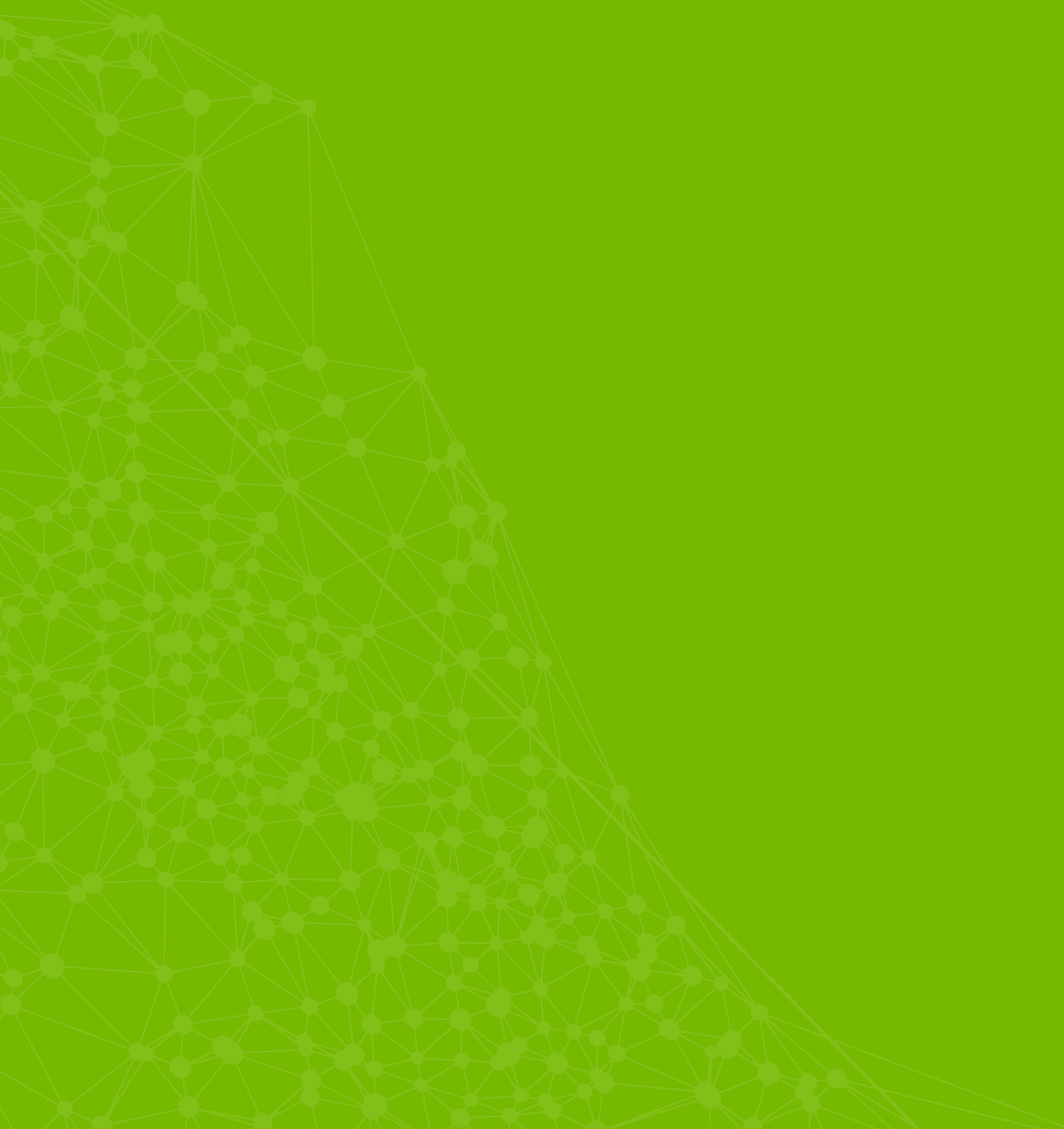




DEEP
LEARNING
INSTITUTE

MULTI-GPU PROGRAMMING FOR CUDA C++

Dr. Momme Allalen | LRZ | 30.11.2021



INTRODUCTION

INTRODUCTION

Main Objectives

Concurrency Strategies

Workshop Structure



MAIN OBJECTIVES



DEEP
LEARNING
INSTITUTE



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

MULTI-GPU PROGRAMMING FOR CUDA C/C++

LINK: <https://courses.nvidia.com/dli-event>

EVENT CODE:

WIFI NAME:

WIFI PASSWORD:

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



CONCURRENCY STRATEGIES

GPU programming is usually a 3-step
process

1. Transfer data to GPU device(s)



copy

The diagram features a black background with a white coordinate system. A vertical white line is on the left, and a horizontal white line with an arrow pointing right is at the bottom. A red horizontal bar with a white border is positioned above the horizontal axis and to the right of the vertical axis. The word 'copy' is written in black text inside the red bar.

2. Perform computation on GPU device(s)



A Gantt chart illustrating the execution of two tasks. The horizontal axis represents time, and the vertical axis represents the processor. The first task, labeled 'copy', is represented by a red bar. The second task, labeled 'compute', is represented by a green bar that starts after the 'copy' task has finished. Both bars have a thin white border.

copy

compute

3. Transfer data back to the host



copy

compute

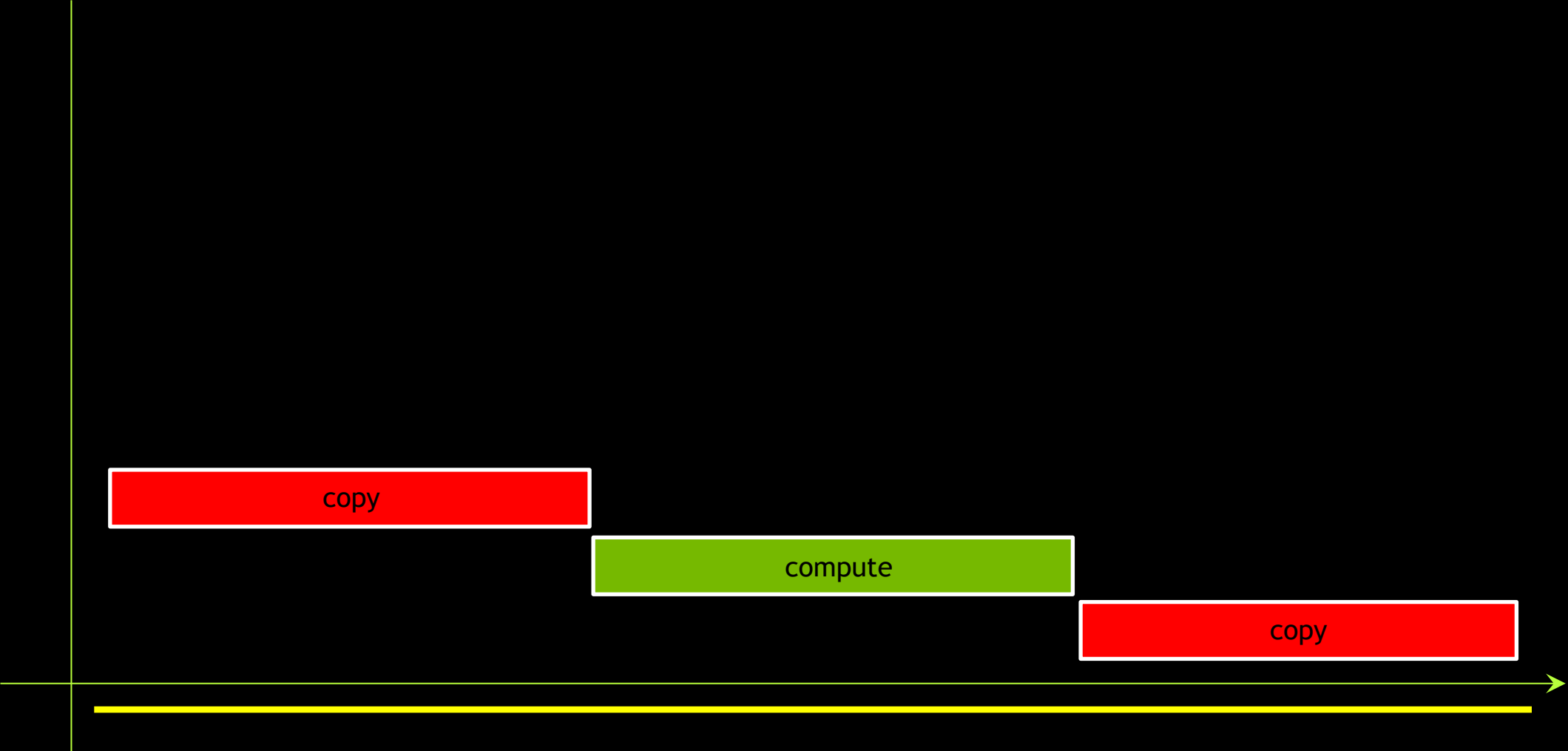
copy

Total runtime is the sum

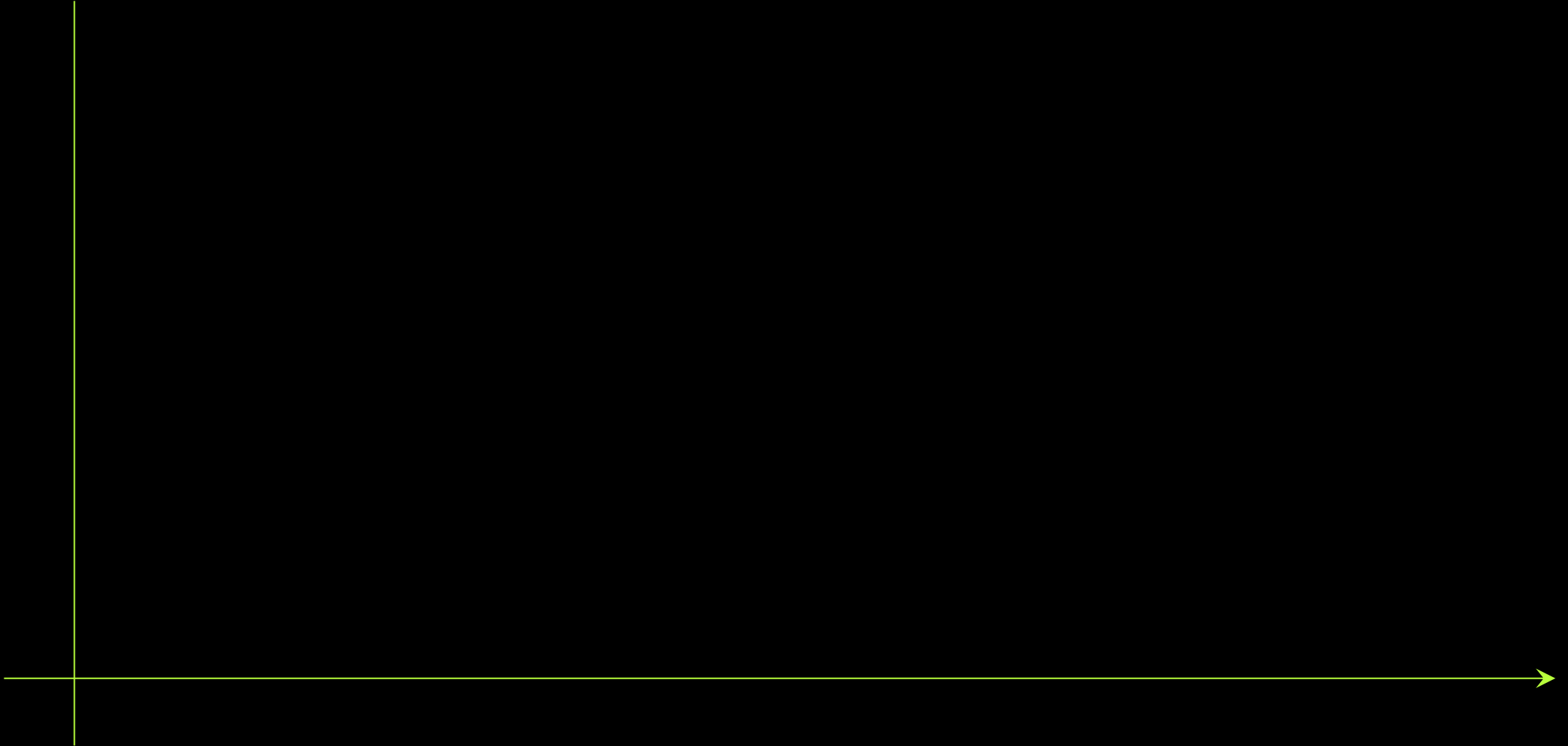
copy

compute

copy



If we can overlap memory transfer and compute...



If we can overlap memory transfer and compute...

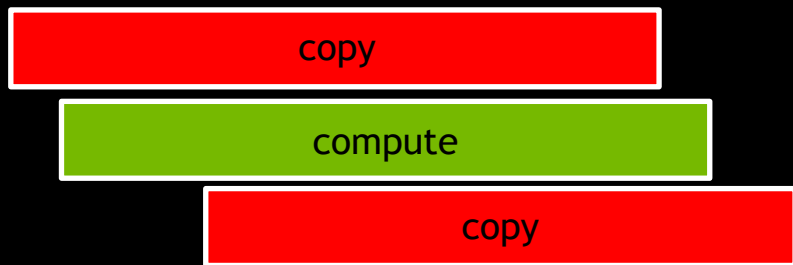


copy

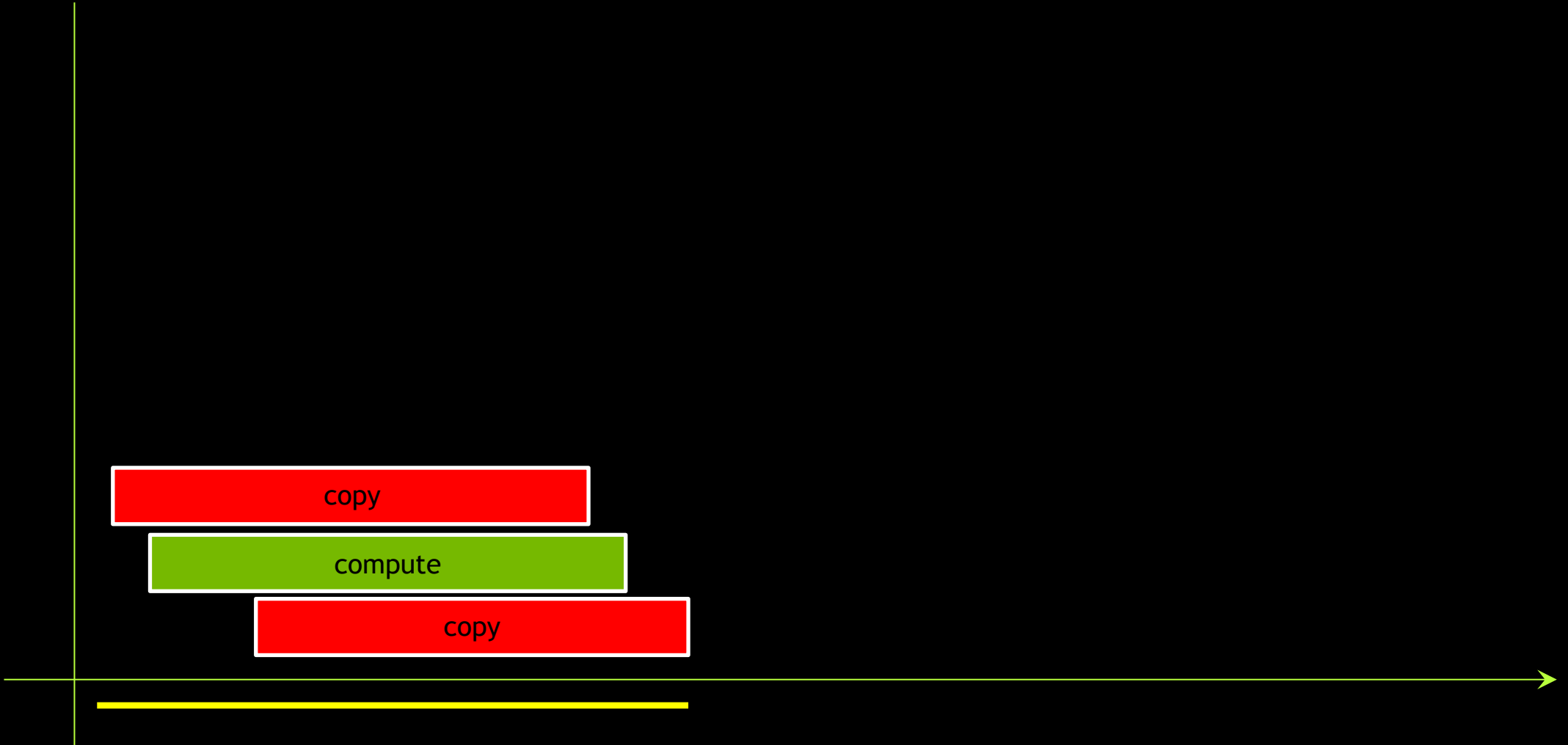
If we can overlap memory transfer and compute...



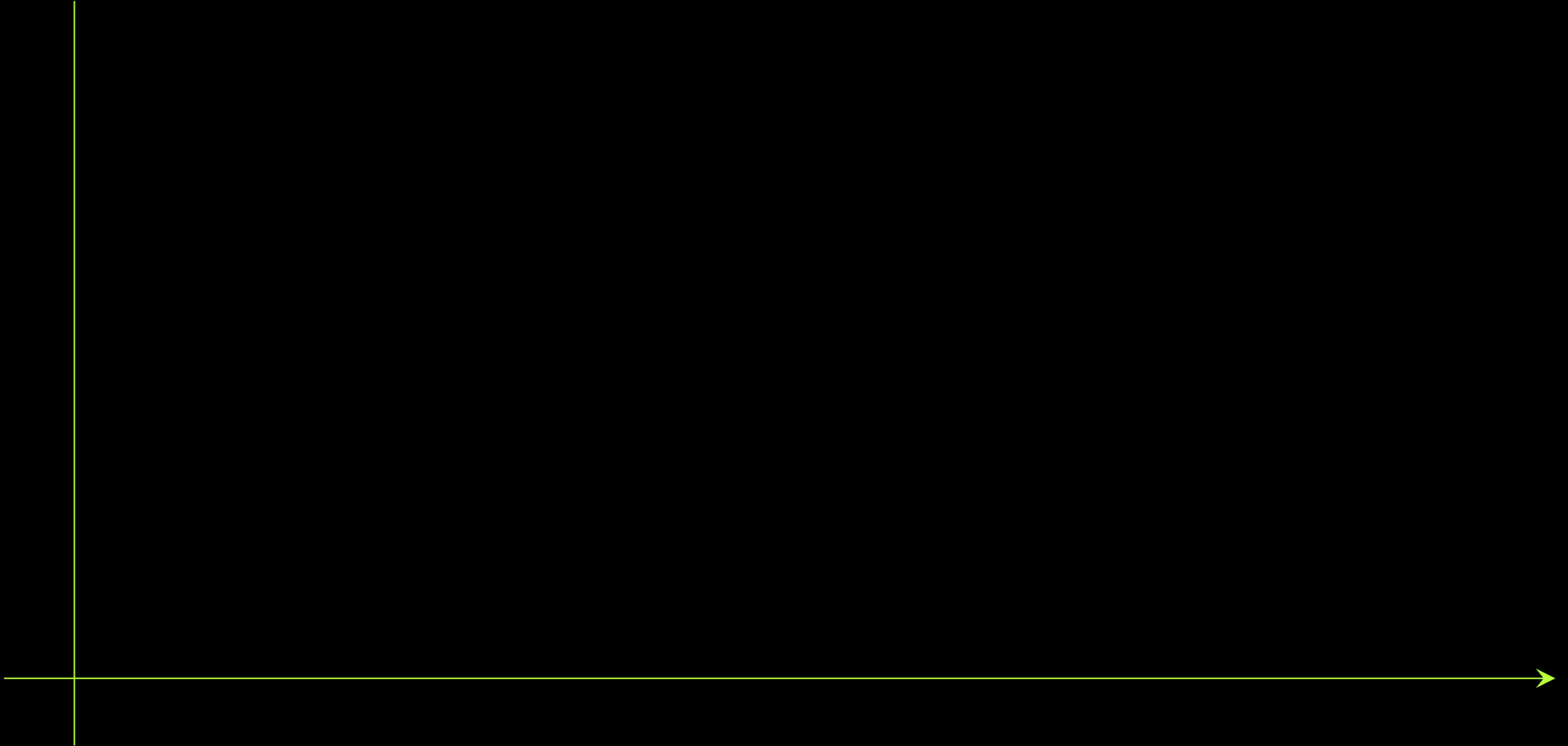
If we can overlap memory transfer and compute...



...total application time will be less



If we can overlap computation on multiple devices...

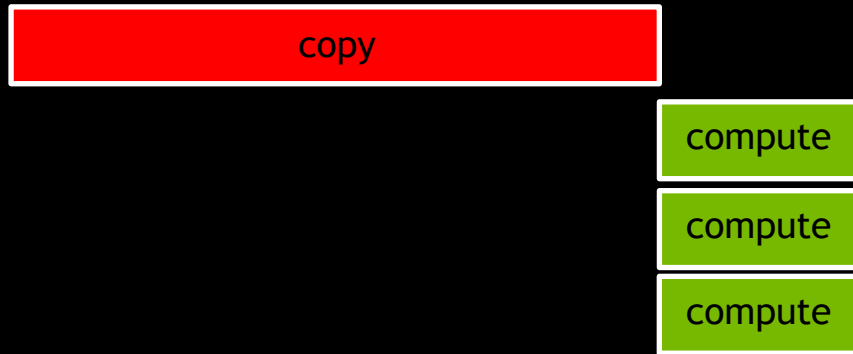


If we can overlap computation on multiple devices...

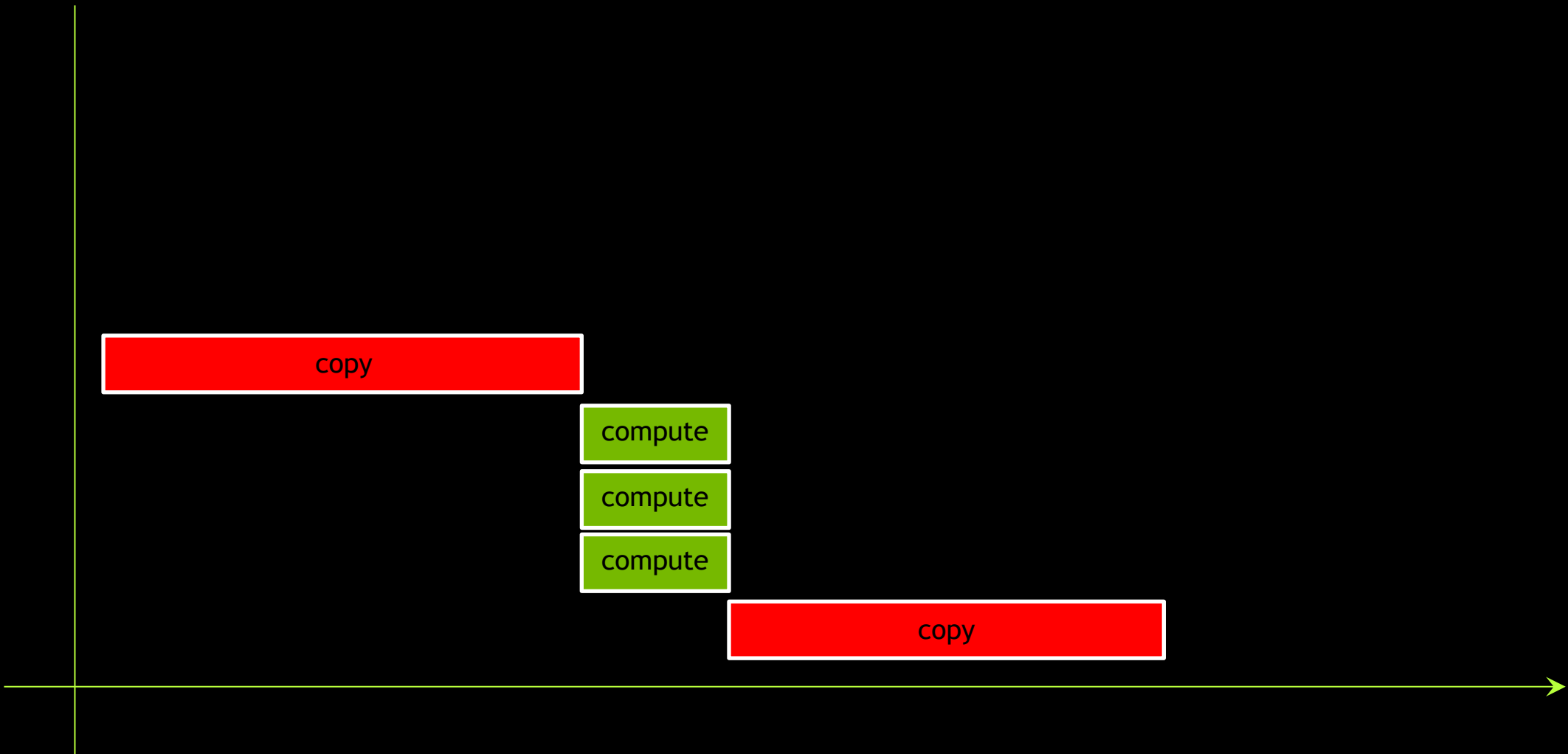


copy

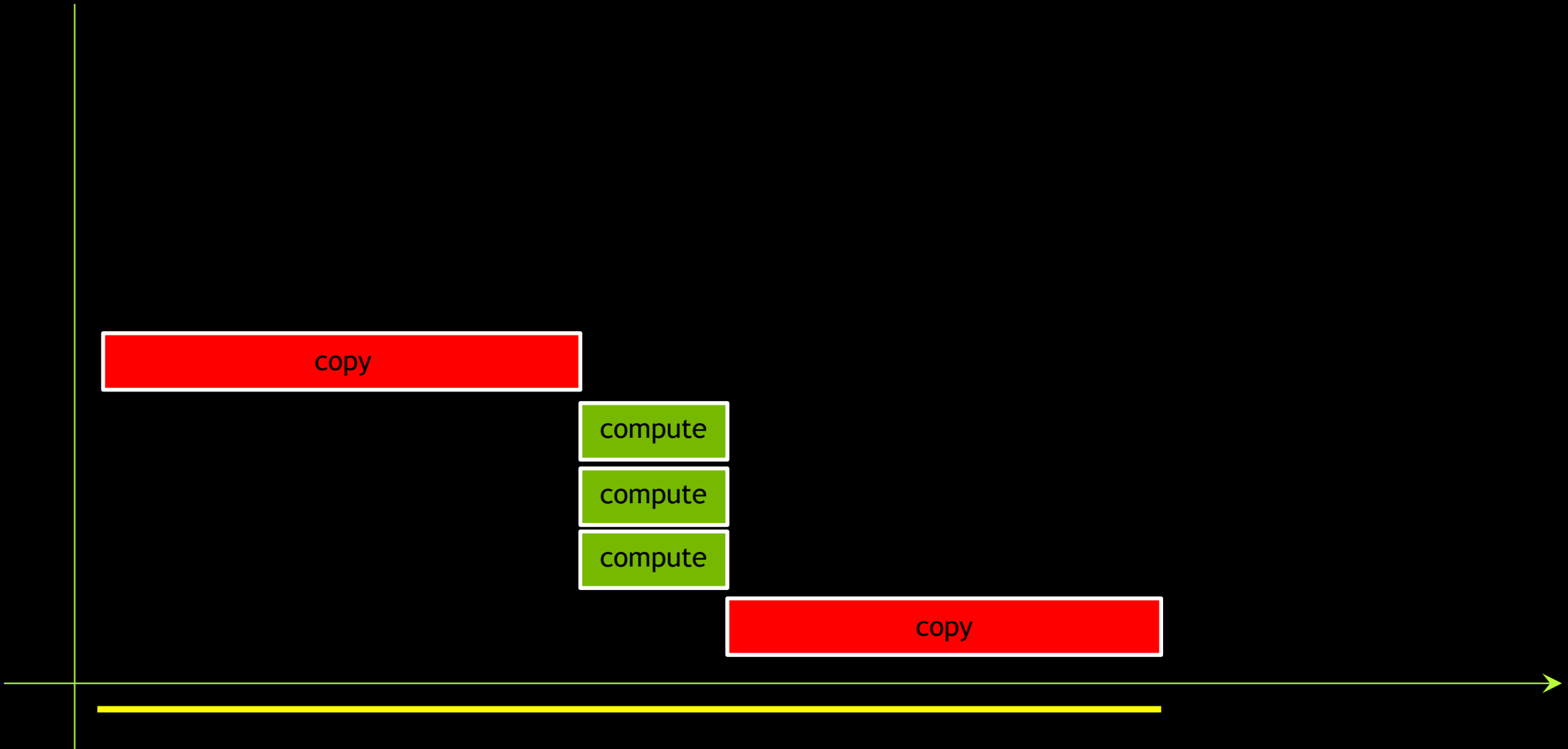
If we can overlap computation on multiple devices...



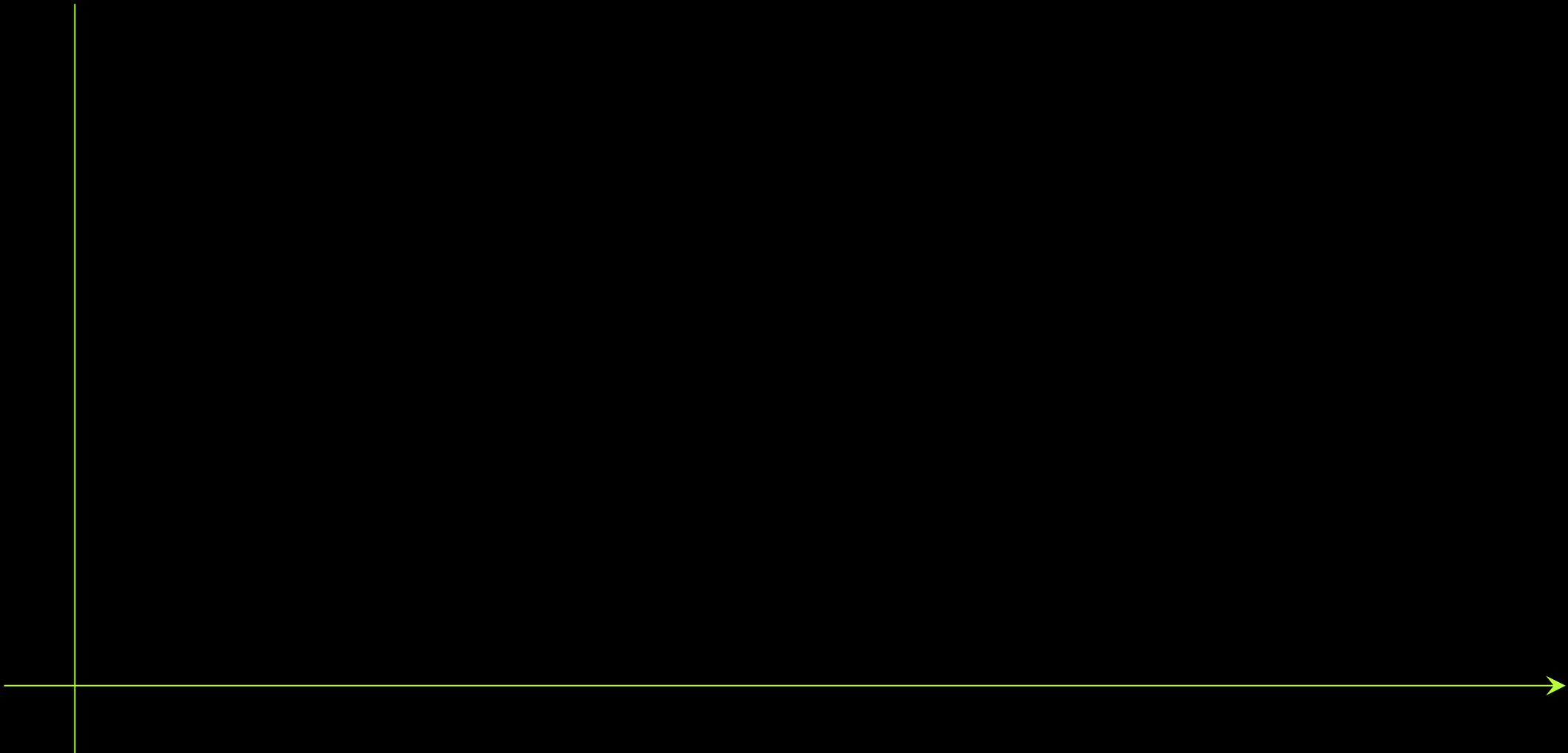
If we can overlap computation on multiple devices...



...total application time will also be less



Combining the 2 strategies...

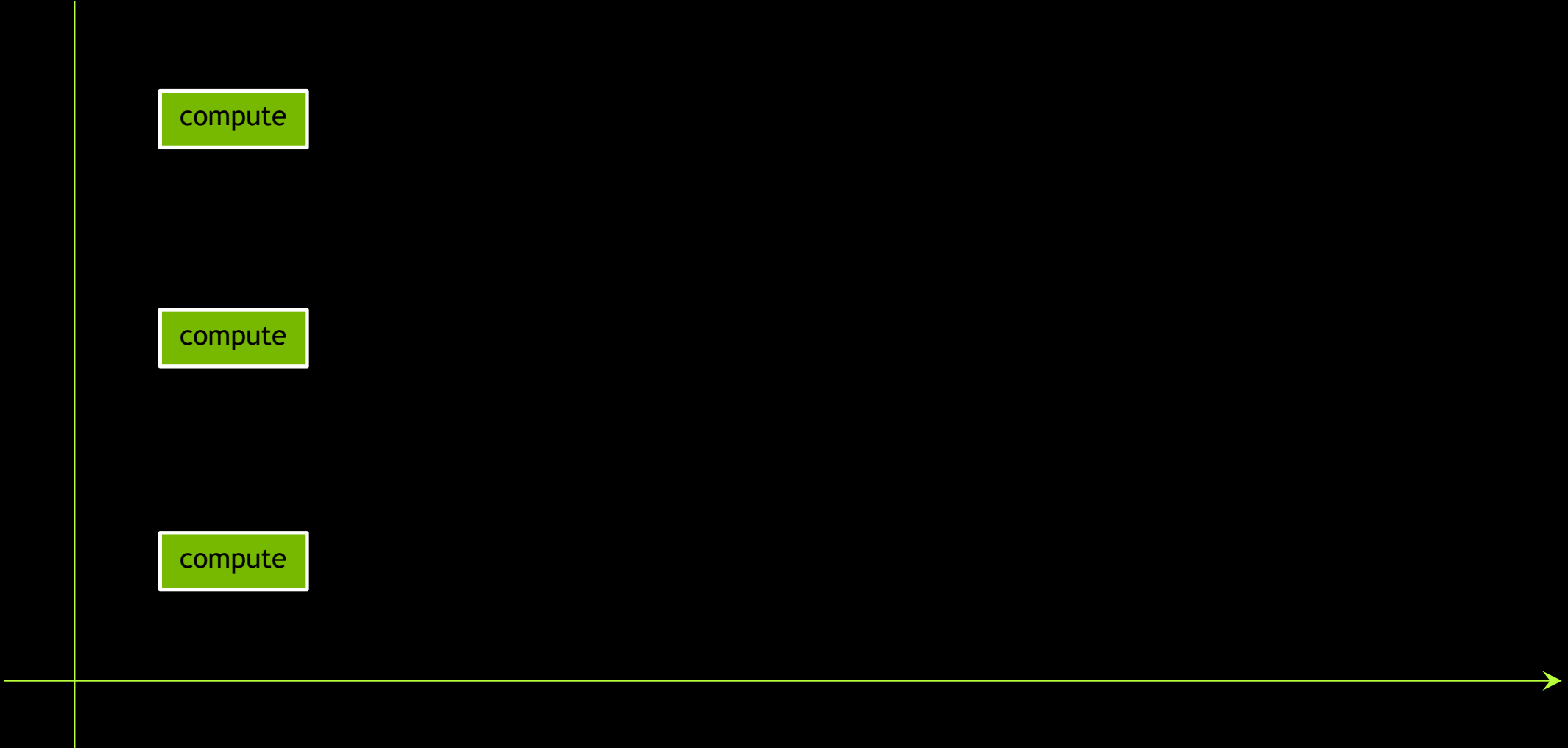


...overlapping compute on multiple devices

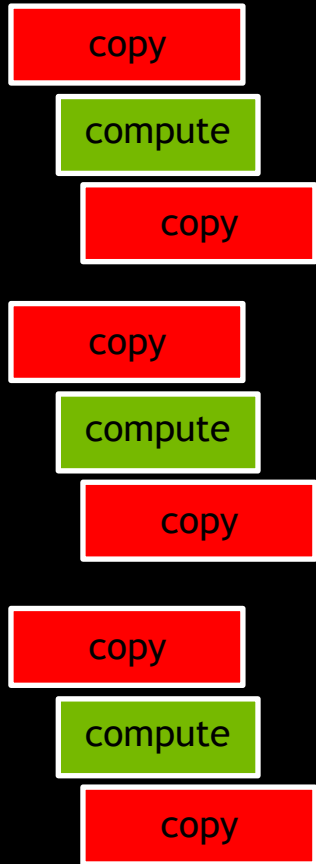
compute

compute

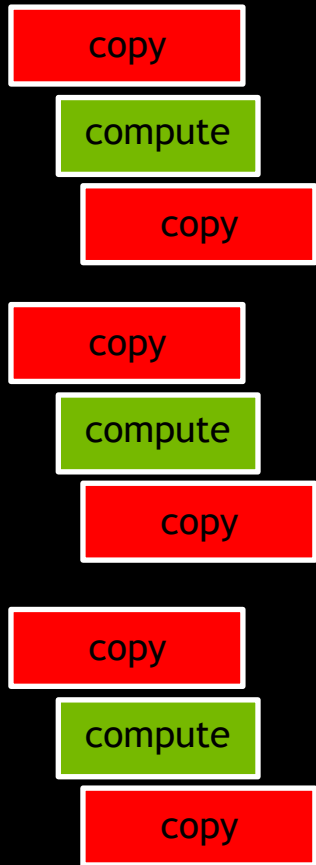
compute



...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



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)

Exercise: Copy/Compute Overlap

Using JupyterLab

Multiple GPUs

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

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)

Exercise: Copy/Compute Overlap

Using JupyterLab

Multiple GPUs

Cipher Application Overview

Considerations for Multiple GPUs

Nsight Systems Setup

Exercise: Multiple GPUs

CUDA Streams

Exercise: Multiple GPUs with Copy/Compute Overlap

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

Multiple GPUs

Cipher Application Overview

Considerations for Multiple GPUs

Nsight Systems Setup

Exercise: Multiple GPUs

CUDA Streams

Exercise: Multiple GPUs with Copy/Compute Overlap

Kernel Launches in Non-Default Streams

Course Survey

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

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

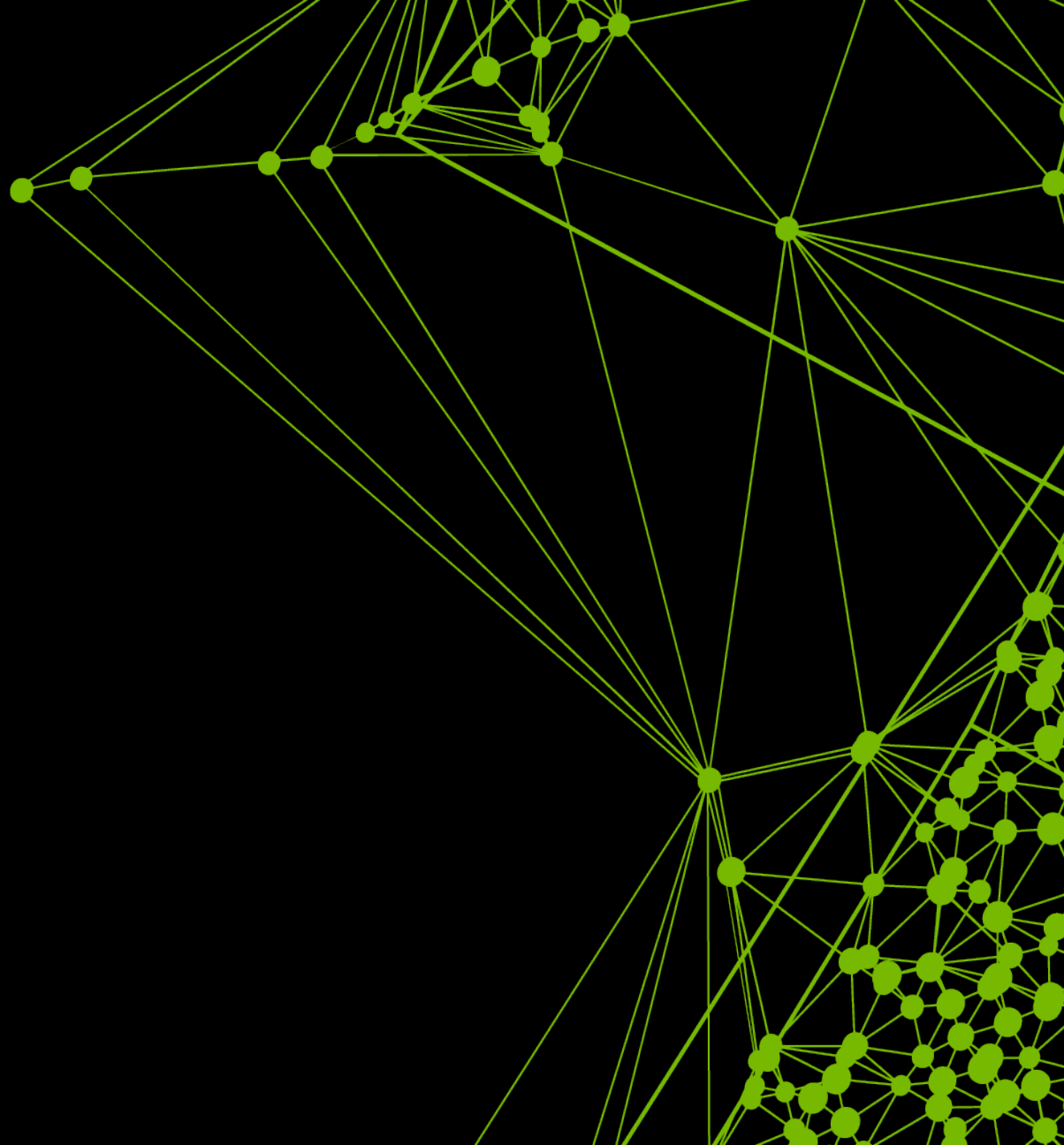
Exercise: Multiple GPUs

Exercise: Multiple GPUs with Copy/Compute Overlap

Course Survey

Course Assessment

Next Steps



nvidia.

DEEP
LEARNING
INSTITUTE

www.nvidia.com/dli