BIG DATA PROCESSING

TERM PROJECT PROPOSAL

Name:	Aditya Patel
Roll Number:	202203027
Name:	Dharmesh Kota
ID:	202203038

Problem Area

The rapid growth of massive datasets from various sources such as social media, weather sensors, and mobile devices has created a pressing need for large-scale data processing in domains like machine learning, climate science, and social media analytics. Many applications in these domains rely on efficient and scalable matrix computations. However, traditional matrix multiplication approaches, like the naive $O(n^3)$ algorithm, become computationally prohibitive with increasing matrix sizes.

Big Data processing frameworks such as Hadoop MapReduce and Spark have emerged as popular solutions for distributed computing. However, existing methods of matrix multiplication, including those in Spark's MLlib and frameworks like Marlin, still suffer from inefficiencies. These methods typically require 8 block multiplications and retain a time complexity of $O(n^3)$, making them less suitable for large-scale datasets.

The primary objective of this project is to address these inefficiencies by implementing **Strassen's Matrix Multiplication Algorithm** on a **distributed Spark framework**. Strassen's algorithm, which reduces the time complexity to $O(n^{2.807})$, uses fewer block multiplications, offering the potential for more efficient matrix computations. This project aims to investigate whether these theoretical improvements translate into practical reductions in execution time when applied to large, distributed datasets.

Expected Outcome

The expected outcomes of this project are as follows:

- **Efficient Implementation**: Develop a distributed implementation of Strassen's Matrix Multiplication Algorithm in Spark, which is expected to outperform traditional block matrix multiplication methods, particularly for large matrix sizes.
- **Reduced Time Complexity**: By reducing the time complexity from $O(n^3)$ to $O(n^{2.807})$, we anticipate faster execution times for larger datasets compared to naive matrix multiplication.
- **Scalability**: Demonstrate improved scalability of the algorithm when deployed on distributed systems, where it effectively handles large datasets while minimizing computational overhead.
- **Comparative Analysis**: Conduct a comprehensive comparison of execution times between the Strassen-based implementation and existing distributed matrix multiplication algorithms in Spark (such as MLlib and Marlin) to evaluate the practical advantages of Strassen's approach.
- **Insight into Applicability**: Gain deeper insights into the real-world applicability of distributed Strassen's algorithm and assess whether the theoretical time complexity gains result in significant improvements in distributed environments.

Selected Readings

Misra, Chandan, Sourangshu Bhattacharya, and Soumya K. Ghosh. ”Stark: Fast and Scalable Strassen's Matrix Multiplication using Apache Spark.” *IEEE Transactions on Big Data*, 2020.