**MTCS-103(P) Parallel Processing**

**Labwork 16 Report**

**Sequential code to perform MM (square matrices):**

1. This file (IterRowOrientedMM.c) contains sequential program to perform iterative roworiented Matrix Multiplication and verified correctness for input size 50. Compiled and ran program for various input sizes. Execution time for various size of matrix is tabulated.
2. This file (RecurBlockwiseMM.c) contains sequential program to perform recursive blockwise Matrix Multiplication and verified correctness for input size 50. Compiled and ran program for various input sizes. Execution time for various size of matrix is tabulated.
3. The execution time for various size of Matrix are recorded.

Table

Description automatically generated

Table

Description automatically generated

1. A qualitative comparison of the two methods based on the performance of your codes.

For iterative roworiented matrix multiplication: During each iteration of the outer i loop, every element of matrix B is read. If matrix B is too large for the cache, then later elements read into cache displace earlier elements read into cache, meaning that in the next iteration of the loop indexed by i, all of the elements of B will need to be read into cache again. Hence once the matrices reach a certain size, the cache hit rate falls dramatically, lowering the performance of the CPU.

For recursive blockwise matrix multiplication: During each iteration of the outer i loop, every element of matrix B is read. If matrix B is too large to fit into cache, we can divide it into four pieces and use the idea of block matrix multiplication to compute C. If block B, j is too large to fit into cache, we can apply this idea recursively until we have blocks that do fit in cache.

1. What can be concluded regarding the relationship of memory size and

execution time, across the memory hierarchy?

For iterative roworiented MM: If memory size is smaller, then the execution time will increase rapidly.

For recursive blockwise MM: If memory size is smaller, then execution time will increase. If memory size is larger, then execution time will decrease.

1. Learnings:

* When matrix B no longer fits in the cache, the performance of the row-oriented matrix multiplication algorithm drops sharply.
* The recursive algorithm maintains high performance, even as the sizes of the matrices grow well beyond the cache capacity.
* The block-oriented matrix multiplication algorithm keeps the cache hit rate high and achieves better performance than the row-oriented algorithm.

**List of files Attached:**

**For iterative roworiented MM:**

**C Parallel file:** IterRowOrientedMM.c

**Compiled:** gcc -o IterRowOrientedMM -fopenmp IterRowOrientedMM.c

**Runned:** ./IterRowOrientedMM inputs50

**For Recursive Blockwise MM:**

**C Parallel file:** RecurBlockwiseMM.c

**Compiled:** gcc -o RecurBlockwiseMM -fopenmp RecurBlockwiseMM.c

**Runned:** ./RecurBlockwiseMM inputs50