BAHRIA UNIVERSITY (KARACHI CAMPUS)

Assignment-01

(Big Data Analytics)

Class: **BSE [4]-7 (B)** **(Morning)**

Course Instructor: **Dr. Salahuddin Shaikh** Submission Date: **31/10/2024**

Date: **(22/10/2024)** Max Marks**:** **5 M (CLO1-2)**

Student's Name: Shoaib Akhter Reg. No: 79290

Assignment Title:

"Advanced Use Cases and Performance Analysis of Hadoop and GFS"

Solution:

1. Case Study Analysis:

In the Big Data landscape, both Facebook and Google have tailored their distributed systems to manage vast data efficiently, although each approaches it uniquely to meet its business requirements.

- **Facebook's Use of Hadoop**:

Facebook relies on Hadoop's distributed file system (HDFS) to manage data generated by its large user base. This data ranges from user interactions—likes, shares, comments—to posts and browsing behaviors. With Hadoop's scalable architecture, Facebook is capable of batch-processing petabytes of data, supporting applications like personalized content recommendations, ad optimization, and overall user experience enhancements. Hadoop's batch processing enables Facebook to analyze log data, generate reports, and gain insights into user behavior patterns, which directly feed into its recommendation and ad-serving algorithms.

- **Google's Use of Colossus (evolved from GFS):**

Google's system, originally the Google File System (GFS) and later improved to Colossus, is designed for real-time data retrieval across distributed applications like Google Search, Gmail, and Google Drive. Colossus allows data replication across multiple servers, ensuring data integrity and high availability even during server outages. Colossus provides Google with reliable storage and fast data access across its services, particularly critical for cloud storage and real-time search functionalities. This system powers its ad network and analytics, providing the backbone for delivering data-driven insights at scale.

- **Analysis:**

Both Facebook and Google use distributed file systems for Big Data, but their approaches differ. Facebook's use of Hadoop emphasizes batch processing, well-suited for large-scale analysis and insights, while Google's Colossus is optimized for real-time data accessibility and high fault tolerance.

• 2. Performance Optimization in Hadoop:

• Data Storage Optimization:

- **Replication Factor:** Adjusting the replication factor (typically set to 3 by default) to balance storage costs and fault tolerance. Reducing it can save storage, but it may affect data availability.
- **Block Size:** Increasing the block size (default 128 MB) can improve processing speed for larger files as it reduces the number of data nodes needed and minimizes the I/O overhead.

• Processing Optimization:

- **MapReduce Tuning:** Fine-tuning parameters such as the number of map and reduce tasks, in-memory sorting buffer sizes, and speculative execution to prevent slow-running nodes from bottlenecking the workflow.
- **Data Compression:** Implementing data compression techniques (e.g., Snappy or LZO) reduces storage requirements and I/O costs, improving processing efficiency for large datasets.

3. Fault Tolerance in GFS:

• Node Failure and Data Recovery:

GFS manages fault tolerance through data replication and chunk servers, where data is divided into fixed-size chunks, each replicated across multiple nodes. When a node fails, a master node reallocates its chunks to other nodes, ensuring data remains available. GFS's master node monitors data replication and restores any missing chunks from replicas as needed.

• Improvements & Alternatives:

Technologies like **Apache Cassandra** and **Amazon S3** offer advanced replication and recovery mechanisms. For example, Cassandra's peer-to-peer architecture minimizes single points of failure, while Amazon S3 employs erasure coding to increase storage efficiency and data resilience without over-replicating data.

4. Comparison of Performance Metrics:

Metric	Hadoop (HDFS)	GFS/Colossus
Data Processing	Optimized for batch processing	Real-time data retrieval
Fault Tolerance	Achieved through replication	Enhanced via data chunk replication
Scalability	High, but depends on block tuning	Extremely high due to advanced replication methods
Storage Efficiency	Moderate; replication-based	High; uses erasure coding in Colossus
Accessibility	Primarily sequential, for big jobs	Designed for immediate data access

5. Conclusion:

Both Hadoop and GFS/Colossus serve as foundational systems in Big Data management, each with distinct advantages. **Hadoop** is advantageous for batch processing, making it suitable for applications requiring extensive data analysis and periodic insights. However, it can lag in real-time applications due to its sequential processing. **GFS/Colossus**, on the other hand, is superior in real-time data retrieval and accessibility, a necessity for Google's real-time services. The challenges both systems face involve managing storage costs, ensuring fault tolerance, and meeting the scalability demands of their respective companies. In today's landscape, advancements like cloud-native storage solutions and distributed, fault-tolerant databases offer complementary solutions that enhance both batch and real-time data handling.