#### FILE ACCESSING MODELS

File access models are methods used for accessing remote files and the unit of data access. A distributed file system may use one of the following models to service a client’s file access request when the accessed file is remote:

##### Remote service model

* + - * Processing of a client’s request is performed at the server’s node.
      * Thus, the client’s request for file access is delivered across the network as a message to the server, the server machine performs the access request, and the result is sent to the client.
      * This need to minimize the number of messages sent and the overhead per message.

##### Data-caching model

* + - * This model attempts to reduce the network traffic of the previous model by caching the data obtained from the server node.
      * This takes advantage of the locality feature of the found in file accesses.
    - A replacement policy such as LRU is used to keep the cache size bounded.
    - This model reduces network traffic it has to deal with the cache coherency problem during writes, because the local cached copy of the data needs to be updated, the original file at the server node needs to be updated and copies in any other caches need to be updated.
    - The data-caching model offers the possibility of increased performance and greater system scalability because it reduces network traffic, contention for the network, and contention for the file servers. Hence almost all distributed file systems implement some form of caching.
    - In file systems that use the data-caching model, an important design issue is to decide the **unit of data transfer.**
    - This refers to the fraction of a file that is transferred to and form clients as a result of single read or write operation.

##### File-level transfer model

* + - * In this model when file data is to be transferred, the entire file is moved.
      * File needs to be transferred only once in response to client request and hence is more efficient than transferring page by page which requires more network protocol overhead.
      * This reduces server load and network traffic since it accesses the server only once. This has better scalability.
      * Once the entire file is cached at the client site, it is immune to server and network failures.
      * This model requires sufficient storage space on the client machine.
      * This approach fails for very large files, especially when the client runs on a diskless workstation.
      * If only a small fraction of a file is needed, moving the entire file is wasteful.

##### Block-level transfer model

* + - * File transfer takes place in file blocks.
      * A file block is a contiguous portion of a file and is of fixed length (can also be a equal to a virtual memory page size).
      * This does not require client nodes to have large storage space.
      * It eliminates the need to copy an entire file when only a small portion of the data is needed.
      * When an entire file is to be accessed, multiple server requests are needed, resulting in more network traffic and more network protocol overhead. NFS uses block-level transfer model.

##### Byte-level transfer model

* + - * Unit of transfer is a byte.
      * Model provides maximum flexibility because it allows storage and retrieval of an arbitrary amount of a file, specified by an offset within a file and length.
      * The drawback is that cache management is harder due to the variable-length data for different access requests.

##### Record-level transfer model

* + - * This model is used with structured files and the unit of transfer is the record.

#### FILE-SHARING SEMANTICS

* + - Multiple users may access a shared file simultaneously.
    - An important design issue for any file system is to define when modifications of file data made by a user are observable by other users.
    - The UNIX semantics is implemented in file systems for single CPU systems because it is the most desirable semantics and because it is easy to serialize all read/write requests.
    - Implementing UNIX semantics in a distributed file system is not easy.
    - One may think that this can be achieved in a distributed system by disallowing files to be cached at client nodes and allowing a shared file to be managed by only one file server that processes all read and write requests for the file strictly in the order in which it receives them.
    - However, even with this approach, there is a possibility that, due to network delays, client requests from different nodes may arrive and get processed at the server node in an order different from the actual order in which the requests were made.
    - Also, having all file access requests processed by a single server and disallowing caching on client nodes is not desirable in practice due to poor performance, poor scalability, and poor reliability of the distributed file system.
    - Hence distributed file systems implement a more relaxed semantics of file sharing.
    - Applications that need to guarantee UNIX semantics should provide mechanisms (e.g. mutex lock etc) themselves and not rely on the underlying semantics of sharing provided by the file system.

#### NAMING IN DISTRIBUTED SYSTEMS

* + - A Name is a string of bits used to refer to an entity.
    - We operate on an entity through its Access Point. The Address is the name of the access point.
    - The naming facility of a distributed operating system enables users and programs to assign character-string names to objects and subsequently use these names to refer to those objects.
    - The locating facility, which is an integral part of the naming facility, maps an object's name to the object's location in a distributed system.
    - The naming and locating facilities jointly form a naming system that provides the users with an abstraction of an object that hides the details of how and where an object is actually located in the network.
    - It provides a further level of abstraction when dealing with object replicas.
    - Given an object name, it returns a set of the locations of the object's replicas.
    - The naming system plays a very important role in achieving the goal of
      * location transparency,
      * facilitating transparent migration
      * Replication of objects, v object sharing.

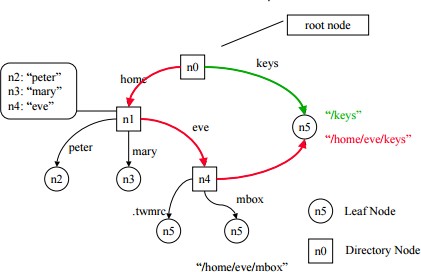
##### Example:

* Telephone as Access Point to a person.
* The Telephone Number then becomes the address of the person.
* Person can have several telephone numbers.
* Entity can have several addresses

##### Identifiers:

* + - * Identifiers are special names that uniquely identify an entity.
      * An identifier refers to at most one entity.
      * Each entity is referred to at most one identifier.
      * An identifier always refers to the same entity (never reused).

##### Namespaces:



**Fig 3.12: Namespaces**

* + - * Names are organized into Name Space.
      * Entities in a structured name space are named by a path name.
      * Leaf nodes represent named entities (e.g., files) and have only incoming edges .
      * Directory nodes have named outgoing edges and define the path used to find a leaf node

##### Attribute based naming

* This allows a user to search for an entity whose name is not known.
* Entities are associated with various attributes, which can have specific values.
* By specifying a collection of <attribute, value> pairs, a user can identify one (or more) entities
* Attribute based naming systems are also referred to as **directory services**, as opposed to naming systems.

##### Naming resolution

It is the process of looking up a name. The path name of the file is given as N:<label-1, label-2,…, label-n>

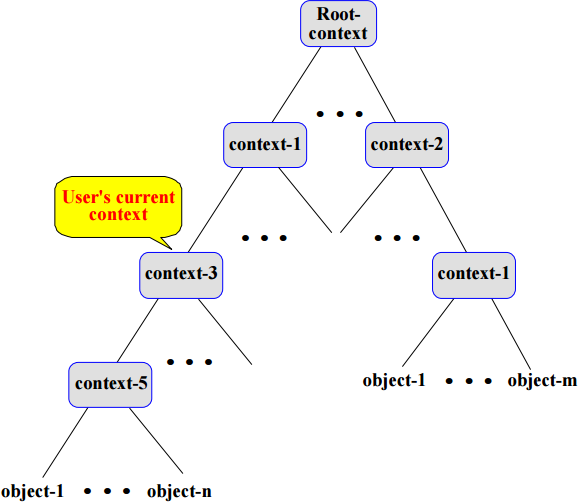


***Name resolution is the process of mapping an object'sname to the object's properties, such as its location.***

* + - Since an object's properties are stored and maintained by the authoritative name servers of that object, name resolution is basically the process of mapping an object's name to the authoritative name servers of that object.
    - Once an authoritative name server of the object has been located, operations can be invoked to read or update the object's properties.
    - Each name agent in a distributed system knows about at least one name server apriori.
    - To get a name resolved, a client first contacts its name agent, which in turn contacts a known name server, which may in turn contact other name servers.

##### Absolute and Relative Names

* A current working context is also known by the shorter names current context or working context.
* According to this concept, a user is always associated with a context that is his or her current context.
* A user can change his or her current context whenever he or she desires.



##### Fig 3.12: Tree structured name space

* An **absolute name** begins at the root context of the name space tree and follows a path down to the specified object, giving the context names on the path.
* On the other hand, a relative name defines a path from the current context to the specified object. It is called a relative name because it is "relative to” (start from) the user's current context.

In this method, a user may specify an object in any of the following ways:

1. Using the full (absolute) name
2. Using a relative name
3. Changing the current context first and then using a relative name

##### Implementing Name Spaces

Three layers used to implement such distributed name spaces

* + Global layer: root node and its children
  + Administrational layer: directory nodes within a single organization
  + Managerial layer

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Global** | **Administrational** | **Managerial** |
| Geographical scale of network | Worldwide | Organization | Department |
| Total number of nodes | Few | Many | Vast numbers |
| Responsiveness of lookups | Seconds | Milliseconds | Immediate |
| Updata propagation | Lazy | Immediate | Immediate |
| Number of replicas | Many | None or few | None |
| Is client-side caching applied? | Yes | Yes | Sometimes |

##### Name Space Distribution Iterative name resolution

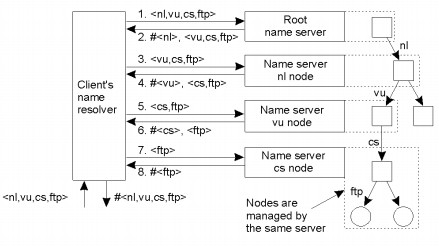
* Recursive name resolution puts a higher performance demand on each name server.
* Too high for global layer name servers
* In this method, the name agent forwards the name resolution request to the name server that stores the first context needed to start the resolution of the given name.
* After this, the name servers that store the contexts of the given pathname are recursively activated one after another until the authority attribute of the named

object is extracted from the context corresponding to the last component name of the pathname.

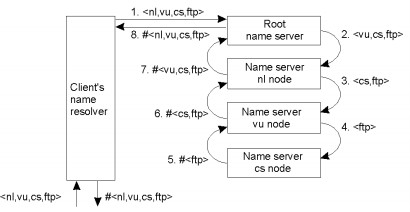
* + - The last name server returns the authority attribute to its previous name server, which then returns it to its own previous name server, and so on.
    - Finally, the fast name server that received the request from the name agent returns the authority attribute to the name agent

##### Advantages of recursive name resolution:

* Caching is more effective
* Communication costs may be reduced



##### Fig 3.12: Principle of iterative name resolution



**Fig 3.13: Principle of iterative name resolution**

##### Recursive Name resolution:

* + In this method, name servers do not call each other directly.
  + Rather, the name agent retains control over the resolution process and one by one calls each of the servers involved in the resolution process.
  + To continue the name resolution, the name agent sends a name resolution request along with the unresolved portion of the name to the next name server.
  + The process continues until the name agent receives the authority attribute of the named object

##### : Name Caches

Caching is an important technique for building responsive, scalable distributed systems. A cache can be maintained either by the client or the server or by both.

##### Types of Name caches

* + - * **Client name caches**

Caching at the client is an effective way of pushing the processing workload from the server out to client devices,if a client has the capacity.

##### Server based cache

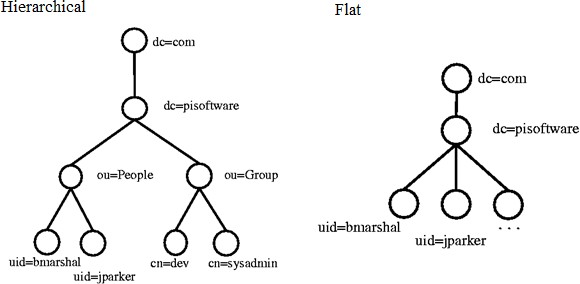
If result data is likely to be reused by multiple clients or if the client devices do not have the capacity then caching at the server is more effective.

##### Multi-level caching

Caches can be maintained at multiple levels. For example, caches can be maintained at all clients and all servers. Use of a cache at one level reduces the number of requests handled at the levels below.

##### Lightweight Directory Access Protocol (LDAP)

LDAP is a standard technology for building computer network directories. A network directory is a specialized database that stores information about devices, applications, people and other aspects of a computer network.

* 1. 

##### Fig 3.14: LDAP Tree structure

**Working of LDAP:**

* + - Client starts an LDAP session by connecting to an LDAP server.
    - The default port on TCP is 389.
    - Client sends operation requests to the server and the server sends responses in turn.
    - With some exceptions the client need not wait for a response before sending the next request. Server may send the responses in any order.
    - The client may request the following operations:
      * Start TLS : Optionally protect the connection with [Transport Layer Security](http://en.wikipedia.org/wiki/Transport_Layer_Security) (TLS), to have a more secure connection
      * Bind - [authenticate](http://en.wikipedia.org/wiki/Authentication) and specify LDAP protocol version
      * Search - search for and/or retrieve directory entries
      * Compare - test if a named entry contains a given attribute value
      * Add a new entry
      * Delete an entry
      * Modify an entry
      * Modify Distinguished Name (DN) - move or rename an entry
      * Abandon - abort a previous request

* + - * Extended Operation - generic operation used to define other operations
      * Unbind - close the connection (not the inverse of Bind)
    - In addition the server may send "Unsolicited Notifications" that are not responses to any request, e.g. before it times out a connection.

##### Directory Structure

* Directory is a tree of directory entries.
* Each entry consists of a set of attributes.
* An attribute has: a name , an attribute type or attribute description and one or more values
* Attributes are defined in a schema.
* Each entry has a unique identifier called Distinguished Name (DN).
* The DN consists of its Relative Distinguished Name (RDN) constructed from some attribute(s) in the entry.
* It is followed by the parent entry's DN.
* Think of the DN as a full filename and the RDN as a relative filename in a folder.
* DN may change over the lifetime of the entry.
* To reliably and unambiguously identify entries, a [UUID](http://en.wikipedia.org/wiki/UUID) might be provided in the set of the entry's operational attributes.

#### REVIEW QUESTIONS

##### PART - A

* + 1. **Define peer-to-peer communications.**

Peer-to-peer (P2P) is a decentralized communications model in which each party has the same capabilities and either party can initiate a communication session.

##### Give the features of Peer to Peer Systems

* + - * Large scale sharing of data and resources
      * No need for centralized management
      * Their design of P2P system must be in such a manner that each user contributes resources to the entire system.
      * All the nodes in apeer-to-peer system have the same functional capabilities and responsibilities.
      * The operation of P2P system does not depend on the centralized management.
      * The choice of an algorithm for theplacement of data across many hosts and the access of the data must balance the workload and ensure the availability without much overhead.

##### List the advantages of P2P systems over client/ server architecture

1. It is easy to install and so is the configuration of computers on the network.
2. All the resources and contents are shared by all the peers, unlike server-client architecture where Server shares all the contents and resources.
3. P2P is more reliable as central dependency is eliminated. Failure of one peer doesn‟t affect the functioning of other peers. In case of Client –Server network, if server goes down whole network gets affected.
4. There is no need for full-time System Administrator. Every user is the administrator of his machine. User can control their shared resources.
5. The over-all cost of building and maintaining this type of network is comparatively very less.

##### List the disadvantages of P2P systems over client/ server architecture

* 1. In this network, the whole system is decentralized thus it is difficult to administer. That is one person cannot determine the whole accessibility setting of whole network.
  2. Security in this system is very less viruses, spywares,trojans, etc malwares can easily transmitted this architecture.
  3. Data recovery or backup is very difficult. Each computer should have its own back-up system
  4. Lot of movies, music and other copyrighted files are transferred using this type of file transfer. P2P is the technology used in torrents.

##### Give the characteristics of peer-to-peer middleware.

* The P2P middleware must possess the following characteristics:
  + Global Scalability
  + Load Balancing
  + Local Optimization
  + Adjusting to dynamic host availability
  + Security of data
  + Anonymity, deniability, and resistance to censorship

##### Define routing overlay.

A routing overlay is a distributed algorithm for a middleware layer responsible for routing requests from any client to a host that holds the object to which the request is addressed.

##### Give the differences between Overlay networks and IP routing

|  |  |
| --- | --- |
| **IP** | **Overlay Network** |
| The scalability of IPV4 is limited to 232 nodes and IPv6 is 2128. | Peer to peer systems can address more objects using GUID. |
| The load balancing is done based on topology. | Traffic patterns are independent of topology since the object locations are randomized. |
| Routing tables are updated asynchronously. | Routing tables are updated both synchronously and synchronously. |
| Failure of one node does not degrade the performance much. Redundancy is introduced in IP. n-fold replication is costly. | Routes and object references can be replicated n-fold, ensuring tolerance of n failures of nodes or connections. |
| Each IP can map to only one node. | Messages are routed to the nearest replica of the target object. |
| Secure addressing is possible only between trusted nodes. | Secure communication is possible between limited trusted systems. |

##### What is Napster?

Napster was developed for peer –to-peer file sharing especially MP3 files.They are not fully peer-to-peer since it used central servers to maintain lists of connected systems and the files they provided, while actual transactions were conducted directly between machines.

##### Give the features of GUID.

* + - They are pure names or opaque identifiers that do not reveal anything about the locations of the objects.
    - They are the building blocks for routing overlays.
    - They are computed from all or part of the state of the object using a function that deliver a value that is very likely to be unique. Uniqueness is then checked against all other GUIDs.
    - They are not human understandable.

##### Give the types of routing overlays.

DHT – Distributed Hash Tables. GUIDs are stored based on hash values.

DOLR – Distributed Object Location and Routing. DOLR is a layer over the DHT that maps GUIDs and address of nodes. GUIDs host address is notified using the Publish() operation.

##### Define pastry.

Pastry is a generic, scalable and efficient substrate for peer-to-peer applications. Pastry nodes form a decentralized, self-organizing and fault-tolerant overlay network within the Internet.

##### Give the Capabilities of Pastry

* Mapping application objects to Pastry nodes
* Inserting objects
* Accessing objects
* Availability and persistence
* Diversity
* Load balancing
* Object caching
* Efficient, scalable information dissemination

##### What is tapestry?

Tapestry is a decentralized distributed system.It is an overlay network that implements simple key-based routing. Each node serves as both an object store and a router that applications can contact to obtain objects.

##### Give the differences between structured and unstructured networks

|  |  |
| --- | --- |
| **Structured Network** | **Unstructured Network** |
| This guarantees to locate objects and | This is self-organizing and resilient |
| can offer time and complexity bounds | to node failure. |
| on this operation. So it has relatively |  |
| low message overhead. |  |
| This needs complexoverlay structures, | Probabilistic and hence cannot offer |
| which can bedifficult and costly to | absolute guarantees on locating |
| achieve, especially in highly dynamic | objects; prone to excessive |
| environments. | messaging overhead which can |
|  | affect scalability |

* 1. **What is Gnutella?**

The Gnutella network is a fully decentralized, peer-to-peer application layer network that facilitates file sharing; and is built around an open protocol developed to enable host discovery, distributed search, and file transfer.

##### What is DFS?

The purpose of a distributed file system (DFS) is to allow users of physically distributed computers to share data and storage resources by using a common file system. A typical configuration for a DFS is a collection of workstations and mainframes connected by a local area network (LAN). Distributed file systems support the sharing of information in the form of files and hardware resources.

##### What are the requirements of distributed file systems?

Transparency is operating in such a way as to not be perceived by users. The distributed file system demands the following transparency requirements:

* Login Transparency: User can log in at any host with uniform login procedure and perceive a uniform view of the file system.
* Access Transparency: Client process on a host has uniform mechanism to access all files in system regardless of files are on local/remote host.
* Location Transparency: The names of the files do not reveal their physical location.
  + - Concurrency Transparency: An update to a file should not have effect on the correct execution of other process that is concurrently sharing a file.
    - Replication Transparency: Files may be replicated to provide redundancy for availability and also to permit concurrent access for efficiency.

##### What are the components in file structures?

File service architecture offers a clear separation of the main concerns in providing access to files is obtained by structuring the file service as three components:

* + - A flat file service
    - A directory service
    - A client module.

##### What is AFS?

This was developed by Carnegie Mellon University as part of Andrew distributed computing environments (in 1986).The public domain implementation is available on Linux (LinuxAFS). It was adopted as a basis for the DCE/DFS file system in the Open Software Foundation (OSF, www.opengroup.org) DEC (Distributed Computing Environment).Like NFS, AFS provides transparent access to remote shared files for UNIX programs running on workstations

##### Define file access modes.

File access models are methods used for accessing remote files and the unit of data access.

##### What is naming and locating facility?

The naming faciliy of a distributed operating system enables users and programs to assign character-string names to objects and subsequently use these names to refer to those objects. The locating faciliy, which is an integral part of the naming facility, maps an object's name to the object's location in a distributed system.

##### Define naming resolution.

Name resolution is the process of mapping an object'sname to the object's properties, such as its location.

##### What is absolute name?

An absolute name begins at the root context of the name space tree and follows a path down to the specified object, giving the context names on the path.

##### What is relative name?

A relative name defines a path from the current context to the specified object. It is called a relative name because it is "relative to” (start from) the user's current context.

##### What are the layers in name spaces?

Three layers used to implement such distributed name spaces

* + - Global layer: root node and its children
    - Administrational layer: directory nodes within a single organization
    - Managerial layer

##### What is LDAP?

LDAP is a standard technology for building computer network directories. A network directory is a specialized database that stores information about devices, applications, people and other aspects of a computer network.

##### PART – B

1. Explain about peer to peer communication.
2. Explain the working of routing overlays.
3. Describe Napster.
4. Write in detail about peer to peer middleware.
5. Explain about pastry.
6. Describe tapestry.
7. Write in detail about distributed file system.
8. Describe file service architecture.
9. Write in detail about Andrew file system.
10. Describe the file accessing models.
11. Elucidate the file sharing semantics.
12. Describe the naming in distributed systems.