Matrix Multiplication using MapReduce

Aim:

To perform Matrix Multiplication using MapReduce in Hadoop framework.

Theory:

Matrix multiplication is a common operation in scientific and data analysis applications. Given two matrices A (m × n) and B (n × p), their product C = A × B is a matrix of size (m × p), where: $C[i][j] = \Sigma (A[i][k] \times B[k][j])$ In MapReduce, this multiplication is distributed across multiple mappers and reducers: • Mapper Phase: Each mapper takes input elements of matrices A and B, and emits key-value pairs in the form (i, j) as key and partial multiplication results as values. • Shuffle & Sort Phase: Framework groups all values with the same key (i, j). • Reducer Phase: The reducer aggregates the partial results for each (i, j) and computes the final sum for C[i][j].

Algorithm / Steps:

1. Input: Two matrices A and B stored as text files in HDFS. Each line format: A,i,j,value B,i,j,value
2. Mapper: Emits key-value pairs for all columns (for A) or rows (for B). 3. Reducer: Collects all values with same key (i,j), multiplies and sums them. 4. Output: Resulting matrix C written to HDFS.

Sample Input:

Matrix A:
A 0 0 1
A 0 1 2
A 1 0 3
A 1 1 4

Matrix B:
B 0 0 5
B 0 1 6
B 1 0 7
B 1 1 8

Reducer Output (Final Result):

Matrix C = A × B
C 0 0 19
C 0 1 22
C 1 0 43
C 1 1 50

Result:

Matrix multiplication was successfully performed using the MapReduce model. The output matrix C is obtained as expected and stored in HDFS.

Viva Questions:

1. What is the main advantage of using MapReduce for matrix multiplication?
→ It distributes computation across multiple nodes for parallel processing.

2. What are the inputs and outputs of Mapper and Reducer?
→ Mapper outputs key-value pairs; Reducer outputs the final matrix elements.

3. Why is Hadoop suitable for this operation?
→ Because it handles large datasets efficiently using distributed computing.

4. What are the real-world applications?
→ Machine Learning, Data Mining, Image Processing, and Scientific Computation.