

GPU Architecture

Pierre Aubert



1980s

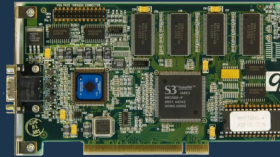


Graphic Processing Unit

1980s



1993



Graphic Processing Unit

1980s



1993



1995



Graphic Processing Unit

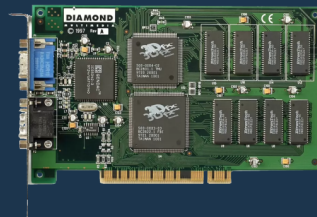
1980s



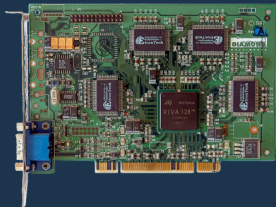
1993



1995



1997



Graphic Processing Unit

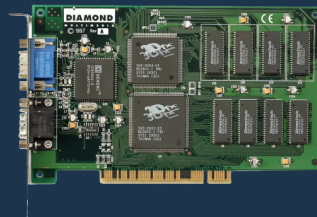
1980s



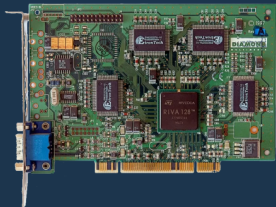
1993



1995



1997



2005

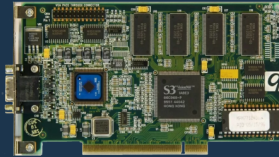


Graphic Processing Unit

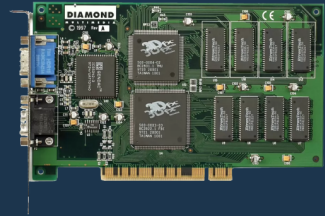
1980s



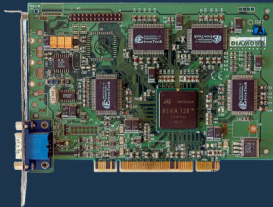
1993



1995



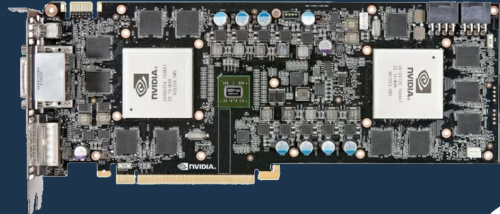
1997



2005



2011



2015



Graphic Processing Unit

2015



2018



Graphic Processing Unit

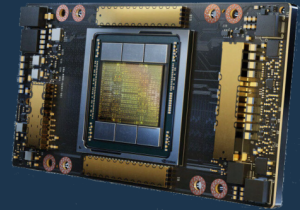
2015



2018



2020



Graphic Processing Unit

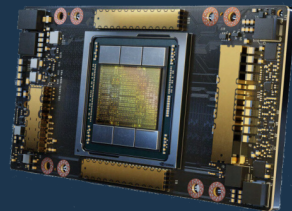
2015



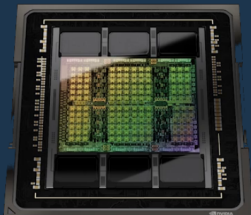
2018



2020



2022-04



Graphic Processing Unit

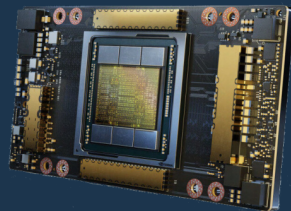
2015



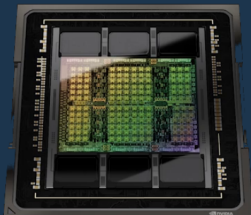
2018



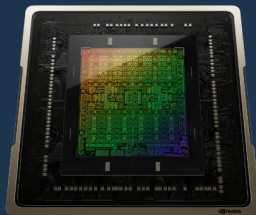
2020



2022-04



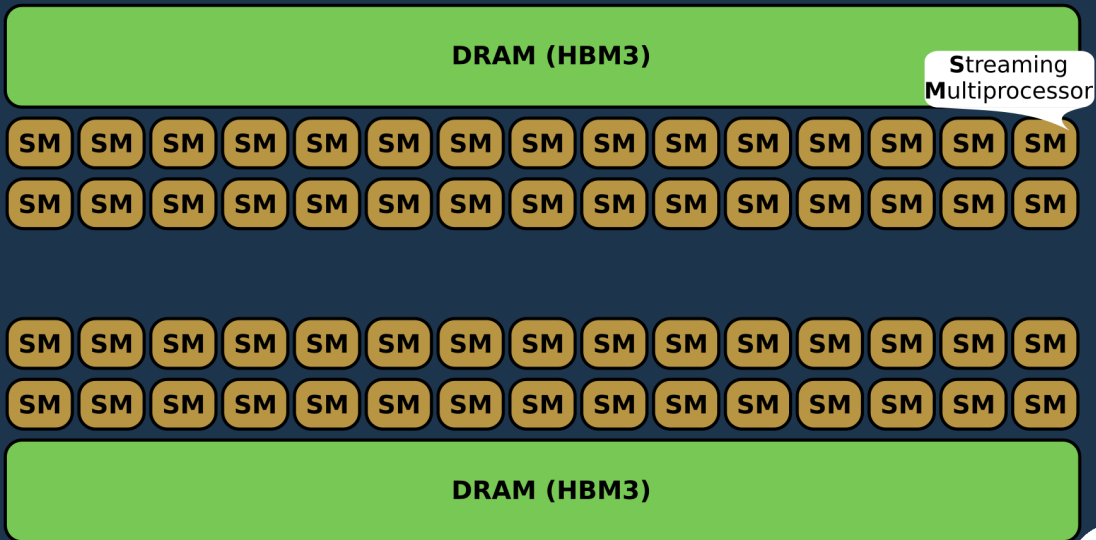
2022-09



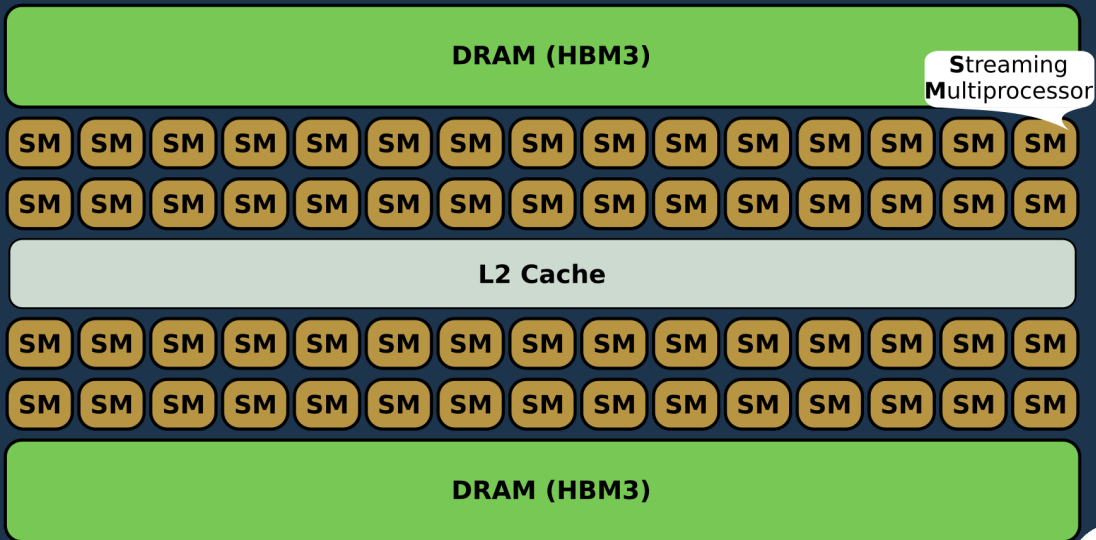
DRAM (HBM3)

DRAM (HBM3)

GPU Architecture



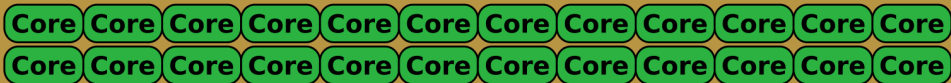
GPU Architecture



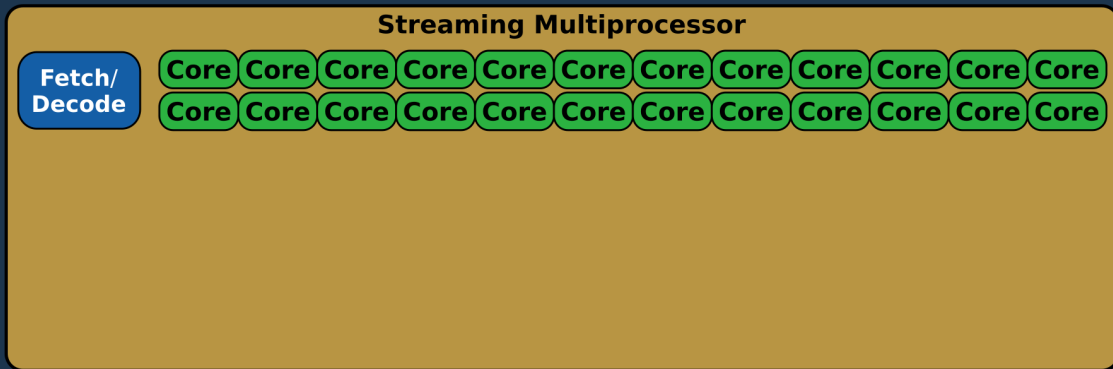
Streaming Multiprocessor

Streaming Multiprocessor

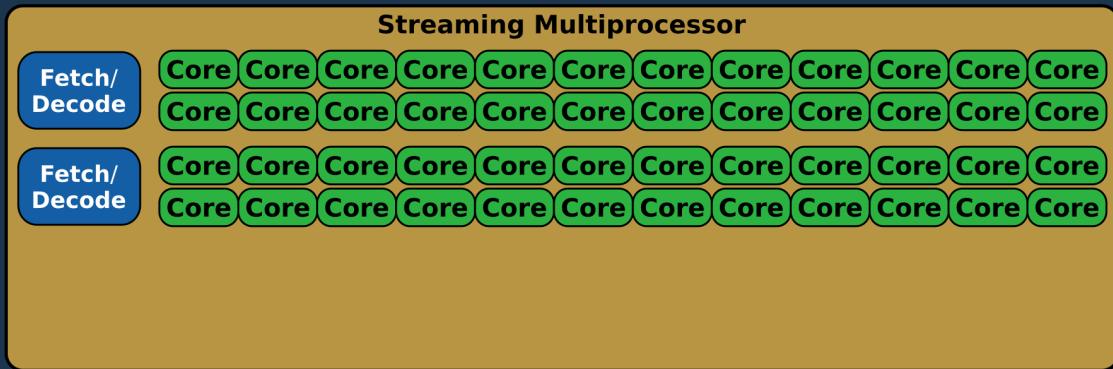
Streaming Multiprocessor



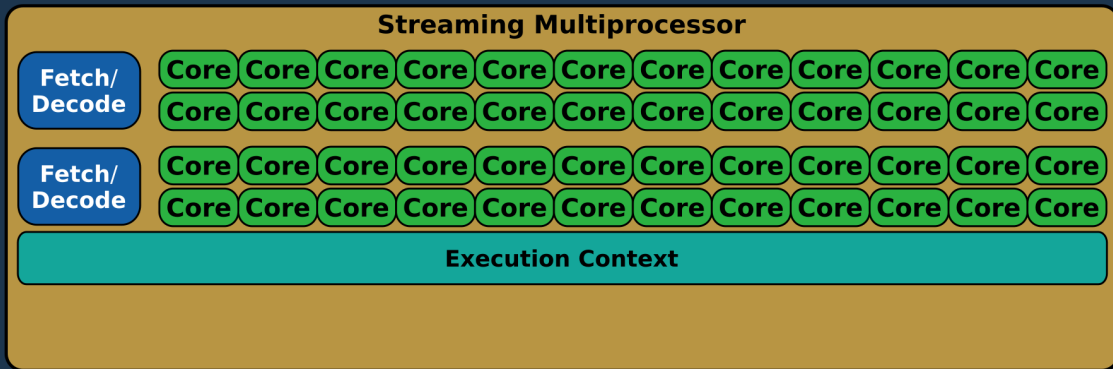
Streaming Multiprocessor



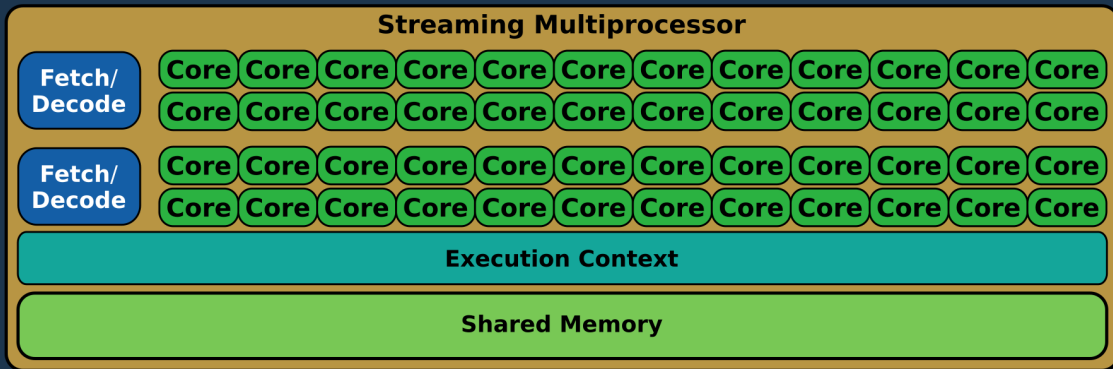
Streaming Multiprocessor



Streaming Multiprocessor

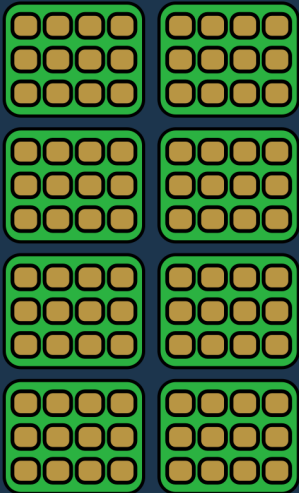


Streaming Multiprocessor

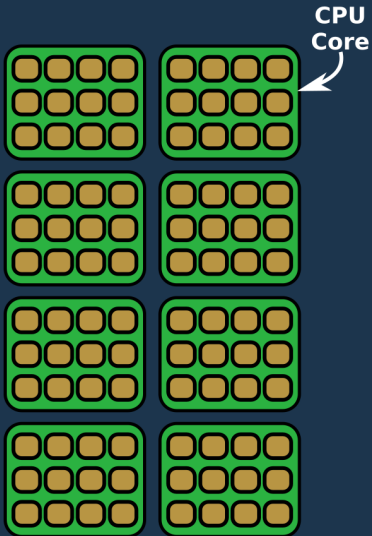


Threads ?

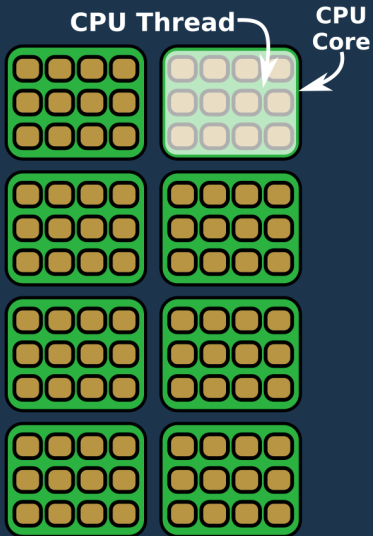
Threads ?



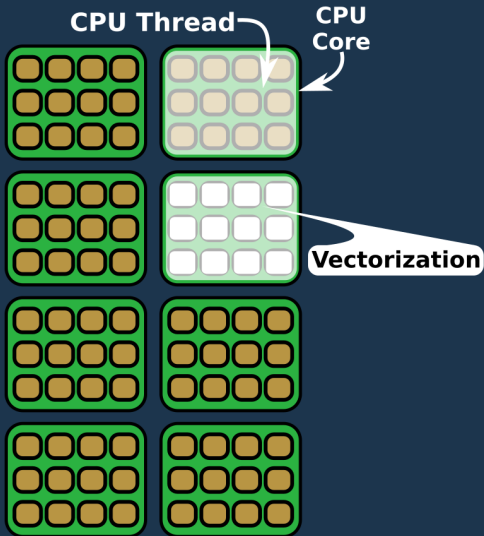
Threads ?



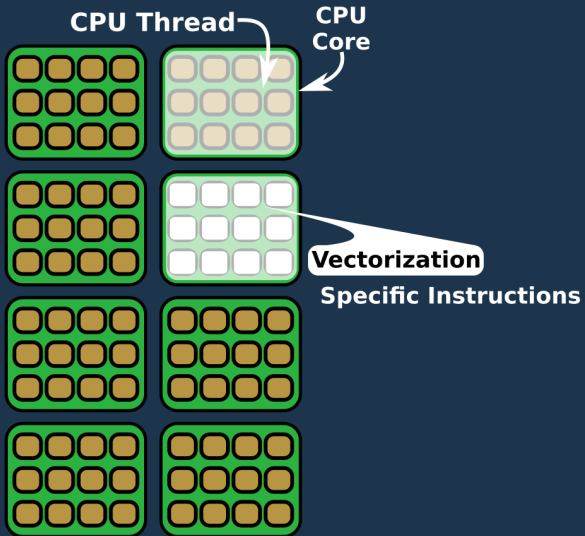
Threads ?



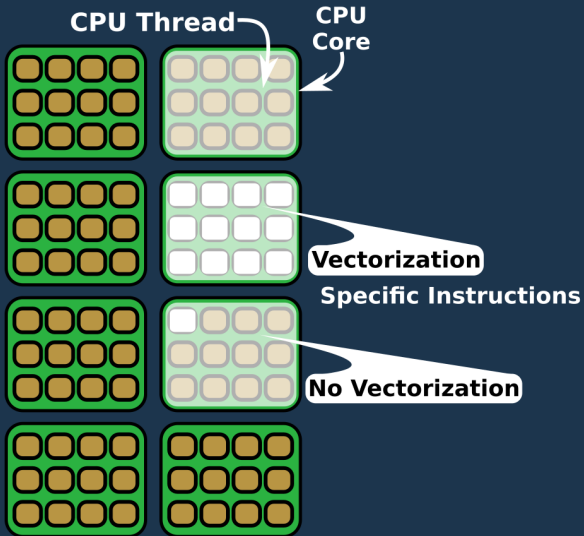
Threads ?



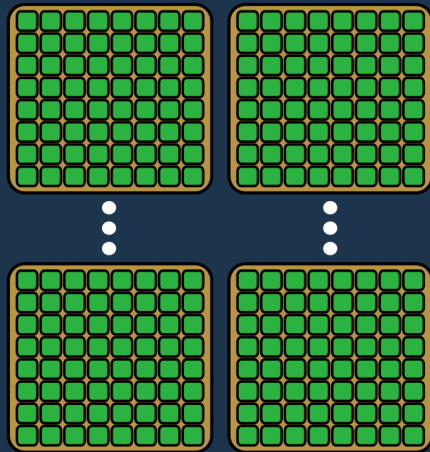
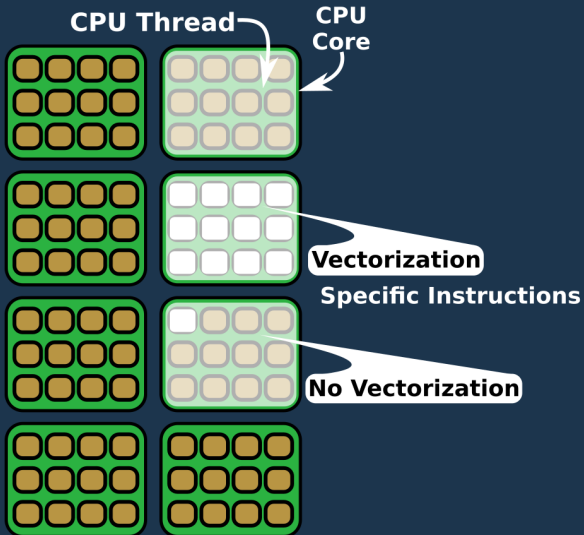
Threads ?



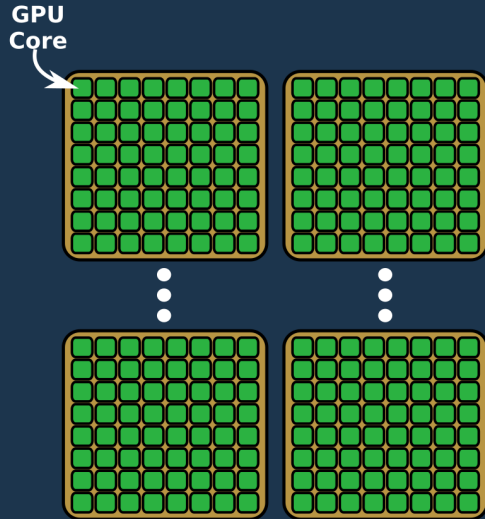
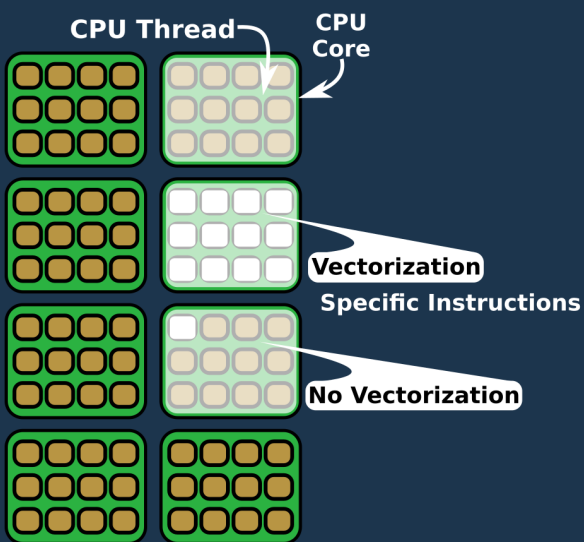
Threads ?



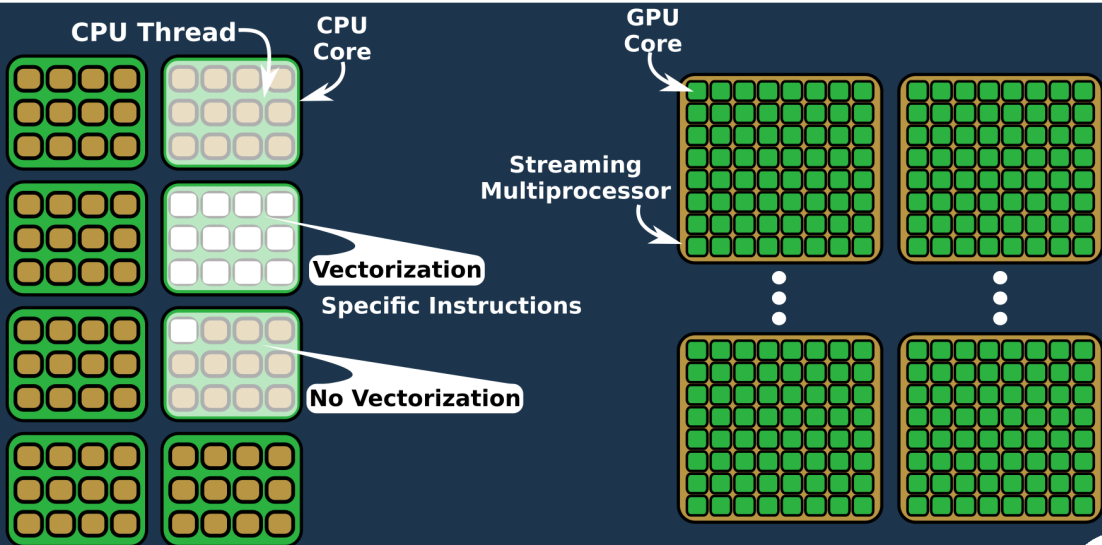
Threads ?



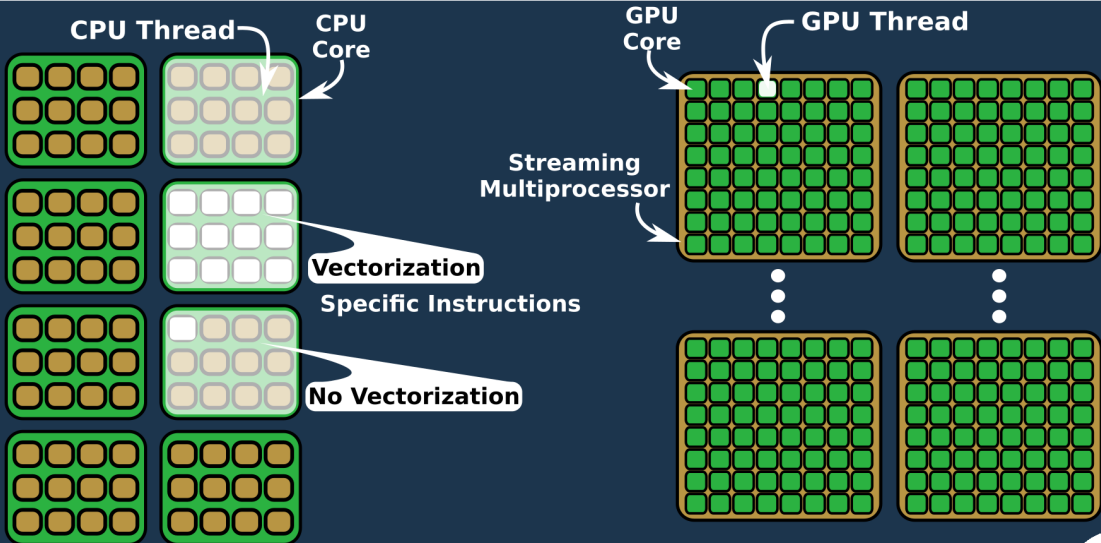
Threads ?



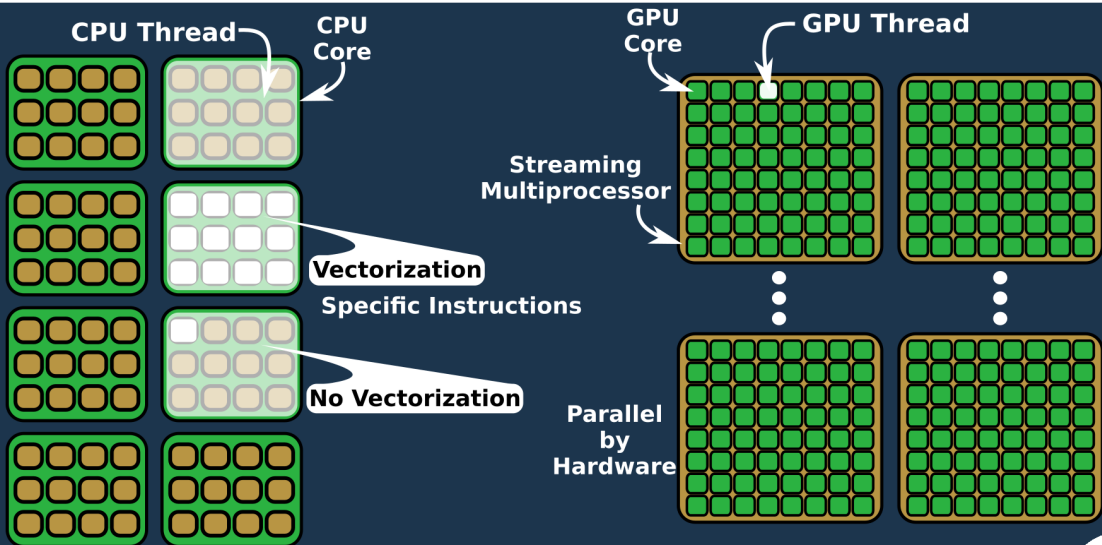
Threads ?



Threads ?



Threads ?



Threads Hierarchy

Threads Hierarchy



Thread

Threads Hierarchy



Thread

Core

Threads Hierarchy

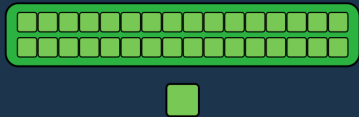


Thread

Core

Executes Computing Kernel

Threads Hierarchy



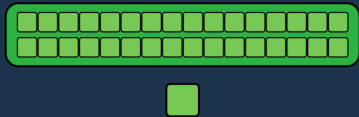
Warp

Thread

Core

Executes Computing Kernel

Threads Hierarchy



Warp

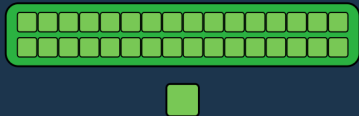
Hardware
Scheduler

Thread

Core

Executes Computing Kernel

Threads Hierarchy



Warp

Hardware
Scheduler

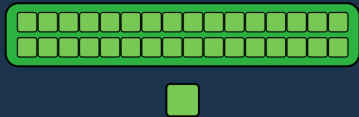
32 **Threads**

Thread

Core

Executes Computing Kernel

Threads Hierarchy



Warp

Thread

Hardware
Scheduler

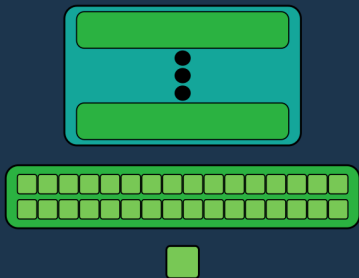
Core

32 **Threads**

Executes Computing Kernel

Synchronous
Same Instruction
Multiple Data

Threads Hierarchy



Block

Warp

Thread

Hardware
Scheduler

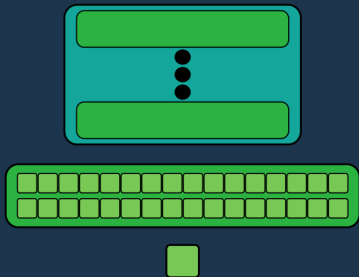
Core

32 **Threads**

Executes Computing Kernel

Synchronous
Same Instruction
Multiple Data

Threads Hierarchy



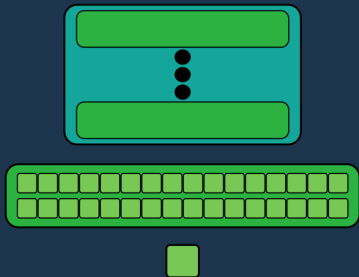
Block Streaming Multiprocessor

Warp Hardware Scheduler

32 Threads **Synchronous**
Same Instruction
Multiple Data

Thread Core Executes Computing Kernel

Threads Hierarchy

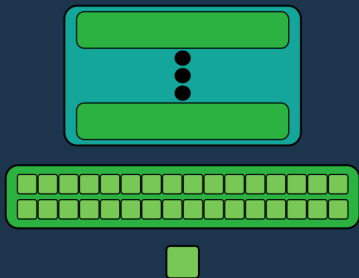


Block Streaming Multiprocessor 2 - 4 **Warps**
64 - 128 **Cores**

Warp Hardware Scheduler 32 **Threads** **Synchronous**
Same Instruction Multiple Data

Thread Core Executes Computing Kernel

Threads Hierarchy



Block Streaming Multiprocessor 2 - 4 **Warps**
64 - 128 **Cores** **Asynchronous**
Same Instruction
Multiple Data

Warp Hardware Scheduler 32 **Threads** **Synchronous**
Same Instruction
Multiple Data

Thread Core Executes Computing Kernel

Threads Hierarchy



Cluster



Block

Streaming
Multiprocessor

2 - 4 **Warps**
64 - 128 **Cores**

Asynchronous
Same Instruction
Multiple Data



Warp

Hardware
Scheduler

32 **Threads**

Synchronous
Same Instruction
Multiple Data

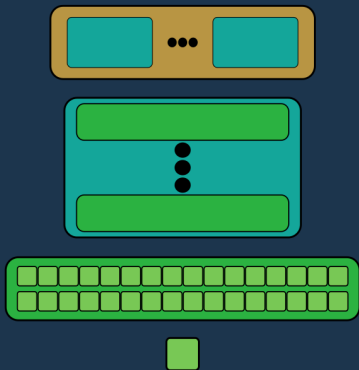


Thread

Core

Executes Computing Kernel

Threads Hierarchy



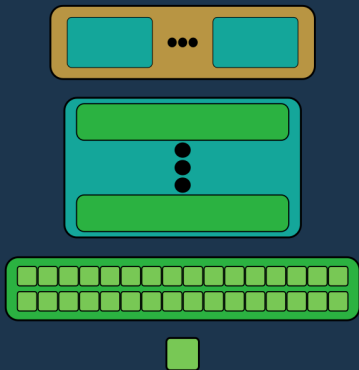
Cluster Blocks
Communication

Block Streaming 2 - 4 **Warps** **Asynchronous**
Multiprocessor 64 - 128 **Cores** Same Instruction
Multiple Data

Warp Hardware 32 **Threads** **Synchronous**
Scheduler Same Instruction
Multiple Data

Thread Core Executes Computing Kernel

Threads Hierarchy



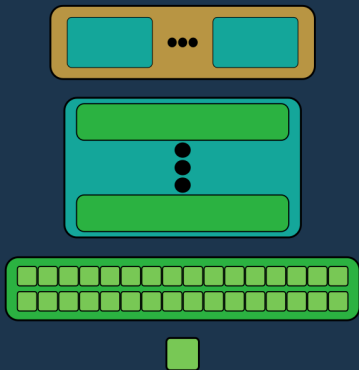
Cluster Blocks
Communication Since :
- **CUDA 12**
- **Hopper**

Block Streaming
Multiprocessor 2 - 4 **Warps** **Asynchronous**
64 - 128 **Cores** Same Instruction
Multiple Data

Warp Hardware
Scheduler 32 **Threads** **Synchronous**
Same Instruction
Multiple Data

Thread Core Executes Computing Kernel

Threads Hierarchy



Cluster	Blocks Communication	Since : - CUDA 12 - Hopper	Hopper : 132 SMs Ampere : 108 SMs
Block	Streaming Multiprocessor	2 - 4 Warps 64 - 128 Cores	Asynchronous Same Instruction Multiple Data
Warp	Hardware Scheduler	32 Threads	Synchronous Same Instruction Multiple Data
Thread	Core	Executes Computing Kernel	

Threads Hierarchy



Grid



Cluster

Blocks
Communication

Since :
- CUDA **12**
- **Hopper**

Hopper : 132 SMs
Ampere : 108 SMs



Block

Streaming
Multiprocessor

2 - 4 **Warps**
64 - 128 **Cores**

Asynchronous
Same Instruction
Multiple Data



Warp

Hardware
Scheduler

32 **Threads**

Synchronous
Same Instruction
Multiple Data



Thread

Core

Executes Computing Kernel

Threads Hierarchy



Grid

Full **GPU**



Cluster

Blocks
Communication

Since :
- **CUDA 12**
- **Hopper**

Hopper : 132 SMs
Ampere : 108 SMs



Block

Streaming
Multiprocessor

2 - 4 **Warps**
64 - 128 **Cores**

Asynchronous
Same Instruction
Multiple Data



Warp

Hardware
Scheduler

32 **Threads**

Synchronous
Same Instruction
Multiple Data

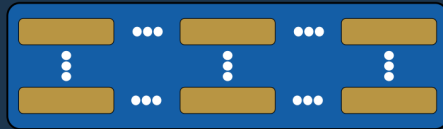


Thread

Core

Executes Computing Kernel

Threads Hierarchy



Grid

Full **GPU**

Hopper : 8448 Cores
Ampere : 6912 Cores

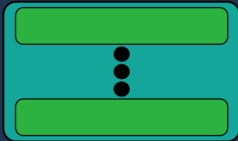


Cluster

Blocks
Communication

Since :
- **CUDA 12**
- **Hopper**

Hopper : 132 SMs
Ampere : 108 SMs



Block

Streaming
Multiprocessor

2 - 4 **Warps**
64 - 128 **Cores**

Asynchronous
Same Instruction
Multiple Data



Warp

Hardware
Scheduler

32 **Threads**

Synchronous
Same Instruction
Multiple Data



Thread

Core

Executes Computing Kernel

Threads Execution

Threads Execution

 Thread

Threads Execution

 Thread



Warp

Threads Execution

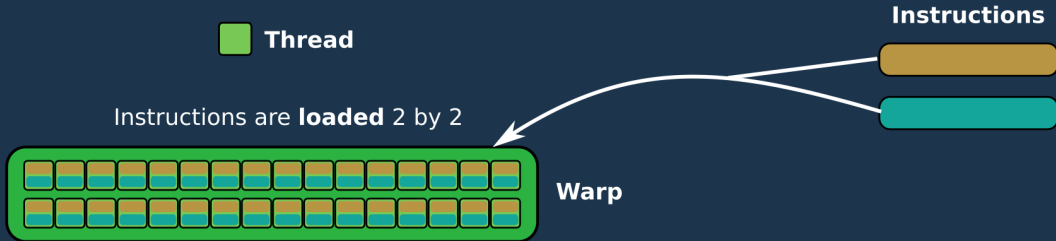
 Thread

Instructions

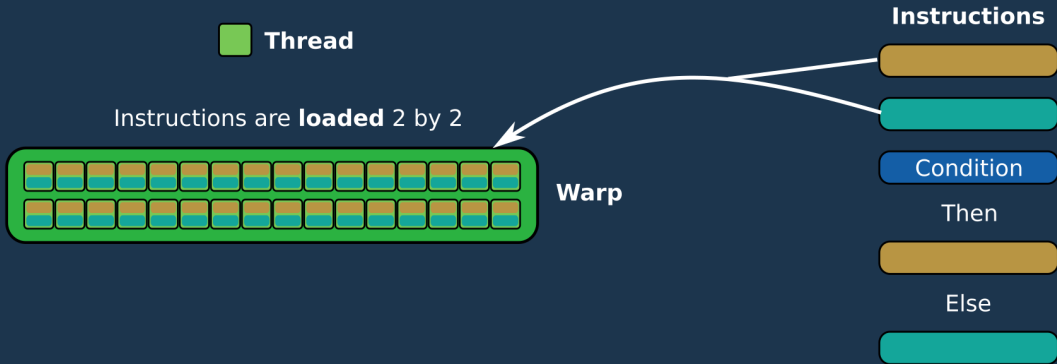


Warp

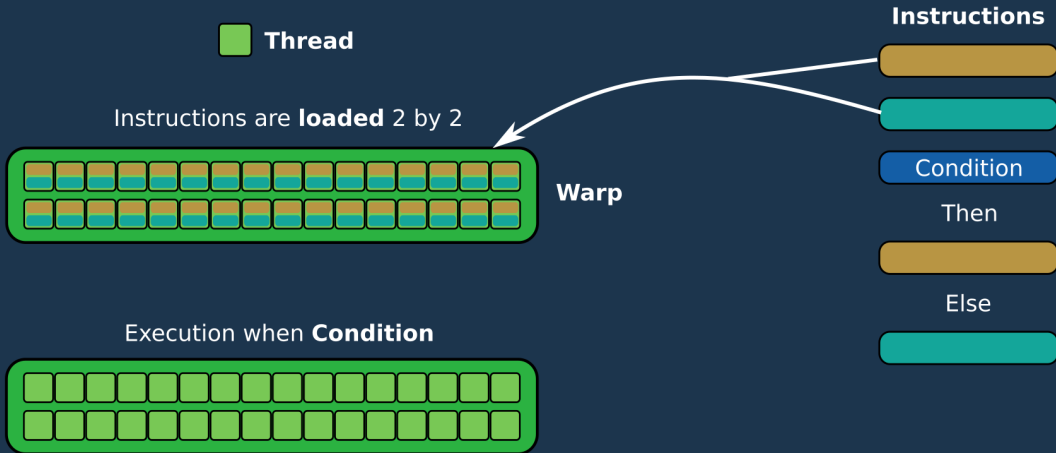
Threads Execution



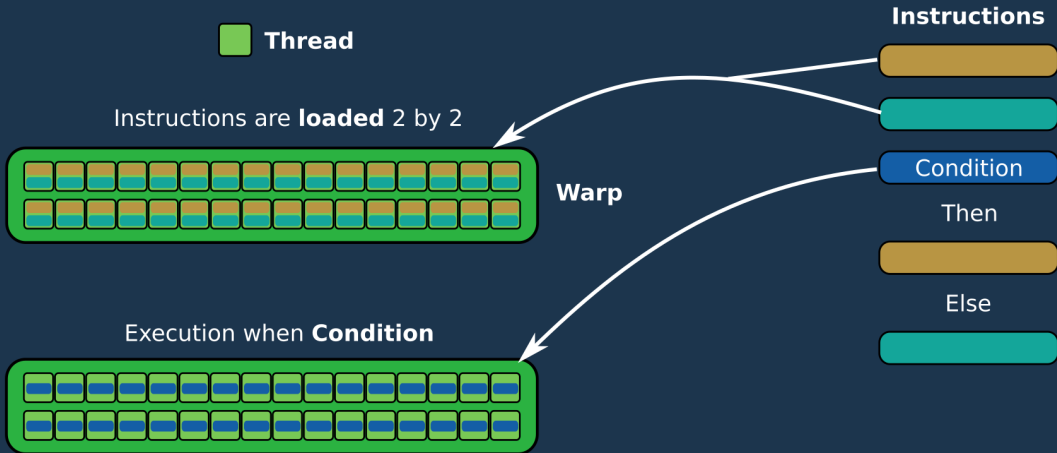
Threads Execution



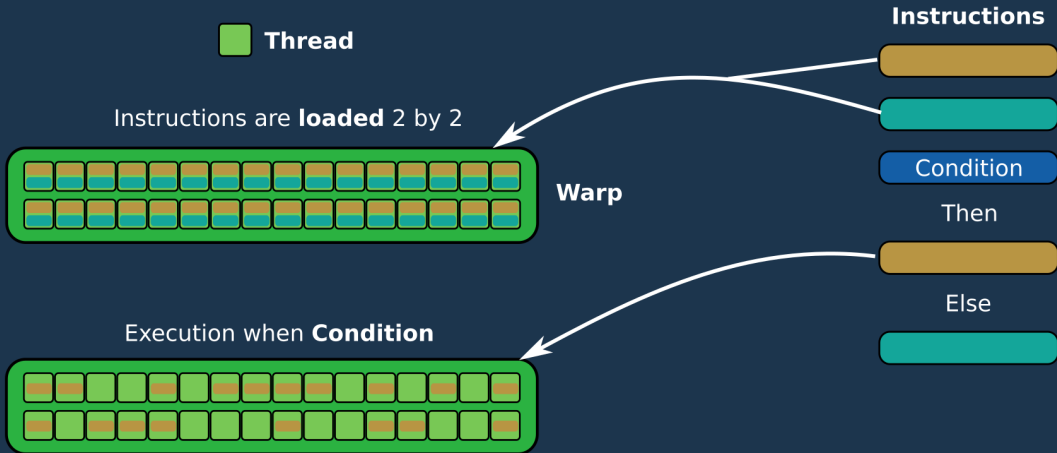
Threads Execution



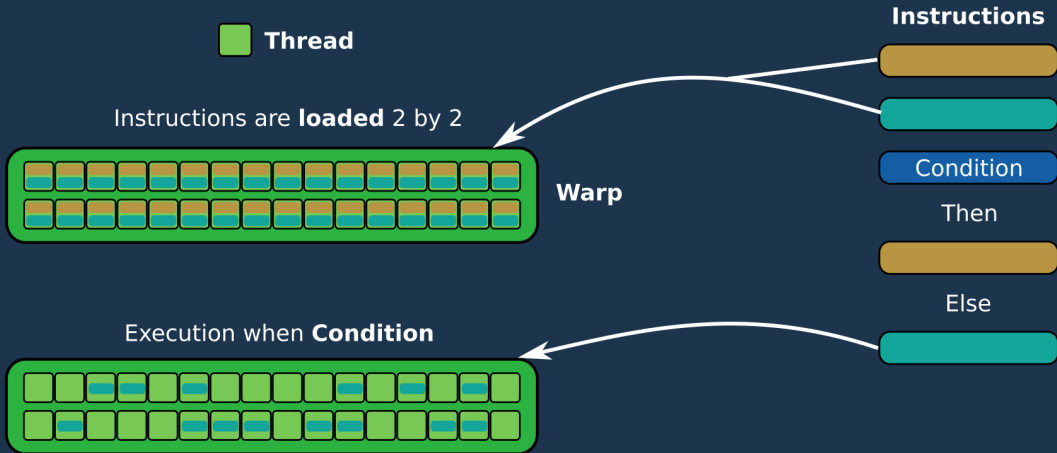
Threads Execution



Threads Execution



Threads Execution



Memory Management

CUDA

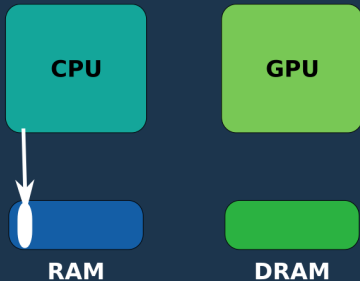


RAM

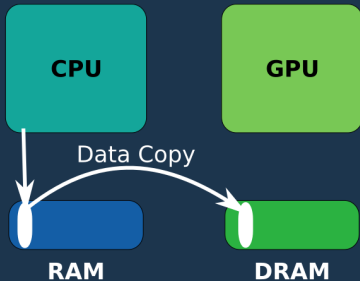


DRAM

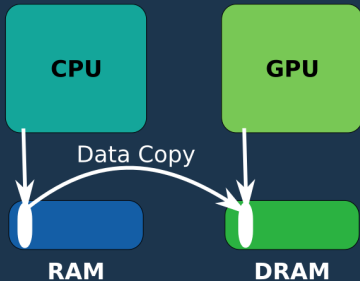
CUDA



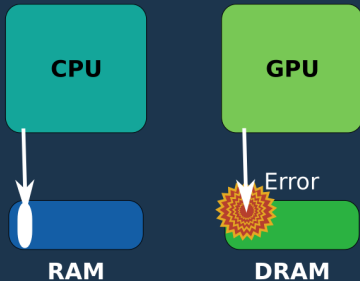
CUDA



CUDA

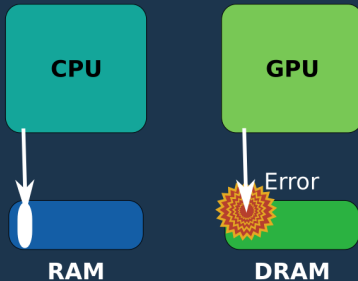


CUDA

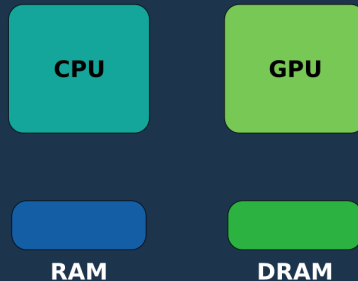


Memory Management

CUDA

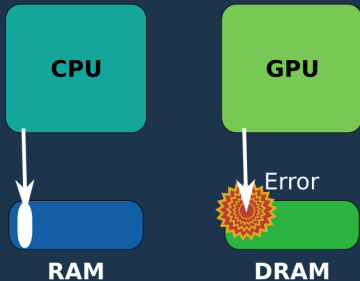


nvc++/nvfortran
Unified Memory

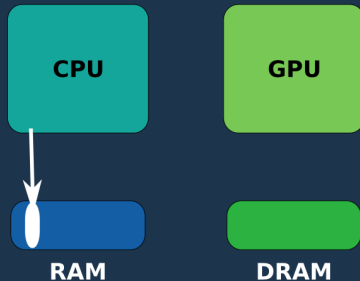


Memory Management

CUDA

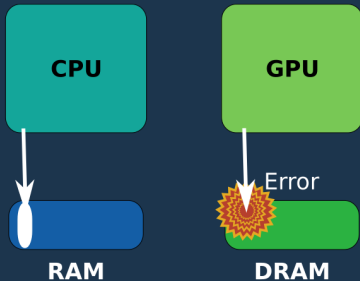


nvc++/nvfortran
Unified Memory

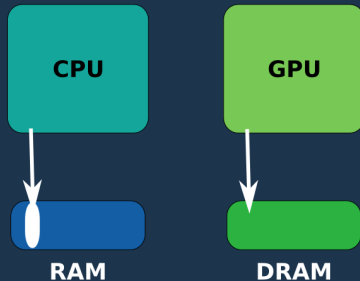


Memory Management

CUDA

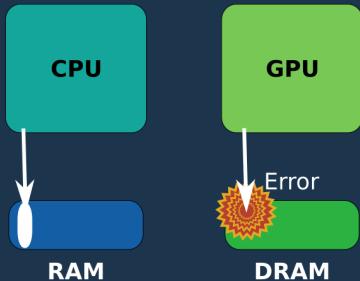


nvc++/nvfortran
Unified Memory

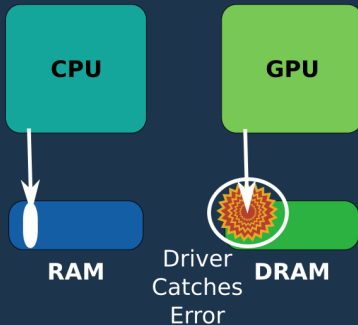


Memory Management

CUDA

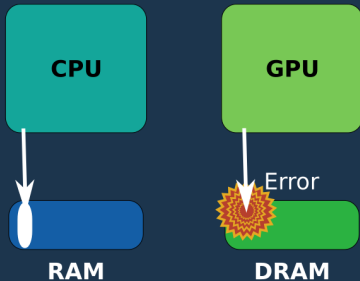


nvc++/nvfortran
Unified Memory

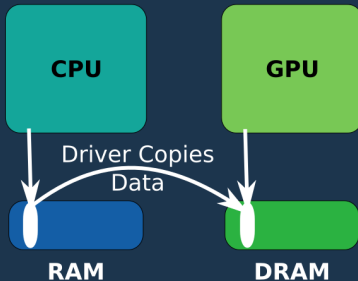


Memory Management

CUDA

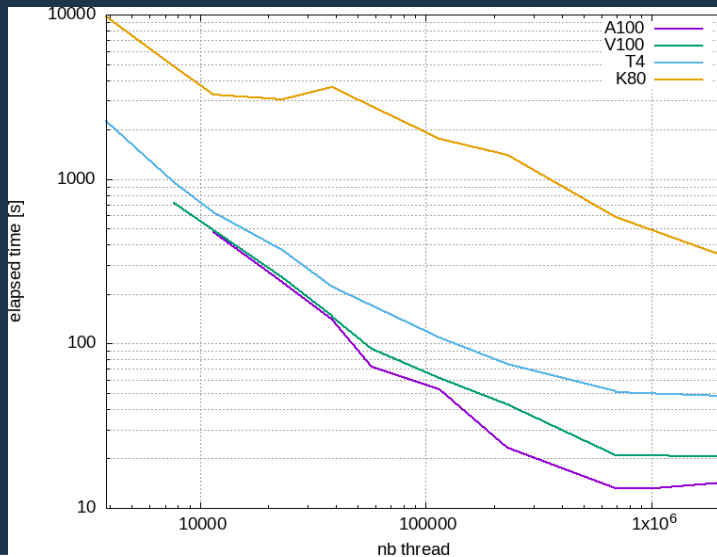


nvc++/nvfortran
Unified Memory

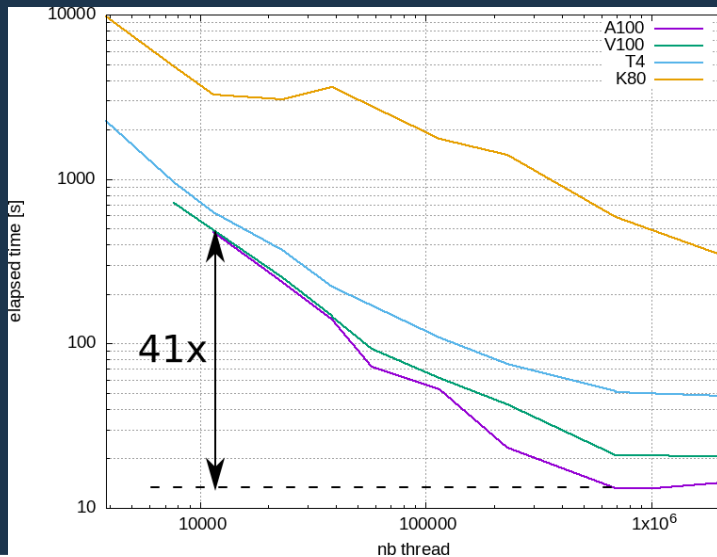


How many threads ?

How many threads ?



How many threads ?



Element



Express Parallelism

Element

Independent Elements => Independent Computing => Parallelism



Express Parallelism

Element Independent Elements => Independent Computing => Parallelism



Particle

Express Parallelism

Element

Independent Elements => Independent Computing => Parallelism



Particle

p_x 

p_y 

p_z 

Express Parallelism

Element

Independent Elements => Independent Computing => Parallelism



Particle

Mix Components :
- Dot Product
- Cross Product

p_x



p_y



p_z



Express Parallelism

Element

Independent Elements => Independent Computing => Parallelism



Particle

Mix Components :
- Dot Product
- Cross Product

p_x



p_y



p_z



Express Parallelism

Element

Independent Elements => Independent Computing => Parallelism



Particle

Mix Components :
- Dot Product
- Cross Product

p_x



p_y



p_z



v_x



v_y



v_z



Express Parallelism

Element

Independent Elements => Independent Computing => Parallelism

Mix Components :
- Dot Product
- Cross Product

Particle

p_x

p_y

p_z

v_x

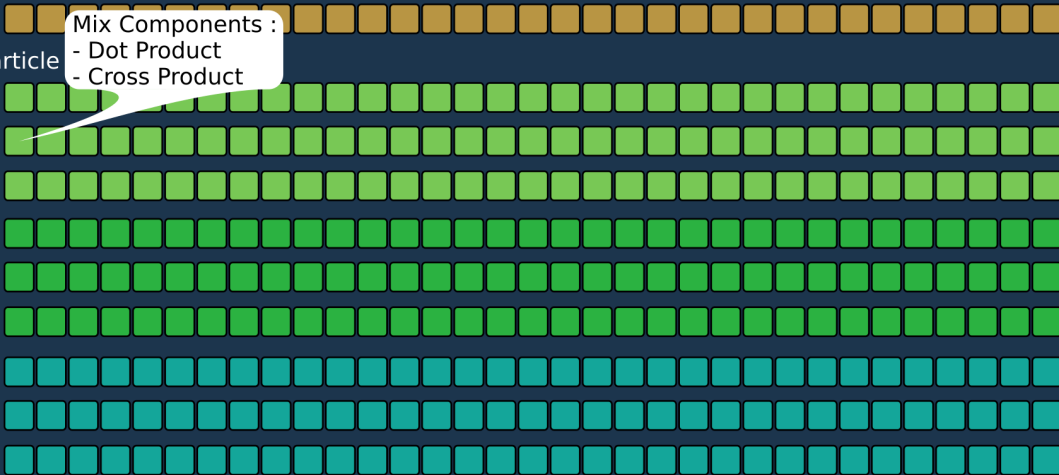
v_y

v_z

a_x

a_y

a_z



Express Parallelism

Element

Independent Elements => Independent Computing => Parallelism

Mix Components :
- Dot Product
- Cross Product

Particle

p_x

p_y

p_z

v_x

v_y

v_z

a_x

a_y

a_z



Express Parallelism

Element

Independent Elements => Independent Computing => Parallelism

Mix Components :
- Dot Product
- Cross Product

Particle

p_x

p_y

p_z

v_x

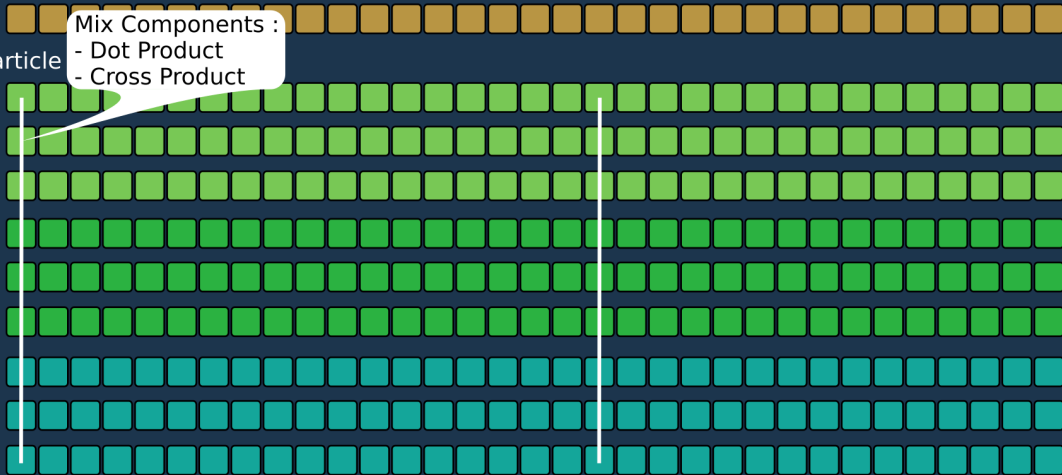
v_y

v_z

a_x

a_y

a_z



Express Parallelism

Element

Independent Elements => Independent Computing => Parallelism

Mix Components :
- Dot Product
- Cross Product

Particle

p_x

p_y

p_z

v_x

v_y

v_z

a_x

a_y

a_z

Vectorisation

Express Parallelism

Element

Independent Elements => Independent Computing => Parallelism

Mix Components :
- Dot Product
- Cross Product

Particle

p_x

p_y

p_z

v_x

v_y

v_z

a_x

a_y

a_z

Vectorisation

Event



Express Parallelism

Event

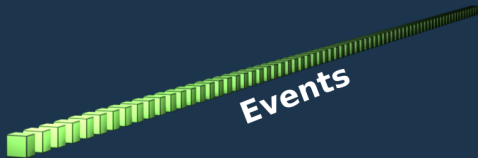
Independent Events => Independent Computing => Parallelism



Express Parallelism

Event

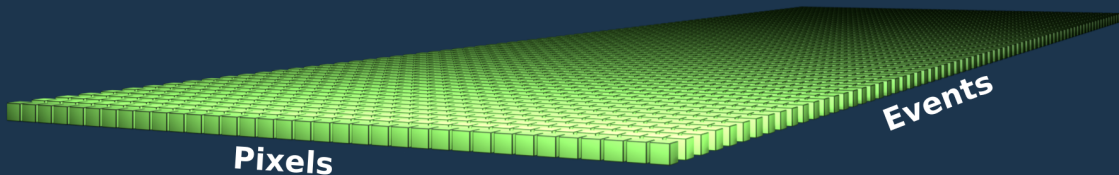
Independent Events => Independent Computing => Parallelism



Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Express Parallelism

Event

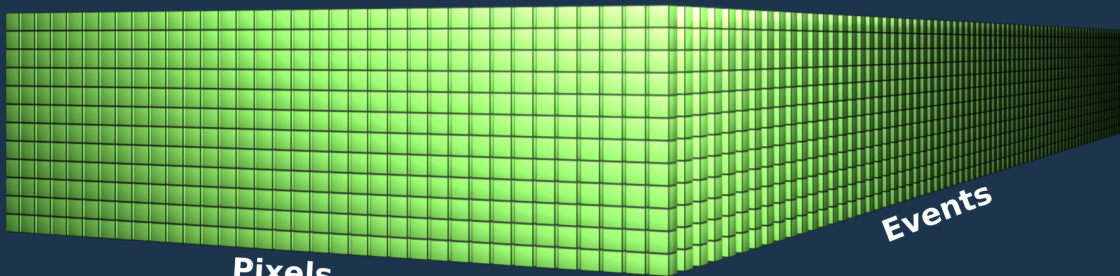
Independent Events => Independent Computing => Parallelism



Slices

Pixels

Events



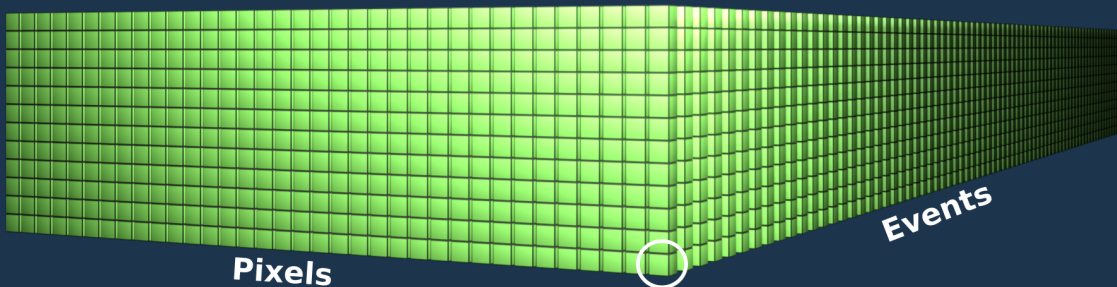
Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Slices



Pixels

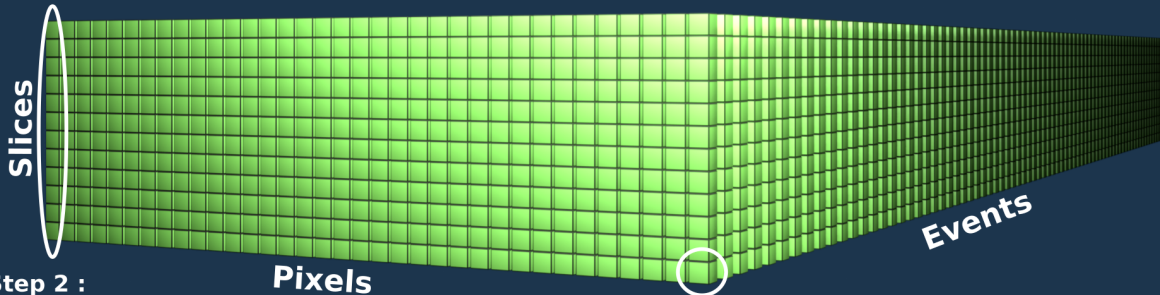
Events

Step 1 : Independent Computing in Calibration

Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Step 2 :

Independent Computing in Integration

Pixels

Step 1 : Independent Computing in Calibration

Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Reduction

Slices

Events

Step 2 :

Independent Computing in Integration

Pixels

Step 1 : Independent Computing in Calibration

Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Reduction

Contiguous Data

Slices

Events

Pixels

Step 2 :

Independent Computing in Integration

Step 1 : Independent Computing in Calibration

Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Reduction

Allows
Vectorization

Contiguous Data

Slices

Events

Pixels

Step 2 :
Independent Computing in Integration

Step 1 : **Independent** Computing in Calibration

Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Reduction

Allows
Vectorization

Even for
Integration

Contiguous Data

Slices

Events

Pixels

Step 2 :

Independent Computing in Integration

Step 1 : Independent Computing in Calibration

Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Reduction

Allows
Vectorization

Even for
Integration

Contiguous Data

Slices

Events

Pixels

Step 2 :

Independent Computing in Integration

Step 1 : Independent Computing in Calibration

Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Reduction

Allows
Vectorization

Even for
Integration

Contiguous Data

Slices

Events

Step 2 :

Independent Computing in Integration

Pixels

Step 1 : Independent Computing in Calibration

Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Reduction

Allows
Vectorization

Even for
Integration

Contiguous Data

Slices

Parallelisms :

- Cores
- Nodes
- Clusters

Events

Pixels

Step 2 :

Independent Computing in Integration

Step 1 : Independent Computing in Calibration

Express Parallelism

Event

Independent Events => Independent Computing => Parallelism



Reduction

Allows
Vectorization

Even for
Integration

Computing Drives Data Storage

Contiguous Data

Slices

Parallelisms :
- Cores
- Nodes
- Clusters

Events

Pixels

Step 2 :
Independent Computing in Integration

Step 1 : **Independent** Computing in Calibration

Conclusion

Conclusion

- Express **as much parallelism** as possible

Conclusion

- Express **as much parallelism** as possible
- Use **Vector**, **Matrix** and **Tensor** to store data

- Express **as much parallelism** as possible
- Use **Vector**, **Matrix** and **Tensor** to store data
- Use **Linear Algebra** computation :
 - **Dot product**
 - **Matrix Multiplication** (GEMM)
 - **Saxpy**
 - **Hadamard Product**

- Express **as much parallelism** as possible
- Use **Vector**, **Matrix** and **Tensor** to store data
- Use **Linear Algebra** computation :
 - **Dot product**
 - **Matrix Multiplication** (GEMM)
 - **Saxpy**
 - **Hadamard Product**

Standard interface

Conclusion

- Express **as much parallelism** as possible
- Use **Vector**, **Matrix** and **Tensor** to store data
- Use **Linear Algebra** computation :
 - **Dot product**
 - **Matrix Multiplication** (GEMM)
 - **Saxpy**
 - **Hadamard Product**

Standard interface

Efficient on **multi cores (CPU)**



TBB



Conclusion

- Express **as much parallelism** as possible
- Use **Vector**, **Matrix** and **Tensor** to store data
- Use **Linear Algebra** computation :
 - **Dot product**
 - **Matrix Multiplication** (GEMM)
 - **Saxpy**
 - **Hadamard Product**

Standard interface

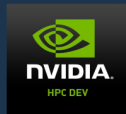
Efficient on **multi cores (CPU)**



TBB



Efficient on **many cores (GPU)**



Vulkan