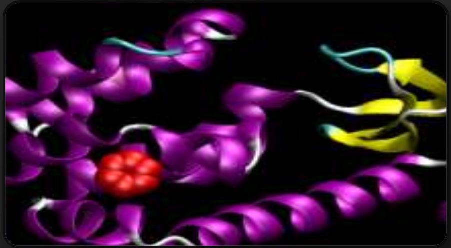
GPU Acceleration Benefits for Scientific Applications

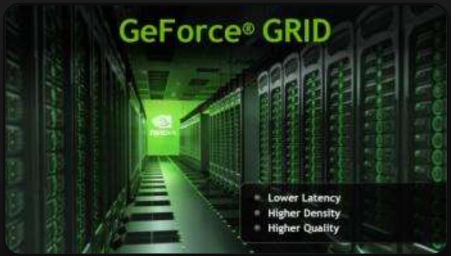
Axel Koehler
Sr. Solution Architect HPC

NVIDIA

NVIDIA: Parallel Computing Company

GPUs: GeForce, Quadro, Tesla

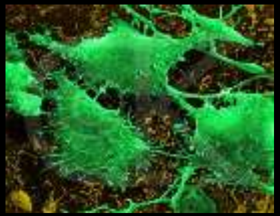






ARM SoCs: Tegra


Continued Demand for Compute Power

Comprehensive Earth System Model

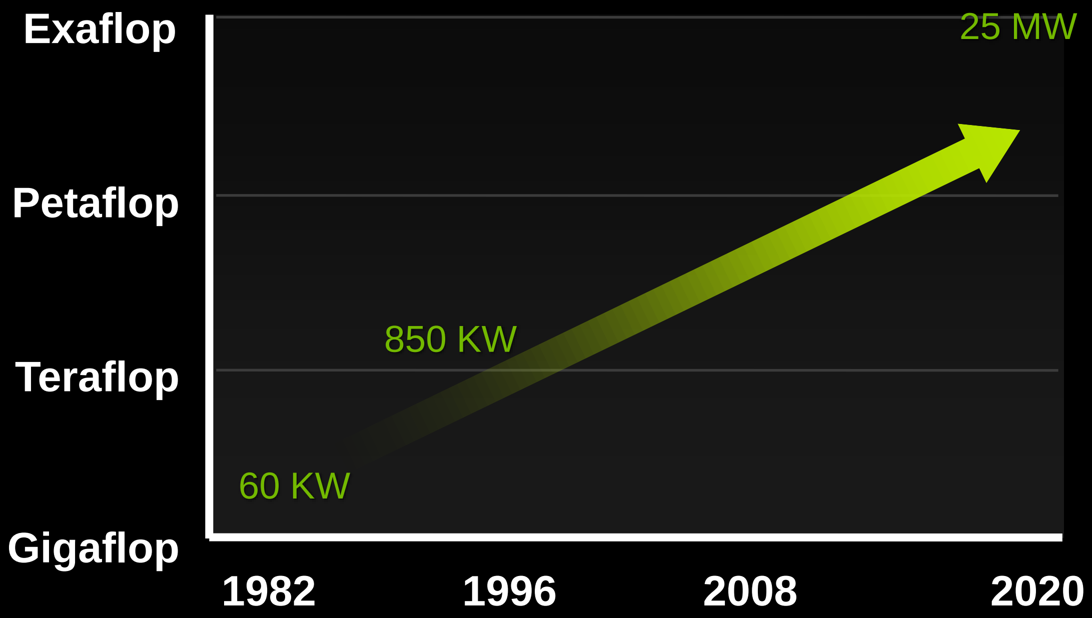


Coupled simulation of entire cells

Simulation of combustion for new high-efficiency, low-emision engines.

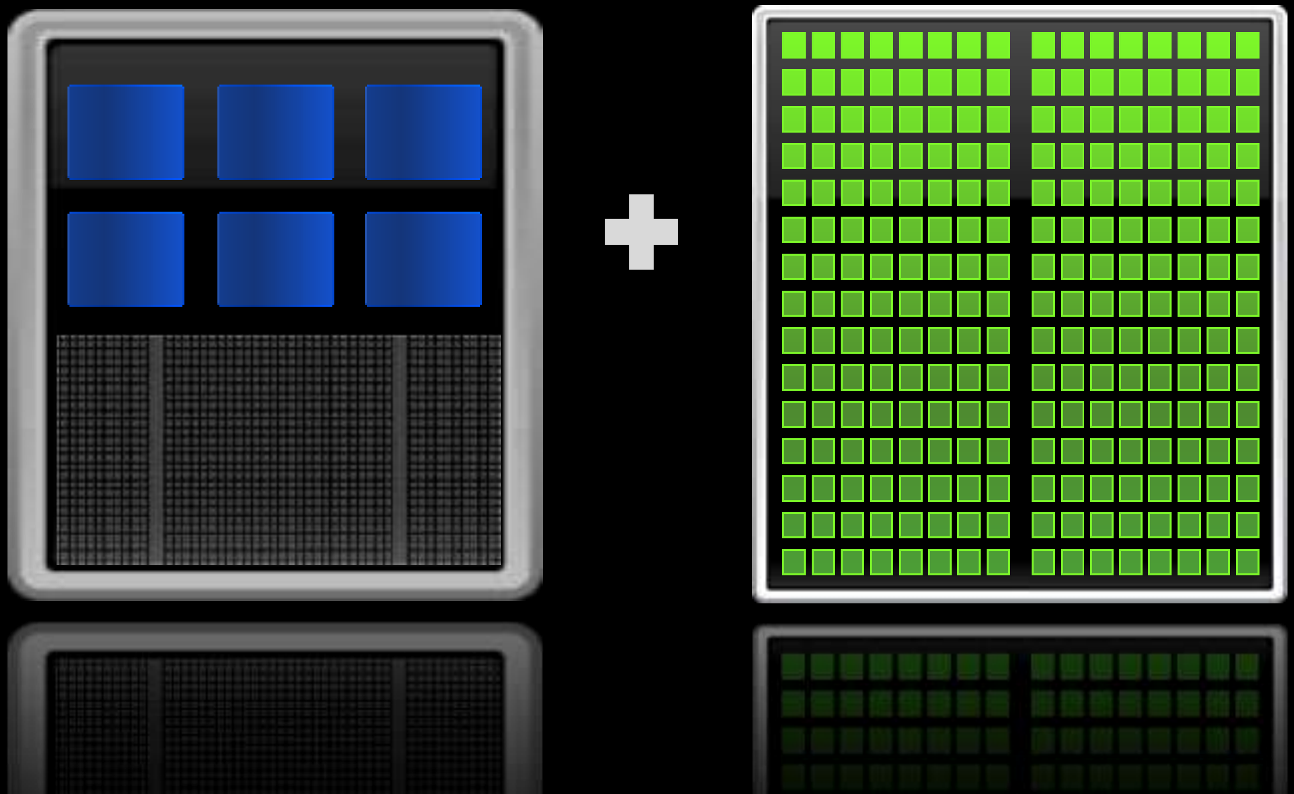
Predictive calculations for supernovae

And the Power Crisis in (Super) Computing



Add GPUs: Accelerate Applications

CPUs: designed to run a few tasks quickly.

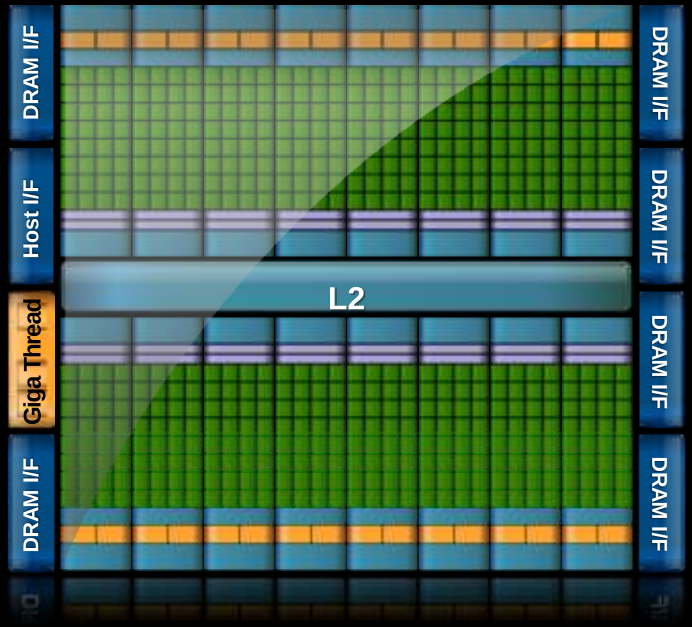


GPUs: designed to run many tasks *efficiently*.

Energy efficient GPU

Performance = Throughput

- Fixed function hardware
  - Transistors are primarily devoted to data processing
  - Less leaky cache
- SIMT thread execution
  - Groups of threads formed into warps which always executing same instruction
  - Some threads become inactive when code path diverges
- Cooperative sharing of units with SIMT
  - eg. fetch instruction on behalf of several threads or read memory location and broadcast to several registers
- Lack of speculation reduces overhead
- Minimal Overhead
  - Hardware managed parallel thread execution and handling of divergence



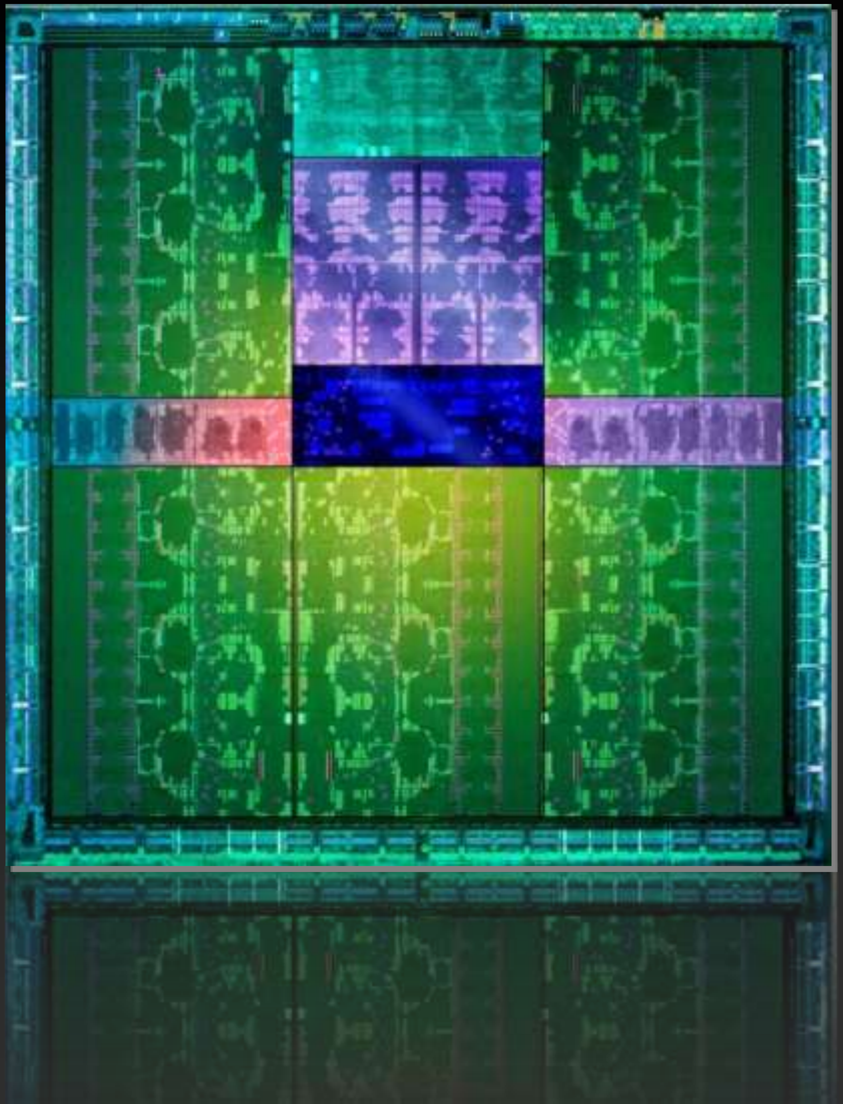
Overarching Goals for GPU Computing

Power
Efficiency

Ease of
Programming
And Portability

Application
Space
Coverage

Power Efficiency



KEPLER

SMX (power efficiency)

Hyper-Q

Dynamic Parallelism

(programmability and application coverage)

Kepler GK110 SMX vs Fermi SM

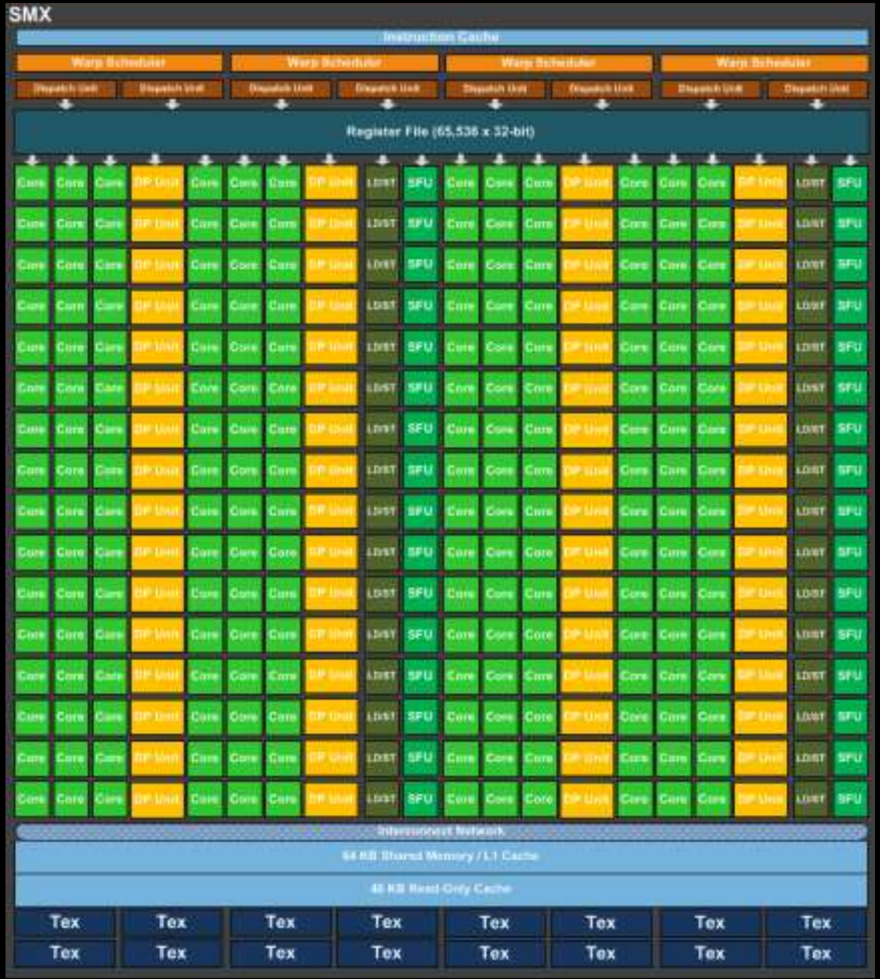
3x sustained perf/W

Ground up redesign for perf/W

6x the SP FP units

4x the DP FP units

Significantly slower FU clocks

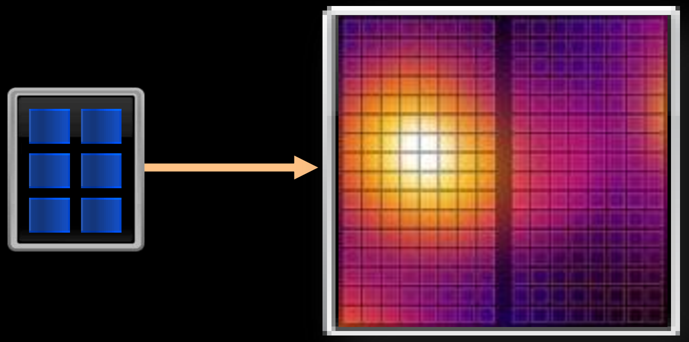


Processors are getting wider, not faster

Better Utilization with Hyper-Q

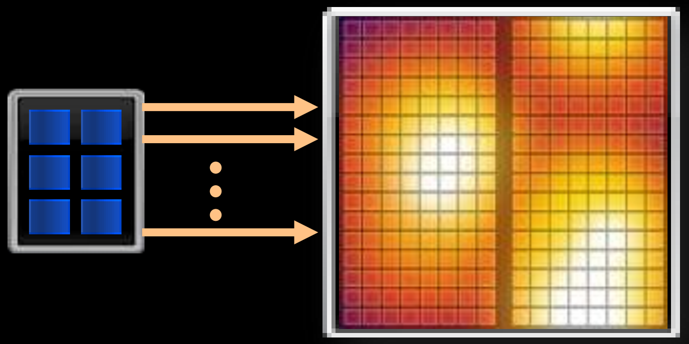
FERMI

1 Work Queue



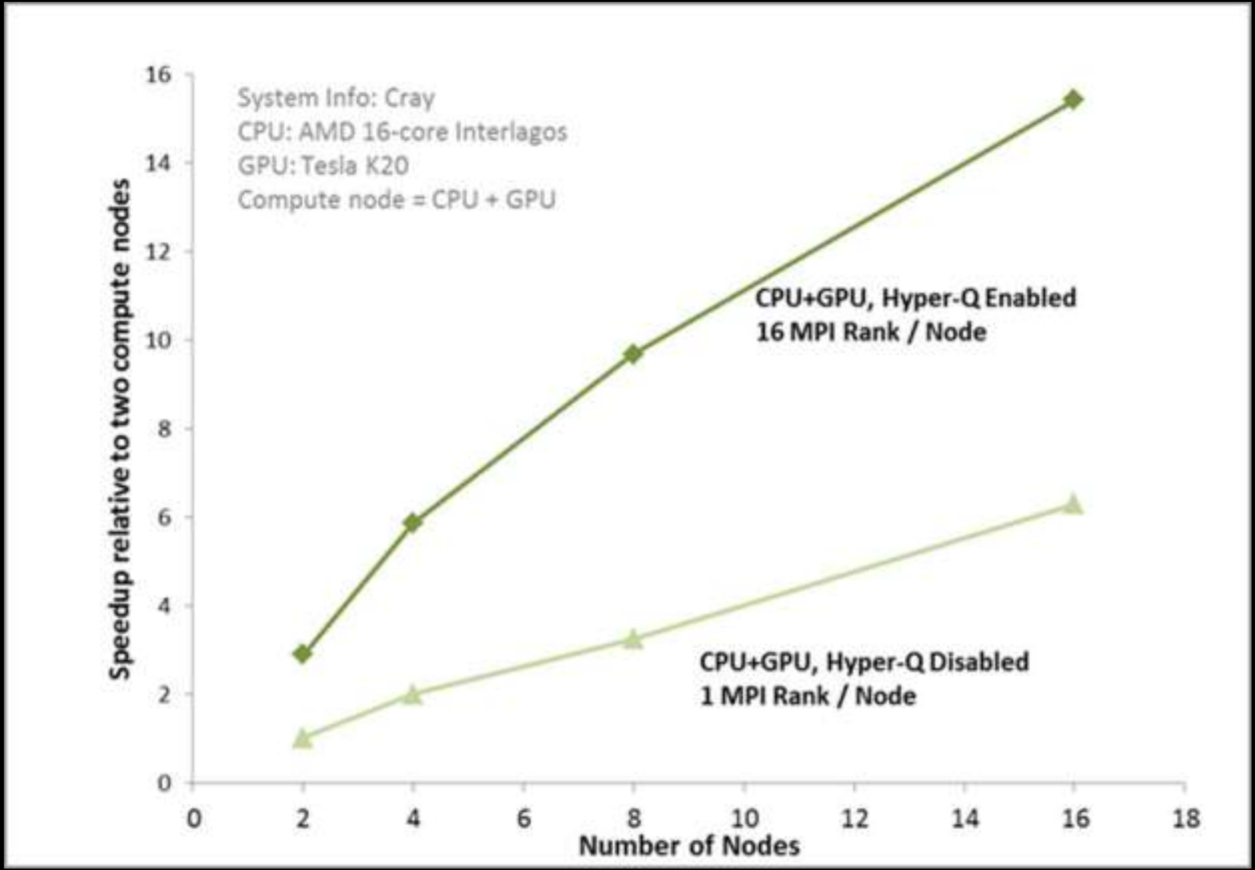
KEPLER

32 Concurrent Work Queues



- **Grid Management Unit selects most appropriate task from up to 32 hardware queues (CUDA streams)**
- **Improves scheduling of concurrently executed grids**
- **Particularly interesting for MPI applications when combined with Multi Process Server, but *not limited to MPI applications***

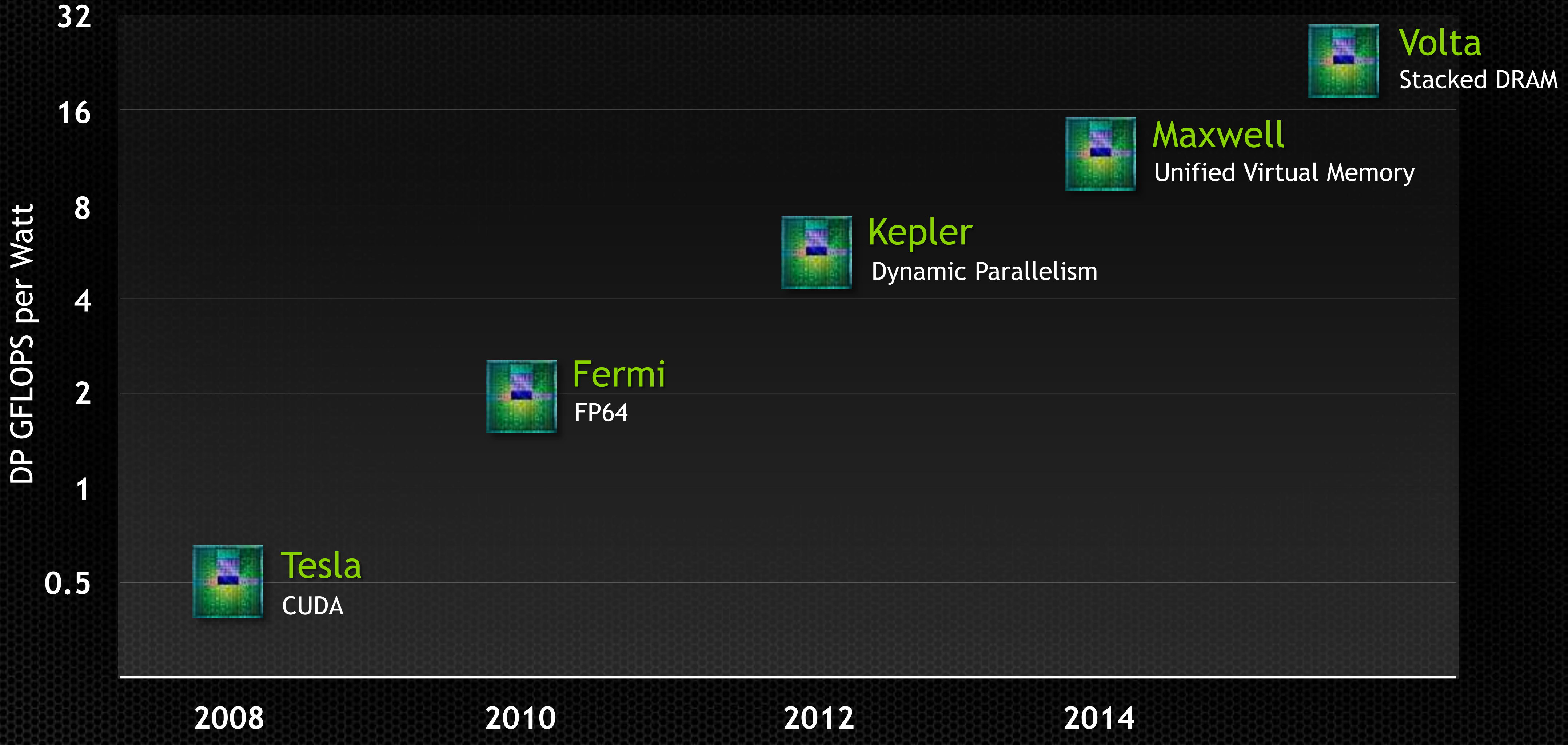
Hyper-Q with Multiple MPI Ranks



- Hyper-Q with multiple MPI ranks leads to 2.5X speedup over single MPI rank using the GPU
- Blog post by Peter Messmer of NVIDIA

http://blogs.nvidia.com/2012/08/unleash-legacy-mpi-codes-with-keplers-hyper-q/

Focus on Power Efficiency

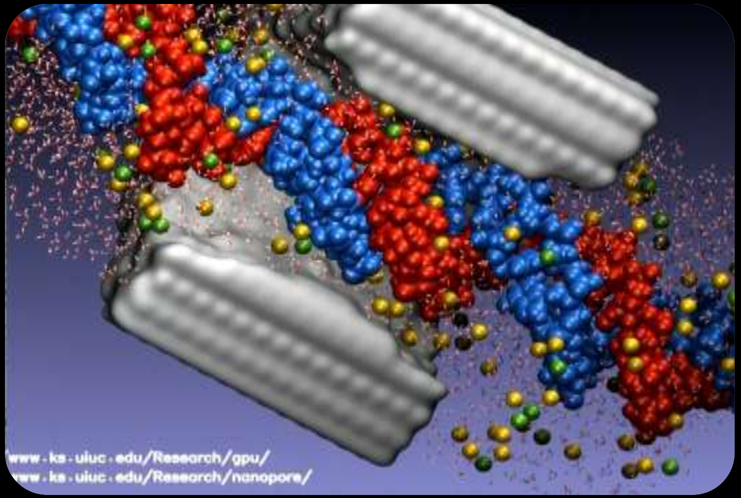


Kayla Development Platform

CUDA 5 | OpenGL 4.3

Kick starts ARM + CUDA Ecosystem

NAMD Ported in 2 Days



https://developer.nvidia.com/kayla-platform

Quad ARM + Kepler GPU

Quad ARM + Any CUDA GPU

Ease of Programming and Portability

Parallel Computing Platform

Multiple Programming Approaches

- Libraries — “Drop-in” Acceleration
- OpenACC Directives — Easily Accelerate Applications
- Programming Languages — Maximum Flexibility

Development Environment

- Parallel Nsight IDE
  Linux, Mac and Windows
  GPU Debugging and Profiling
- CUDA-GDB debugger
  NVIDIA Visual Profiler
- Third Party Tools
  DDT, TotalView,
  Vampir, ...

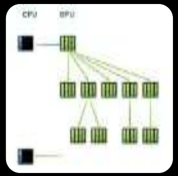
Compiler

- Open Compiler Tools
- Enables compiling new languages to CUDA platform, and CUDA languages to other architectures
- OpenACC Compiler

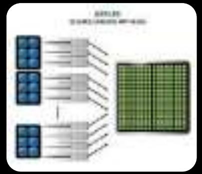
Hardware Capabilities

SMX

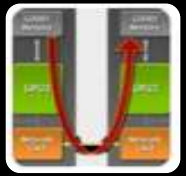
DynamicParallelism



HyperQ



GPUDirect



GPU Accelerated Libraries

“Drop-in” Acceleration for your Applications

Linear Algebra
FFT, BLAS,
SPARSE, Matrix

NVIDIA cuFFT, cuBLAS, cuSPARSE | CULA tools | MAGMA | CUSP

Numerical & Math
RAND, Statistics

IMSL Fortran Numerical Library | NVIDIA Math Lib | ArrayFire | NVIDIA cuRAND

Data Struct. & AI
Sort, Scan, Zero Sum

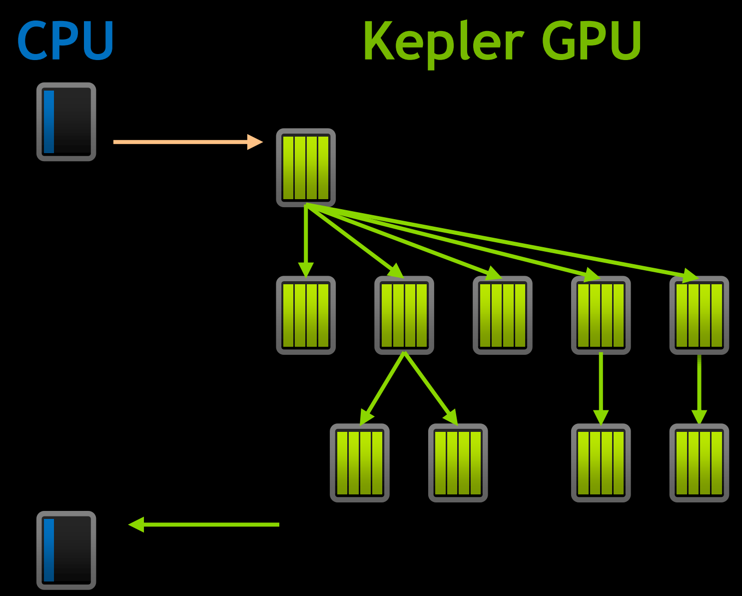
Thrust | GPU AI – Board Games | GPU AI – Path Finding

Visual Processing
Image & Video

NVIDIA NPP | NVIDIA Video Encode | Sundog Software

Dynamic Parallelism

Simpler Code, More General, Higher Performance

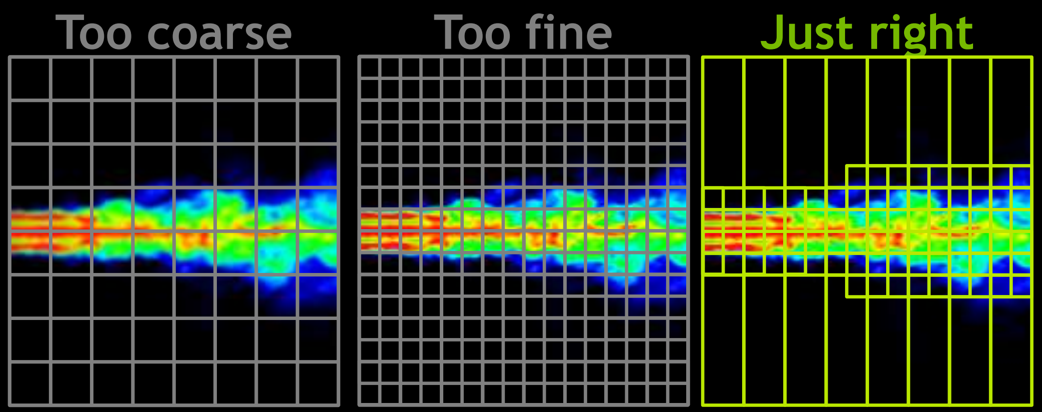


Launch new kernels from the GPU

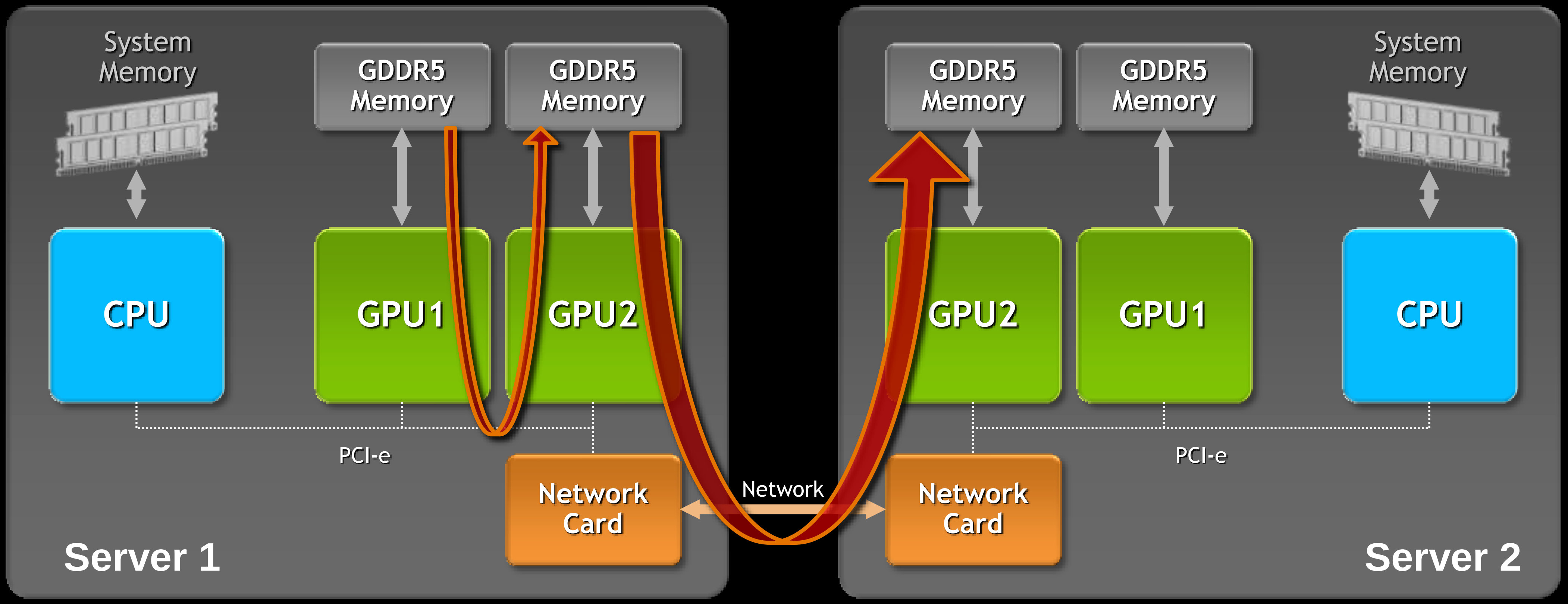
- Dynamically - based on run-time data
- Simultaneously - from multiple threads at once
- Independently - each thread can launch a different grid

Better load balancing for dynamic workloads

- when work-per-block is data-dependent (e.g. Adaptive Mesh CFD)



Kepler Enables Full NVIDIA GPUDirect™ RDMA



NVIDIA GPUDirect™ RDMA

- True RDMA support for GPU memory
  - NIC performs DMA (GPU DMA engines remain available for CUDA use)
- No hardware changes in NIC
- System BIOS should support Large BARs
- GPU and NIC have to be installed on the same IO Hub (QPI doesn't support it)
- GPUDirect ™ RDMA for communication with other PCI devices (eg. Flash memory devices)
  - Requires adopting GPUDirect-Interop API in vendor software stack
  - Documentation "Developing a Linux Kernel Module using RDMA for GPUDirect" is available at http://docs.nvidia.com/cuda/gpudirect-rdma/index.html
- GPUDirect ™ RDMA support available on Tesla and Quadro Kepler class hardware with CUDA 5 and later

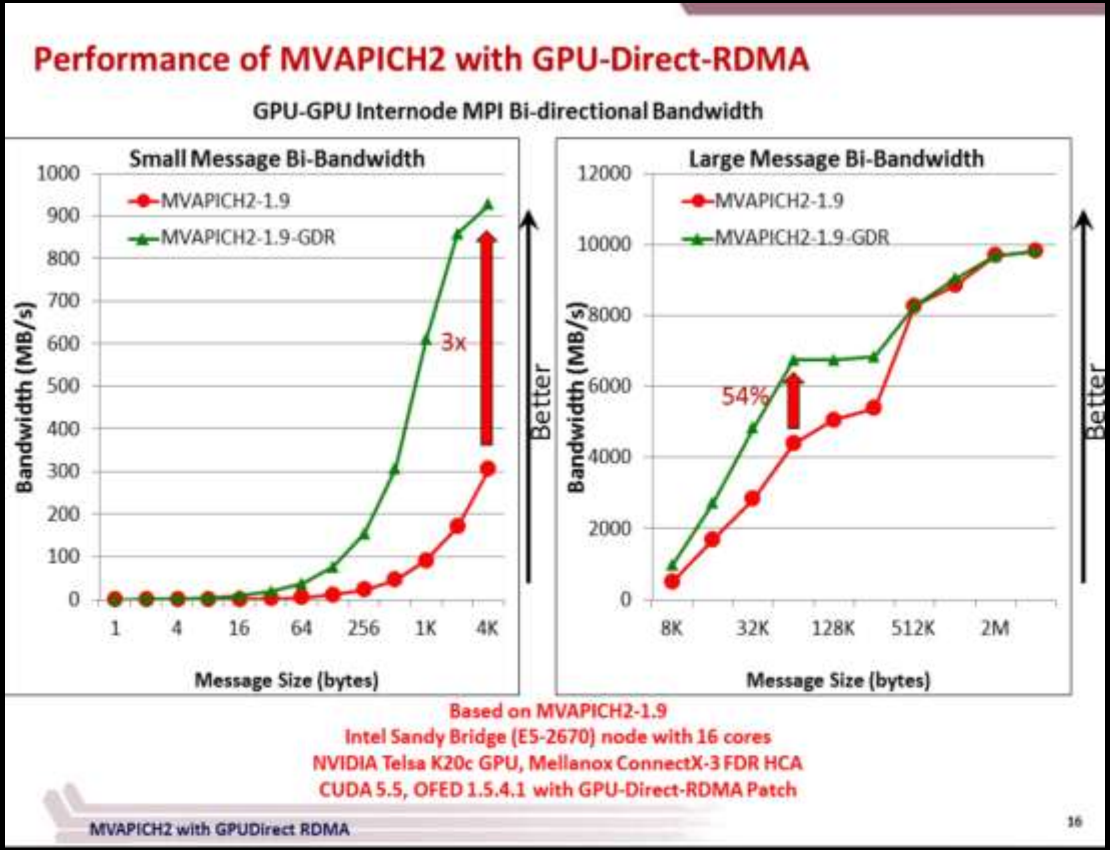
Mellanox Infiniband with GPUDirect™ RDMA

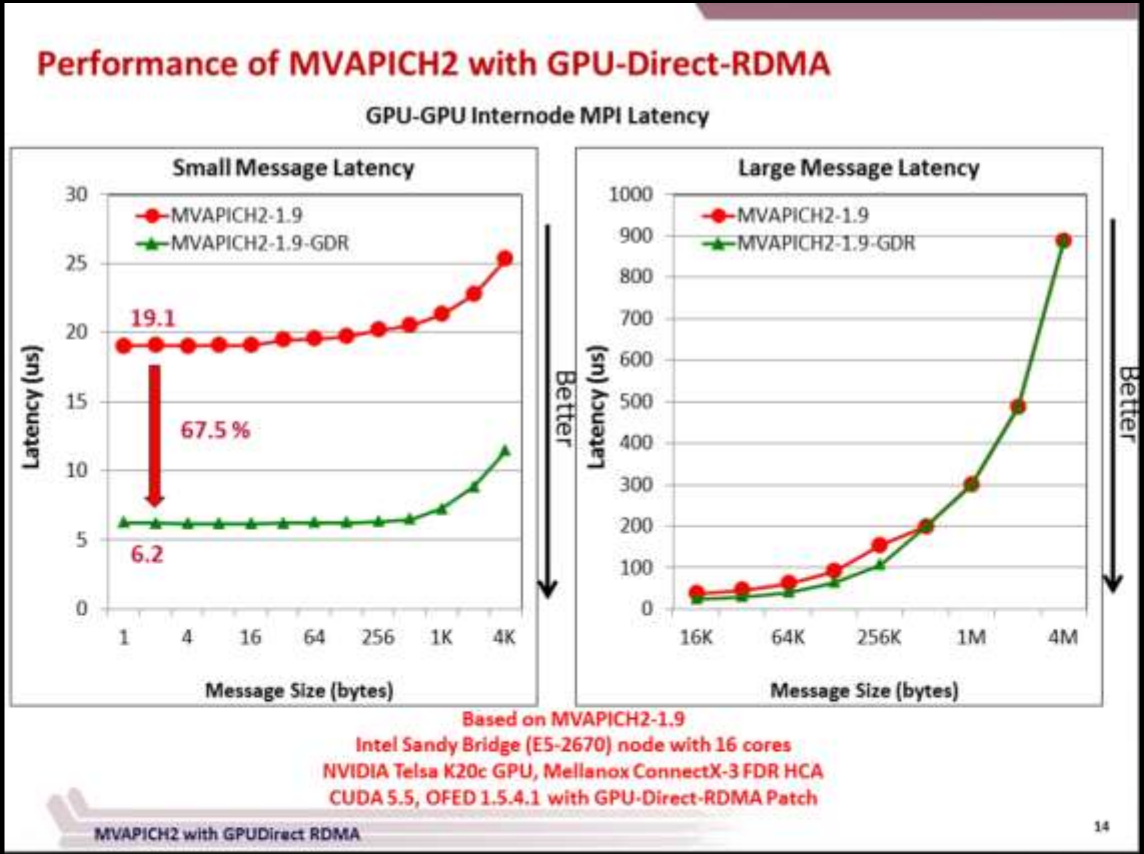
- Alpha release of Mellanox GPUDirect (GDR) MLNX_OFED driver is available
  - Alpha release works with CUDA 5.0 or CUDA 5.5
  - Final release will be based on CUDA 6.0 (Beta later in 2013)
  - Supported on any ConnectX adapter that use the MLX4 driver
- MVAPICH2-GDR (based on MVAPICH2 1.9) release can be used with this IB driver release (see later slides)
- Request Mellanox GDR driver and MVAPICH2-GDR via email to hpc@mellanox.com

MPI support for GPUDirect™ RDMA

- **MVAPICH2-GDR (based on MVAPICH2 1.9) supports GPUDirect™ RDMA**
  - **Hybrid design takes advantage of GPU Direct ™ RDMA**
  - **Uses host based buffered design in current MVAPICH2 for reads (Alleviates Sandybridge chipset bottleneck)**
- **MVAPICH2 team is working on multiple enhancements (collectives, datatypes, one-sided) to exploit the advantages of GPUDirect™ RDMA**
- **As Mellanox GDR driver matures, successive versions of MVAPICH2-GDR with enhancements will be made available to the community**

MVAPICH2 Performance with GPUDirect™ RDMA





Latency

Bi-Directional Bandwidth

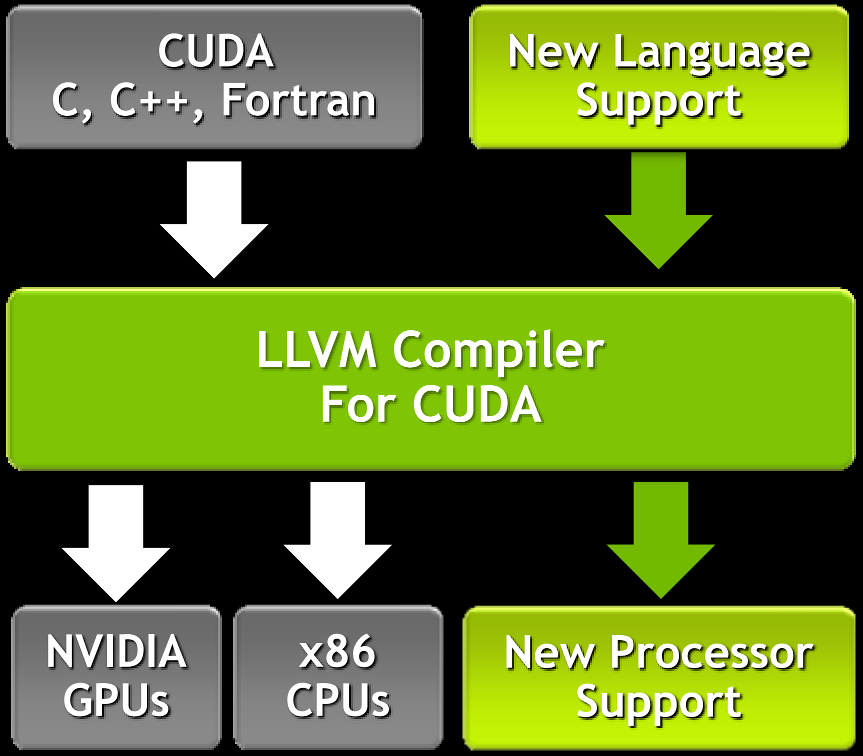
CUDA Compiler Contributed to Open Source LLVM

Developers want to build front-ends for

Java, Python, R, DSLs

Target other processors like

ARM, FPGA, GPUs, x86

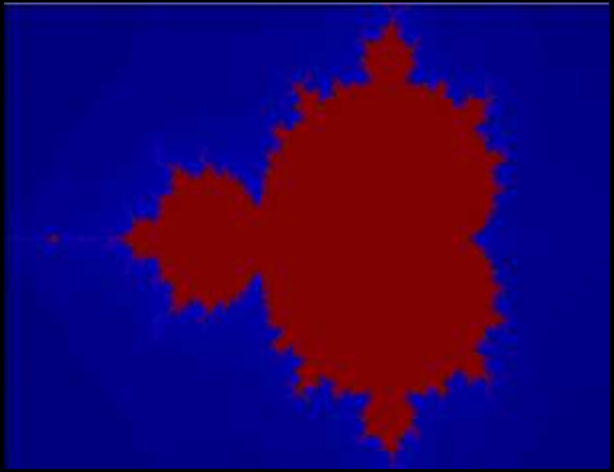


LLVM COMPILER INFRASTRUCTURE

Enabling More Programming Languages

CUDA Python

NumbaPro



```
@cuda.jit(restype=uint8, argtypes=[f8, f8, uint32], device=True)
def mandel(x, y, max_iters):
  zr, zi = 0.0, 0.0
  for i in range(max_iters):
    newzr = (zr*zr-zi*zi)+x
    zi = 2*zr*zi+y
    zr = newzr
    if (zr*zr+zi*zi) >= 4:
       return i
  return 255

@cuda.jit(argtypes=[uint8[:,:], f8, f8, f8, f8, uint32])
def mandel_kernel(img, xmin, xmax ymin, ymax, iters):
  x, y = cuda.grid(2)
  if x < img.shape[0] and y < img.shape[1]:
    img[y, x] = mandel(min_x+x*((max_x-min_x)/img.shape[0]),
                       min_y+y*((max_y-min_y)/img.shape[1]), iters)

gimage = np.zeros((1024, 1024), dtype = np.uint8)
d_image = cuda.to_device(gimage)
mandel_kernel[(32,32), (32,32)](d_image, -2.0, 1.0, -1.0, 1.0, 20)
d_image.to_host()
```

CUDA Programming, Python Syntax

Domain-specific Languages

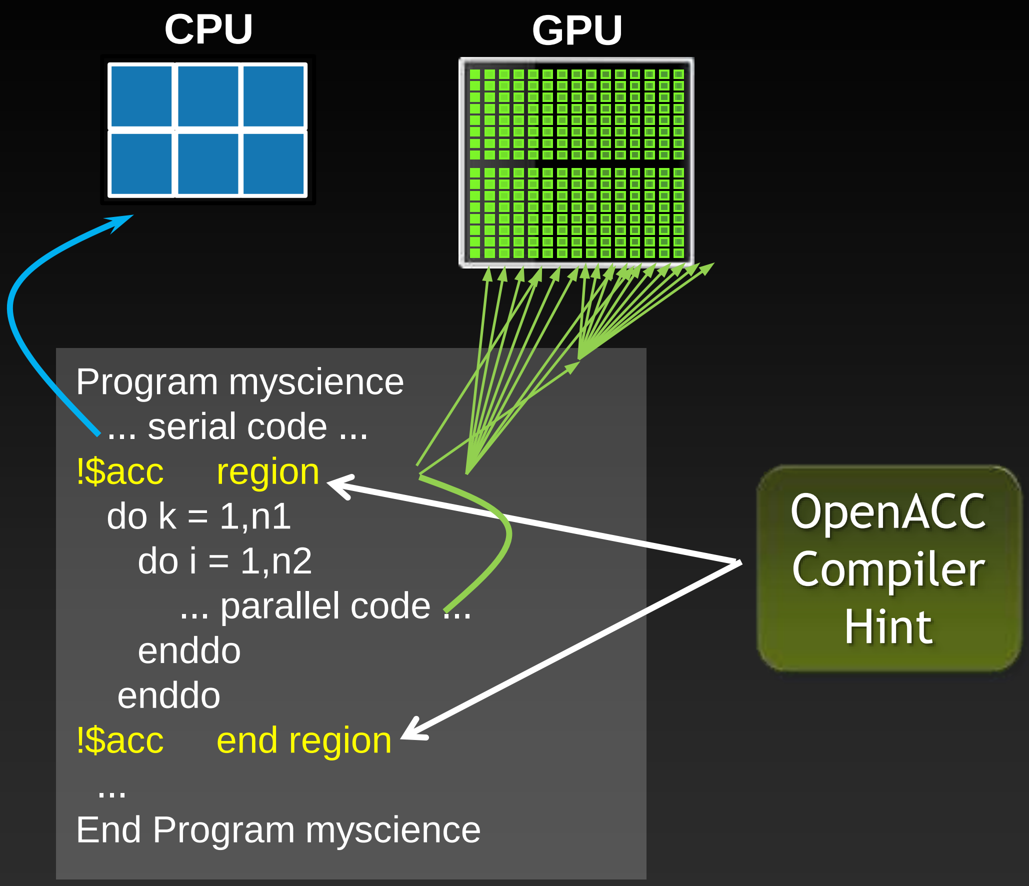
MATLAB

R Statistical
Computing Language

Liszt

A DSL for solving
mesh-based PDEs

OpenACC Directives



Your original Fortran or C code

Easy, Open, Powerful

- Simple Compiler hints
- Works on multicore CPUs & many core GPUs
- Compiler Parallelizes code

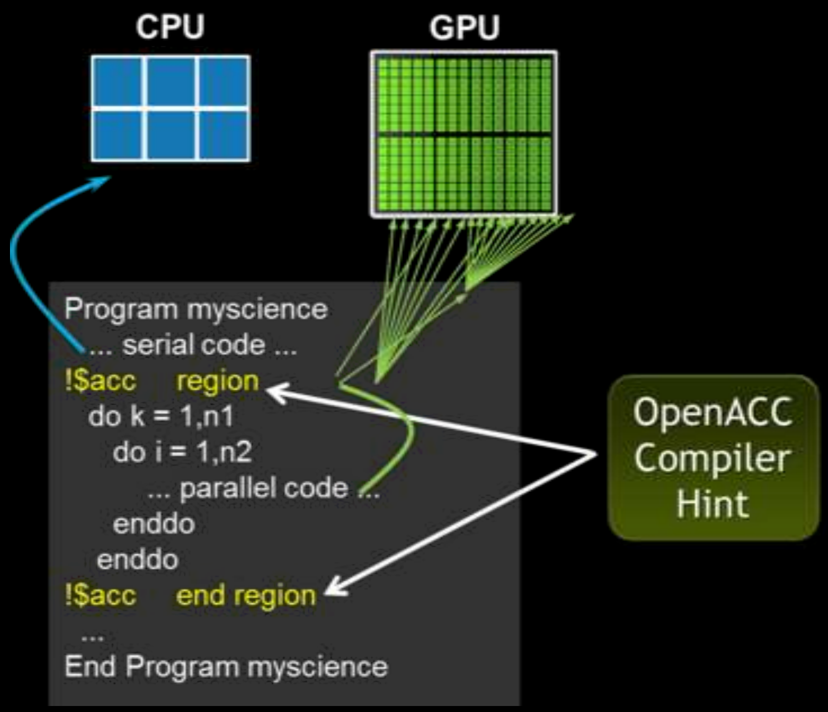
http://www.openacc.org

NVIDIA. CRAY THE SUPERCOMPUTER COMPANY PGI CAPS

Additions for OpenACC 2.0

- Procedure calls
- Separate compilation
- Nested parallelism
- Device-specific tuning, multiple devices
- Data management features and global data
- Multiple host thread support
- Loop directive additions
- Asynchronous behavior additions
- New API routines for target platforms

    (CUDA, OpenCL, Intel Coprocessor Offload Infrastructure)



See http://www.openacc.org/sites/default/files/OpenACC-2.0-draft.pdf

Application Space Coverage

Wide Adoption of Tesla GPUs

Oil and gas

Reverse Time Migration
Kirchoff Time Migration
Reservoir Sim

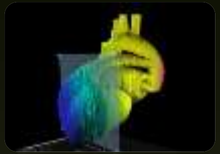
Edu/Research

Astrophysics
Lattice QCD
Molecular Dynamics
Weather / Climate Modeling

Government

Signal Processing
Satellite Imaging
Video Analytics
Synthetic Aperture Radar

Life Sciences



Bio-chemistry
Bio-informatics
Material Science
Sequence Analysis
Genomics

Finance

Risk Analytics
Monte Carlo
Options Pricing
Insurance modeling

Manufacturing

Structural Mechanics
Computational Fluid Dynamics
Machine Vision
Electromagnetics

Catalog GPU-Accelerated Applications

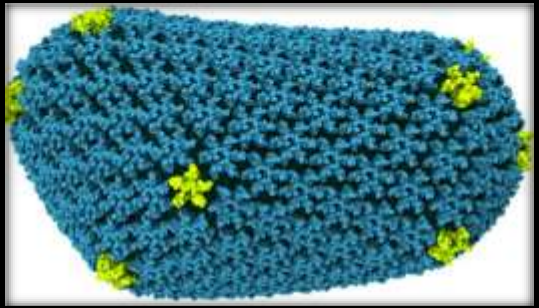
http://www.nvidia.com/teslaapps/

POPULAR GPU-ACCELERATED APPLICATIONS, Continued

Application	Description	Supported Features	Expected Speed Up*	Multi-GPU Support	Release Status
Physics					
Chroma	General purpose LQCD application	Wilson-clover fermions, Krylov solvers, Domain-decomposition	5-6x	Yes	Available now
MILC	General purpose LQCD application	Staggered fermions, Krylov solvers, Gauge-link fattening	5-6x	Yes	Available now
Computational Fluid Dynamics					
Altair AcuSolve	General purpose CFD flow solver	Linear equation solver	2x	Yes	Available now
Autodesk Moldflow	Optimize design of plastic parts and injection molds	Linear equation solver	2x	Single Only	Available now
FEFLO (GMU-Lohner)	Navier-Stokes flow solver based on unstructured grids for modeling both compressible and incompressible flows	Explicit solver	10x	Yes	In Development
FluiDyna LBultra	Computing physical flows in and around solid bodies	LBM, particle CFD	20x	Yes	Available now
FluiDyna Culises-OpenFOAM	Computing physical flows with Culises — a software library with special algorithms for solving systems of equations	Linear equation solvers	3x Solver	Single Only	Available now
Prometech Particleworks	Fluid simulation for free surface flow like Tsunami, material processing and liquids	MPS, Particle CFD	4x-9x	Yes	Available now
S3D (Sandia NL S3D)	Massively parallel direct numerical solver (DNS) for the full compressible Navier-Stokes	Chemistry kernel	8x SP, 5x DP kernel	Yes	In Development
Turbostream	Ultrafast CFD solver for turbomachines	Explicit solver	19x	Yes	Available now
Vratis SpeedIT-OpenFOAM Solver	Set of accelerated solvers for sparse linear systems of equations	Linear equation solvers	3x Solver	Yes	Available now
Computational Structural Mechanics					
Abaqus/Standard	Simulation and analysis tool for structural mechanics	Linear equation solver	1.5-2.5x	Single Only	Available now
ANSYS Mechanical	Simulation and analysis tool for structural mechanics	Linear equation solver	2x	Single Only	Available now
Impetus Afea	Predicts large deformations of structures and components exposed to extreme loading conditions	Linear equation solver, SPH	10x SPH, 2x Total	Yes	Available now
LS-DYNA Implicit	Multiphysics simulation package used	Linear equation solver	3x	Yes	In Development
MSC Nastran	Simulation and analysis tool for structural mechanics	Linear equation solver	1.4-2x	Yes	Available now
Marc	Simulation and analysis tool for structural mechanics	Linear equation solver	1.5x	Yes	In Development
RADIOSS Implicit	Used to maximize durability, NVH, crash, safety, manufacturability and fluid-structure interaction performance	Linear equation solver	2x	Single Only	In Development

Recent Scientific Breakthroughs using GPUs

Breakthrough in HIV research



Discover the chemical structure of HIV's capsid to build more effective drugs

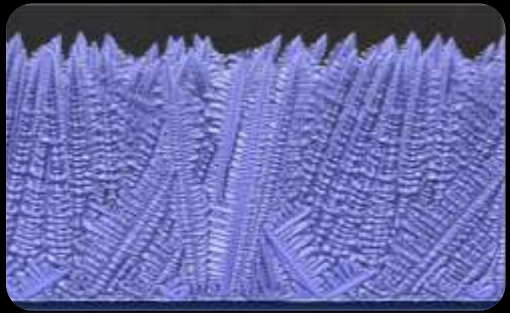
Run at NCSA Blue Waters (3000 GPUs)

Fastest simulation for Silicon for Solar Cells

More efficient & cost-effective solar cells

1.87 Petaflop / sec perf on 7168 GPUs on Tianhe-1A,

Gordon Bell Prize Stronger, Lighter Metals

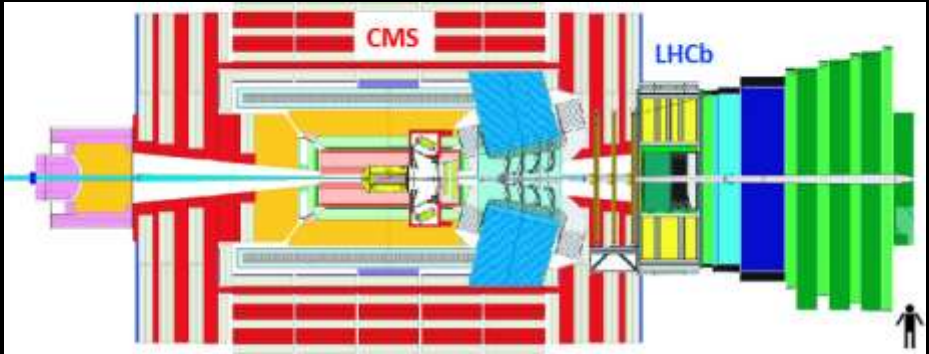


Lighter, Stronger Metals for More Fuel-Efficient Cars

4224 GPUs at Tokyo Tech, Japan

GPUs for control systems

- GPUs are used in many experiments for controlling
  - Examples:
  - Triggering and tracking for CERN experiments
  - Signal processing for Lofar or Square Kilometre Array (SKA)



GPUs and Big Data Analytics

- GPUs Today
  - Computational acceleration for Big Data
  - Visualization
  - Accelerating the Cloud + Mobile transformation
- GPUs Tomorrow
  - Converged architecture for Big Data and Compute

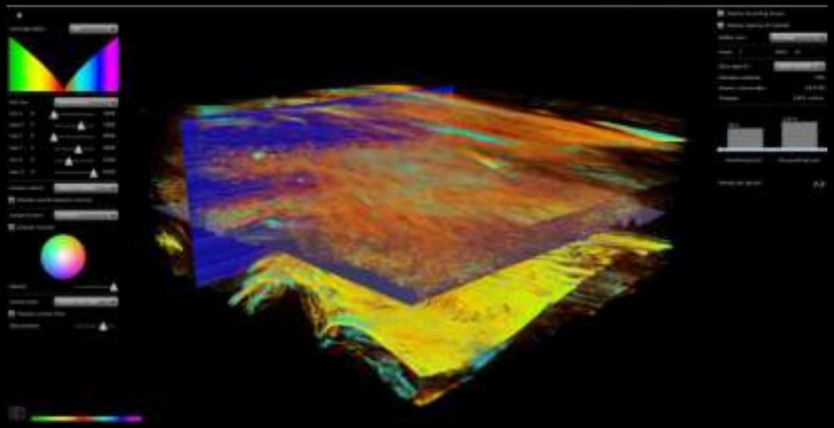
Analyzing Twitter — salesforce

Searching Audio — shazam

Visual Shopping — CORTEXICA vision systems, ebay

Real-time Video Delivery — ELEMENTAL, ESPN

Integration of Compute and Visualisation

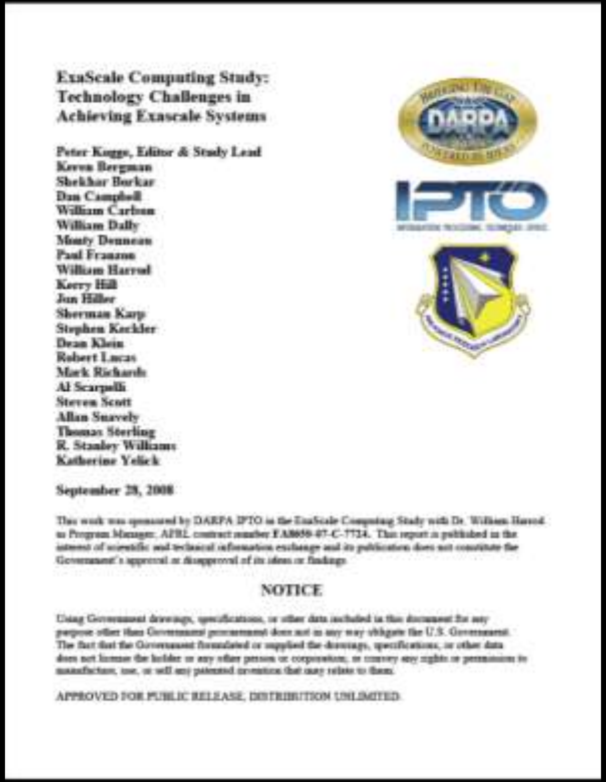


- GPU Operation Mode “All_On” enables graphics capabilities for K20/K20X server GPUs
  - nvidia-smi --gom=0

- NVIDIA indeX - Scalable Big Data Visualization
- Remote visualization tools like ParaView

The Future

DARPA Study Identifies Four Challenges for ExaScale Computing


ExaScale Computing Study:
Technology Challenges in
Achieving Exascale Systems

Peter Kogge, Editor & Study Lead
Keren Bergman
Shekhar Borkar
Dan Campbell
William Carlson
William Dally
Monty Denneau
Paul Franzon
William Harrod
Kerry Hill
Jon Hiller
Sherman Karp
Stephen Keckler
Dean Klein
Robert Lucas
Mark Richards
Al Scarpelli
Steven Scott
Allan Snavely
Thomas Sterling
R. Stanley Williams
Katherine Yelick

September 28, 2008

This work was sponsored by DARPA IPTO in the ExaScale Computing Study with Dr. William Harrod as Program Manager; AFRL contract number FA8650-07-C-7724. This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

NOTICE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.

Available at
www.darpa.mil/ipto/personnel/docs/ExaScale_Study_Initial.pdf

Report published September 28, 2008:

- **Four Major Challenges**
  - Energy and Power challenge
  - Memory and Storage challenge
  - Concurrency and Locality challenge
  - Resiliency challenge
- **Number one issue is *power***
  - Extrapolations of current architectures and technology indicate over 100MW for an Exaflop!
  - Power also constrains what we can put on a chip

Which Takes More Energy?

Performing a 64-bit floating-point FMA:

	893,500.288914668
×	43.90230564772498
=	39,226,722.78026233027699
+	2.02789331400154
=	39,226,724.80815564

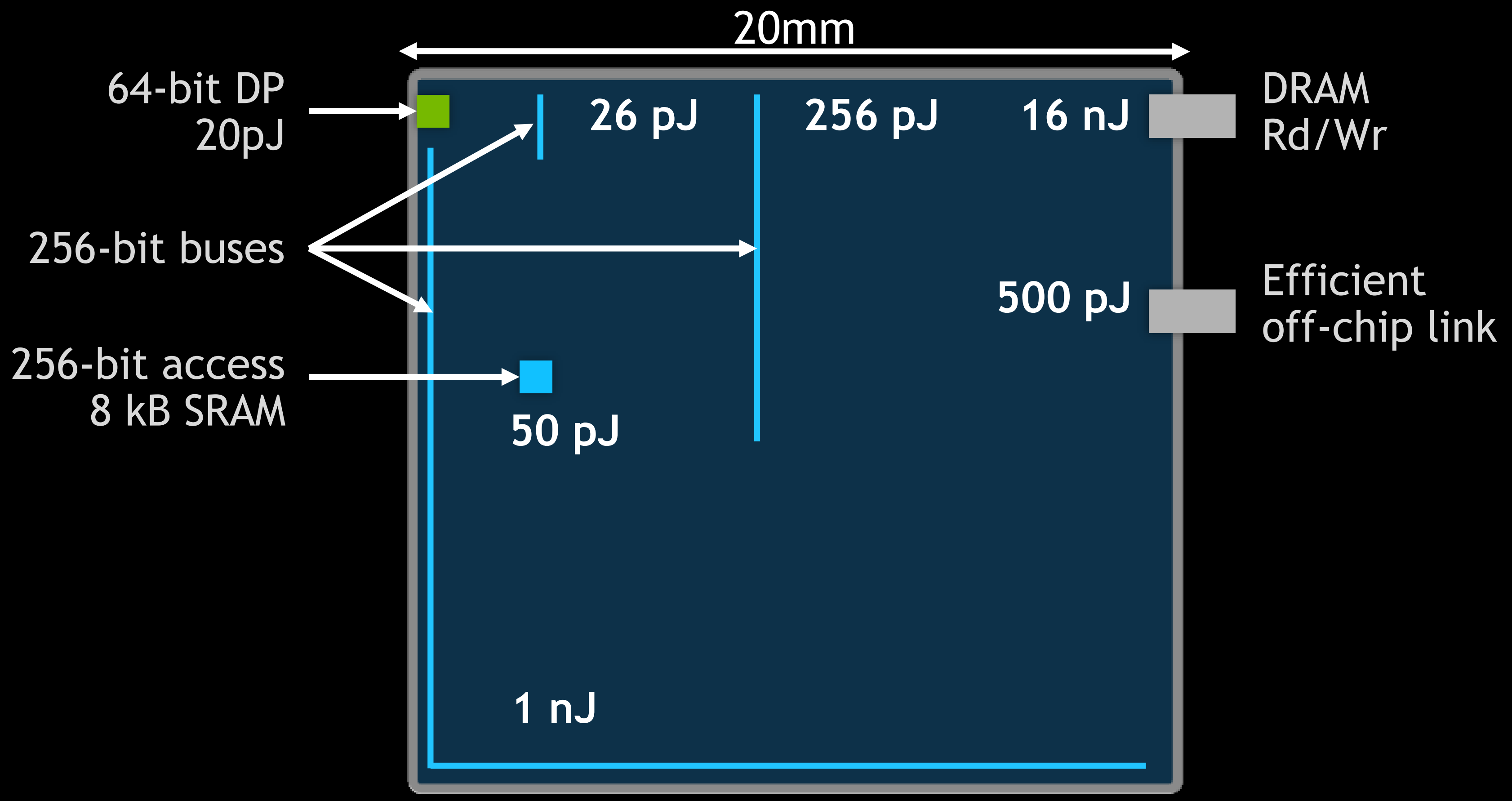
Or moving the three 64-bit operands 20 mm across the die:

This one takes over 4.7x the energy today (40nm)!

It’s getting worse: in10nm, relative cost will be 17x!

Loading the data from off chip takes >> 100x the energy.

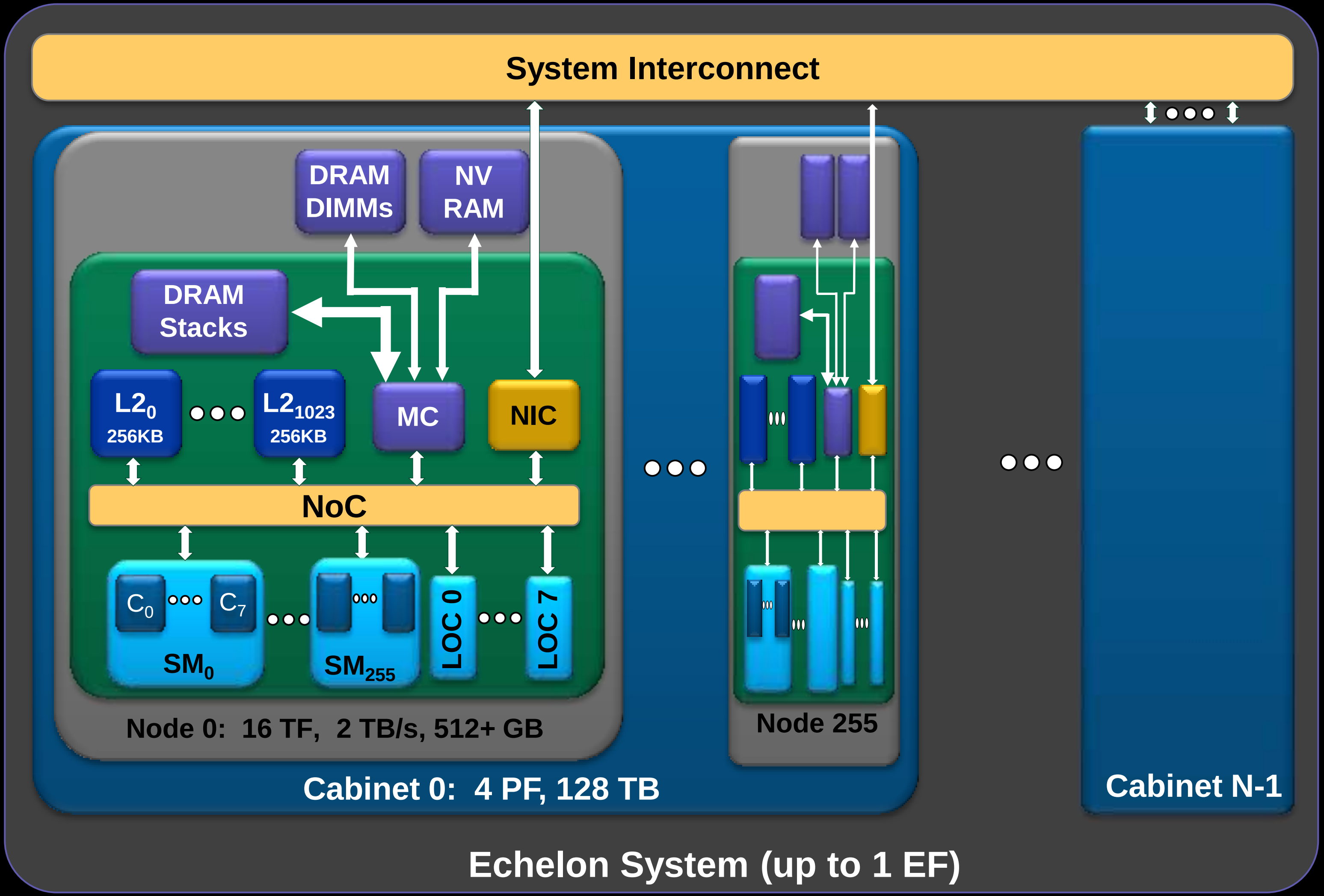
Communication Takes More Energy Than Arithmetic



What is important for the future?

- Its not about the FLOPS
- Its about data movements
- Algorithms should be designed to perform more work per unit data movement
- Programming systems should further optimize this data movement
- Architectures should facilitate this by providing an exposed hierarchy and efficient communication

2018 Vision: 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

Conclusion

- **Power is the main HPC constraint**
  - Vast majority of work *must* be done by cores designed for efficiency
- **NVIDIA GPU's are already designed for energy efficiency**
- **Data movement dominates the power**
  - **Locality at all levels and reduction of overhead is necessary**
- **GPU computing has a sustainable model**
  - Aligned with technology trends, supported by consumer markets
- **GPUs are the path to the tightly-coupled hybrid processor future**

Thank you.

Axel Koehler
akoehler@nvidia.com