

1

INTRODUCTION TO DISTRIBUTED SYSTEMS

1.1 WHAT IS DISTRIBUTED SYSTEM AND DISTRIBUTED COMPUTING?

The process of computation was started from working on a single processor. This uni-processor computing can be termed as **centralized computing**. As the demand for the increased processing capability grew high, multiprocessor systems came to existence. The advent of multiprocessor systems, led to the development of **distributed systems** with high degree of scalability and resource sharing. The modern day parallel computing is a subset of distributed computing.

A distributed system is a collection of independent computers, interconnected via a network, capable of collaborating on a task. Distributed computing is computing performed in a distributed system.

Distributed computing is widely used due to advancements in machines and faster and cheaper networks. In distributed systems, the entire network will be viewed as a computer. The multiple systems connected to the network will appear as a single system to the user. Thus the distributed systems hide the complexity of the underlying architecture to the user.

The definition of distributed systems deals with two aspects that:

- **Deals with hardware:** The machines linked in a distributed system are autonomous.
- **Deals with software:** A distributed system gives an impression to the users that they are dealing with a single system.

Features of Distributed Systems:

- Communication is hidden from users
- Applications interact in uniform and consistent way

1.2 Introduction of Distributed System

- High degree of scalability
- A distributed system is functionally equivalent to the systems of which it is composed.
- Resource sharing is possible in distributed systems.
- Distributed systems act as fault tolerant systems
- Enhanced performance

Issues in distributed systems

- Concurrency
- Distributed system function in a heterogeneous environment. So adaptability is a major issue.
- Latency
- Memory considerations: The distributed systems work on both local and shared memory.
- Synchronization issues
- Applications must need to adapt gracefully without affecting other parts of the systems in case of failures.
- Since they are widespread, security is a major issue.
- Limits imposed on scalability
- They are less transparent.
- Knowledge about the dynamic network topology is a must.

QOS parameters

The distributed systems must offer the following QOS:

- Performance
- Reliability
- Availability
- Security

Differences between centralized and distributed systems

Centralized Systems	Distributed Systems
In Centralized Systems, several jobs are done on a particular central processing unit(CPU)	In Distributed Systems, jobs are distributed among several processors. The Processor are interconnected by a computer network
They have shared memory and shared variables.	They have no global state (i.e.) no shared memory and no shared variables.
Clocking is present.	No global clock.

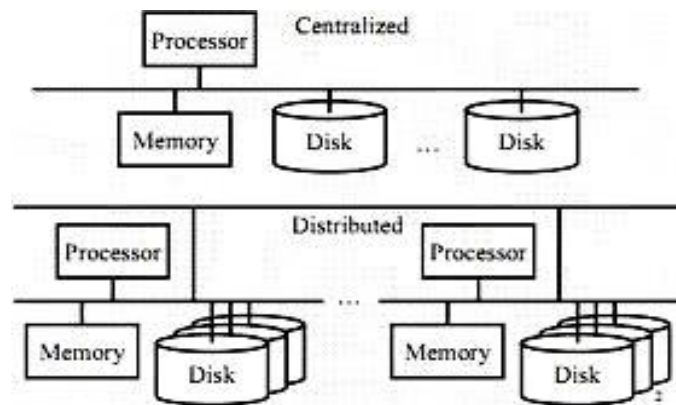


Fig 1.1: Distributed and Centralized systems

1.2 SOME EXAMPLES OF DISTRIBUTED SYSTEMS

Today’s internet exists on the distributed systems. There are numerous applications of distributed systems. Some of them are discussed below:

Web Search

- ✓ The task of a web search engine is to index the entire contents of the World Wide Web.
- ✓ Distributed search is a search engine model in which the tasks of Web crawling, indexing and query processing are distributed among multiple computers and networks.
- ✓ The search engines were supported by a single supercomputer . But in recent years, they have moved to a distributed model.
- ✓ Google search relies upon thousands of computers crawling the Web from multiple locations all over the world.

1.4 Introduction of Distributed System

- ✓ In Google's distributed search system, each computer involved in indexing crawls and reviews a portion of the Web, taking a URL and following every link available from it.
- ✓ The computer gathers the crawled results from the URLs and sends that information back to a centralized server in compressed format.
- ✓ The centralized server then coordinates that information in a database, along with information from other computers involved in indexing.
- ✓ When a user types a query into the search field, Google's domain name server (DNS) software relays the query to the most logical cluster of computers, based on factors such as its proximity to the user or how busy it is.
- ✓ At the recipient cluster, the Web server software distributes the query to hundreds or thousands of computers to search simultaneously.
- ✓ Hundreds of computers scan the database index to find all relevant records.
- ✓ The index server compiles the results, the document server pulls together the titles and summaries and the page builder creates the search result pages.
- ✓ The following features of Google search makes it act as distributed system:
 - The physical infrastructure with very large numbers of networked computers .
 - Highly distributed file system that supports very large files.
 - Availability of structured distributed storage system for fast access to data.
 - Distributed locking and agreement.
 - Works on a programming model that supports the management of large parallel and distributed computations.

Massively multiplayer online games (MMOGs)

- ✓ These games simulate real-life as much as possible.
- ✓ As such it is necessary to constantly evolve the game world using a set of laws.
- ✓ These laws are a complex set of rules that the game engine applies with every clock tick.
- ✓ The virtual world consists not only of human players but also of all game elements that are not living objects.
- ✓ These elements are immutable and include the area terrain, trees, mountains, rivers, etc.

- ✓ MMOGs must be able to handle a very large number of simultaneous users.
- ✓ As the information transferred between the players and the game server is large, the bandwidth required to support a huge number of players is enormous.
- ✓ Very large virtual worlds require huge computational power to simulate the existence of life (AI Algorithms).
- ✓ No single processor machine can handle the computational load required.
- ✓ These gaming applications work on both client server and distributed architectures.
- ✓ In distributed architectures, the large virtual world of the game is split into different smaller areas and each area is to be handled by a separate physical machine (server).
- ✓ Therefore both the bandwidth and computational load is spread out on many machines, thus making the application distributed.

Financial Trading

- ✓ The financial trading is now moving to distributed systems.
- ✓ These systems require frequent modifications, in response to the communication.
- ✓ Processing the events in distributed systems, demands reliability.
- ✓ So distributed event based systems are used.
- ✓ A series of event feeds are given by the financial institution.
- ✓ The events may be in different formats, employ different technologies etc.
- ✓ So proper adaptors are a must.
- ✓ Pattern detection is also very important aspect in financial trading.
- ✓ The **Complex Event Processing (CEP)** is an automatized way of composing event together into logical, temporal or spatial patterns.

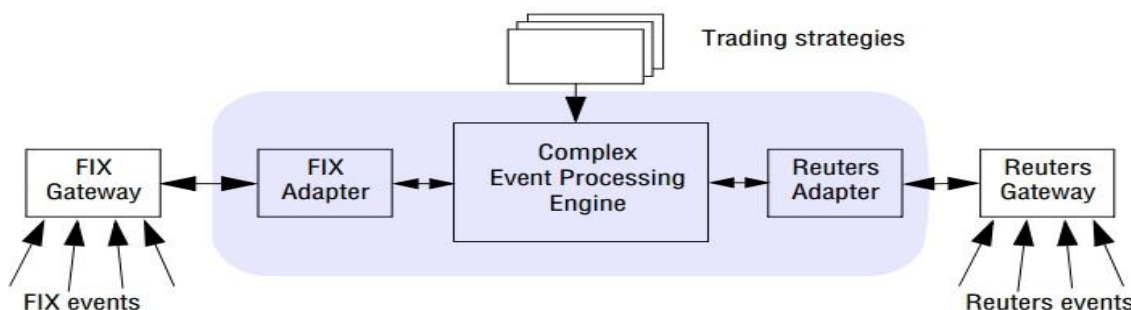


Fig 1.2: Financial Trading System

1.3 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.

1.3.1 Pervasive networking and the modern Internet

Pervasive computing devices are completely connected and constantly available.

- ✓ The products that are connected to the pervasive network are easily available.
- ✓ The main goal of pervasive computing is to create an environment where the connectivity of devices is embedded in such a way that the connectivity is unobtrusive and always available.
- ✓ Internet can be seen as a large distributed system.

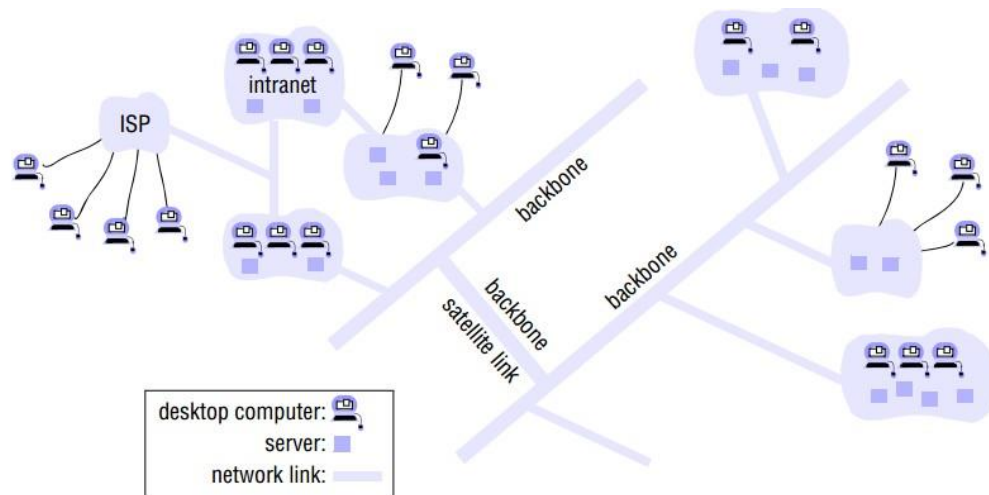


Fig 1.3: Internet

- ✓ Internet is actually a collection of numerous sub networks operated by companies and other organizations and typically protected by firewalls.
- ✓ The role of a firewall is to protect an intranet by preventing unauthorized messages from leaving or entering.

- ✓ Internet Service Providers (ISPs) are companies that provide broadband links and other types of connection to individual users and small organizations, enabling them to access services anywhere in the Internet as well as providing local services such as email and web hosting.
- ✓ The intranets are linked together by backbones network link with a high transmission capacity, employing satellite connections, fibre optic cables and other high-bandwidth circuits.

1.3.2 Mobile and ubiquitous computing

- ✓ Internet is now build into numerous small devices from laptops to watches.
- ✓ These devices must have high degree of portability. Mobile computing supports this.
- ✓ Mobile computing is the performances of computing tasks while the user dynamically changing his geographic location.
- ✓ Ubiquitous computing (small computing) means that all small computing devices will eventually become so pervasive in everyday objects.

Differences between ubiquitous computing and mobile computing

Ubiquitous Computing	Mobile Computing
They could connect tens/hundreds of computing devices in every room/person, becoming “invisible” and part of the environment – WANs, LANs, PANs – networking in small spaces	They could connect a few devices for every person, small enough to carry around – devices connected to cellular networks or WLANs
They could connect even the non- mobile devices and offer various forms of communication.	They are actually a subset of ubiquitous computing.
They could support all form of devices that are connected to the internet from laptops to watches.	They support only conventional, discrete computers and devices.

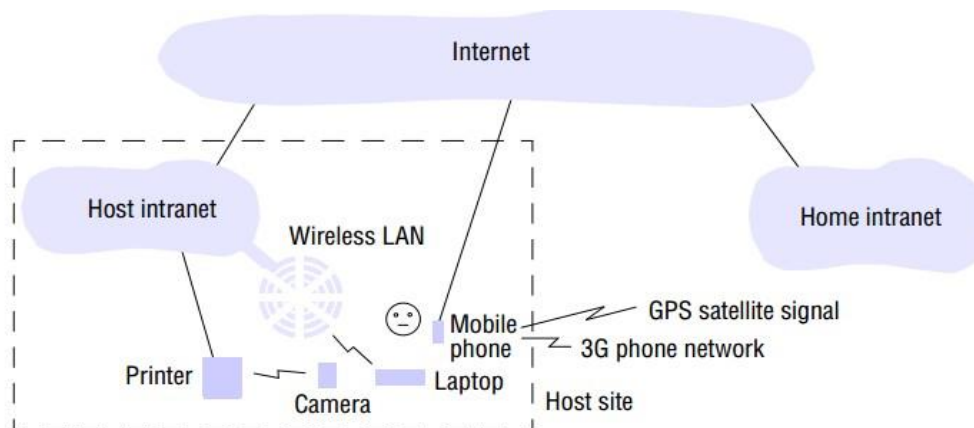


Fig 1.4: Mobile devices in an environment

- ✓ The user in the above given environment has access to three forms of wireless connection:
 - * The laptop connecting to the host's wireless LAN.
 - * Mobile (cellular) telephone connected to the Internet. The phone gives access to the Web and other Internet services.
 - * Digital camera that communicates over a personal area wireless network.
- ✓ This scenario demands associations between devices are routinely created and destroyed called as **spontaneous interoperation**. This interoperation must be fast and convenient.

1.3.3 Distributed multimedia systems

- ✓ A distributed system supports the storage, transmission and presentation of discrete media types.
- ✓ A distributed multimedia system should be able to perform the same functions for continuous media types such as audio and video.
- ✓ It should be able to store and locate audio or video files, to transmit them across the network to support the presentation of the media types to the user and optionally also to share the media types across a group of users.
- ✓ The processing of such media files includes handling of temporal dimensions and integrity of media.
- ✓ The distributed multimedia computing allows a wide range of new multimedia services and applications to be provided on the desktop.

- ✓ This includes 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.

- ✓ Webcasting demands the following changes in the infrastructure:
 - Support for an range of encoding and encryption formats.
 - Support for range of mechanisms to ensure that the desired quality of service can be met.
 - Adaptability to associated resource management strategies.
 - Providing adaptation strategies to deal with the situation where QOS is difficult to achieve.

1.3.4 Distributed computing as a utility

- ✓ Distributed systems are seen as a utility like water and electricity.
- ✓ The resources are provided by appropriate service suppliers and effectively rented by the end user.
- ✓ The services may be physical or logical services.
- ✓ Physical resources such as storage and processing can be made available to networked computers through data centres.
- ✓ Operating system virtualization is a key enabling technology that users may actually be provided with services by a virtual rather than a physical node.
- ✓ Software services rental, services such as email and distributed calendars.
- ✓ Cloud computing is the child of resource sharing.

A cloud is defined as a set of Internet-based application, storage and computing services sufficient to support most users' needs, thus enabling them to largely or totally dispense with local data storage and application software with local data storage and application software.

- ✓ Clouds are generally implemented on cluster computers.
- ✓ A cluster computer is a set of interconnected computers that cooperate closely to provide a single, integrated high performance computing capability.

1.4 RESOURCE SHARING

- ❖ An important goal of a distributed system is to effectively utilize the collective resources of the system, namely, the memory and the processors of the individual nodes.
- ❖ Users think in terms of shared resources such as a search engine without regard for the server or servers that provide these.
- ❖ A search engine on the Web provides a facility to users throughout the world, users who need never come into contact with one another directly.
- ❖ In **computer-supported cooperative working (CSCW)**, a group of users who cooperate directly share resources such as documents in a small, closed group.
- ❖ The pattern of sharing and the geographic distribution of particular users determine what mechanisms the system must supply to coordinate users' actions.
- ❖ The **service** manages a collection of related resources and presents their functionality to users and applications.
- ❖ Resources in a distributed system are physically encapsulated within computers and can only be accessed from other computers by means of communication.
- ❖ Sharing is done by a program that offers a communication interface enabling the resource to be accessed and updated reliably and consistently.
- ❖ In a client server paradigm, server refers to a running program on a networked computer that accepts requests from programs running on other computers to perform a service and responds appropriately.
- ❖ The requesting processes are referred to as clients.
- ❖ A complete interaction between a client and a server, from the point when the client sends its request to when it receives the server's response, is called a **remote invocation**.

1.5 CHALLENGES IN DISTRIBUTED SYSTEMS

a) Heterogeneity

Heterogeneity means the diversity of the distributed systems in terms of hardware, software, platform, etc. Modern distributed systems will likely to be operating with different:

- Hardware devices: computers, tablets, mobile phones, embedded devices, etc.

- Operating System: Ms Windows, Linux, Mac, Unix, etc.
- Network: Local network, the Internet, wireless network, satellite links, etc.
- Programming languages: Java, C/C++, Python, PHP, etc.
- Different roles of software developers, designers, system managers
- Middleware: 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. Eg: CORBA, RMI.
- Heterogeneity in mobile code: Mobile code is used to refer to program code that can be transferred from one computer to another and run at the destination. Eg: Java applets.



Fig 1.5: Challenges in Distributed systems

b) Transparency

Distributed systems designers must hide the complexity of the systems. Adding abstraction layer is particularly useful in distributed systems. While users hit search in google.com, they never notice that their query goes through a complex process before google shows them a result. Some terms of transparency in distributed systems are:

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located

1.12 Introduction of Distributed System

Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource may be copied in several places
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or a disk

The access and location transparency is collectively referred as **network transparency**.

c) Openness

If the well-defined interfaces for a system are published, it is easier for developers to add new features or replace sub-systems in the future. Example: Twitter and Facebook have API that allows developers to develop their software. The following are key points in openness:

- * key interfaces are published.
- * 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.

d) Concurrency

Distributed Systems usually is multi-users environment. In order to maximize concurrency, resource handling components should anticipate as they will be accessed by competing users. Concurrency prevents the system to become unstable when users compete to view or update data.

e) Security

Every system must consider strong security measurement. Distributed Systems somehow deals with sensitive information; so secure mechanism must be in place. The following attacks are more common in distributed systems:

- * **Denial of service attacks:** When the requested service is not available at the time of request it is Denial of Service (DOS) attack. This attack is done by bombarding the service with a large number of useless requests that the serious users are unable to use it.

- * **Security of mobile code:** Mobile code needs to be handled with care since they are transmitted in an open environment.

f) Scalability

Distributed systems must be scalable as the number of user increases.

A system is said to be scalable if it can handle the addition of users and resources without suffering a noticeable loss of performance or increase in administrative complexity.

Scalability has 3 dimensions:

- * **Size:** Number of users and resources to be processed. Problem associated with this is overloading.
- * **Geography:** Distance between users and resources. Problem associated is communication reliability
- * **Administration:** As the size of distributed systems increases, many of the system needs to be controlled. Problem associated is administrative mess

Scalability often conflicts with small system performance. Claim of scalability in such system is often abused. The following are the scalability challenges:

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

g) Resilience to Failure (Fault tolerance)

Distributed Systems involves a lot of collaborating components (hardware, software, communication). So there is a huge possibility of partial or total failure. The failures are handled in series of steps:

- **Detecting failures:** Some failures like checksum can be detected.
- **Masking failures:** Some failures that have been detected can be hidden or made less severe. Examples of hiding failures include retransmission of messages and maintaining a redundant copy of same data.
- **Tolerating failures:** All the failures cannot be handled. Some failures must be accepted by the user. Example of this is waiting for a video file to be streamed in.

- **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.
- **Redundancy:** Services can be made to tolerate failures by the use of redundant components. Examples for this includes: maintenance of two different paths between same source and destination.
- Availability is also a major concern in the fault tolerance. The **availability** of a system is a measure of the proportion of time that it is available for use. It is a useful performance metric.

h) Quality of Service:

The distributed systems must confirm the following non functional requirements:

- ❖ **Reliability:** A reliable distributed system is designed to be as fault tolerant as possible. Reliability is the quality of a measurement indicating the degree to which the measure is consistent.
- ❖ **Security:** Security is the degree of resistance to, or protection from, harm. It applies to any vulnerable and valuable asset, such as a person, dwelling, community, nation, or organization. Distributed systems spread across wide geographic locations. So security is a major concern.
- ❖ **Adaptability:** The frequent changing of configurations and resource availability demands the distributed system to be highly adaptable.

1.6 WORLD WIDE WEB (WWW)

1.6.1 History and Development of WWW

- ✓ The World Wide Web (WWW) can be viewed as a huge distributed system consisting of millions of clients and servers for accessing linked documents.
- ✓ Servers maintain collections of documents, while clients provide users an easy to-use interface for presenting and accessing those documents.
- ✓ The Web started as a project at CERN, the European Particle Physics Laboratory in Geneva, to let its large and geographically dispersed group of researchers provide access to shared documents using a simple hypertext system.
- ✓ A document in a WWW could be anything that could be displayed on a user's computer terminal, such as personal notes, reports, figures, blueprints, drawings, and so on.
- ✓ By linking documents to each other, it became easy to integrate documents from different projects into a new document without the necessity for centralized changes.

- ✓ The Web gradually grew worldwide encompassing sites other than high energy physics, but popularity really increased when graphical user interfaces became available, notably Mosaic (Vetter et al., 1994).
- ✓ Mosaic provided an easy-to-use interface to present and access documents by merely clicking the mouse.
- ✓ A document was fetched from a server, transferred to a client, and presented on the screen.
- ✓ To a user, there was conceptually no difference between a document stored locally or in another part of the world. This is transparent distribution.
- ✓ Since 1994, Web developments are primarily initiated and controlled by the World Wide Web Consortium, which is a collaboration between CERN and M.I.T.
- ✓ This consortium is responsible for standardizing protocols, improving interoperability, and further enhancing the capabilities of the Web.
- ✓ The webpages are portable and open.

1.6.2 Components of WWW

➤ HTML

- ✓ The Web documents are expressed by means of a special language called HyperText Markup Language (HTML).
- ✓ HTML provides keywords to structure a document into different sections.
- ✓ One of its most powerful features is the ability to express parts of a document in the form of a script.
- ✓ When a document is parsed, it is internally stored as a rooted tree, called a **parse tree**, in which each node represents an element of that document.
- ✓ Each node is required to implement a standard interface containing methods for accessing its content, returning references to parent and child nodes, and so on. This standard representation is also known as the **Document Object Model** or DOM or **dynamic HTML**.
- ✓ An alternative language that also matches the DOM is XML (Extensible Markup Language).
- ✓ XML is used only to structure a document; it contains no keywords.
- ✓ XML can be used to define arbitrary structures. In other words, it provides the means to define different document types.

➤ **Uniform Resource Locators (URL)**

The pages of a website can usually be accessed from a simple Uniform Resource Locator (URL) otherwise called as web address.

- ✓ URL's are a way of identifying information on a server.
- ✓ A URL gives the protocol, the domain, the directory, and even the file.
- ✓ A URL consists of the following parts:
 - protocol (such as http:// or ftp://)
 - host name (the Web server's IP address or domain name)
 - directory (i.e. folder)
 - file name

There are two forms of URL as listed below:

- **Absolute URL:**

Absolute URL is a complete address of a resource on the web. This completed address comprises of protocol used, server name, path name and file name.

Example: `http:// www.abc.com / xyz /index.htm.`

- http is the protocol.
- abc.com is the server name.
- index.htm is the file name.

The protocol part tells the web browser how to handle the file. Other protocols also that can be used to create URL are: FTP, https, Gophe, mailto, news

- **Relative URL**

Relative URL is a partial address of a webpage. Unlike absolute URL, the protocol and server part are omitted from relative URL. Relative URLs are used for internal links i.e. to create links to file that are part of same website as the WebPages on which you are placing the link.

Example:

To link an image on `abc.com/xyz/internet_referemce_models`, we can use the relative URL which can take the form like `/internet_technologies/internet-osi_model.jpg`.

➤ **Hyper Text Transfer Protocol (HTTP)**

- ✓ HTTP is a communication protocol.
- ✓ It defines mechanism for communication between browser and the web server. It is also called request and response protocol because the communication between browser and server takes place in request and response pairs.
- ✓ It is a **stateless** protocol (i.e.) the history of the communication between server and client is not stored in any form.

HTTP Request

- ✓ HTTP request comprises of lines which contains: Request line, Header Fields and Message body.
- ✓ The first line i.e. the Request line specifies the request method i.e. Get or Post.
- ✓ The second line specifies the header which indicates the domain name of the server from where index.htm is retrieved.

HTTP Response

- ✓ Like HTTP request, HTTP response also has certain structure.
- ✓ HTTP response contains: Status line, Headers and Message body.

➤ **Publishing a resource:**

- ✓ The methods for publishing resources on the Web are dependent upon the web server implementation.
- ✓ The simplest method of publishing a resource on the Web is to place the corresponding file in a directory that the web server can access.
- ✓ It is common for such concerns to be hidden from users when they generate content.
- ✓ The database or file system on which the product pages are based is transparent.

➤ **Downloaded code:**

- ✓ The designers of web services require some service-related code to run inside the browser, at the user's computer.
- ✓ Javascript is an example for downloaded code.
- ✓ A Javascript enhanced page can give the user immediate feedback on invalid entries, instead of forcing the user to check the values at the server, which would take much longer.

1.18 Introduction of Distributed System

- ✓ AJAX (Asynchronous Javascript And XML) is used in cases where synchronization is not a major concern.
 - ✓ Applet is also a form of downloaded code.
 - ✓ It is an application which the browser automatically downloads and runs when it fetches a corresponding web page.
- **Web services**
- ✓ Web services allow exchange of information between applications on the web.
 - ✓ Using web services, applications can easily interact with each other.
 - ✓ The web services are offered using concept of Utility Computing.

REVIEW QUESTIONS

PART-A

1. Define centralized computing.

The process of computation was started from working on a single processor. This uni-processor computing can be termed as centralized computing.

2. Define distributed systems.

A distributed system is a collection of independent computers, interconnected via a network, capable of collaborating on a task. Distributed computing is computing performed in a distributed system.

3. List the features of Distributed Systems.

- Communication is hidden from users
- Applications interact in uniform and consistent way
- High degree of scalability
- A distributed system is functionally equivalent to the systems of which it is composed.
- Resource sharing is possible in distributed systems.
- Distributed systems act as fault tolerant systems
- Enhanced performance

4. What are the issues in distributed systems?

- Concurrency
- Distributed system function in a heterogeneous environment. So adaptability is a major issue.
- Latency
- Memory considerations: The distributed systems work on both local and shared memory.
- Synchronization issues
- Applications must need to adapt gracefully without affecting other parts of the systems in case of failures.
- Since they are widespread, security is a major issue.

- Limits imposed on scalability
- They are less transparent.
- Knowledge about the dynamic network topology is a must.

5. Mention the QOS parameters of DS.

The distributed systems must offer the following QOS:

- Performance
- Reliability
- Availability
- Security

6. Give the differences between centralized and distributed systems

Centralized Systems	Distributed Systems
In Centralized Systems, several jobs are done on a particular central processing unit(CPU)	In Distributed Systems, jobs are distributed among several processors. The Processor are interconnected by a computer network
They have shared memory and shared variables.	They have no global state (i.e.) no shared memory and no shared variables.
Clocking is present.	No global clock.

7. What is distributed search?

Distributed search is a search engine model in which the tasks of Web crawling, indexing and query processing are distributed among multiple computers and networks.

8. List the features of Google DS.

The physical infrastructure with very large numbers of networked computers .

Highly distributed file system that supports very large files.

Availability of structured distributed storage system for fast access to data.

Distributed locking and agreement.

Works on a programming model that supports the management of large parallel and distributed computations.

9. What is MMOG?

These games simulate real-life as much as possible. As such it is necessary to constantly evolve the game world using a set of laws. These laws are a complex set of rules that the game engine applies with every clock tick

10. What is CEP?

The Complex Event Processing (CEP) is an automatized way of composing event together into logical, temporal or spatial patterns.

11. What is pervasive computing?

Pervasive computing devices are completely connected and constantly available.

12. Define ISP.

Internet Service Providers (ISPs) are companies that provide broadband links and other types of connection to individual users and small organizations, enabling them to access services anywhere in the Internet as well as providing local services such as email and web hosting.

13. Define Ubiquitous computing .

Ubiquitous computing (small computing) means that all small computing devices will eventually become so pervasive in everyday objects.

14. Give the differences between ubiquitous computing and mobile computing

Ubiquitous Computing	Mobile Computing
They could connect tens/hundreds of computing devices in every room/person, becoming “invisible” and part of the environment – WANs, LANs, PANs – networking in small spaces	They could connect a few devices for every person, small enough to carry around – devices connected to cellular networks or WLANs
They could connect even the non- mobile devices and offer various forms of communication.	They are actually a subset of ubiquitous computing.
They could support all form of devices that are connected to the internet from laptops to watches.	They support only conventional, discrete computers and devices.

15. Define webcasting.

Webcasting is the ability to broadcast continuous media, typically audio or video, over the Internet.

16. List the requirements of web casting.

Support for an range of encoding and encryption formats.

Support for range of mechanisms to ensure that the desired quality of service can be met.

Adaptability to associated resource management strategies.

Providing adaptation strategies to deal with the situation where QOS is difficult to achieve.

17. Define cloud.

A cloud is defined as a set of Internet-based application, storage and computing services sufficient to support most users' needs, thus enabling them to largely or totally dispense with local data storage and application software with local data storage and application software.

18. What is CSCW?

In computer-supported cooperative working (CSCW), a group of users who cooperate directly share resources such as documents in a small, closed group.

19. Define remote invocation.

A complete interaction between a client and a server, from the point when the client sends its request to when it receives the server's response, is called a remote invocation.

20. List the challenges in DS.

- ↑ Heterogeneity
- ↑ Transparency
- ↑ Openness
- ↑ Concurrency
- ↑ Security
- ↑ Scalability
- ↑ Fault tolerance
- ↑ Providing QOS

21. What are the dimensions of scalability?

- * Size: Number of users and resources to be processed. Problem associated with this is overloading.

- * Geography: Distance between users and resources. Problem associated is communication reliability
- * Administration: As the size of distributed systems increases, many of the system needs to be controlled. Problem associated is administrative mess

22. List the challenges in scalability.

The following are the scalability challenges:

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

23. Give the features in fault tolerance.

Detecting failures: Some failures like checksum can be detected.

Masking failures: Some failures that have been detected can be hidden or made less severe. Examples of hiding failures include retransmission of messages and maintaining a redundant copy of same data.

Tolerating failures: All the failures cannot be handled. Some failures must be accepted by the user. Example of this is waiting for a video file to be streamed in.

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.

Redundancy: Services can be made to tolerate failures by the use of redundant components. Examples for this includes: maintenance of two different paths between same source and destination.

24. What is parse tree?

When a document is parsed, it is internally stored as a rooted tree, called a parse tree, in which each node represents an element of that document.

25. Define URL.

The pages of a website can usually be accessed from a simple Uniform Resource Locator (URL) otherwise called as web address.

PART – B

1. What are the issues in distributed systems?
2. Explain web search as a distributed system.
3. Brief about MMOGs.
4. Describe financial trading.
5. Explain the trends in distributed systems.
6. Elucidate the resource sharing nature of distributed system.
7. Brief about the challenges in distributed system.
8. Describe WWW.
9. Explain HTML.