Mca IV Sem – Distributed file system

Introduction

- In a computer system, a file is named object that comes into existence by explicit creation and is immune to temporary failure in the system and perziz until Explicitly destroyed.
- A file system is a subsystem of an operating system that perform file management activities va organization, storing, retieval, naming, sharing and protection of files.
- It allows programmers to use set of operation that characterized the file
- A File system provides an abstraction of storage device which is convenient for storing and retrieving information.
- It performs 2 main purposes:
    - Permanent storage of information – by storing a file into secondary storage media.
    - Sharing of information – information can e shared with different application.

Types of Service

Distributed file system typically provide 3 types of services:

1. Storage Service - it deals with allocation and management of space on a secondary storage devices that is used storage of files in the file system. It provides a logical view of the storage system by providing operation s for storing and retrieval data in them. Most system use magnetic disk as secondary storage devices. Therefore, the storage service is also known as disk service. Many system use fixed size blocks and hence is also known as block Service of the system.

2. True file service – It concerned with the operation on individual file such as operation for accessing and modifying the data in files and for creating snd deleting files. To perform these primitive file operation, it includes certain mechanism as File caching, file replication, concurrency control, data consistency, access control etc.

3. Name service - It provide a mapping between text names for files and references to files, file IDs. Name service is also known as Directory services. The directory service is responsible for performing directory related activities such as creation and deletion of directories, adding a new file to directory, deleting a file from directory, changing name etc.

Desirable Features of Good Distributed File System

1. Transparency : there are 4 types of transparency.
    - Structure transparency – generally DFS (distributed File System) use multiple file servers for better performance, scalability, reliability. Each file several is normally user process or kernel process. In multiple file server, the overall structure should be transparent to the clients. The client should not know the location and storage devices used.

- Access transparency – both local and remote files should be accessible in the same way. That is DFS should not distinguish between the local and remote files and the file system should automatically locate an accessed file and arrange for the transport of data to the client, s site.
- Naming transparency – the name of a file should give no hint as to where the file is located. A file should be allowed to move from one node to another in DFS without having the change the name of the file.
- Replication transparency – if the file is replicated on multiple nodes, both the existence of multiple copies and their locations should be hidden from the clients.

2. User mobility : In DFS, a user should not be forced to work on a specific node but should have the flexibility to work on different nodes at different times.
3. Performance : the performance of file system is usually measured as the average time needed to satisfy the client request. In centralized system, time include accessing time of secondary storage device And CPU Processing time. However in DFS it also include the network communication overhead.
4. Simplicity and ease of use : In an ideal design, the semantics of file system should be simple and easier to be used by the client. DFS should be able to support whole range of applications.
5. Scalability : It is known that distributed file system will grow by adding new machine or interconnecting more networks. Hence a good DFS should be designed to cope up with growth of nodes and users in system.
6. High availability: DFS should continue to function even in partial failure. Hence DFS require design in which both control and data are distributed. It should also have multiple and independent file server controlling multiple and independent storage device.
7. High reliability : DFS should automatically create backup copies ofor critical files that can used in the event of loss of original ones.
8. Data integrity : a file often shared by multiple users. Fir shared files integrity of file should ne maintain by proper synchronized form of concurrency control mechanism.
9. Security : A distributed file system should be secure Against any unauthorized access.
10. Heterogeneity : As a result of large scale, heterogeneity becomes inevitable in DFS. It provide flexibility to the users to use different computer platform for different applications. Therefore, a DFS should designed to allow a variety of workstation to work together.

File Model

DFS uses different file model listed below.

1. Unstructured and structured files :
    - Unstructured files- in this model, a file is unstructured sequence of data. There is no substructure and contents are also uninterpreted sequence of bytes. Hence the interpretation of the meaning and structure of data is stores in the files are entirely up to the application program.

    Structured files- in this model, a file is structured sequence of data. The records of different files of the same file system can be of different size, each having different

Properties. Here record is the smallest unit of file and are accesses in two ways. Thus structured files are of two types:

a) Non indexed records – here file records are accessed by specifying it's position within the files. For eg: the fifth record from beginning, second record from the end etc.
b) Indexed records – records have one or more key fields and can be addressed by specifying the value of the keys. For eg : B- tree or other suitable data structure or hash table etc.

2. Mutable and Immutable Files
   - Mutable – In this model, an update is perform on a file, it overwrite the it's old contents to produce the new contents. That is each file is altered by each update operation to keep only single stores file. Most OS uses the mutable file model.
   - Immutable – In this model, a file once created can only be deleted. It means if any update occurs, then a new version of files is created keeping the old version file without any change.

Flie Access Model

The manner in which clients request are processed depends on the file accessing methods a) remote files b) data transfer.

a) Accessing remote files – this model process the client's request for a file which is a remote file. This had two methods of processing.
   - Remote service model- this is also called server mode. When client request fir a file, the server machine performs the access request at server node snd forward it to client.
   - Data caching model – this is also known as client mode. When client request for a file, the data is copied from server's bide to client's node and is cached there. In order to keep the cache size bounded, LRU(least recently used) technique is used.

b) Unit of Data transfer – this model using data caching model depending on unit of data transfer. Unit of data transfer shows the fraction of files is transfer Ed to and from client's as result of single read and write operation. There are 4 types transfer model.
   - File level transfer model – in this model, whenever there is an operation requires file data to be transferred across the network between client and server, the whole file is moved.
     Advantages :
     - ✓ Transmitting an entire file for single request in single response is much better than pageby page transmission to several request.
     - ✓ It has better Scalability as it requires fewer accesss to file server.
     - ✓ Disk access routines on the server can be better optimized.
     - ✓ It becomes immune to server and network failure once whole file is received at the client's end.

- ✓ Finally it also helps in supporting heterogeneous workstation.

Disadvantages

- ✓ It requires sufficient storage space on the client's node fir storing all the required files in whole.
- ✓ If only small part of data file is required, there whole file transfer is of no use.
- ✓ Handling large size file transfer on diskless workstation is difficult.

- Block level transfer model – In this model, data file transfer between client and server take place in units of file blocks. A file block is contiguous portion of file and are fixed in length. Generally block size is equals to page size.
  Advantages
  - ✓ It needed less storage space.
  - ✓ It works for diskless workstation.
  - ✓ If small portion of file data is needed then only that block is transferred.

  Disadvantages
  - ✓ Performance goes down as when requires whole file, then block transfer requires lot of transfer operation.
  - ✓ It shows poor scalability.
  - ✓ It is not immu to network failure or server failure.

- Byte level transfer model – this model include byte level transfer of data. It provides maximum flexibility in storage and retrieval if file data.
- Record level transfer model – this model is commonly used in structured file model. It includes data transfer in units of records. Above three model ate used for unstructured.