Introduction to Distributed File Systems (DFS)

- Definition of DFS
- Importance of DFS in distributed computing
- Common use cases

- Definition of DFSA Distributed File System (DFS) is a file system that allows multiple users and applications to access and manage files stored across multiple servers or locations as if they were on a single local system. It provides seamless access to distributed storage while abstracting complexities such as network communication and data replication.Importance of DFS in Distributed ComputingDFS plays a crucial role in distributed computing by enabling efficient data sharing, redundancy, and scalability. It enhances performance by balancing the load among multiple storage nodes, ensures data availability through replication, and provides fault tolerance, reducing the risk of data loss in case of server failures. Additionally, DFS facilitates collaboration by allowing remote users to work with shared files in a unified system.

Features of DFS

- Transparency (Access, Location, Replication, Failure)
- Scalability
- Fault Tolerance
- Security

File Models in DFS

- Types of File Models:
  - Flat File Model
  - Hierarchical Model
  - Structured Model
- Comparison of different file models

File Models in DFS

- Types of File Models:
  - Flat File Model
  - Hierarchical Model
  - Structured Model
- Comparison of different file models

File Accessing Models

- Remote File Access Model
- Upload/Download Model
- Remote Procedure Call (RPC) Model
- Benefits and challenges of each model

5: File Caching Schemes

- Definition of File Caching
- Client-Side Caching
- Server-Side Caching
- Write-back vs. Write-through caching
- Cache Consistency mechanisms

File Replication

- Importance of file replication
- Replication strategies
  - Primary Copy Model
  - Quorum-based Replication
  - Epidemic Replication
- Trade-offs between consistency and availability

Case Study: Distributed File System (DSF)

- Overview of a real-world DFS
- Architecture and key features
- Performance considerations

Network File System (NFS)

- Introduction to NFS
- NFS Architecture
- Working Mechanism
- Pros and Cons of NFS

Designing Distributed Systems: Google Case Study

- Google File System (GFS) Overview
- Key Design Principles
  - Scalability
  - Fault Tolerance
  - High Performance
- GFS Architecture and Components
- Comparison with traditional DFS

Conclusion

- Summary of DFS and Name Services
- Future Trends in DFS
- Final thoughts and Q&A