International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Performance Evaluation of Apache Hadoop and Apache Spark in Large-Scale Data Processing

Guide :- Mrs. Anjali Dandekar anjali.dandekar@ruparel.edu Assistant Professor

MES "D.G Ruparel College of Arts, Science and Commerce"

Matunga West

Sr. No.	Author Name
1	**Mrs. Shrutika Dave**
2	**Mrs. Ruchi Bandekar**
3	**Ms. Dipesh Zagade**
4	**Ms. Vedant Mhapankar**

Abstract

In the era of big data, efficient processing of large datasets is essential for organizations. Apache Hadoop and Apache Spark are two widely used frameworks for distributed data processing. This paper evaluates their performance based on speed, execution model, resource utilization, and real-world applications. The findings suggest that Spark outperforms Hadoop in terms of speed due to its in-memory processing, while Hadoop remains a reliable solution for large-scale batch processing.

Keywords

Keywords: Big Data, Apache Hadoop, Apache Spark, Distributed Computing, In-Memory Processing, Performance Analysis, Data Processing Frameworks

I. Introduction

The rapid growth of data from digital platforms has created a need for scalable processing systems. Traditional databases cannot handle such volumes efficiently, leading to the development of distributed frameworks like Hadoop and Spark.

Hadoop introduced a disk-based processing model using MapReduce, whereas Spark improved performance by enabling in-memory computation. This paper compares both frameworks to evaluate their efficiency in large-scale data processing.

II. Overview of Technologies

2.1 Apache Hadoop

Hadoop is an open-source framework that processes large datasets using distributed storage (HDFS) and the MapReduce model. It ensures fault tolerance but suffers from slower execution due to heavy disk usage.

2.2 Apache Spark

Spark is a fast and flexible data processing engine that uses in-memory computation. It supports real-time analytics, machine learning, and iterative processing, making it more efficient than Hadoop in many scenarios.

III. Comparative Performance Analysis

3.1 Execution Time Comparison

Comparative Analysis Table		
Feature	**Apache Hadoop**	**Apache Spark**
Processing Type	Batch Processing	Batch + Real-time Processing
Data Storage	Uses HDFS	Uses HDFS / other storage systems
Programming Complexity	High (MapReduce coding)	Low (Simple APIs)
Fault Tolerance	High (Data replication)	High (Lineage-based recovery)
Cost	Low (Disk storage)	Higher (Requires more RAM)
Use Cases	Data warehousing, ETL	ML, streaming, real-time analytics

Performance Comparison

1. Processing Speed Comparison (Relative Scale)

Hadoop : [redacted] **(Slow) Spark :** [redacted] **(Fast)**

Spark performs up to 10–100x faster than Hadoop in many scenarios due to in-memory computation.

2. Resource Utilization

Hadoop (Disk Usage) : [redacted]

Spark (RAM Usage) : [redacted]

Hadoop relies heavily on disk I/O, while Spark depends on RAM, making it faster but more resource-intensive.

3. Ease of Development

Hadoop : ▓▓▓▒▒▒▒▒▒▒▒▒ (Complex) Spark : ▓▓▓▓▓▓▓▓▓▒ (Easy)

Spark provides user-friendly APIs (Python, Scala), reducing development time.

IV. Detailed Analysis

Speed and Efficiency

Spark outperforms Hadoop due to its in-memory processing. Hadoop writes intermediate data to disk, increasing latency. Spark avoids this by storing data in memory, resulting in faster execution.

Architecture Difference

Hadoop follows a **MapReduce model**, while Spark uses a **DAG (Directed Acyclic Graph)** model. DAG allows better optimization and parallel execution of tasks.

Cost Consideration

Hadoop is more cost-effective as it uses cheaper disk storage. Spark requires high RAM, which increases infrastructure costs, especially for large clusters.

Application Type	Best Choice	Reason
Batch Processing	Hadoop	Efficient for large static data
Real-time Analytics	Spark	Fast processing speed
Machine Learning	Spark	Built-in ML libraries
Data Storage	Hadoop	Reliable distributed storage
Interactive Queries	Spark	Quick response time

V. Discussion

The comparison clearly shows that Spark provides superior performance in terms of speed and flexibility. However, Hadoop remains relevant due to its strong storage capabilities and lower cost.

In practical environments, both technologies are often used together. Hadoop handles storage through HDFS, while Spark processes the data efficiently. This hybrid approach balances cost and performance.

VI. Future Scope

Future research can explore integration with cloud platforms and improvements in memory optimization to further enhance performance.

VII. Conclusion

Apache Hadoop and Apache Spark both play important roles in big data processing. Hadoop is suitable for large-scale storage and batch processing, while Spark is ideal for real-time analytics and high-speed computation. The choice depends on application requirements, budget, and processing needs.

References

1. White, T. (2015). Hadoop: The Definitive Guide.
2. Zaharia, M. et al. (2016). Apache Spark: A Unified Engine for Big Data Processing.
3. Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing.
4. Karau, H. et al. (2015). Learning Spark.