nVISION 08

THE WORLD OF VISUAL COMPUTING

High Performance Computing

David B. Kirk, Chief Scientist, NVIDIA

NVIDIA

A History of Innovation

- Invented the Graphics Processing Unit (GPU)
- Pioneered programmable shading
- Over 2000 patents*

1995	1999	2002	2003	2004	2005	2006-2007	2008
NV1	GeForce 256	GeForce4	GeForce FX	GeForce 6	GeForce 7	GeForce 8	GeForce GTX 200
1 Million Transistors	22 Million Transistors	63 Million Transistors	130 Million Transistors	222 Million Transistors	302 Million Transistors	754 Million Transistors	1.4 Billion Transistors

nvision 08
THE WORLD OF VISUAL COMPUTING

Real-time Ray Tracing Demo

- Real system
- NVSG-driven animation and interaction
- Programmable shading
- Modeled in Maya, imported through COLLADA
- Fully ray traced

2 million polygons
Bump-mapping
Movable light source
5 bounce reflection/refraction
Adaptive antialiasing

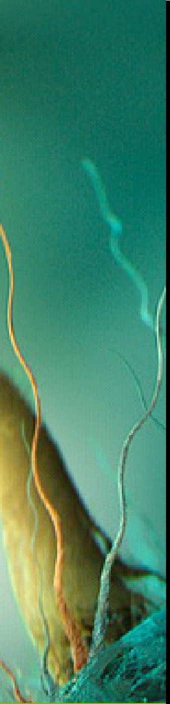
NVIDIA


Poznaj świadectwa z dokumentacji procesu
beatyfikacyjnego i kanonizacyjnego Jana Pawła II
CUDA
już od 7 marca
CUD
Książka wraz z filmem VCD „Cud” dostępna w księgarniach,
parafiach, salonach EMPiK, w księgarniach internetowych
i na stronie www.stanislawbm.pl tel. 012 429 52 17

CUDA^{TM} Uses Kernels and Threads for Fast Parallel Execution

- Parallel portions of an application are executed on the GPU as kernels
  - One kernel is executed at a time
  - Many threads execute each kernel

- Differences between CUDA and CPU threads
  - CUDA threads are extremely lightweight
    - Very little creation overhead
    - Instant switching
  - CUDA uses 1000s of threads to achieve efficiency



Simple "C" Description For Parallelism

```
void saxpy_serial(int n, float a, float *x, float *y)
{
    for (int i = 0; i < n; ++i)
        y[i] = a*x[i] + y[i];
}
// Invoke serial SAXPY kernel
saxpy_serial(n, 2.0, x, y);
```

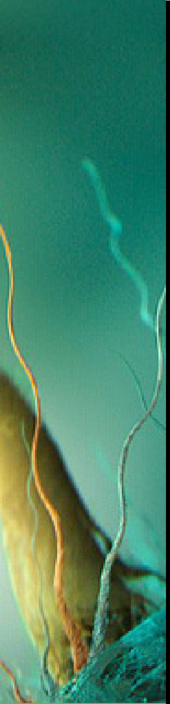
Standard C Code

```
__global__ void saxpy_parallel(int n, float a, float *x, float *y)
{
    int i = blockIdx.x*blockDim.x + threadIdx.x;
    if (i < n)  y[i] = a*x[i] + y[i];
}
// Invoke parallel SAXPY kernel with 256 threads/block
int nblocks = (n + 255) / 256;
saxpy_parallel<<<nblocks, 256>>>(n, 2.0, x, y);
```

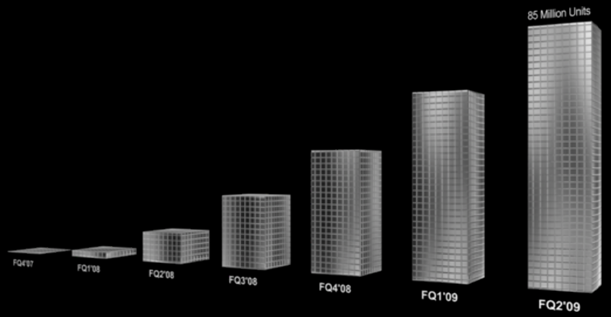
Parallel C Code

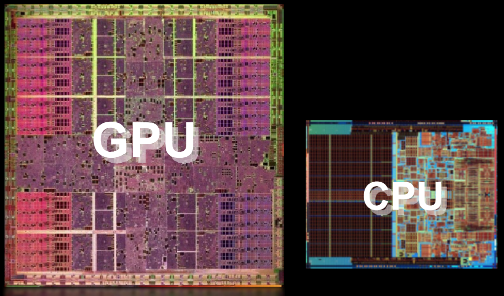
The Key to Computing on the GPU

- Standard high level language support
  - C, soon C++ and Fortran
  - Standard and domain specific libraries
- Hardware Thread Management
  - No switching overhead
  - Hide instruction and memory latency
- Shared memory
  - User-managed data cache
  - Thread communication / cooperation within blocks
- Runtime and tool support
  - Loader, Memory Allocation
  - C stdlib



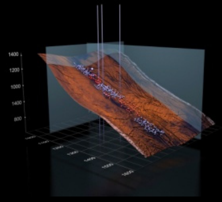
nvision 08
THE WORLD OF VISUAL COMPUTING

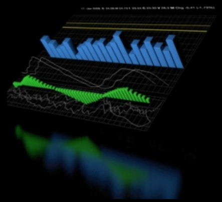


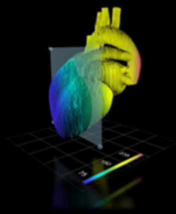


Heterogeneous Computing

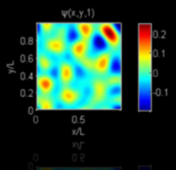
CUDA


Oil & Gas

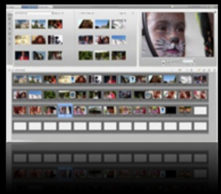

Finance


Medical


Biophysics

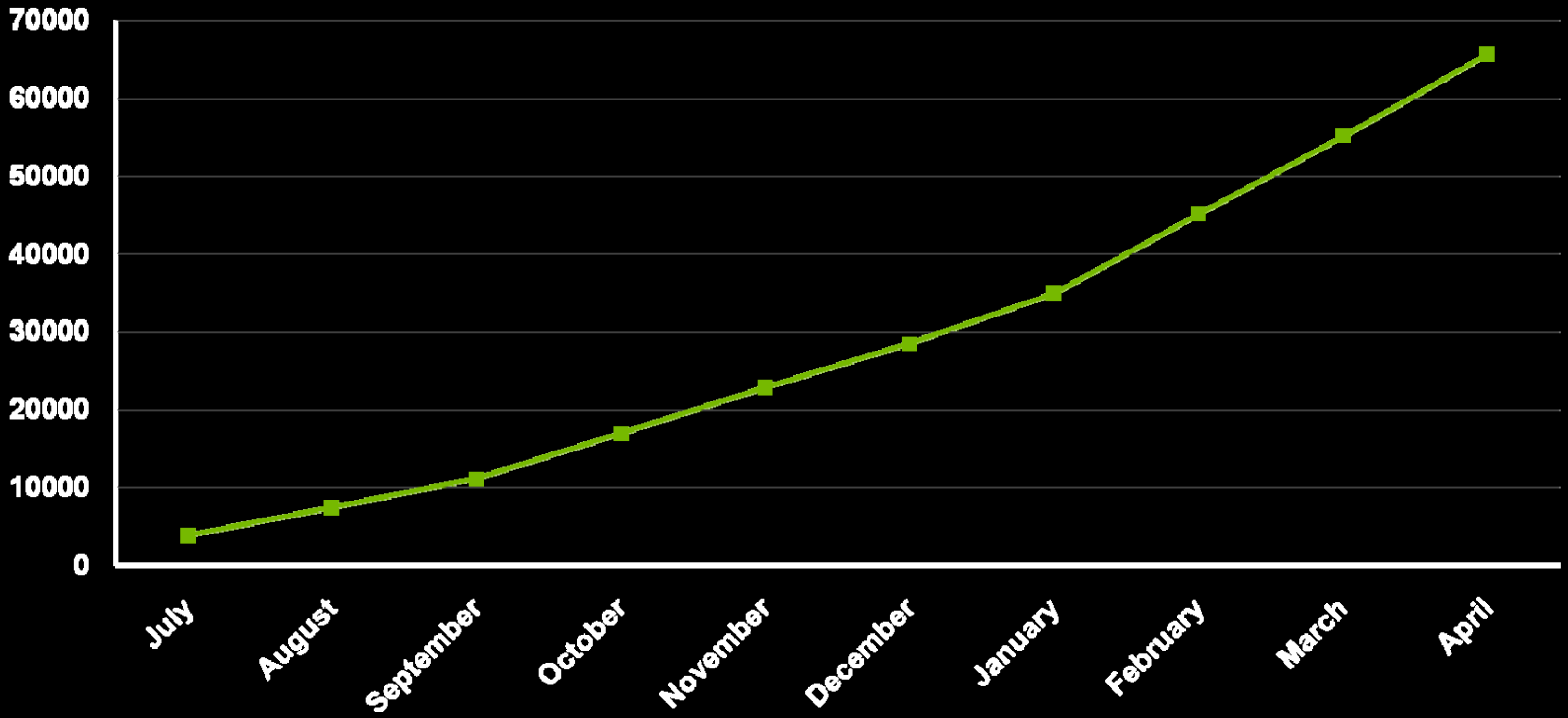

Numerics


Audio


Video


Imaging

CUDA Compiler Downloads



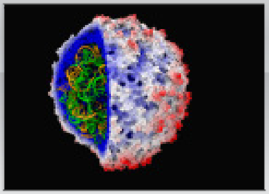
Universities Teaching Parallel Programming With CUDA

- Duke
- Erlangen
- ETH Zurich
- Georgia Tech
- Grove City College
- Harvard
- IIIT
- IIT
- Illinois Urbana-Champaign
- INRIA
- Iowa
- ITESM
- Johns Hopkins
- **Kent State**
- **Kyoto**
- **Lund**
- **Maryland**
- **McGill**
- **MIT**
- **North Carolina - Chapel Hill**
- **North Carolina State**
- **Northeastern**
- **Oregon State**
- **Pennsylvania**
- **Polimi**
- **Purdue**
- Santa Clara
- Stanford
- Stuttgart
- Suny
- Tokyo
- TU-Vienna
- USC
- Utah
- Virginia
- Washington
- Waterloo
- Western Australia
- Williams College
- Wisconsin

Wide Developer Acceptance

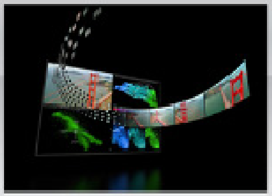
146X

Interactive visualization of volumetric white matter connectivity



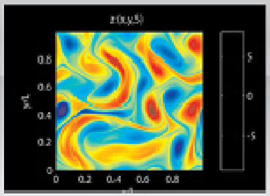
36X

Ionic placement for molecular dynamics simulation on GPU



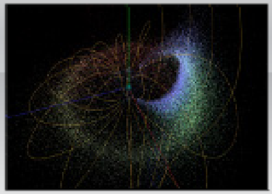
19X

Transcoding HD video stream to H.264



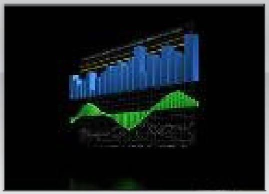
17X

Simulation in Matlab using .mex file CUDA function



100X

Astrophysics N-body simulation

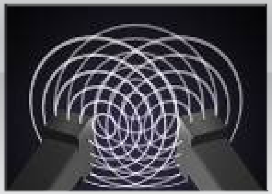


149X

Financial simulation of LIBOR model with swaptions

47X

GLAME@lab: An M-script API for linear Algebra operations on GPU

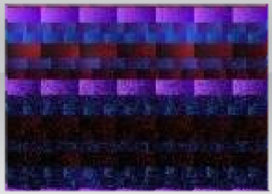


20X

Ultrasound medical imaging for cancer diagnostics

24X

Highly optimized object oriented molecular dynamics

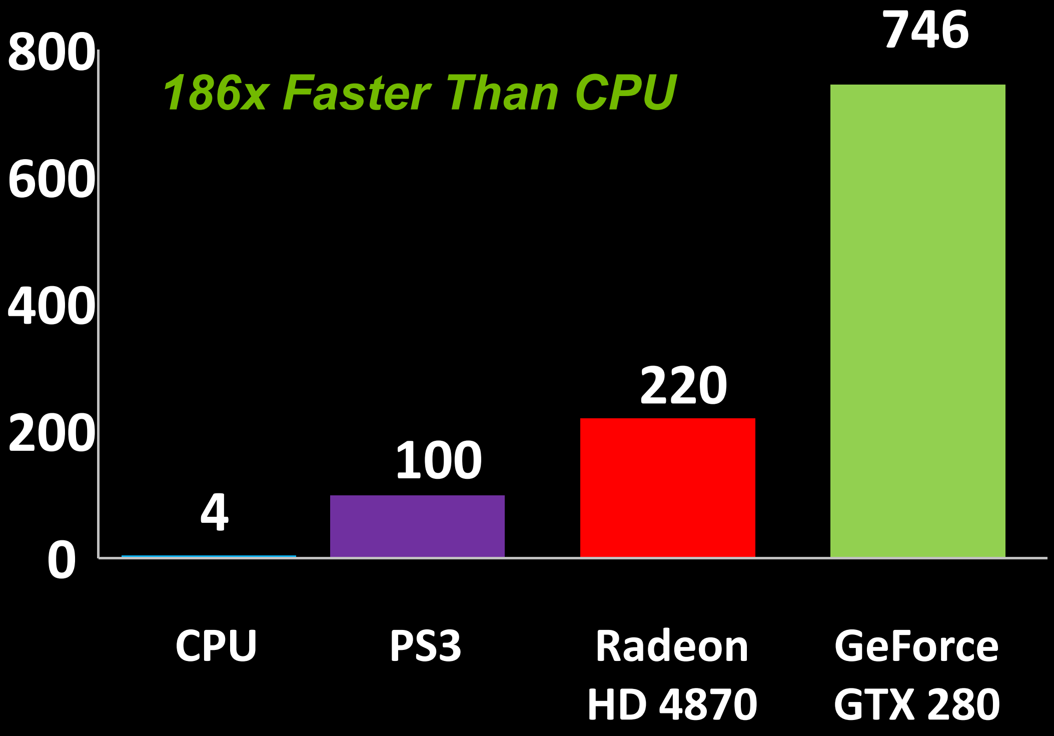


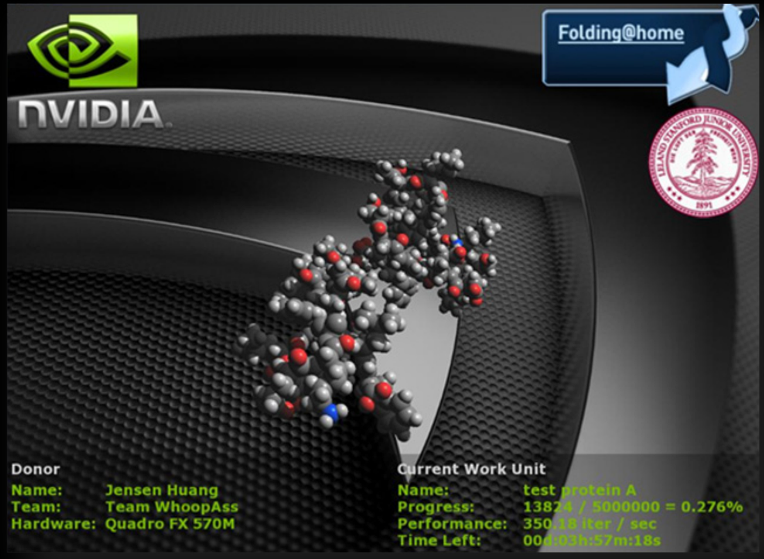
30X

Cmatch exact string matching to find similar proteins and gene sequences

nvision 08
THE WORLD OF VISUAL COMPUTING

Folding@home on GeForce® / CUDA





CUDA Zone

nVISION 08
THE WORLD OF VISUAL COMPUTING

Faster is not “just Faster”

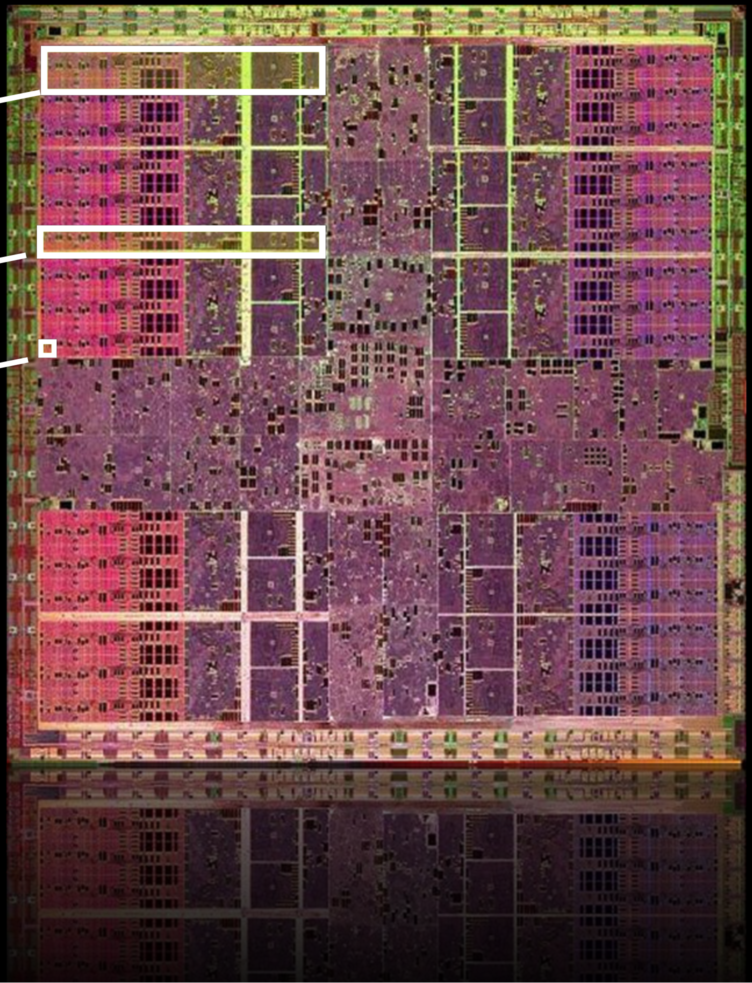
- 2-3X faster is “just faster”
  - Do a little more, wait a little less
  - Doesn’t change how you work
- 5-10x faster is “significant”
  - Worth upgrading
  - Worth re-writing (parts of) the application
- 100x+ faster is “fundamentally different”
  - Worth considering a new platform
  - Worth re-architecting the application
  - Makes new applications possible
  - Drives “time to discovery” and creates fundamental changes in Science

Tesla™ T10: 1.4 Billion Transistors

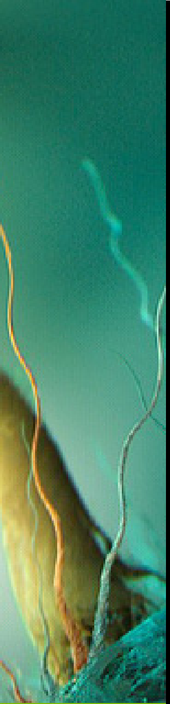
Thread Processor Cluster (TPC)

Thread Processor Array (TPA)

Thread Processor



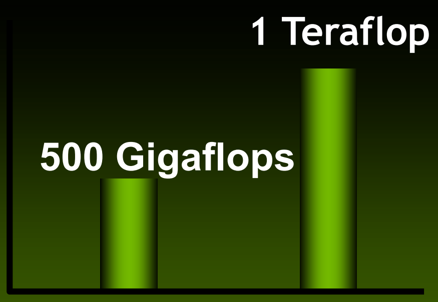
Die Photo of Tesla T10

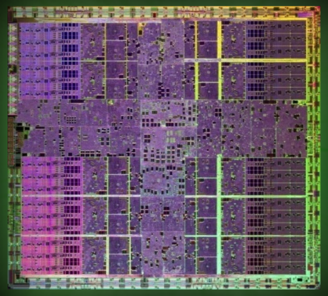


NVISION 08
THE WORLD OF VISUAL COMPUTING

Tesla 10-Series

Double the Performance





Double the Memory

Double Precision

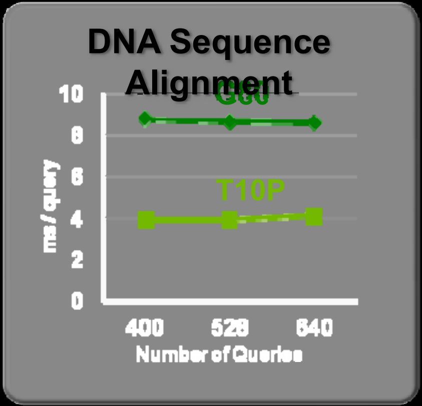
Finance Science Design

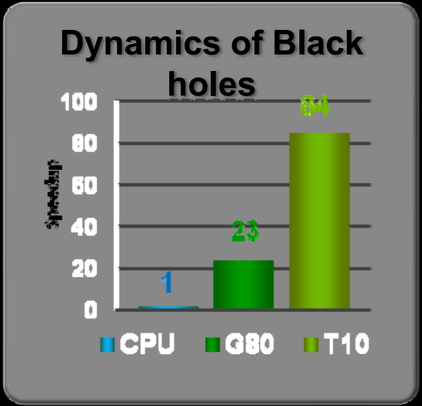
NVIDIA

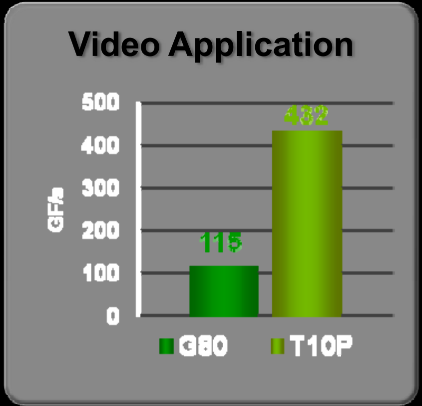
Tesla T10 Double Precision Floating Point

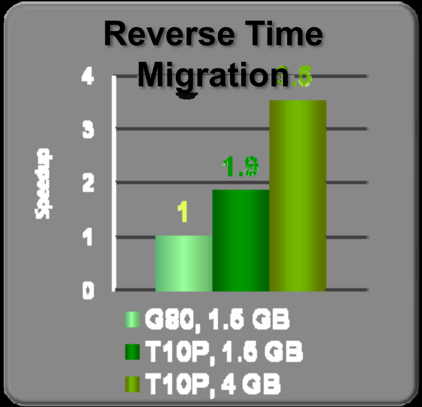
Precision	IEEE 754
Rounding modes for FADD	All 4 IEEE, round to nearest, zero, inf, -inf
Denormal handling	Full speed
NaN support	Yes
Overflow and Infinity support	Yes
Flags	No
FMA	Yes
Square root	Software with low-latency FMA-based convergence
Division	Software with low-latency FMA-based convergence
Reciprocal estimate accuracy	24 bit
Reciprocal sqrt estimate accuracy	23 bit
log2(x) and 2^x estimates accuracy	23 bit

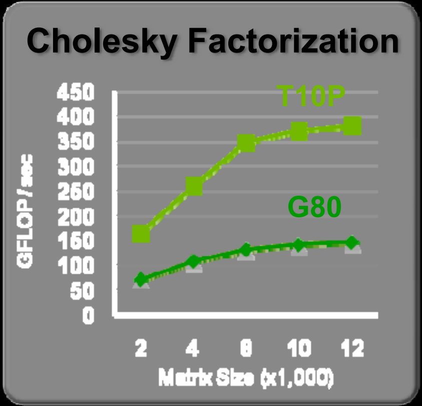
Double the Performance Using T10

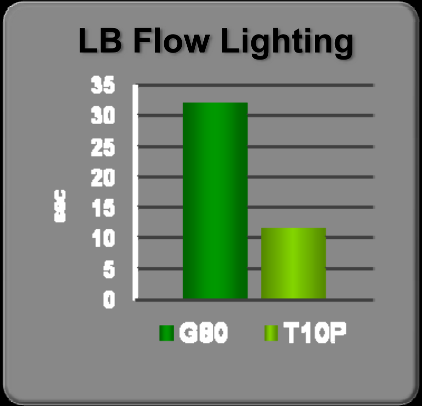


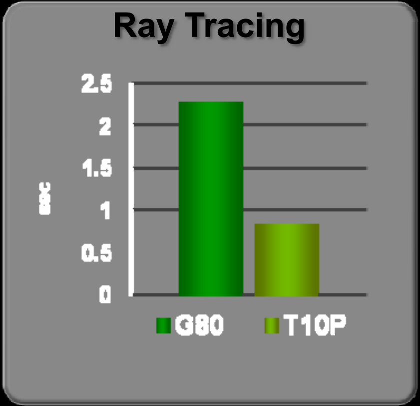








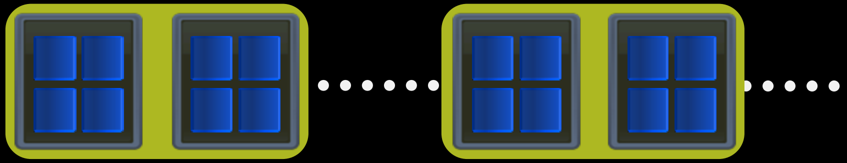




How to Get to 100x?

Traditional Data Center Cluster

Quad-core CPU



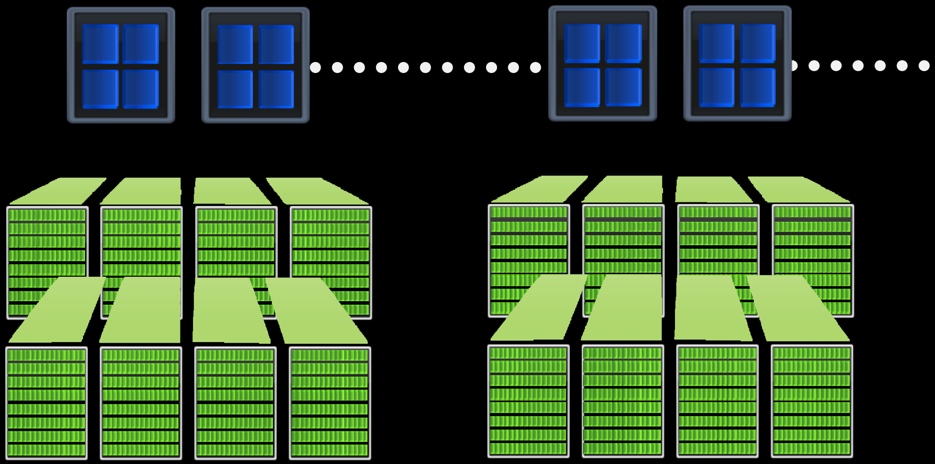
1000's of cores
1000's of servers

cores per server

More Servers To Get More Performance

Heterogeneous Computing Cluster

10,000's processors per cluster



1928 processors **1928 processors**

- **Hess**
- **NCSA / UIUC**
- **JFCOM**
- **SAIC**
- **University of North Carolina**
- **Max Plank Institute**
- **Rice University**
- **University of Maryland**
- **GusGus**
- **Eotvas University**
- **University of Wuppertal**
- **IPE/Chinese Academy of Sciences**
- **Cell phone manufacturers**

Building a 100TF datacenter

CPU 1U Server		Tesla 1U System
4 CPU cores	**4 GPUs: 960 cores**	
0.07 Teraflop	4 Teraflops	
$ 2000	$ 8000	
400 W	800 W	
1429 CPU servers	**25 CPU servers** **25 Tesla systems**	
$ 3.1 M	$ 0.31 M	*10x lower cost*
571 KW	27 KW	*21x lower power*

NVIDIA

Tesla S1070 1U System

4 Teraflops[1]

800 watts[2]

[1] single precision
[2] typical power

Tesla C1060 Computing Processor

957 Gigaflops[1]

160 watts[2]

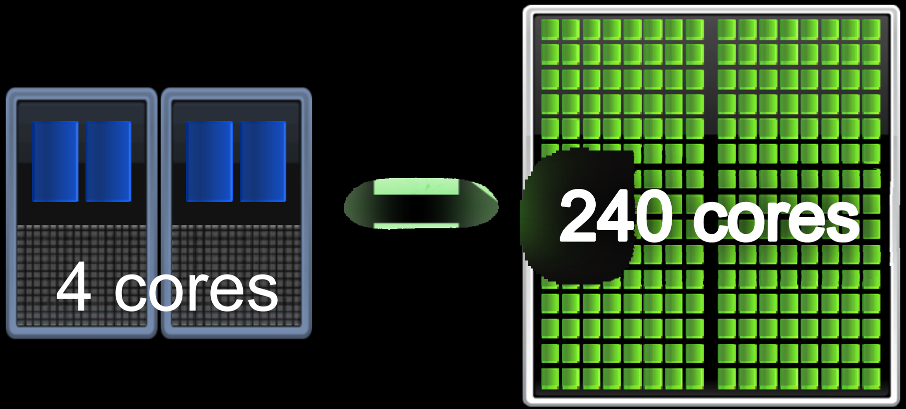
[1] single precision
[2] typical power

Application Software
Industry Standard C Language

Libraries

cuFFT | cuBLAS | cuDPP

System	CUDA Compiler	CUDA Tools
1U PCI-E Switch	C Fortran Multi-core	Debugger Profiler

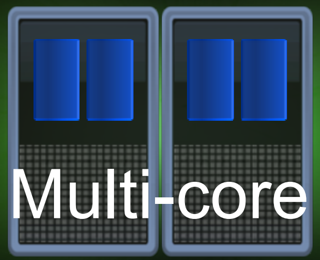


CUDA Source Code

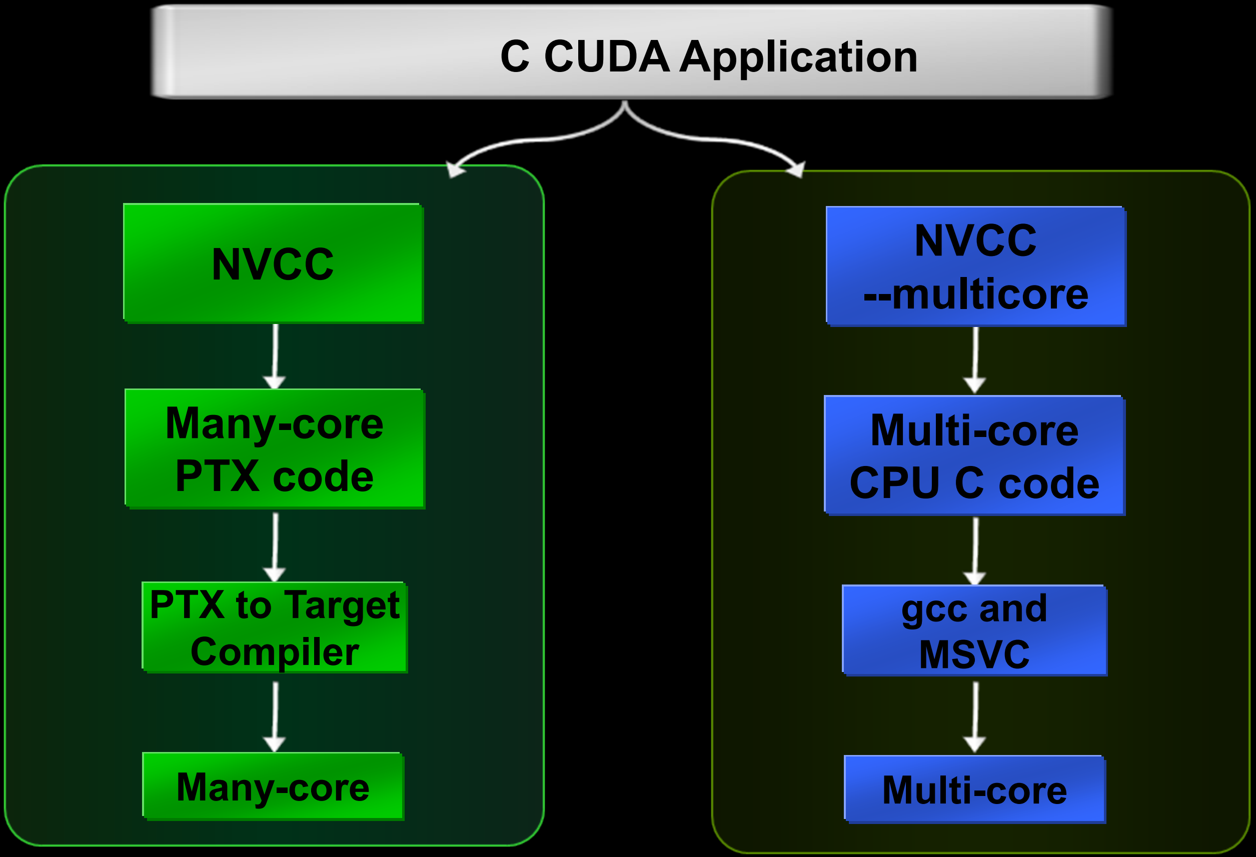
Industry Standard C Language

Industry Standard Libraries

CUDA Compiler	Standard
C Fortran	Debugger Profiler



CUDA 2.1: Many-core + Multi-core support



CUDA Everywhere!