IN3200/IN4200 Exercise Set 4

Measuring the memory bandwidth

Write a C program that makes use of the following kernel loop

```
for (i=0; i<N, i++)
  A[i] = s*B[i];
```

to measure the realistically achievable memory bandwidth of a CPU.

Strided access of arrays

How will the performance, measured as number of FP operations per second, change if the above loop is modified as follows? And why?

```
for (i=0; i<N, i+=stride)
  A[i] = s*B[i];
```

Matrix-matrix multiplication

If $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{B}$ is an $m \times p$ matrix,

$$\mathbf{A} = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,m-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,m-1} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} b_{0,0} & b_{0,1} & \cdots & b_{0,p-1} \\ b_{1,0} & b_{1,1} & \cdots & b_{1,p-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m-1,0} & b_{m-1,1} & \cdots & b_{m-1,p-1} \end{pmatrix}$$

then the matrix-matrix multiplication $\mathbf{AB}$ will produce another matrix $\mathbf{C}$ of dimension $n \times p$,

$$\mathbf{C} = \mathbf{AB} = \begin{pmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,p-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,p-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n-1,0} & c_{n-1,1} & \cdots & c_{n-1,p-1} \end{pmatrix}$$

where each entry in $\mathbf{C}$ is calculated by

$$c_{i,j} = a_{i,0}b_{0,j} + a_{i,1}b_{1,j} + \ldots + a_{i,m-1}b_{m-1,j} \qquad 0 \le i \le n-1, 0 \le j \le p-1$$

(a) Write a C program to implement the matrix-matrix multiplication. The matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ is each allocated as a two-dimensional array with an underlying one-dimensional contiguous storage of the matrix entries.

(b) Estimate the amount of cache line loads (and stores) from (to) main memory that will be incurred by such a matrix-matrix multiplication.

(c) How can loop-unrolling help to improve the performance?