Important Characteristics of Distributed File Systems

1. Scalability:

- DFS can be horizontally scaled by adding more nodes to the system, which allows it to handle more and more data as well as users.

- Examples: Hadoop Distributed File System (HDFS), Google File System (GFS).

2. Fault Tolerance:

- DFS offers data availability in the presence of node failures. This is achieved through replication (maintenance of multiple copies of data on various nodes) or erasure coding (reconstruction of data from fragments).

- HDFS replicates data blocks across multiple nodes.

3. High Availability:

- DFS provides that data is always available even under hardware failure or maintenance through redundancy and failover.

4. Transparency:

- DFS keeps users in the dark with regard to complexity of the underlying system. Some of the main types of transparency are:

- Location Transparency: Users aren't required to be aware of where the data is stored physically.

- Access Transparency: Users access the file system as they would a local file system.

5. Consistency:

- DFS must present all users with the same view of the data, even if multiple clients are accessing or modifying files simultaneously. This is typically achieved using locks or versioning.

6. Performance:

- DFS is optimized for low latency and high throughput, especially for large files and parallel access patterns.