SC5809 - Advanced High Performance Computing

Problem Set #1 – Serial optimization

Part 1

Define the two performance models for a dense matrix-matrix multiplication algorithm under each of the following assumptions:

1. Assume data is read once and remains into cache. No additional costs to access the data in cache.
2. Assume that all data fits into cache except for one of the matrices.

Report the models in terms of cost per floating point operation (c), read from memory (r), write to memory (w) and size of the matrices (N).

Choose (and explain) the best values for c, r, w and calculate the time estimate based on the previous models.

Notes:

- For simplicity, use only square matrices.
- Ignore cache lines related issues in the model.
- In estimating the time, consider r=w.

Part 2

Write a code that performs the calculation in Part 1 and compare the actual time with the estimated ones.

Notes:

- Run your code with different matrix sizes where the biggest size must not fit into L3 cache.
- To avoid the compiler to replace the matrix-matrix multiply with an optimized version, add a call to a dummy function declared in a different file in the outer loop (i.e. after the `mysecond()` call).
- Compile with `-O0` to avoid additional compiler optimization.

Part 3

Optimize the code in Part 2 by means of the techniques illustrated in class, especially blocking. Report the and discuss the timings.