Operating Systems–2: Spring 2024
Programming Assignment1:
Efficient Matrix Squaring
Report

Prabhat - CO22BTECH11009

Program Overview:

The overview of the program is as follows:

- The program contains two code files, namely **Assgn1_Src-CO22BTECH11009_chunk.cpp** and **Assgn1_Src-CO22BTECH11009_mixed.cpp**. The first one performs parallel matrix multiplication using the chunks method, while the later one performs the same using the mixed method.

- The code files take one input file named **inp.txt**, which contains the number of rows of the square matrix A (N), the number of threads (K), and the matrix A.

- Using these, the code files perform parallel matrix multiplication to produce the output files **out_chunk.txt** by the chunk method and **out_mixed.txt** by the mixed method. These output files contain the resultant square matrix C, such that C = A * A, and the time taken to perform the threads by these methods.

Chunk Method:

The low-level design of the program is as follows:

- N is the size of the vectors A and C.
- K is the number of threads.
- p is the size of each chunk, calculated by N/K.
- A loop is used to create K threads. Each thread performs a multiplication operation on a different chunk of the vector A.
- The start and end variables determine the range of indices that each thread will work on.
- If it's the last thread (i == K - 1), it will handle all remaining elements (in case N is not a multiple of K).
- The multiply function multiplies elements in vector A and stores the result in vector C.
- After all threads are created, another loop is used to join all threads, ensuring they all complete their execution before the function chunk returns.

Mixed Method:

The low-level design of the program is as follows:

- N is the size of the vectors A and C.
- K is the number of threads.
- A loop is used to create K threads. Each thread performs a multiplication operation on a different set of elements of the vector A.
- The multiply function multiplies elements in vector A and stores the result in vector C. It is called with N, i, and K as arguments, which suggests that each thread is working on every K'th element starting from the i'th element.
- After all threads are created, another loop is used to join all threads, ensuring they all complete their execution before the function mixed returns.

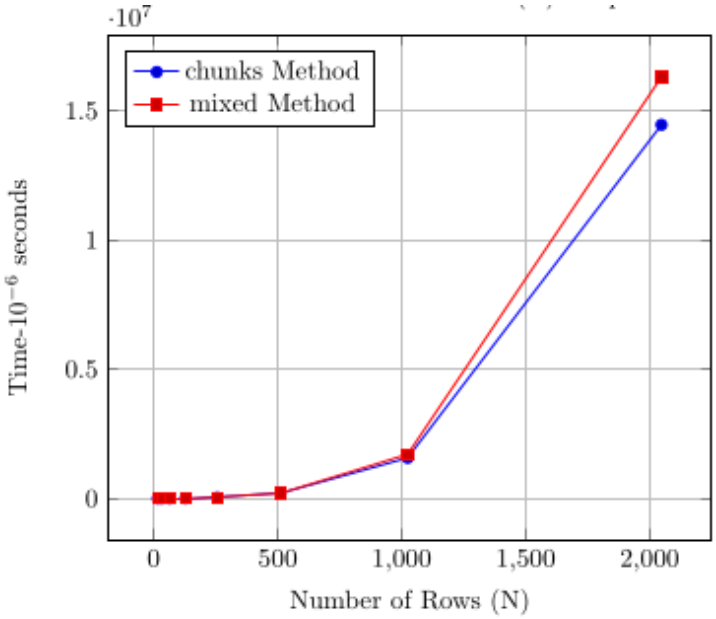

Figure 1: Time vs Number of Rows,

Observations:

The observations made from the Graph are as follows:

(A) Figure 1 (Time vs Number of Rows, N):

- For both methods, chunks method and mixed method, as the number of rows increases, the time taken for program completion also increases. This suggests that the computational complexity of both methods increases with the size of the matrix.
- The increase in time is much steeper for the "mixed Method" compared to the "chunks Method". This indicates that the "mixed Method" has a higher computational complexity and is less efficient than the "chunks Method" for larger matrices.
- The "chunks Method" appears to be more efficient for this task, especially as the size of the matrix increases.

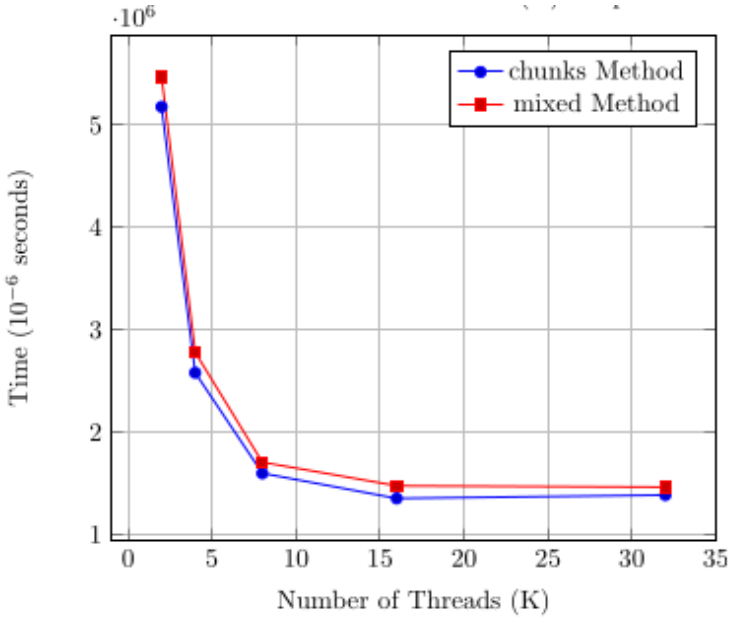

Figure 2: Time vs Number of Threads, K

Observations:

The observations made from the Graph are as follows:

(B) Figure 2 (Time vs Number of Threads, K):

- For both methods, chunks method and mixed method, as the number of threads increases, the time taken for program completion decreases significantly. This suggests that both methods benefit from parallel processing.
- The decrease in time is much steeper for the "chunks Method" compared to the "mixed Method". This indicates that the "chunks Method" more efficient than the "mixed Method" for larger number of threads.
- However, both methods show similar trends, suggesting that the number of threads is a more significant factor in time performance than the method used.