Introduction to Accelerated/Hybrid Computing with GPGPU Architectures

Carlos J. Barrios H., PhD

@carlosjaimebh

@SuperCCamp

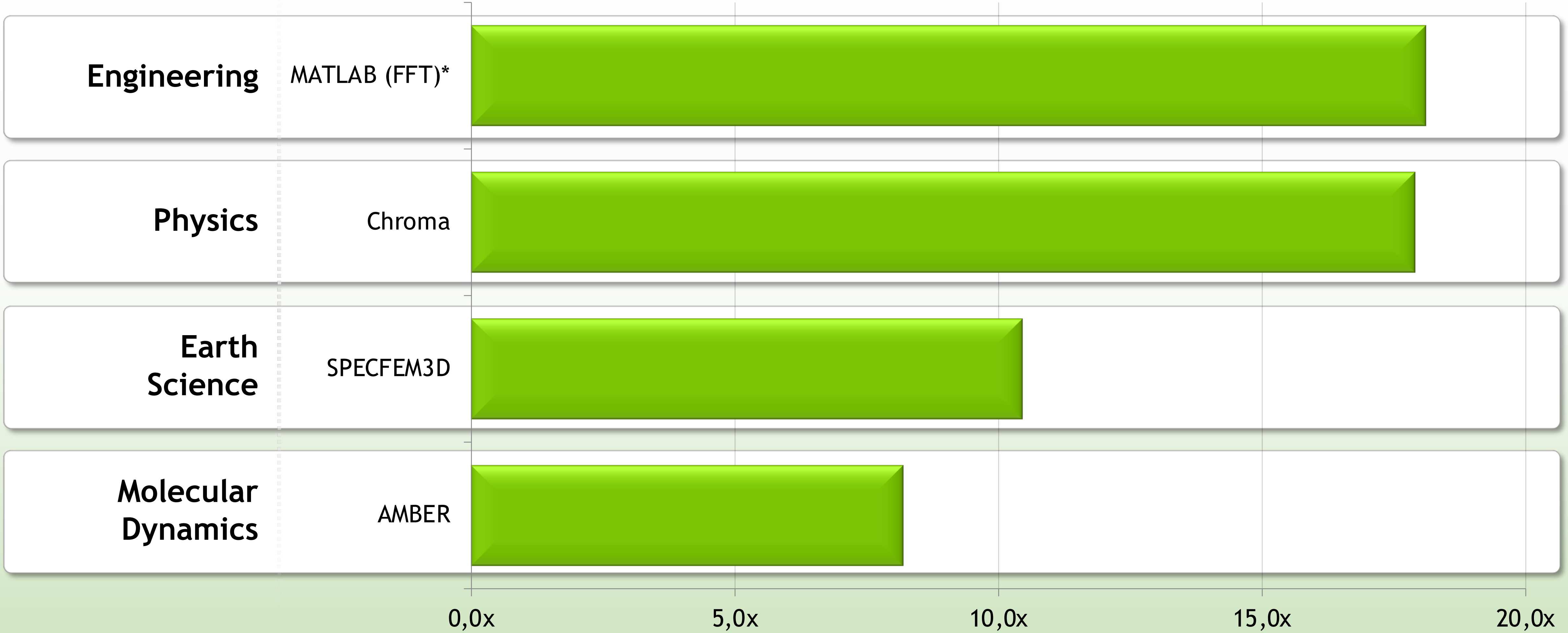
NVIDIA

DEEP
LEARNING
INSTITUTE

UNIVERSITY
AMBASSADOR

Fastest Performance on Scientific Applications

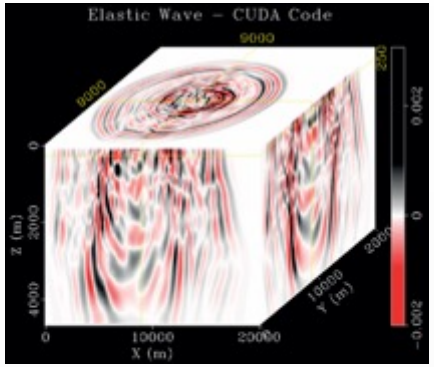
Comparing Tesla K20X Speed-Up over Sandy Bridge CPUs

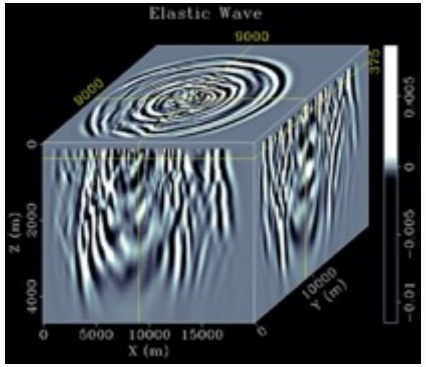


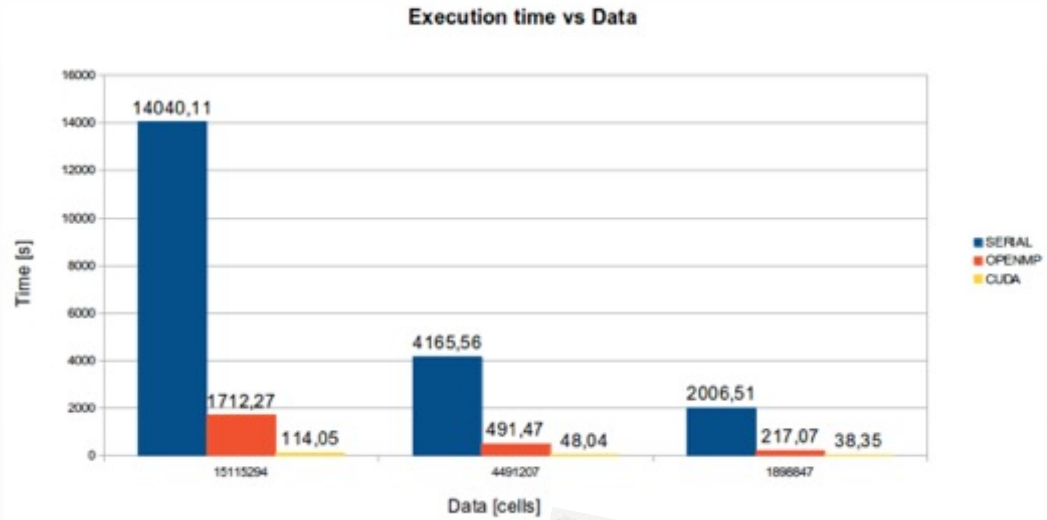
CPU results: Dual socket E5-2687w, 3.10 GHz, GPU results: Dual socket E5-2687w + 2 Tesla K20X GPUs
*MATLAB results comparing one i7-2600K CPU vs with Tesla K20 GPU
Disclaimer: Non-NVIDIA implementations may not have been fully optimized

© NVIDIA 2013

Interesting @SC3UIS Experiences



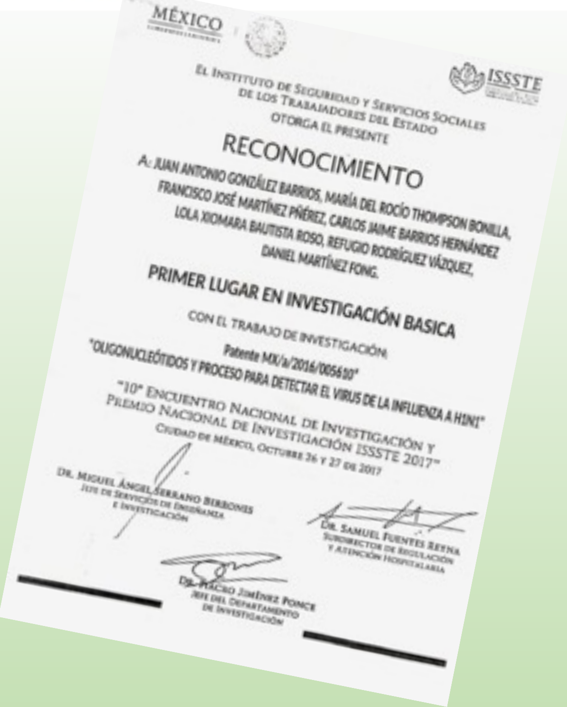




Processing and Visualization for Oil Reservoirs (3D seismic modelling in isotropic and heterogeneous media)

For 10 Millons of bases

0. Original App 3 Months
1. App (3 Weeks)
2. App (2- Days)
3. App (4 Minutes)



MÉXICO

ISSSTE

EL INSTITUTO DE SEGURIDAD Y SERVICIOS SOCIALES DE LOS TRABAJADORES DEL ESTADO
OTORGA EL PRESENTE

RECONOCIMIENTO

A: JUAN ANTONIO GONZÁLEZ BARRIOS, MARÍA DEL ROCÍO THOMPSON BONILLA, FRANCISCO JOSÉ MARTÍNEZ PÉREZ, CARLOS JAIME BARRIOS HERNÁNDEZ LOLA XIOMARA BAUTISTA ROSO, REFUGIO RODRÍGUEZ VÁZQUEZ, DANIEL MARTÍNEZ FONG.

PRIMER LUGAR EN INVESTIGACIÓN BASICA

CON EL TRABAJO DE INVESTIGACIÓN
Patente MX/a/2016/005610"
"OLIGONUCLEÓTIDOS Y PROCESO PARA DETECTAR EL VIRUS DE LA INFLUENZA A H1N1"
"10° ENCUENTRO NACIONAL DE INVESTIGACIÓN Y PREMIO NACIONAL DE INVESTIGACIÓN ISSSTE 2017"
CIUDAD DE MÉXICO, OCTUBRE 26 Y 27 DE 2017

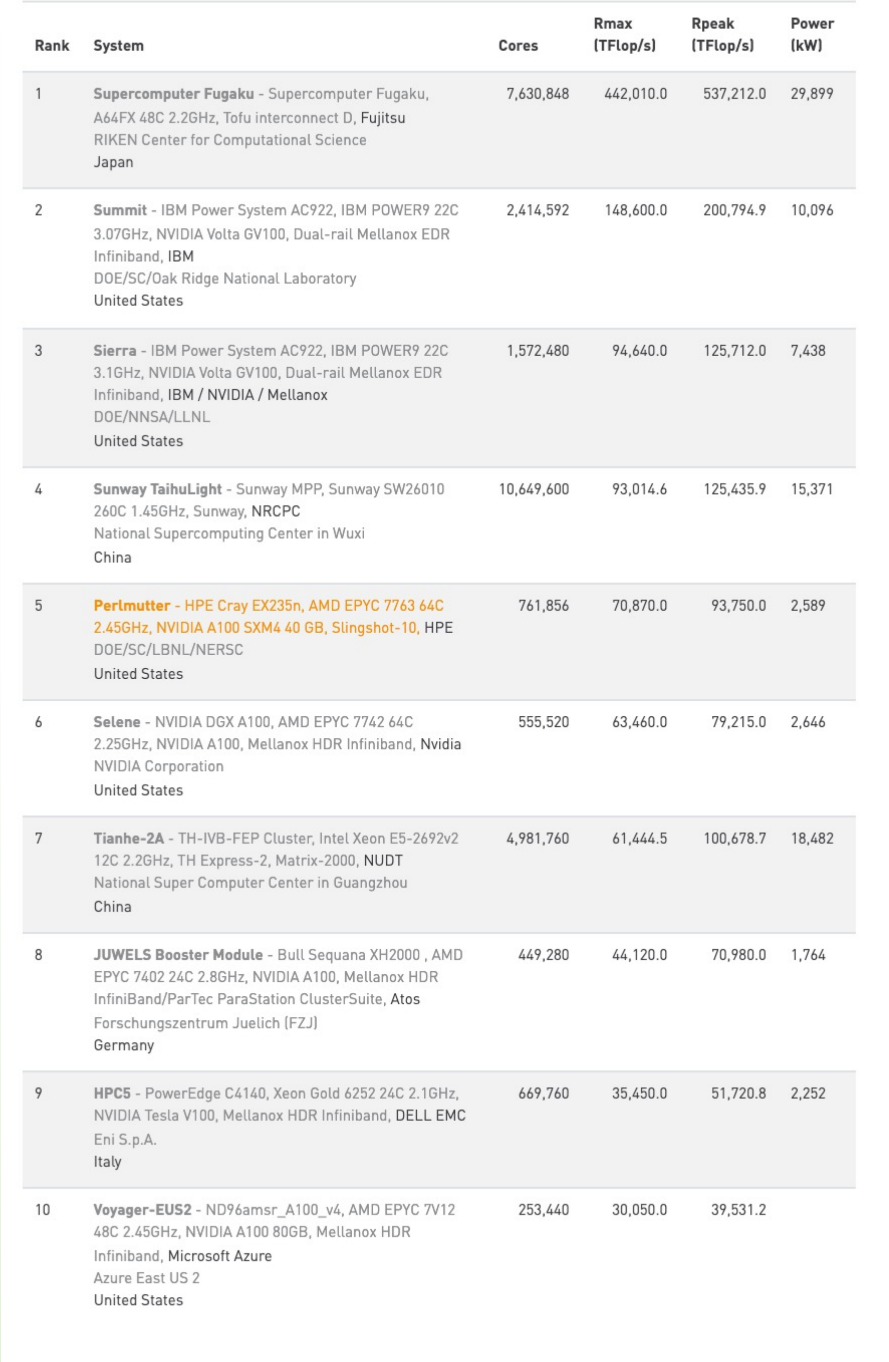
DR. MIGUEL ÁNGEL SERRANO BERRONES
JEFE DE SERVICIOS DE ENSEÑANZA E INVESTIGACIÓN

DR. SAMUEL FUENTES REYNA
SUBDIRECTOR DE REGULACIÓN Y ATENCIÓN HOSPITALARIA

DR. MACRO JIMÉNEZ PONCE
JEFE DEL DEPARTAMENTO DE INVESTIGACIÓN

Processing Genomic Data for Mexican Flu AHN1 Discovering

About Top500 List -2021



Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	**Perlmutter** - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	761,856	70,870.0	93,750.0	2,589
6	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63,460.0	79,215.0	2,646
7	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
8	**JUWELS Booster Module** - Bull Sequana XH2000 , AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR InfiniBand/ParTec ParaStation ClusterSuite, Atos Forschungszentrum Juelich (FZJ) Germany	449,280	44,120.0	70,980.0	1,764
9	**HPC5** - PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, DELL EMC Eni S.p.A. Italy	669,760	35,450.0	51,720.8	2,252
10	**Voyager-EUS2** - ND96amsr_A100_v4, AMD EPYC 7V12 48C 2.45GHz, NVIDIA A100 80GB, Mellanox HDR Infiniband, Microsoft Azure Azure East US 2 United States	253,440	30,050.0	39,531.2	

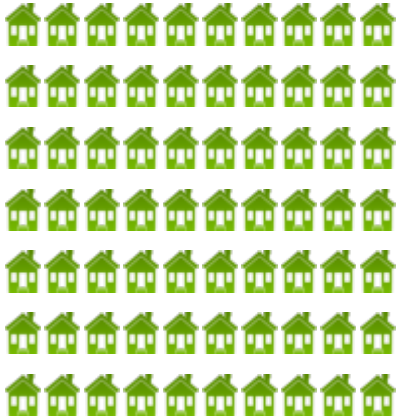
- 9/10 Powerful Machines are MPP Clusters
- 7/10 are Hybrid Machines with Accelerators
  - 5 NVIDIA GPU Technology
    - 3 Different Generations (Keppler, Pascal and Volta)
  - 2 Chinesse PU’s Technology
    - 1 Combines GPUs + MICs

Why Computing Perf/Watt Matters?

2.3 PFlops

7.0 Megawatts

7000 homes

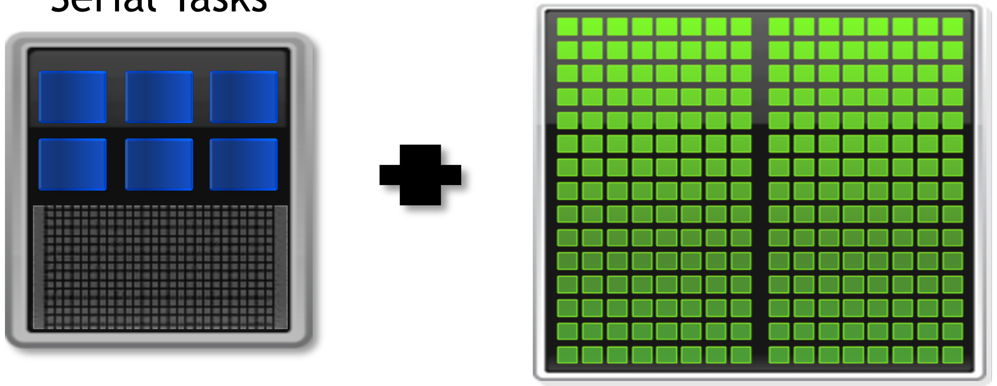


7.0 Megawatts

Traditional CPUs are not economically feasible

CPU
Optimized for
Serial Tasks

GPU Accelerator
Optimized for Many
Parallel Tasks

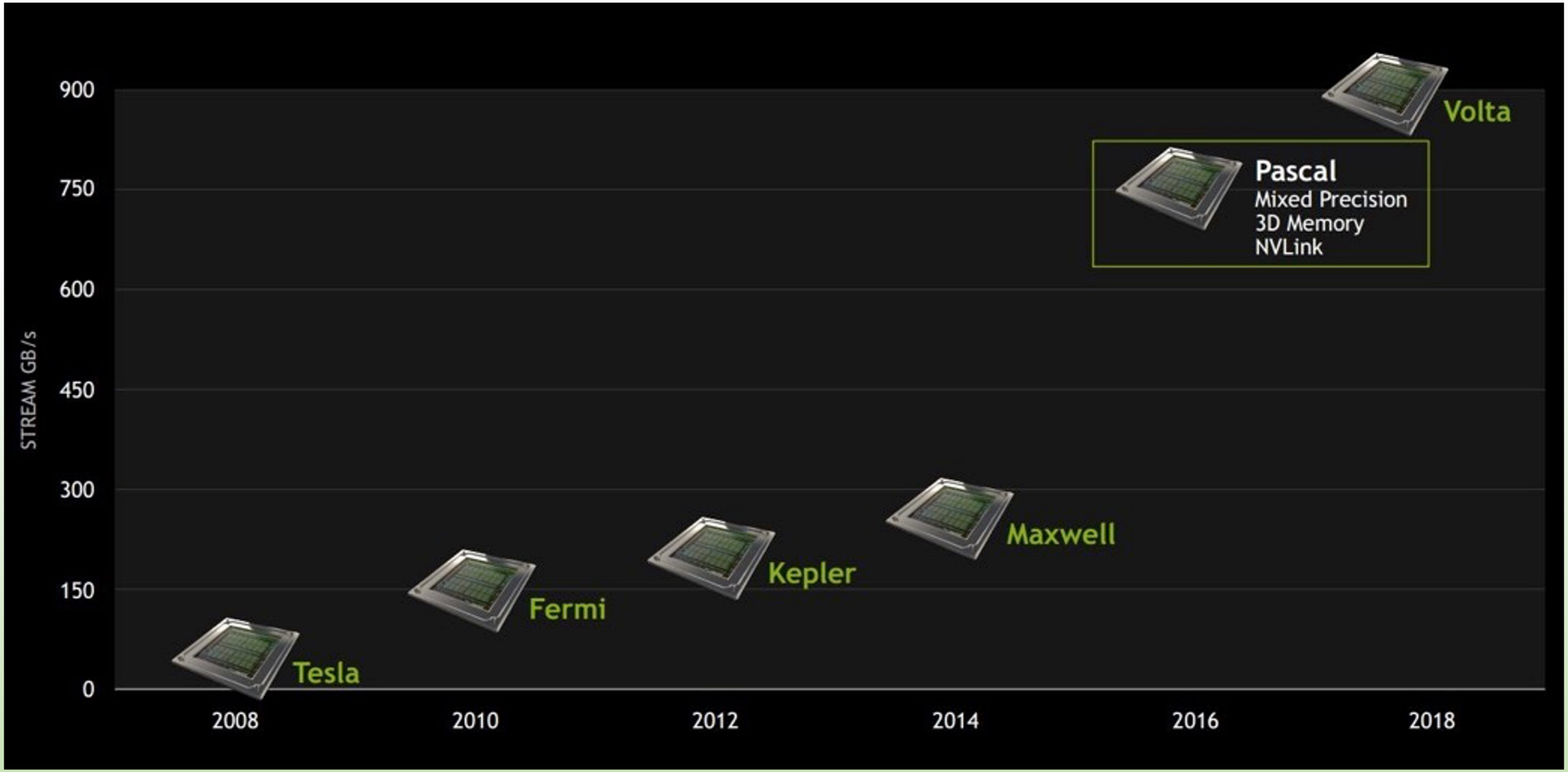


10x performance/socket

> 5x energy efficiency

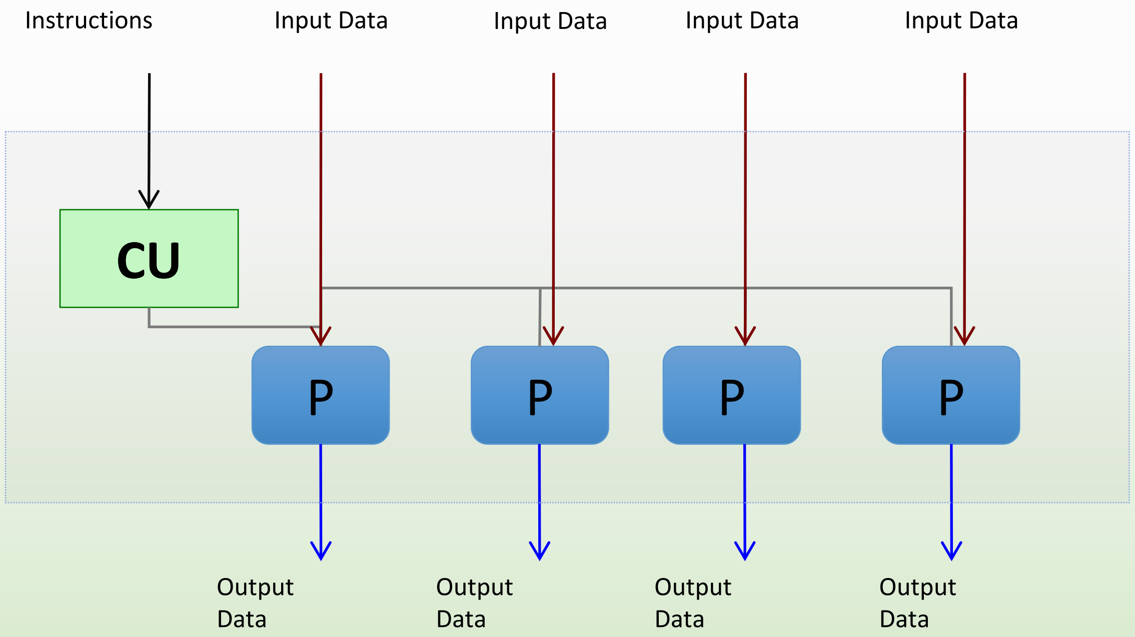
Era of GPU-accelerated computing is here

10 Years NVIDIA GPUs Development



Remember Architectural Systems Facts (From Flynn’s Taxonomy)

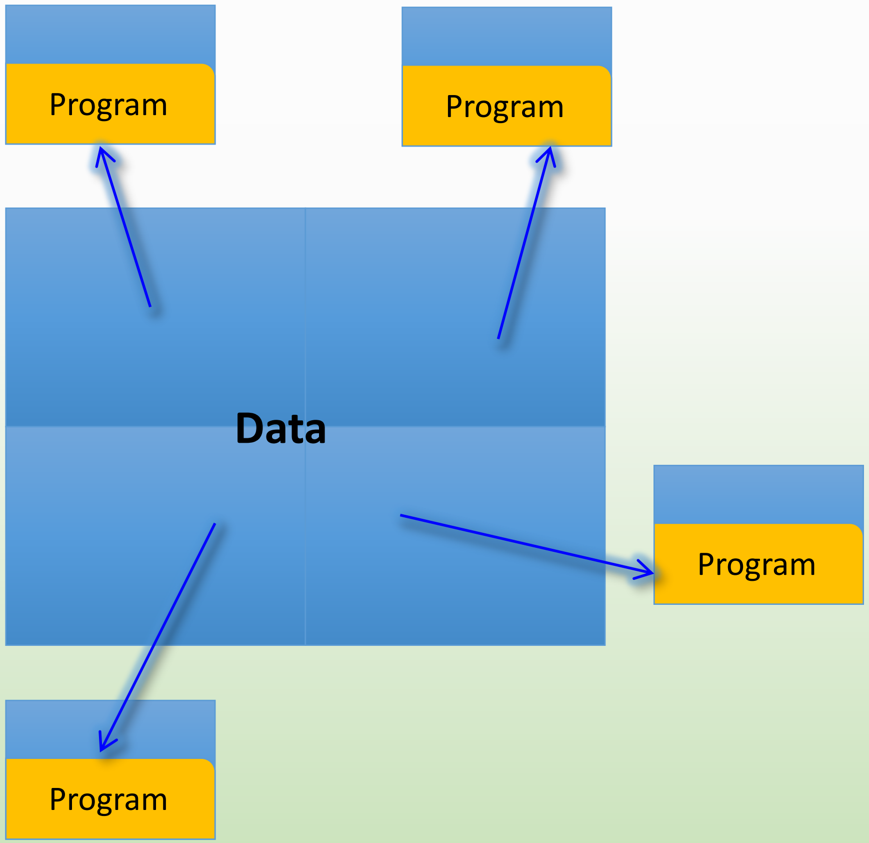
SPMD: Parallel Processing Units execute the same program on multiple parts of data



SIMD

+

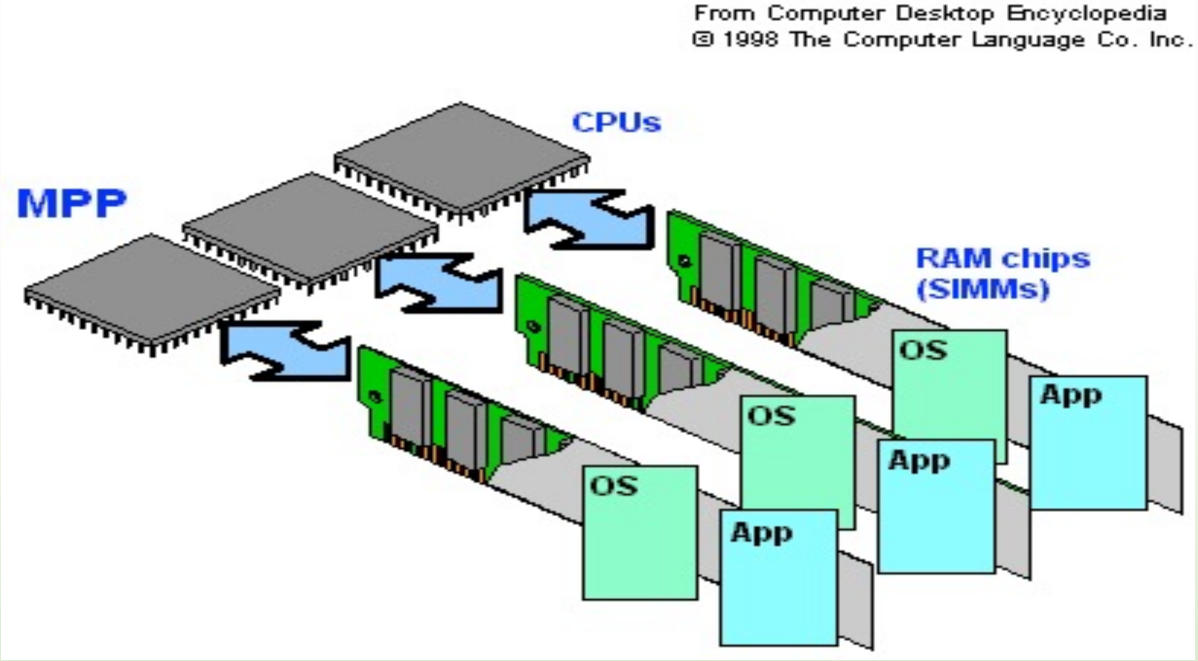
SIMD: All processors units are executing the same instructions in any instant.



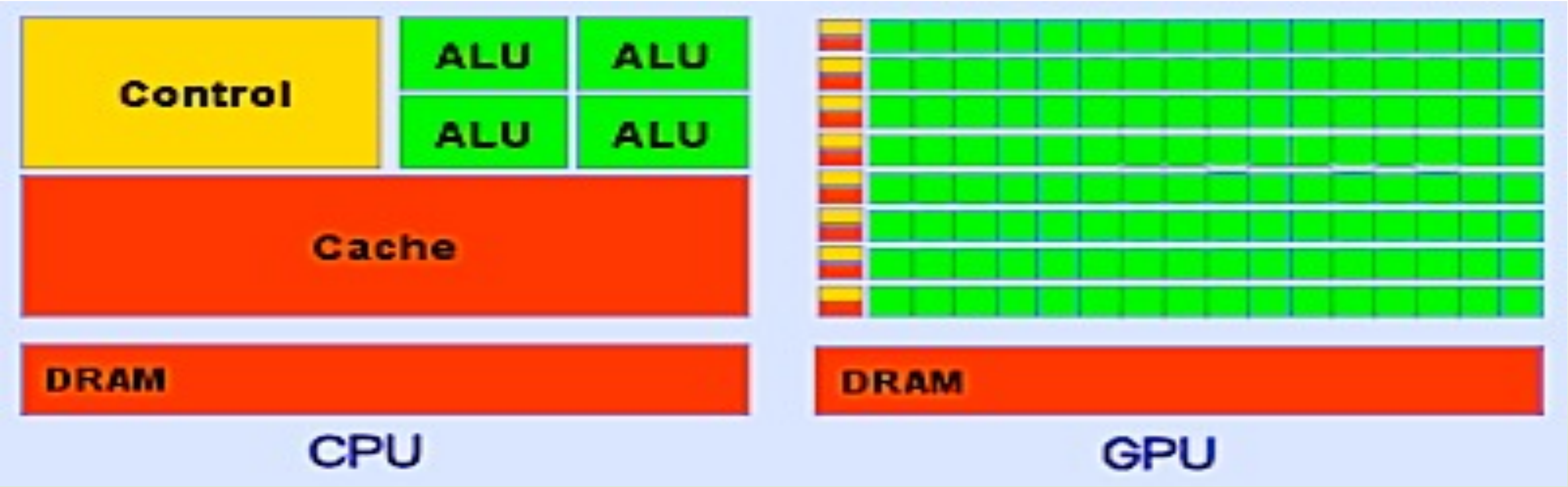
Processor

Massive Parallel Processing (MPP)

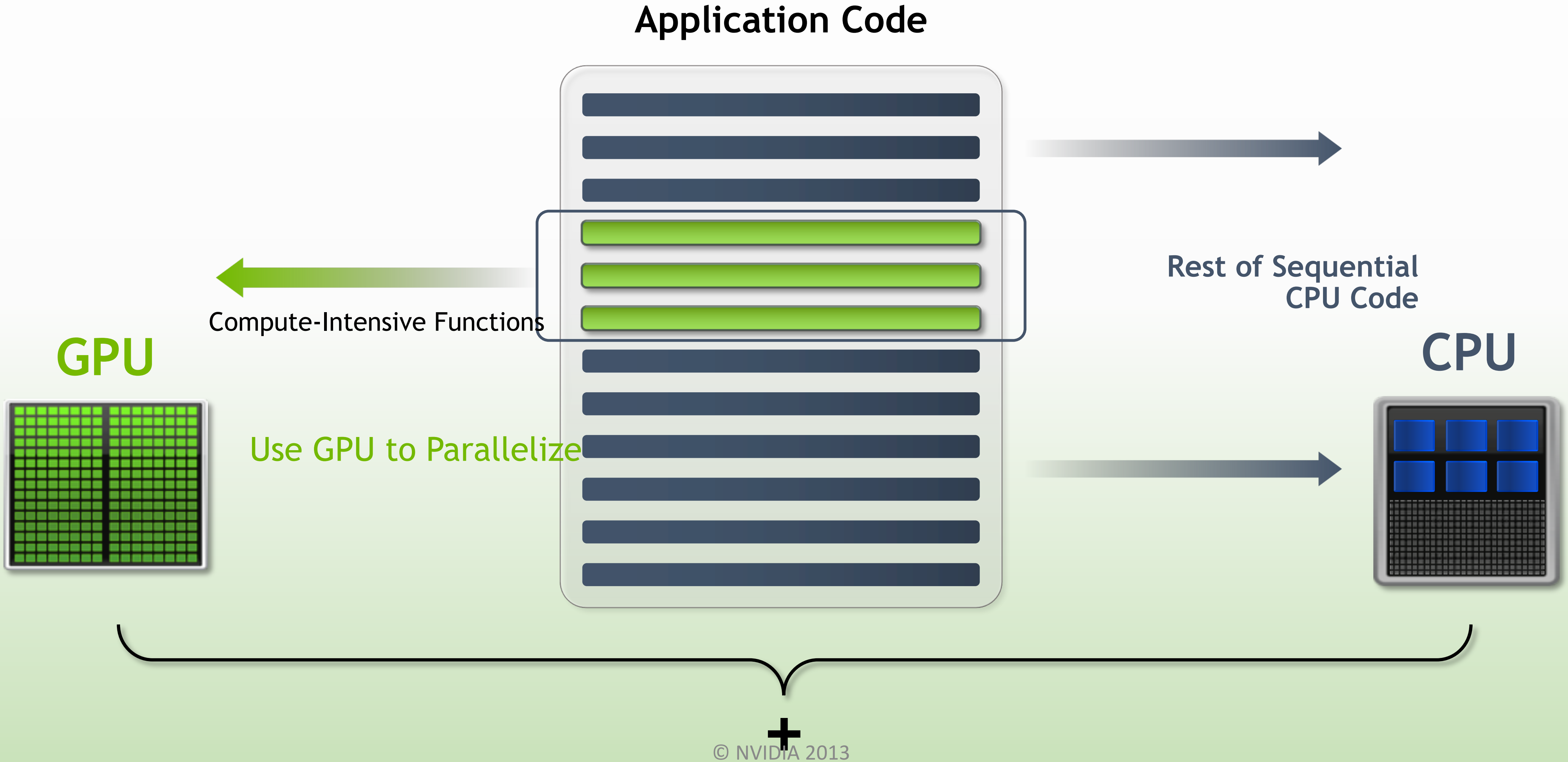
- Computer system with many independent arithmetic units or entire microprocessors, that run in parallel
- MPPA is a MIMD (Multiple Instruction streams, Multiple Data) architecture, with distributed memory accessed locally, not shared globally


From Computer Desktop Encyclopedia
© 1998 The Computer Language Co. Inc.

CPUs and GPUs Architecture

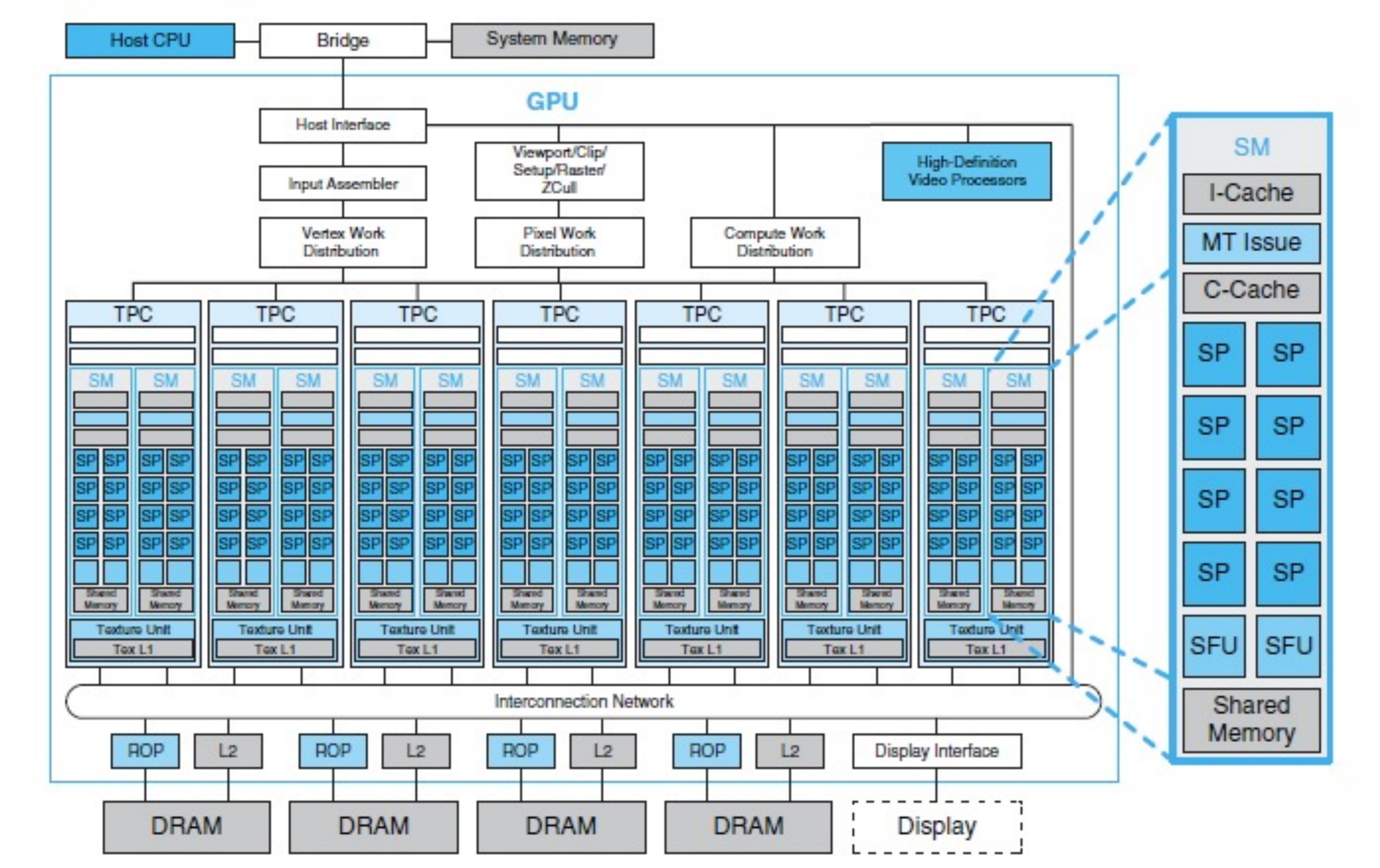


Small Changes, Big Speed-up



© NVIDIA 2013

NVIDIA TESLA® Architecture



NVIDIA TESLA™ Graphics and Computing Architecture Features

- TESLA™ shader processors are fully programmable
  - Large instructions memory
  - Cache Instructions
  - Logic Sequence Instructions
- TESLA™ to non-graphics programs:
  - Hierarchical Parallel Threads
  - Barrier Synchronization
  - Atomic Operators (Manage Highly Parallel Computing Work)

Heterogeneous Computing

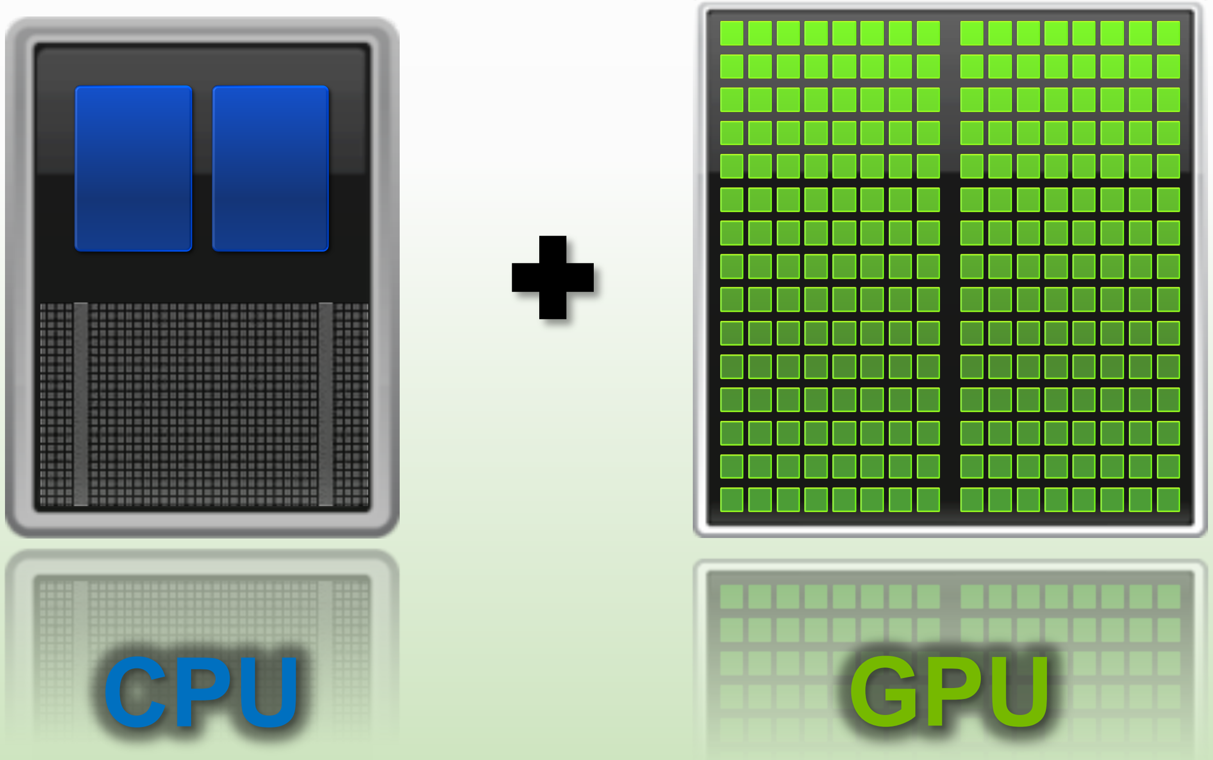
- Terminology:
  - *Host* The CPU and its memory (host memory)
  - *Device* The GPU and its memory (device memory)

Host

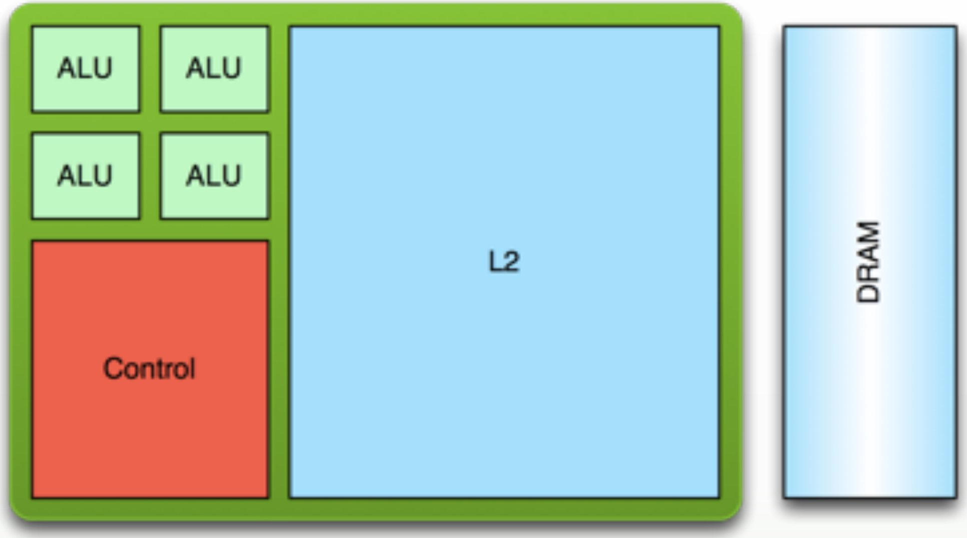
Device

GPGPU Accelerate Computing

Latency Processor + Throughput processor

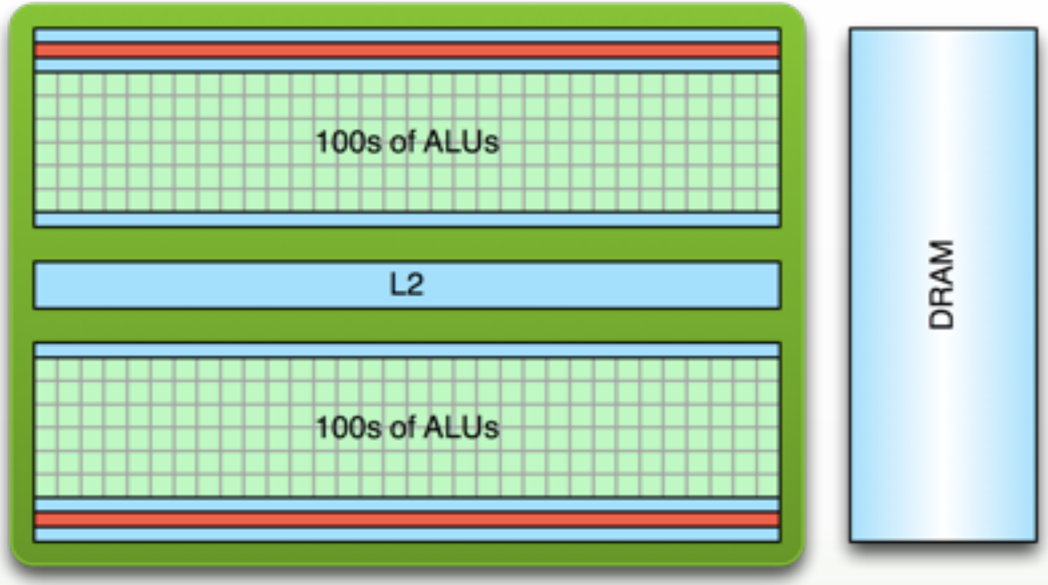


Low Latency or High Throughput?



CPU

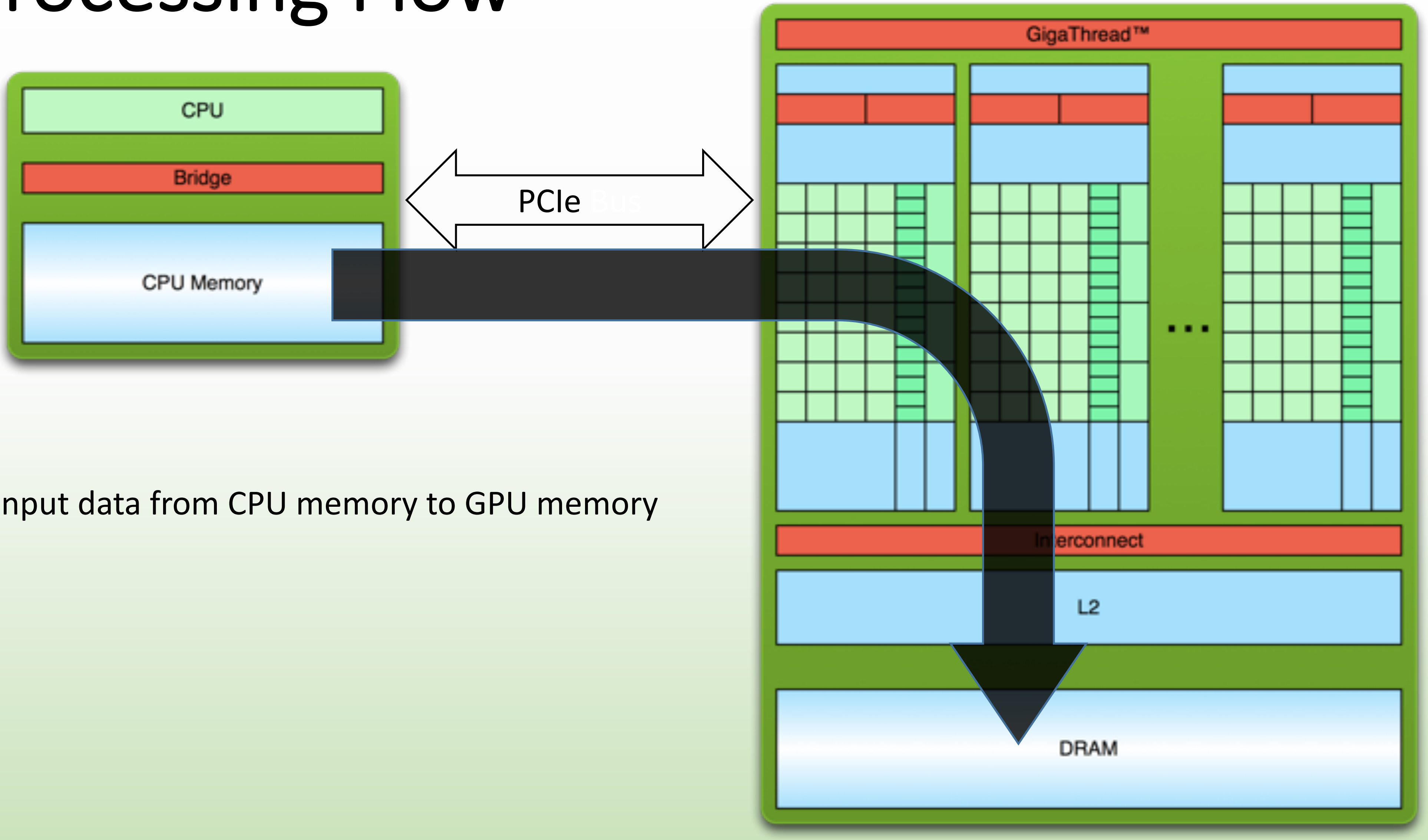
- Optimized for low-latency access to cached data sets
- Control logic for out-of-order and speculative execution



GPU

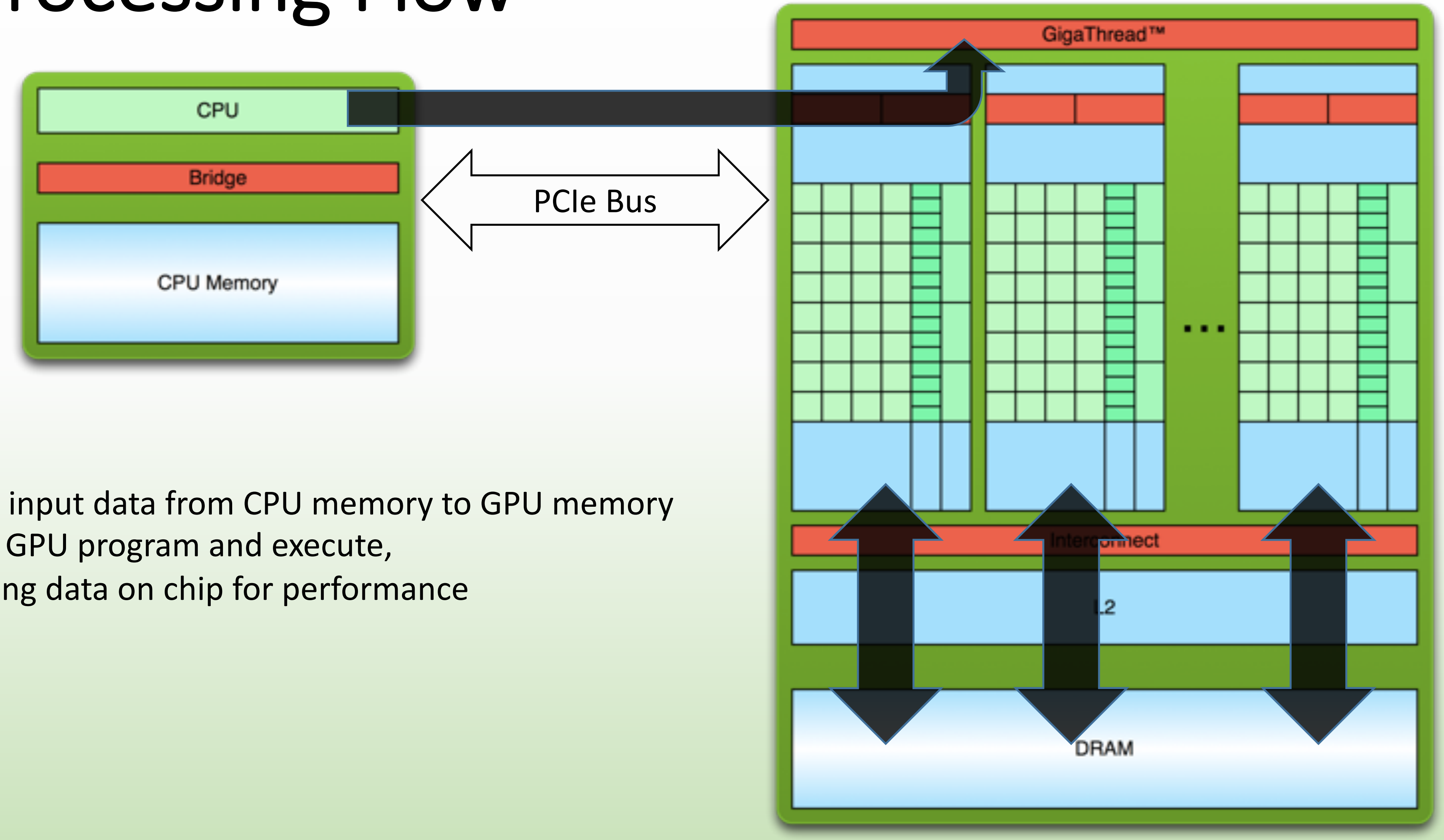
- Optimized for data-parallel, throughput computation
- Architecture tolerant of memory latency
- More transistors dedicated to computation

Processing Flow



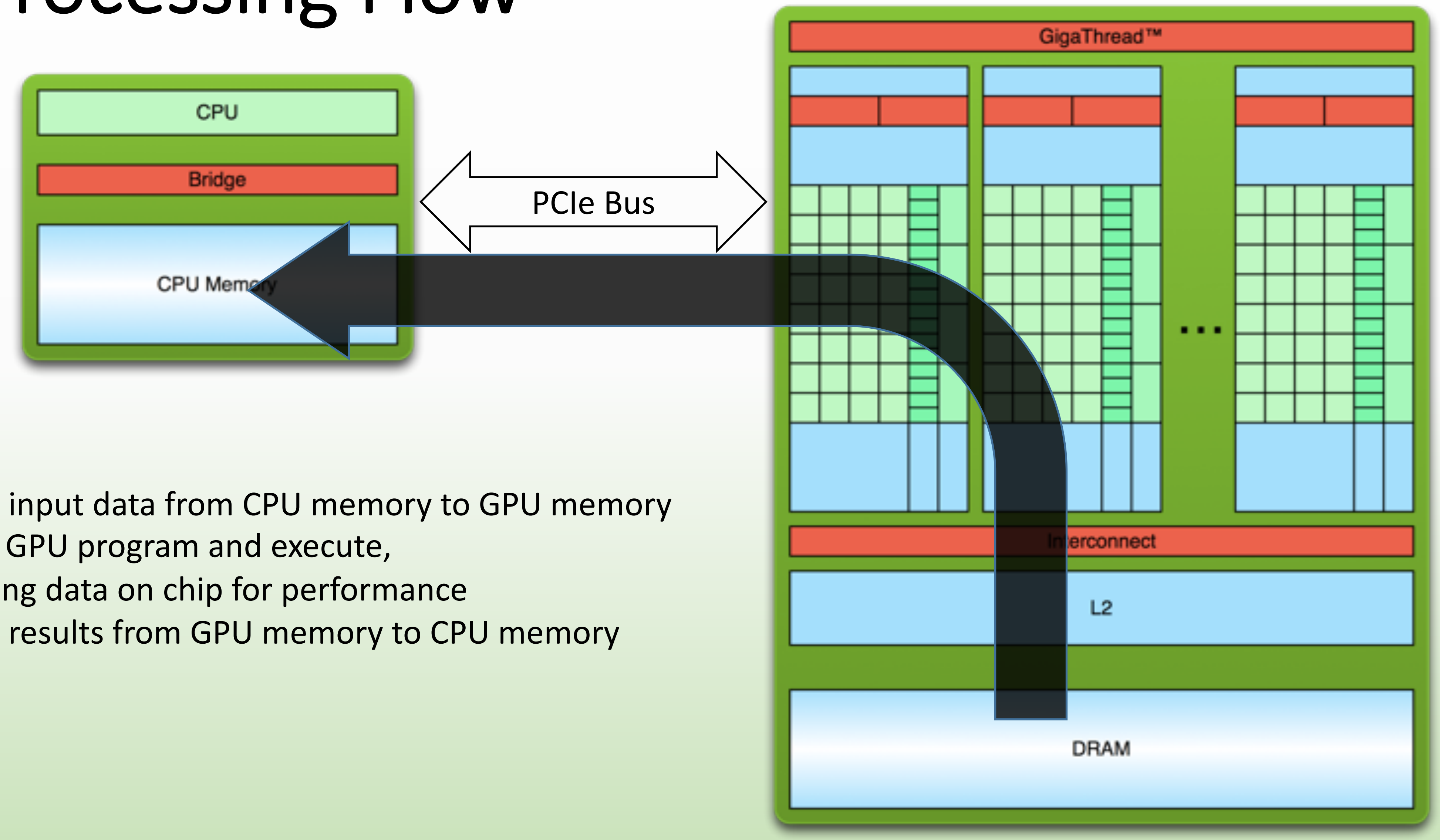
1. Copy input data from CPU memory to GPU memory

Processing Flow



1. Copy input data from CPU memory to GPU memory
2. Load GPU program and execute,
   caching data on chip for performance

Processing Flow



1. Copy input data from CPU memory to GPU memory
2. Load GPU program and execute,
   caching data on chip for performance
3. Copy results from GPU memory to CPU memory

CUDA Parallel Computing Platform

www.nvidia.com/getcuda

Programming Approaches

Libraries	OpenACC Directives	Programming Languages
"Drop-in" Acceleration	**Easily Accelerate Apps**	**Maximum Flexibility**

Development Environment

Nsight IDE
Linux, Mac and Windows
GPU Debugging and Profiling

CUDA-GDB debugger
Nsight Visual Profiler

Open Compiler Tool Chain

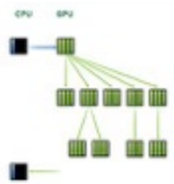
LLVM COMPILER INFRASTRUCTURE

Enables compiling new languages to CUDA platform, and CUDA languages to other architectures

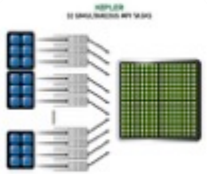
Hardware Capabilities

SMX

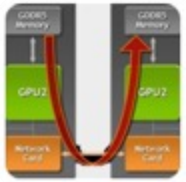
Dynamic Parallelism



HyperQ

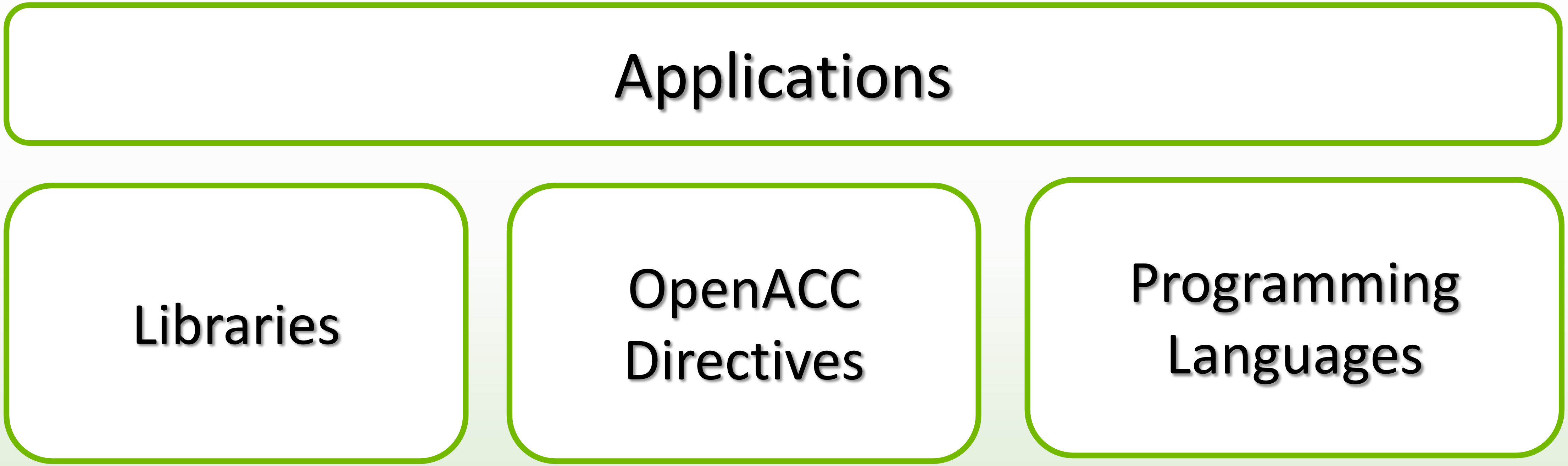


GPUDirect



© NVIDIA 2013

3 Ways to Accelerate Applications



Applications

Libraries

OpenACC Directives

Programming Languages

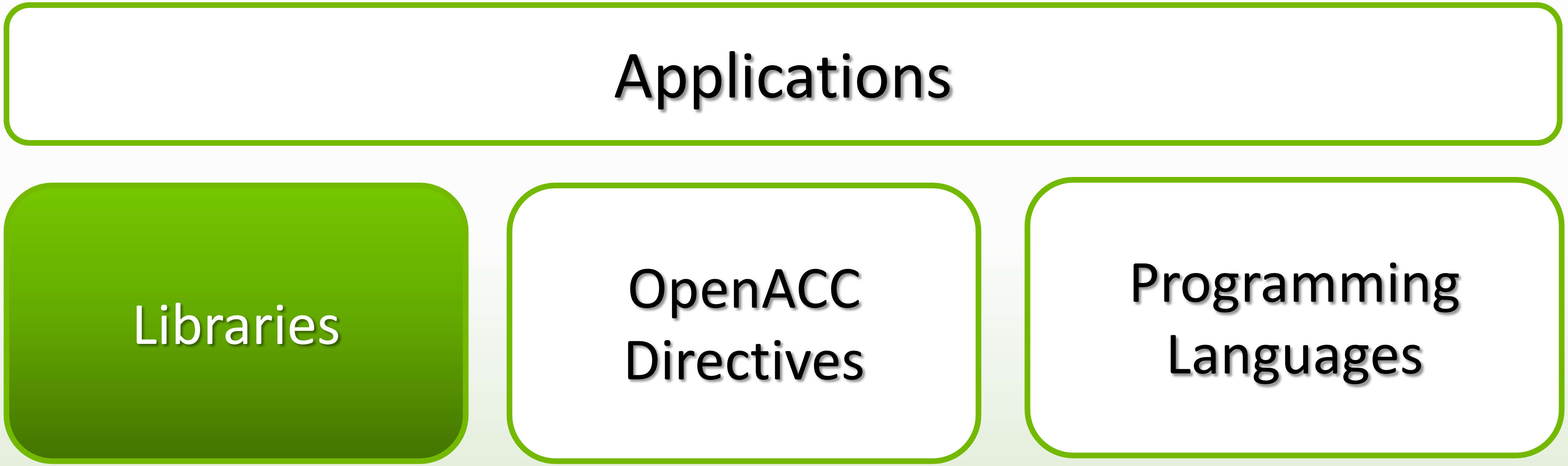
"Drop-in" Acceleration

Easily Accelerate Applications

Maximum Flexibility

© NVIDIA 2013

3 Ways to Accelerate Applications



Applications

Libraries

OpenACC Directives

Programming Languages

"Drop-in" Acceleration

Easily Accelerate Applications

Maximum Flexibility

© NVIDIA 2013

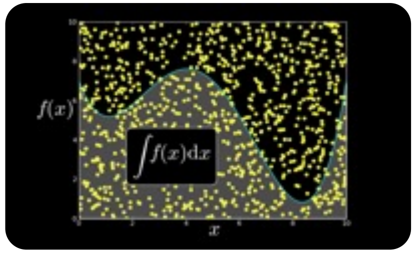
Libraries: Easy, High-Quality Acceleration

- **Ease of use:** Using libraries enables GPU acceleration without in-depth knowledge of GPU programming
- **“Drop-in”:** Many GPU-accelerated libraries follow standard APIs, thus enabling acceleration with minimal code changes
- **Quality:** Libraries offer high-quality implementations of functions encountered in a broad range of applications
- **Performance:** NVIDIA libraries are tuned by experts

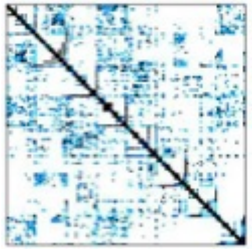
© NVIDIA 2013

Some GPU-accelerated Libraries

NVIDIA cuBLAS



NVIDIA cuRAND



NVIDIA cuSPARSE

NVIDIA NPP

GPU VSIPL

Vector Signal
Image Processing

CULA | tools

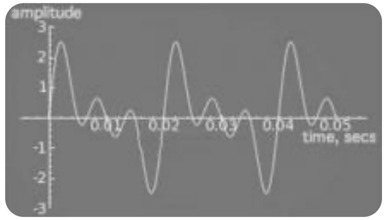
GPU Accelerated
Linear Algebra

MAGMA

ICL UT

Matrix Algebra on GPU
and Multicore

open source initiative



NVIDIA cuFFT

ROGUE WAVE
SOFTWARE

IMSL Library

ArrayFire Matrix
Computations

CUSP

Sparse Linear
Algebra

open source initiative

Thrust

C++ STL Features
for CUDA

open source initiative

© NVIDIA 2013

3 Steps to CUDA-accelerated application

- **Step 1:** Substitute library calls with equivalent CUDA library calls

```
saxpy ( … )                cublasSaxpy ( … )
```

- **Step 2:** Manage data locality

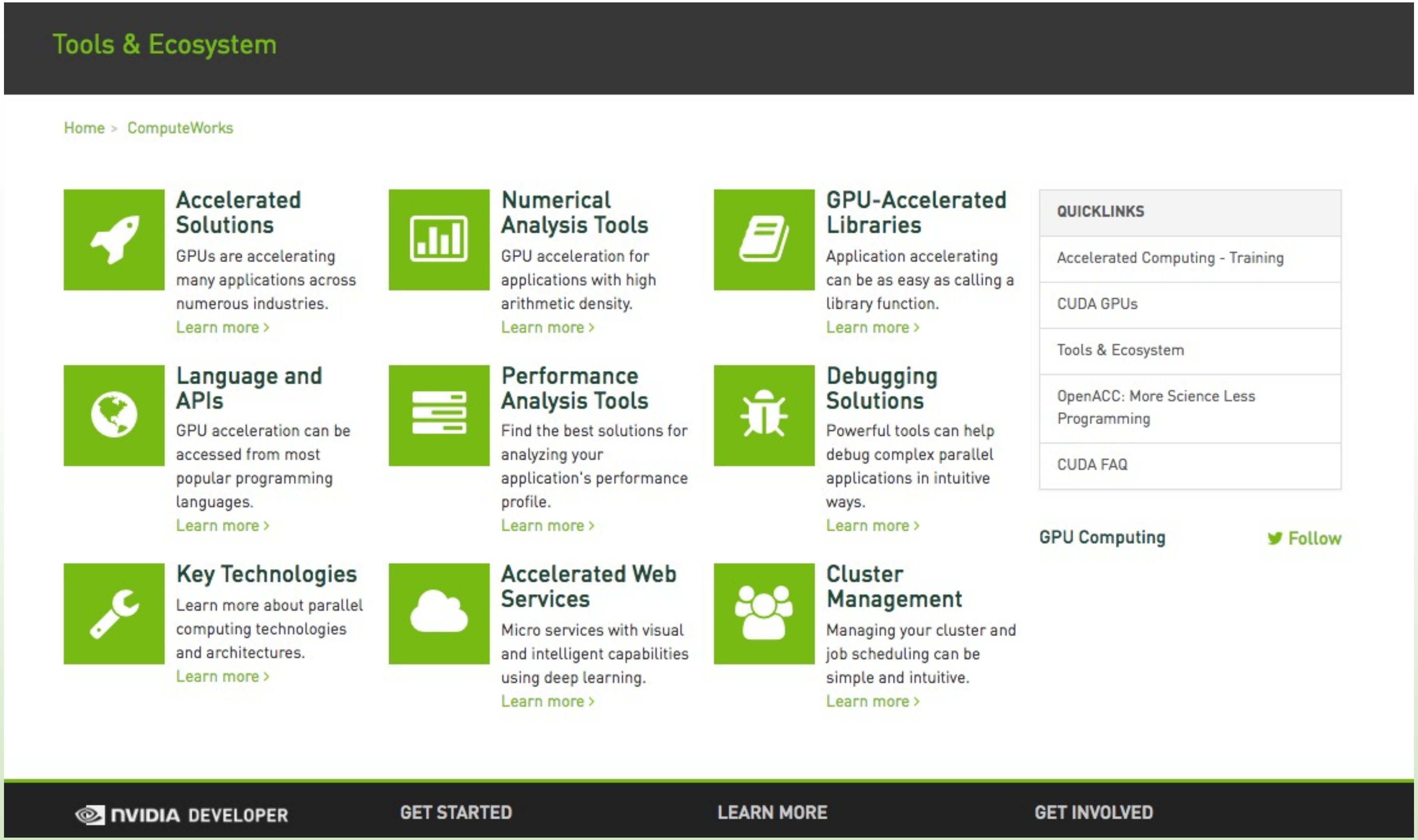
  - with CUDA: `cudaMalloc(), cudaMemcpy(), etc.`
  - with CUBLAS: `cublasAlloc(), cublasSetVector(), etc.`

- **Step 3:** Rebuild and link the CUDA-accelerated library

```
nvcc myobj.o –l cublas
```

© NVIDIA 2013

Explore the CUDA (Libraries) Ecosystem



Tools & Ecosystem

Home > ComputeWorks

Accelerated Solutions
GPUs are accelerating many applications across numerous industries.
Learn more >

Numerical Analysis Tools
GPU acceleration for applications with high arithmetic density.
Learn more >

GPU-Accelerated Libraries
Application accelerating can be as easy as calling a library function.
Learn more >

Language and APIs
GPU acceleration can be accessed from most popular programming languages.
Learn more >

Performance Analysis Tools
Find the best solutions for analyzing your application's performance profile.
Learn more >

Debugging Solutions
Powerful tools can help debug complex parallel applications in intuitive ways.
Learn more >

Key Technologies
Learn more about parallel computing technologies and architectures.
Learn more >

Accelerated Web Services
Micro services with visual and intelligent capabilities using deep learning.
Learn more >

Cluster Management
Managing your cluster and job scheduling can be simple and intuitive.
Learn more >

QUICKLINKS

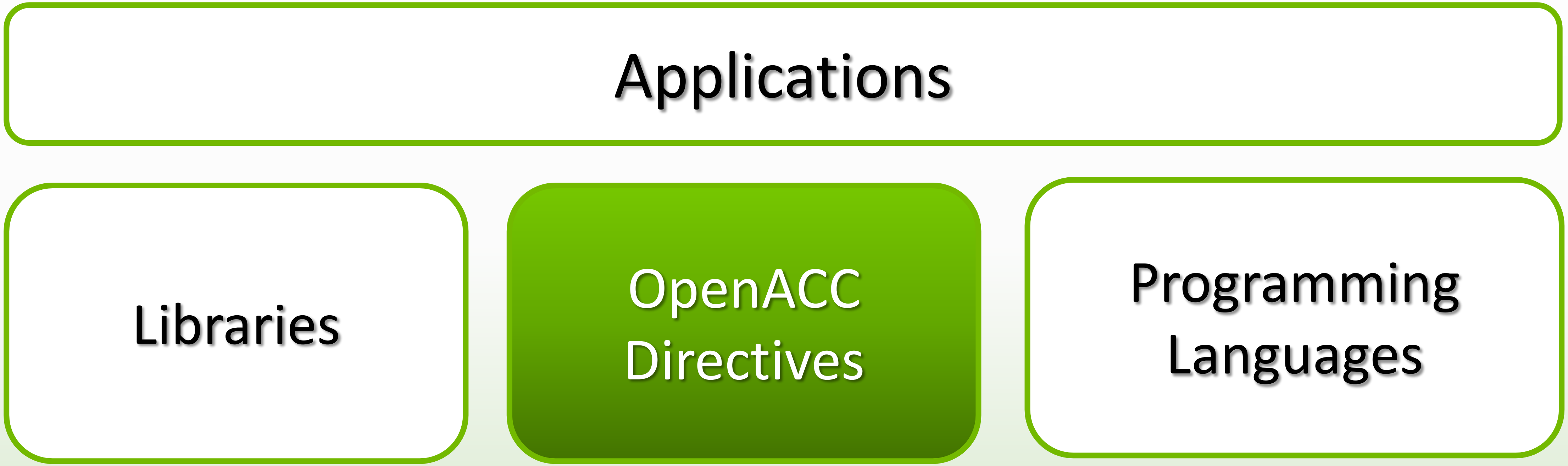
- Accelerated Computing - Training
- CUDA GPUs
- Tools & Ecosystem
- OpenACC: More Science Less Programming
- CUDA FAQ

GPU Computing Follow

NVIDIA DEVELOPER GET STARTED LEARN MORE GET INVOLVED

- CUDA Tools and Ecosystem described in detail on NVIDIA Developer Zone: developer.nvidia.com/cuda-tools-ecosystem

3 Ways to Accelerate Applications



Applications

Libraries

OpenACC Directives

Programming Languages

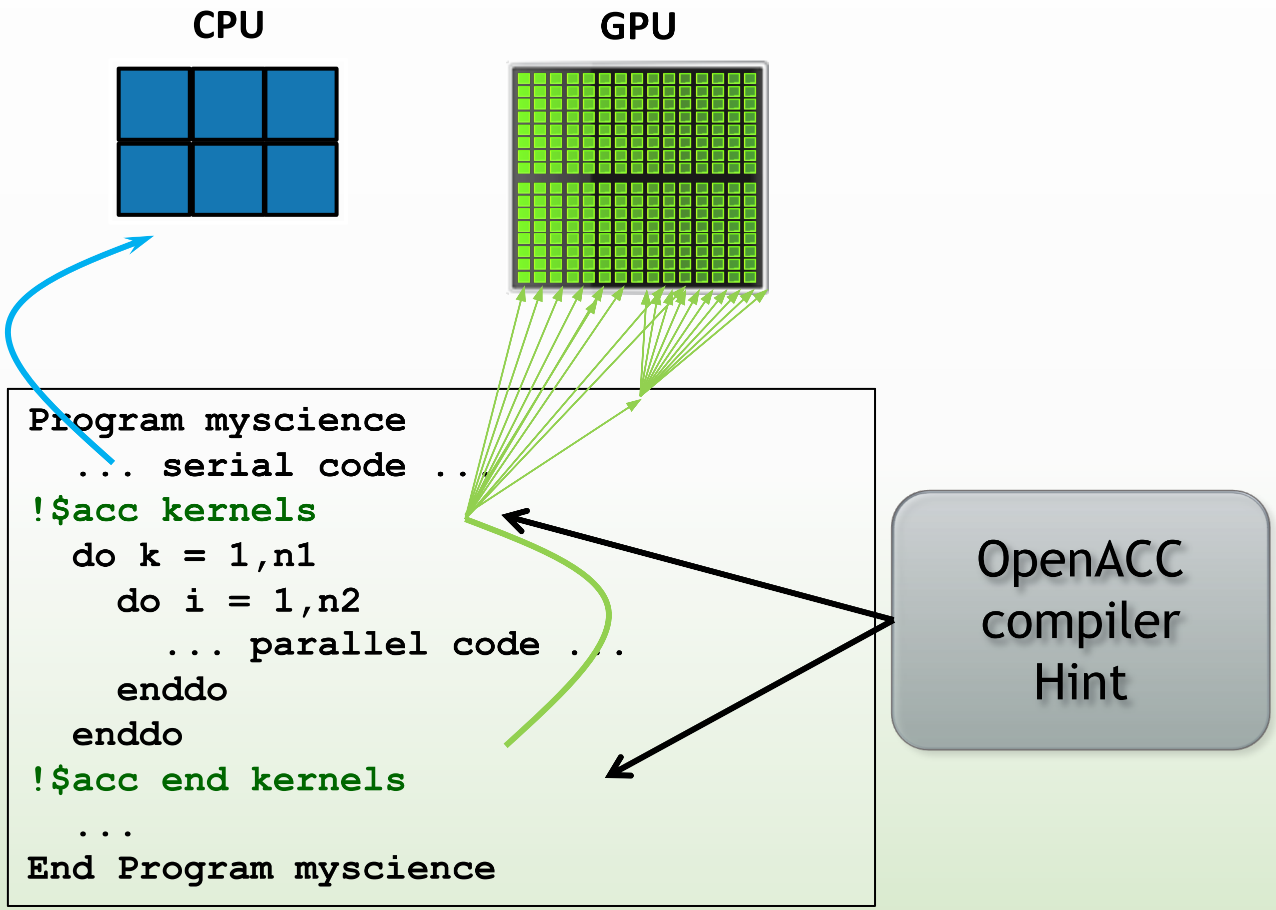
“Drop-in” Acceleration

Easily Accelerate Applications

Maximum Flexibility

© NVIDIA 2013

OpenACC Directives



```
Program myscience
  ... serial code ...
!$acc kernels
  do k = 1,n1
    do i = 1,n2
      ... parallel code ...
    enddo
  enddo
!$acc end kernels
  ...
End Program myscience
```

Your original
Fortran or C code

Simple Compiler hints

Compiler Parallelizes code

Works on many-core GPUs &
multicore CPUs

© NVIDIA 2013

OpenACC

The Standard for GPU Directives

OpenACC

DIRECTIVES FOR ACCELERATORS

- **Easy:** Directives are the easy path to accelerate compute intensive applications

- **Open:** OpenACC is an open GPU directives standard, making GPU programming straightforward and portable across parallel and multi-core processors

- **Powerful:** GPU Directives allow complete access to the massive parallel power of a GPU

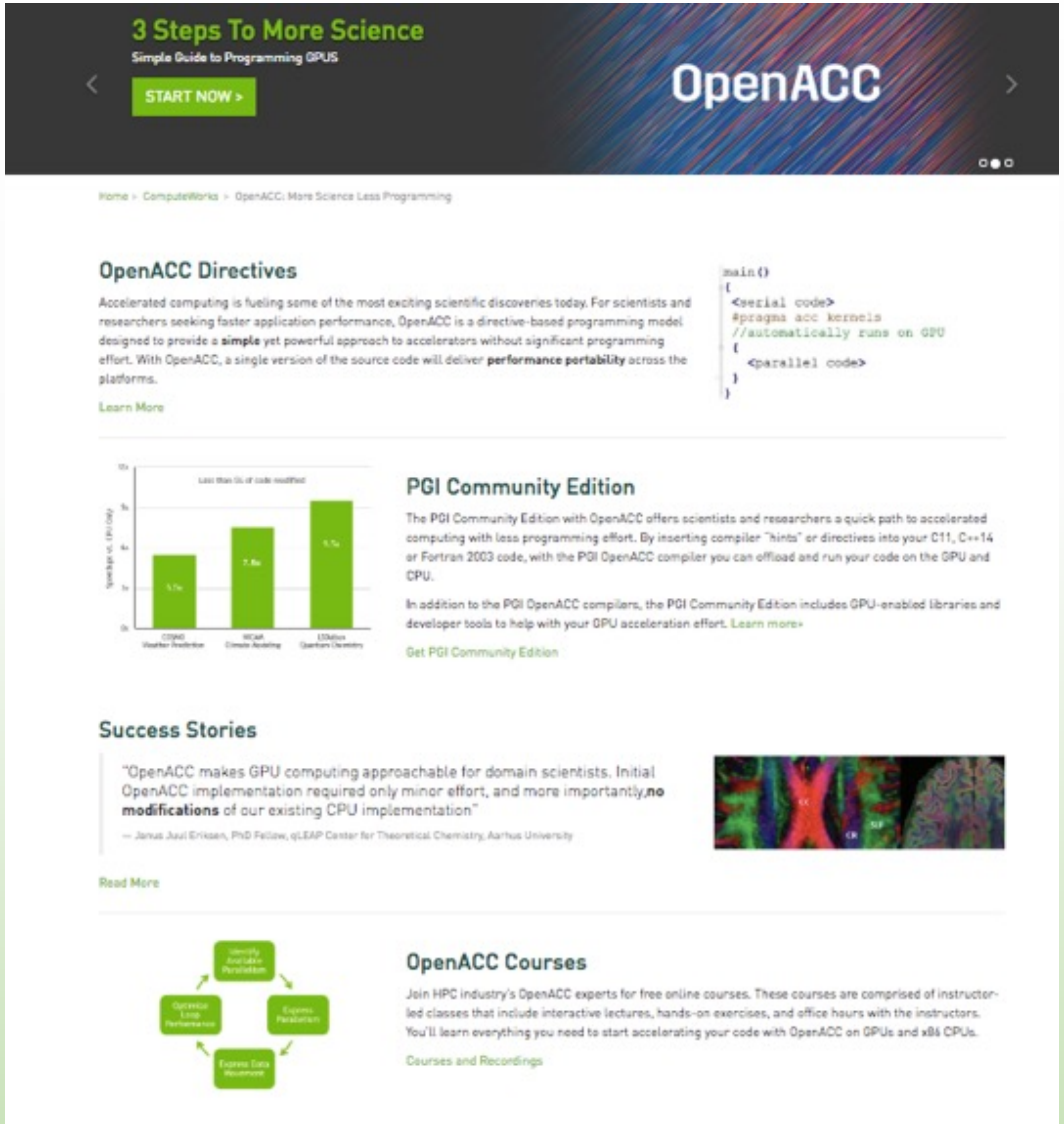
© NVIDIA 2013

Start Now with OpenACC Directives

Free trial license to PGI Accelerator

Tools for quick ramp

https://developer.nvidia.com/openacc



3 Steps To More Science

Simple Guide to Programming GPUS

START NOW >

OpenACC

Home > ComputeWorks > OpenACC: More Science Less Programming

OpenACC Directives

Accelerated computing is fueling some of the most exciting scientific discoveries today. For scientists and researchers seeking faster application performance, OpenACC is a directive-based programming model designed to provide a **simple** yet powerful approach to accelerators without significant programming effort. With OpenACC, a single version of the source code will deliver **performance portability** across the platforms.

Learn More

```
main()
{
 <serial code>
 #pragma acc kernels
 //automatically runs on GPU
 {
   <parallel code>
 }
}
```

Less than 5% of code modified
Speedup vs. CPU Only
COSMO Weather Prediction
NICAM Climate Modeling
LSDalton Quantum Chemistry
Identify Available Parallelism
Express Parallelism
Express Data Movement
Optimize Loop Performance

PGI Community Edition

The PGI Community Edition with OpenACC offers scientists and researchers a quick path to accelerated computing with less programming effort. By inserting compiler "hints" or directives into your C11, C++14 or Fortran 2003 code, with the PGI OpenACC compiler you can offload and run your code on the GPU and CPU.

In addition to the PGI OpenACC compilers, the PGI Community Edition includes GPU-enabled libraries and developer tools to help with your GPU acceleration effort. Learn more›

Get PGI Community Edition

Success Stories

> "OpenACC makes GPU computing approachable for domain scientists. Initial OpenACC implementation required only minor effort, and more importantly,**no modifications** of our existing CPU implementation"
>
> — Janus Juul Eriksen, PhD Fellow, qLEAP Center for Theoretical Chemistry, Aarhus University

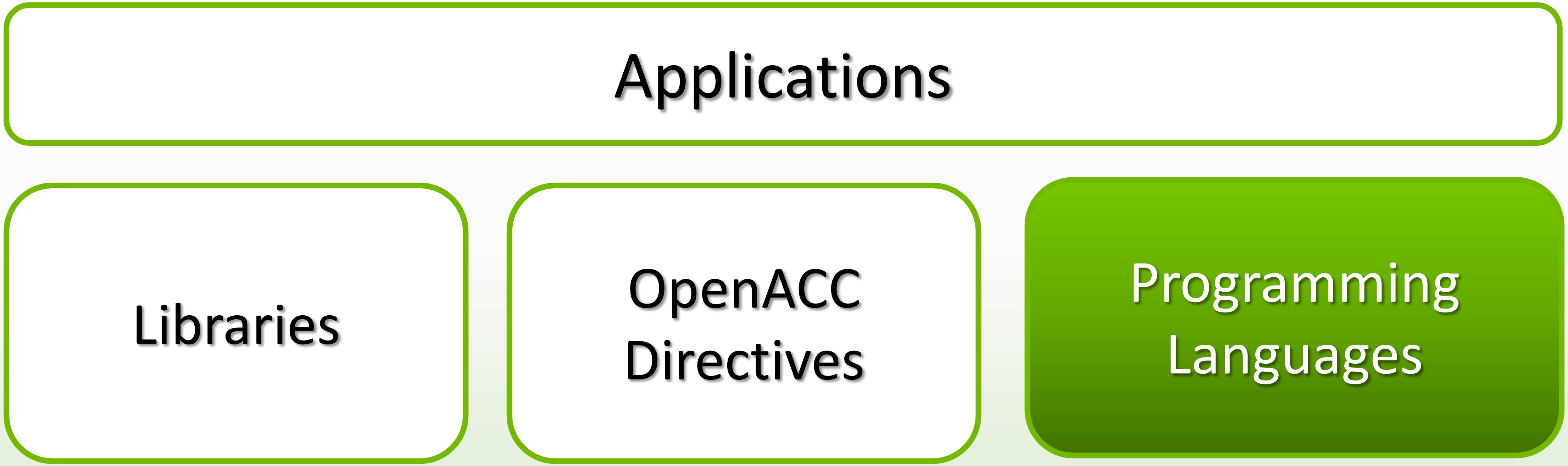
Read More

OpenACC Courses

Join HPC industry's OpenACC experts for free online courses. These courses are comprised of instructor-led classes that include interactive lectures, hands-on exercises, and office hours with the instructors. You'll learn everything you need to start accelerating your code with OpenACC on GPUs and x86 CPUs.

Courses and Recordings

3 Ways to Accelerate Applications



Applications

Libraries

OpenACC Directives

Programming Languages

"Drop-in" Acceleration

Easily Accelerate Applications

Maximum Flexibility + Best Performance

© NVIDIA 2013

GPU Programming Languages

Numerical analytics	▶	MATLAB, Mathematica, LabVIEW
Fortran	▶	OpenACC, CUDA Fortran
C	▶	OpenACC, CUDA C
C++	▶	Thrust, CUDA C++
Python	▶	PyCUDA, Copperhead
F#	▶	Alea.cuBase

© NVIDIA 2013

Learn More

These languages are supported on all CUDA-capable GPUs.

You might already have a CUDA-capable GPU in your laptop or desktop PC!

CUDA C/C++
http://developer.nvidia.com/cuda-toolkit

Thrust C++ Template Library
http://developer.nvidia.com/thrust

CUDA Fortran
http://developer.nvidia.com/cuda-toolkit

PyCUDA (Python)
http://mathema.tician.de/software/pycuda

GPU.ORG Different Resources
http://gpgpu.org

MATLAB
http://www.mathworks.com/discovery/matlab-gpu.html

Mathematica
http://www.wolfram.com/mathematica/new-in-8/cuda-and-opencl-support/ or
http://www.wolfram.com/gridmathematica/

Getting Started

- Download CUDA Toolkit & SDK: https://developer.nvidia.com/cuda-downloads
- Nsight IDE (Eclipse or Visual Studio): http://www.nvidia.com/object/nsight.html

- General GPU Computing Community: http://gpgpu.org/

- Programming Guide/Best Practices:
  - docs.nvidia.com

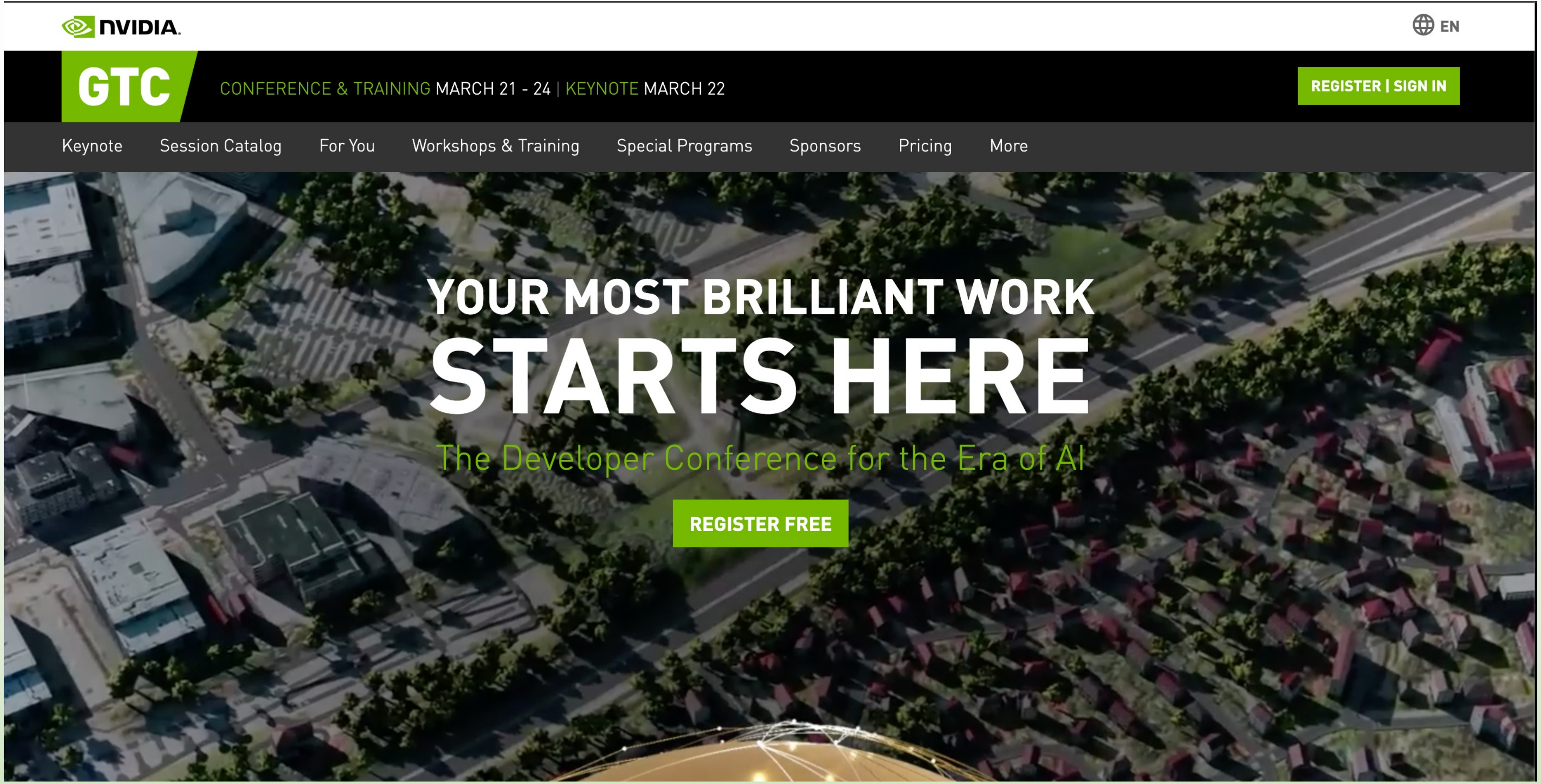
- Questions:
  - NVIDIA Developer forums: devtalk.nvidia.com
  - Search or ask on: www.stackoverflow.com/tags/cuda

- Developer Community: https://developer.nvidia.com/ (Join Now!)

Thank you!
@carlosjaimebh

NVIDIA DEEP LEARNING INSTITUTE

www.nvidia.com/dli



CONFERENCE & TRAINING MARCH 21 - 24 | KEYNOTE MARCH 22

YOUR MOST BRILLIANT WORK

STARTS HERE

The Developer Conference for the Era of AI

https://www.nvidia.com/gtc