Empty Book Template

Ryan Hou

Invalid Date

Table of contents

Preface

My notes on ???.

Resources

Some relevant resources:

- CUDA Code Samples
- Intro to CUDA - Josh Holloway

Textbooks:

- Book 1

1 Introduction

1.1 Perspective

1.2 High Level Ideas

2 Hierarchy & Concepts

Grid -> Block -> (Warps) -> Threads

Host Code

- Runs on CPU
- Serial
- Launches CUDA kernels

Device Code:

- Runs on GPU
- Parallel
-

3 Syntax

Kernel Launch:

```
// Specify block and grid dimensions
dim3 grid_size(x, y, z);
dim3 block_size(x, y, z);

// Launch kernel
kernelName<<< grid_size, block_size >>> (...);
```

4 Program Flow

- Host code
  - do tasks on the host
  - prepare for kernel launch
  - Allocate memory on the device
  - Copy data from host to device
  - Launch the kernel
  - Copy data from the device to the host

4.1 Example

```
int main ( void ) {
    cudaMalloc(...);

    cudaMemcpy(...);


    kernel_0<<<grid_size0, blk_size0>>>(...);

    cudaMemcpy(...);

    return 0;
}
```

```
int main( void ) {

    // Declare variables
    int *h_c, *d_c;

    // Allocate memory on the device
    cudaMalloc( (void**)&d_c, sizeof(int) );

    // Copy data to the device
```

```
    cudaMemcpy(d_c, h_c, sizeof(int), cudaMemcpyHostToDevice );

    // Configuration Parameters
    dim3 grid_size(1);
    dim3 block_size(1);

    // Launch the Kernel
    kernel<<<grid_size, block_size>>>(...);

    // Copy data back to host
    cudaMemcpy( h_c, d_c, sizeof(int), cudaMemcpyDeviceToHost );

    // De-allocate memory
    cudaFree( d_c );
    free( h_c );

    return 0;
}
```

4.2 Convention

Variables that live on host start with `h_`

Variables that live on device start with `d_`

5 For Loop Example

```
// Kernel Definition
__global__ void increment_gpu(int *a, int N)
{
    int i = threadIdx.x;
    if (i < N)
        a[i] = a[i] + 1;
}

int main( void )
{
    int h_a[N] = // ...

    // Allocate arrays in Device memory
    int* d_a;
    cudaMalloc((void**)&d_a, N * sizeof(int));

    // Copy memory from Host to Device
    cudaMemcpy(d_a, h_a, N * sizeof(int), cudaMemcpyHostToDevice);

    // Block and Grid dimensions
    dim3 grid_size(1);
    dim3 block_size(N);

    // Launch Kernel
    increment_gpu<<<grid_size, block_size>>>(d_a, N);

    // Copy data back and cleanup (not shown in the image)

    return 0;
}
```

6 Summary

In summary...

References