# Big Data - Chapter 2 (Detailed 10 Marks Answers)

### 1. Differentiate between File System and Distributed File System

A File System is designed to manage and store files within a single computer, while a Distributed File System (DFS) extends file management across multiple systems connected through a network.

Differences:

1. Storage Location: A file system stores data on a single system, whereas DFS stores data on multiple computers.
2. Accessibility: A file system is accessible only within one system, while DFS allows network-wide access.
3. Fault Tolerance: File systems may lose data on system failure, but DFS ensures reliability through replication.
4. Scalability: Traditional file systems have limited scalability, while DFS can grow by adding more nodes.
5. Performance: File systems handle limited data, while DFS handles massive datasets using parallelism.
6. Examples: File systems - NTFS, FAT32; DFS - HDFS, Google File System (GFS).

Conclusion: Distributed File Systems are more efficient, reliable, and scalable, making them suitable for Big Data environments.

### 2. Write down features of Distributed File System

A Distributed File System (DFS) allows users to access and manage files stored on multiple computers as if they were on a single machine. It is essential for Big Data applications.

Key Features:

1. Transparency: Users can access data seamlessly without knowing the physical storage location.
2. Fault Tolerance: Data is replicated across nodes to prevent data loss in case of failure.
3. Scalability: New nodes can be added to increase storage and performance capacity.

4. Concurrency: Multiple users can access data simultaneously without conflict.

5. Data Replication: Ensures data availability even if some nodes fail.

6. Load Balancing: Distributes workload evenly across servers for better performance.

Conclusion: DFS ensures high performance, availability, and reliability, making it ideal for handling Big Data.

**3. Write down popular models of BIG DATA**

Big Data models represent frameworks for storing, processing, and analyzing vast amounts of information. The most popular models are:

1. Batch Processing Model: Processes large volumes of data at once. Example: Hadoop MapReduce.

2. Stream Processing Model: Handles real-time data streams. Example: Apache Storm, Spark Streaming.

3. Lambda Architecture: Combines batch and real-time processing for high efficiency.

4. Kappa Architecture: Focuses entirely on stream processing for continuous data flow.

5. Graph Model: Represents data as nodes and edges for relationship analysis. Example: Neo4j.

Conclusion: These models enable organizations to process and analyze data efficiently, based on their needs and data characteristics.

**4. Write down challenges of BIG DATA**

Handling Big Data involves several challenges due to its massive size and complexity.

Major Challenges:

1. Data Storage: Storing petabytes of data requires large-scale distributed systems.

2. Data Integration: Combining structured and unstructured data from various sources.

3. Data Security: Protecting data from unauthorized access and breaches.

4. Data Quality: Maintaining consistency, accuracy, and completeness.

5. Processing Speed: Real-time analytics require fast data processing tools.

6. Cost: Setting up Big Data infrastructure is expensive.

Conclusion: Overcoming these challenges requires advanced tools like Hadoop, Spark, and cloud-based solutions.

**5. Write note on the following:**

**(a) The Age of Internet Computing**

**(b) High Throughput Computing**

(a) The Age of Internet Computing:

Internet computing refers to using distributed internet-based systems to share resources and process data collaboratively. It forms the foundation for cloud computing and Big Data analytics.

Features:

- Provides global connectivity.
- Enables remote access and collaboration.
- Reduces infrastructure cost using virtualized resources.

Applications: E-commerce, social media, online education.

(b) High Throughput Computing:

High Throughput Computing (HTC) focuses on executing a large number of computing tasks over long periods. It aims for maximum computation capacity rather than quick responses.

Features:

- Handles large workloads with high efficiency.
- Utilizes distributed computing resources.
- Ideal for scientific simulations and data-intensive research.

Conclusion: Internet Computing connects global systems, while HTC ensures continuous large-scale computation for Big Data and scientific applications.

**6. What are the advantages and disadvantages of Distributed File System?**

A Distributed File System offers numerous benefits but also faces certain limitations.

Advantages:

1. Scalability: Can easily expand storage by adding more systems.
2. Fault Tolerance: Ensures data availability through replication.
3. Performance: Supports parallel data access for faster processing.
4. Transparency: Provides users with a unified view of distributed data.
5. Data Sharing: Facilitates easy sharing of files among multiple users.

Disadvantages:

1. Complexity: Requires sophisticated management and synchronization.
2. Security Issues: More prone to unauthorized access if not secured properly.
3. Network Dependency: Performance depends on network stability.
4. Cost: Setting up and maintaining a DFS can be expensive.

Conclusion: Despite its challenges, DFS remains a powerful system for managing Big Data efficiently across distributed environments.