Jisy Raju
Assistant Professor, CE Cherthala

CS407 Distributed Computing

Module 1

Evolution of Distributed Computing -Issues in designing a distributed system- Challenges- Minicomputer model –Workstation model - Workstation-Server model–
Processor - pool model- Trends in distributed systems

1.1 Evolution of Distributed Computing

At the very beginning, one computer could only do one particular task at a time (batch processing). Then multiprogramming introduced, which can execute multiple programs but there is only one processor, there can be no true simultaneous execution of different programs. If we need multiple tasks to be done in parallel, we need to have multiple computers running in parallel. Multiprocessor run multiple programs simultaneously by sharing the same memory,so it is tightly coupled system. But running them parallel was not enough for building a truly distributed system since it requires a mechanism to communicate between different computers.

Distributed Computing is a field of computer science that studies distributed systems. A distributed system is a model in which components located on networked computers communicate and coordinate their actions by passing messages. Distributed system is said to be loosely coupled because each processor has its own local memory.

Characteristics of Distributed System

- **Concurrency:** Tasks carry out independently
- **No global clock**:Each system has its own clock and it coordinate their actions by exchanging messages
- **Independent failures**:when some systems fail, others may not know, does not stop the running of the whole system

Example Of Distributed Systems

1. **The Internet**
2. **Mobile Computing**
3. **Intranet**
4. **Multiplayer online game**

1. The Internet: A vast interconnected collection of computer networks of many different types.

Programs running on the computers connected to it interact by passing messages, employing a

common means of communication

2. Mobile Computing: Mobile computing (also called nomadic computing) is the performance of computing tasks while

the user is on the move, or visiting places other than their usual environment.

3. Intranet: A portion of the Internet that is separately administered and has a boundary that can be configured to enforce local security policies and composed of several LANs linked by backbone connections.

1.2 Issues in designing a distributed system/ Challenges

In distributed system, there is no global clock among the multiple processor. So it is very hard to schedule the processor. Designing a distributed system does not come as easy and straight forward. A number of challenges need to be overcome in order to get the ideal system. The major **challenges** in distributed systems are listed below:

- Heterogeneity
- Openness
- Security
- Scalability
- Failure handling
- Concurrency
- Transparency

Heterogeneity :

The Internet enables users to access services and run applications over a heterogeneous collection of computers and networks. Heterogeneity (that is, variety and difference) applies to all of the following:

• Networks;

• Computer hardware;

• Operating systems;

• Programming languages;

• Implementations by different developers.

Different programming languages use different representations for characters and data structures such as arrays and records. These differences must be addressed if programs written in different languages are to be able to communicate with one another. Programs written by different developers cannot communicate with one another unless they use common standards.

-<u>Middleware</u> : The term middleware applies to a software layer that provides a programming abstraction as well as masking the heterogeneity of the underlying networks, hardware, operating systems and programming languages.

-<u>Heterogeneity and mobile code</u> : The term mobile code is used to refer to program code that can be transferred from one computer to another and run at the destination – Java applets are an example. Code suitable for running on one computer is not necessarily suitable for running on another because executable programs are normally specific both to the instruction set and to the host operating system.

Openness:

The openness of a computer system is the characteristic that determines whether the system can be extended and re-implemented in various ways. The openness of distributed systems is determined primarily by the degree to which new resource-sharing services can be added and be made available for use by a variety of client programs.

To summarize:

• Open systems are characterized by the fact that their key interfaces are published.

• Open distributed systems are based on the provision of a uniform communication mechanism and published interfaces for access to shared resources.

• Open distributed systems can be constructed from heterogeneous hardware and software, possibly from different vendors

Security:

Security for information resources has three components:

-confidentiality (protection against disclosure to unauthorized individuals)

-integrity (protection against alteration or corruption)

-availability (protection against interference with the means to access the resources).

However, security challenges have not yet been fully met: for Denial of service attacks' and Security of mobile code'.

Scalability:

Distributed systems operate effectively and efficiently at many different scales, ranging from a small intranet to the Internet. A system is described as scalable if it will remain effective when there is a significant increase in the number of resources and the number of users. The number of computers and servers in the Internet has increased dramatically.

The design of scalable distributed systems presents the following challenges:

- Controlling the cost of physical resources
- Controlling the performance loss
- Preventing software resources running out

- Avoiding performance bottlenecks

Failure handling

Computer systems sometimes fail. When faults occur in hardware or software, programs may produce incorrect results or may stop before they have completed the intended computation. Failures in a distributed system are partial – that is, some components fail while others continue to function. Therefore the handling of failures is particularly difficult.

- Detecting failures: Some failures can be detected. For example, checksums can be used to detect corrupted data in a message or a file
- Masking failures: Some failures that have been detected can be hidden or made less severe. Two examples of hiding failures:
    1. Messages can be retransmitted when they fail to arrive.
    2. File data can be written to a pair of disks so that if one is corrupted, the other may still be correct.
- Tolerating failures: Most of the services in the Internet do exhibit failures – it would not be practical for them to attempt to detect and hide all of the failures that might occur in such a large network with so many components. when a web browser cannot contact a web server, it does not make the user wait for ever while it keeps on trying – it informs the user about the problem, leaving them free to try again later.
- Recovery from failures: Recovery involves the design of software so that the state of

permanent data can be recovered or 'rolled back' after a server has crashed

Concurrency

Both services and applications provide resources that can be shared by clients in a distributed system. There is therefore a possibility that several clients will attempt to access a shared resource at the same time

Transparency

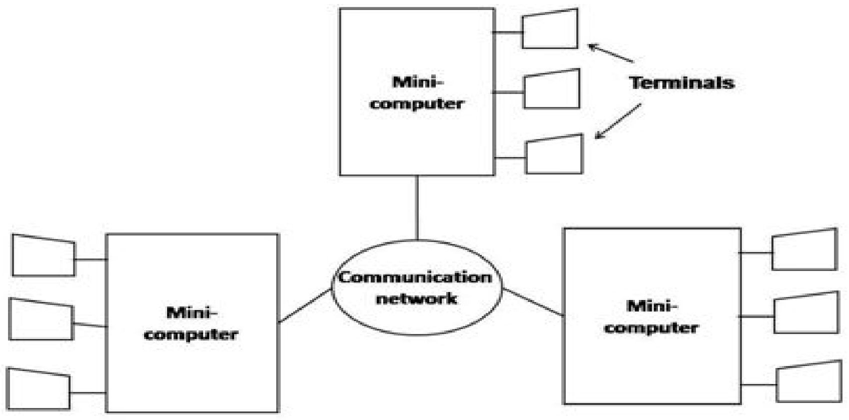
Transparency is defined as the concealment from the user and the application programmer of the separation of components in a distributed system, so that the system is perceived as a whole rather than as a collection of independent components.

- **Access transparency**: enables local and remote resources to be accessed using identical operations.
- **Location transparency** :enables resources to be accessed without knowledge of their
physical or network location (for example, which building or IP address).
- **Concurrency transparency**: enables several processes to operate concurrently using
shared resources without interference between them.
- **Replication transparency:** enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.
- **Failure transparency** : enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.
- **Mobility transparency:** allows the movement of resources and clients within a system
without affecting the operation of users or programs.
- **Performance transparency:** allows the system to be reconfigured to improve performance as loads vary.

The most important transparency are access and location transparency. They are referred together as **network transparency.**

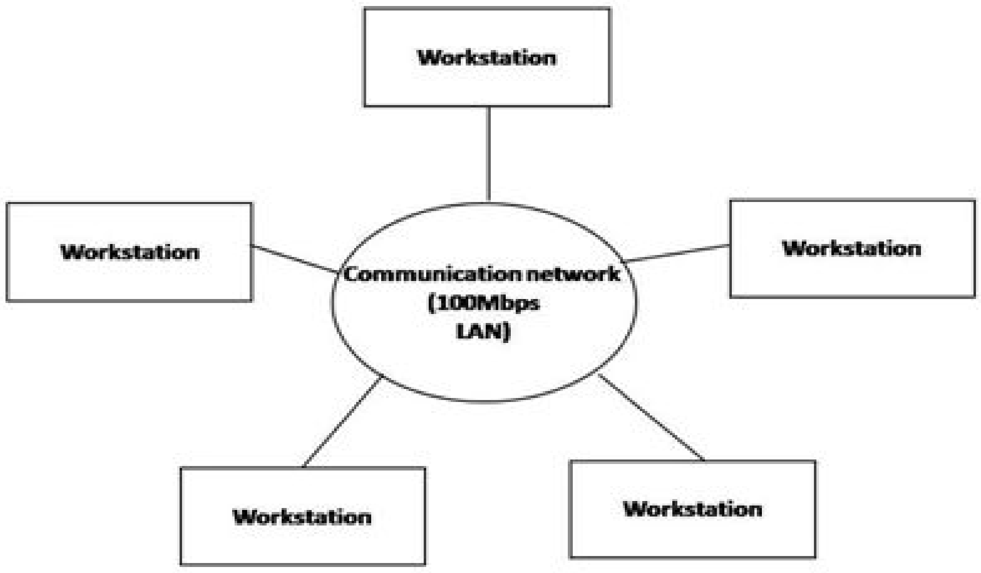
1.3 Distributed computing system models

1.3.1 Minicomputer Model



- The minicomputer model is a simple extension of the centralized time-sharing system.
- A distributed computing system based on this model consists of a few minicomputers interconnected by a communication network were each minicomputer usually has multiple users simultaneously logged on to it.
- Several interactive terminals are connected to each minicomputer.Each user logged on to one specific minicomputer has remote access to other minicomputers.
- The network allows a user to access remote resources that are available on some machine other than the one on to which the user is currently logged.The minicomputer model may be used when resource sharing with remote users is desired.
- The early ARPA net is an example of a distributed computing system based on the minicomputer model.

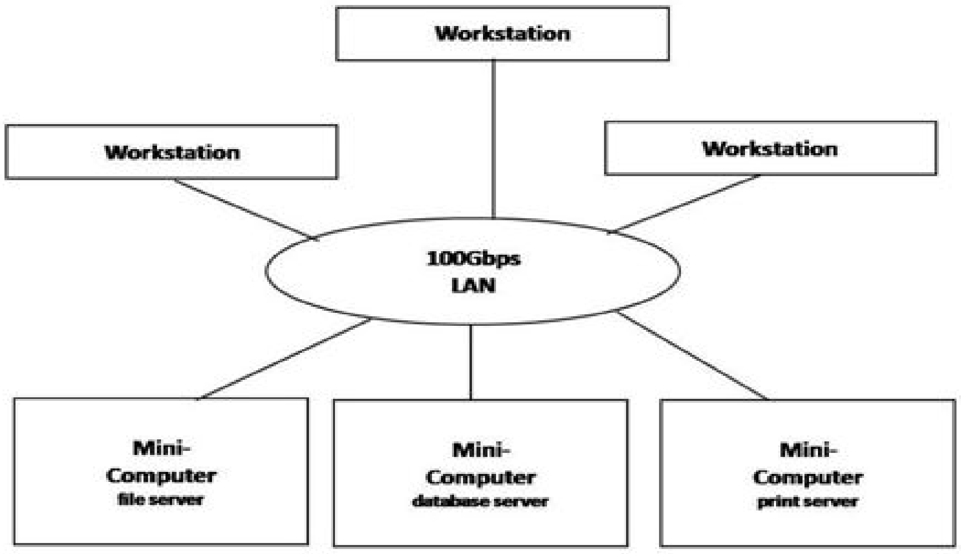
1.3.2 Workstation Model



- A distributed computing system based on the workstation model consists of several workstations interconnected by a communication network.
- An organization may have several workstations located throughout an infrastructure were each workstation is equipped with its own disk & serves as a single-user computer.
- In such an environment,at any one time a significant proportion of the workstations are idle which results in the waste of large amounts of CPU time.
- Therefore,the idea of the workstation model is to interconnect all these workstations by a high-speed LAN so that idle workstations may be used to process jobs of users who are logged onto other workstations & do not have sufficient processing power at their own workstations to get their jobs processed efficiently.
- Example:Sprite system & Xerox PARC.

1.3.3 Workstation–Server Model

- The workstation model is a network of personal workstations having its own disk & a local file system.


Workstation
Workstation
Workstation
100Gbps LAN
Mini-Computer file server
Mini-Computer database server
Mini-Computer print server

- A workstation with its own local disk is usually called a diskful workstation & a workstation without a local disk is called a diskless workstation.Diskless workstations have become more popular in network environments than diskful workstations,making the workstation-server model more popular than the workstation model for building distributed computing systems.
- A distributed computing system based on the workstation-server model consists of a few minicomputers & several workstations interconnected by a communication network.
- In this model,a user logs onto a workstation called his or her home workstation.Normal computation activities required by the user's processes are performed at the user's home workstation,but requests for services provided by special servers are sent to a server providing that type of service that performs the user's requested activity & returns the result of request processing to the user's workstation.
- Therefore,in this model,the user's processes need not migrated to the server machines for getting the work done by those machines.
- Example:The V-System.

Advantages:

1. User has guarenteed response time

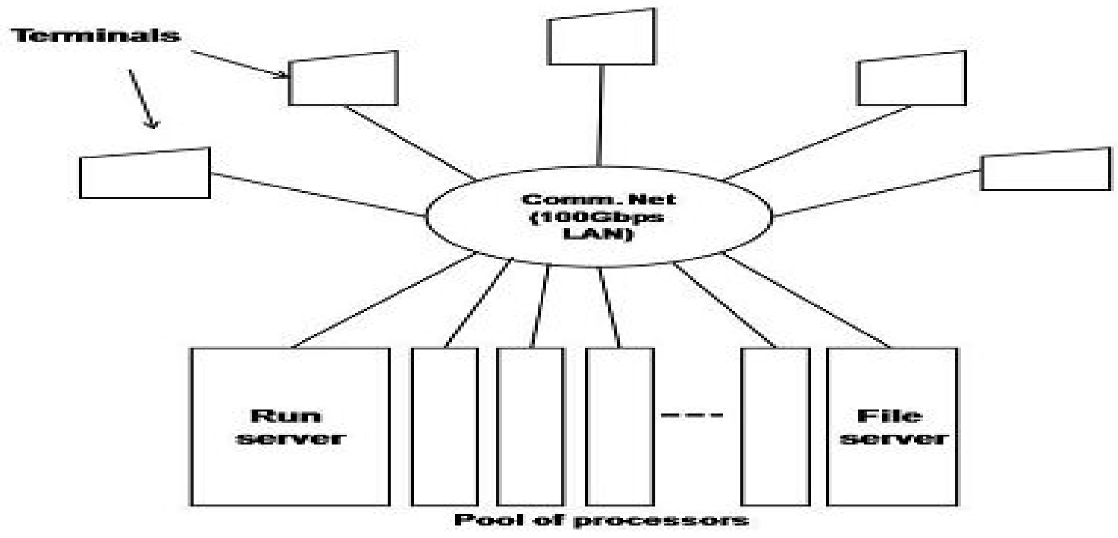
2. Does not need process migration facility due to **client-server mode**l of communication

3. Users have flexibility to use any workstation and access any files.

4. Backup and hardware maintenance are easier with diskless workstation.

1.3.4 Processor–Pool Model

- The processor-pool model is based on the observation that most of the time a user does not need any computing power but once in a while the user may need a very large amount of computing power for a short time.
- Therefore,unlike the workstation-server model in which a processor is allocated to each user,in processor-pool model the processors are pooled together to be shared by the users as needed.
- The pool of processors consists of a large number of microcomputers & minicomputers attached to the network.
- Each processor in the pool has its own memory to load & run a system program or an application program of the distributed computing system.
- In this model no home machine is present & the user does not log onto any machine.
- This model has better utilization of processing power & greater flexibility.
- Example:Amoeba & the Cambridge Distributed Computing System.



1.3.5 Hybrid Model

- The workstation-server model has a large number of computer users only performing simple interactive tasks &-executing small programs.
- In a working environment that has groups of users who often perform jobs needing massive computation,the processor-pool model is more attractive & suitable.
- To combine Advantages of workstation-server & processor-pool models,a hybrid model can be used to build a distributed system.
- The processors in the pool can be allocated dynamically for computations that are too large or require several computers for execution.
- The hybrid model gives guaranteed response to interactive jobs allowing them to be more processed in local workstations of the users

1.4 Trends in distributed systems

Distributed systems are undergoing a period of significant change and this can be traced back to a number of influential trends:

• The emergence of pervasive networking technology

• The emergence of ubiquitous computing coupled with the desire to support user mobility in distributed systems

• The increasing demand for multimedia services

• The view of distributed systems as a utility.

1.4.1 Pervasive networking and the modern Internet

The modern Internet is a vast interconnected collection of computer networks of many different types, with the range of types increasing all the time. Example, a wide range of wireless communication technologies such as WiFi, WiMAX, Bluetooth and third-generation mobile phone networks. The net result is that networking has become a pervasive resource and devices can be connected (if desired) at any time and in any place. The role of a *firewall* is to protect an intranet by preventing unauthorized messages from leaving or entering. A firewall is implemented by filtering incoming and outgoing messages.

1.4.2 Mobile and ubiquitous computing

Technological advances in device miniaturization and wireless networking have led increasingly to the integration of small and portable computing devices into distributed systems. These devices include:

- Laptop computers.
- Handheld devices, including mobile phones, smart phones, GPS-enabled devices, pagers, personal digital assistants (PDAs), video cameras and digital cameras.
- Wearable devices, such as smart watches with functionality similar to a PDA.
- Devices embedded in appliances such as washing machines, hi-fi systems, cars and refrigerators.

The portability of many of these devices, together with their ability to connect conveniently to networks in different places, makes mobile computing possible.
Mobile computing is the performance of computing tasks while the user is on the move, or visiting places other than their usual environment.

1.4.3 Distributed multimedia systems

Another important trend is the requirement to support multimedia services in distributed systems. The benefits of distributed multimedia computing are considerable in that a wide range of new (multimedia) services and applications can be provided on the desktop, including access to live or pre-recorded television broadcasts, access to film libraries offering video-on-demand services, access to music libraries, the provision of audio and video conferencing facilities and integrated telephony features including IP telephony or related technologies such as Skype, a peer-to-peer alternative to IP telephony.
Webcasting is an application of distributed multimedia technology. Webcasting is the ability to broadcast continuous media, typically audio or video, over the Internet.

1.4.4 Distributed computing as a utility

With the increasing maturity of distributed systems infrastructure, a number of companies are promoting the view of distributed resources as a commodity or utility, drawing the analogy between distributed resources.
eg. Cloud computing, grid computing
cloud computing is to provide the service *such as a database, networking, software,* storage, *servers, and many more*. at a lower rate and increase returns.

grid computing basically focuses on networks to solve some complex problems and has a large-scale goal.