

Programowanie procesorów graficznych w CUDA. Laboratorium 2.

Mnożenie macierzy

Mnożenie macierzy w pamięci globalnej

Razem z niniejszym plikiem umieszczony jest program `matrixMultiplication.cu`. Program mnoży macierze $A \cdot B$, a wynik zapisuje w C , zrzucanej do pliku. Do programu dołączone są pliki `A.txt` i `B.txt`, które reprezentują macierze 16×16 , oraz pliki `A_18x18` i `B_18x18` z nieco większymi macierzami.

Kernel w programie korzysta z pamięci globalnej:

```
//matrix multiplication kernel
__global__ void MatMulKernel(Matrix A, Matrix B, Matrix C)
{
    // each thread computes one element of C and acumulates results to
    Cvalue
    float Cvalue = 0;
    int row = blockIdx.y * blockDim.y + threadIdx.y;
    int col = blockIdx.x * blockDim.x + threadIdx.x;

    if ((row >= A.height) || (col >= B.width)) {return;}

    for (int e=0; e<A.width; e++)
        Cvalue += A.elements[row*A.width + e] *
        B.elements[e*B.width + col];
    C.elements[row*C.width + col] = Cvalue;
}
```

Macierze zapisane są za pomocą struktur:

```
typedef struct
{
    int width;
    int height;
    float *elements;
} Matrix;
```

Proszę dopisać pętlę mnożenia macierzy w CPU

Proszę dopisać do programu funkcję pomiaru czasu i wykonać mnożenie na CPU i GPU dla obu zestawów danych . Jeśli pomiary czasu są za małe należy powtórzyć mnożenie w pętli.

Proszę rozciągnąć dane $\times 4$ (w dowolny sposób) tak żeby reprezentowały macierze 32×32 , ewentualnie 36×36 i powtórzyć obliczenia.