Project #1

The purpose of this project is to investigate simple but efficient technique for improving performance for matrix multiplication operations. Consider matrix multiplication

\[ C = A \cdot B \]

with all matrixes of size \( N \times N \). Using the so-called “tiling” technique the matrixes \( A \) and \( B \) are partitioned into a smaller chunks so that additions and multiplications are performed on these smaller chunks to obtain elements of matrix \( C \). For example, when matrixes are partitioned into 4 equal submatrixes

\[
\begin{bmatrix}
C & C_{12} \\
C_{21} & C_{22}
\end{bmatrix}
= \begin{bmatrix}
A & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\begin{bmatrix}
B & B_{12} \\
B_{21} & B_{22}
\end{bmatrix}
\]

matrix \( C \) is obtained by calculating

\[
\begin{align*}
C_{11} &= A_{11} \cdot B_{11} + A_{12} \cdot B_{21} \\
C_{12} &= A_{11} \cdot B_{12} + A_{12} \cdot B_{22} \\
C_{21} &= A_{21} \cdot B_{11} + A_{22} \cdot B_{21} \\
C_{22} &= A_{21} \cdot B_{12} + A_{22} \cdot B_{22}
\end{align*}
\]

The particular steps for this project are:

1) Write a program in C language for multiplication of two matrixes of single-precision floating point numbers with linear size \( N = 2^k \) using tiling technique with parameter \( K=2^k \) which defines the linear size of submatrixes. Assume row major order for matrix \( A \) and column major order for matrix \( B \).

2) Measure the execution time of matrix multiplication for values of \( n = 3, \ldots, 10 \) and value of \( k \) from 0 (i.e. tile of size 1) to \( n \) (no tiling). To avoid any uncertainty which might be introduced by operating system and other factors run each experiment several times (say 10) and then take average execution time.

3) Make a figure showing execution time versus tile size for several values of \( n \). For convenience, use log-log graph with x axis values corresponding to tile size \( k \) and y axis corresponding to \( \log[<\text{execution time}>] \) and show all curves (8 total) for different values of \( n \) on the same graph.

4) Discuss the results, in particular make your best effort in explaining the graph in the context of the complexity of the algorithm and memory hierarchy of system that you run experiment on (hint: you may have to consider the number of multiplication and addition operations involved and the size of cache)

5) Submit your report electronically to TA by the time specified on the class website.

In general, you may use any computer to run this experiment but to get the best advice from TA you are encouraged to use Computer Lab facilities (in HFH) and use gcc compiler provided with Linux. Make sure that optimization options are turned off for the compiler.