CSC 424
MINI AND MACRO-COMPUTER

INTRODUCTION

A distributed system is an application that executes a collection of protocols to coordinates the actions of multiple processes on a network such that components cooperate together to perform a single or small set of related tasks.

Distributed system can also be referred to as the process of running a single computational task on more than one distinct computer. There are a lot of advantages in distributed system including the ability to connect remote users with remote resources in an open and scalable way. A distributed system must be reliable and for it to be truly reliable it must possess the following characteristics:

- Scalable
- Consistent
- Fault tolerant
- Recoverable
- Highly available
- Secure
- Predictable performance

Some of the disadvantages of distributed systems are:

- Virus susceptibility
- Backup difficulty
- Multiple point of failures

CENTRALIZED COMPUTING

This is done at a central location using terminals that are attached to a central computer. The computer itself may control all the peripherals directly (if they are physically connected to the central computer) or they may be attached via a terminal server. Alternatively if the terminals have the capability, they may be able to connect to the central computer over the network.

Centralized computing offers greater security because all the processing is controlled in a central location. In addition, if one terminal breaks down the user can simply go to another terminal and log in again and all their files will still be accessible. The central computer performs the computing functions and controls the remote terminals.

The disadvantage of this system is that this system of computing relies totally on the central computer, should the central computer crash, the entire system will be unavailable (go down). Accessing network may also be slow.

COLLABORATIVE COMPUTING

This is also called cooperative computing, it enables computers in a distributed computing environment to share processing power in addition to data, resources and services. It is a term describing a variety of activities where people interact with one another using desktops, laptops, palmtops and sophisticated digital cellular phones. As computers are best at handling data and representing information, person-to-person communication is enriched by an ability to share, modify or collaboratively create data and information.

Collaborative computing is extremely fast, it enables complete documentation, it can also be used as an application development interface and it allows multiple users.

ADVANTAGES OF DISTRIBUTED SYSTEMS OVER CENTRALIZED SYSTEMS

1. **RELIABILITY**: If one machine crashes, the system as a whole can still survive.

2. **SPEED**: A distributed system may have more total computing power than a mainframe

3. **OPEN SYSTEM**: Since it is an open system it is always ready to communicate with other systems. An open system that scales has an advantage over a perfectly closed and self-contained system.

4. **ECONOMIC**: Collection of microprocessors offers a better price or performance than mainframes.

5. **INCREMENTAL GROWTH**: Computing power can be added in small increments.

DISTRIBUTED DATABASE

A Distributed Database is a database that is stored in more than one physical location. Parts (Partition d/b) or copies (Replication d/b) are physically stored in one location and other parts or copies are stored and maintained in other locations.

Distributed Database can be implemented in four main ways:

Partitioned Database: A **partition** is a division of a logical database or its constituting elements into distinct independent parts. Database partitioning is normally done for manageability, performance or availability reasons. Each partition may be spread over multiple nodes, and users at the node can perform local transactions on the partition. This increases performance for sites that have regular transactions involving certain views of data, while maintaining availability and security. The partitioning can be done by either building separate smaller databases (each with its own tables, indices, and transaction logs), or by splitting selected elements.

Duplicate Database: The duplicate database can be either identical to the original database or contain only a subset of the original table-spaces. A duplicate database is a copy of the target database that you can run independently for a variety of purposes. A duplicate database cannot be updated continually or periodically by using archived logs from the primary database and does not support recovery and failover options.

Central Index Distributed Database: The central database contains only an index of entries. The actual records are held at the remote site. A query to the central database will locate where the record is held. This system is used by very large databases.

Network Distributed Database: A distributed database is a database in which storage devices are not all attached to a common CPU. It may be stored in multiple computers located in the same physical location, or may be dispersed over a network of interconnected computers. Collections of data (e.g. in a database) can be distributed across multiple physical locations. A distributed database can reside on network servers on the Internet, on corporate intranets or extranets, or on other company networks.

Advantages of Distributed system

- **Data sharing**: Allow many users to access to a common database
- **Resource sharing**: Allow expensive peripherals like colour printers
- **Communication**: Enhance human-to-human communication e.g. email, chat e.t.c.

Disadvantages of Distributed System

- **Flexibility:** Spread the workload over the available machines
- **Software:** Difficult to develop software for distributed systems
- **Network:** Saturation, loss transmissions
- **Security:** Easy access also applies to secret data

EXAMPLES OF DISTRIBUTED SYSTEMS

INTERNET: It is interconnected collection of computer network of many different types like WAN, MAN, LAN e.t.c.

Design Issues

- The design issue of the internet communication mechanism is a major technical achievement that enables programs running anywhere to address message program anywhere else.
- The internet is a large distributed system that enables users wherever they are to make use of service such as WWW, email and file transfer. This set of services is open ended as it can be extended by the addition of server computer and new type of service.

INTRANET: An intranet is a portion of the internet that is separately administered and has boundary that can be configured to enforce the local security policies. It is composed of several local area network linked by back bone connection. The intranet is connected to the internet via a router, which allows the user inside the intranet to access the services elsewhere such as web or email. It also allows the user in other internet to access the services it provides. Many organization need to protect their own services from unauthorized use by possibly malicious user.

Design Issues

- File services are needed to enable to share data.
- Firewall tends to impede to legitimate access to services when resource sharing between internal and external users is required, firewall must be completed by the use of fine grain security mechanism.

MOBILE FACILITY: It is an integration of small and portable computing device into distributed system. Examples of various mobile computing devices are:

a) Laptop computers
b) Handheld device like PDA Personal Digital Assistant, Mobile phone pagers, video camera and digital camera.

NETWORK OF WORKSTATIONS: This is a computer network that connects several computer workstations together with special software forming a cluster.

Design issues

- Personal workstations + processors not assigned to specific users
- Single file system, with all files accessible from all machines in the same way and using the same path name.
- For a certain command the system can look for the best place (workstation) to execute it.

AUTOMATIC BANKING (Teller machine) SYSTEM

In teller machine, the data is shared by the server on different machine of Banks. A concept of distributed system in which all the data of a server in a bank will be stored and also be a shared data to main server.

- Primary requirements security and reliability
- Consistency of replicated data
- Concurrent transactions (operations which involve accounts in different banks; simultaneous access from several users etc.
- Fault tolerance.

AUTOMATIVE SYSTEM (A Distributed Real-time System)

A distributed real-time system is an integrated system comprising a set of dedicated hardware that monitors, acts and reacts on events within specified time period. Its requirements are dictated by the outside environment, not by the computer. This type of system is characterized under;

a) The internet
    - Heterogeneous network of computers and applications
    - Implemented through the Internet Protocol Stack
    - Typical configuration

b) Distributed Multimedia-Systems
    - Often use internet infrastructure such as video , audio , text , heterogeneous data sources and sinks that need to be synchronized in real time
    - Often use Distribution services such as multicast e.g. tele-teaching tools, video-conferencing, video and audio on demand.

c) Intranets
    - Locally administered network
    - Interfaces with the internet
    - Firewalls
    - Provides services internally and externally

CHALLENGES IN THE DESIGN OF DISTRIBUTED SYSTEM

The following are the challenges being faced while designing distributed system.

1. Heterogeneity: It's underlying network infrastructure, computer hardware and software (e.g. Operating systems), programming languages (in particular data representation).
2. Openness
   - Ensuring extensibility and maintainability of the systems
   - Adherence to standard interface
3. Security
   - Privacy
   - Authentication
   - Availability
4. Scalability
   - Handling increasing number of files and users
   - Growth of storage space.
5. Handling of failures
   - Detection(may be impossible)
   - Exception handling(e.g. time-outs when waiting for a web resource)
   - Redundancy of data storage
   - Redundant routes in network
   - Replication of name tables in multiple domain name servers
6. Concurrency
   - Consistent scheduling of concurrent threads(so that dependencies are preserved e.g. in concurrent transitions)
   - Avoidance of dead and life lock problems.
7. Transparency: concealing the heterogeneous and distributed nature of the system so that it appears to the user like one system.

Resource sharing and the web challenges

Resources may be shared either in the form of printer, scanner, machine and so on.
Terms used in the web

A. **Services**: it is a distinct of a computer system that manages a collection of related resources and present functionality to users. For instance, we can access the shared file service to send document through the printing service.
B. **Server**: it means a running program on a networked computer that accepts request from program running on other computer to perform a service and respond appropriately.

WWW (World Wide Web)

It is an evolving system for publishing and accessing resources and service across the internet. Among the web browsers are Mozilla, fire fox, internet explorer, etc. and are used to retrieve and view documents of many types, view video streams and so on.

Properties of WWW (World Wide Web)

1. It is an open system and it can be extended and implemented in new ways without distributing its existing functionality.
2. The web is open with respect to the type of resources that can be published and shared on it.

Web characteristics

Heterogeneity

The internet enables users to access services and run application over a heterogeneity collection of computers network. It is applicable on the following:

a) Computer network
b) Computer hardware
c) Operating system
d) Programming language

The internet consist of many different sort of network, their differences are masked by fact that all of the computers attached to them use the Internet Protocols to communicate with one another.

Openness

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

Security

1. Information security: it depends on three components;
   a. Confidentiality: it is confidential and to protect the unauthorized individual.
   b. Integration: it deals with the protection against alteration and corruption.
   c. Availability: it is protection against interference with the mean to access the resource. Example: In banking, users send their credit card number across the internet
2. Denial of service attacks: this is a security problem whereby a user may wish to disrupt a service for some reasons.
3. Security on mobile code: it needs to be handled with care.

Failure handling

Software, hardware or program may produce incorrect result or stop before completing an intended computation due to the system failure
The following techniques could be employed in dealing with failure.

a. **Detecting failure**: its failure can be detected.
b. **Masking failure**: it is failure due to detect and can be hidden or made less server.

c. **Redundancy**: services can be made to tolerate failure by the use of redundant components. There should always be at least two different routes between any two routes on the internet.

Scalability

A system is scalable if it will remain effective when there is significant increase in the number of resources and the number of users. For a system with user to be scalable, the quantity of physical resources required to support them should be O(n) that is proportional to n.

Transparency

This is 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 collection of independent component.

DISTRIBUTED FILE SYSTEMS (DFS)

A **distributed file system** or **network file system** is any file system that allows access to files from multiple hosts sharing via a computer network. This type of file system is similar to the Distributed Objects i.e. software modules that are designed to work together, but reside either in multiple computers connected via a network or in different processes inside the same computer. The major difference between a DFS paradigm and a DO paradigm is that the resources (files) in DFS are much longer lived; DFS provides users with access to files and directories that are provided by one or more file servers.

ADVANTAGES OF DFS

A. Transparency means that any form of distributed system should hide its distributed nature from its users, appearing and functioning as a normal centralized system.
   There are many types of transparency:
   - Access transparency
   - Location transparency
   - Migration transparency
   - Relocation transparency
   - Replication transparency
   - Concurrent transparency
   - Persistence transparency
   - Security transparency

B. Flexibility
C. Reliability
D. Performance
E. Scalability

CLIENT'S PERSPECTIVE: FILE SERVICES

The file Service interface shows files as un-interpreted sequence of bytes that are associated with a set of attributes including information regarding security. There are two models involved in which the user has a choice, the upload / download model.

The Upload model: Users download files from the server and make modifications to them before uploading them back to the server. And this may cause problems when there are different users performing this operation.

The Download Model: All operations are performed at the server itself, with client simply sending commands to the server. This model makes it possible for the server to order all operations and therefore allow concurrent modifications to the files. The problem with this option is that users only have access to files that are connected with the server such that if there is loss of connection with the network, the user loses access to the files.

FILE ACCESS SEMANTICS

Due to the distributed nature of a DFS thereby making it difficult for user to see remote files just like local ones, we consider several semantics:

1. UNIX Semantics
2. Session Semantics
3. Immutable Files
4. Atomic Transactions

However, Atomic transaction is the best used although it is expensive.

SERVER'S PERSPECTIVE

Usage pattern about the expected use of a file system can be used to guide the design of a DFS. Also, implementation trades may depend on the requirements of a DFS which include supporting a large file system, supporting many users, the need for high performance, and the need for fault tolerance.

STATEFUL OR STATELESS SERVERS

The file servers that implement a distributed file service can be stateless or stateful. Stateless file servers do not store any session state i.e. every user request is treated independently, and not as part of a new or existing session. Stateful servers stores session state and may keep track of activities on the server.

Advantages of Stateless Server

a. They can easily recover from failure.
b. If users crash the server, they are not stuck with abandoned opened or locked files
c. The server implementation remains simple because it does not have to implement the state accounting associated with opening, closing and locking files.

Advantages of Stateful Server

i. They can provide better performance for clients.
ii. The server can make use of knowledge of access pattern to perform read-ahead and do other optimizations.
iii. They can also order users extra services such as file locking and remember read and write positions.

REPLICATION

The main approach to improving the performance and fault tolerance of a DFS is to replicate its content. A replicating DFS maintains multiple copies of files on different servers. This can prevent data loss, protect a system against down time of single server, and distributed the overall workload.

There are three approaches to replication in a DFS

1. Explicit Replication: In this case the client explicitly writes files to multiple servers.
2. Lazy File Replication: The server automatically copies files to other servers after the files are written.
3. Group File Replication: write request are simultaneously sent to a group of servers.

CACHING

In a DFS caching involves storing either a whole file or the result of file service operations. Caching can be performed at two locations:

- Server-Side
- Client-Side

- **Server-Side:** it makes use of file caching provided by the host operating system. It is transparent to the server and helps to improve the server's performance by reducing costly distance access.

- **Client-Side:**
  - **On disk Caching:** it involves the creation of temporary file on the client's disk.
  - **In-memory caching:** this stores the result of requests in the client-machine's memory.

NETWORK FILE SYSTEM (NFS)

NFS is a remote access DFS. The servers are stateless and it support caching but not replication. NFS can be used on diskless workstations so doesn't require local disk space for caching files. It does, however, support client-side caching and allows both file contents as well as file attributes to be cached.

ANDREW FILE SYSTEM (AFS)

AFS is a DFS that came out of the Andrew Research project at Carnagie Mellon University (CMU). Files and directories are stored on a collection of trusted servers called vice. Client processes accessing AFS redirected by the file system layer to a local user-level process called Venice (the AFS daemon), which then connect to the servers. The servers serve whole files, which are cached as a whole on the clients' local disks. For cached files a callback is installed on the corresponding server. After a process finishes modifying a file by closing it, the changes are written back to the server. The server then uses the callbacks to invalidate the file in other clients' caches, thereby reducing client's access time.

CODA

Coda is an experimental DFS developed at CMU. It provides client-side of whole files. The caching is implemented in a user-level cache process called Venus. In contrast to AFS, coda supports disconnected operation. This is done my making sure that a client always has up-to-date cached copies of files that they might require. This process is called file **Hoarding.** The system builds a user hoard database which it uses to update frequently used files using a process called a Hoard Walk. Coda allows files to be replicated on read/write servers.

GOOGLE FILE SYSTEM (GFS)

The GFS is a DFS developed to support a system with very different requirement than traditionally assumed when developing file systems. GFS support operations that typically involves large amounts of data, run distributed over very large clusters, and include much concurrent access to files. One of the key assumptions driving the design is that, because very large clusters are used, failure (of hardware or software resulting in crashes or corrupt data) is a regular occurrence rather than anomaly.

CEPH

This is a scalable, high performance research DFS. It targets systems with huge amounts of data (petascale systems) and like GFS, assumes that node failures are the norm, not an exception. It has three key design features. First, it decouples data and metadata by using a mapping function that maps from a file's unique ID too intelligent object storage devices (OSD) which stores the file's data, thus eliminating the need to store explicit allocation lists. Secondly, it adaptively and intelligently distributes responsibility of metadata to a cluster of metadata servers. Thirdly, ceph uses intelligent OSD's to reliably and automatically store data.