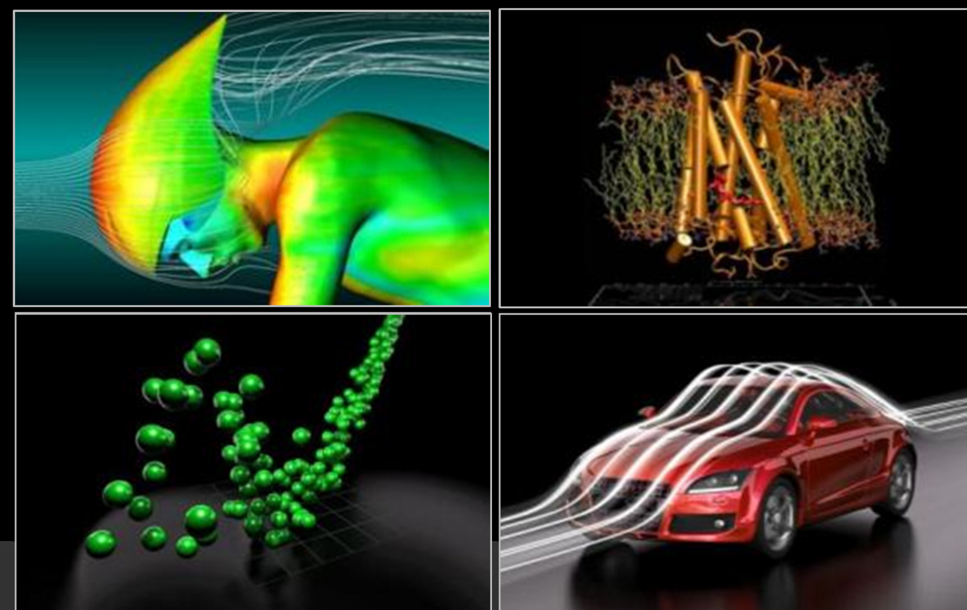
GPU Computing Will Fundamentally Change Science



David B. Kirk, PhD
NVIDIA Fellow

VISUALIZATION

QUADRO™

PARALLEL COMPUTING

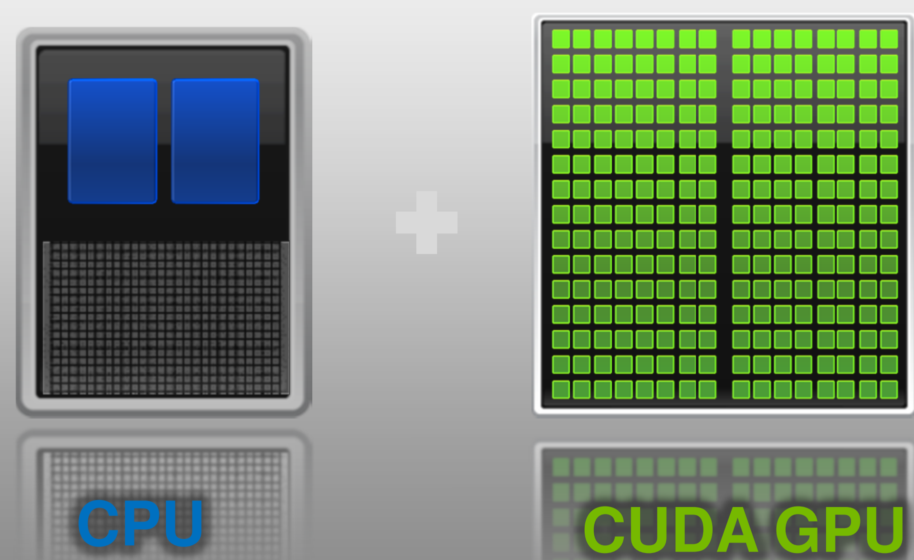
TESLA™

PERSONAL COMPUTING

GeForce™, TEGRA™

CUDA GPU Accelerates Computing

Choose the Right Processor for the Right Job

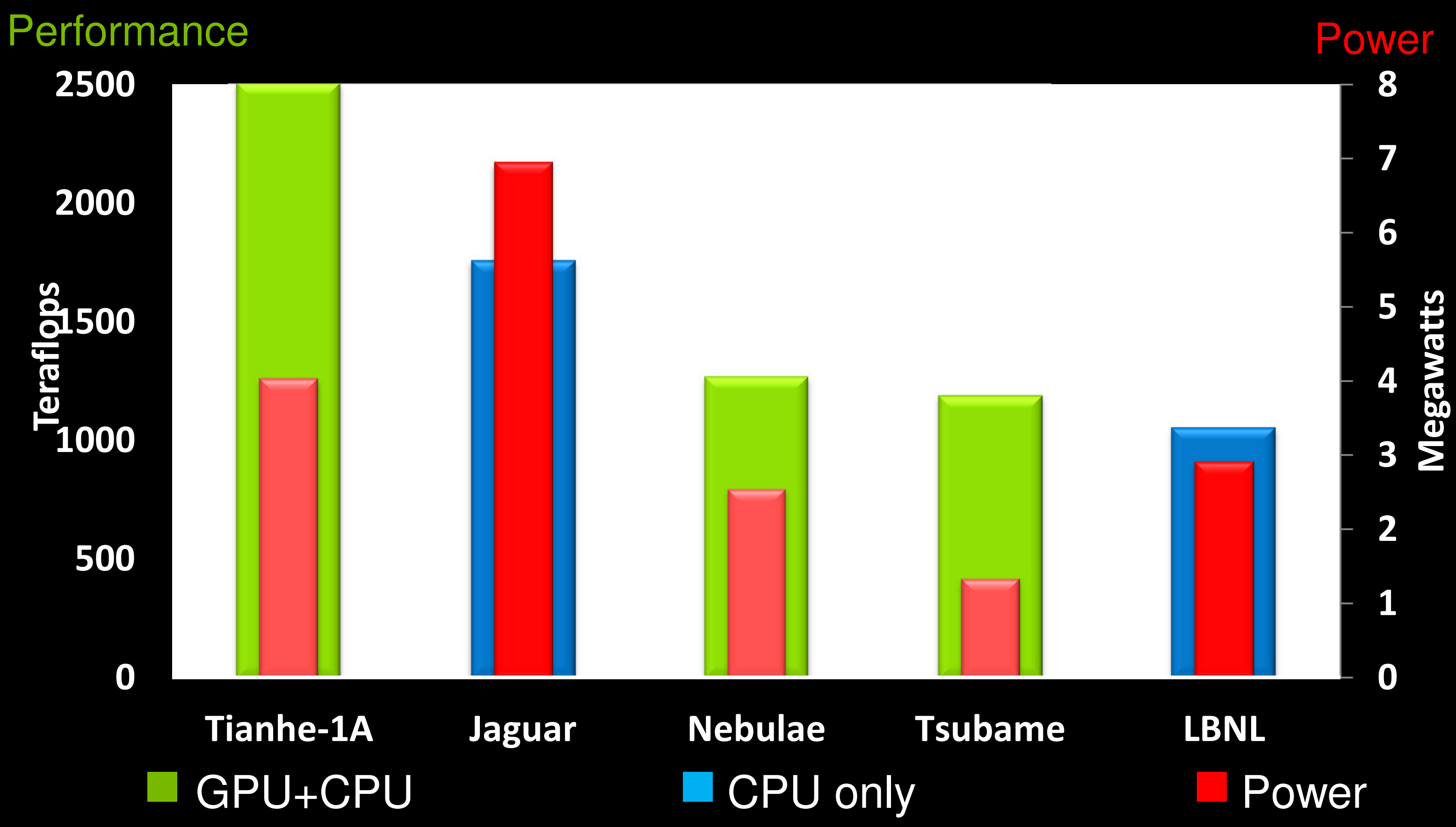


Several sequential cores | Hundreds of parallel cores

GPUs Accelerate Oak Ridge's Titan

- World's top open science lab embraces GPUs to develop world's leading supercomputer
- Cray XK6 machine will use up to 18,000 NVIDIA Tesla GPUs
- Could exceed 20 petaflops, over 2x as fast and 3x as energy-efficient as "K", last year's leader
- Paving the way to exascale

GPU Supercomputers: More Power Efficient



World's Fastest Molecular Dynamics Simulation

Sustained Performance of 1.87 Petaflops/s

Institute of Process Engineering (IPE)

Chinese Academy of Sciences (CAS)

Simulation for Crystalline Silicon

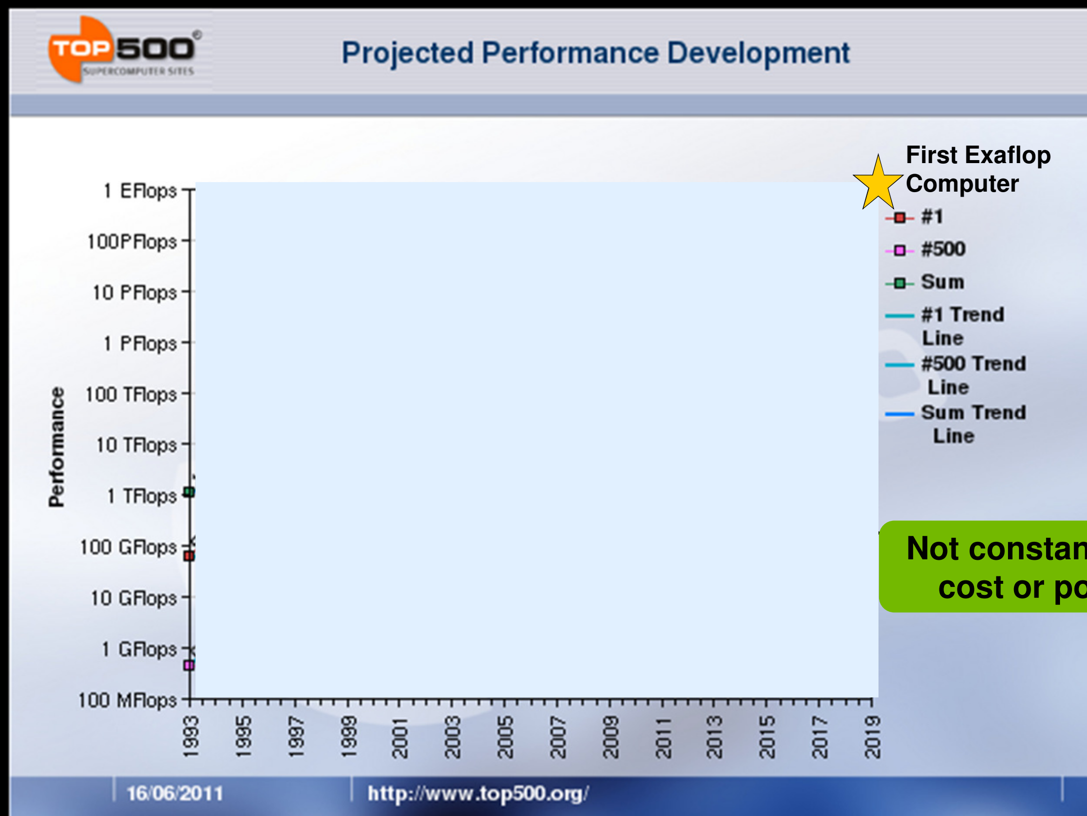
Used for Photovoltaic cells & Semiconductors

Used all 7168 Tesla GPUs on Tianhe-1A GPU Supercomputer

1000+ GPU Clusters Around the World

Exaflop Expectations

CM5
~200 KW

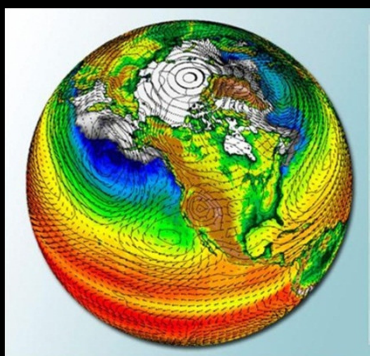


K Computer
~10 MW

Not constant size, cost or power

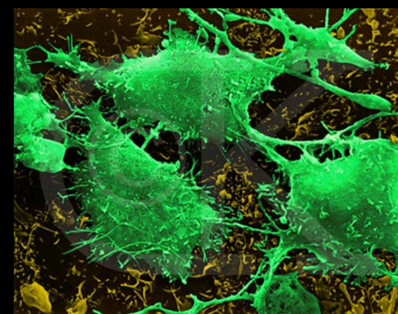
More Powerful Computing Enables Transformational Science Results

From Individual Scientists/Engineers to World Class Teams



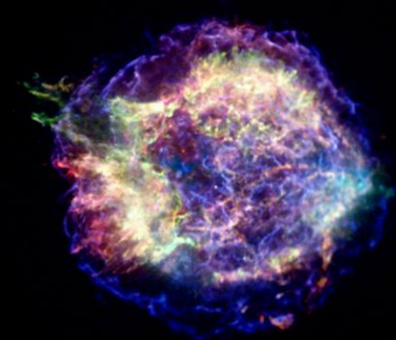
Comprehensive Earth System Model at 1KM scale, enabling modeling of cloud convection and ocean eddies.

First-principles simulation of combustion for new high-efficiency, low-emision engines.



Coupled simulation of entire cells at molecular, genetic, chemical and biological levels.

Predictive calculations for thermonuclear and core-collapse supernovae, allowing confirmation of theoretical models.



(Exascale science challenges)

Power: This Time It's Different

In the Good Old Days

Leakage was not important, and voltage scaled with feature size

$L' = L/2$
$V' = V/2$
$E' = CV^2 = E/8$
$f' = 2f$
$D' = 1/L^2 = 4D$
$P' = P$

Halve L and get 4x the transistors and 8x the capability for the same power!

MF to GF to TF and almost to PF

Technology was giving us **68%** per year in perf/W!

Processors realized ~50% per year in perf/W…
(spent it on single thread performance)

The New Reality

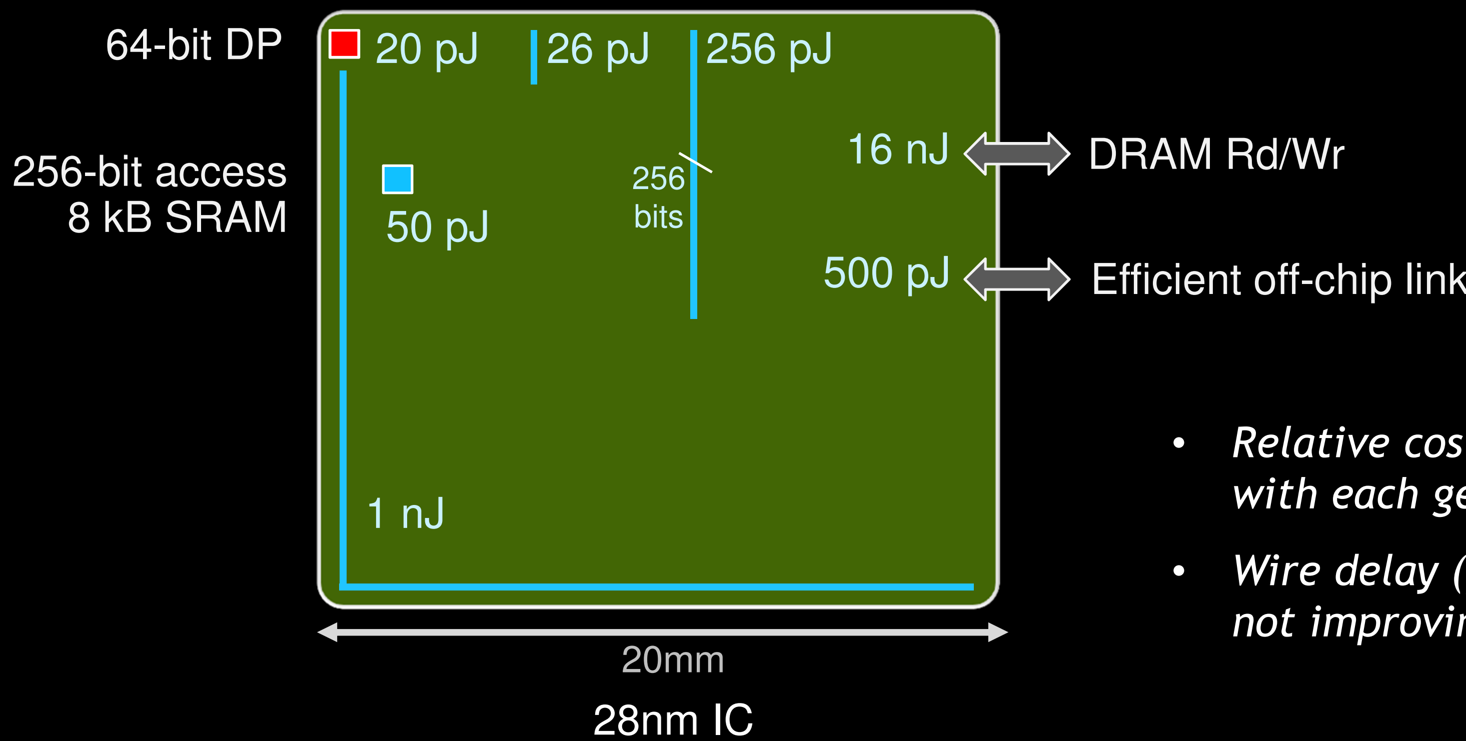
Leakage has limited threshold voltage, largely ending voltage scaling

Halve L and get 2x the capability for the same power.

At constant voltage, technology gives us only **19%** per year in perf/W

The High Cost of Data Movement

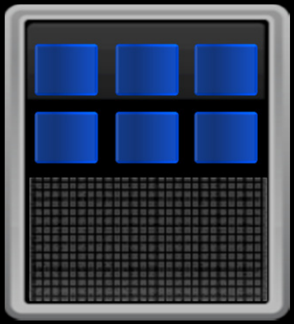
Fetching operands costs more than computing on them



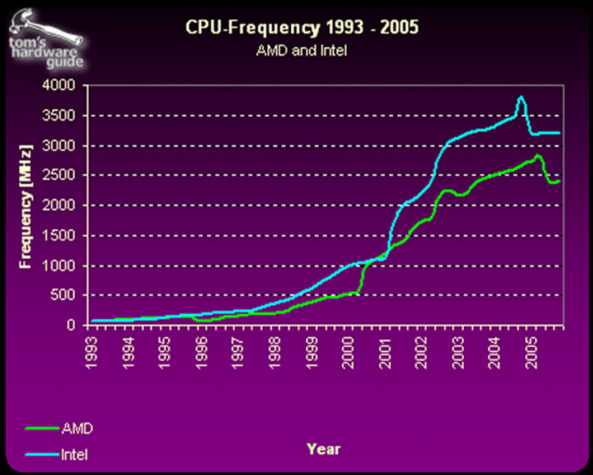
- *Relative cost grows with each generation*
- *Wire delay (ps/mm) not improving*

So, What To Do?

1) Stop making it worse...



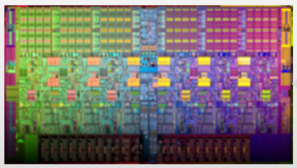
Multicore CPUs



But still only about 2% of CPU power spent on flops

2) Unwind all that complexity we threw at single thread performance

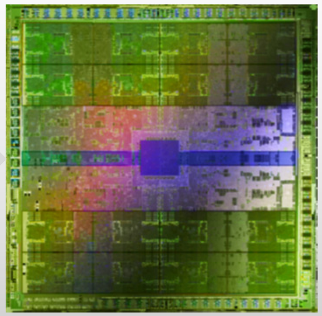
HPC Going Hybrid



x86 CPU
Fast single threads
(serial work)

PCIe

Westmere
32nm
1.7 nJ/flop



Fermi
40nm
225 pJ/flop

GPU
Extreme power-efficiency
(throughput work)

- Do most work by cores optimized for **extreme energy efficiency**
- Still need a few cores optimized for **fast serial work**

And memory hierarchies are getting deeper…

Major Software Implications

Computers are not getting faster… just **wider**

⇒ *Need to cast all HPC apps as throughput problems, and expose massive parallelism*

Locality across nodes is not the problem

… *vertical* locality is

⇒ *Need to expose locality & explicitly manage memory hierarchy*

(programming model) *(compiler, runtime, auto-tuner)*

Science per Watt

=

Performance per Watt

+

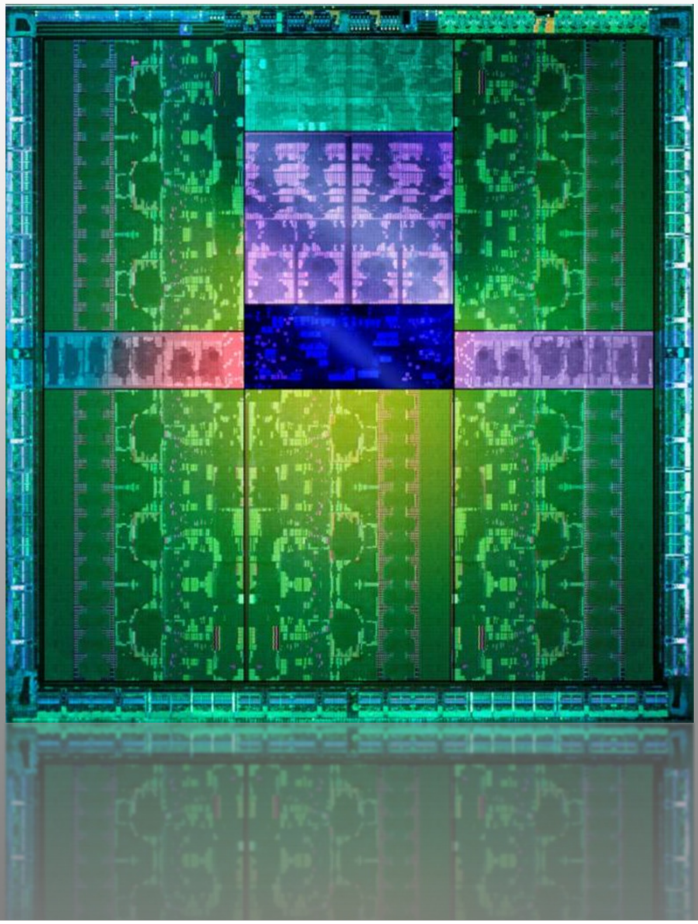
Programmability

Long Term Goals for Tesla

Power
Efficiency

Ease of
Programming
And Portability

Application
Space
Coverage



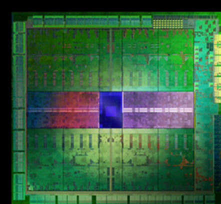
KEPLER

SMX *(power efficiency)*

Hyper-Q

Dynamic Parallelism *(programmability and application coverage)*

Tesla K10



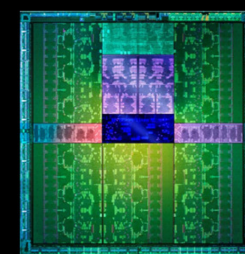
Dual GK104 GPUs

3x Single Precision

Video, Signal, Life Sciences, Seismic

Available Now

Tesla K20



GK110 GPU

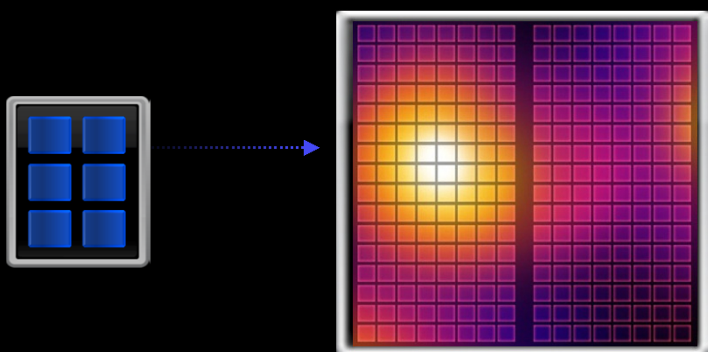
3x Double Precision

CFD, FEA, Finance, Physics, etc.

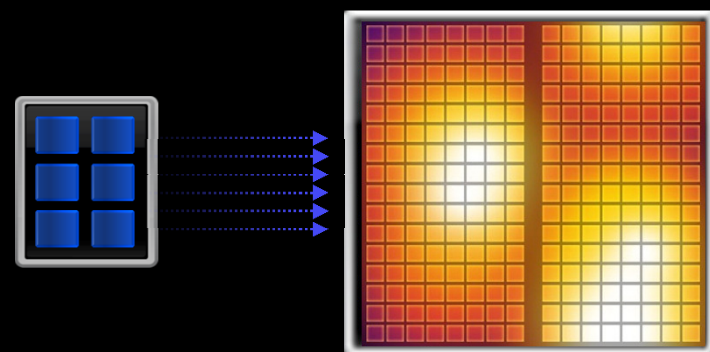
Available Q4 2012

Hyper-Q

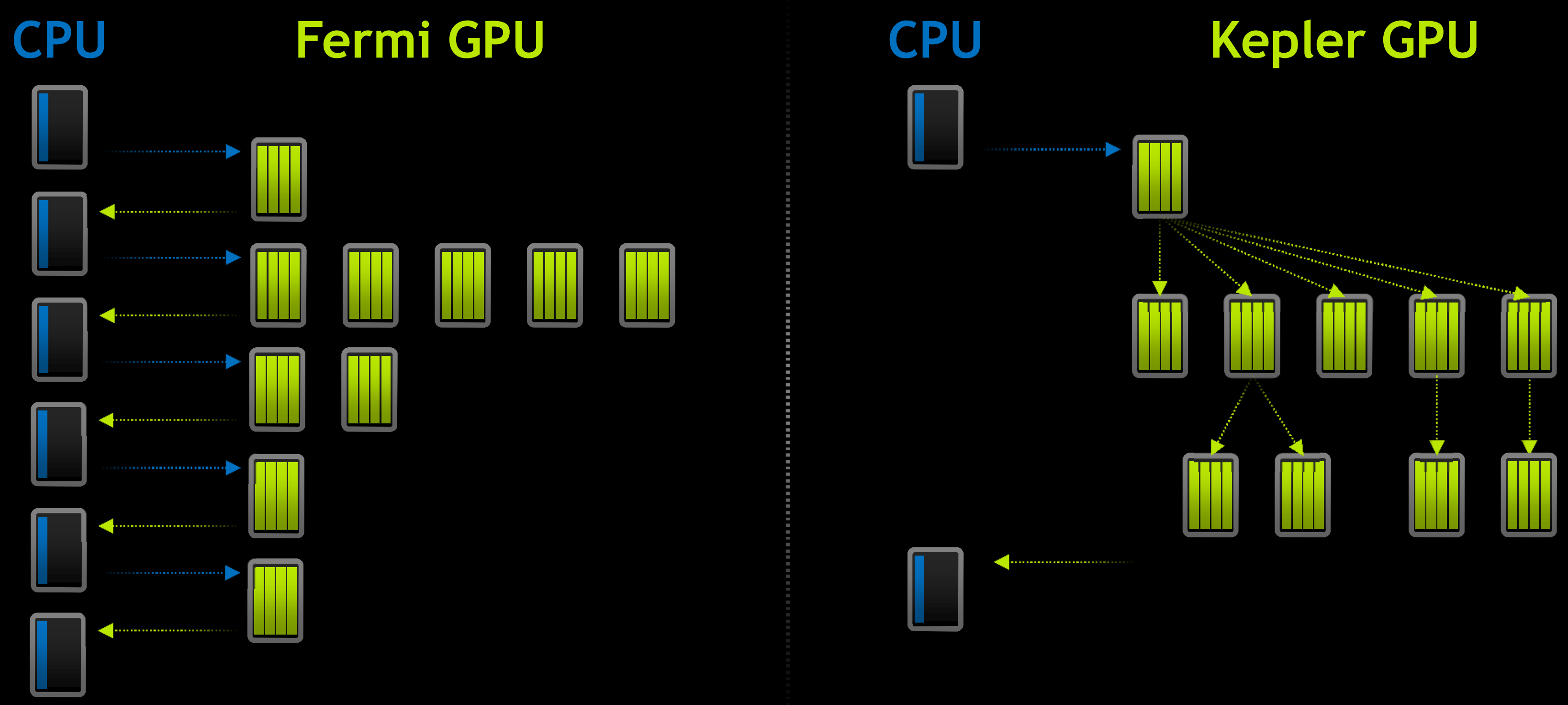
FERMI
1 Work Queue



KEPLER
32 Concurrent Work Queues



Dynamic Parallelism



Kepler GK110 SMX vs Fermi SM

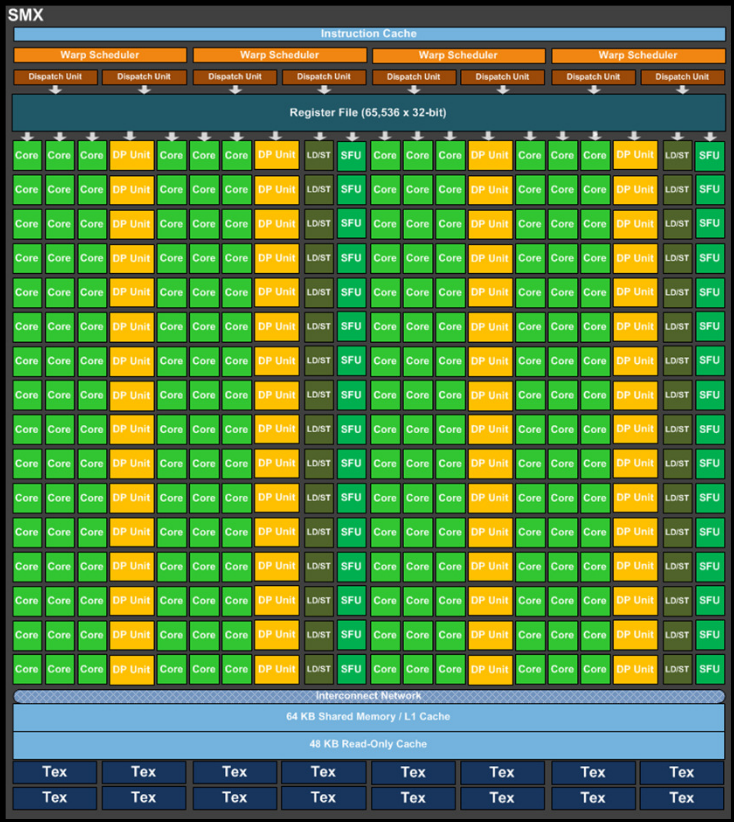
3x sustained perf/W

Ground up redesign for perf/W

6x the SP FP units

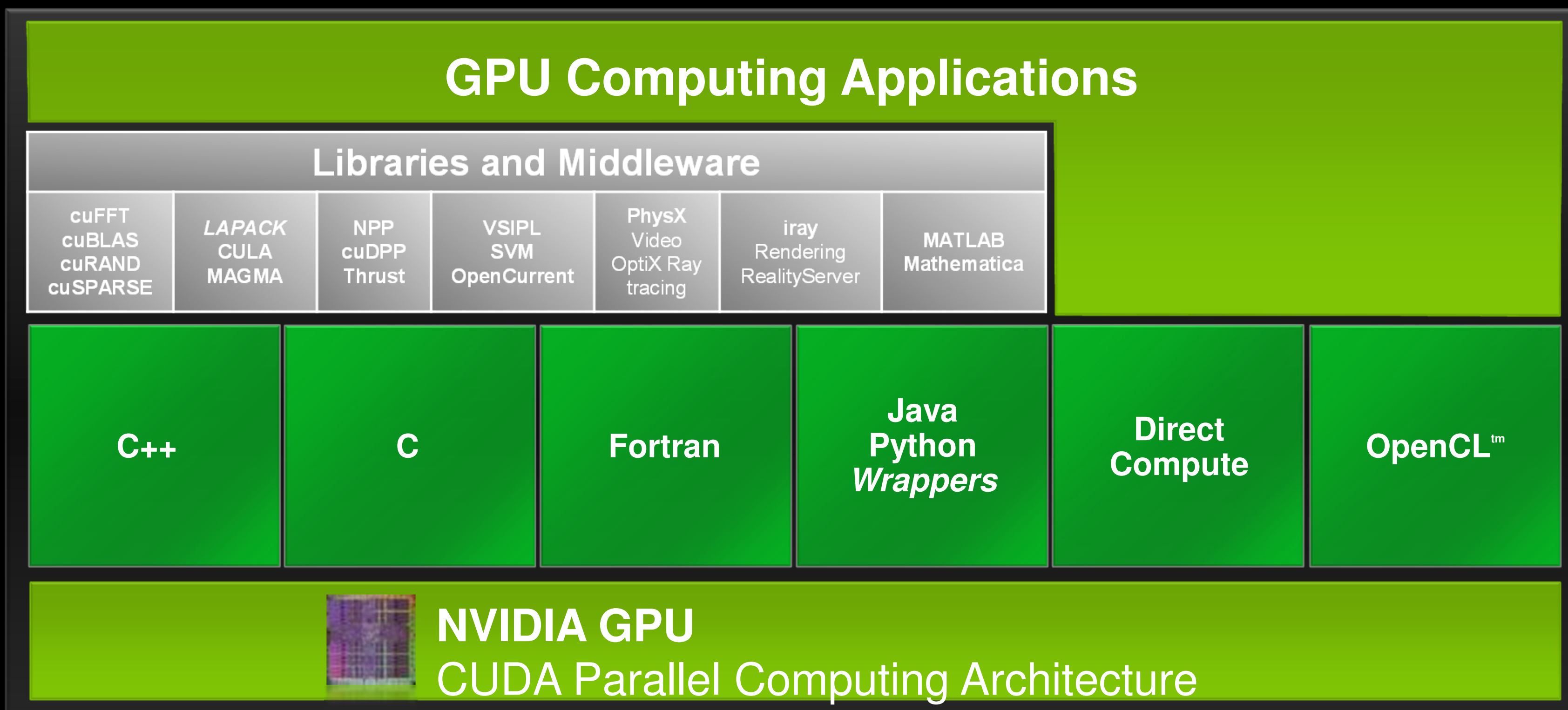
4x the DP FP units

Significantly slower FU clocks

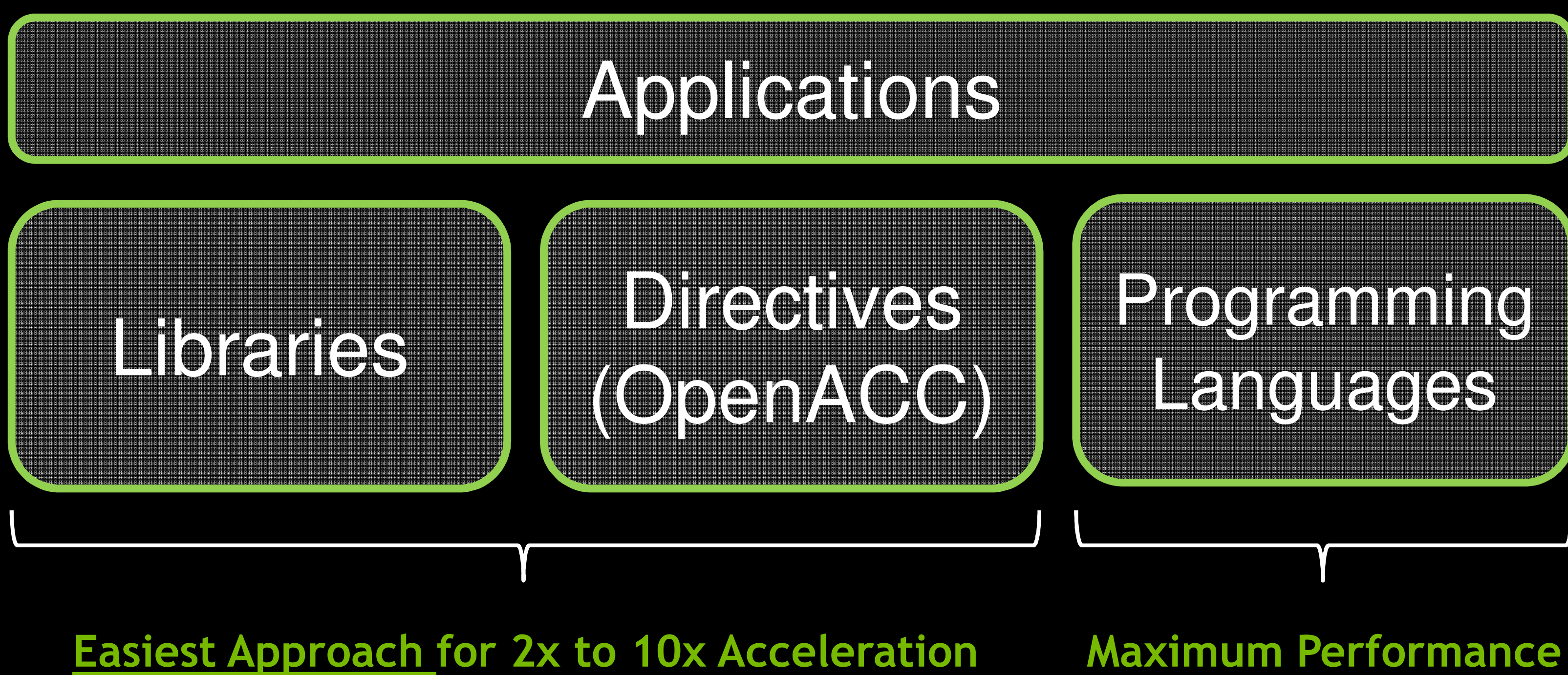


Programming GPUs

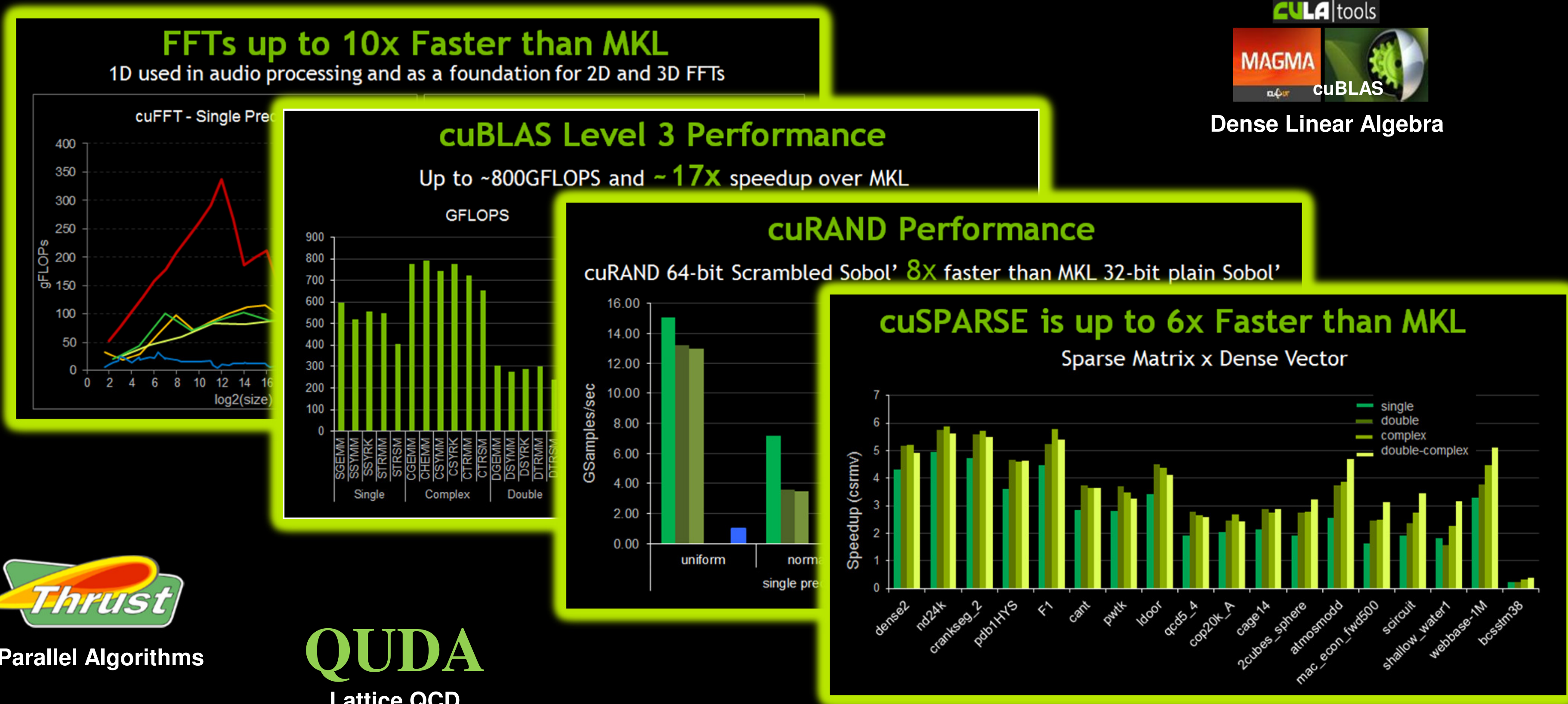
CUDA: Easy to Use Parallel Programming Model



3 Ways to Accelerate Your Apps



GPU Libraries: Simply Use and Accelerate



MAGMA cuBLAS CULA tools

Dense Linear Algebra

Thrust

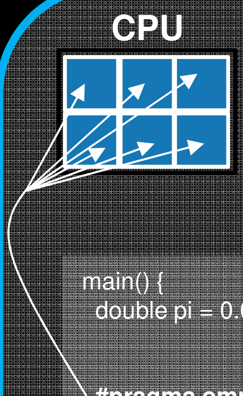
Parallel Algorithms

QUDA

Lattice QCD

Directives: Add One Line of Code

OpenMP

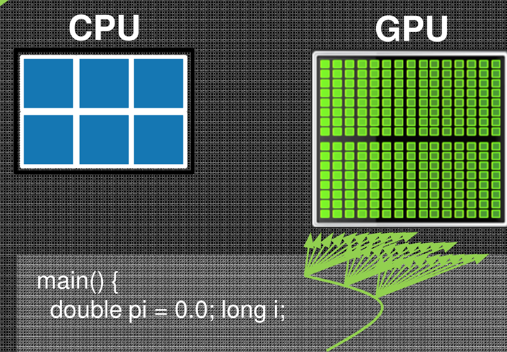



```
main() {
  double pi = 0.0; long i;


  #pragma omp parallel for reduction(+:pi)
  for (i=0; i<N; i++)
  {
    double t = (double)((i+0.05)/N);
    pi += 4.0/(1.0+t*t);
  }

  printf("pi = %f\n", pi/N);
}
```

GPU Directives*




```
main() {
  double pi = 0.0; long i;

  #pragma omp acc_region_loop
  #pragma omp parallel for reduction(+:pi)
  for (i=0; i<N; i++)
  {
    double t = (double)((i+0.05)/N);
    pi += 4.0/(1.0+t*t);
  }
  #pragma omp end acc_region_loop
  printf("pi = %f\n", pi/N);
}
```

*Directives from Cray

C for CUDA : C with a few keywords

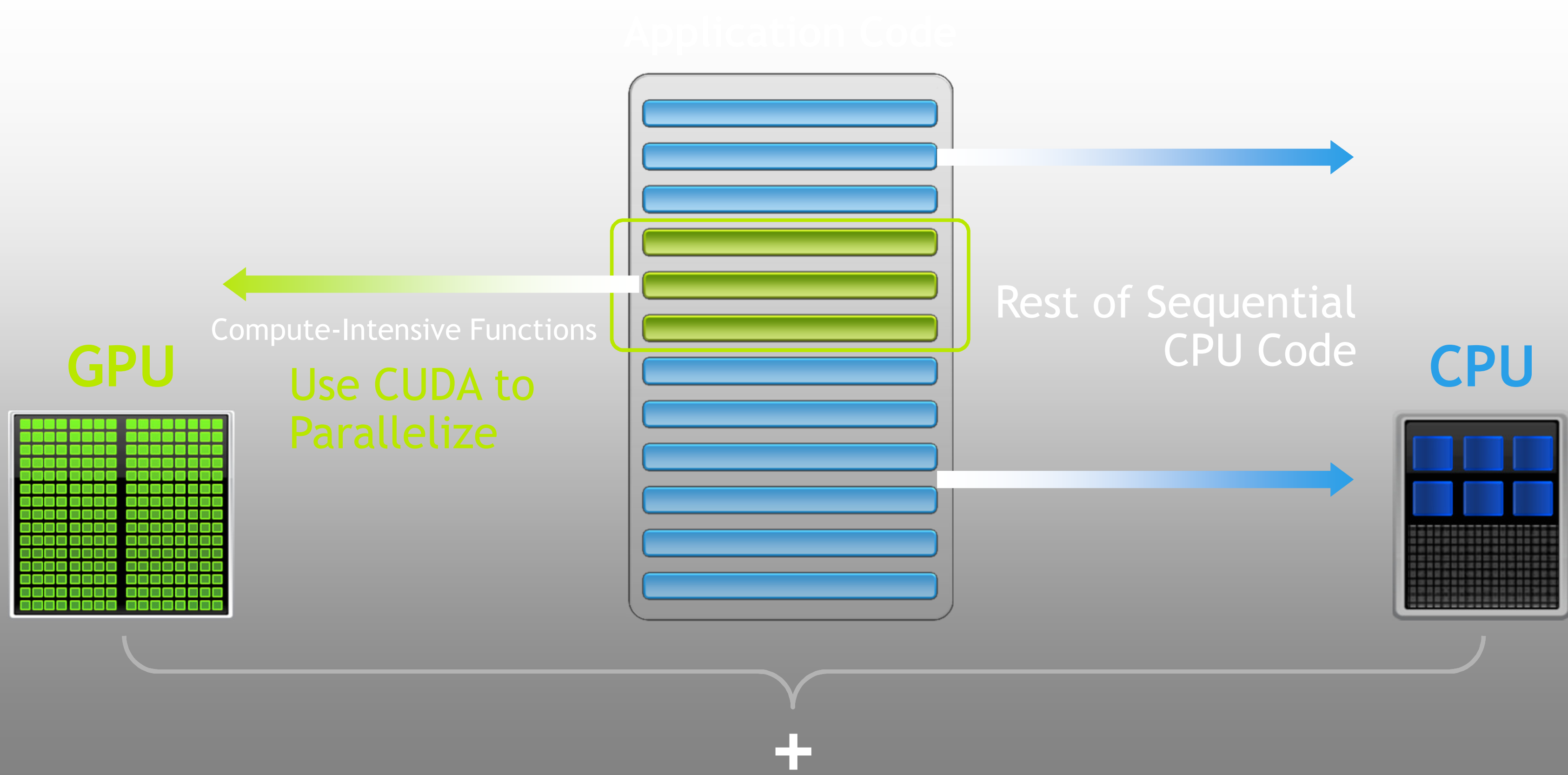
```
void saxpy_serial(int n, float a, float *x, float *y)
{
    for (int i = 0; i < n; ++i)
        y[i] = a*x[i] + y[i];
}
// Invoke serial SAXPY kernel
saxpy_serial(n, 2.0, x, y);
```

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);
```

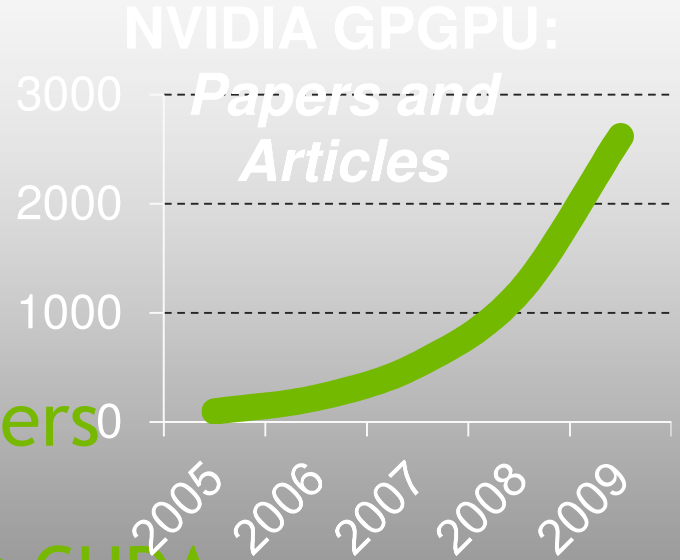
Parallel C Code

Minimum Change, Big Speed-up



CUDA By the Numbers:

300,000,000	CUDA Capable GPUs
1,000,000	CUDA Toolkit Downloads
100,000	Active CUDA Developers
460	Universities Teaching CUDA
100	% OEMs offer CUDA GPU PCs



Getting Started with GPUs

TRY

Try CUDA 4.0 on your Notebook or Desktop with CUDA Enabled GPU

DEVELOP

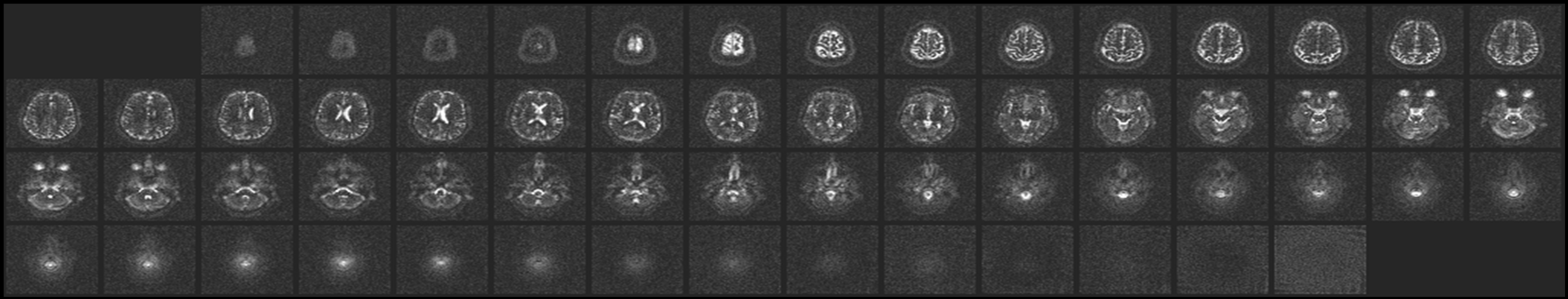
Optimize HPC Apps on Compute Workstation with Tesla GPUs

DEPLOY

Run apps in production with GPU Compute Cluster

Making Science Better, not just Faster

An Exciting Revolution - Sodium Map of the Brain

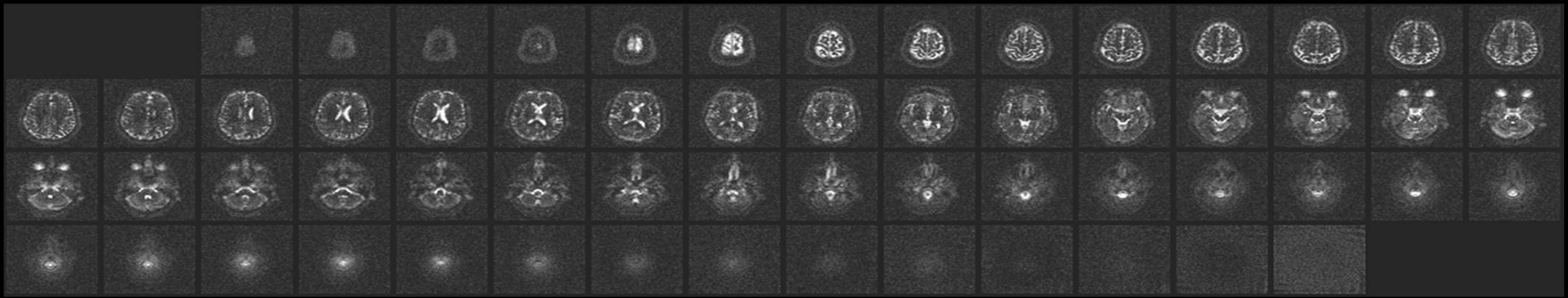


Courtesy of Keith Thulborn and Ian Atkinson, Center for MR Research, University of Illinois at Chicago

- **Images of sodium in the brain**
  - **Sodium is one of the most regulated substance in human tissues**
  - **Any significant shift in sodium concentration signals cell death**
  - **Much less abundant than water in human tissues, about 1/2000**
  - **Very large number of samples are needed for good SNR**
  - **Requires high-quality reconstruction, currently considered impractical**

Thanks: Wen-mei Hwu

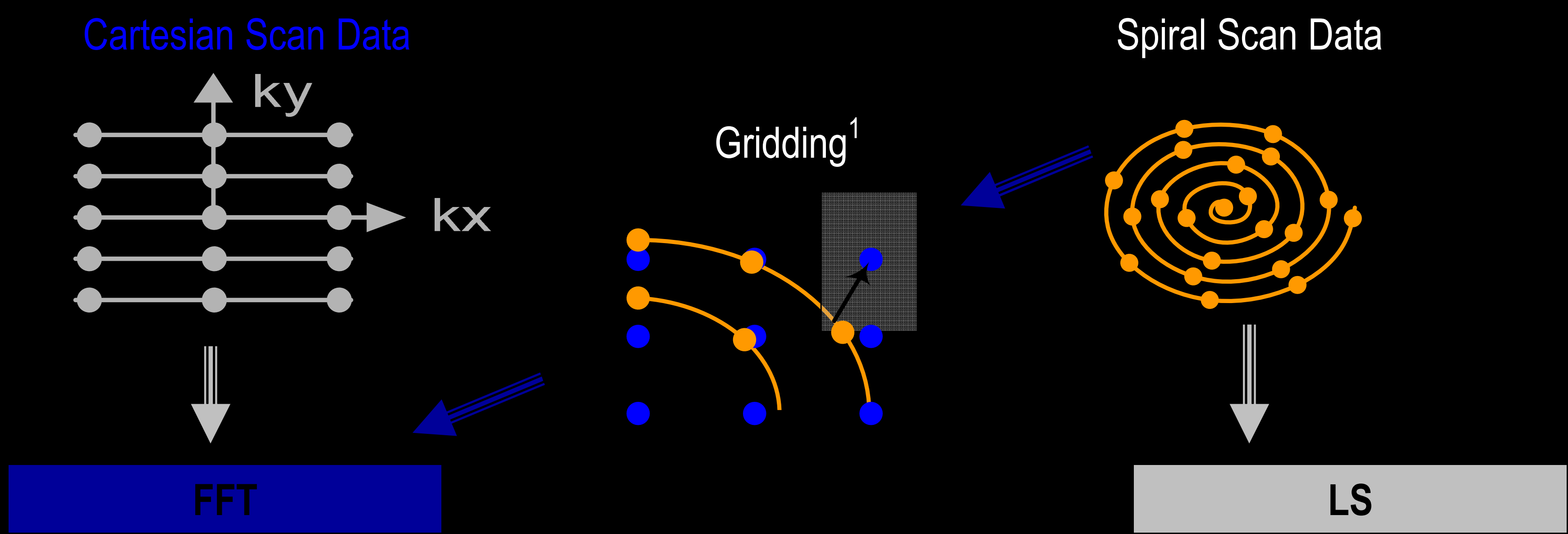
An Exciting Revolution - Sodium Map of the Brain



Courtesy of Keith Thulborn and Ian Atkinson, Center for MR Research, University of Illinois at Chicago

- Enables study of brain-cell viability before anatomic changes occur in stroke and cancer treatment.
  - Drastic improvement of timeliness of treatment decision
  - Minutes for stroke and days for oncology.

Reconstructing MR Images



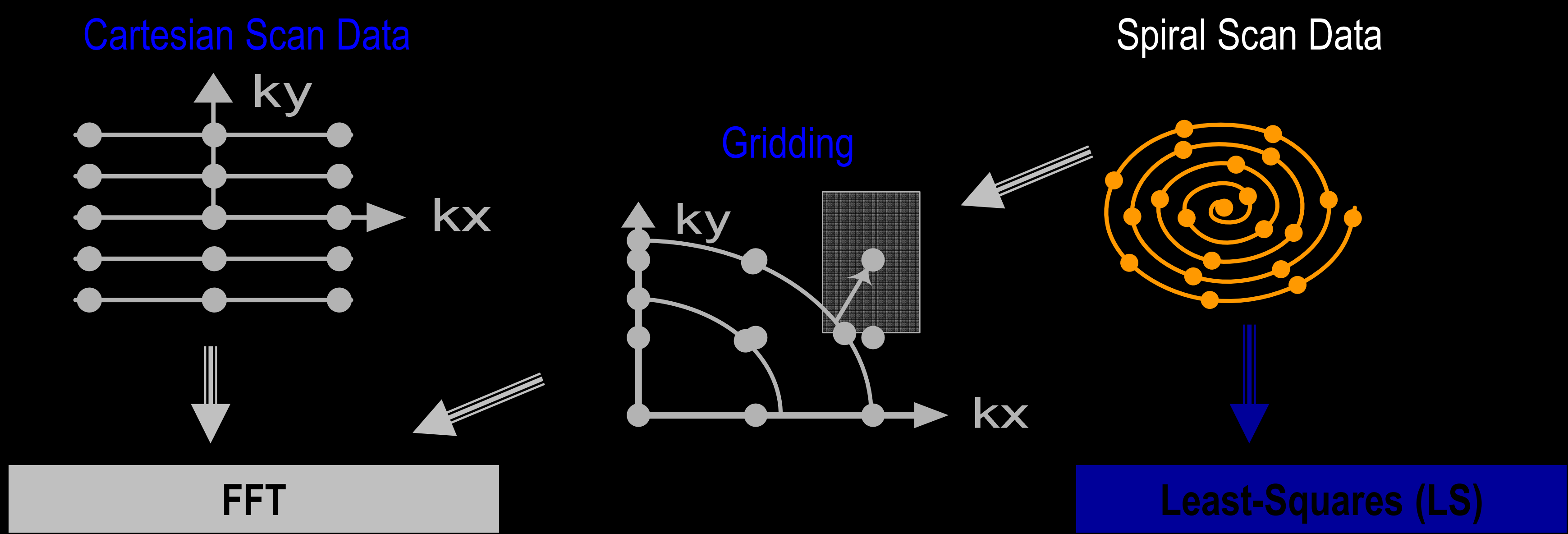
Spiral scan data + Gridding + FFT:

Fast scan, fast reconstruction, good images

Can become realtime with about 10X speedup.

[1] Based on Fig 1 of Lustig et al, Fast Spiral Fourier Transform for Iterative MR Image Reconstruction, IEEE Int'l Symp. on Biomedical Imaging, 2004

Reconstructing MR Images



Spiral scan data + LS

Superior images at expense of significantly more computation; several hundred times slower than gridding.

Traditionally considered impractical!

Conclusion: Three Options for CUDA Adoption

- "Accelerate" Legacy Codes
  - Call CUBLAS/CUFFT/thrust/matlab/PGI pragmas/etc.

    => good work for domain scientists (minimal CS required)

- Rewrite / Create new codes
  - Opportunity for clever algorithmic thinking

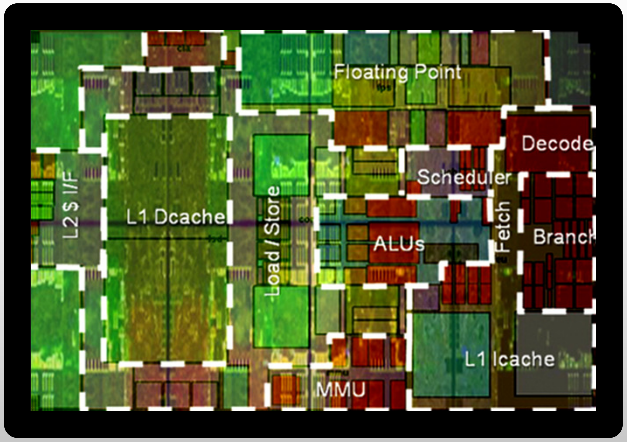
    => good work for computer scientists (minimal domain knowledge required)

- Rethink Numerical Methods & Algorithms
  - Potential for biggest performance advantage

    => Interdisciplinary: requires CS *and* domain insight

    => Exciting time to be a computational scientist

The Future



Project Denver

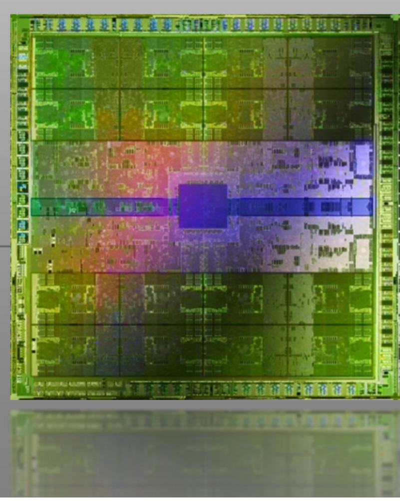
NVIDIA-Designed
High Performance ARM CPU

GPU

200pJ/Instruction

Optimized for Throughput

Explicit Management of On-chip Memory

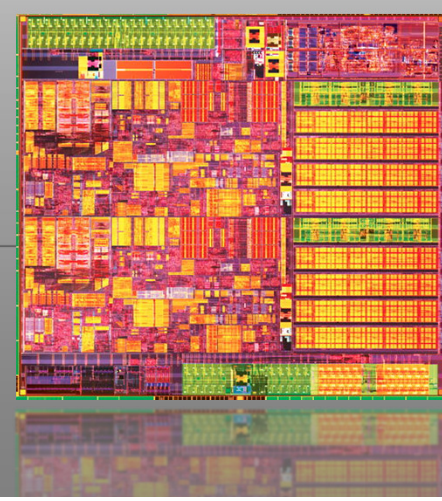


CPU

2nJ/Instruction

Optimized for Latency

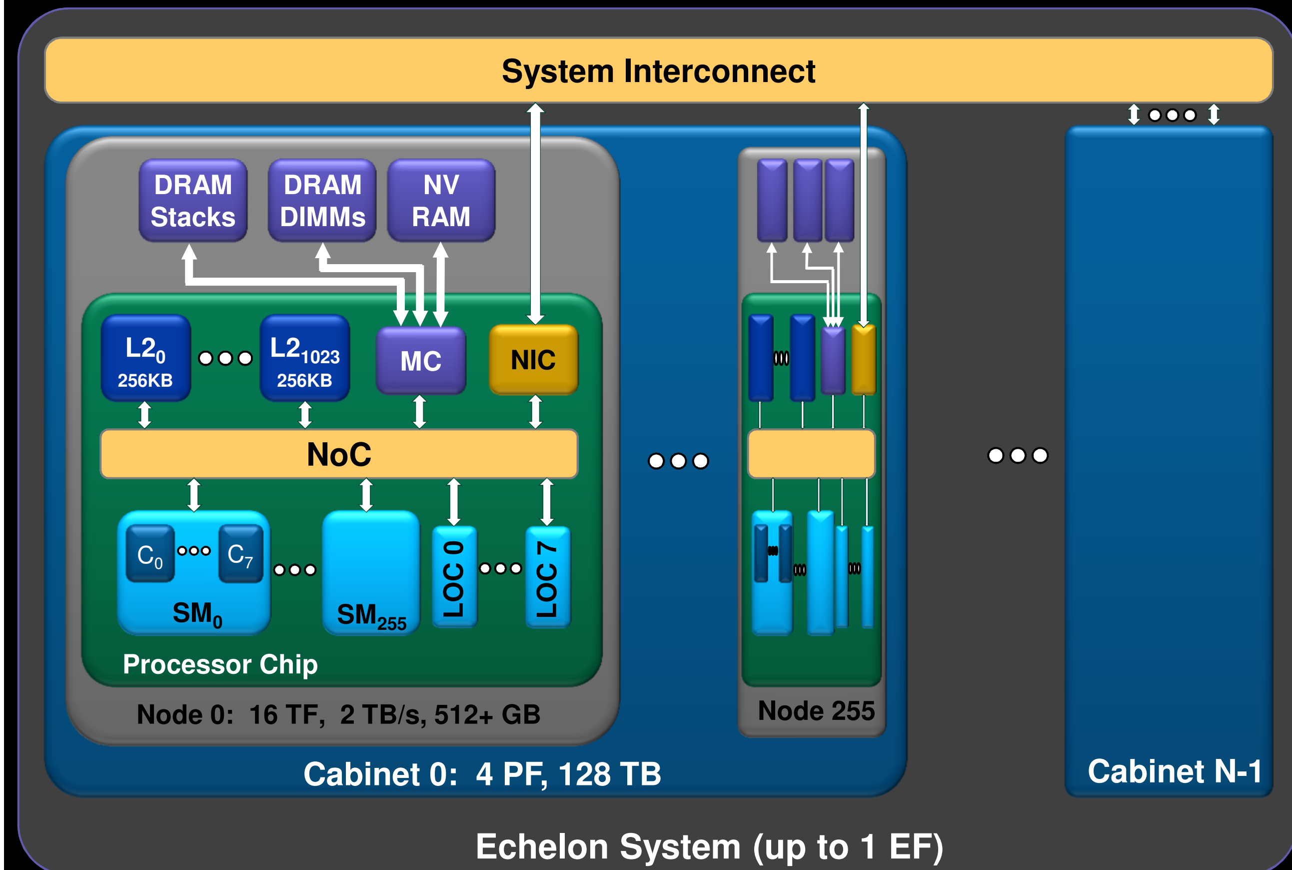
Caches



Echelon

NVIDIA's Extreme-Scale Computing Project
DARPA UHPC Program
Targeting 2018

Echelon Compute Node & System



Key architectural features:

- Malleable memory hierarchy
- Hierarchical register files
- Hierarchical thread scheduling
- Place coherency/consistency
- Temporal SIMT & scalarization
- PGAS memory
- HW accelerated queues
- Active messages
- AMOs everywhere
- Collective engines
- Streamlined LOC/TOC interaction

Academic Program Goals

Engage with external researchers

Learn **from emerging research ideas**

Guide researchers working on important problems

Ignite **disciplines with** the power of GPUs and CUDA

research.nvidia.com

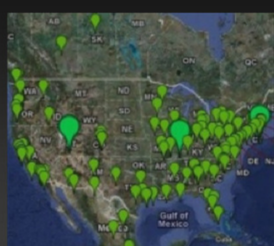
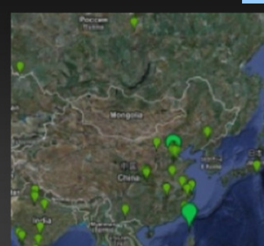
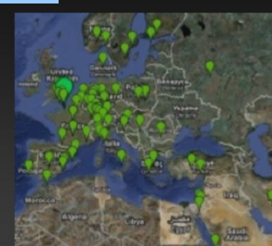
Support
- CUDA Centers
- Academic Partnerships
- Graduate Fellowships
- Internships & Coops

Resources
- CUDA Courses and Training
- CUDA Zone
- Developer Zone

Discuss
- Research Summit at GTC
- CUDA Forums
- twitter.com/gpucomputing

Advancing The Parallel Computing Revolution

The CUDA Parallel Programming Model has been taught in 400+ Universities


NVIDIA

SuperPhones to SuperComputers

Thank You