OS2 Assignment :1

Programming assignment on p_thread

Goal:- This assignment aims to perform parallel matrix multiplication in C++.

a)Low -level design of my program:

Three method I used to show use of multithreadiing.

1)chunk method:

Where I divided matrix into into chunks of size p = N/K. Then thread1 will compute .the values of the rows 1 to p of C; thread2 will compute the values of the rows p + 1 to 2* p; thread3 will compute the values of the rows 2*p + 1 to 3*p and so on. Thus, the thread thi will be responsible for computing the rows corresponding to chunk i.

Ex: N=10, k=4
So p=N/K=2 . and 4 threads.
So thread1 will do calculator for 1 and 2
Thread2 —> 3 and 4
thread 3—>5 and 6
Thread 4—>7 ,8,9,10
IN my code the last thread will cover all remaining rows.

2)mixed method:

Here, the rows of the C matrix are evenly distributed among the threads. Thread1 will be responsible for the following rows of the C matrix: 1, k+1, 2*k+1 ....,. Similarly, thread2 will be responsible for the following rows of the C matrix: 2, k+2, 2*k+2 ....,. This pattern continues for all the threads.

Ex: N=10, k=4
So p=N/K=2 and . 4 threads.
So thread1 will do calculations for 1, 5 and 9.
Thread2 —> 2 , 6 and 10
thread 3—>3 and 7
Thread 4—>4and ,8

3)Extra method(random selection of rows):

Here , if K threads are made. Then each thread will choose at least N/K rows randomly. And do its computation. Last thread will do the calculation of all remaining rows. And all threads together will complete the matrix.

ex:N=10 and K=4

So except last, all threads will choose 2 rows.

Let thread1 choose 5 and 6.

So reaming thread options are {1,2,3,4,7,8,9,10}

And goes on..

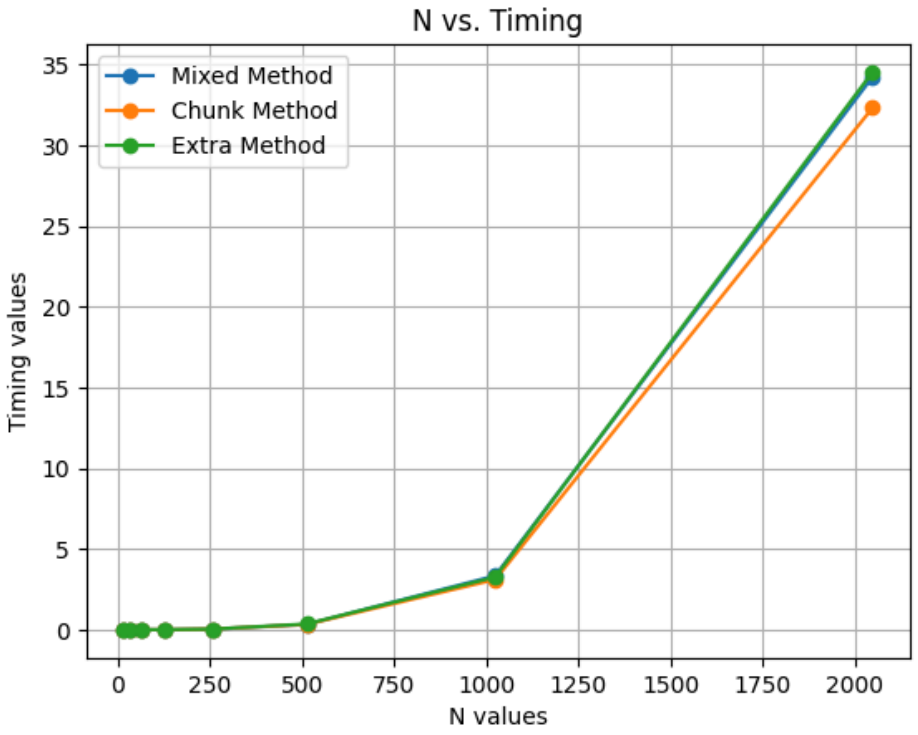
For last threads 4 rows will remains.and last thread will do their calculations

(b) The graph plots and its analysis.

Time vs. Size, N:

The graph below illustrates the time taken for matrix squaring as the size of the input matrix (N) increases. It compares the performance of chunk ,mixed approach and random choosing row approach.

- From the graph we can see although they have nearly equal timing but. Mixed approach and Random(Extra method) approach have slightly greater timing. Timing also varies from matrix to matrix.
- Time is increasing with the value of N as more calculations need to be done for higher N.



-

Time vs. Threads (K):

- Generally time decreases with increase in number of threads. And we can see it.
- However, the performance benefit of parallelism is more noticeable for larger matrix sizes.
- For all threads methods graphs are nearly the same.but for some values we can see mixed chunk and random method are more efficient.