NVIDIA GPUs in Earth System Modelling

Thomas Bradley

Agenda: GPU Developments for CWO

- Motivation for GPUs in CWO
- Parallelisation Considerations
- GPU Technology Roadmap

MOTIVATION FOR GPUS IN CWO

NVIDIA GPUs Power 3 of Top 5 Supercomputers

#2 : Tianhe-1A
7168 Tesla GPU's 2.5 PFLOPS

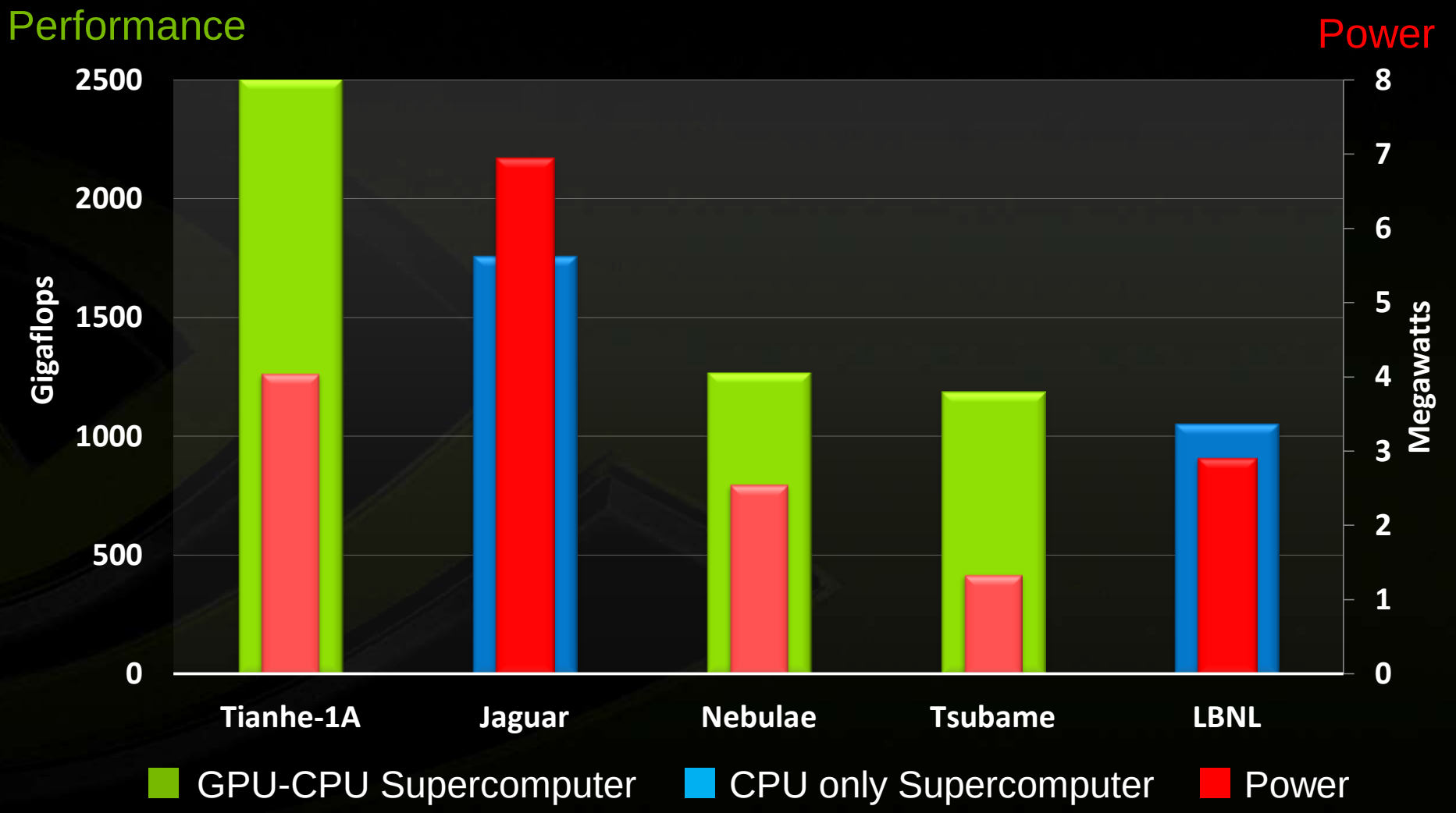
#4 : Nebulae
4650 Tesla GPU's 1.2 PFLOPS

#5 : Tsubame 2.0
4224 Tesla GPUs 1.194 PFLOPS

"We not only created the world's fastest computer, but also implemented a heterogeneous computing architecture incorporating CPU and GPU, this is a new innovation."

Premier Wen Jiabao
Public comments acknowledging Tianhe-1A

GPU Systems: More Power Efficient (for HPL)



Comparison with Top Supercomputer K in Japan

K Computer: Custom SPARC Processors

8.1 PetaFlop

68,500 CPUs

672 Racks

10 Megawatt

$700 Million

2.3x better flops/rack

1.06x better flop/watt

2.6x better $/flop

Tsubame: Intel CPUs + NVIDIA Tesla

1.2 PetaFlop

2K CPUs, 4K GPUs

44 Racks

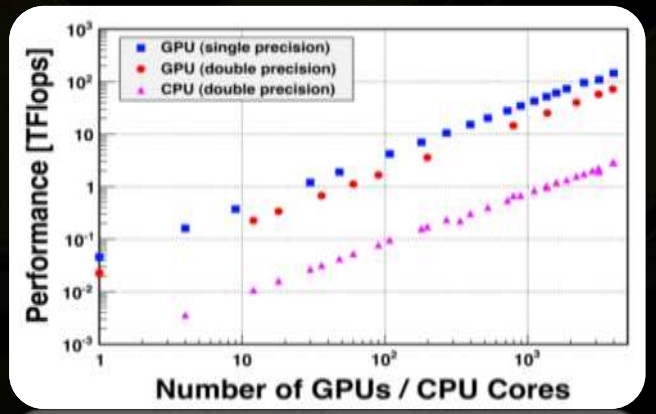
1.4 Megawatt

$40 Million

Real Science on GPUs: ASUCA NWP on Tsubame

Tsubame 2.0
Tokyo Institute of Technology

- 1.19 Petaflops
- 4,224 Tesla M2050 GPUs



3990 Tesla M2050s
145.0 Tflops SP
76.1 Tflops DP

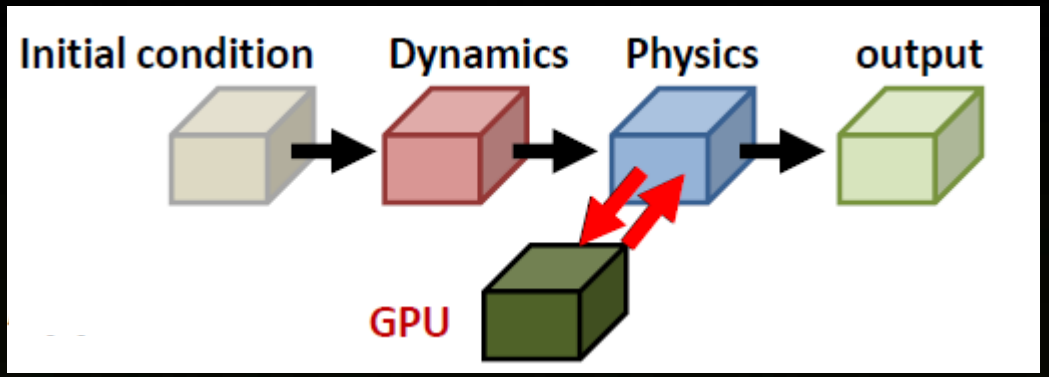
Simulation on Tsubame 2.0, TiTech Supercomputer

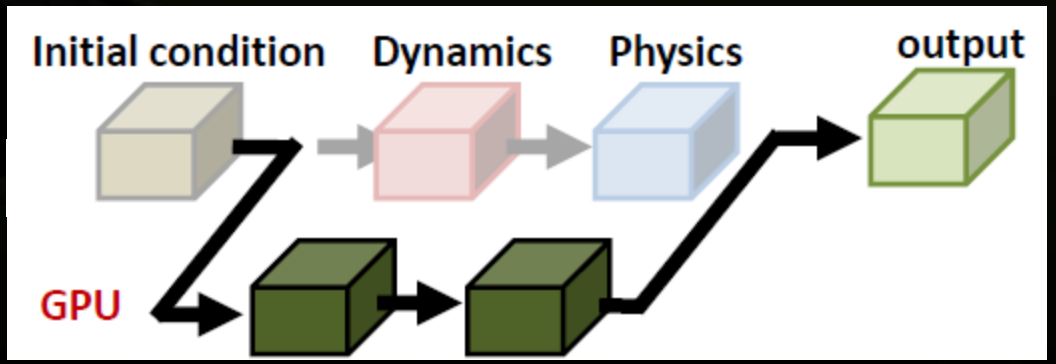
CWO Performance: Full GPU Approach

Physics Only
- WRF
- COSMO (1)

Dynamics Only
- COSMO (2)
- ICON
- NIM (sample physics)
- CAM
- HOMME
- HIRLAM

Full GPU Approach
- ASUCA
- GEOS-5
- GRAPE





NVIDIA Features GPUs at Conferences

Supercomputing 2010 | Nov 2010 | New Orleans, LA

COSMO: GPU Considerations for Next Generation Weather Simulations
Thomas Schulthess, Swiss National Supercomputing Centre (CSCS)

ASUCA: Full GPU Implementation of Weather Prediction Code on TSUBAME Supercomputer
Takayuki Aoki, GSIC of Tokyo Institute of Technology (TiTech)

NIM: Using GPUs to Run Next-Generation Weather Models
Mark Govett, National Oceanic and Atmospheric Administration (NOAA)

***BoF*:** GPUs and Numerical Weather Prediction *(organized by CSCS and NVIDIA)*
Featured organizations: TiTech (ASUCA), NASA (GEOS-5), NOAA (NIM), Cray, PGI

NVIDIA GPU Technology Conference | Sep 2010 | San Jose, CA

GPU TECHNOLOGY CONFERENCE

ASUCA: Full GPU Implementation of Weather Prediction Code on TSUBAME Supercomputer
Takayuki Aoki, GSIC of Tokyo Institute of Technology (TiTech)

NIM: Using GPUs to Run Next-Generation Weather Models
Mark Govett, National Oceanic and Atmospheric Administration (NOAA)

MITgcm: Designing a Geoscience Accelerator Library Accessible from High Level Languages
Chris Hill, Massachusetts Institute of Technology (MIT)

PARALLELISATION CONSIDERATIONS

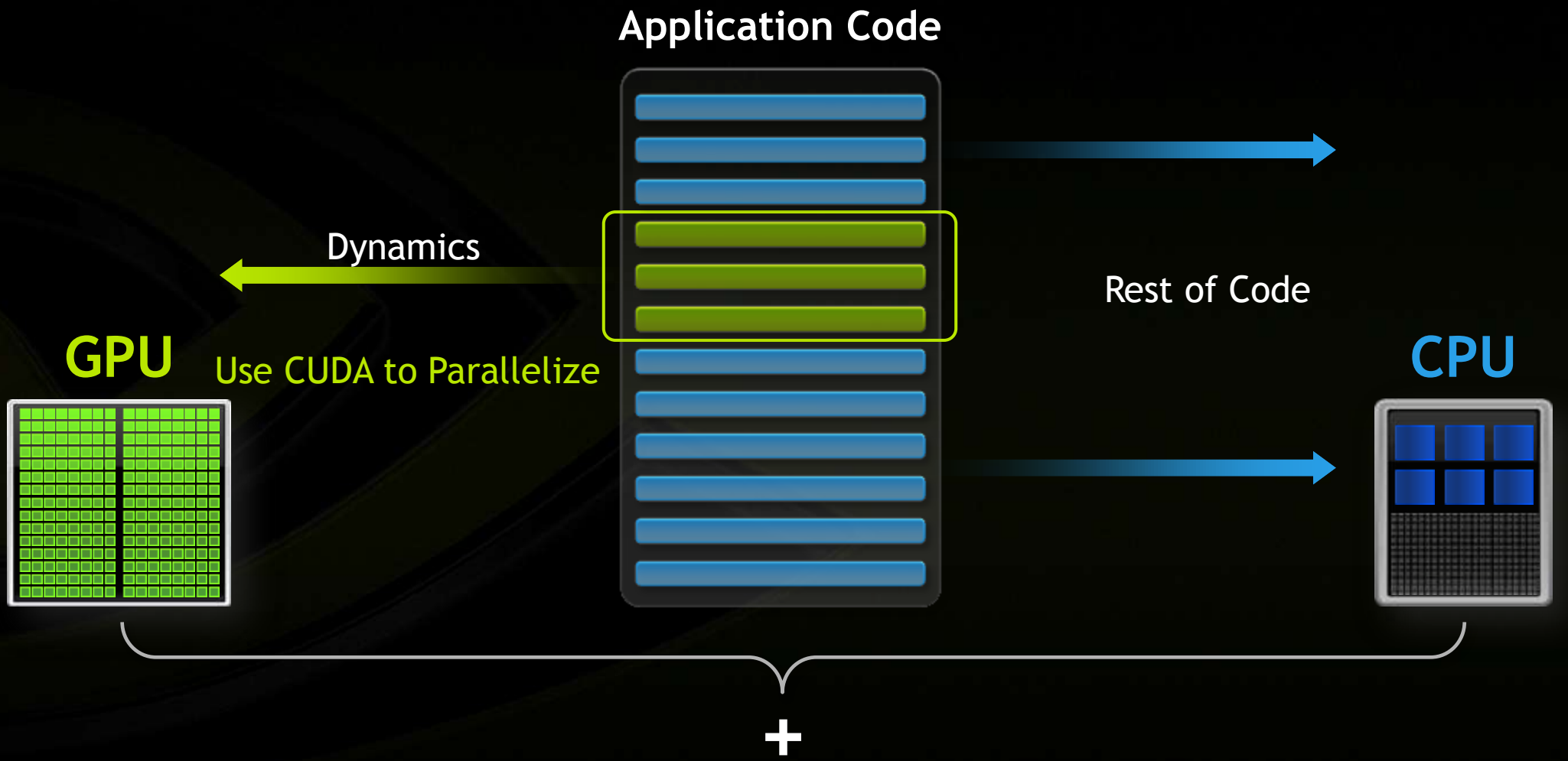
GPU Considerations for CWO Codes

- Initial efforts are mostly implicit linear solvers on GPU
  - If linear solver ~50% of profile time – only 2x speed-up is possible
  - More of application must be moved to GPUs for additional benefit
  - *Explicit schemes* – no linear algebra, no solver, operations on stencil
- Most codes are parallel and scale across multiple CPU cores
  - Multi-core CPUs can contribute to parallel matrix assembly, others
- Most codes use a domain decomposition parallel method
  - Fits GPU model very well and preserves costly MPI investment

Options for Parallel Programming of GPUs

Approach	Examples
Applications	MATLAB, Mathematica, LabVIEW
Libraries	FFT, BLAS, SPARSE, RNG, IMSL, CUSP, etc.
Directives	PGI Accelerator, HMPP, Cray, F2C-Acc
Wrappers	PyCUDA, CUDA.NET, jCUDA
Languages	CUDA C/C++, PGI CUDA Fortran, GPU.net
APIs	CUDA C/C++, OpenCL

Most Implementations Focus on Dynamical Core



Sparse Iterative Solvers for Dynamics

- Sparse-matrix vector multiply (SpMV) & BLAS1
  - Memory-bound
- GPU can deliver good SpMV performance
  - ~10-20 Gflops for unstructured matrices in double precision
- Best sparse matrix data structure on GPU different from CPU
  - Explore for your specific case
- A massively parallel preconditioner is key:

  Lectures: Jon Cohen at IMA Workshop: "Thinking parallel: sparse iterative solvers with CUDA"

  Nathan Bell (4-parts) at PASI: "Iterative methods for sparse linear systems on GPU"

Typical Sparse Matrix Formats

(DIA) Diagonal

(ELL) ELLPACK

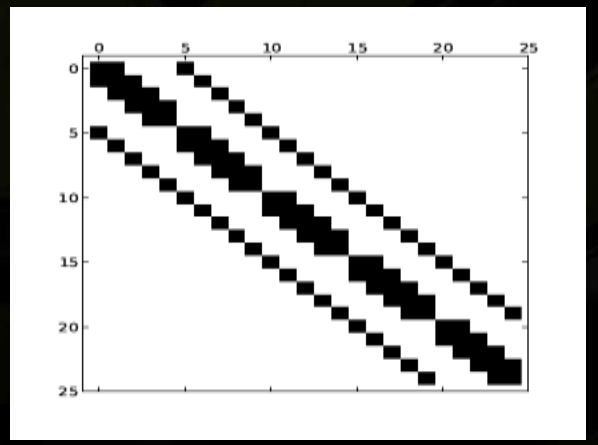
(CSR) Compressed Row

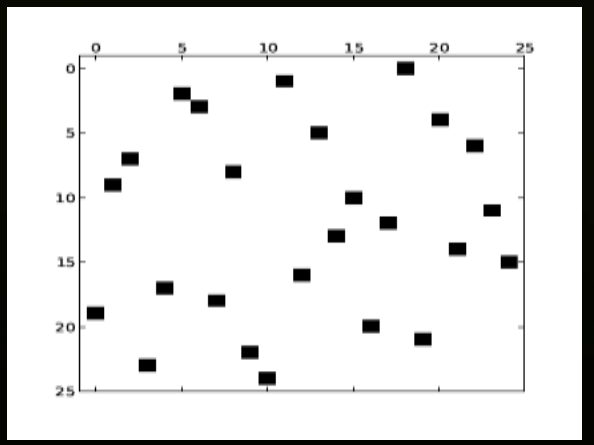
(HYB) Hybrid

(COO) Coordinate

Structured

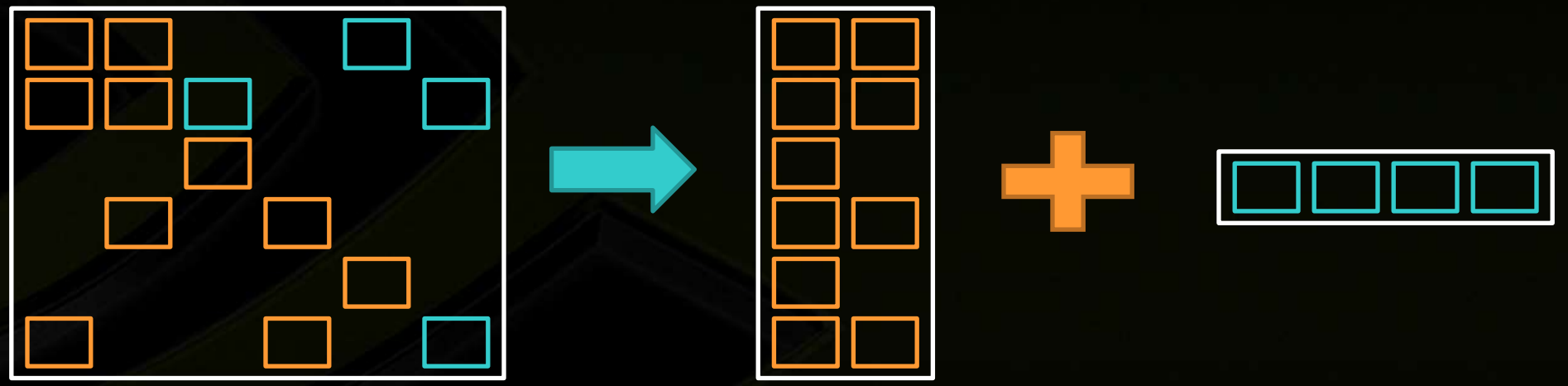
Unstructured





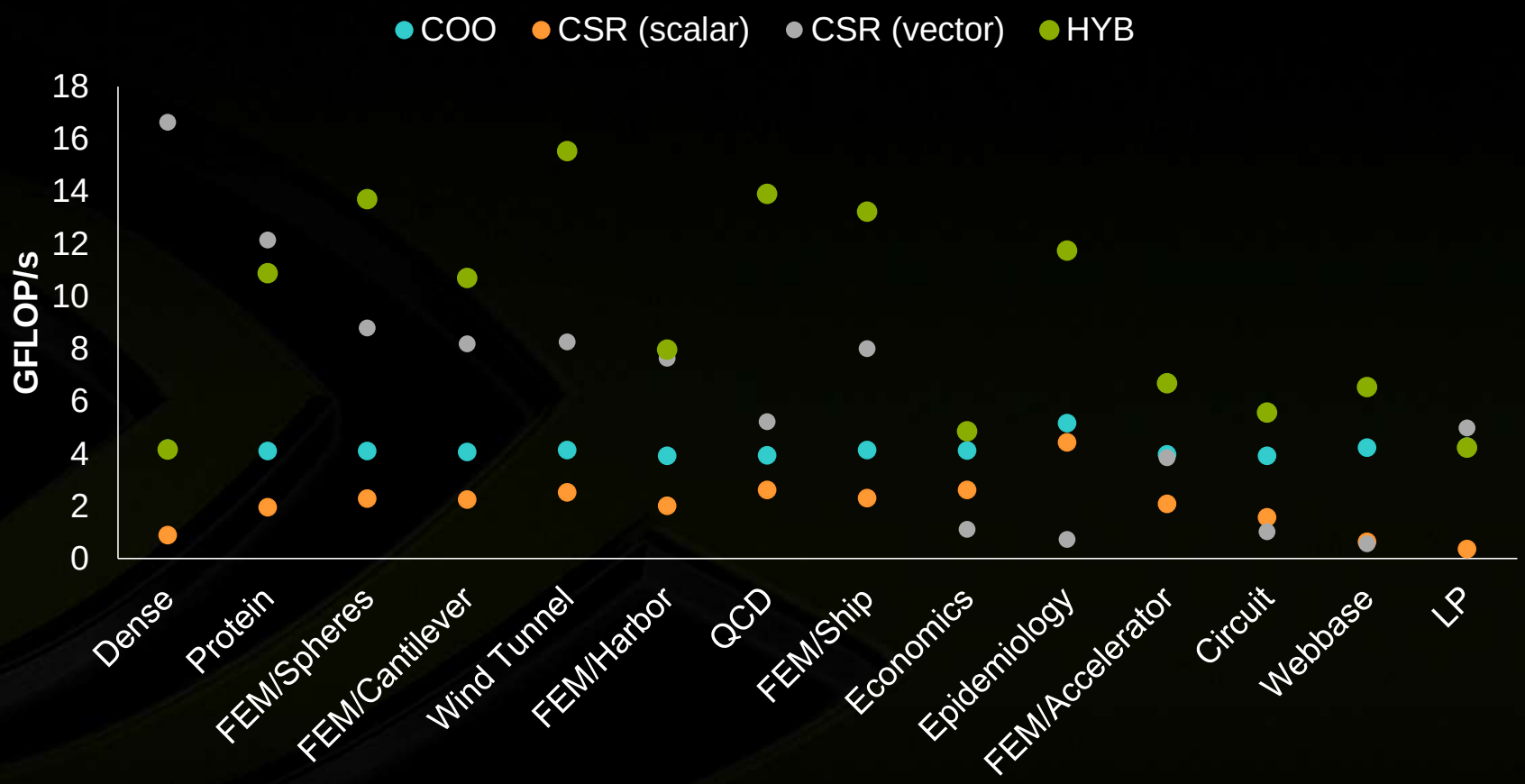
Hybrid Sparse Matrix Format for GPUs

- ELL handles *typical* entries
- COO handles *exceptional* entries
  - Implemented with segmented reduction



- Some overheads in matrix format conversion, can be hidden if the solver does O(100) of iterations

SpMV Performance for Unstructured Matrices



Flops=2*nz/t, BW = (2*sizeof(double)+size(int))/t

Soul of NVIDIA's GPU Roadmap

NVIDIA

Increase Performance / Watt

Make Parallel Programming Easier

Run more of the Application on the GPU

Tesla CUDA GPU Roadmap

NVIDIA Announced “Project Denver” Jan 2011

NVIDIA Announces "Project Denver" to Build Custom CPU Cores Based on ARM Architecture, Targeting Personal Computers to Supercomputers

NVIDIA Licenses ARM Architecture to Build Next-Generation Processors That Add a CPU to the GPU

LAS VEGAS, NV -- (Marketwire) -- 01/05/2011 -- **CES 2011** -- NVIDIA announced today that it plans to build high-performance ARM® based CPU cores, designed to support future products ranging from personal computers and servers to workstations and supercomputers.

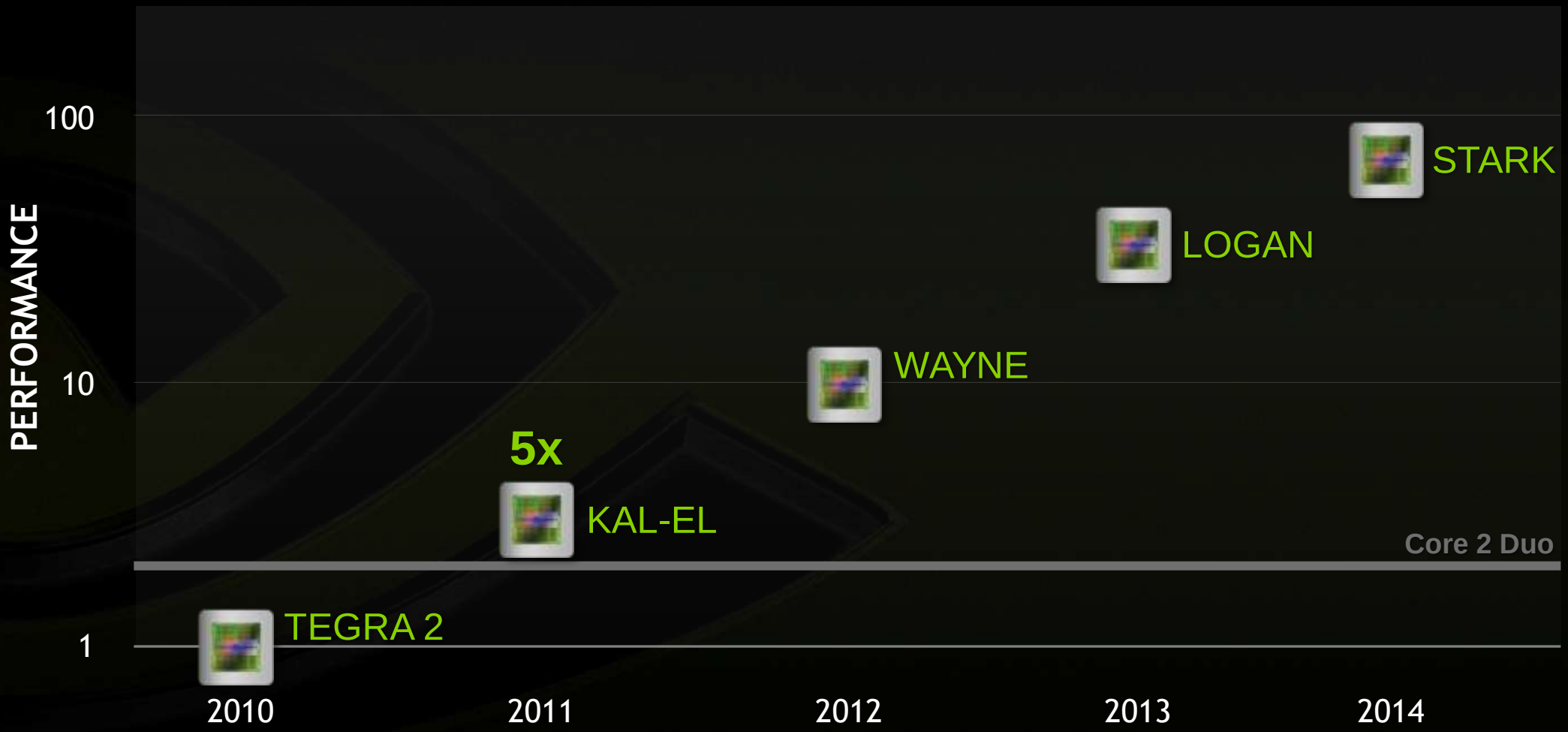
It's true folks, NVIDIA's building a CPU! Madness! The future just got a lot more exciting.

http://www.engadget.com/2011/01/05/nvidia-announces-project-denver-arm-cpu-for-the-desktop/

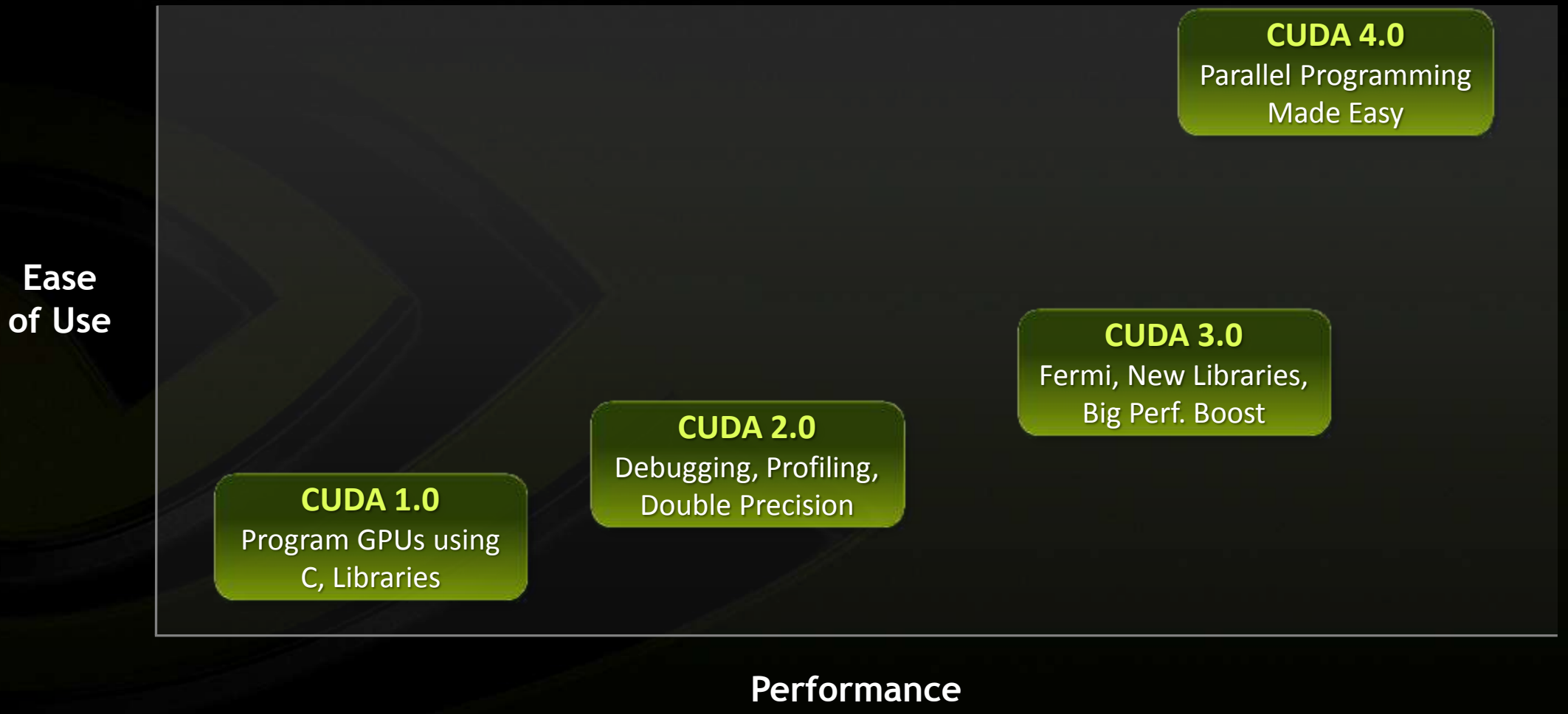
An ARM processor coupled with an NVIDIA GPU represents the computing platform of the future. A high-performance CPU with a standard instruction set will run the serial parts of applications and provide compatibility while a highly-parallel, highly-efficient GPU will run the parallel portions of programs.

The result is that future systems - from the thinnest laptops to the biggest data centers, and everything in between — will deliver an outstanding combination of performance and power efficiency.

Tegra Roadmap



CUDA 4.0: Big Leap In Usability



NVIDIA GPUDirect™: *Eliminating CPU Overhead*

Accelerated Communication
with Network and Storage Devices

- Direct access to CUDA memory for 3rd party devices
- Eliminates unnecessary memory copies & CPU overhead
- Supported by Mellanox and Qlogic

Supported since CUDA 3.1

Peer-to-Peer Communication
Between GPUs

- Peer-to-Peer memory access, transfers & synchronization
- Less code, higher programmer productivity

New in CUDA 4.0

http://developer.nvidia.com/gpudirect

MPI Integration of NVIDIA GPUDirect™

- **MPI libraries with support for NVIDIA GPUDirect and Unified Virtual Addressing (UVA) enables:**
  - **MPI transfer primitives copy data directly to/from GPU memory**
  - **MPI library can differentiate between device memory and host memory without any hints from the user**
  - **Programmer productivity: less application code for data transfers**

Code without MPI integration

At Sender:

```
cudaMemcpy(s_buf, s_device, size, cudaMemcpyDeviceToHost);
MPI_Send(s_buf, size, MPI_CHAR, 1, 1, MPI_COMM_WORLD);
```

At Receiver:

```
MPI_Recv(r_buf, size, MPI_CHAR, 0, 1, MPI_COMM_WORLD, &req);
cudaMemcpy(r_device, r_buf, size, cudaMemcpyHostToDevice);
```

Code with MPI integration

At Sender:

```
MPI_Send(s_device, size, …);
```

At Receiver:

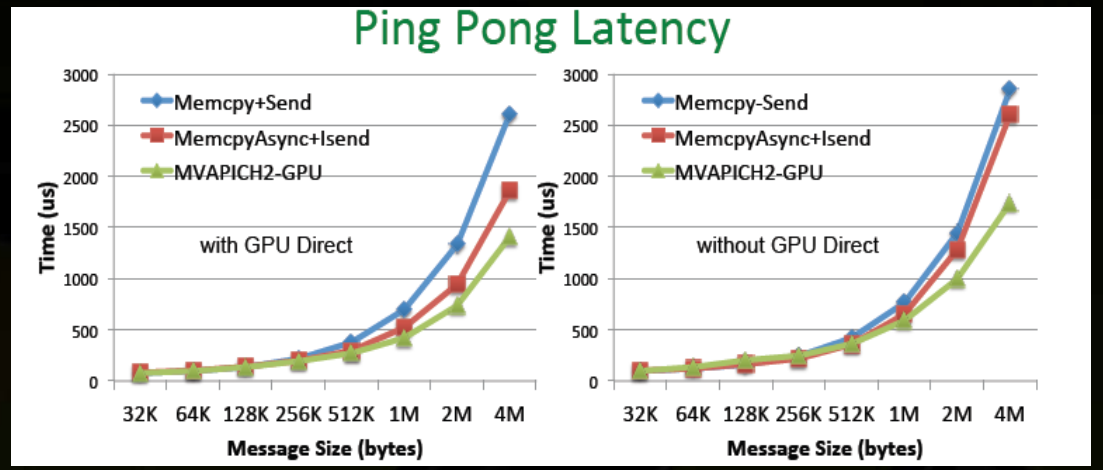
```
MPI_Recv(r_device, size, …);
```

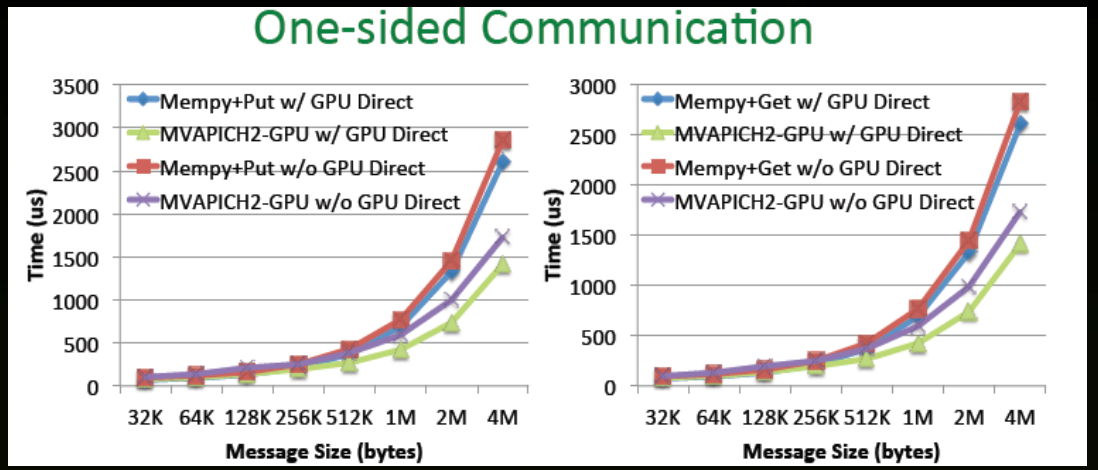
Open MPI

- Transfer data directly to/from CUDA device memory via MPI calls
- Code is currently available in the Open MPI trunk, available at:
  - http://www.open-mpi.org/nightly/trunk (contributed by NVIDIA)
- More details in the Open MPI FAQ
  - Features: http://www.open-mpi.org/faq/?category=running#mpi-cuda-support
  - Build Instructions: http://www.open-mpi.org/faq/?category=building#build-cuda

MVAPICH2-GPU

- Upcoming MVAPICH2 support for GPU-GPU communication with
  - Memory detection and overlap CUDA copy and RDMA transfer





With GPUDirect
- 45% improvement compared to Memcpy+Send (4MB)
- 24% improvement compared to MemcpyAsync+Isend (4MB)

Without GPUDirect
- 38% improvement compared to Memcpy+send (4MB)
- 33% improvement compared to MemcpyAsync+Isend (4MB)

With GPUDirect
- 45% improvement compared to Memcpy+Put

Without GPUDirect
- 39% improvement compared with Memcpy+Put

Similar improvement for Get operation

Major improvement in programming

Measurements from:
H. Wang, S. Potluri, M. Luo, A. Singh, S. Sur and D. K. Panda, *"MVAPICH2-GPU: Optimized GPU to GPU Communication for InfiniBand Clusters"*, Int'l Supercomputing Conference 2011 (ISC), Hamburg

http://mvapich.cse.ohio-state.edu/

GPUDirect: Further Information

- http://developer.nvidia.com/gpudirect
  - More details including supported configurations
  - Instructions
  - System design guidelines
- Also talk to NVIDIA Solution Architects

QUESTIONS