



DEEP
LEARNING
INSTITUTE

MULTI-GPU PROGRAMMING FOR CUDA C++

An abstract geometric pattern consisting of numerous small dots connected by thin lines, forming a complex, interconnected network. The pattern is primarily located on the left side of the image, with some lines extending towards the center. The background is a solid, vibrant green color.

INTRODUCTION

INTRODUCTION

Main Objectives

Concurrency Strategies

Workshop Structure



MAIN OBJECTIVES

MAIN OBJECTIVES

Increase performance for Single-Node CUDA C/C++ applications by exploiting, and then combining, 2 concurrency strategies offered to CUDA programmers.

MAIN OBJECTIVES

Increase performance for Single-Node CUDA C/C++ applications by exploiting, and then combining, 2 concurrency strategies offered to CUDA programmers:

- 1) Overlapping memory transfers to and from the GPU with computations on the GPU

MAIN OBJECTIVES

Increase performance for Single-Node CUDA C/C++ applications by exploiting, and then combining, 2 concurrency strategies offered to CUDA programmers:

- 1) Overlapping memory transfers to and from the GPU with computations on the GPU
- 2) Performing computations concurrently on more than one GPU

An abstract geometric pattern consisting of numerous small dots connected by thin lines, forming a complex, interconnected network. This pattern is located on the left side of the image, set against a solid green background.

CONCURRENCY STRATEGIES

GPU programming is usually a 3-step
process

1. Transfer data to GPU device(s)



A diagram on a black background. A red rectangular box with a black border is positioned in the lower-left area. The word "copy" is written in black text inside the box. To the left of the box is a vertical blue line. Below the box is a horizontal blue line that ends in an arrowhead pointing to the right.

copy

2. Perform computation on GPU device(s)



A horizontal timeline is shown with a vertical line at the start and an arrow pointing to the right. Two rectangular blocks are placed on the timeline. The first block is red and labeled 'copy'. The second block is green and labeled 'compute', starting after the 'copy' block ends.

copy

compute

3. Transfer data back to the host

copy

compute

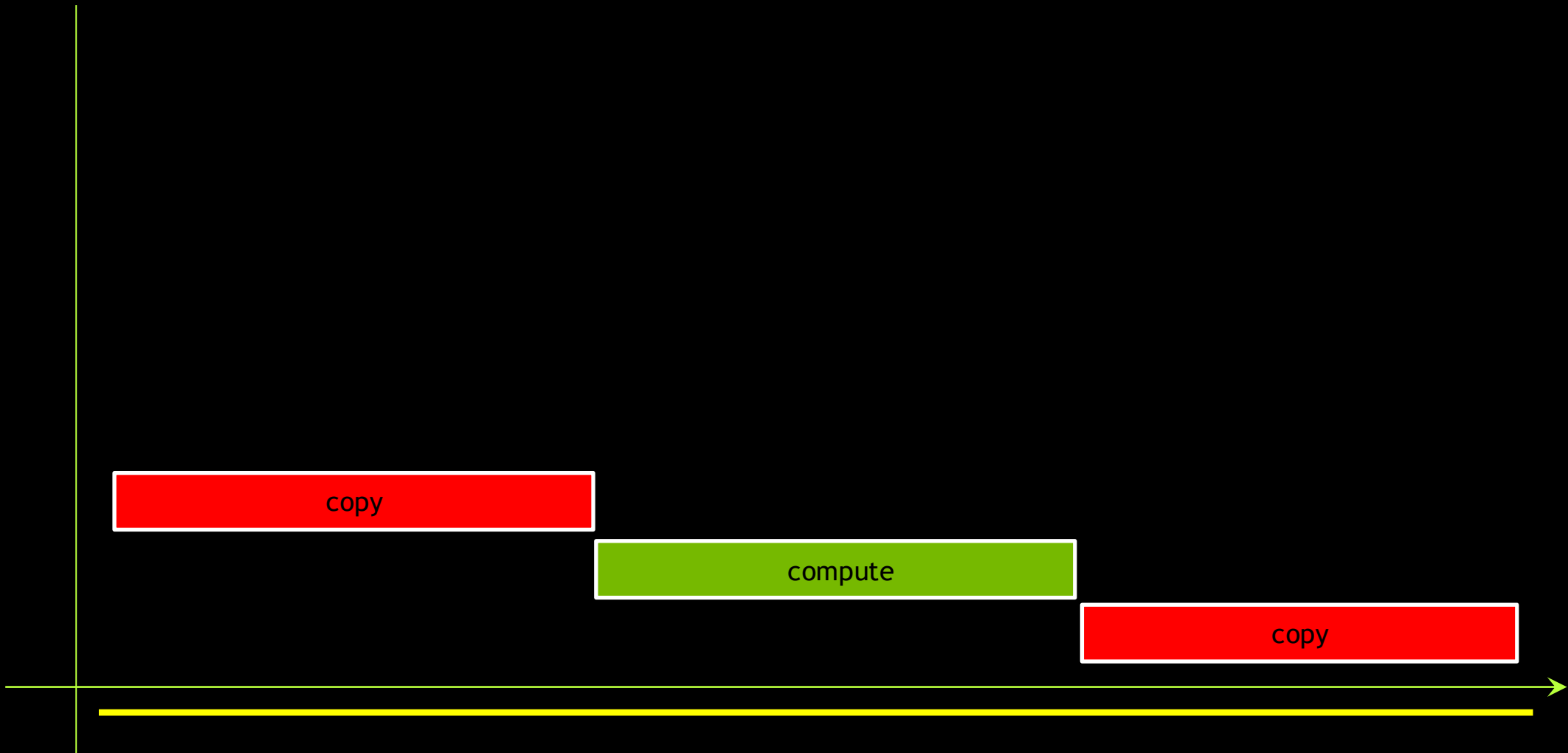
copy

Total runtime is the sum

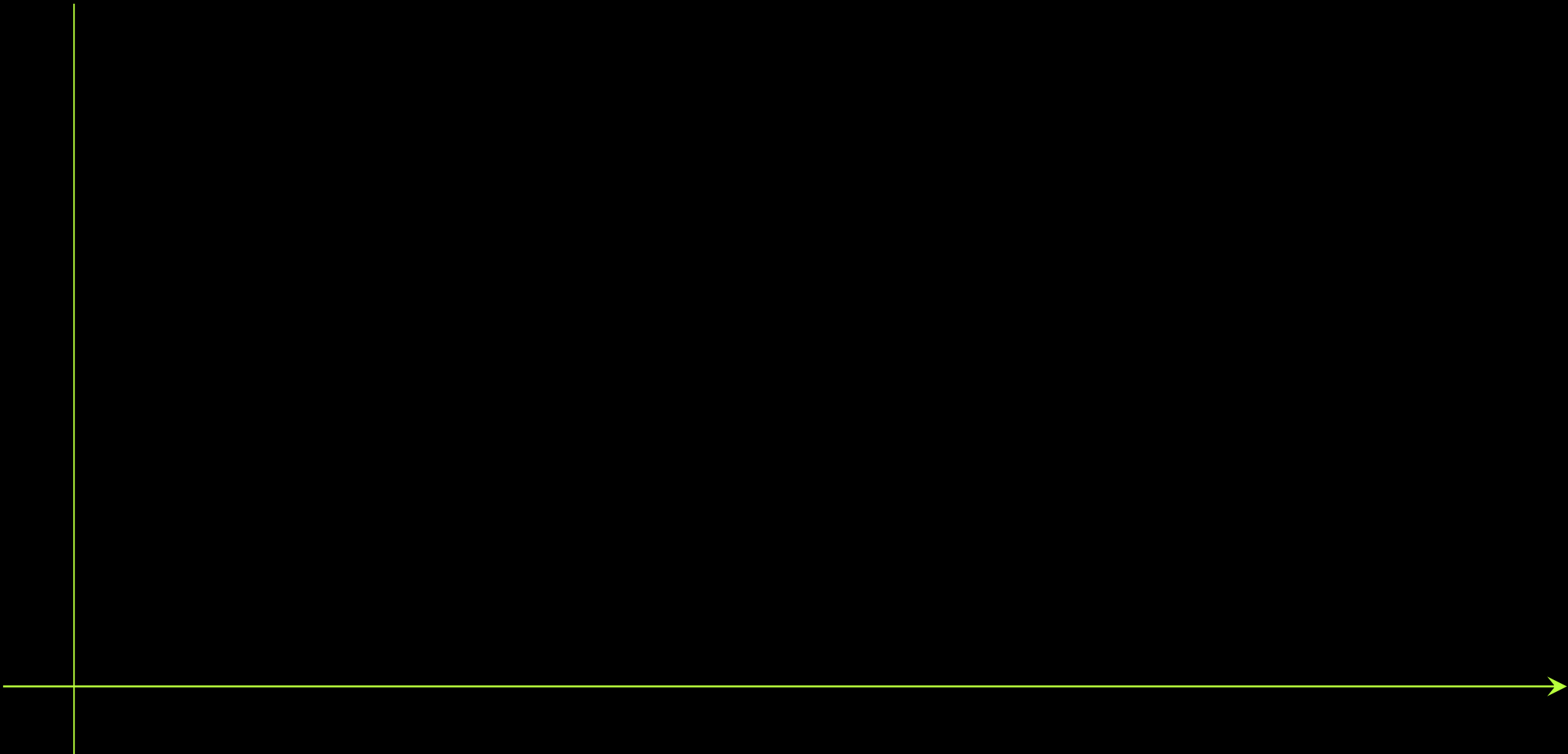
copy

compute

copy



If we can overlap memory transfer and
compute...



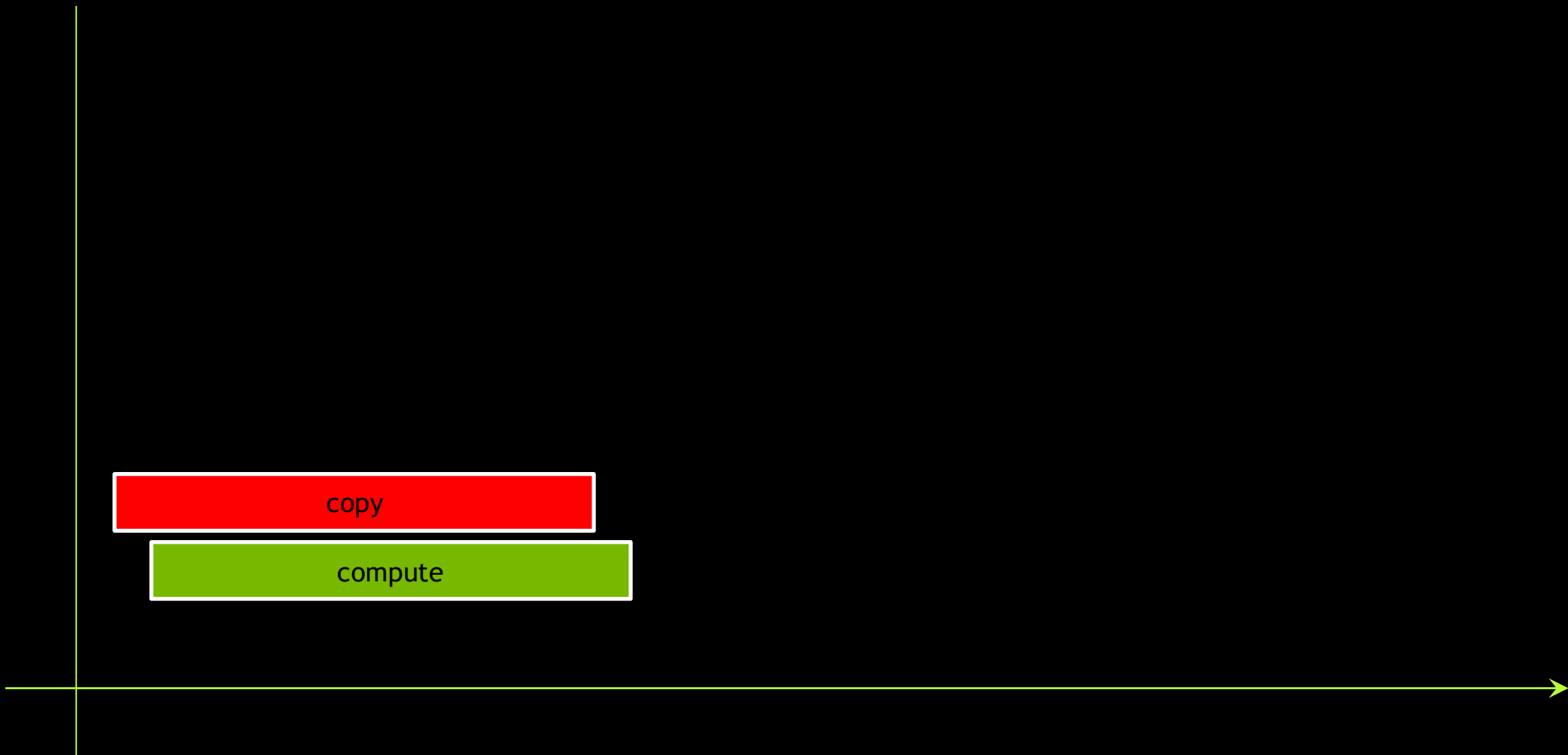
If we can overlap memory transfer and
compute...



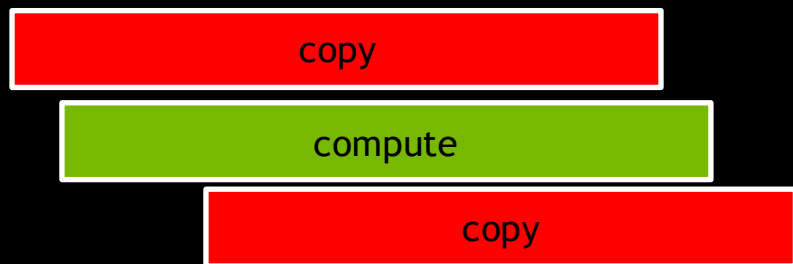
A diagram illustrating memory transfer and computation. It features a black background with a white coordinate system consisting of a vertical y-axis and a horizontal x-axis. The x-axis has an arrow pointing to the right. A red rectangular bar with a black border is positioned in the lower-left area of the coordinate system. The word "copy" is written in black text inside the red bar.

copy

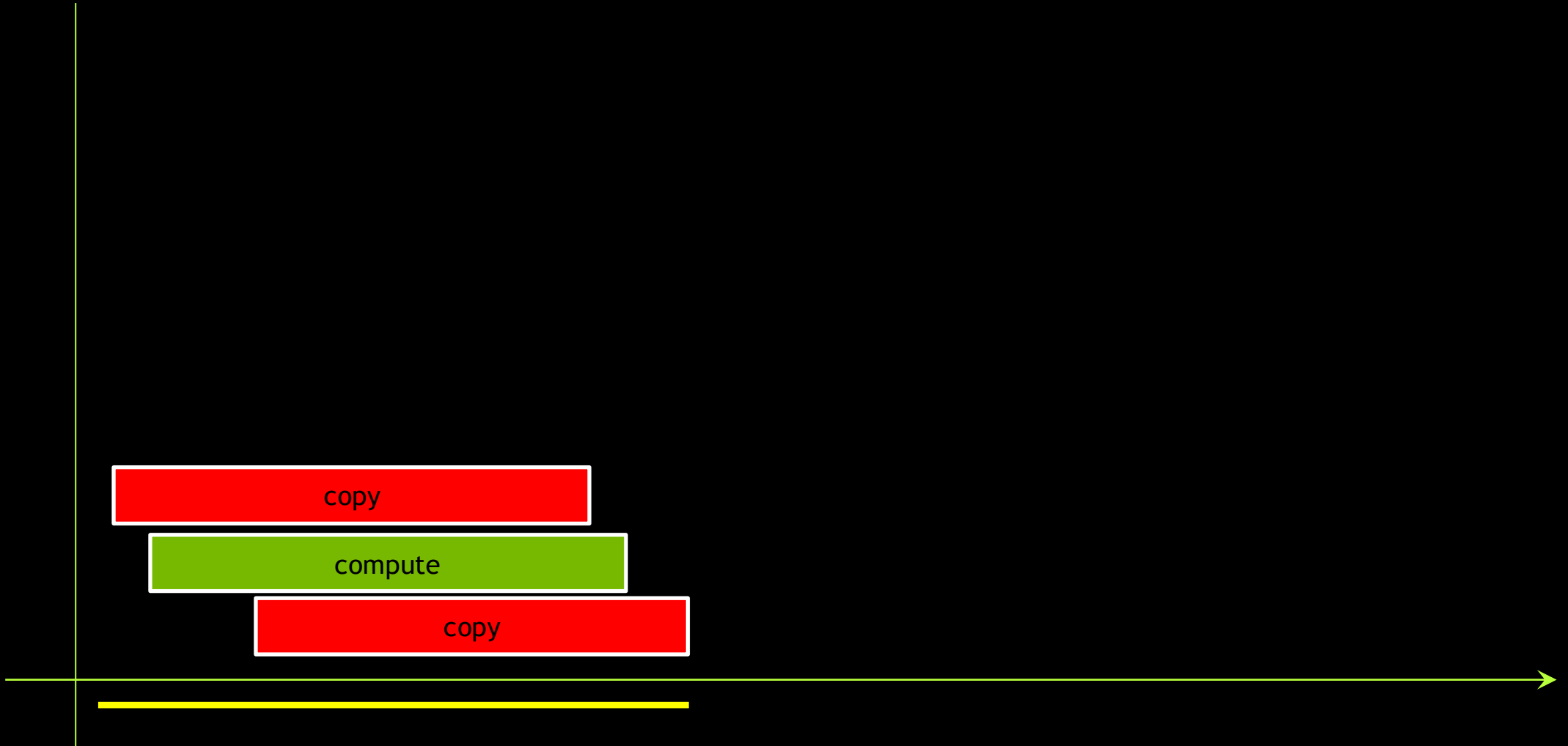
If we can overlap memory transfer and compute...



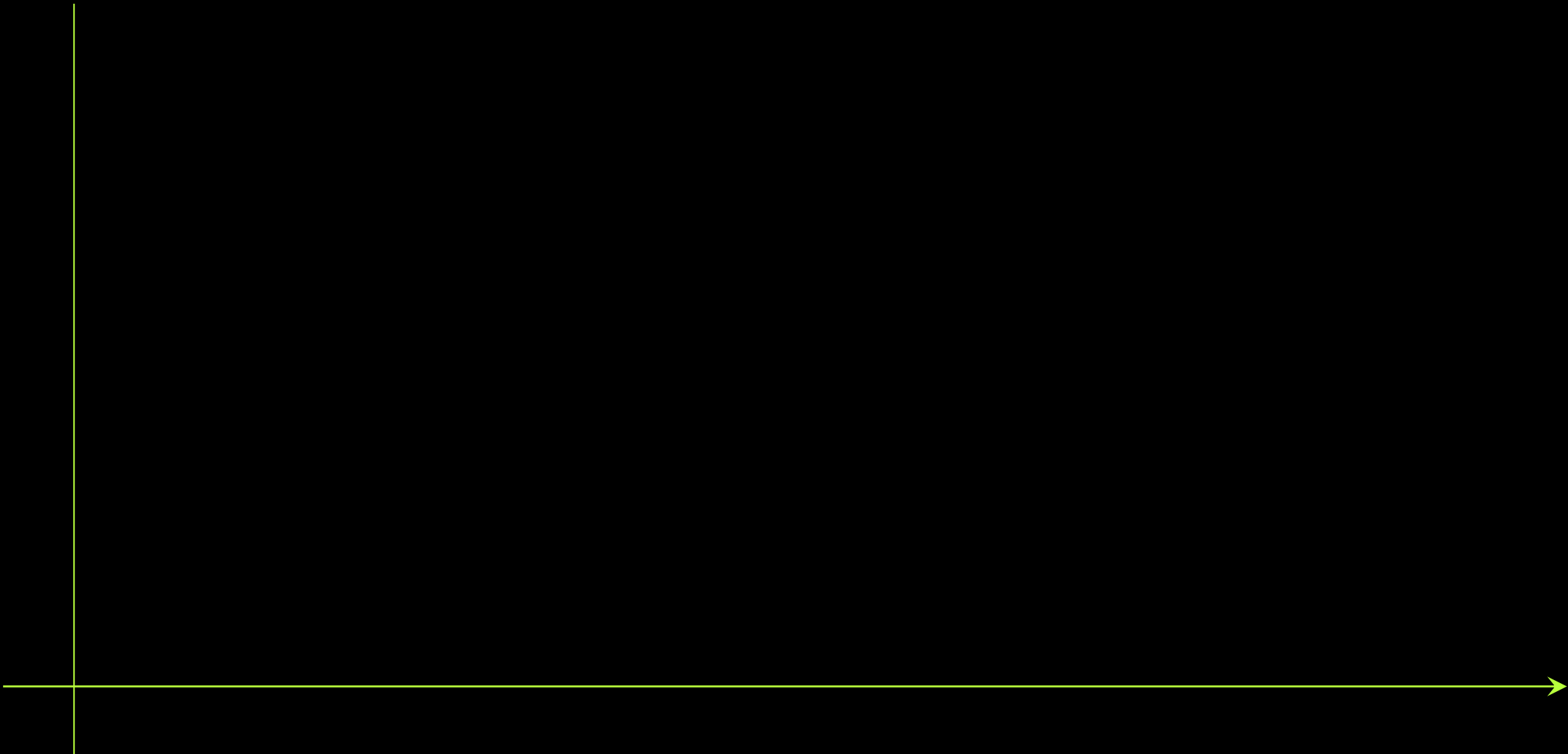
If we can overlap memory transfer and compute...



...total application time will be less



If we can overlap computation on
multiple devices...



If we can overlap computation on
multiple devices...



A diagram illustrating a task on a coordinate system. The system has a vertical y-axis and a horizontal x-axis, both represented by thin black lines. The x-axis ends with an arrow pointing to the right. A solid red horizontal bar is positioned in the upper-left area of the coordinate system. The word "copy" is written in black text inside the red bar.

copy

If we can overlap computation on multiple devices...



A Gantt chart diagram illustrating task execution. A vertical yellow line on the left represents the start of time. A horizontal yellow line at the bottom represents the timeline, ending in an arrow. A red bar labeled 'copy' starts at the vertical line and extends to the right. To the right of the 'copy' bar, three green bars labeled 'compute' are stacked vertically, each starting at the end of the 'copy' bar and extending to the right.

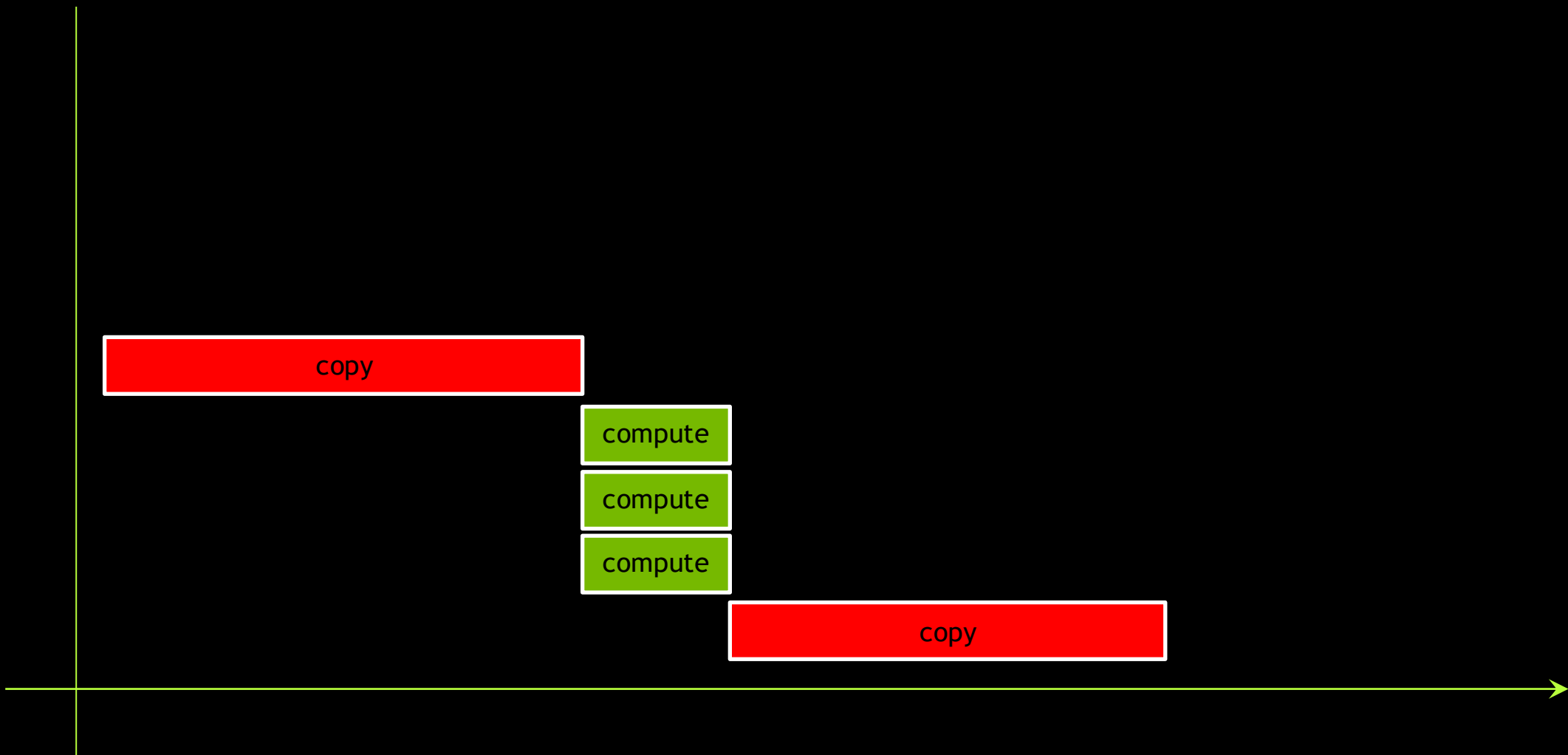
copy

compute

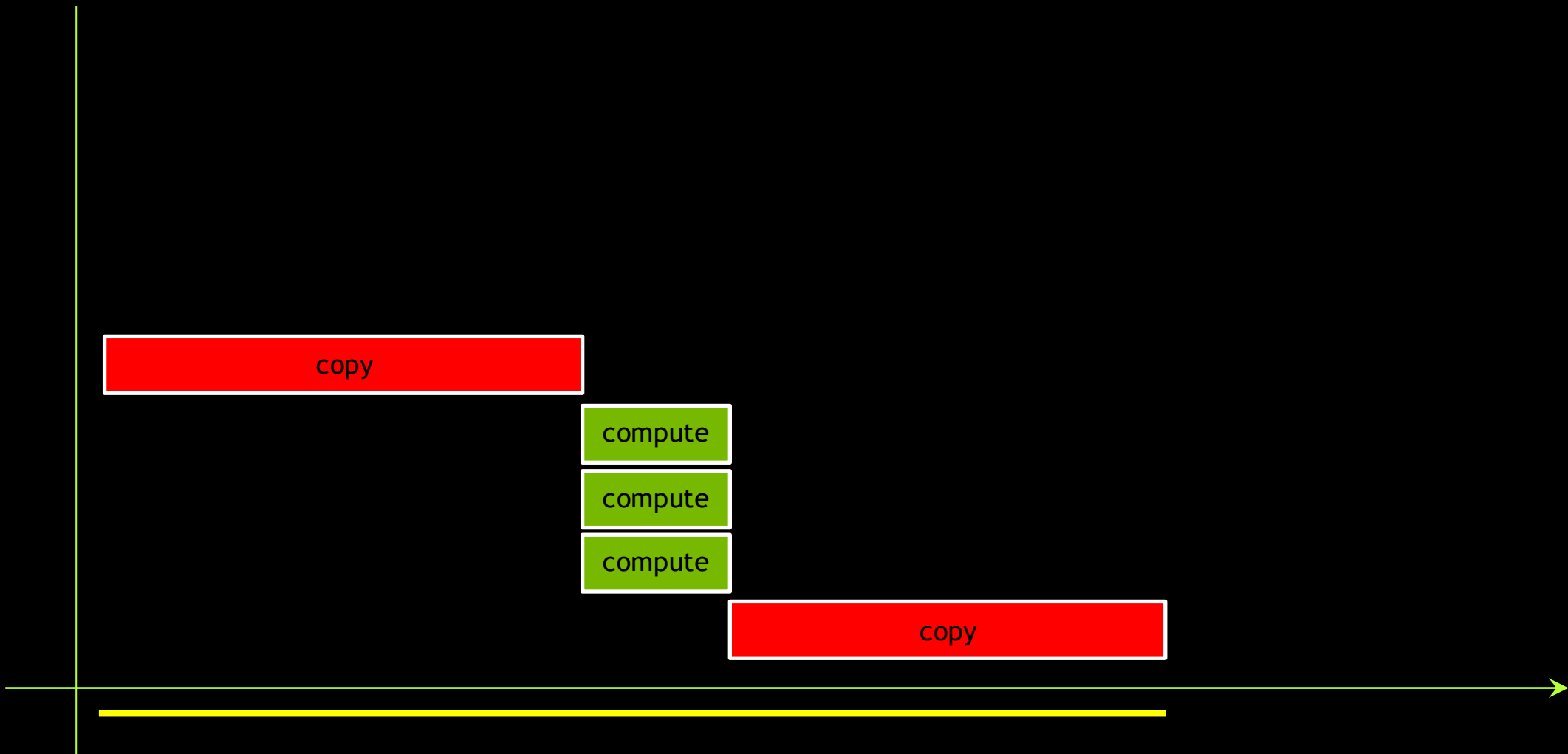
compute

compute

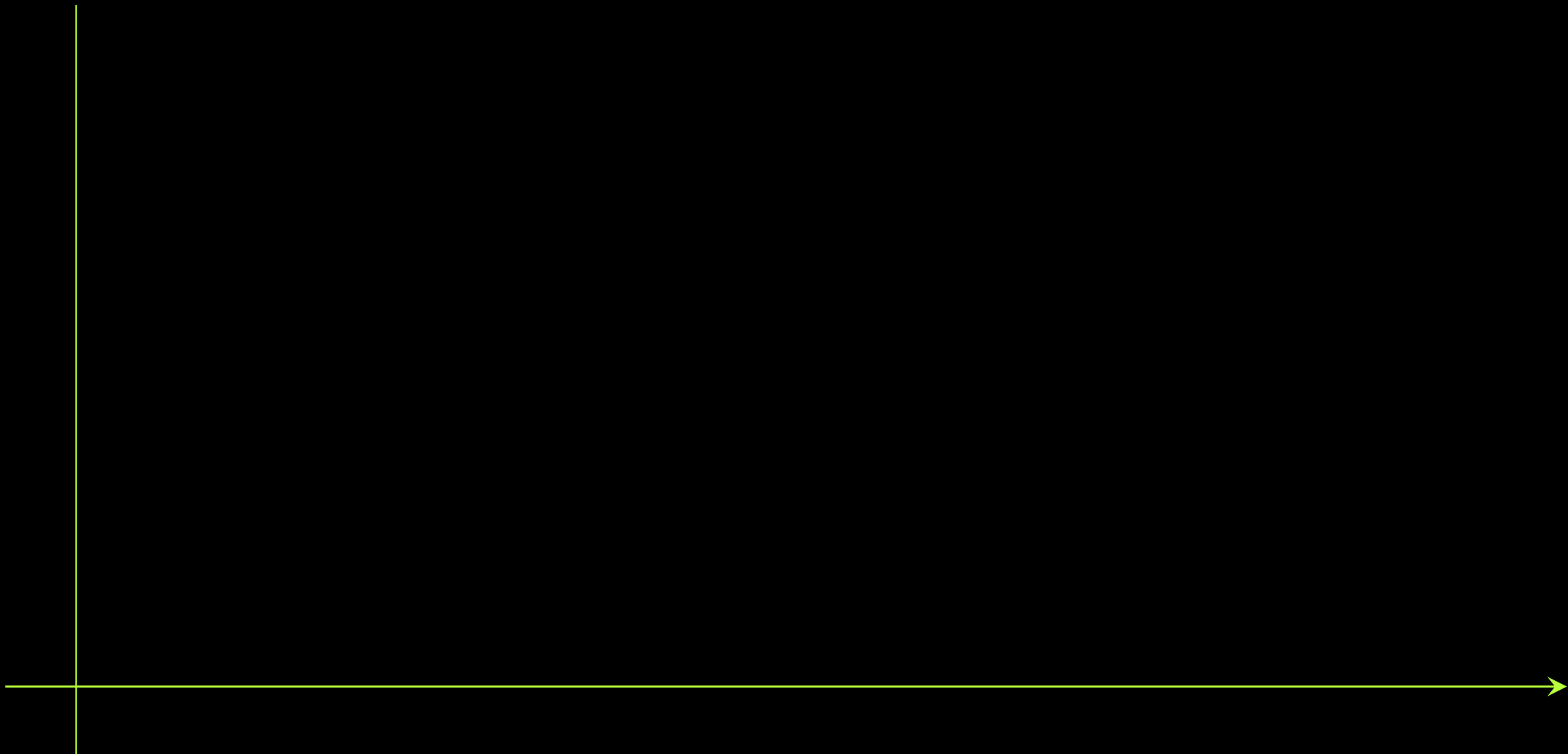
If we can overlap computation on multiple devices...



...total application time will also be less



Combining the 2 strategies...



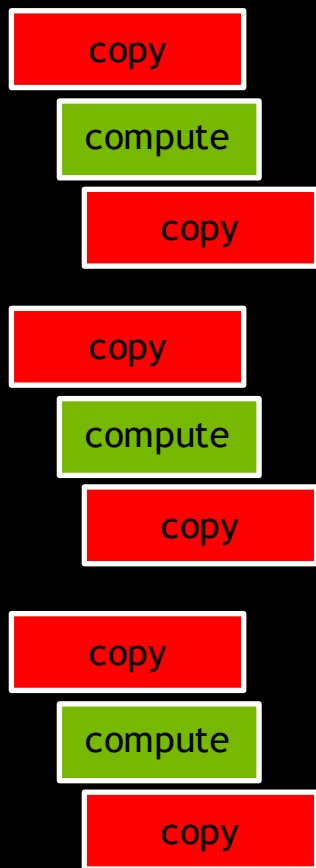
...overlapping compute on multiple
devices

compute

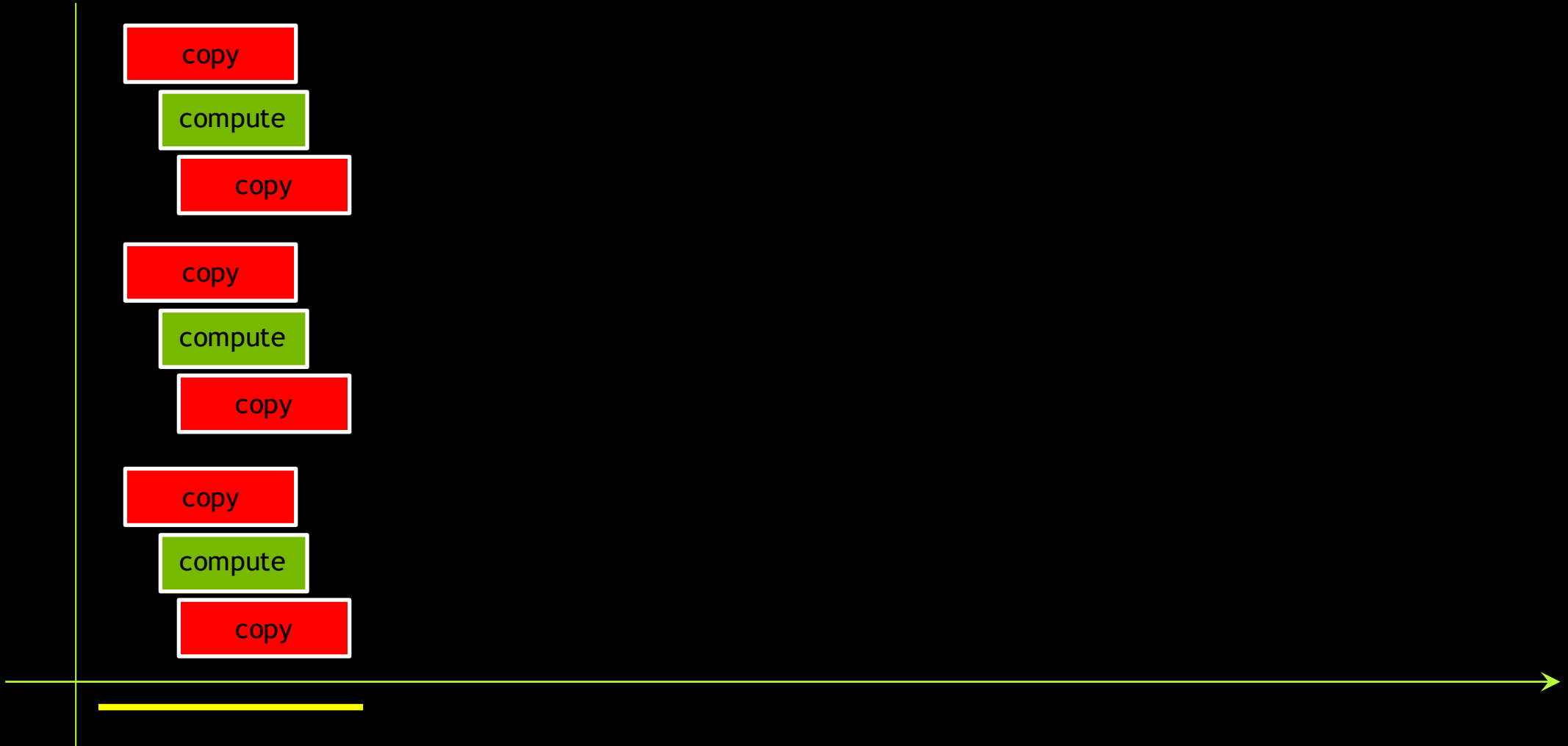
compute

compute

...and copy with each device's compute



...total application time will be even less



MAIN OBJECTIVES

Increase performance for Single-Node CUDA C/C++ applications by exploiting, and then combining, 2 concurrency strategies offered to CUDA programmers:

- 1) Overlapping memory transfers to and from the GPU with computations on the GPU
- 2) Performing computations concurrently on more than one GPU

An abstract geometric pattern consisting of numerous small dots connected by thin lines, forming a complex, interconnected network. The pattern is primarily located on the left side of the image, with some lines extending towards the center. The background is a solid, vibrant green color.

WORKSHOP STRUCTURE

WORKSHOP STRUCTURE

Introduction (this section)

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

Memory Copies in Non-Default Streams

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

Memory Copies in Non-Default Streams

Considerations for Copy/Compute Overlap

WORKSHOP STRUCTURE

Introduction (this section)

Exercise: Copy/Compute Overlap

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

Memory Copies in Non-Default Streams

Considerations for Copy/Compute Overlap

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

Memory Copies in Non-Default Streams

Considerations for Copy/Compute Overlap

Exercise: Copy/Compute Overlap

Multiple GPUs

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

Memory Copies in Non-Default Streams

Considerations for Copy/Compute Overlap

Exercise: Copy/Compute Overlap

Multiple GPUs

Considerations for Multiple GPUs

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

Memory Copies in Non-Default Streams

Considerations for Copy/Compute Overlap

Exercise: Copy/Compute Overlap

Multiple GPUs

Considerations for Multiple GPUs

Exercise: Multiple GPUs

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

Memory Copies in Non-Default Streams

Considerations for Copy/Compute Overlap

Exercise: Copy/Compute Overlap

Multiple GPUs

Considerations for Multiple GPUs

Exercise: Multiple GPUs

Exercise: Multiple GPUs with Copy/Compute Overlap

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

Memory Copies in Non-Default Streams

Considerations for Copy/Compute Overlap

Exercise: Copy/Compute Overlap

Multiple GPUs

Considerations for Multiple GPUs

Exercise: Multiple GPUs

Exercise: Multiple GPUs with Copy/Compute Overlap

Course Survey

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

Memory Copies in Non-Default Streams

Considerations for Copy/Compute Overlap

Exercise: Copy/Compute Overlap

Multiple GPUs

Considerations for Multiple GPUs

Exercise: Multiple GPUs

Exercise: Multiple GPUs with Copy/Compute Overlap

Course Survey

Course Assessment

WORKSHOP STRUCTURE

Introduction (this section)

Using JupyterLab

Cipher Application Overview

Nsight Systems Setup

CUDA Streams

Kernel Launches in Non-Default Streams

Memory Copies in Non-Default Streams

Considerations for Copy/Compute Overlap

Exercise: Copy/Compute Overlap

Multiple GPUs

Considerations for Multiple GPUs

Exercise: Multiple GPUs

Exercise: Multiple GPUs with Copy/Compute Overlap

Course Survey

Course Assessment

Next Steps



DEEP
LEARNING
INSTITUTE

www.nvidia.com/dli

