Appendix – Benchmark

Control-implementation and Main Function:

```
public static int[][] multiplyWithoutThreads(int[][] A, int[][] B) {
    int aRows = A.length;
    int aColumns = A[0].length;
    int bRows = B.length;
    int bColumns = B[0].length;

    if (aColumns != bRows) {
        throw new IllegalArgumentException("A:Columns: " + aColumns + " did not match B:Rows " + bRows +
".");
    }

    int[][] C = new int[aRows][bColumns];

    for (int row = 0; row < aRows; row++) {
        for (int col = 0; col < bColumns; col++) {
            for (int k = 0; k < aColumns; k++) {
                C[row][col] += A[row][k] * B[k][col];
            }
        }
    }

    return C;
}

public static void main(String[] args) {
    // Define matrix size
    final int MATRIX_SIZE = 1000;  // Adjust based on your testing needs

    // Create two random matrices
    int[][] A = generateRandomMatrix(MATRIX_SIZE, MATRIX_SIZE);
    int[][] B = generateRandomMatrix(MATRIX_SIZE, MATRIX_SIZE);
    // Measure time for matrix multiplication using threads
    long startTimeWithThreads = System.currentTimeMillis();
    int[][] resultWithThreads = multiplyWithThreads(A, B);
    long endTimeWithThreads = System.currentTimeMillis();

    // Measure time for matrix multiplication without threads
    long startTimeWithoutThreads = System.currentTimeMillis();
    int[][] resultWithoutThreads = multiplyWithoutThreads(A, B);
    long endTimeWithoutThreads = System.currentTimeMillis();

    // Compare the results of both methods (for correctness)
    boolean areEqual = matricesAreEqual(resultWithThreads, resultWithoutThreads);
    System.out.println("Matrices equal: " + areEqual);

    // Print the time taken for both methods
    System.out.println("Time taken with threads: " + (endTimeWithThreads - startTimeWithThreads) +
"ms");
    System.out.println("Time taken without threads: " + (endTimeWithoutThreads -
startTimeWithoutThreads) + "ms");
}

// Helper function to generate random matrices
public static int[][] generateRandomMatrix(int rows, int cols) {
    int[][] matrix = new int[rows][cols];
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            matrix[i][j] = (int) (Math.random() * 10);  // Random values between 0 and 9
        }
    }
    return matrix;
}

// Helper function to check if two matrices are equal
public static boolean matricesAreEqual(int[][] A, int[][] B) {
    if (A.length != B.length || A[0].length != B[0].length) {
        return false;
    }
    for (int i = 0; i < A.length; i++) {
        for (int j = 0; j < A[0].length; j++) {
            if (A[i][j] != B[i][j]) {
                return false;
            }
        }
    }
    return true;
}
```