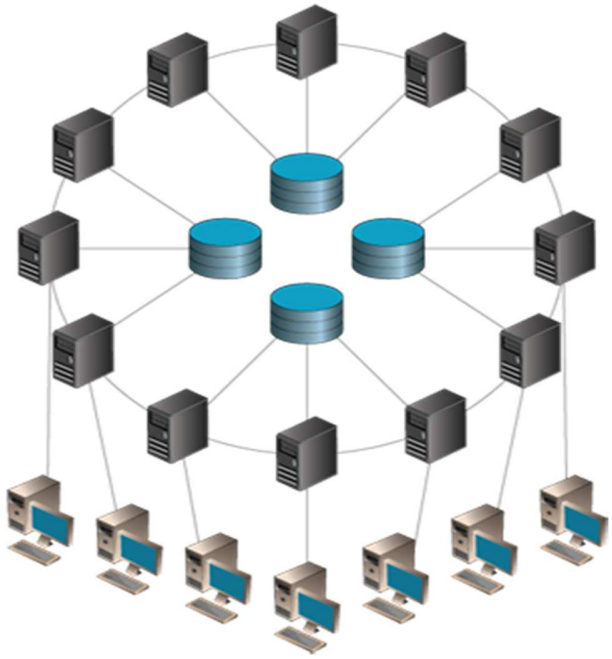
International Journal of Research (IJR) Vol-1, Issue-9, October 2014 **ISSN 2348-6848**

Distributed Operating System

Shilpa Yadav; Tanushree & Yashika Arora

A Distributed operating system is software over collection of communicating, networked, independent and with physically separate computational nodes. Each of the individual nodes holds one specific software subset of the global aggregate operating system. Each of the subset is a composite of the two specific service provisioners. The first is a universal minimal kernel, or microkernel, which directly controls that particular node's hardware. The Second is higher-level collection of a system management component which coordinates node's individual and the collaborative activities. These components consider microkernel functions and support user applications.

The collection of management component and microkernel work together. They support system goals of integrating the multiple resources and the processing functionality into a stable and an efficient system. This absolute integration of individual nodes into global system is called as transparency, or a single system image; which describes the illusion provided to the users of global system's appearance as single computational entity.



A distributed Operating System provides all the essential services and the functionalities required in an Operating System, adding the attributes and appropriate configurations which allows to support the additional requirements like increased scale and availability. For a user, a distributed Operating System works in a manner which is similar to a single-node, i.e. monolithic operating system. Although it consists of multiple nodes, it seems to users and applications as a single-node.

Distributed Operating System is a kind of model where the distributed applications are running on numerous computers linked by

connections. A distributed operating system is an expansion of the network operating system which supports higher levels of communication and unification of the machines on the network.

For users this system seems an ordinary centralized operating system but it runs on multiple and independent CPUs (Central Processing Unit).

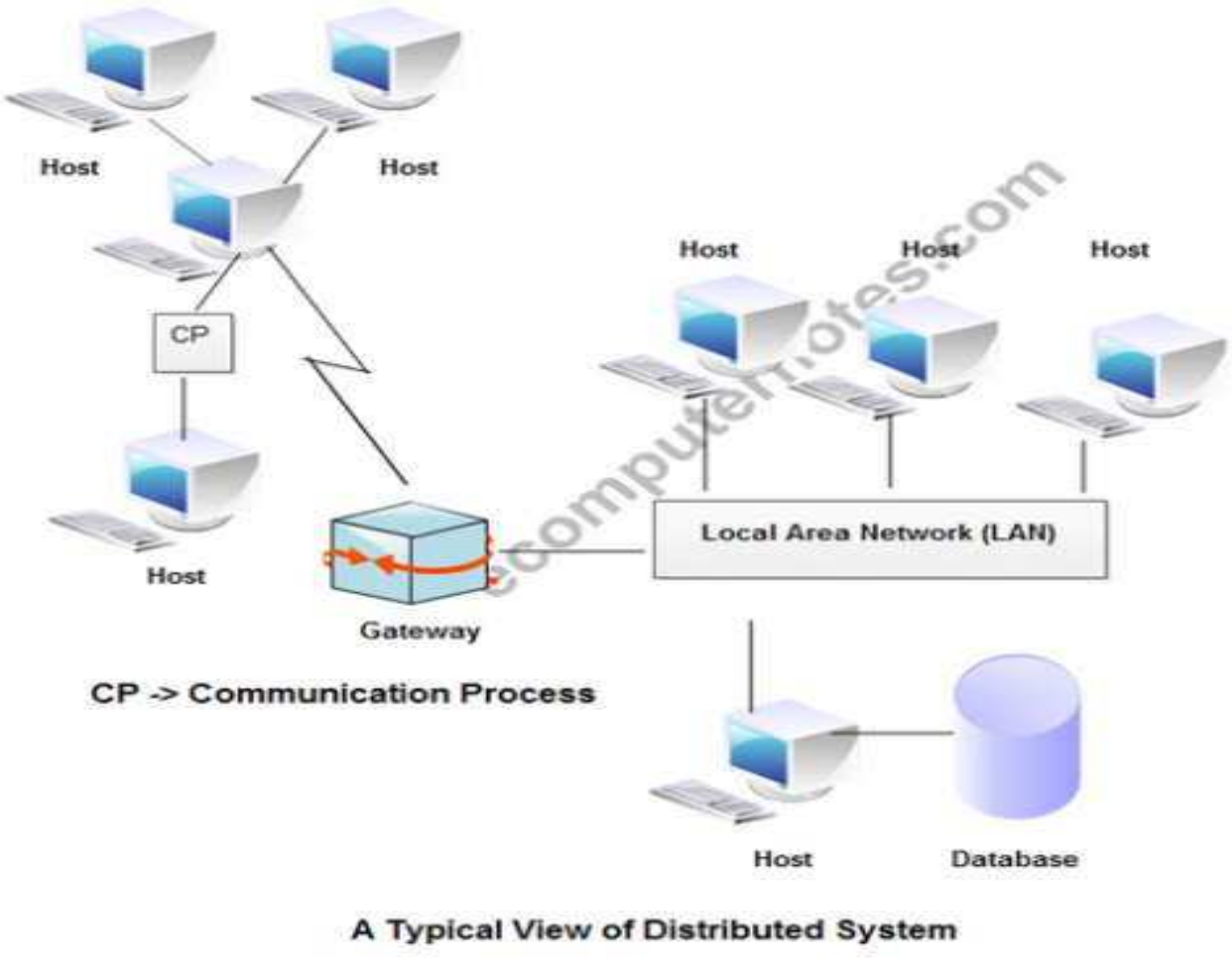


Figure 1: A sample of Distributed System (Source: ecomputernotes.com)

These systems are known as loosely coupled systems where each single processor has its own local memory and the processors communicate with each another through many communication lines, like high speed buses and telephone lines. The 'loosely coupled systems', mean that the computers possess no hardware connections at Central Processing Unit - memory bus level, but are linked by external interfaces which runs under the control of a software.

The Distributed Operating System involves collection of autonomous computer systems, which are capable of communicating and combining with each other through a Local Area Network / Wide Area Network. A Distributed Operating System gives a virtual machine abstraction to its users and a wide sharing of resources such as computational capacity, Input/Output and files etc.

The structure shown contains a set of independent computer systems and workstations which are connected via communication systems, but this structure doesn't make it a distributed system because it is a software, and not a hardware, and this is the thing which determines whether a system is distributed or not.

The end user of a genuine distributed system should not know that on which machine their programs are running and where the files are stored. LOCUS and MICROS are best examples of a distributed operating systems.

LOCUS operating system makes it possible to access the local and distant files in a uniform manner. This feature enables a user to log on any of the node of the network and to use the resources in a network without the referral of his/her location.

The MICROS operating system provides sharing of the resources in an automated manner. The jobs were allotted to the different nodes of whole system to balance the load on different nodes.

Given below are some of the examples of a distributed operating system:

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

2. DYNIX operating system working on Sequent Symmetry multiprocessor computer.

3. AIX operating system for the IBM RS/6000 computers.

4. Solaris operating system for the SUN multiprocessor workstations.

5. Mach/OS is both multithreading and multitasking - UNIX compatible operating system;

6. OSF/1 operating system was developed by Open Foundation Software: UNIX compatible software.

Distributed systems provides some of the advantages:

1 Sharing resources.

2 Computation speedup.

3 Communication

4 Reliability.

Distributed systems are potentially much more reliable as compared to a central system because if a system has only single instance of some vital component, such as CPU, hard disk, or a network interface, and if that component fails, then the system will go down. When there are numerous instances, the system might be able to continue despite of occasional failures. In addition to the hardware failures, one could also consider the software failures. Distributed systems allows both the hardware and software errors to be dealt with.

Figure 2: Advantages of the Distrubuted System

Item	Description
Economics	Microprocessors offer a better price/performance than mainframes
Speed	A distributed system may have more total computing power than a mainframe
Inherent distribution	Some applications involve spatially separated machines
Reliability	If one machine crashes, the system as a whole can still survive
Incremental growth	Computing power can be added in small increments

Multiprocessors (MIMD computers which use shared memory architecture), multicomputers which are connected through static or dynamic connection networks (MIMD computers use message passing architecture) and the workstations are connected through a local area network are the examples of a distributed systems.

Distributed system is organized by a distributed operating system. A distributed operating system organizes the system shared resources which are used by many processes, the process scheduling task (how processes are allocated on the processors which are available), the communication and synchronization in the middle of running processes and so on. The software for the parallel computers could be tightly coupled or loosely coupled. The loosely coupled software allows the computers and end users of distributed system to be independent of each other but have a limited probability to cooperate. An example of such system is a group of computers which connected along a local network. Each computer has its own memory. There are some of the shared resources such as files and printers. If the interconnection network breaks down, then the individual computers could be used but then we will not have some of the features like printing to a non-local printer.

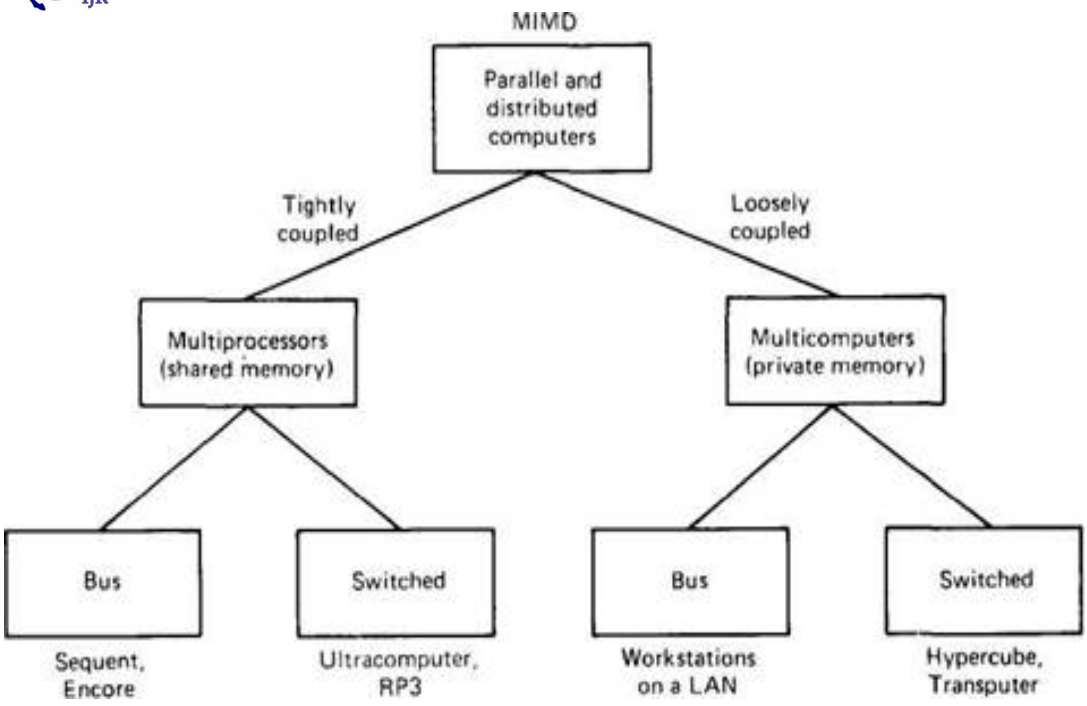
HARDWARE CONCEPTS:

All the distributed systems consist of many Central Processing Unit (CPU) , there are various different ways hardware could be regulated, especially in terms of how they communicate and how they are interconnected.

There are various classification schemes for multiple Central Processing Unit computer systems which have been advanced over the years, but not a single of them have been really caught on and been widely adopted. Possibly the most usually cited taxonomy is Flynn's (1972).Despite the fact, it is fairly elementary. What Flynn did is?? He focused on two characteristics that he considered important. One was the number of instruction streams and the other characteristic was the number of data streams. A computer which has single instruction stream and single data stream is known as SISD. All the traditional uniprocessor i.e., those having only one CPU falls in the category of SISD, ranging from the personal computers(PCs) to large mainframes.

The second category is SIMD i.e. Single Instruction Multiple Data stream. This kind refers to the array processors with single instruction unit which fetches an instruction, and after that commands many data units to carry out in parallel, and each with its own data. These kind of machines are useful for a computation which repeats the identical calculation on numerous sets of data. Like: Addition of all the elements of 64 unassociated vectors. Some of the supercomputers are SIMD.

The third category is MISD i.e. multiple instruction and single data stream. No familiar computer fits into this model.



And finally, comes the MIMD, which substantially means a group of distinct computers, each having its own program counter, data and program. All the distributed systems are MIMD, and so this categorization system is not exceedingly useful for our purposes.

And Flynn stopped here, after knowing about SISD, SIMD, MISD and MIMD.

Now we talk about Multiple Instruction Multiple Data stream and divide all MIMD computers into two categories: One category contains those which have shared memory, generally called **multiprocessors,** and the other category is of those which sometimes do not have memory and are generally called **multicomputers**. The vital difference is that in a multiprocessor, we have a single virtual address space which is shared by all the Central Processing Units (CPUs). If a CPU writes, for example, value 44 to address 1000, and other CPU is eventually reading from the address 1000 will get the value 44. All machines share the same memory.

In variance, in a multicomputer, machine has its private memory. If a CPU writes the value 44 to address 1000, while another CPU reads the address 1000 it will get the value which was there before. The write of 44 will not affect *its* memory at all. A usual example of a multicomputer is collection of personal computers (PCs) connected by a network.

Each of this classification can be further categorized based on the framework of the interconnection network. In Figure we describe the two categories as Switched and Bus**.** By bus we indicate that it has a single network, cable, backplane, bus or

other medium which connects all the machines.

Reference:

[1] Mullender, S. J., Van Rossum, G., Tananbaum, A. S., Van Renesse, R., & Van Staveren, H. (1990). Amoeba: A distributed operating system for the 1990s. *Computer*, *23*(5), 44-53.

[2] Barak, A., Guday, S., & Wheeler, R. G. (1993). *The MOSIX distributed operating system: load balancing for UNIX*. Springer-Verlag New York, Inc..

[3] Walker, B., Popek, G., English, R., Kline, C., & Thiel, G. (1983, October). The LOCUS distributed operating system. In *ACM SIGOPS Operating Systems Review* (Vol. 17, No. 5, pp. 49-70). Acm.

[4] Tanenbaum, A. S., Van Renesse, R., Van Staveren, H., Sharp, G. J., & Mullender, S. J. (1990). Experiences with the Amoeba distributed operating system. *Communications of the ACM*, *33*(12), 46-63.

[5] Kaashoek, M. F., & Tanenbaum, A. S. (1991, May). Group communication in the Amoeba distributed operating system. In *Distributed Computing Systems, 1991., 11th International Conference on* (pp. 222-230). IEEE.

[6] Rozier, M., Abrossimov, V., Armand, F., Boule, I., Gien, M., Guillemont, M., ... & Neuhauser, W. (1991). Overview of the chorus distributed operating systems. In *Computing Systems*.