Heterogenous Program

Sivasankar Arul, IT4Innovations

June/2021

Univerza v Ljubljani

TU WIEN TECHNISCHE UNIVERSITÄT WIEN

CINECA consorzio interuniversitario

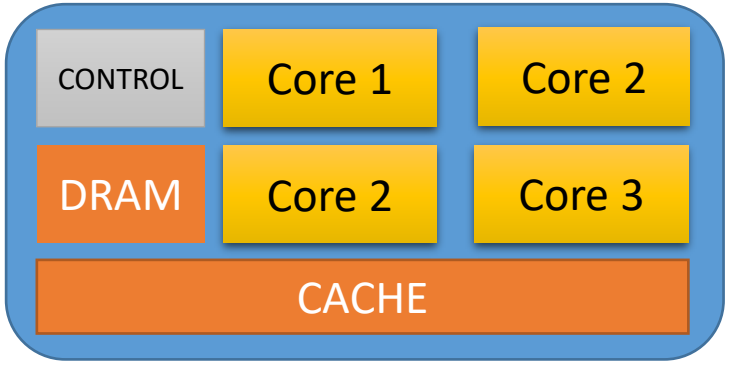
VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Co-funded by the Erasmus+ Programme of the European Union

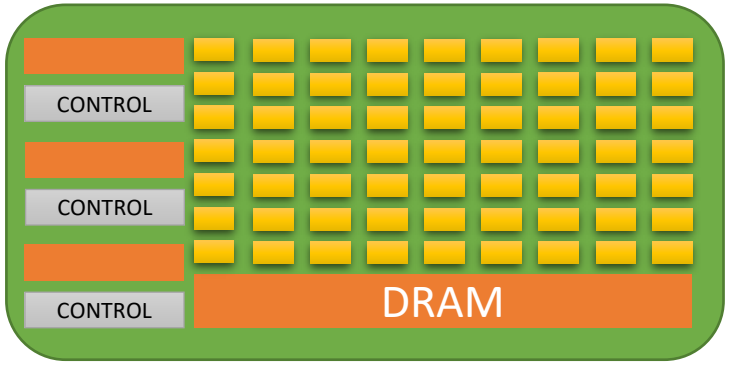
This project has been funded with support from the European Commission.
This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Host, Device

CPU - Host



GPU – Device



Heterogenous Program

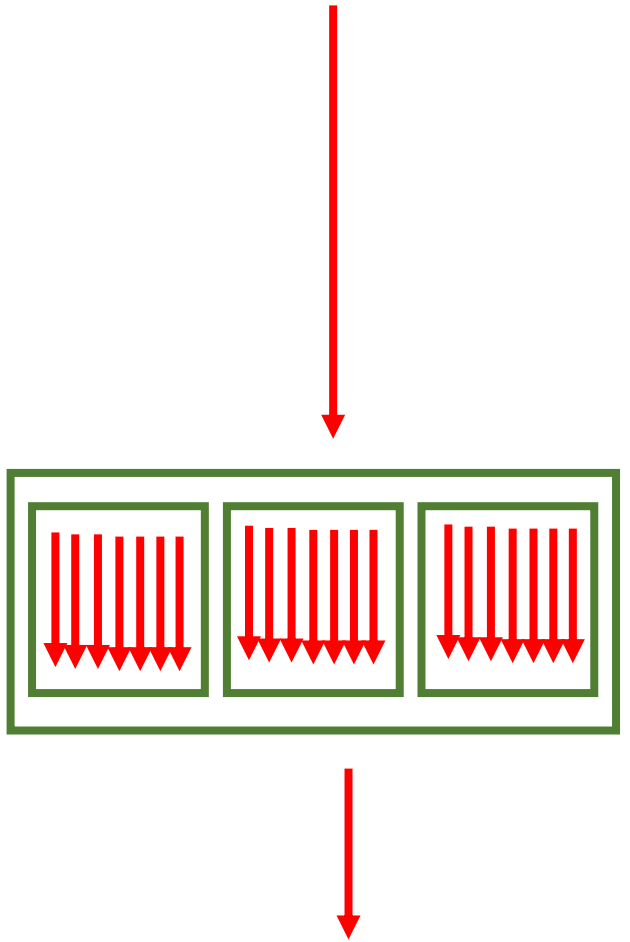

```
int main(){
    float *a, *b, *out, *d_a, *d_b, *d_out;

    // Allocate host memory
    a   = (float*)malloc(sizeof(float) * array_size);
    b   = (float*)malloc(sizeof(float) * array_size);
    out = (float*)malloc(sizeof(float) * array_size);

    // Initialize array
    for(int i = 0; i < array_size; i++){
        a[i] = 1.0f;     b[i] = 2.0f;}

    // Allocate device memory
    cudaMalloc((void**)&d_a, sizeof(float)*array_size);
    cudaMalloc((void**)&d_b, sizeof(float)*array_size);
    cudaMalloc((void**)&d_out, sizeof(float)*array_size);

    // Transfer data from host to device memory
    cudaMemcpy(d_a, a, sizeof(float)*array_size, cudaMemcpyHostToDevice);
    cudaMemcpy(d_b, b, sizeof(float)*array_size, cudaMemcpyHostToDevice);

    int block_size = 256;
    int grid_size  = (array_size + block_size) / block_size;
    // Vector addition
    vector_add<<<grid_size, block_size>>>(d_out, d_a, d_b, array_size);

    // Transfer data from device to host memory
    cudaMemcpy(out, d_out, sizeof(float)*array_size, cudaMemcpyDeviceToHost);

    // Deallocate device memory
    cudaFree(d_a);
    cudaFree(d_b);
    cudaFree(d_out);

    // Deallocate host memory
    free(a);
    free(b);
    free(out);
}
```

Serial Code

Parallel Code

Serial Code

Host – CPU, Device - GPU



Serial code in host

Parallel Code in device

Serial code in host

Heterogenous Program

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

```
int main(){
    float *a, *b, *out;
    float *d_a, *d_b, *d_out;

    a   = (float*)malloc(sizeof(float) * N);
    b   = (float*)malloc(sizeof(float) * N);
    out = (float*)malloc(sizeof(float) * N);

    // Initialize array
    for(int i = 0; i < N; i++){
        a[i] = 1.0f;
        b[i] = 2.0f;
    }
```

Memory Allocation in Host

```
    // Allocate device memore for a
    cudaMalloc((void**)&d_a,sizeof(float)*N);
    cudaMalloc((void**)&d_b,sizeof(float)*N);
    cudaMalloc((void**)&d_out,sizeof(float)*N);
```

Memory Allocation in Device

```
    // Transfer data from host to device memory
    cudaMemcpy(d_a,a, sizeof(float)*N, cudaMemcpyHostToDevice);
    cudaMemcpy(d_b,b, sizeof(float)*N, cudaMemcpyHostToDevice);
```

Data transfer from Host to Device

```
// Main function
int block_size = 256;
int grid_size  = (N+block_size)/block_size;
vector_add<<<grid_size,block_size>>>(d_out, d_a, d_b, N);
```

Computation in Device

```
cudaMemcpy(out, d_out, sizeof(float)*N, cudaMemcpyDeviceToHost);
```

Data transfer from Device to Host

```
    // Deallocate device memory
    cudaFree(d_a);
    cudaFree(d_b);
    cudaFree(d_out);

    // Deallocate host memory
    free(a);
    free(b);
    free(out);
```

Deallocation of Memory

Thank you for your attention!

http://sctrain.eu/

Univerza v Ljubljani

TU WIEN TECHNISCHE UNIVERSITÄT WIEN

CINECA consorzio interuniversitario

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission.
This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.