all requested object.

In case of unstructured P2P systems, it is seen that:-

- The information or data depends on randomized algorithms for developing an overlay network.
- There occur several unstructured P2P systems that will try to maintain a random graph.

The basic idea of unstructured P2P systems is that every node is required to contact a randomly selected other node which:-

- allows each peer to maintain a partial view of network that carries n nodes
- will select each node P periodically and also node Q from its partial view
- will exchange P and Q information and members from their respective partial views

The hybrid architecture of Client-server combined with P2P where edge server architectures used for Content Delivery Networks is shown in fig 1.9

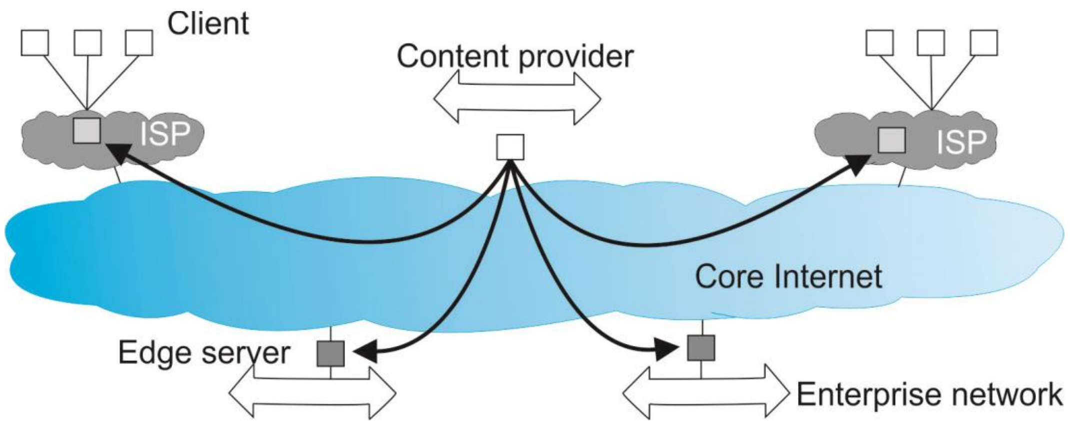

Fig 1.9 Content Delivery Network

In another hybrid architecture of C/S with P2P which is a Bit Torrent, where users cooperate in file distribution is shown in fig 1.10

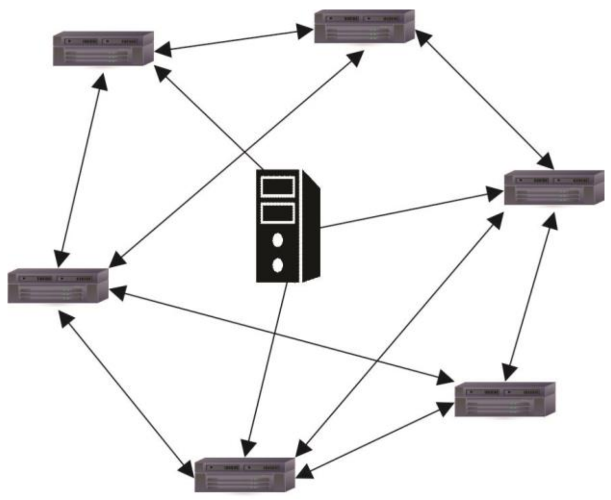

Fig 1.10 hybrid architecture of C/S with P2P

It is seen that once a node finds a path to download a file, it will joins a swarm of downloader which will parallel get file chunks from the source, and will further distribute such chunks among them. It can be shown by fig 1.11.

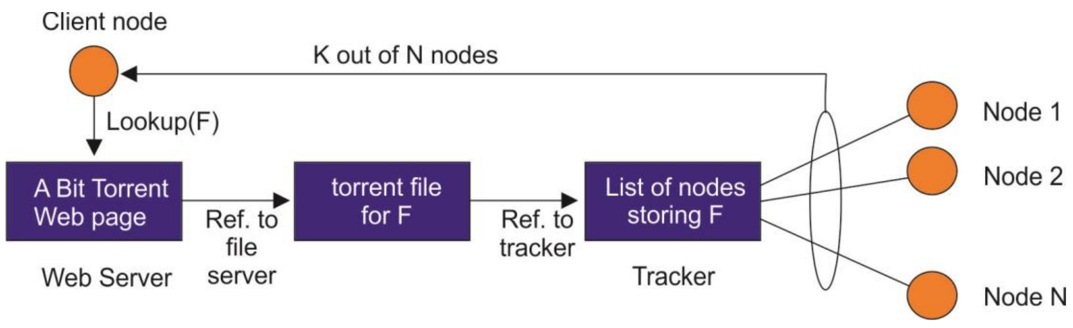

Fig 1.11 architecture of distributed O/S

The architecture of distributed O/S comprises of multiple computers which:

- Will not share memory or Clock.
- Will communicate by exchanging of messages over a network.

The advantages of distributed system are:-

- They can easily share resource related to hardware and software.
- They have better performance that led to rapid response time and higher system throughput.
- They are capable of improved reliability and availability.
- They can expand by modular expansion.
- They have low set up price and have better performance.
- They are traditional time-sharing systems.

The distributed operating system appears as a centralized OS for a single machine that can run on multiple independent computers. They are clear with the transparency when they are handled by users.

The distributed operating system has certain drawbacks:

- They lack in latest and current global knowledge.
- They are poor in naming.
- They possess low scalability.
- They are much compatible.

- They have less process synchronization.
- They have week resource management.
- They are not secure.
- They possess poor structuring of OS.

There exist numerous levels of compatibility:

1. **Binary level -** They are restrictive if same binary is to be executed on all computers.
2. **Execution level -** In case of an execution level, if similar source code gets compiled then they can execute on all computers.
3. **Protocol level -** In case of protocol level, when each computer can perform on different OS, then they:
    - support common protocols for interaction
    - least restrictive

The structure of distributed operating system comprises of -

Monolithic kernel

In this, every computer OS kernel contains necessary services.

Such type of kernel is not advisable for distributed systems.

Collective kernel

In this, the structure of microkernel or the nucleus runs on many computers.

In this the OS services can be done as individual processes which are not necessarily for all computers.

Here the microkernel supports interaction among processes by way of messages.

Object-oriented OS

In this type of structure, the services performed by OS are implemented in shape of an object.

Check your progress 2

1. In an object-based style system, there exists a connector as____.

   a. RPC

   b. RIM

   c. RCP

   d. PCR

2. Which is not an activity of event driven architecture?

   a. production

   b. organisation

   c. consumption

   d. detection

3. The shared data space architecture is__________.

   a. data cantered architecture

   b. event-based architecture

   c. data cantered + event-based architecture

   d. none

4. In which peer-to-peer system the nodes are arranged having a particular distributed data structure.

   a. Structured P2P

   b. Unstructured P2P

   c. Hybrid P2P

   d. all

1.4 Models of distributed system

There are various types of distribute operating system models such as:-

- Workstation-server Model - Workstation may be a standalone system or a part of a network
- Processor-pool Model - Provides processing power on a demand basis

- Integrated Hybrid Model - Workstations used as processor pools

Workstation-server Model

The workstation model is a basic arrangement where system comprises of workstations that will be high end personal computers that are spread across the building or campus and are joined or connected through high speed LAN. This type of arrangement is shown in fig 1.12.

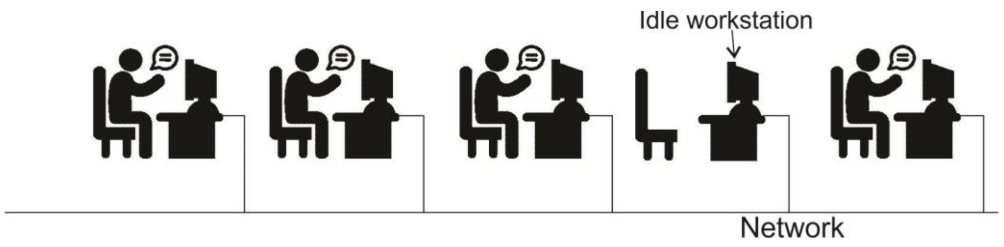

Fig 1.12 workstation server model

Here we see that some of the workstations in offices are particularly for single user, whereas others can be for public during the course of a day. We see that in such cases, a workstation can either have single user logged into it or will remain idle.

It is found that in some systems, the workstations contain local disks and some are without disk. Such type of computers is called as diskless workstations where as workstations containing a disk is often known as diskful or disky workstations. It is found that the workstations having no disk, will store file in the one or more remote file servers. These workstations will requests for read and write of files that are sent to file server which after performing sends the file back.

Such type of diskless workstations is found in universities and companies for several reasons. The workstation with small or slow disks is normally more expensive as compared to one or two file servers that carries huge, fast disks and accessed over the LAN.

Apart from this, the diskless workstations are popular as their maintenance is easy. When in this case, a compiler comes out, and then system administrator can easily install it on small number of file servers in respective machines. In this case taking backup of files and maintain the hardware is somewhat easy with centrally placed hard disk.

Finally, it is seen that a diskless workstations will supply symmetry and flexibility. Here the user can walk up to any workstation in system and can easily logged in. It is examined that when all files are stored on local disks, then by accessing any workstation will be easier to get files as compared to getting your own. If the workstations contain private disks, such disks can be used in either of ways:

1. Paging and temporary files.
2. Paging, temporary files, and system binaries.
3. Paging, temporary files, system binaries, and file caching.
4. Complete local file system.

From above, the first design is based on observation that since it is easier to keep all user files on single file servers, disks also requires paging for temporary files. This model is used only for paging and files that are temporary, unshared, and can be leftover during end of login session.

The next arrangement is an alternative of the first one as in this, the local disks keeps the executable programs like compilers, text editors and electronic mail handlers. If a program is called, it will be obtained from the local disk as a replacement for file server which lowers the network load. As such programs rarely changes, they get installed on local disks and can be stored for future use. On obtaining the new release of system program, these will be shown on all machines.

The third option uses local disks for open a cache. In this, users can download files from file servers to its own disks which will be read and write by him and can again be uploaded at the end of login session. The idea of this arrangement is to keep long-term storage in a particular place, but reduce network load by keeping files locally at the time they are used.

The last option shows that every machine will ideally have its own self-contained file system that can be mounted or accessed by other machines file systems. The idea is that every machine is on the whole self-contained and that contact with the outside world is inadequate. This organization makes available a uniform as well as guaranteed reply time for the user further more position little load on the network. The disadvantage is to facilitate sharing is additional difficult, and the resulting system is largely close to network operating system as compared to a true transparent distributed operating system.

The advantages of the workstation model are multiple and comprehensible. The model is undoubtedly easy to comprehend. Users have a predetermined amount of committed computing power, more over accordingly guaranteed response time. Complicated graphics programs can be extremely fast, in view of the fact that they can have straight access to the screen. Every user has a great degree of independence and be able to distribute his workstation's assets as he sees fit. Local disks put in to this independence, furthermore make it likely to carry on working to a smaller or better degree even in the countenance of file server collides.

Processor Pool Model

Even though by means of idle workstations put in a little computing power to the system, it does not deal with supplementary primary issues. An optional move towards this is to build a processor pool, which is rack full of CPUs in machine room that can be animatedly owed to users on order. The processor pool comes near as shown in Fig. 1.13.

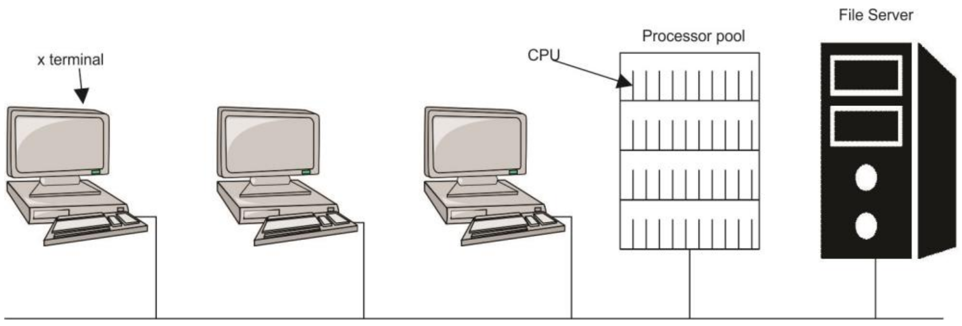

Fig 1.13 processor pool model

In its place, giving users personal workstations, this model gives high-performance graphics terminals like X terminals. This design is based on the inspection that what numerous users actually would like is high-quality graphical interface as well as high-quality performance. Theoretically, it is greatly faster to usual timesharing as compared to personal computer model, even though it is built with up to date technology involving low cost microprocessors.

The inspiration for the processor pool thought comes as of taking the diskless workstation proposal a step extra. If the file system be able to centralise in a little number of file servers to increase economies of scale, it have to be possible to do the similar thing for computer servers. Besides putting all the CPUs in a large rack in the machine room, power supply as well as other packaging costs is able to abridged, giving more computing power for a specified amount of money. In addition, it authorizes the use of cheaper X terminals, and decouples the amount of users as of number of workstations. The model in addition allocate for simple incremental growth. If the computing pack increases by 10%, you can immediately buy 10% additional processors along with them in the pool.

In result, we are converting all computing power into idle workstations with the purpose of to access vigorously. Users can be allocate as several CPUs as they need for little periods, after which they are revisit to the pool as a result that the other users can have them. There is no idea of possession here as all the processors fit in equally to everyone.

The biggest fight for centralizing the computing power in a processor pool comes as of queuing theory. A queuing scheme is a circumstance in which users produce accidental requests for work from a server. When the server is full of activity, the users queue for service as well as process in turn. Frequent examples of queuing systems are:-

- Bakeries
- Airport check-in counters
- Supermarket check-out counters

The basics of such system are shown in the Fig. 1-14.

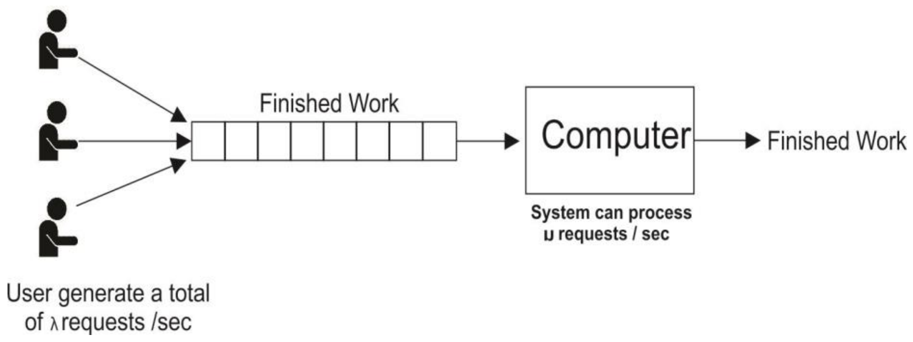

Fig 1.14 queuing systems

Queuing systems are helpful for the reason that they can be easily modelled analytically. Allow us call the whole input rate requests per second, as of all the users combined. Allow us call the rate at which the server can practice the requests. For steady operation, we should have >. If the server be able to handle 100requests/sec, other than the users continuously generate 110requests/sec, then the queue will produce with no bound.

Integrated Hybrid Model

A probable negotiation is to give each user with a personal workstation plus to have a processor pool in addition. While this solution is further costly than either a clean workstation model or a clean processor pool model, having the advantages of both.

Interactive work can be completed on workstations, giving certain reply. Idle workstations, on the other hand, are not exploiting, making for a simpler system design. They are now left idle. In its place, all non-interactive procedures run on the processor pool, because they do all serious computing in all-purpose. This model makes available for fast interactive response, an efficient use of resources, and a straightforward design.

Check your progress 3

1. A workstation server model is a_______.

   a. single system c. multiple system

   b. two systems d. all

2. Which model provides processing power on request?

   a. Processor-pool Model c. Integrated Hybrid Model

   b. workstation server model d. none

3. The disk in workstations can be used by____.

   a. Paging c. System binaries

   b. Temporary files d. all

1.5 Let Us Sum Up

In this unit we have learned:

- That a Distributed Operating System is a model where applications are running on multiple computers linked by communications.
- It is studied that a workstation model is a basic arrangement where system comprises of workstations which are high end personal computers spread across the building or campus and are joined or connected through high speed LAN.
- A distributed operating system is an extension of the network operating system.
- Queuing systems are helpful as they can be easily modelled analytically.

1.6 Answers for Check Your Progress

Check your progress 1

Answers: (1-a), (2-c), (3-b)

Check your progress 2

Answers: (1-a), (2-b), (3-c), (4-a)

Check your progress 3

Answers: (1-d), (2-a), (3-d)

1.7 Glossary

1. **Structured P2P -** where the nodes are arranged having a particular distributed data structure.
2. **Unstructured P2P -** where the nodes have arbitrarily selected other close nodes.
3. **Hybrid P2P -** where some nodes are presented as special functions in a good organized manner.
4. **Workstation-server Model -** Workstation may be a standalone system or a part of a network.
5. **Processor-pool Model -** Provides processing power on a demand basis.
6. **Integrated Hybrid Model -** Workstations used as processor pools.

1.8 Assignment

Design a Processor-pool Model in your institute.

1.9 Activities

Create an activity on Unstructured P2P.

1.10 Case Study

Is you institute carries Workstation-server Model.

1.11 Further Readings

1. Distributed Systems, Principles and Paradigms by Tanenbaum.
2. Distributed Systems, Concepts and Design by Coulouris, Dollimore, Kindberg.