CS301 HPC : Lab-3

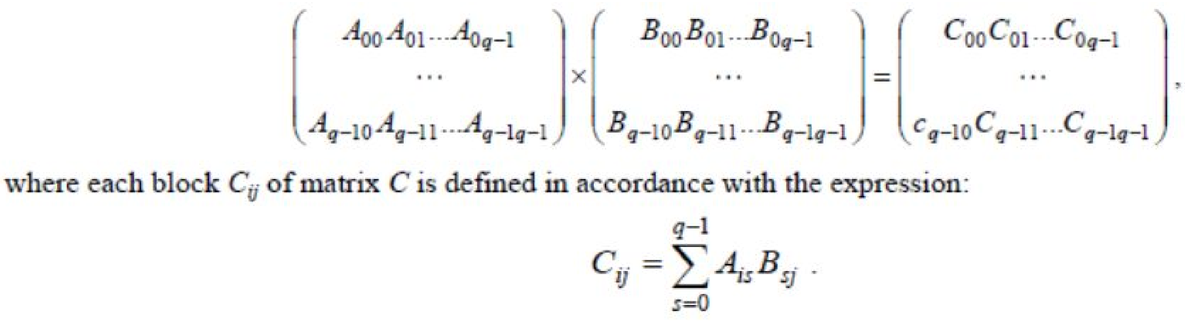
Parallel Matrix Multiplication (MM):

- Parallel implementation of the serial code with different loop ordering which was done in the previous assignment.
- Create n x n matrix; take n in multiples of 2 while increasing the size of matrix [e.g. 4, 8, 16, 32, 64 go up to 1024 or even 2048].
- Compare with serial results to check whether the output of the parallel code is correct or not.
- For the serial implementation, the code was optimized by utilizing the cache locality *(Transpose of the matrix before multiplication).* Check if you can get similar performance improvements.

Parallel Block Matrix Multiplication:

- Parallel implementation of the serial code of Block matrix multiplication which was done earlier.
- Vary block sizes and comment on the changes in performance.

You can assume all the matrices are square of n×n size, the number of vertical blocks and the number of horizontal blocks are the same and are equal to q (i.e. the size of all block is equal to k×k, k=n/q). Multiplying the matrices A and B as blocks may be presented as follows:



$$\begin{pmatrix} A_{00}A_{01}\cdots A_{0q-1} \\ \cdots \\ A_{q-10}A_{q-11}\cdots A_{q-1q-1} \end{pmatrix} \times \begin{pmatrix} B_{00}B_{01}\cdots B_{0q-1} \\ \cdots \\ B_{q-10}B_{q-11}\cdots B_{q-1q-1} \end{pmatrix} = \begin{pmatrix} C_{00}C_{01}\cdots C_{0q-1} \\ \cdots \\ c_{q-10}C_{q-11}\cdots C_{q-1q-1} \end{pmatrix},$$

where each block C_{ij} of matrix C is defined in accordance with the expression:

$$C_{ij} = \sum_{s=0}^{q-1} A_{is} B_{sj} \ .$$

The basic subtask = computing of the elements of a block of the matrix C. The basic subtasks should have the corresponding sets of rows of the matrix A and columns of the matrix B. Remember, we have two outer loops that give us the location of the block, then two inner loops to visit each of the elements of that block, and finally a loop to do the calculation. Do you think “critical” is necessary during OpenMP computation?

Submission Guidelines:

Doubts to be addressed through google-discussion-board.

Make your report brief and to the point; highlight the 10 important issues listed below (observations + supporting explanations).

Report generation using Using ***www.letshpc.org***

All the assignments have to be supplemented with a **brief write-up or ppt with the following details** (wherever necessary):

1. Context:
   - Brief description of the problem.
   - Complexity of the algorithm (serial).
   - Possible speedup (theoretical).
   - Profiling information (e.g. gprof). Serial time.
   - Optimization strategy.
   - Problems faced in parallelization and possible solutions.
2. Hardware details: CPU model, memory information details, no of cores, compiler, optimization flags if used, precision used.
3. Input parameters. Output. Make sure results from serial and parallel are same.
4. Parallel overhead time. (openmp version on 1 core vs serial without openmp)
5. Problem Size vs Time (Serial, parallel) **curve**. Speedup curve. Observations and comments about the results.
6. If more than one implementation, curves for all algorithms in the same plot.
7. Wherever necessary use log scale and auxiliary units.
8. Problem size vs. speedup curve.
9. No. of cores vs. speedup curve for a couple of problem sizes.
10. Measure performance in MFLOPS/sec.

Assignments include submission of codes [optimized serial and parallel codes (multiple versions if applicable) with necessary comments inside the code].
