CSE 611 HW4 - Cache Performance Evaluation and the Impact of Loop Orders (due Thursday Oct 1st, 2020)

Introduction

In this assignment, you will implement a matrix multiplication algorithm in *C* or *C++* and evaluate its performance with different loop access orders. Since different loop orders can affect the access orders of matrix elements from the memory, which can eventually impact the cache locality and performance.

Matrix multiplication is the core operation of many linear algebra algorithms, thus efficient implementation of matrix multiplication is crucial for many applications in applied sciences to achieve high performance.

Matrices are 2-dimensional data structures wherein each data element is accessed via two indices. To multiply two matrices, we can simply use 3 nested loops, assuming that matrices A, B, and C are all n-by-n and stored in one-dimensional column-major arrays:

```
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        for (int k = 0; k < n; k++)
            C[i+j*n] += A[i+k*n] * B[k+j*n];
```

In the above code, the *k* loop is the innermost loop, and we say the loop order is *i-j-k*. However, for matrix multiplication, the loops can be arranged in different orders, including *i-k-j*, *j-i-k*, *j-k-i*, *k-i-j*, *k-j-i*, and all of them are correct from software development's perspective. However, different loop orders many results in quite different performance results, which should be concerns for performance-critical software or developers who are aware of computer architectures.

In this assignment, you need to implement and evaluation the performance of matrix multiplication with all these 6 different loop orders for 5000*5000 matrices. Which loop order(s) perform the best, and why? Which loop order(s) perform the worst and why? Which loop order(s) are in between, and why? Your explanation should be specific.

Hint: Use malloc() to allocate memory space for the matrices. You can use clock( ) to measure the time in C/C++, which can be seen at http://www.cplusplus.com/reference/ctime/clock/.

Submission:

You need to submit both a report and your source code file.

The report should include the following sections/contents:

- Title, date, and your name.
- detailed description of experimental setup
- experimental results
- result explanation and analysis
- discussion of any problem(s) you encountered and how did you solve the problem(s).
- conclusions or suggestions.
- Also, please attach the outputs of your experiments.