TSUBAME2.0: A Tiny and Greenest Petaflops Supercomputer

Satoshi Matsuoka
Global Scientific Information and Computing Center (GSIC)
Tokyo Institute of Technology (Tokyo Tech.)

Booth #1127

Booth Presentations SC10

Nov 2010

The TSUBAME 1.0 "Supercomputing Grid Cluster" Spring 2006

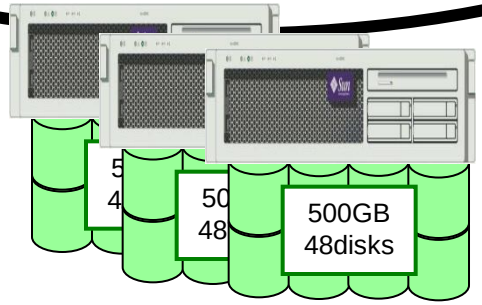
Unified IB network

Voltaire ISR9288 Infiniband 10Gbps
x2 (DDR next ver.)
~1310+50 Ports
~13.5Terabits/s (3Tbits bisection)

"Fastest Supercomputer in Asia" 7th on the 27th Top500@38.18TF

10Gbps+External Network

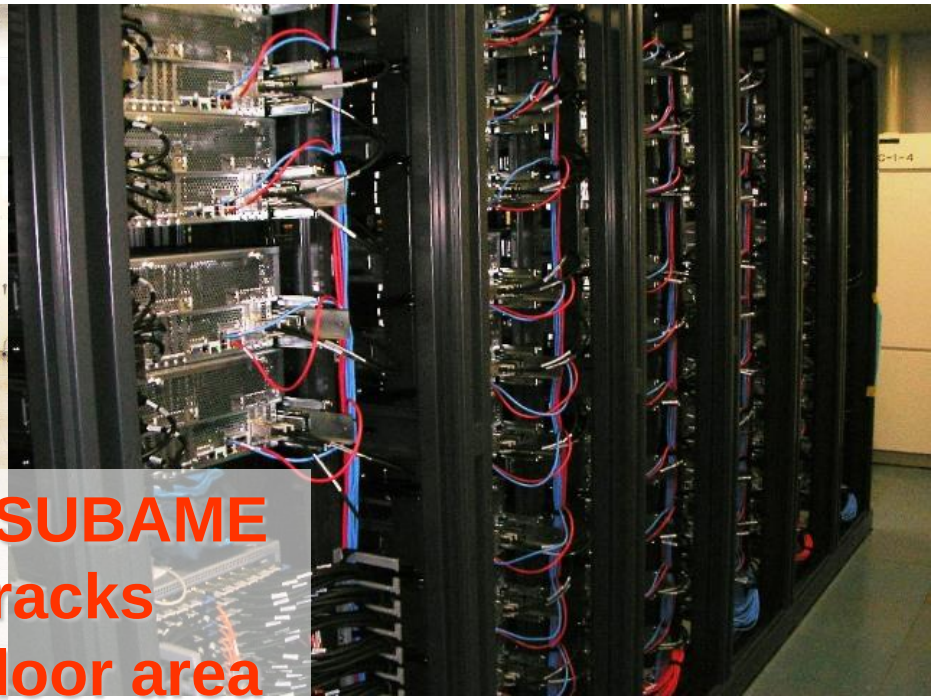
NEC SX-8i
(for porting)



Sun Galaxy 4 (Opteron Dual core 8-socket)
10480core/655Nodes
21.4Terabytes
50.4TeraFlops
OS Linux (SuSE 9, 10)
NAREGI Grid MW

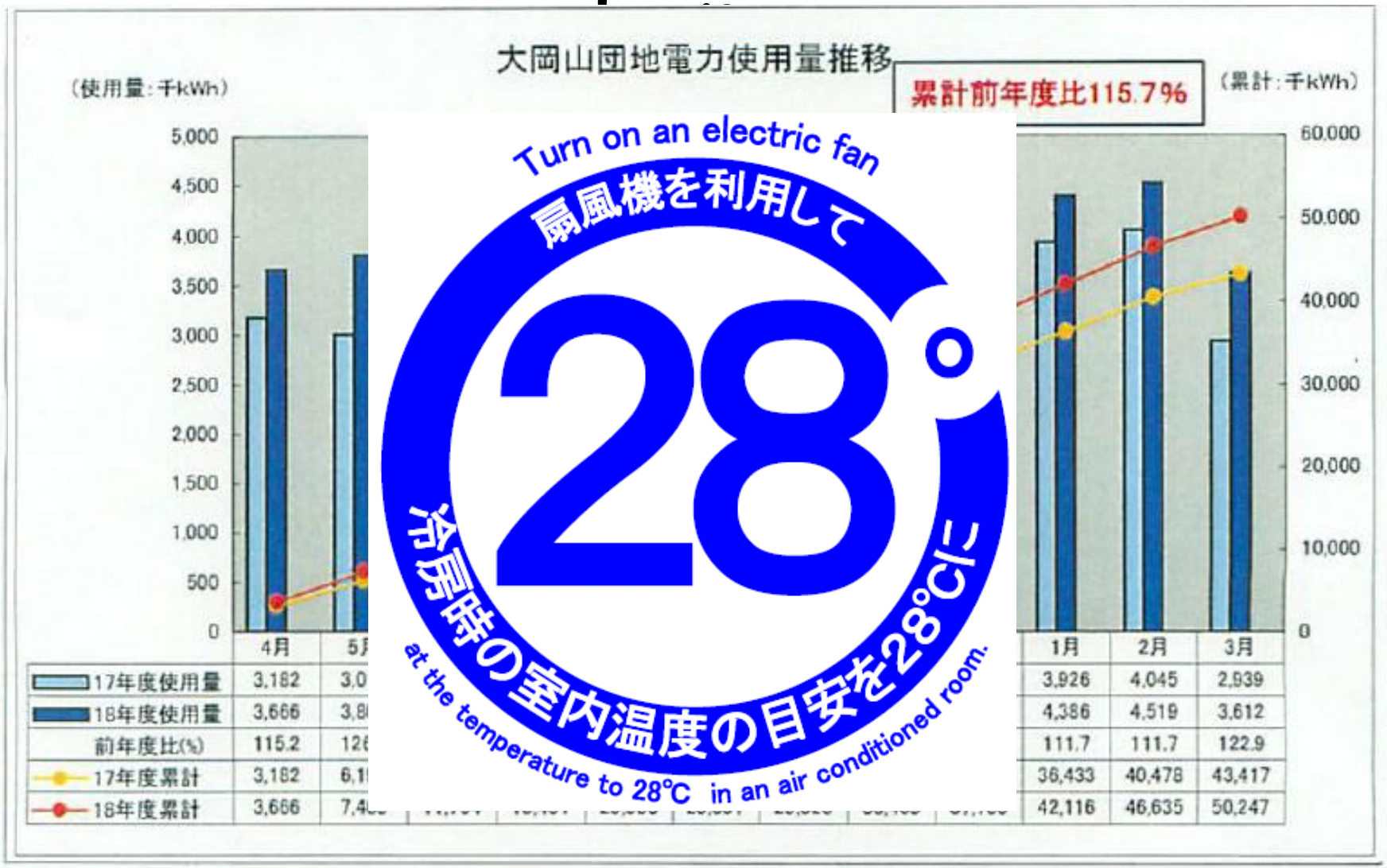
Storage
1.0 Petabyte (Sun "Thumper")
0.1Petabyte (NEC iStore)
Lustre FS, NFS, WebDAV (over IP)
50GB/s aggregate I/O BW

ClearSpeed CSX600
SIMD accelerator
360 boards,
35TeraFlops(Current))



Titech TSUBAME
~76 racks
350m2 floor area
1.2 MW (peak),
PUE=1.44

You know you have a problem



Biggest Problem is Power…

Machine	CPU Cores	Watts	Peak GFLOPS	Peak MFLOPS/ Watt	Watts/ CPU Core	Ratio c.f. TSUBAME
TSUBAME(Opteron)	10480	800,000	50,400	63.00	76.34	
TSUBAME2006 (w/360CSs)	11,200	810,000	79,430	98.06	72.32	
TSUBAME2007 (w/648CSs)	11,776	820,000	102,200	124.63	69.63	1.00
Earth Simulator	5120	6,000,000	40,000	6.67	1171.88	0.05
ASCI Purple (LLNL)	12240	6,000,000	77,824	12.97	490.20	0.10
AIST Supercluster (Opteron)	3188	522,240	14400	27.57	163.81	0.22
LLNL BG/L (rack)	2048	25,000	5734.4	229.38	12.21	1.84
Next Gen BG/P (rack)	4096	30,000	16384	546.13	7.32	4.38
TSUBAME 2.0 (2010Q3/4)	160,000	810,000	1,024,000	1264.20	5.06	10.14

TSUBAME 2.0 x24 improvement in 4.5 years…? ➔ ~ x1000 over 10 years

Scaling Peta to Exa Design?

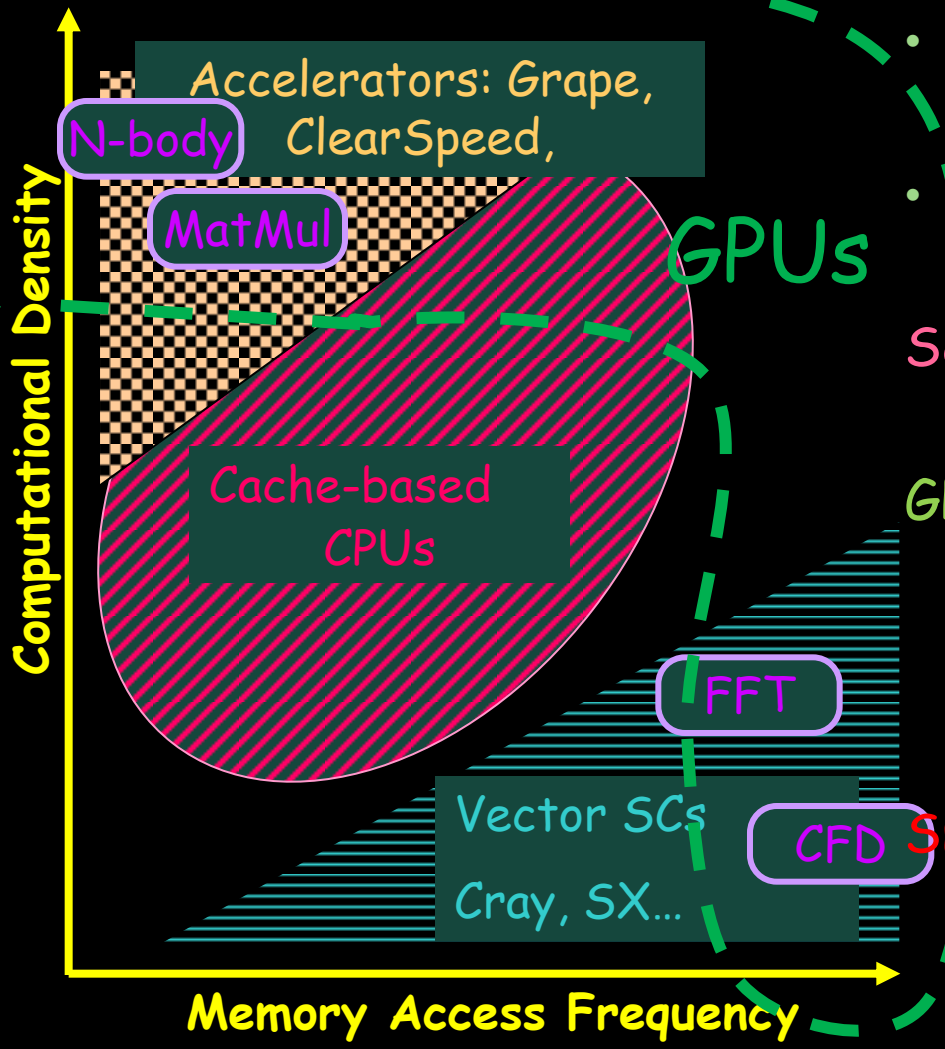
- ***Shorten*** latency as much as possible
  - Extreme multi-core incl. vectors
  - "Fat" nodes, exploit short-distance interconnection
  - Direct cross-node DMA (e.g., put/get for PGAS)
- ***Hide*** latency if cannot be shortened
  - Dynamic multithreading (Old: dataflow, New: GPUs)
  - Trade Bandwidth for Latency (so we do need BW…)
  - Departure from simple mesh system scaling
- ***Change*** Latency-Starved Algorithms
  - From implicit Methods to direct/hybrid methods
  - Structural locality, extraploation, stochastics (MC)
  - Need good programming model/lang/system for this…

GPUs as Commodity Massively Parallel Vector Processors

- E.g., NVIDIA Tesla, AMD Firestream
  - High Peak Performance > 1TFlops
    - Good for tightly coupled code e.g. Nbody
  - **High Memory bandwidth (>100GB/s)**
    - **Good for sparse codes e.g. CFD**
  - **Low latency over shared memory**
    - **Thousands threads hide latency w/zero overhead**
  - **Slow and Parallel and Efficient vector engines for HPC**
  - Restrictions: Limited non-stream memory access, PCI-express overhead, programming model etc.

How do we exploit them given vector computing experiences?

GPUs as Modern-Day Vector Engines



Two types of HPC Workloads

- High Computational Density
  => Traditional Accelerators
- High Memory Access Frequency
  => Traditional Vector SCs

Scalar CPUs are so-so at both
=> Massive system for speedup

GPUs are both ***modern-day vector engines and high compute density accelerators***
=> Efficient Element of Next-Gen Supercomputers

Small memory, limited CPU-GPU BW, high-overhead communication with CPUs and other GPUs, lack of system-level SW support incl. fault tolerance, programming, ...

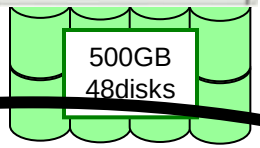
Our Research@ Tokyo Tech

TSUBAME 1.2 Experimental Evolution (Oct. 2008)

The first "Petascale" SC in Japan

Voltaire ISR9288 Infiniband x8
10Gbps x2 ~1310+50 Ports
~13.5Terabits/s
(3Tbits bisection)

NEC SX-8i



Storage
1.5 Petabyte (Sun x4500 x 60)
0.1Petabyte (NEC iStore)
Lustre FS, NFS, CIF, WebDAV (over IP)
60GB/s aggregate I/O BW

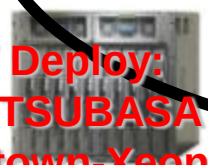
10Gbps+External NW

Unified Infiniband network

87.1TF Linpack (The First GPU Supercomputer on Top500) [IPDPS10]

Sun x4600 (16 Opteron Cores)
32~128 GBytes/Node
10480core/655Nodes
21.4TeraBytes
50.4TeraFlops
OS Linux (SuSE 9, 10)
NAREGI Grid MW

NEW Deploy:
GCOE TSUBASA
Harpertown-Xeon
90Node 720CPU
8.2TeraFlops



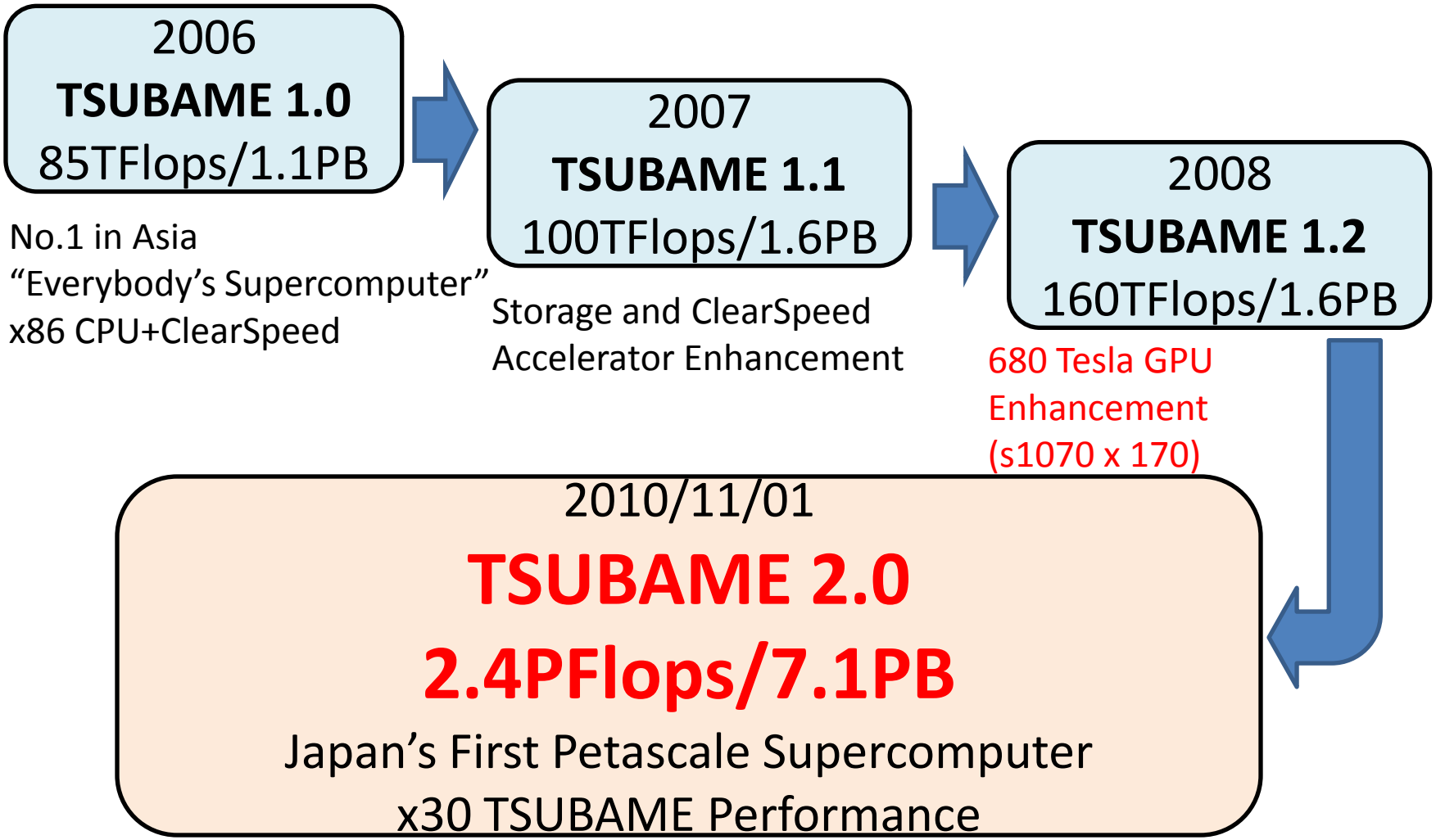
PCI-e

ClearSpeed CSX600
SIMD accelerator
648 boards,
52.2TeraFlops
SFP/DFP

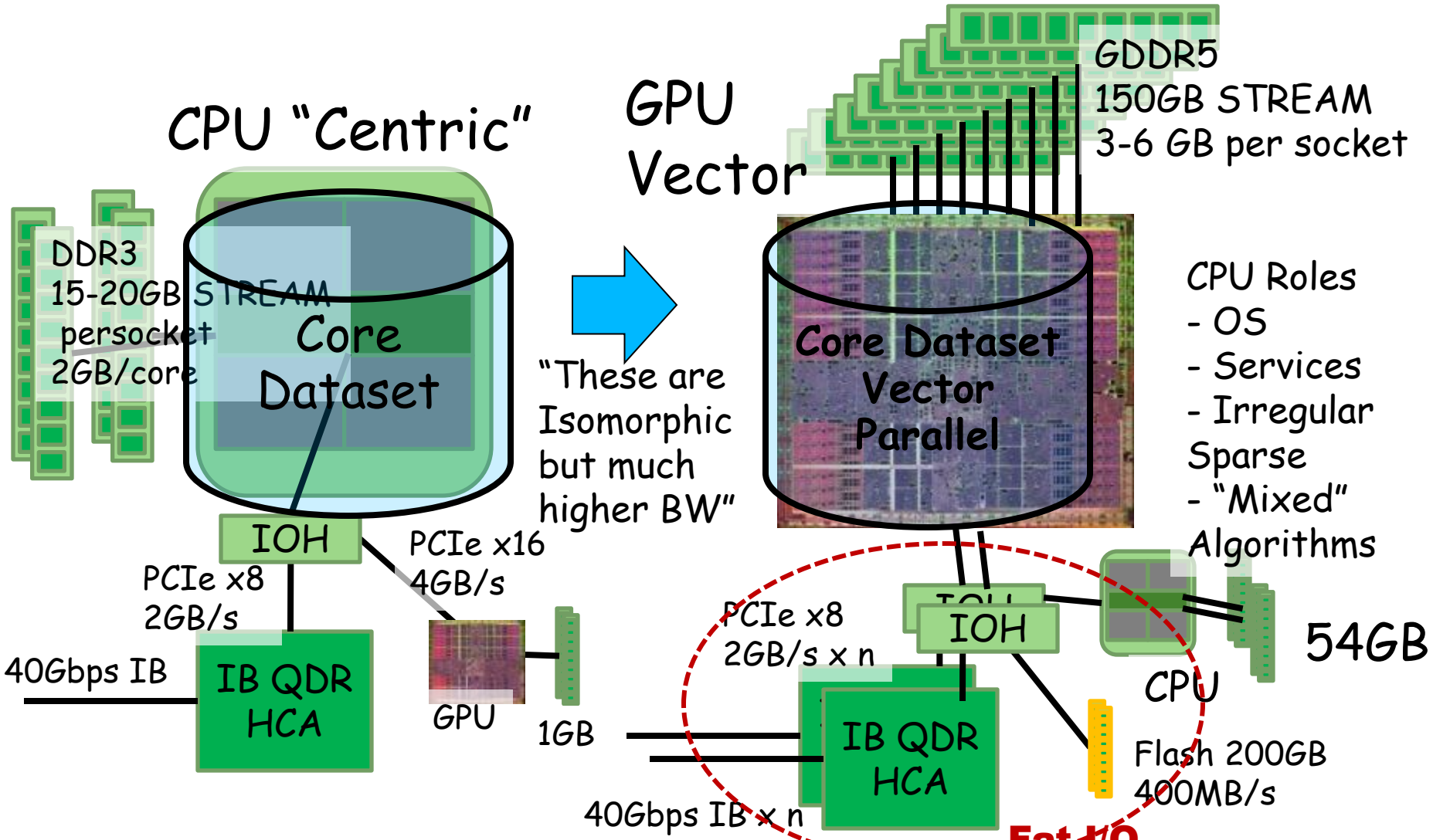
170 Nvidia Tesla 1070, ~680 Tesla cards
High Performance in Many BW-Intensive Apps
10% power increase over TSUBAME 1.0


680 Unit Tesla Installation…
While TSUBAME in Production Service (!)

History of TSUBAME



From TSUBAME 1.2 to 2.0: From CPU Centric to GPU Centric Nodes for Scaling High Bandwidth in Network & I/O!!

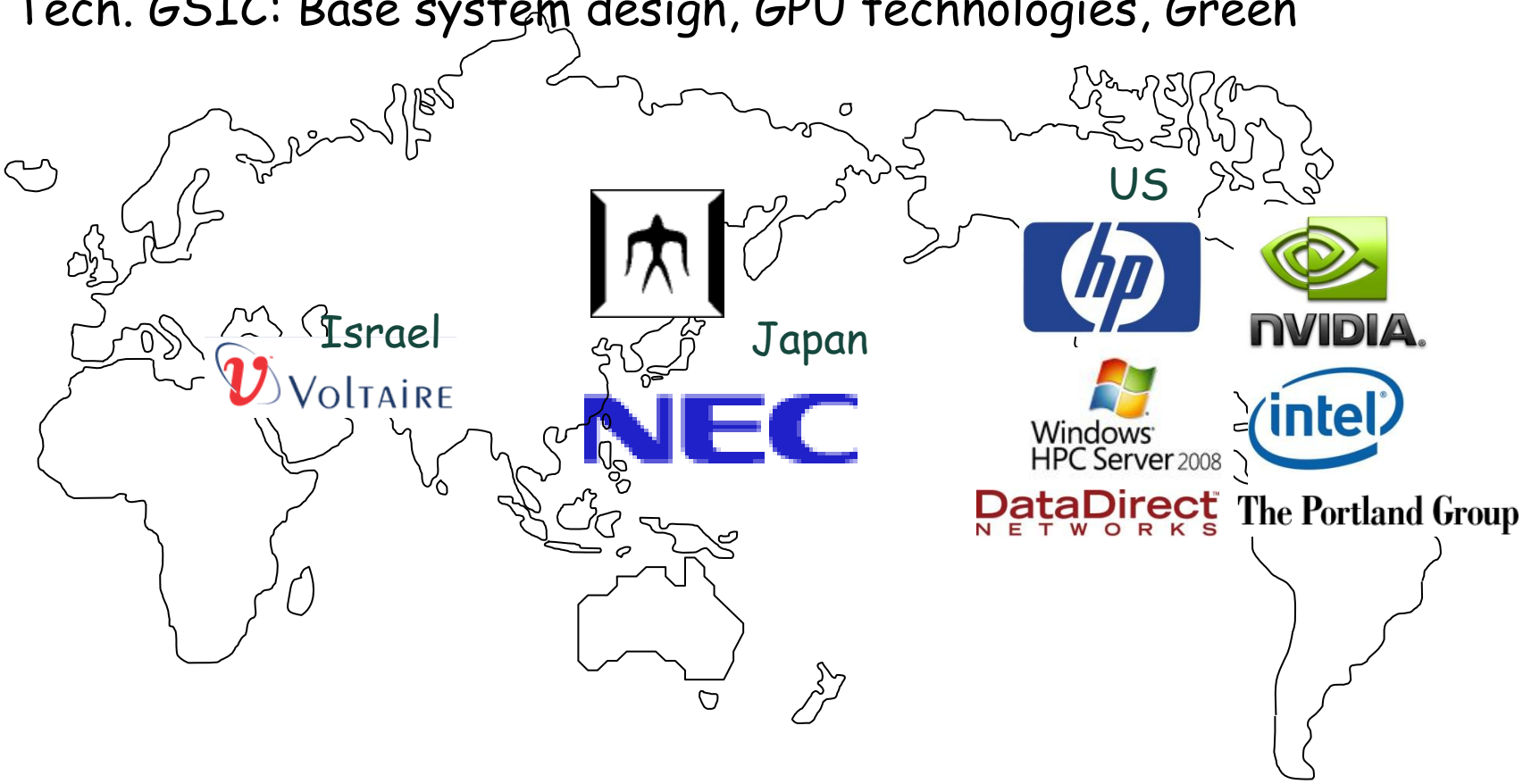


Tsubame 2: Start with Clean Slate!!!

- Achieve **30x** performance over Tsubame 1.0
- Support diverse research workloads
  - Weather, informatics, protein modeling, etc.
- Balanced design for ***scale***
  - Bandwidth to GPUs, CPU and GPU high memory capacity
  - High-bandwidth, low latency, full bi-section network
  - Solid state disks for high bandwidth, scalable storage I/O
- Green: stay within power threshold
  - System estimated to require 1.8 MW with PUE of 1.2
- Density (GPU per CPU footprint) for costs and space
- Expand support for Tokyo Tech's 2000 users; provide dynamic "cloud-like" provisioning for both Microsoft® Windows® and Linux workloads

TSUBAME2.0 Global Partnership

NEC: Main Contractor, overall system integration, Cloud-SC mgmt.
HP: Node R&D, Green; Microsoft: WindowsHPC, Cloud & VM
NVIDIA: Fermi GPUs, CUDA; Voltaire: QDR Infiniband Network
DDN: Large-Scale Storage; Intel: Westmere & Nehalem-EX CPUs
PGI: GPU Vectorizing Compiler
Tokyo Tech. GSIC: Base system design, GPU technologies, Green



Highlights of TSUBAME 2.0 Design (Oct. 2010) w/NEC-HP

- 2.4 PF multi-core x86 + Fermi GPGPU
  - 1432 nodes, Intel Westmere/Nehalem EX
  - 4224 NVIDIA Tesla (Fermi) M2050 GPUs
  - ~80,000 total CPU and GPU "cores", High Bandwidth
  - **1.9 million "CUDA cores", 32K x 4K = 130 million CUDA threads(!)**

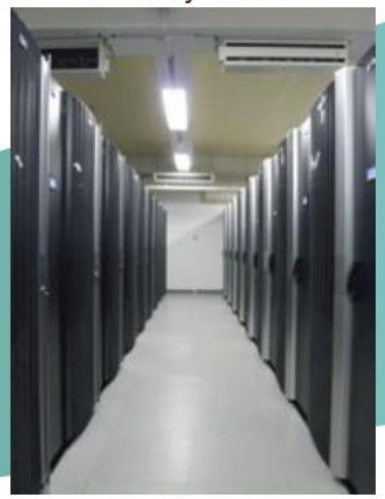
- 0.72 Petabyte/s aggregate mem BW,
  - Effective 0.3-0.5 Bytes/Flop, restrained memory capacity (100TB)
- Optical Dual-Rail IB-QDR BW, full bisection BW(Fat Tree)
  - 200Tbits/s, Likely fastest in the world, still scalable
- Flash/node, ~200TB (1PB in future), 660GB/s I/O BW
  - >7 PB IB attached HDDs, 15PB Total HFS incl. LTO tape
- Low power & efficient cooling, comparable to TSUBAME 1.0 (~1MW); PUE = 1.28 (60% better c.f. TSUBAME1)
- Virtualization and Dynamic Provisioning of Windows HPC + Linux, job migration, etc.

TSUBAME2.0 Nov. 1, 2011
See Live @ Tokyo Tech Booth #1127

TSUBAME2.0: A GPU-centric Green 2.4 Petaflops Supercomputer

Tsubame 2.0: "Tiny" footprint, very power efficient
- Floorspace less than 200m² (2,100 ft²)
- Top-class power efficient machine on the Green 500

System
(42 Racks)
1408 GPU Compute Nodes,
34 Nehalem "Fat Memory" Nodes



2.4 PFLOPS
80 TB

Rack
(8 Node Chassis)

53.6 TFLOPS
1.7 TB/3.2 TB

Node Chassis
(4 Compute Nodes)

6.7 TFLOPS
220 GB/412 GB

Compute Node
(2 CPUs,3 GPUs)

1.6 TFLOPS
55 GB/103 GB

Chip
(CPU ,GPU)

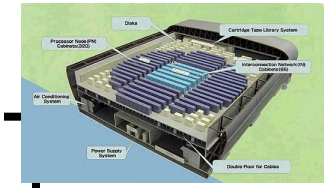
CPU(Westmere EP)
76.8 GFLOPS

GPUs(Tesla M2050)
515 GFLOPS
3 GB

Integrated by NEC Corporation

TSUBAME 2.0 Performance

Earth Simulator ⇒ TSUBAME 4years x 40 Downsizing



TSUBAME ⇒ TSUBAME 2.0 x40 downsizing again?

TSUBAME2.0 2.4 PF (2010-2H)

Japanese NLP >10PF(2012)

US >10P (2011~12)

US/EU Petascales (Peak) (2008)

US NSF/DoE (2009 2H)

U-Tokyo, Kyoto-U, Tsukuba T2K 61-140TF (2008)

Others

BlueGene/L 360TF(2005)

TSUBAME 1.2 Upgrade (3Q2008) 170TF

TSUBAME 1.1 Upgrade 111TF, 1.6 PB, 128GB nodes (1Q2008)

TSUBAME 1.0 80TF (1Q2006)

Earth Simulator 40TF (2002)

Titech (Tokyo Tech) Campus Grid

1.3TF

KEK 59TF BG/L+SR11100

10PF

1PF

100TF

10TF

1TF

2002 2004 2006 2008 2010 2012

TSUBAME2.0 World Rankings (Nov. 2010 PREDICTION for Green500!!!)

The Top 500 (Absolute Performance)

#1 : ~2.5 PetaFlops: China Defense Univ. Dawning Tianhe 1-A
#2 : 1.76 Petaflops: US ORNL Cray XT5 Jaguar
#3 : 1.27 PetaFlops: China Shenzen SC Nebulae
#4 : 1.19 PetaFlops: Japan Tokyo Tech. HP/NEC TSUBAME2.0
#5 : 1.054 PetaFlops: US LLBL Cray XE6 Hopper
#~33 (#2 Japan) : 0.191 Petaflops : JAEA Fujitsu

Top500 BoF
Tue 16th
17:30~

The Green 500 (Performance/Power Efficiency)

#1: ???
#2: ???
#3: ???
#4: ???
#5: ???

TSUBAME2.0
> 900MFlops/W

Green 500
BoF
Thu 18th
12:15~

"Little Green 500" collaboration w/Microsoft
Achieved 1.037 Gigaflops/W in power efficient experimental config.

Petaflops? Gigaflops/W?

x66,000 faster
x3 power efficient
<<
x44,000 Data

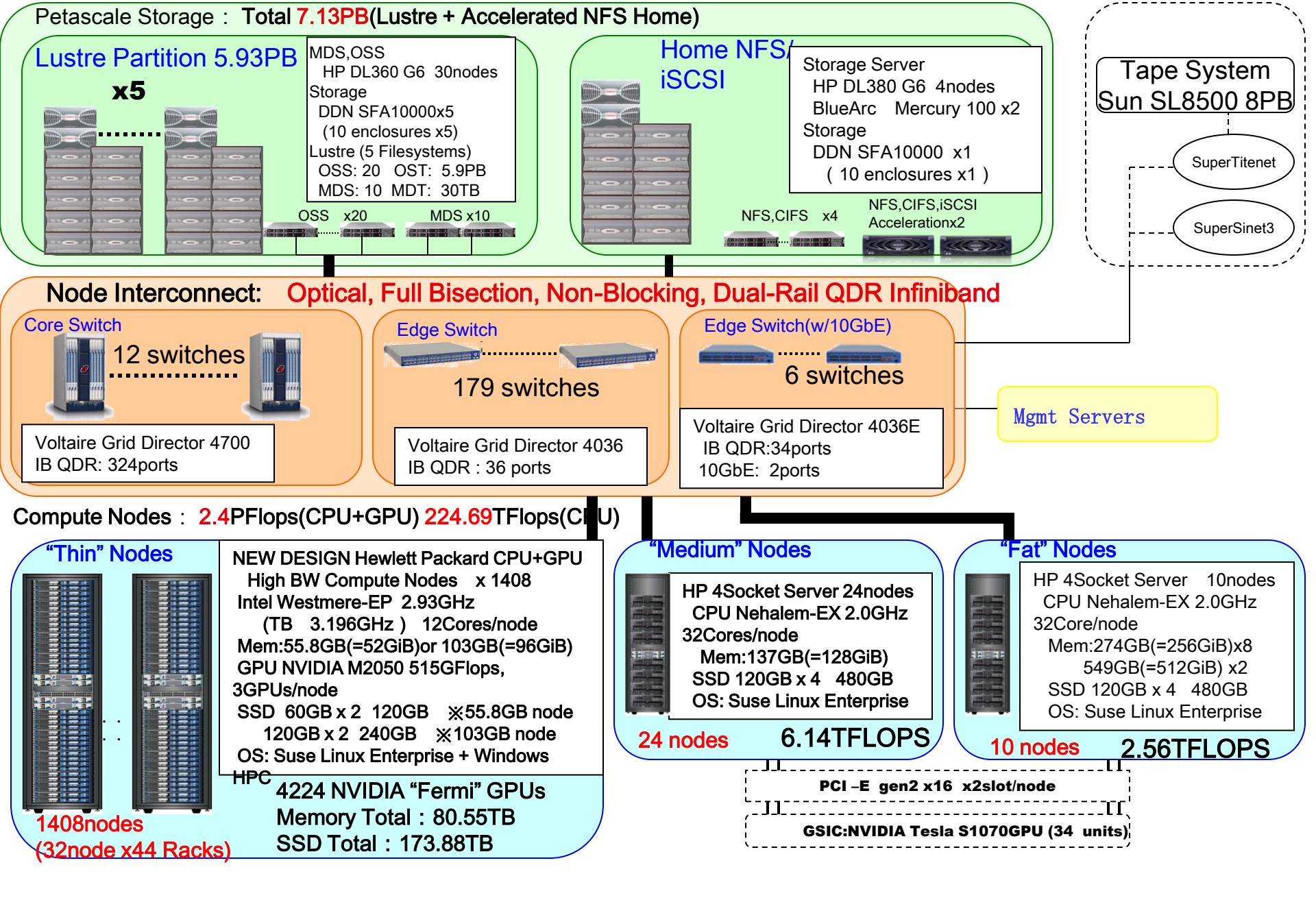
Laptop: SONY Vaio type Z (VPCZ1)
CPU: Intel Core i7 620M (2.66GHz)
MEMORY: DDR3-1066 4GBx2
OS: Microsoft Windows 7 Ultimate 64bit
HPL: Intel(R) Optimized LINPACK Benchmark for Windows (10.2.6.015)
256GB HDD

18.1 GigaFlops Linpack
369 MegaFlops/W

Supercomputer: TSUBAME 2.0
CPU: 2714 Intel Westmere 2.93 Ghz
GPU: 4071 nVidia Fermi M2050
MEMORY: DDR3-1333 80TB + GDDR5 12TB
OS: SuSE Linux 11 + Windows HPC Server R2
HPL: Tokyo Tech Heterogeneous HPL
11PB Hierarchical Storage

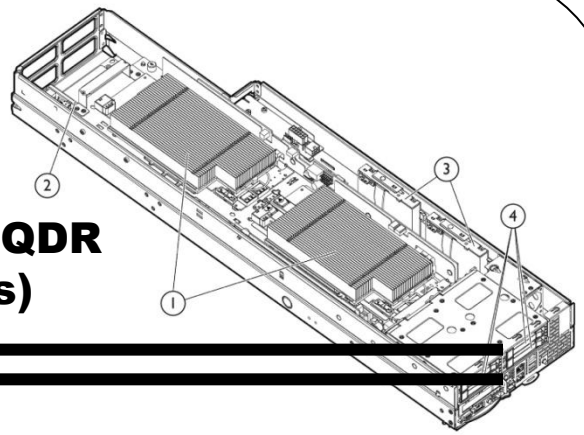
1.192 PetaFlops Linpack
1043 MegaFlops/W

TSUBAME2.0 System Overview (2.4 Pflops/15PB)



TSUBAME2.0 Compute Nodes

Thin Node



Infiniband QDR x2 (80Gbps)

HP SL390G7 (Developed for TSUBAME 2.0)
GPU: NVIDIA Fermi M2050 x 3
515GFlops, 3GByte memory /GPU
CPU: Intel Westmere-EP 2.93GHz x2 (12cores/node)
Memory: 54, 96 GB DDR3-1333
SSD：60GBx2, 120GBx2

1408nodes：

4224GPUs：59,136 SIMD Vector Cores, 2175.36TFlops (Double FP)

2816CPUs, 16,896 Scalar Cores: 215.99TFlops

Total： 2391.35TFLOPS

Memory： 80.6TB (CPU) + 12.7TB (GPU)

SSD： 173.9TB

IB QDR

PCI-e Gen2x16 x2
NVIDIA Tesla S1070 GPU

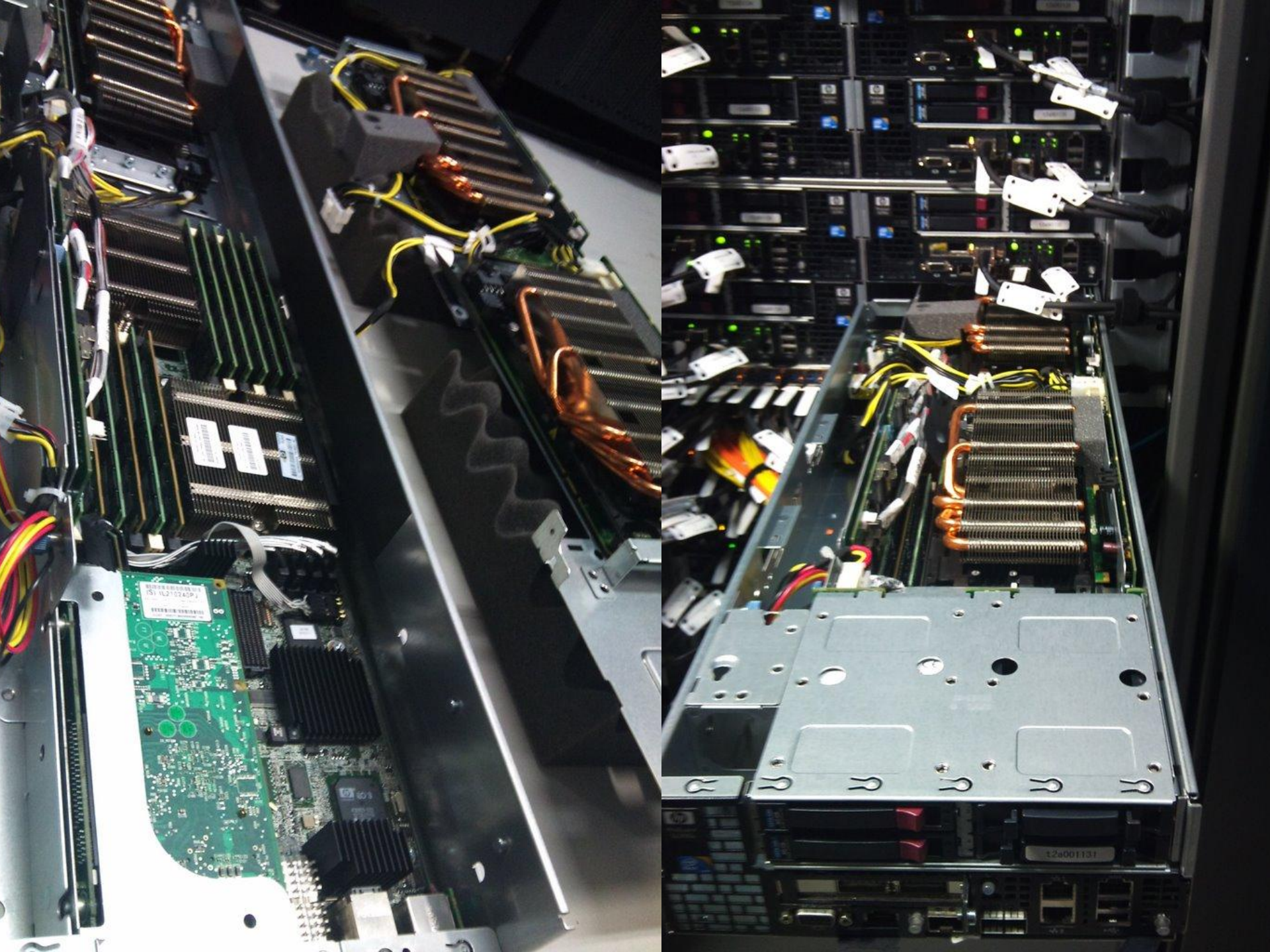
HP 4 Socket Server
CPU: Intel Nehalem-EX 2.0GHz x4 (32cores/node)
Memory: 128, 256, 512GB DDR3-1066
SSD：120GB x4 (480GB/node)

34 nodes: 8.7TFlops

Memory: 6.0TB+GPU

SSD: 16TB+

Total Perf
2.4PFlops
Mem： ~100TB
SSD: ~200TB



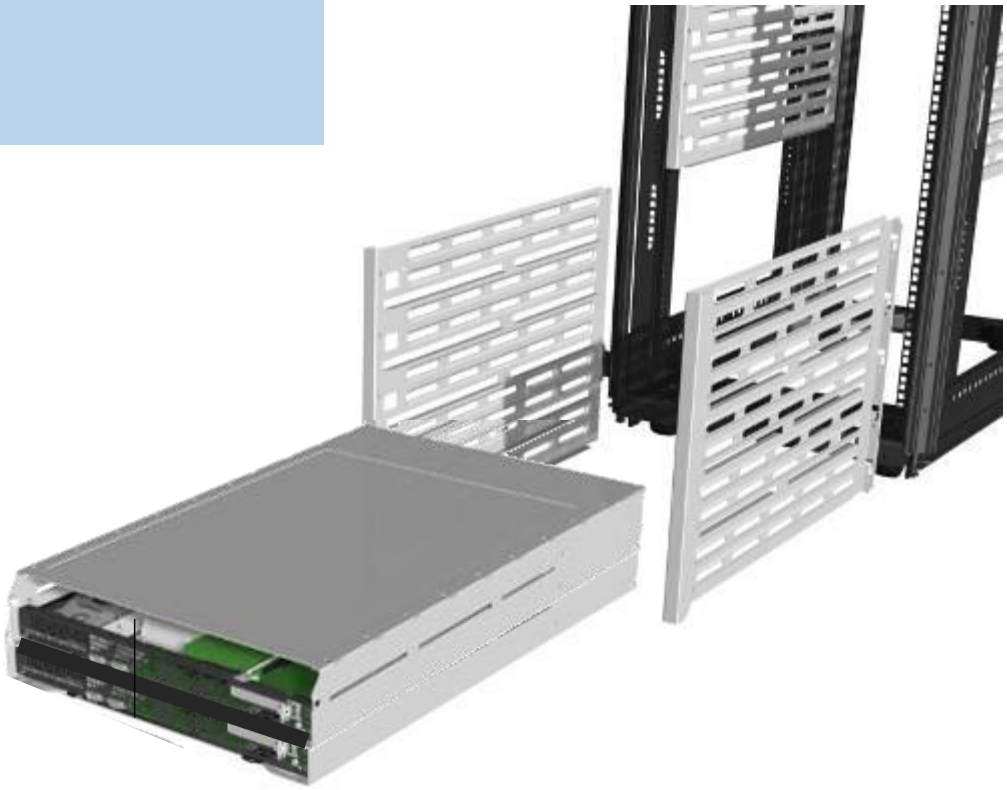

MADE IN TOKYO

GPU-optimized node in HP Skinless Packaging

Typical Tsubame2 node:

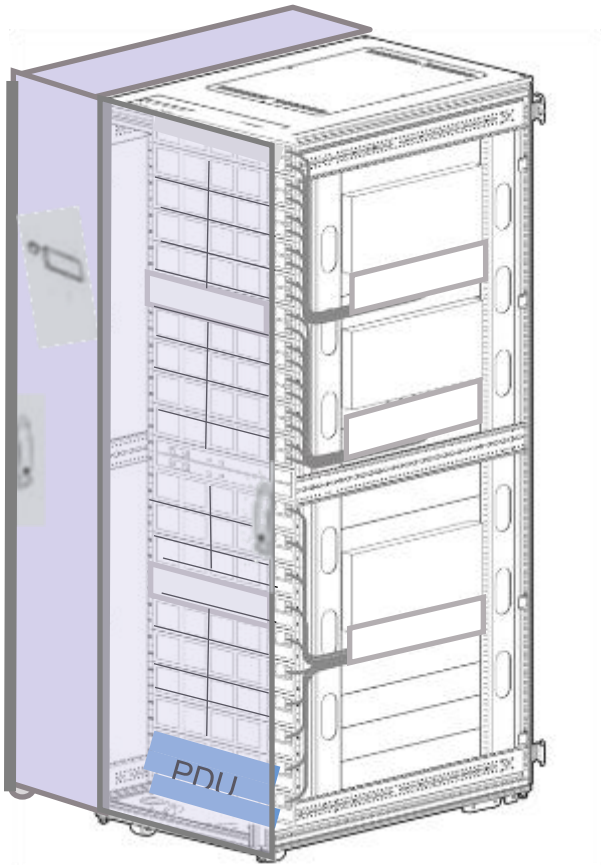
- 2 Intel® Xeon® Processor X5670 (six-core 2.93Ghz)
- 54 GB or 96 GB memory
- 2 SSDs per node
- Dual Rail IB
- 3 NVIDIA M2050 GPU board

- Nodes go into chassis with shared fans and power
- Chassis mounted into lightweight racks
- Easy assembly
- Lower weight
- Reduced heat retention
- Modular and flexible
- Standard 19” racks



Compute Rack Building Block for Tsubame2

HP Modular Cooling System G2 rack



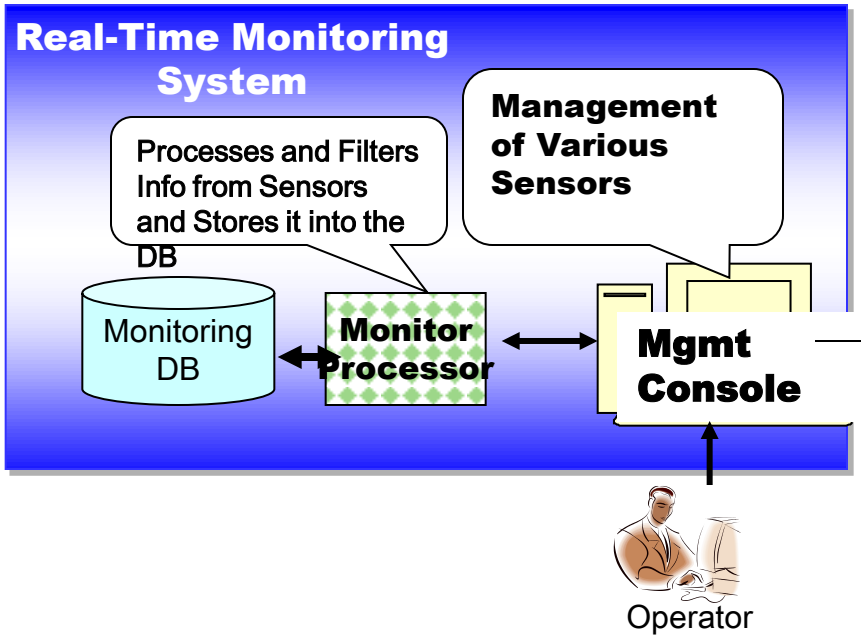
- 42U HP Modular Cooling System G2 rack
- 30 nodes per rack – 90 GPUs
- 8 chassis with Advanced Power Management
- 1 HP Network Switch for shared console and admin local area network
- 2 Airflow Dam (Liquid Cooled)
- 4 Voltaire 4036 Leaf Switch
- Power distribution units

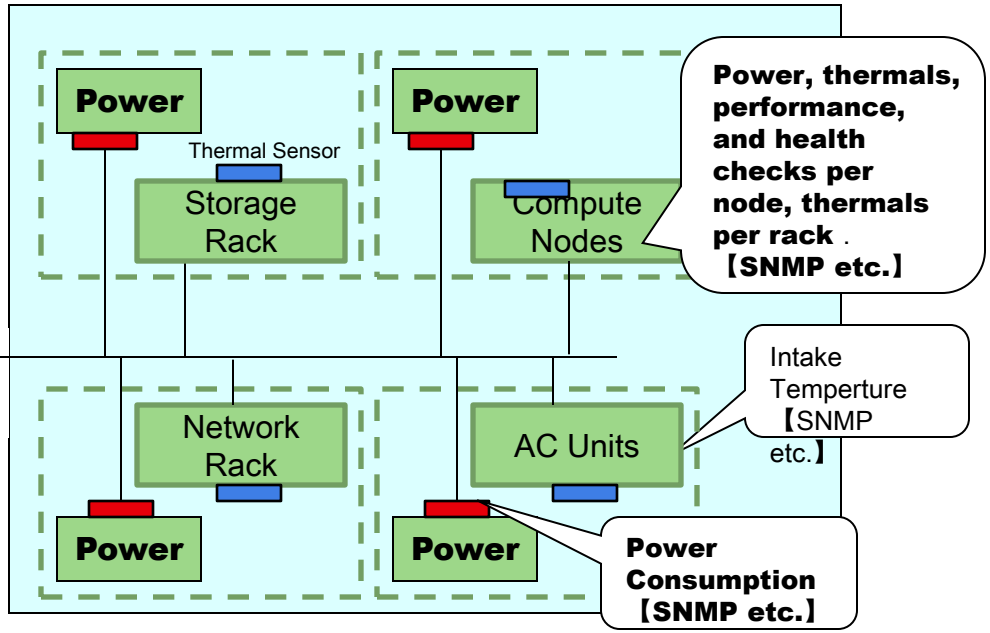
Power per rack approximately **35KW**

Green and Reliable SC

- Monitoring Sensors (incl. GPUs)
  - Thermals
  - Power Consumption
  - Utilization
  - HW and SW Health checks

- **Advanced Power Management**
  - **Dynamic Power Capping**
  - **Power monitoring**
  - **Node level power off/on**
- **Thermal logic technologies**
  - **Shared power and fans**
  - **Energy-efficient fans**
  - **Three-phase load balancing**
- **94% Platinum Common Slot Power Supplies**





Cooling: Enhanced HP Modular Cooling System G2

HP's water-cooled rack

Completely closed racks with their own heat exchanger.

1.5 x width of normal rack+rear ext.

Cooling for high density deployments

35kW of cooling capacity single rack

- **Highest Rack Heat Density ever**
- **3000CFM Intake airflow with 7C chiller water**

up to 2000 lbs of IT equipment

Uniform air flow across the front of the servers

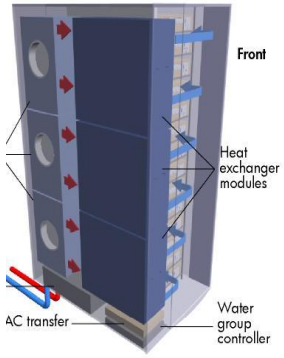
Automatic door opening mechanism controlling both racks

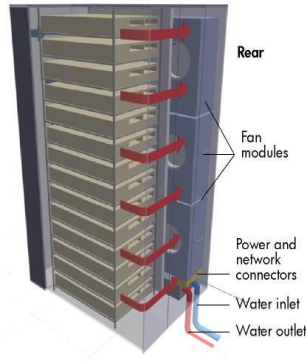
Adjustable temperature set point

Removes 95% to 97% of heat inside racks

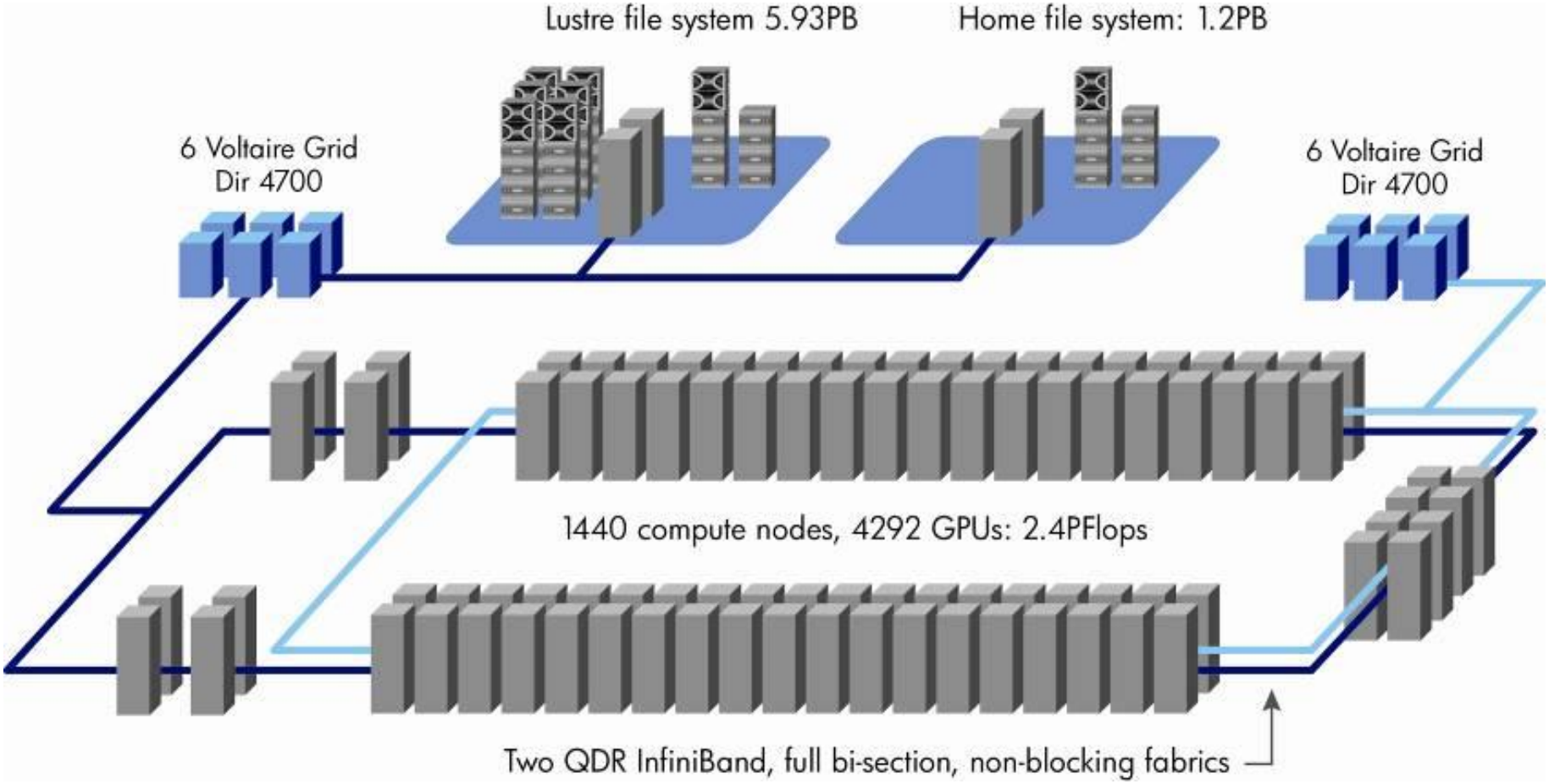
Polycarbonate front door reduces

~= Entire Earth Simulator (rack = 50TF)

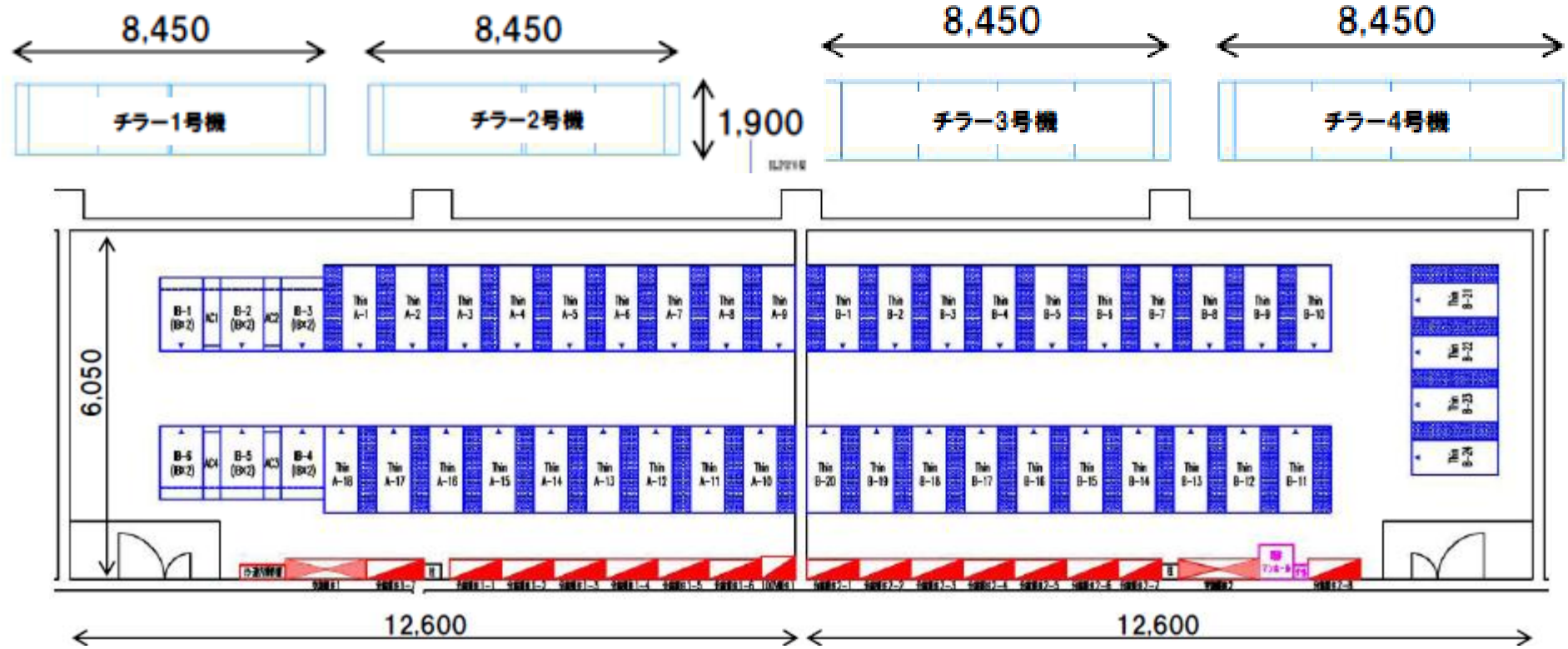




Pulling It All Together



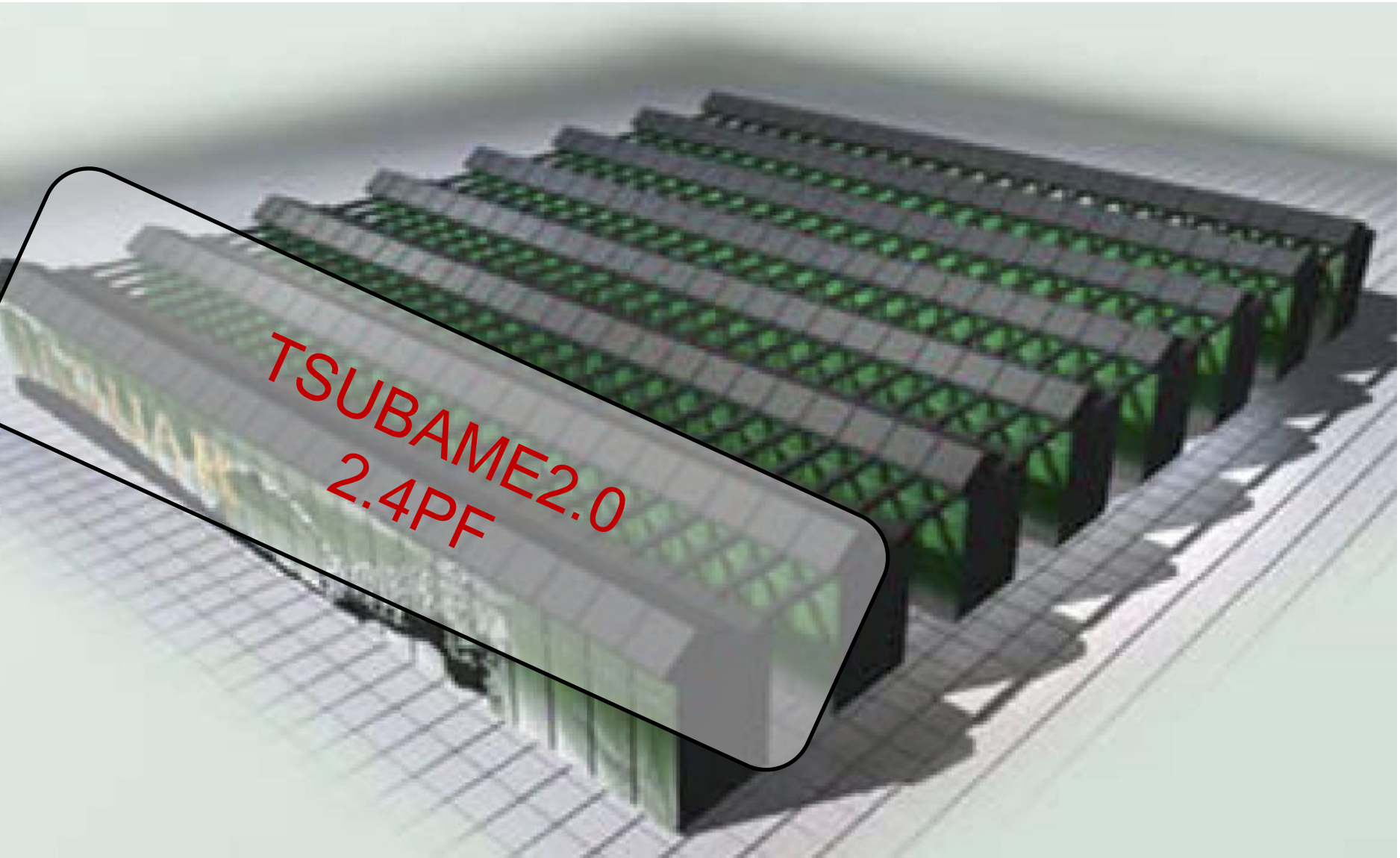
TSUBAME2.0 Layout (200m2 for main compute nodes)



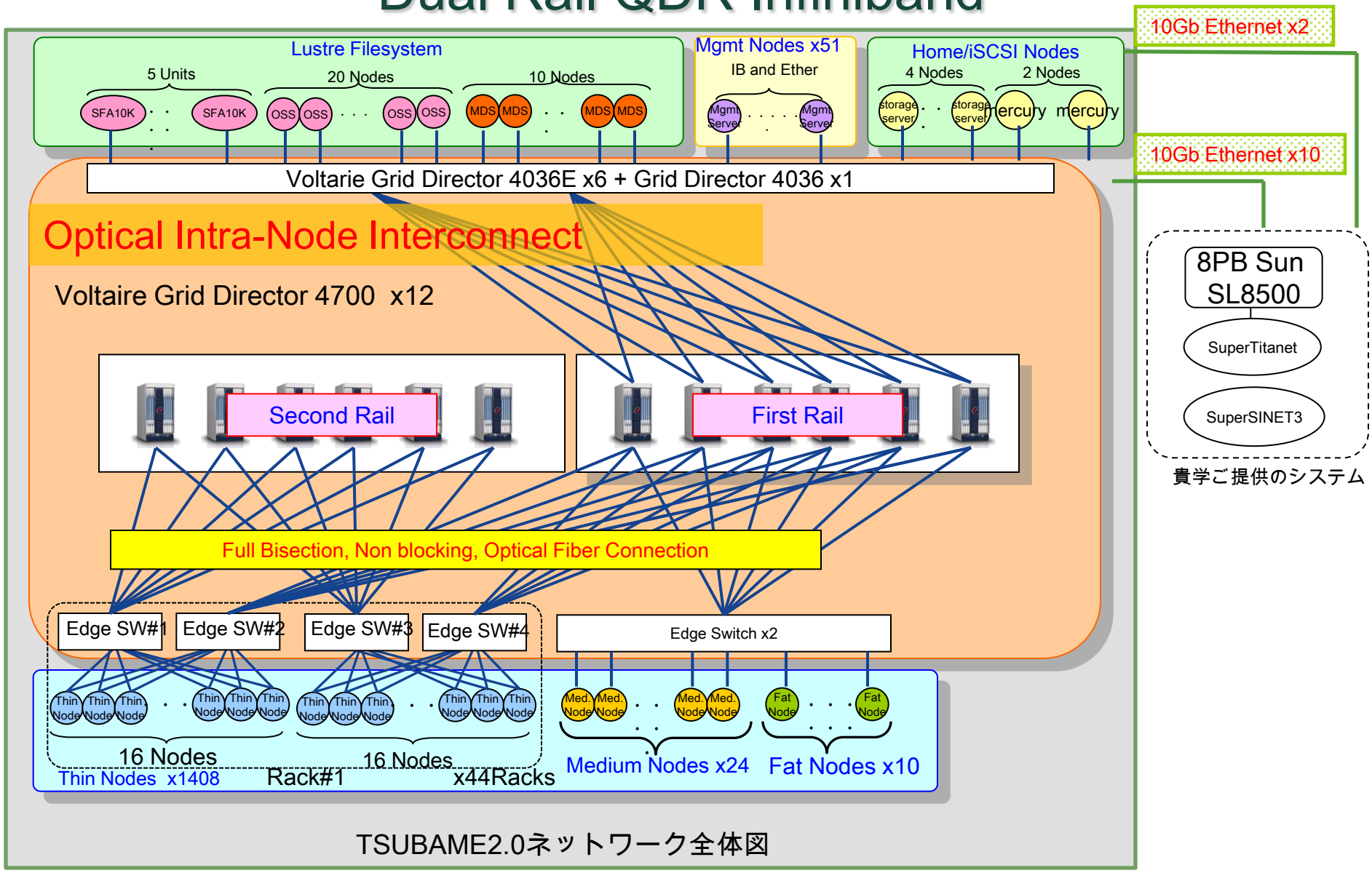

Push Here
Do Not
Push


2.4 Petaflops, 1408 nodes
~50 compute racks + 6 switch racks
Two Rooms, Total 160m²
1.4MW (Max, Linpack), 0.48MW (Idle)

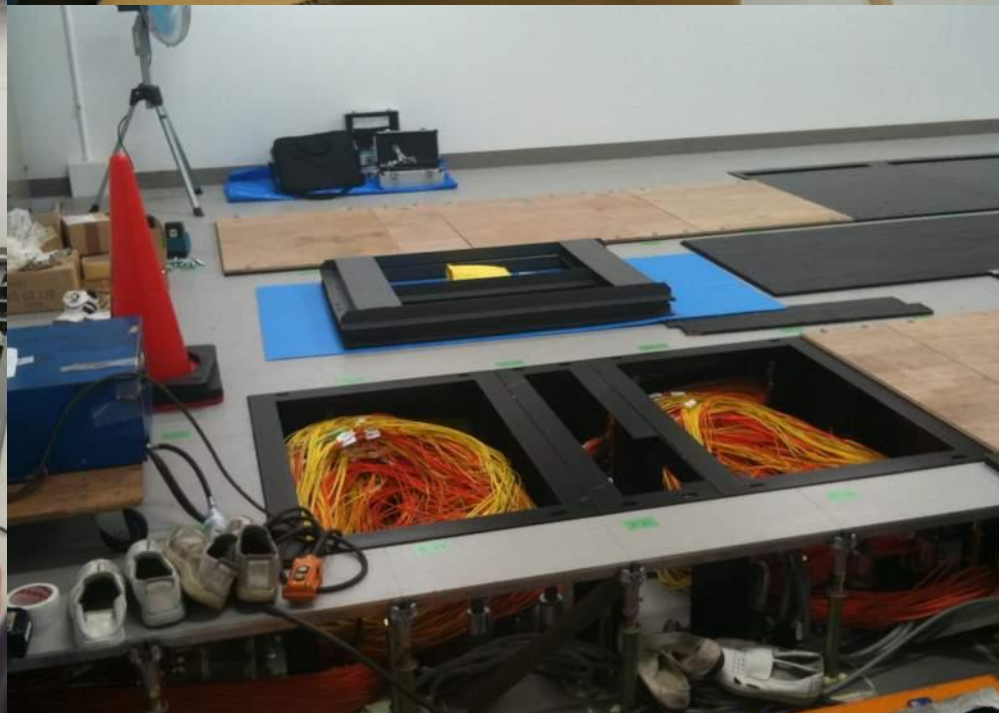
ORNL Jaguar and Tsubame 2.0
Similar Peak Performance, 1/4 the Size and Power



TSUBAME 2.0 Full Bisection Fat Tree, Optical, Dual Rail QDR Infiniband



TSUBAME2.0ネットワーク全体図



TSUBAME2.0 Storage

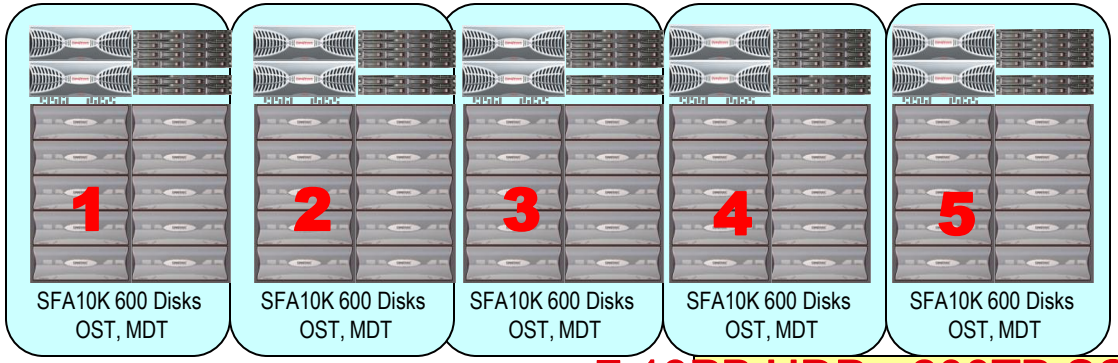
Multi-Petabyte storage consisting of Luster Parallel Filesystem Partition and NFS/CIFS/iSCSI Home Partition + Node SSD Acceleration

Lustre Parallel Filesystem Partition, 5.93PB

MDS：HP DL360 G6 x10
- -CPU:Intel Westmere-EP x2 socket(12 Cores)
- -Memory：51GB(=48GiB)
- -IB HCA:IB 4X QDR PCI-e G2 x1port

OSS：HP DL360 G6 x20
- -CPU:Intel Westmere-EP x2 socket(12 Cores)
- -Memory：25GB(=24GiB)
- -IB HCA：IB 4X QDR PCI-e G2 x2port

Storage：DDN SFA10000 x5
- -Total Capacity：5.93PB
  2TB SATA x 2950 Disks + 600GB SAS x 50 Disks

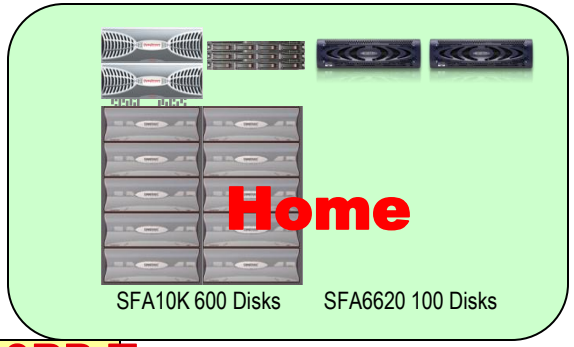


Home Partition 1.2PB

NFS/CIFS：HP DL380 G6　x4
- -CPU:Intel Westmere-EP x2 socket(12 Cores)
- -Memory：51GB(=48GiB)
- -IB HCA:IB 4X QDR PCI-e G2 x2port

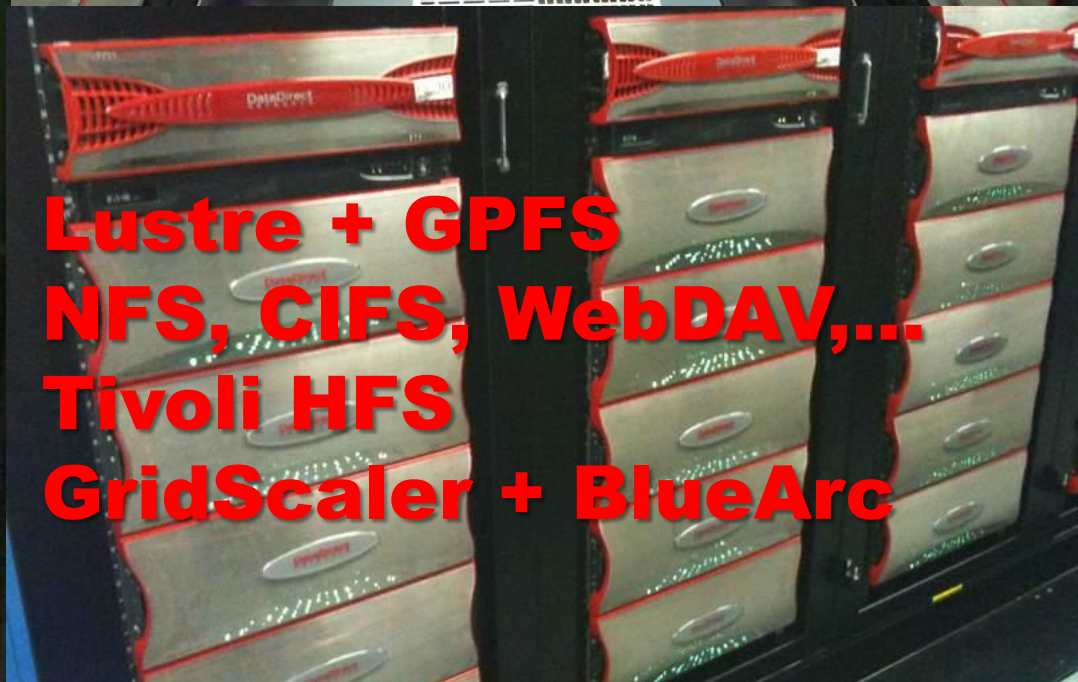
NFS/CIFS/iSCSI：BlueArc Mercury100 x2
- -10GbE x2

ストレージ：DDN SFA10000 x1
- -Total Capacity：1.2PB
  2TB SATA x 600 Disks



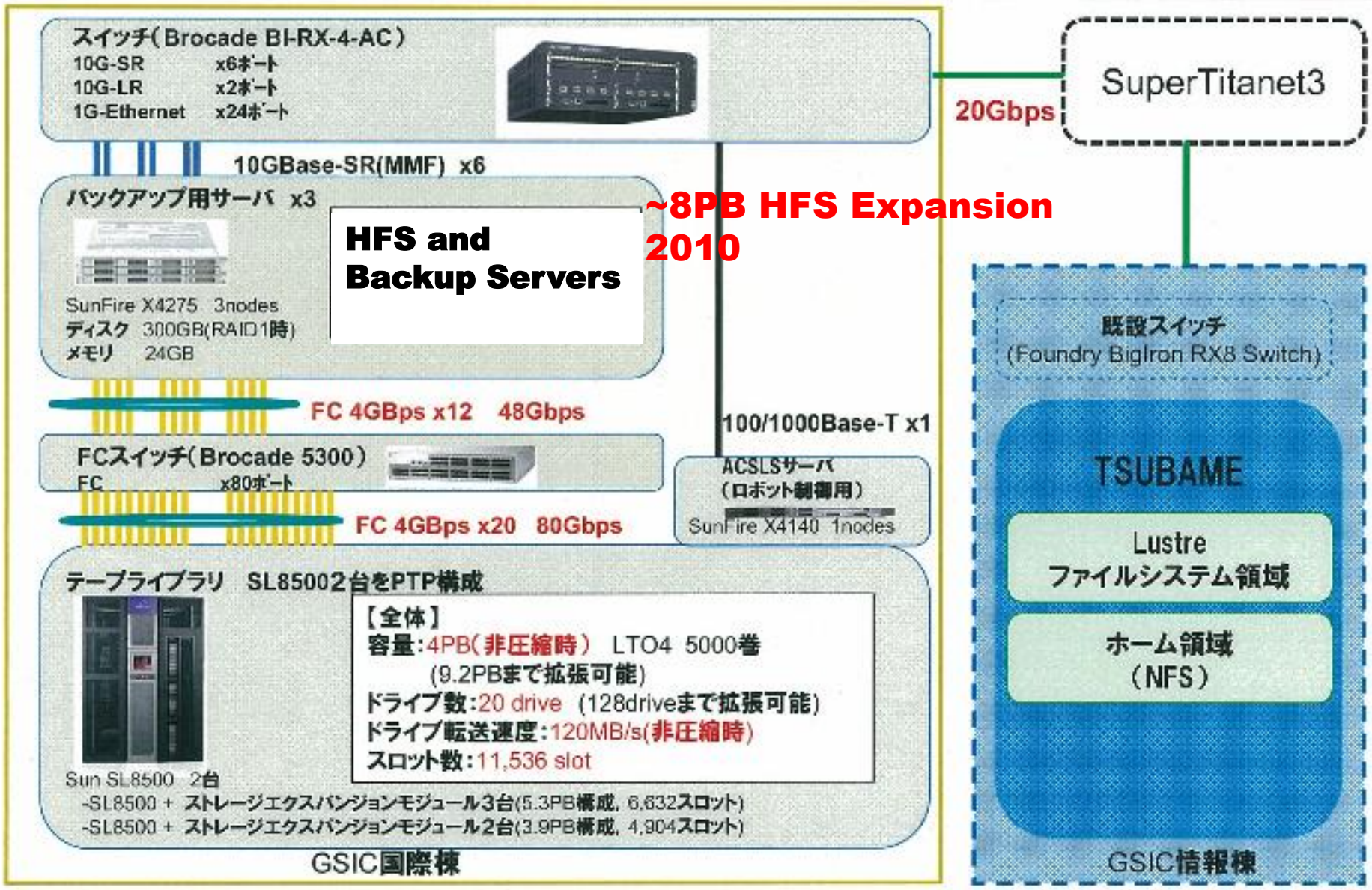
7.13PB HDD +200TB SSD + 8PB Tape

200 TB SSD (SLC, 1PB MLC future)
7.1 PB HDD (Highly redundunt)
4-8 PB Tape (HFS+Backup)
All Inifiniband+10GbE Connected



Lustre + GPFS
NFS, CIFS, WebDAV,...
Tivoli HFS
GridScaler + BlueArc

TSUBAME2.0 Tape System
HFS of over15PB and Scalable



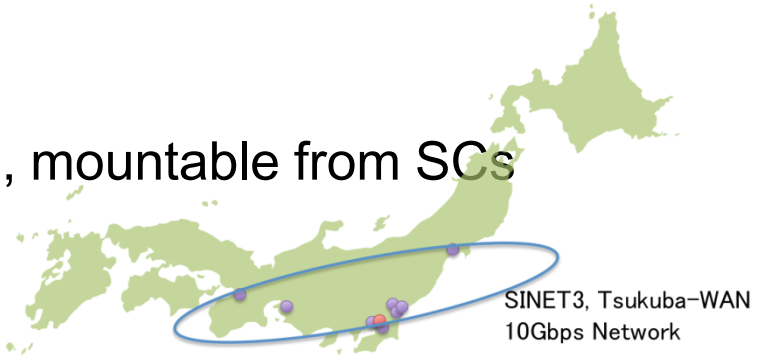
RENKEI-POP Distributed Storage

- Objective: SC Centers-Wide Distributed Storage on SINET National NW
  - RENKEI (MEXT e-Science)Project and R&D and nationwide deployment of RENKEI-PoPs(Point of Presence)
  - Data transfer server engine with large capacity and fast I/O
  - “RENKEI-Cloud” on SINET3 with various Grid/Cloud Services including Gfarm distributed filesystem
  - (

CPU	Core i7 975 Extreme (3.33 GHz)
Memory	12GB (DDR3 PC3-10600 , 2GB*6)
NIC	10GbE (without TCP/IP Offload Engine)
System Disk	500GB HDD
SSD RAID	30TB (RAID 5, 2TB HDD x 16)

Tokyo Inst. Tech.	Osaka Univ.
National Inst. Inform.	KEK
Nagoya Univ.	Univ. Tsukuba
AIST	Tohoku Univ.

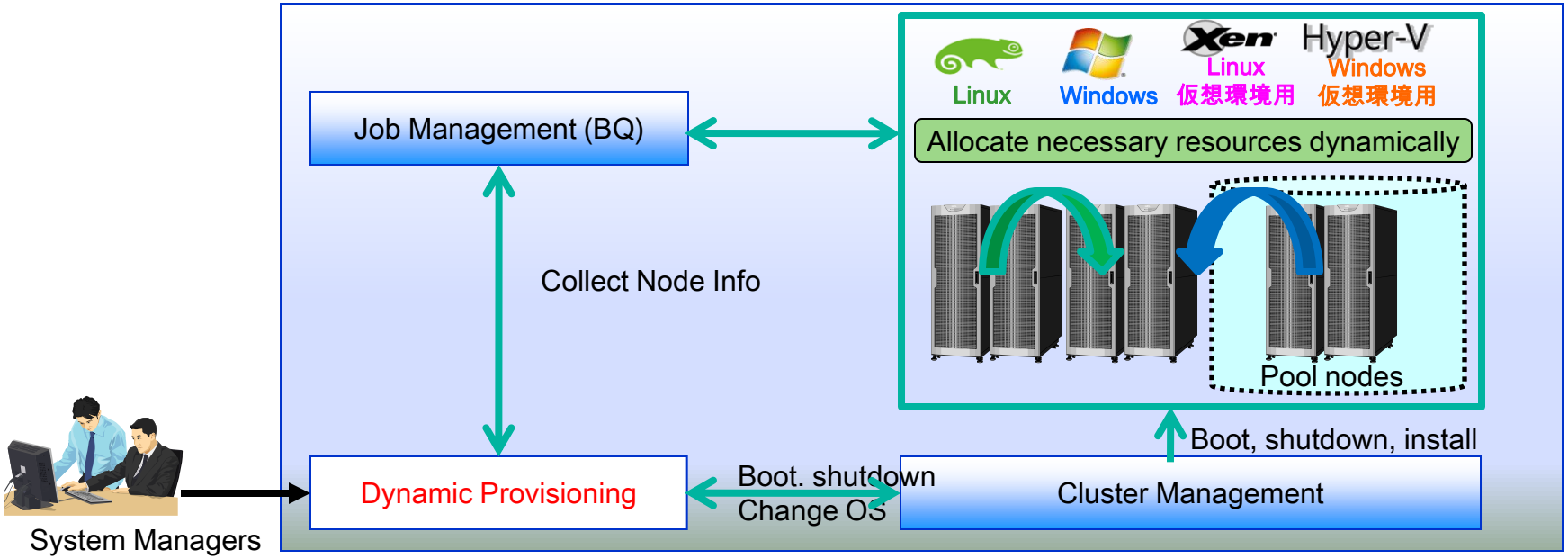
e, mountable from SCs



Cloud Provisioning Supercomputing Nodes

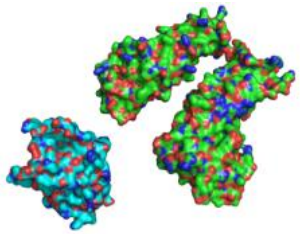
Tsubame 2.0 will have a dynamic provisioning system that works with batch queue systems and cluster management systems to dynamically provision compute resources

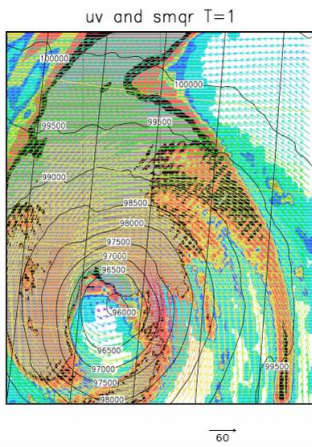
- Pool nodes can be dynamicaly allocated to various supercomputing services
- Linux and Windows batch queue systems coordinate to manage the nodes
- Dynamic increase/decrease of nodes allocated to particular services
  ※Increse / Decrease of Windows HPC and Linux nodes of different configurations
- Virtual Nodes on VMs on respective OSes can be subject to dynamically allocated and scheduled.



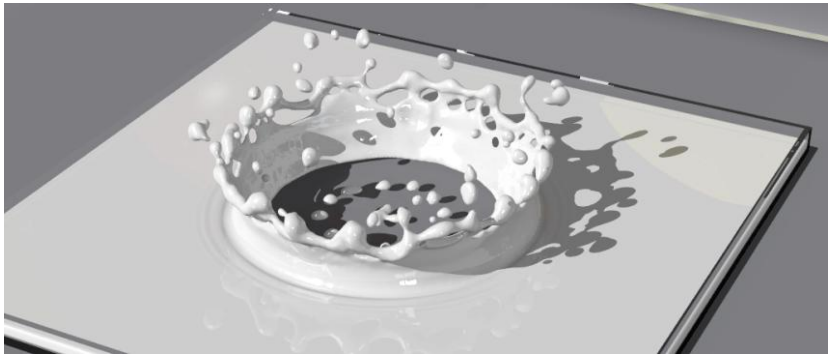
TSUBAME2.0 Estimated Performances

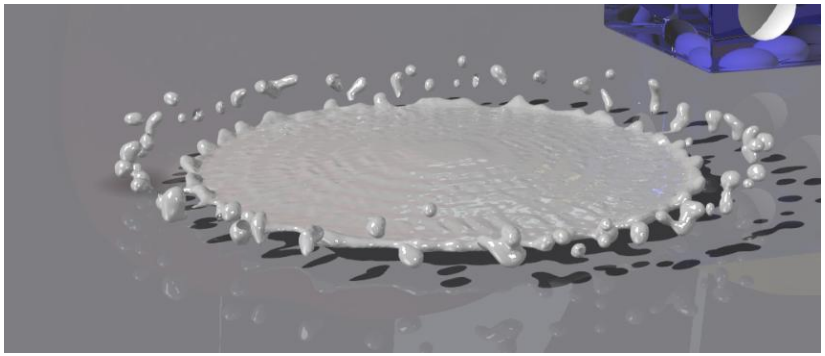
- 1.192 TFlops Linpack [IEEE IPDPS 2010]
  - Top ranks Green 500?
- ~0.5 PF 3D Protein Rigid Docking (Node 3-D FFT) [SC08, SC09]
- 145Tlops ASUCA Weather Forecast [SC10 Best Student Paper Finalist]
- Multiscale Simulation of Cardiovascular flows [SC10 Gordon Bell Finalist]
- Various FEM, Genomics, MD/MO (w/IMS), CS-Apps: search, optimization,





Gas-Liquid Two-Phase Flow





Fluid Equations

- Navier-Stokes solver : Fractional Step
- Time integration : 3rd TVD Runge-Kutta
- Advection term : 5th WENO
- Diffusion term : 4th FD
- Poisson : AMG-BiCGstab
- Surface tension : CSF model
- Surface capture : CLSVOF(THINC + Level-Set)

Continuum eq.

$$\nabla \cdot \mathbf{u} = 0$$

Momentum eq.

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \frac{1}{\rho}\mathbf{F}$$

Level-Set advection

$$\frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla)\,\phi = 0$$

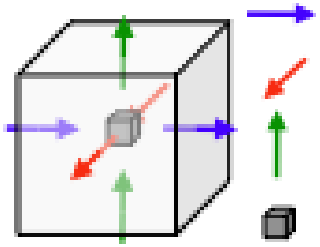
VOF continuum eq.

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{u}\psi) = 0$$

Level-Set re-initialization

$$\frac{\partial \phi}{\partial \tau} = sgn(\phi)\,(1 - |\nabla \phi|)$$

Staggered variable position



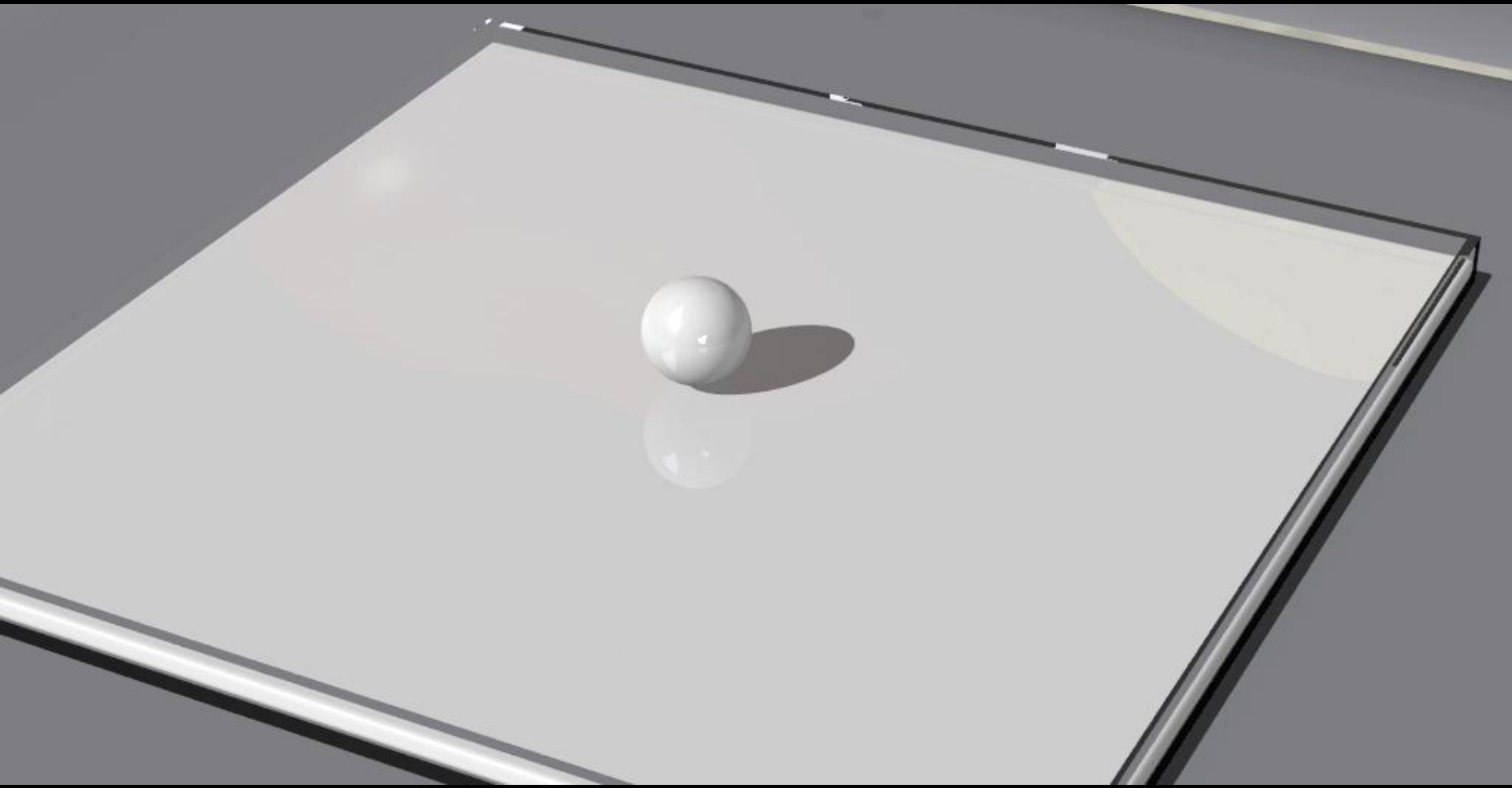
u : velocity on x-direction

v : velocity on y-direction

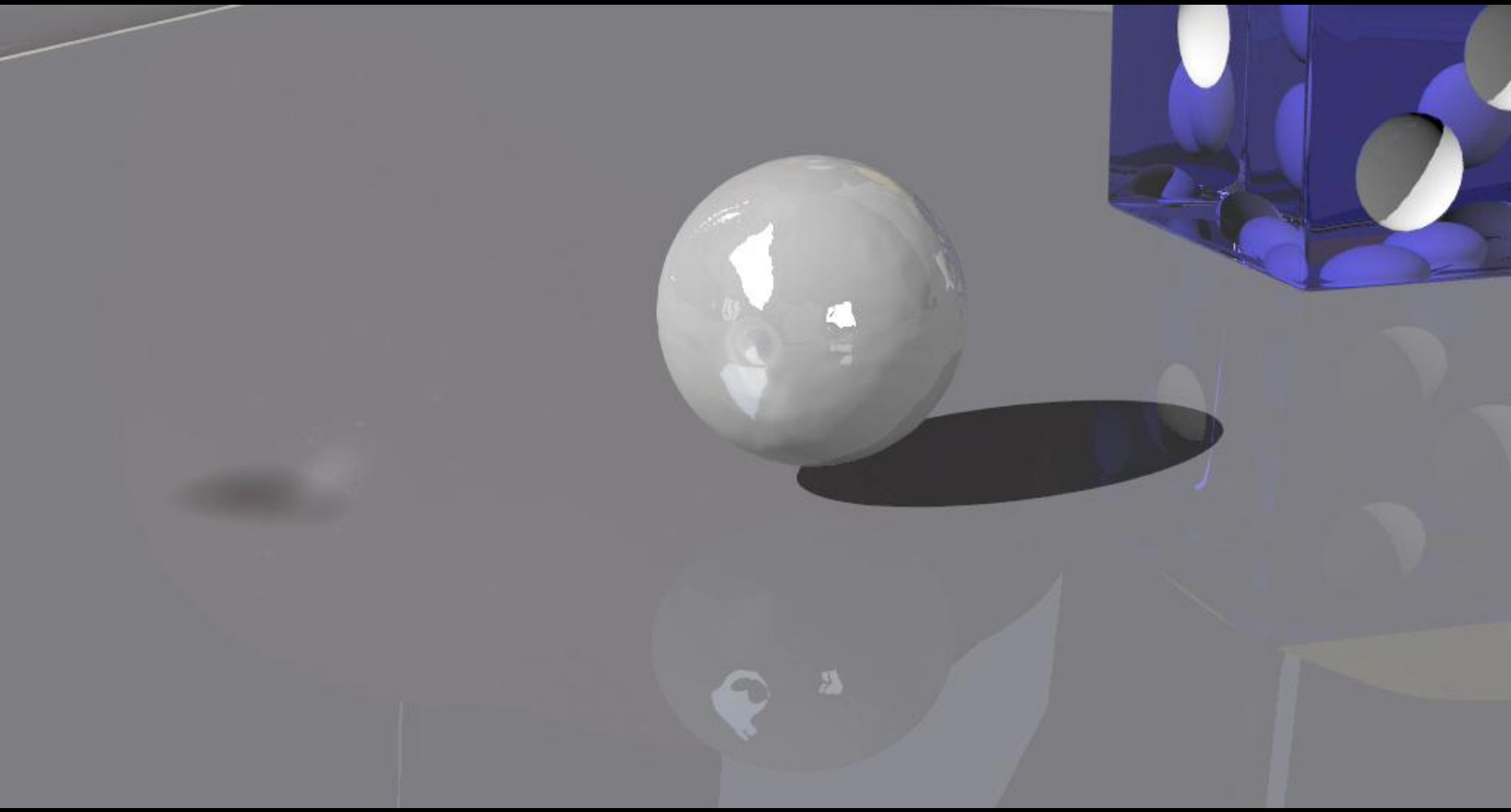
w : velocity on z-direction

p,φ,ψρ : scalar variables

Milk Crown



Drop on dry floor



Dendrite Solidification

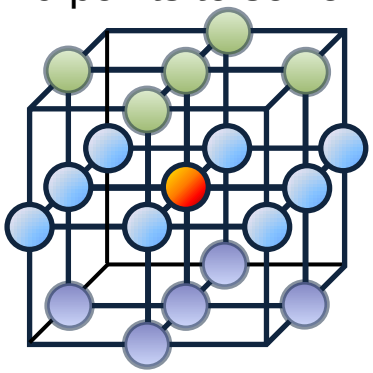
Allen-Cahn Equation based on Phase Field Model

$$\frac{\partial \phi}{\partial t} = M\left[\frac{\partial}{\partial x}\left(\varepsilon^2\frac{\partial \phi}{\partial x}+\varepsilon\frac{\partial \varepsilon}{\partial \phi_x}|\nabla\phi|^2\right)+\frac{\partial}{\partial y}\left(\varepsilon^2\frac{\partial \phi}{\partial y}+\varepsilon\frac{\partial \varepsilon}{\partial \phi_y}|\nabla\phi|^2\right)+\frac{\partial}{\partial z}\left(\varepsilon^2\frac{\partial \phi}{\partial z}+\varepsilon\frac{\partial \varepsilon}{\partial \phi_z}|\nabla\phi|^2\right)+4W\phi(1-\phi)\left\{\phi-\frac{1}{2}+\beta+a\chi\right\}\right]$$

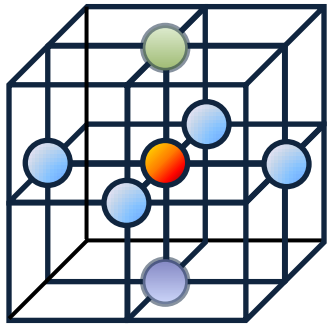
Thermal Conduction

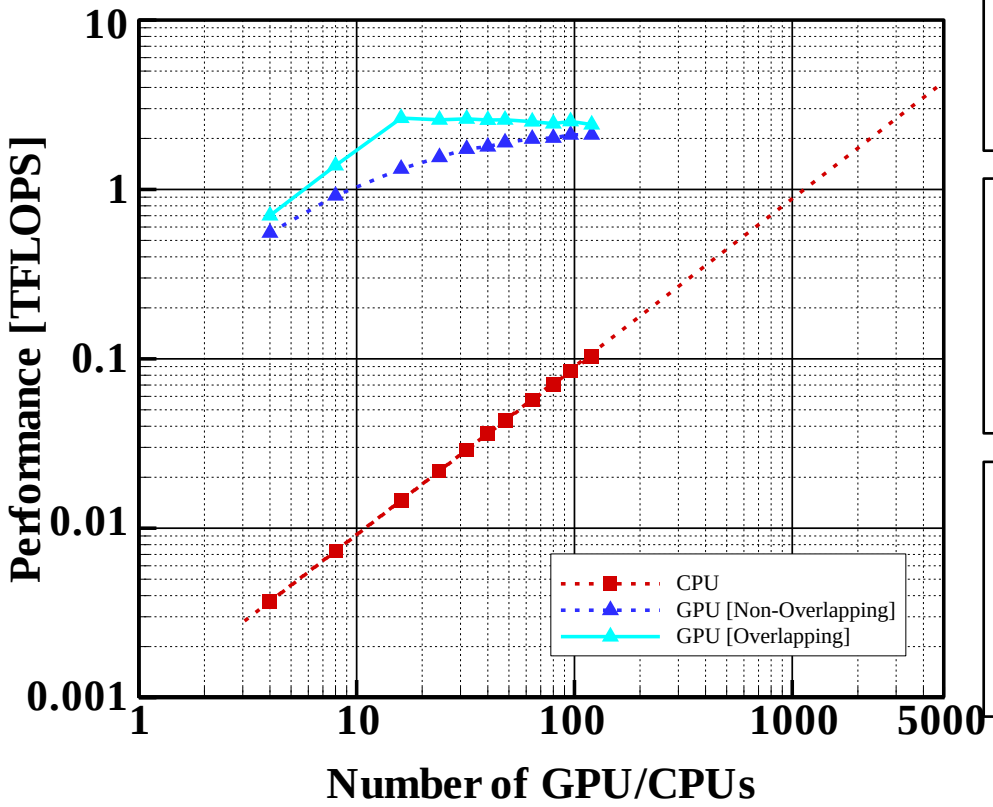
$$\frac{\partial T}{\partial t} = k\nabla^2 T + \frac{L}{C}30\phi^2(1-\phi)\frac{\partial \phi}{\partial t}$$

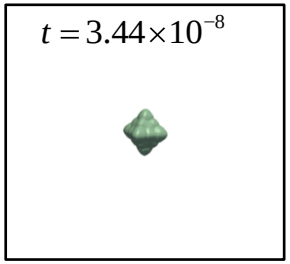
ϕ : 19 points to solve

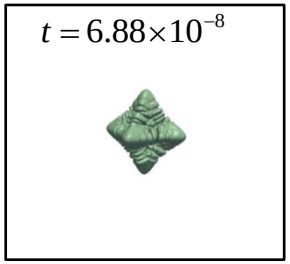


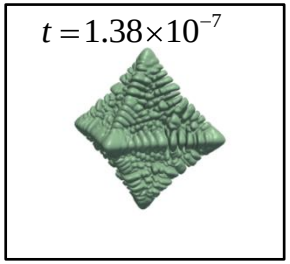
T : 7 points to solve

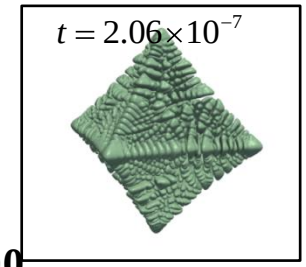


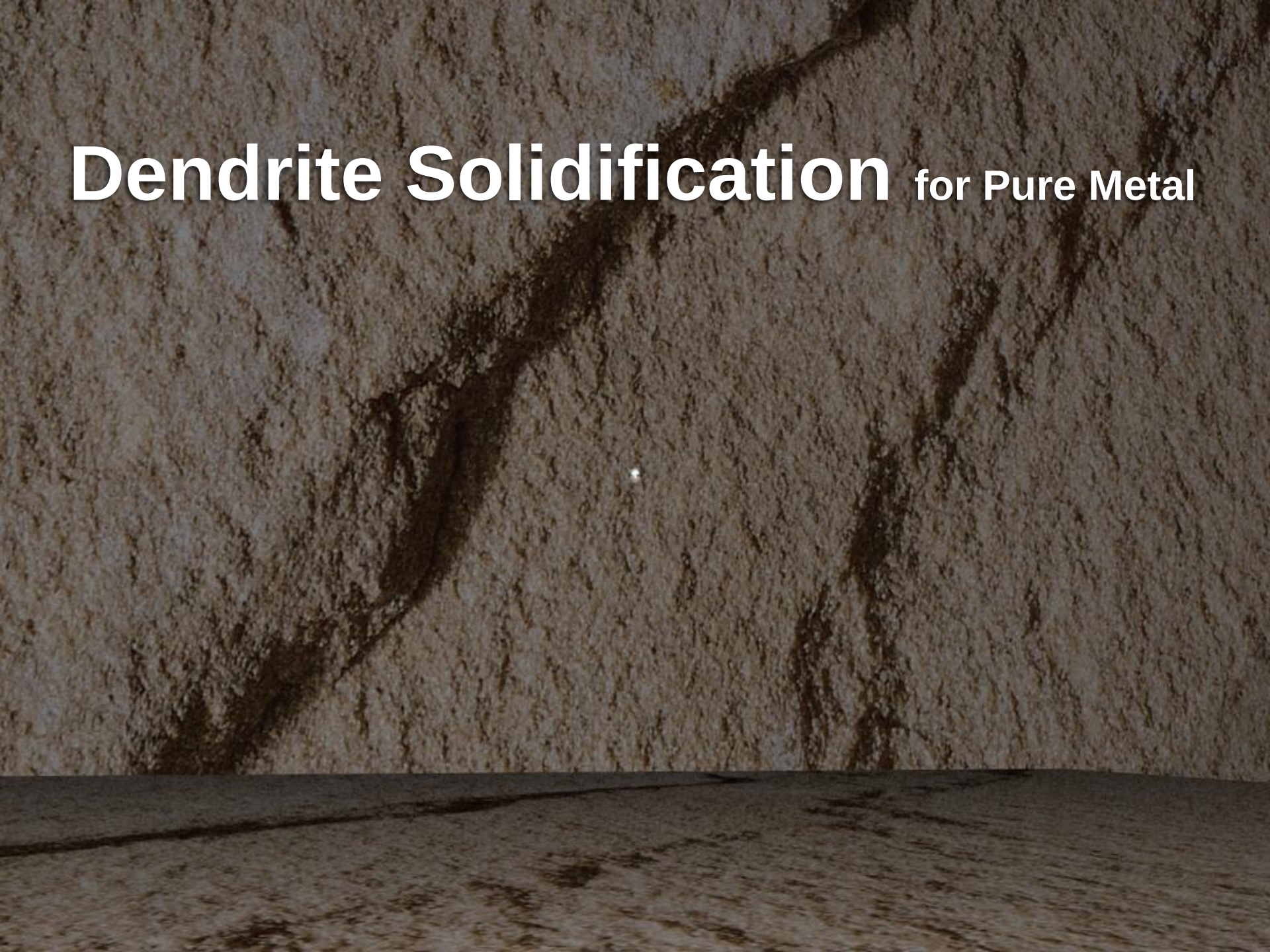











Dendrite Solidification for Pure Metal

Pulmonary Airflow Study

Prof. Takayuki Aoki
Collaboration with Tohoku University

Lattice Boltzmann Method

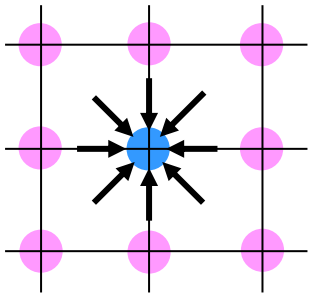
$$\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i = -\frac{1}{\lambda}\left(f_i - f_i^{eq}\right) \quad f_i^{eq} = \rho w_i\left[1 + \frac{3}{c^2}\left(\mathbf{e}_i \cdot \mathbf{u}\right) + \frac{9}{2c^4}\left(\mathbf{e}_i \cdot \mathbf{u}\right)^2 - \frac{3}{2c^2}\left(\mathbf{u} \cdot \mathbf{u}\right)\right]$$

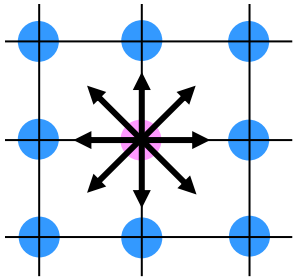
Collision step:

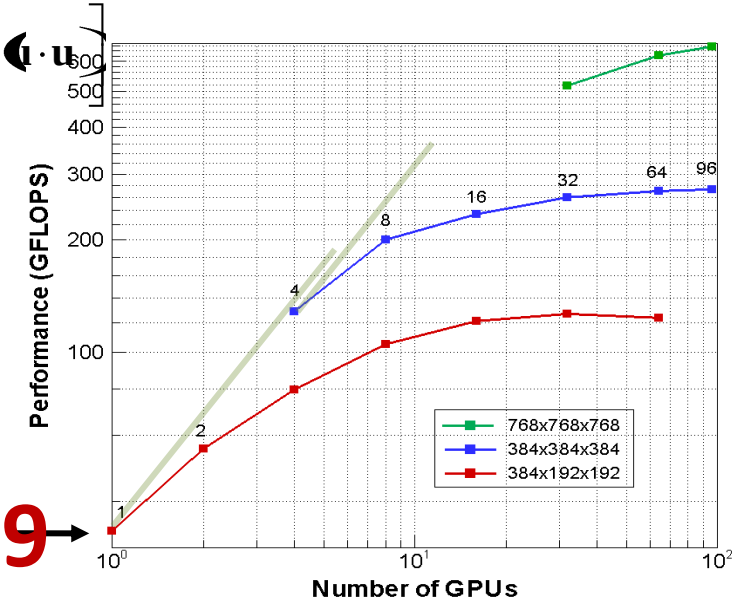
$$\bar{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) - \frac{1}{\tau}\left(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)\right)$$

Streaming step:

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = \bar{f}_i(\mathbf{x}, t)$$

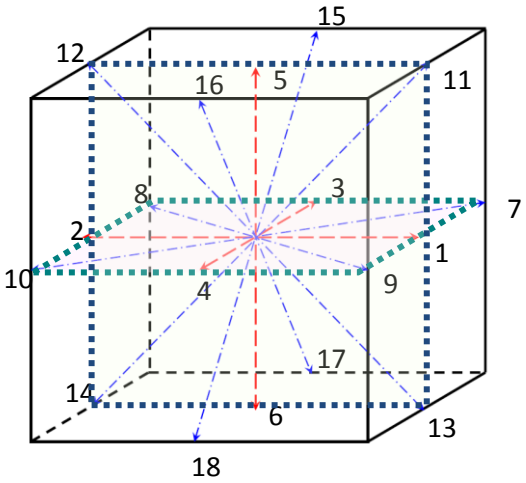




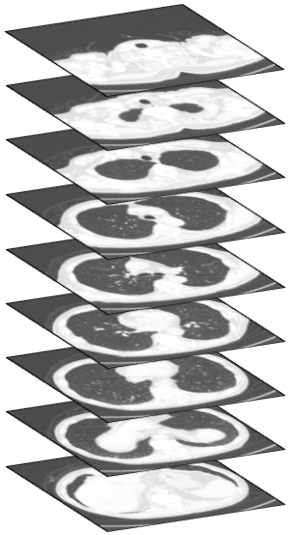


×89

D3Q19



X-Ray CT images



Airway structure Extraction

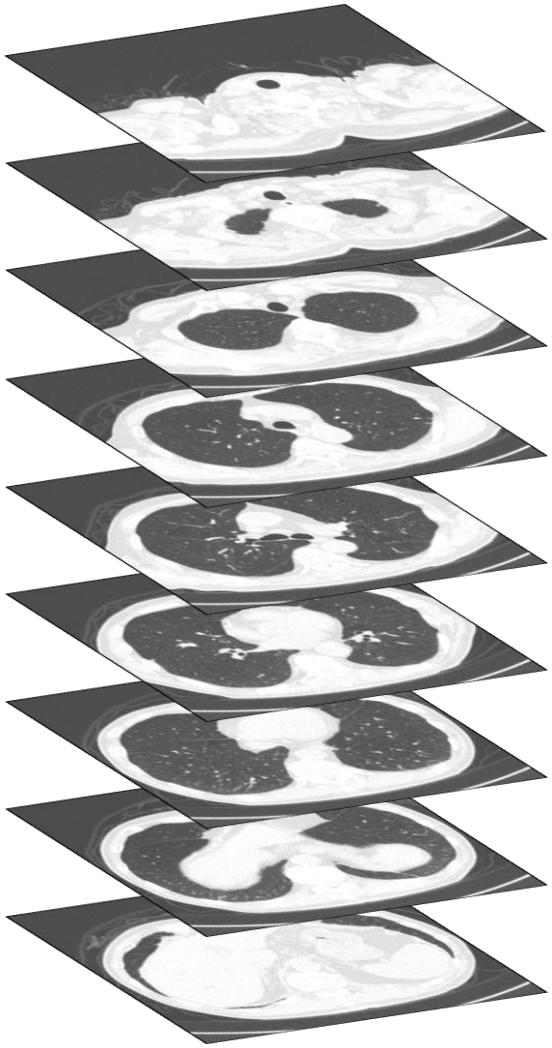
Lattice Boltzmann GPU computing

Pulmonary Airflow Study

Collaboration with Tohoku University

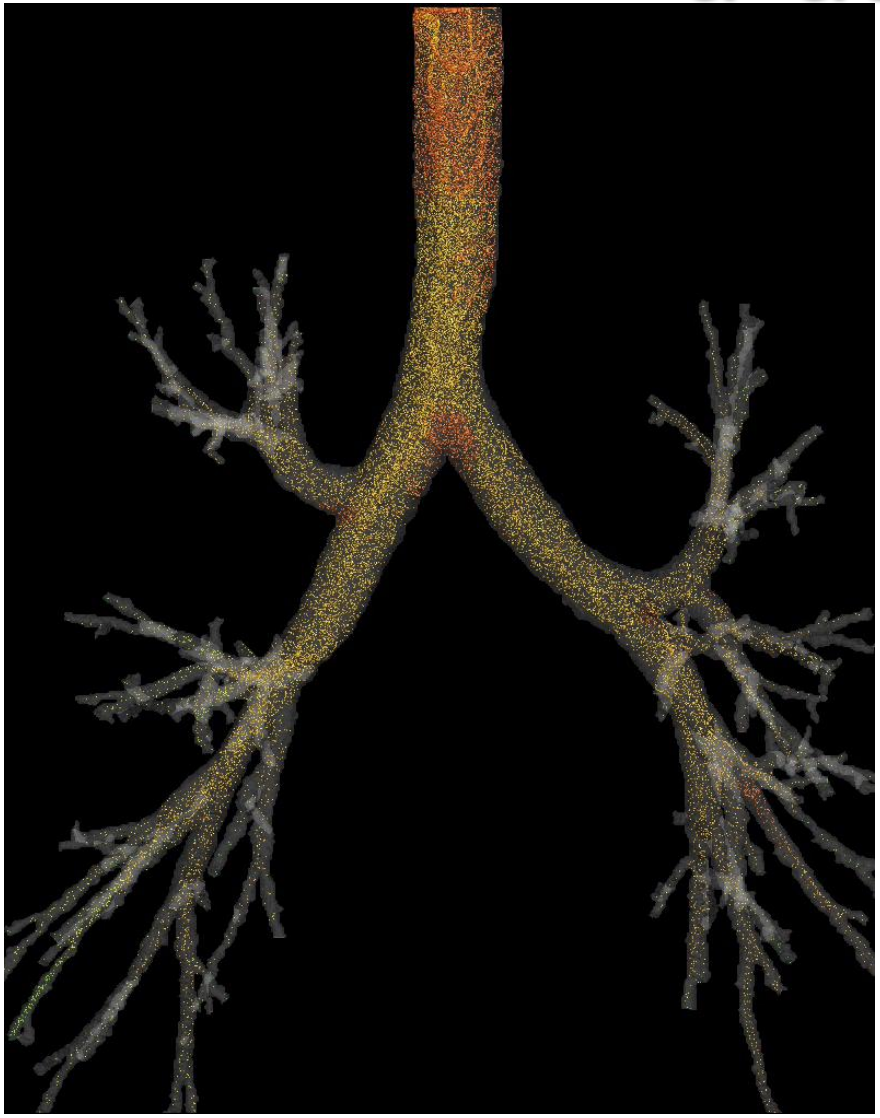
GP GPU

X-Ray CT images



Airway structure Extraction

Lattice Boltzmann GPU computing



Next Generation Numerical Weather Prediction[SC10]

Collaboration: Japan Meteorological Agency

Meso-scale Atmosphere Model:

Cloud Resolving Non-hydrostatic model
[Shimokawabe et. al. SC10 BSP Finalist]

ex. WRF(Weather Research and Forecast)

Typhoon ~ 1000km

1~ 10km
Tornado,
Down burst,
Heavy Rain

WSM5 (WRF Single Moment 5-tracer) Microphysics*

Represents condensation, precipitation and thermodynamic effects of latent heat release

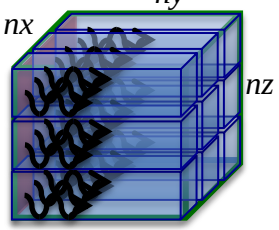
1 % of lines of code, 25 % of elapsed time

⇒ 20 x boost in microphysics (1.2 - 1.3 x overall improvement)

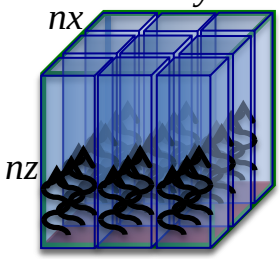
ASUCA : full GPU Implementation
developed by Japan Meteorological Agency

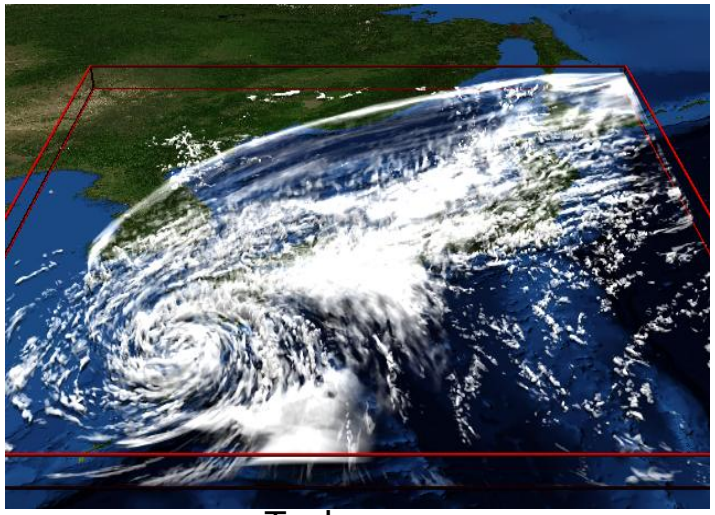
TSUBAME 2.0 : 145 Tflops
World Record !!!

Block Division for Advection

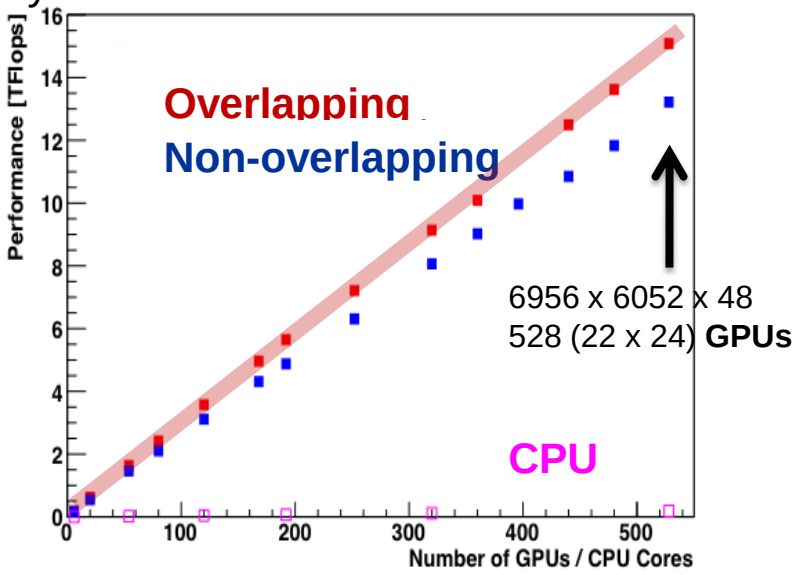


for 1-D Helmholtz eq.

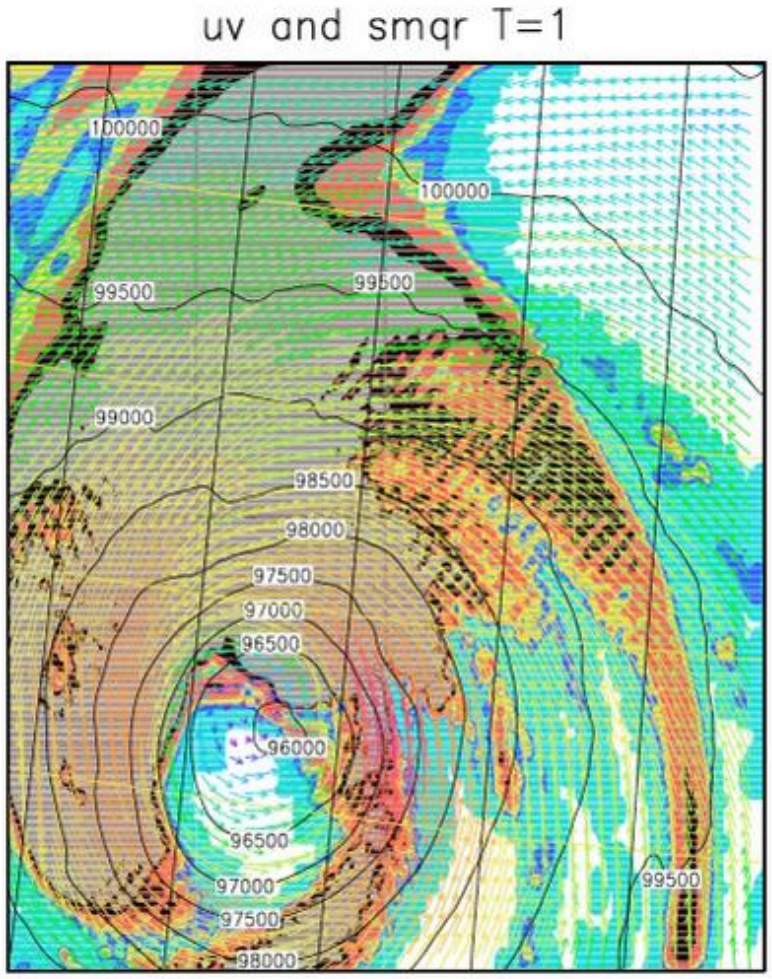




Typhoon



ASUCA Typhoon Simulation 2km mesh 3164×3028×48



70 minutes wallclock time for 6 hour simulation time (x5 faster)

Expect real-time with 0.5km mesh (for Clouds) on TSUBAME2.0

MUPHY : Multi-Physics Simulator

[Massimo Bernaschi]

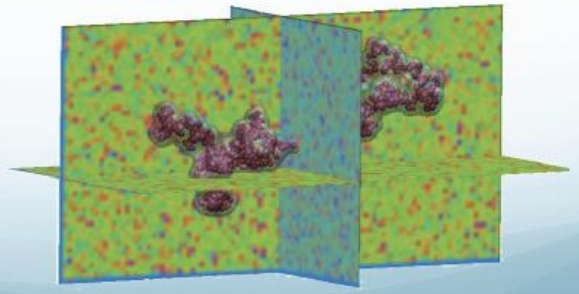
road spectrum of ***fluid-particle*** coupling mechanisms

brary of particle and molecular representations

brary of fluid types

o-polymers, colloidal suspensions, gels, biofluids, …

ulti-Platform

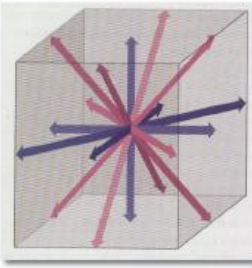


Blood Plasma via Lattice Boltzmann

From (minimal) Boltzmann equation

$$(\partial_t + \vec{v}\partial_x) f(\vec{x},\vec{v},t) = -\omega(f - f^{eq})(\vec{x},\vec{v},t)$$

Collision + Streaming of a set of discrete velocities



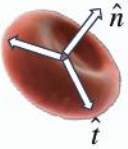
Superset of the Navier-Stokes dynamics

Exact streaming (no self-convected): uniform mesh

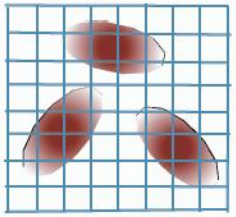
Complexity O(N)

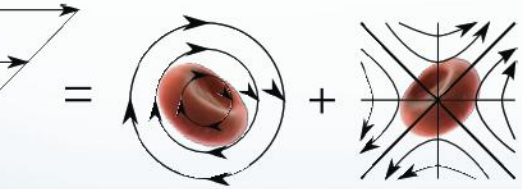
Enable complex geometries

Ellipsoidal Suspensions of RBC



$$\tilde{\delta}_{\xi_\parallel \xi_\perp}(x) = \prod_{\alpha=x,y,z} \tilde{\delta}_{\xi_\alpha}((Qx)_\alpha)$$





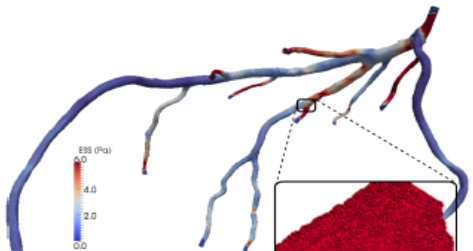
- ✓Tensorial drag
- ✓Tank Treading
- ✓Complex Rheology

Strip off DOFs: O(10) per RBC

MUPHY Performance on the TiTech GPU cluster

The performance of 1536 GPUs for the Lattice Boltzmann kernel is the same as the whole Jülich Bluegene/P system (294712 cores!)

# of GPUs	time	efficiency
256	76.16	N.A.
512	38.52	98.86%
1024	19.95	95.37%
1536	13.43	94.43%



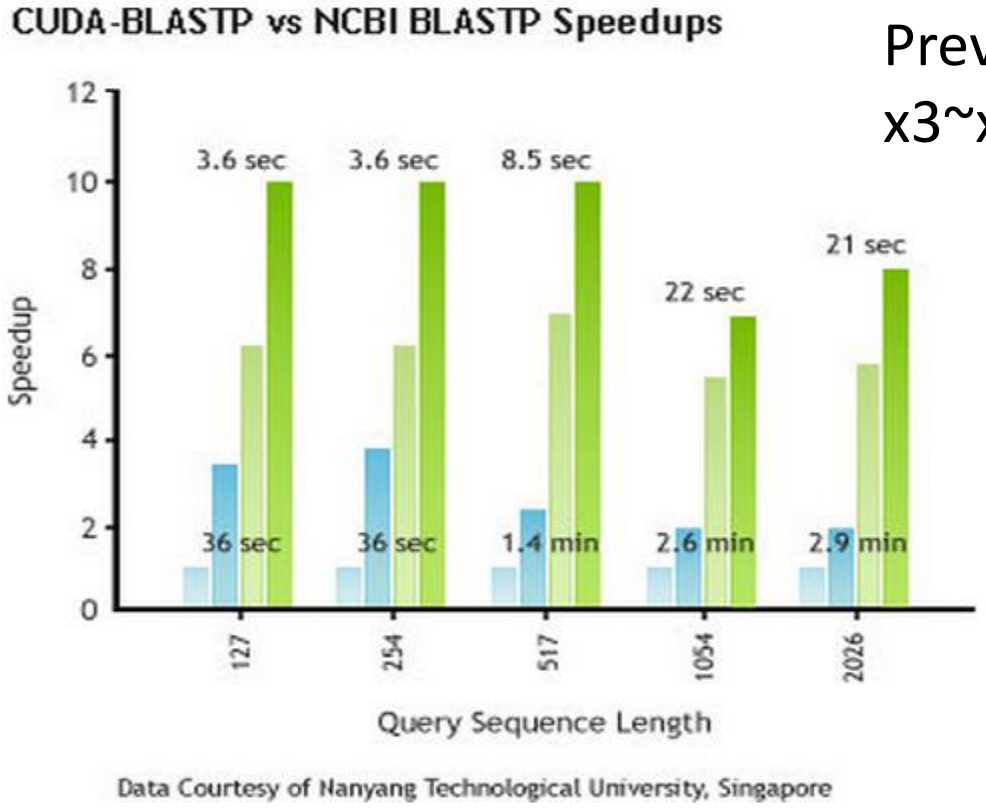
x3 40 rack BG/P

# of GPUs	time
256	648.23
512	327.97

LB and Molecular Dynamics

Bioinformatics: BLAST on GPUs

- CUDA-BLASTP (NTU)
  http://www.nvidia.com/object/blastp_on_tesla.html



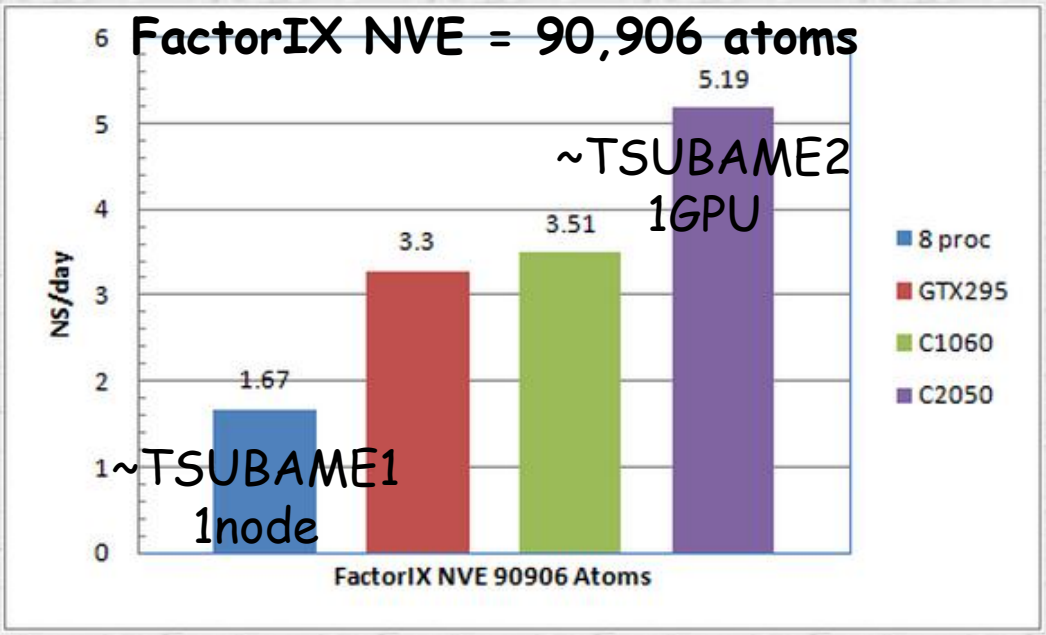
Previous-Generation CPU/GPU
x3~x5 GPU per socket speedup

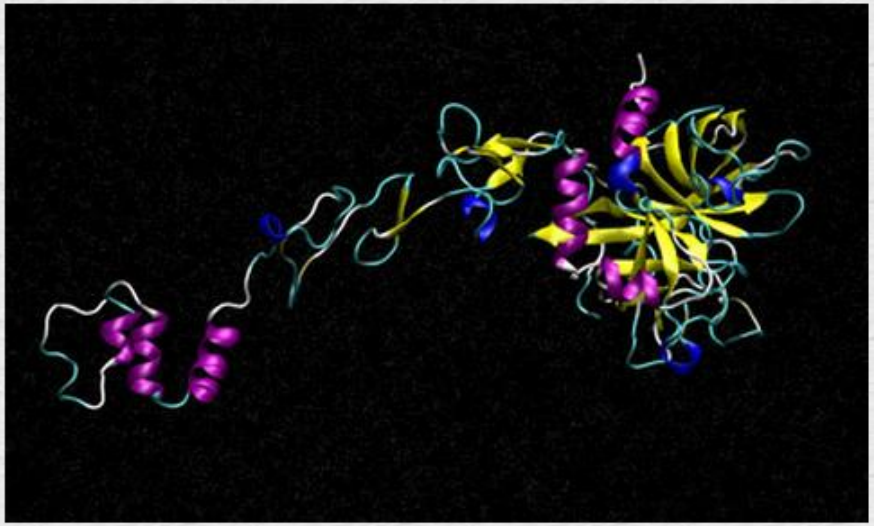
TSUBAME2.0
equivalent to 16,000
sockets (100,000 cores)

- GPU-BLAST (CMU)
  http://eudoxus.cheme.cmu.edu/gpublast/gpublast.html
  "4 times speedup on Fermi GPUs"

GPU Enabled Amber 11

- 1GPU ~= 8 Core CPU node x 4~20
  - Already in service on TSUBAME2.0
  - Waiting for Multi-GPU version presented at GTC 2010…
- Equivalent to 300,000 CPU cores on TSUBAME2.0
  - Assuming x10 speedup per socket





Electric Power Consumption
for Multiple-GPU Applications

GP GPU

CFD Applications

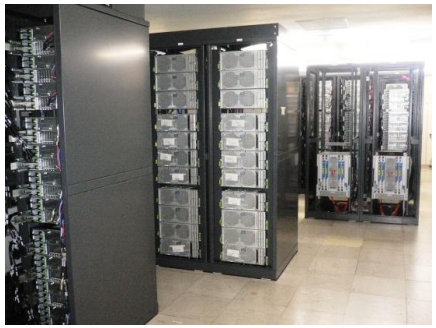

MAX 200W/GPU

Comparing to TSUBAME Opteron CPU 1 core,

Weather Prediction: x80 (10 GPU = 1000 CPU)

Lattice Boltzmann: ~ x100

Phase Field Model: x170


TSUBAME ~
1MW/10000CPU

When we assume the acceleration is typically x100,

1 GPU < 200W for our applications

100 GPU : 20 kW

10000 CPU core of TSUBAME 1.2 : 1MW (1000kW)

1/50 Ultra low Power

CompView

GSIC
Global Scientific Information and Computing Center

But it is not just HW Design…SOFTWARE(!)

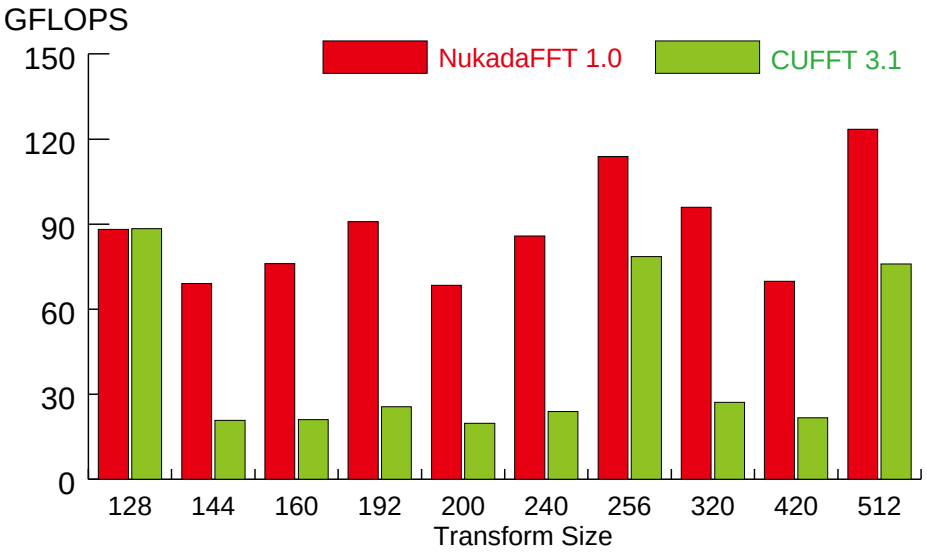
- The ***entire software stack*** must preserve bandwidth, hide latency, lower power, heighten reliability for Petascaling
- Example: TSUBAME1.2, Inter-node GPU ⇔ GPU achieves only 100-200MB/s in real apps
  - c.f. 1-2GB/s HW dual rail IB capability
  - Overhead due to unpinnable buffer memory?
  - New Mellanox driver will partially resolve this?
  - Still need programming model for GPU ⇔ GPU
- Our SW research as CUDA CoE (and other projects such as Ultra Low Power HPC)

Tokyo Tech CCOE
1. Auto-Tuning GPU Numerical Libraries

NukadaFFT 1.0 release !? [SC08,SC09]

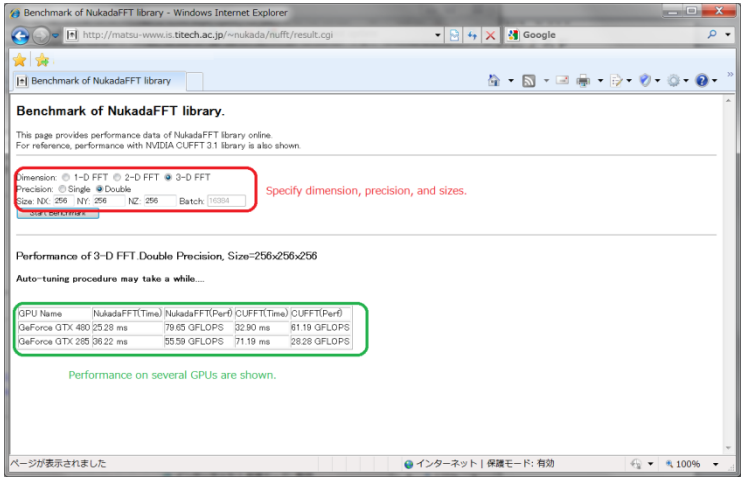
NukadaFFT library is a very fast auto-tuning FFT library for CUDA GPUs.

Tuning Parameters:
(1) Factorization of transform size.
(2) # of threads launched per SM.
(3) Padding insertion patters for shared memory
The library generates many kinds of FFT kernels and selects the fastest one. (exhaustive)



Performance of 1-D FFT.
(Double Precision, batch=32,768, GPU=GeForce GTX 480.)

For more details, you can see GTC 2010 Research Poster, or catch the author.



You can try online benchmarking as for the size you are interested in.

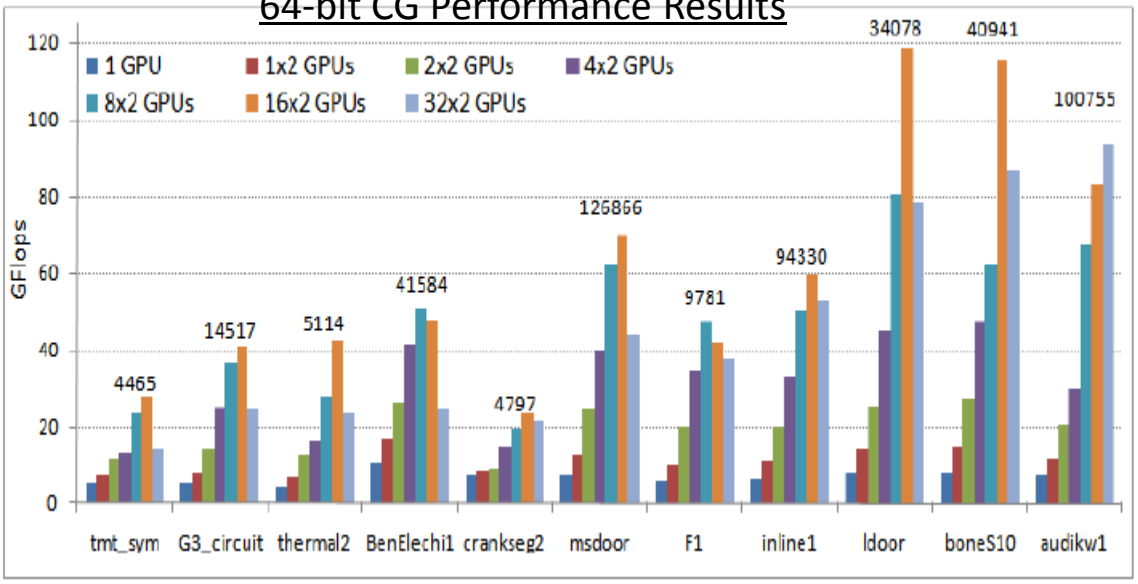
http://matsu-www.is.titech.ac.jp/~nukada/nufft/

"Auto-Tuning" Sparse Iterative Solvers on a Multi-GPU Clusters

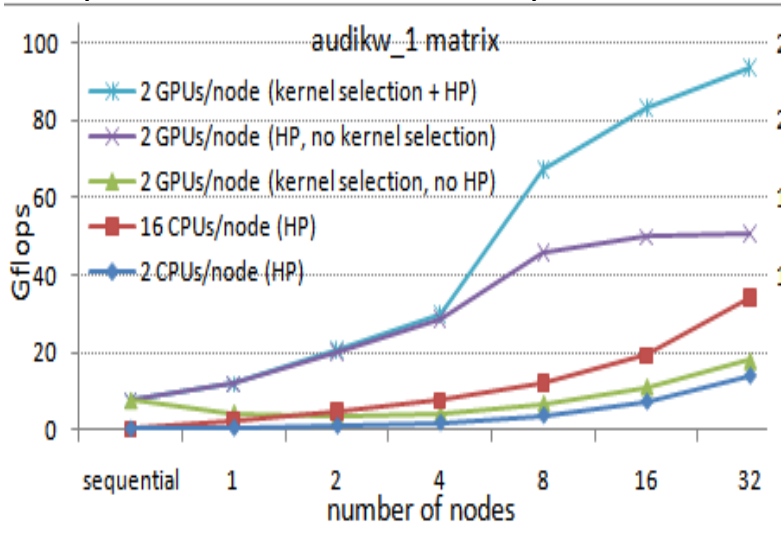
(A. Cevahir, A. Nukada, S. Matsuoka)

- High communication due to irregular sparsity
  - GPU computing is fast, communication bottleneck is more severe
- We propose techniques to achieve scalable parallelization
  - A new sparse matrix-vector multiplication (MxV) kernel [*ICCS'09*]
  - **Auto-selection of optimal running MxV kernel** [*ISC'10*]
  - Minimization of communication between GPUs [*ISC'10*]
- Two methods: **CG and PageRank (on 100CPUs)**

64-bit CG Performance Results



Comparisons of different CG implementations



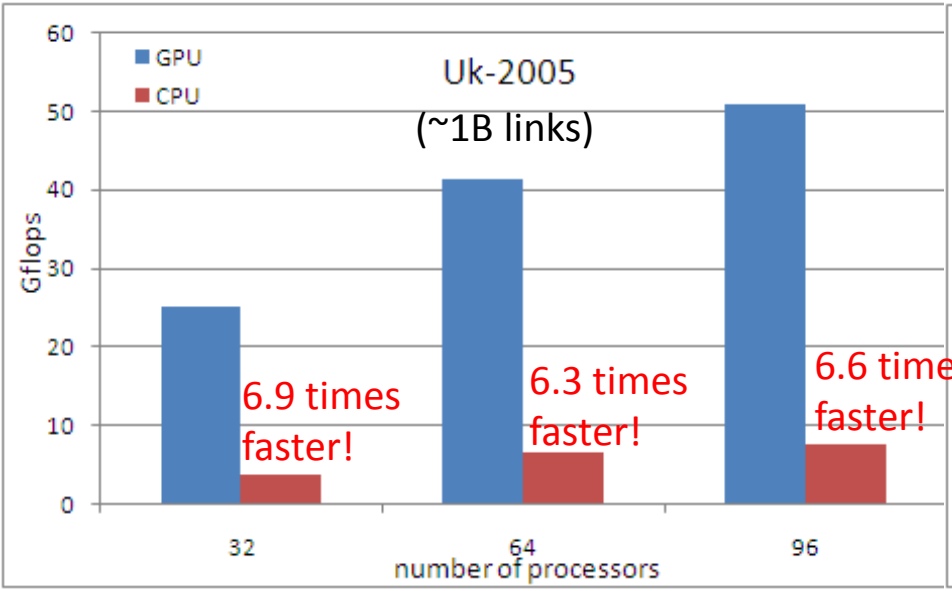
PageRank Results

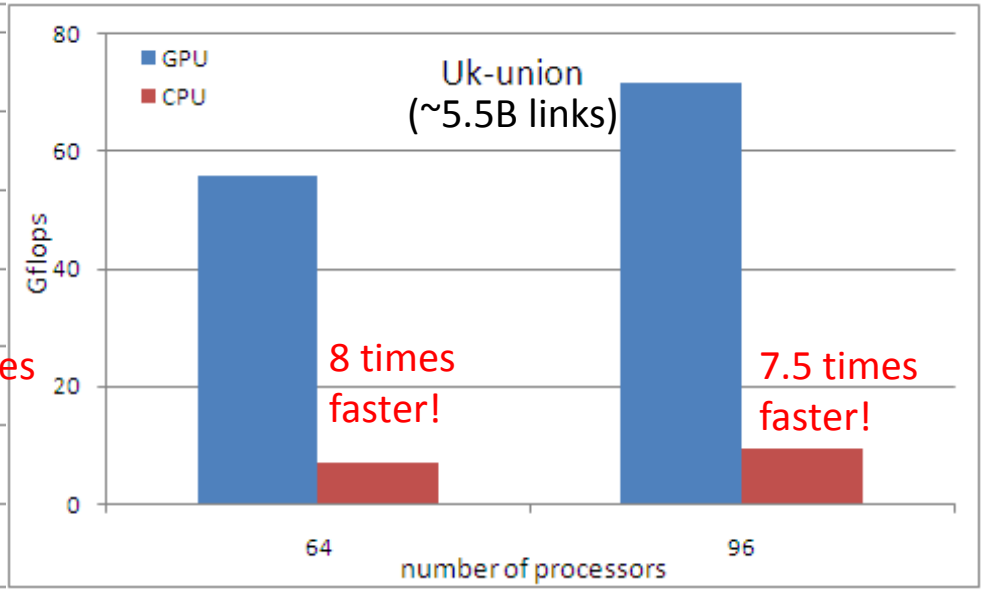
(A. Cevahir, A. Nukada, S. Matsuoka)

- Huge data is processed -- bigger data scales better
- A state-of-the-art PageRank method is implemented

(based on *Cevahir, Aykanat, Turk, Cambazoglu: IEEE TPDS June 2010*)

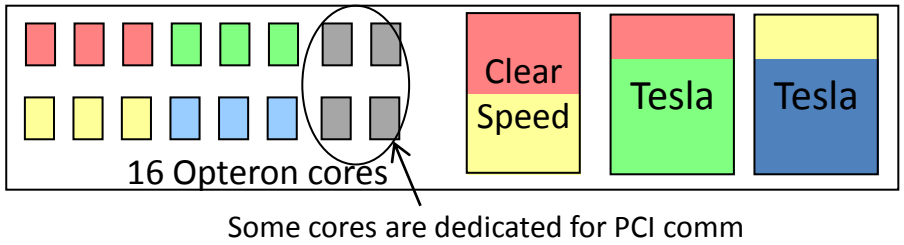
(Problem bandwidth restricted – socket - to - socket comparison)





High performance Linpack on heterogeneous Supercomputers [IPDPS 08,10]

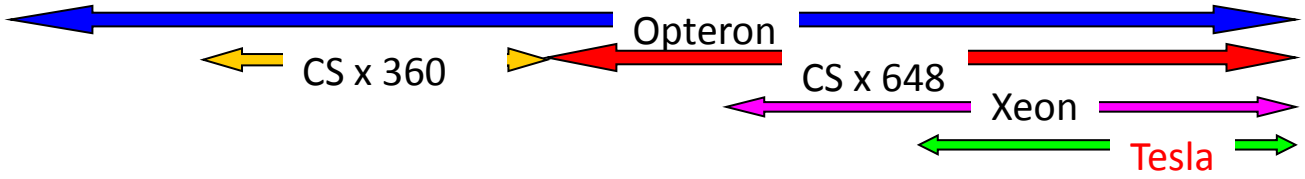
- Dense linear computation on large heterogeneous systems
  - All types of processors are used for kernel (DGEMM)
  - Even supporting "heterogeneous nodes"
  - Load balancing techniques, tuning for reducing PCIe communication



Mapping between MPI processes and heterogeneous processors

- >10,000 CPU cores, >600 Teslas, >600 ClearSpeed are cooperatively used
  - 87.01TFlops, 53% efficiency → 7 times improvements on Top 500 ranking

	Jun06	Nov06	Jun07	Nov07	Jun08	Nov08	Jun09
Linpack speed	38.18TF	47.38	48.88	56.43	67.70	77.48	87.01
Rank	7	9	14	16	24	29	41



Tokyo Tech CCOE
2. GPU Power Performance Modeling and Heterogeneous Scheduling

Measuring GPU Power Consumption

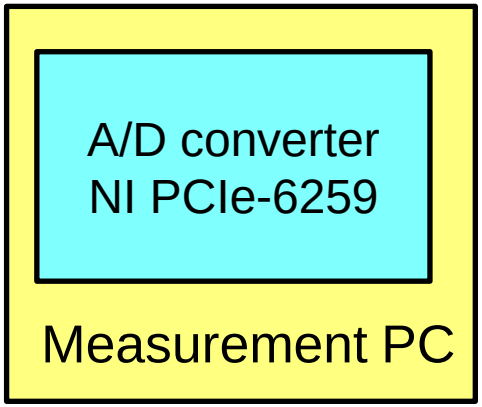
- Two power sources
  - Via PCI Express: < 75 W
  - Direct inputs from PSU: < 240 W
- Uses current clamp sensors around the power inputs

Precise and Accurate Measurement with Current Probes

Attaches current sensors to two power lines in PCIe

+

Direct power inputs from PSU



Reads currents at 100 us interval

Statistical Modeling of GPU Power Consumption [IEEE HPPAC10]

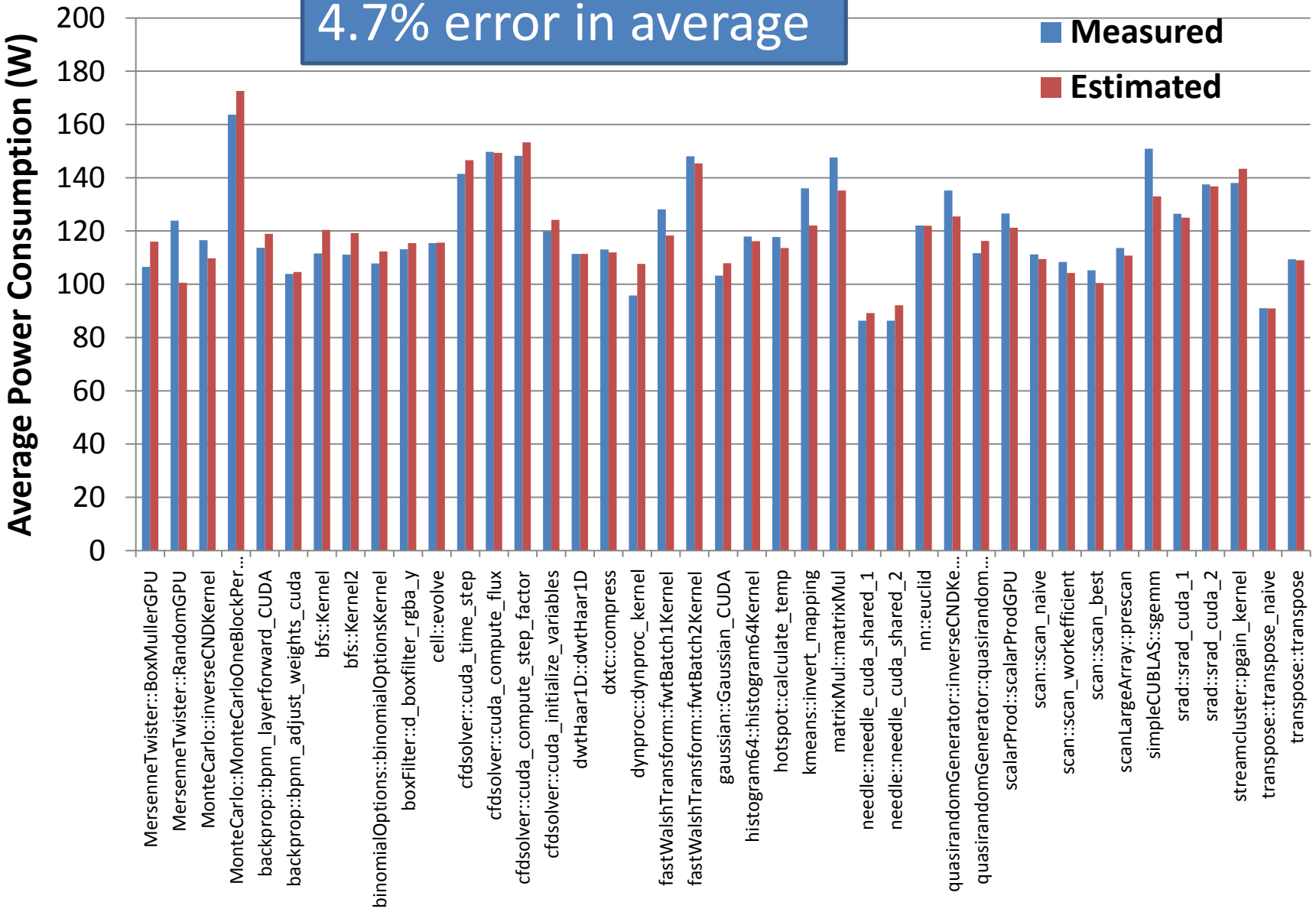
- Regularized linear regression
  - Finds linear correlation between per-second PMC values and average power of a kernel execution
  - Aggregates 3 kinds of global memory loads/stores
    - gld: gld_32b + gld_64b * 2 + gld_128b * 4
  - Regularization for avoiding overfitting to training data (Ridge Regression [10])

Average power of a kernel (Watts)

$$p = \sum_{i=1}^{n} \alpha_i c_i + \beta$$

Per-second PMC values

Actual vs. Estimated Power



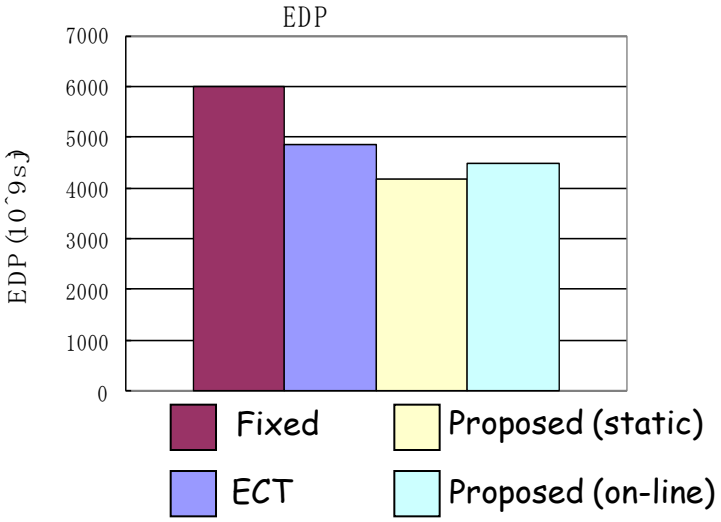
Low Power Scheduling in GPU Clusters

[IEEE IPDPS-HPPAC 09]

Objective : Optimize CPU/GPU Heterogeneous

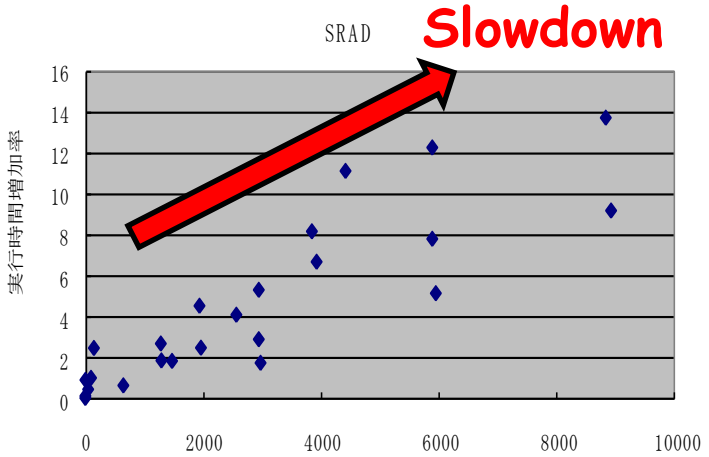
- Optimally schedule mixed sets of jobs executable on either CPU or GPU but w/different performance & power
- Assume GPU accel. factor (%) known

30% Improvement Energy-Delay Product



TODO: More realistic environment

- Different app. power profile
- PCI bus vs. memory conflict
  - GPU applications slow down by 10 % or more when co-scheduled with memory-intensive CPU app.

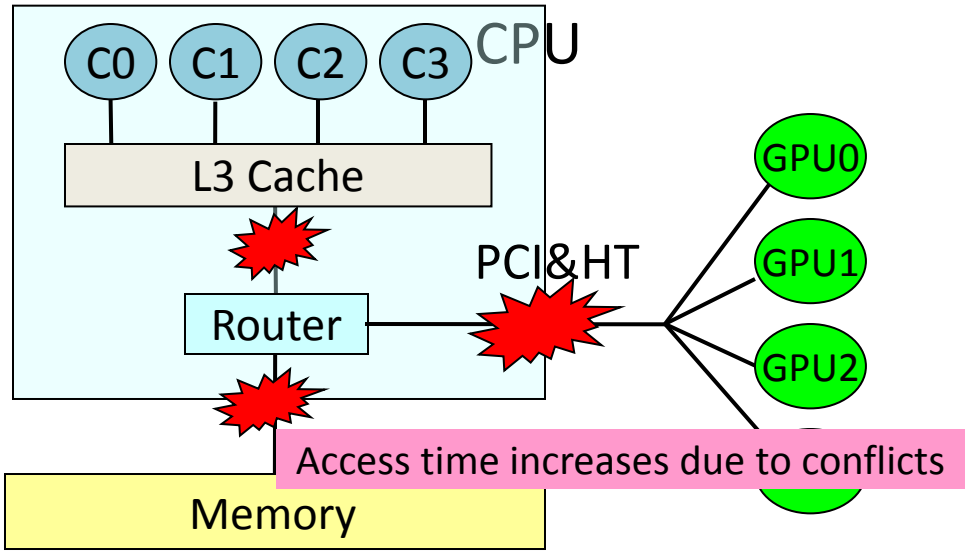


Intelligent Task Scheduling for GPU clusters

- considering conflicts on memory bus and PCI-E bus -

T. Hamano, et al. IPSJ SIGHPC-124 (domestic)

4 processes share a node.



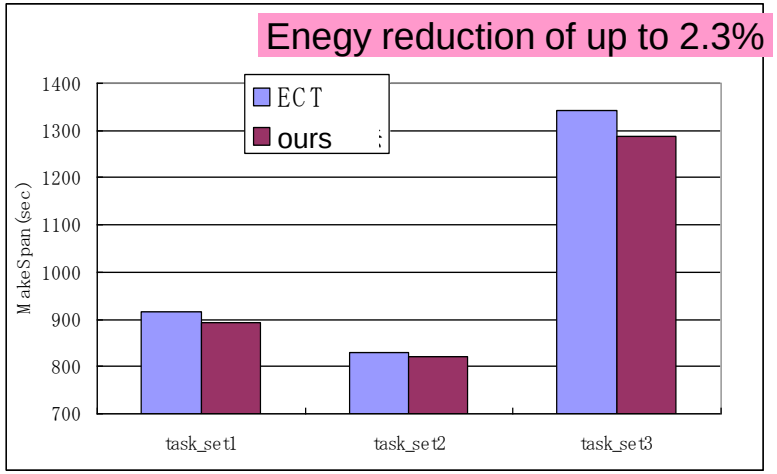
Status of each process is one of the following:
(1) Memory access
(2) PCI transfer (assume blocking cuMemcpy)
(3) No memory access
Speed-down ratio of a process = r (percentage of each status)×α (conflict coefficient)

$$\sum_{s0}\sum_{s1}\sum_{s2}\sum_{s3}\alpha(s0,\{s1,s2,s3\})r_0(s0)r_1(s1)r_2(s2)r_3(s3)$$

We integrated this model into ECT(Earliest Complete Time) scheduling.

Assuming the following information of each job is known
- Execution time (at stand alone case)
- Total memory access (from performance counter)
- Total PCI-E transfer (from CUDA profiler)

Actual execution time is estimated by our model.



Tokyo Tech CCOE
3. GPU Fault Tolerance

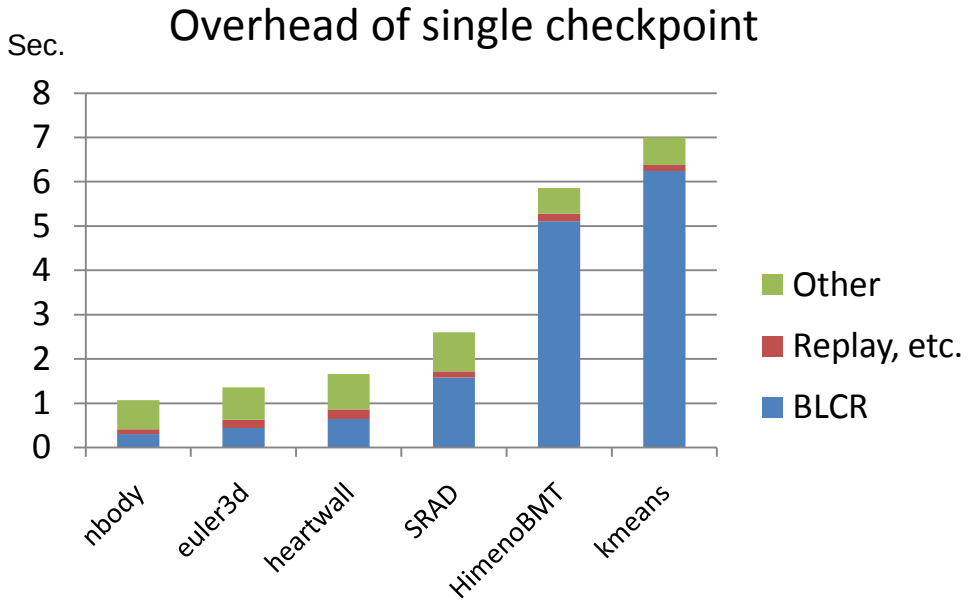
NVCR : A Transparent Checkpoint/Restart for CUDA

CUDA Checkpoint using BLCR (Takizawa, 2009)
- Save all data on CUDA resource.
- Destroy all CUDA context.
- BLCR
- Reallocate CUDA resource.
- Restore data.

After reallocation, the address or handle may differ from previous one.
Handles are Opaque pointers : can be virtualized.
Addresses must be visible for user application.

Our answer is `REPLAY'.
NVCR records calls of all APIs related to device memory address such as;
cuMem*(), cuArray*(), cuModule*().

We added some optimization of reducing the API records. For example, locally resolved alloc-free pairs are removed from the record.



The time for Replaying is quite negligible compared with the main BLCR procedure to write image to HDD.

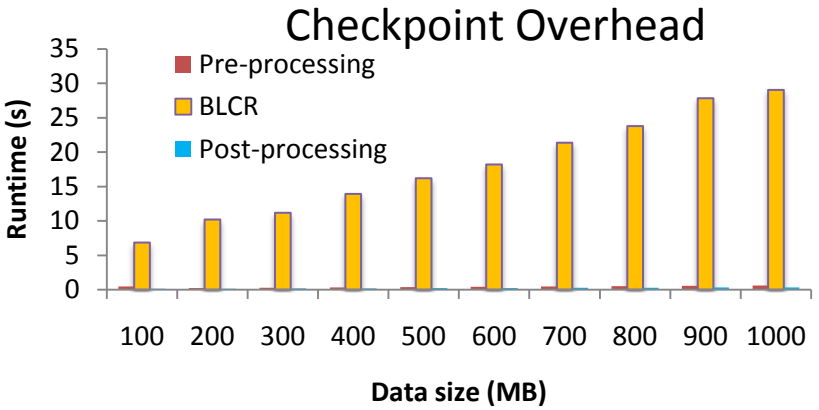
NVCR supports
- All CUDA 3.0 APIs
- OpenMPI
- Both CUDA runtime and CUDA driver API.

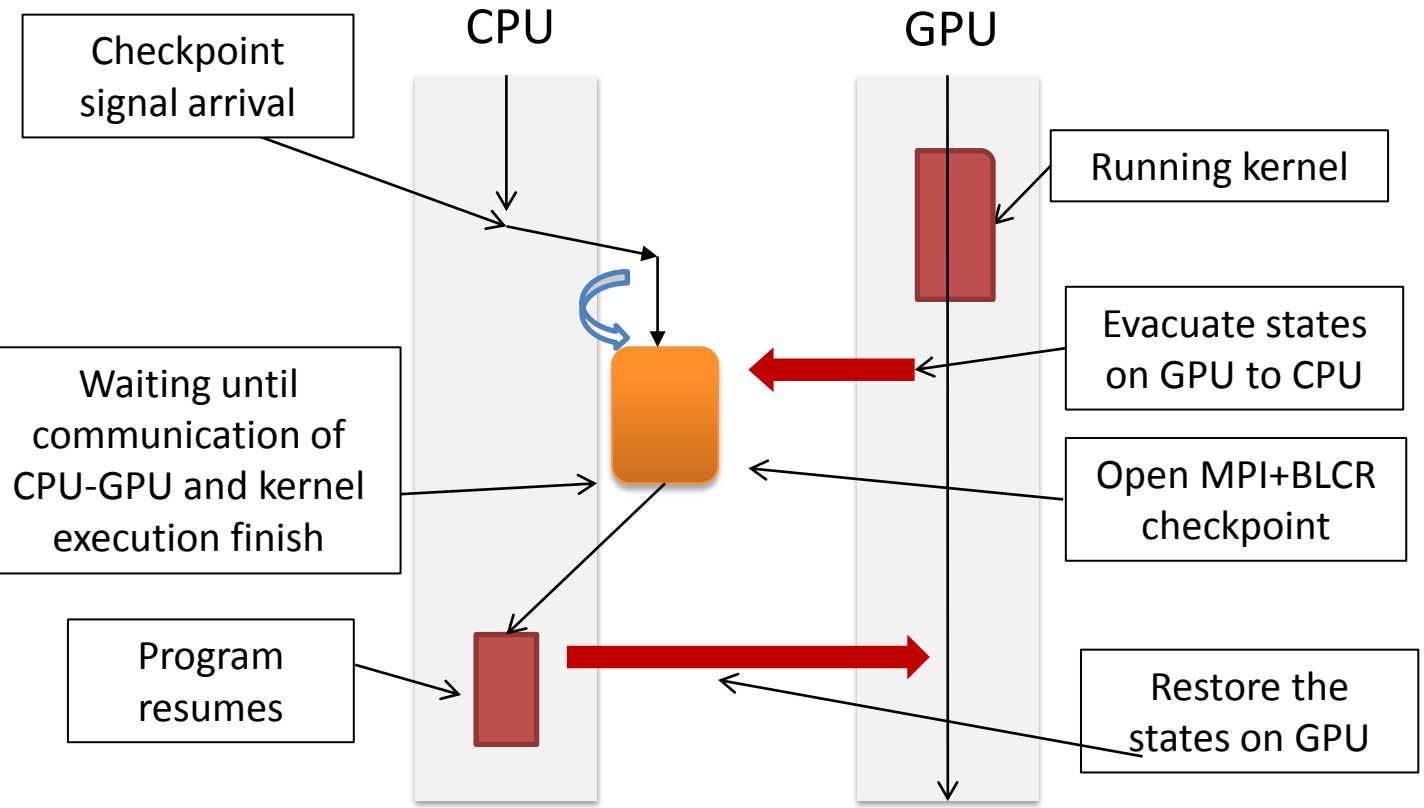
CUDA pinned memory is supported partially … Full support requires NVIDIA's official support.

MPI CUDA Checkpointing

Checkpointer for MPI+CUDA applications

- Very important in large-scale GPU systems
- Supports a majority of the CUDA RT API
- Based on BLCR and OpenMPI



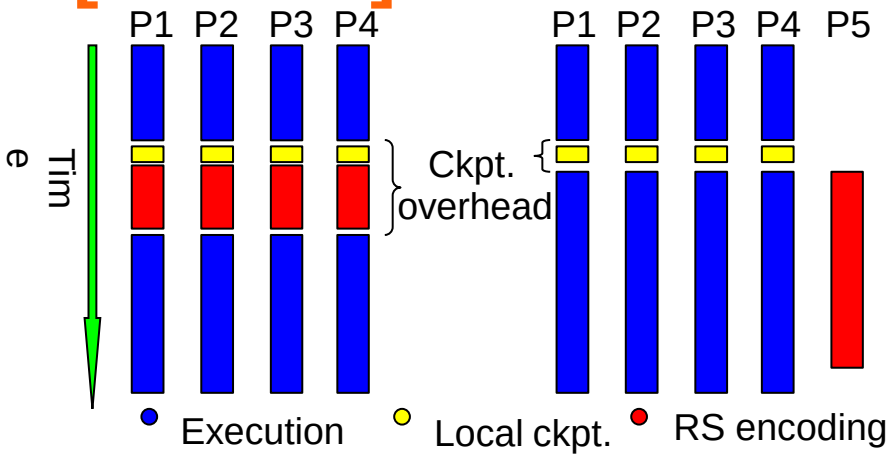


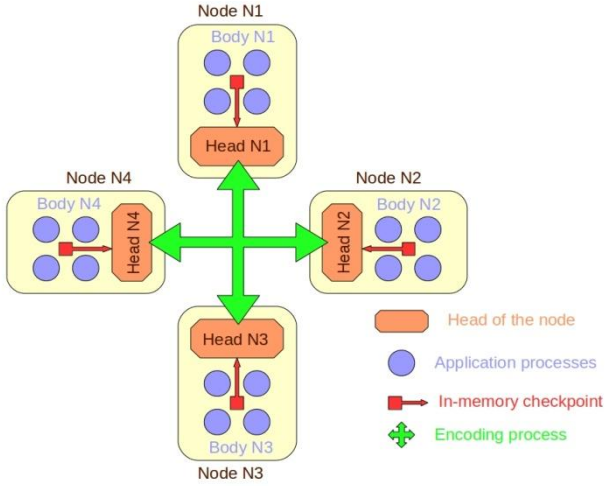
- GPU state saving and restoring has little overall performance impact.
- Can be significantly faster with our scalable checkpointing algorithms

More details at the Research Poster Session

Hybrid Diskless Checkpoint

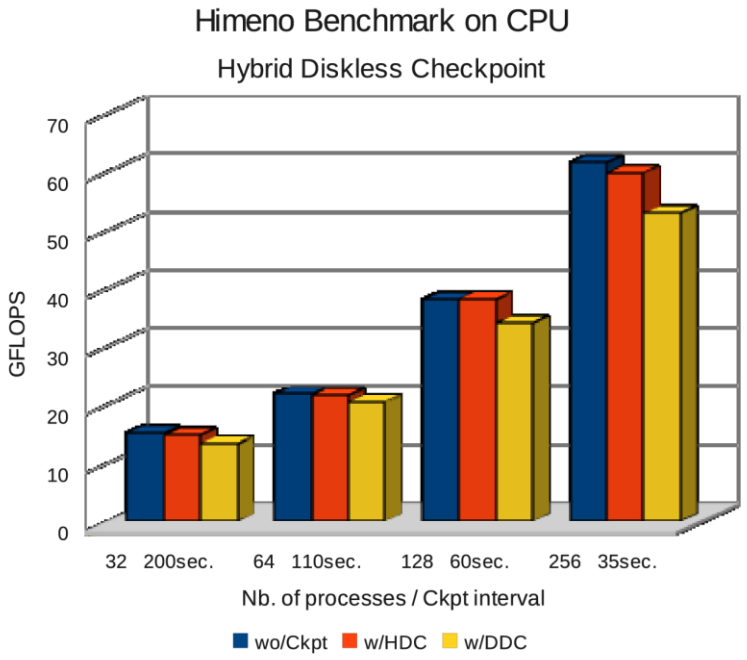
- ***Problem:*** *Decrease the ckpt. overhead with erasure codes on large-scale GPU systems.*
- Scalable encoding algorithm and efficient group mapping **[CCGrid10]**
- Fast encoding work on idle resource (CPU/GPU) **[HiPC10]**

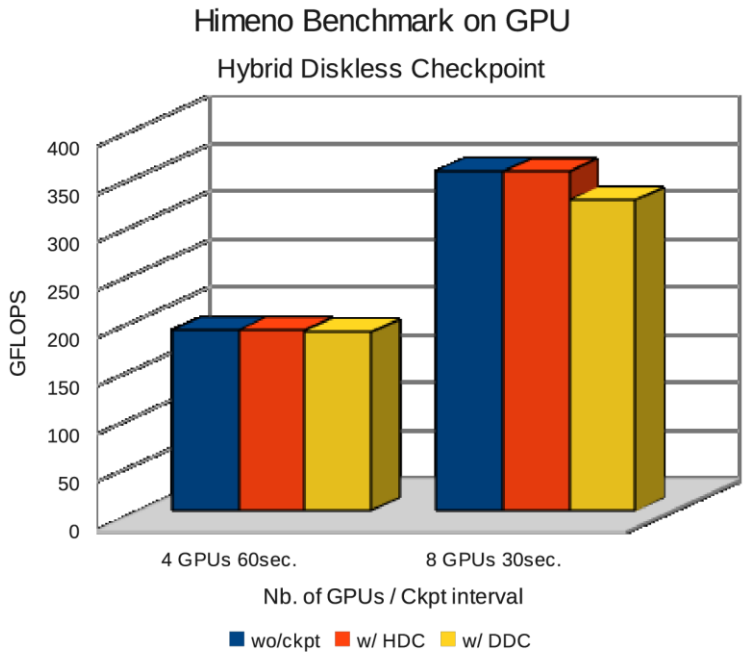




Hybrid Diskless Checkpoint

- Less than 3% of ckpt. overhead using idle resources





- We are currently combining this technique with our MPI CUDA Checkpoint.

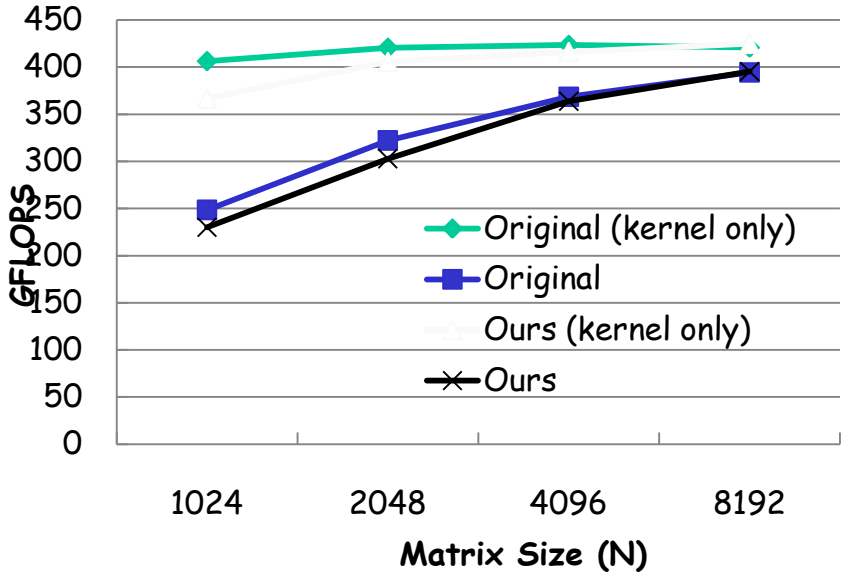
Software Framework for GPGPU Memory FT [IEEE IPDPS 2010]

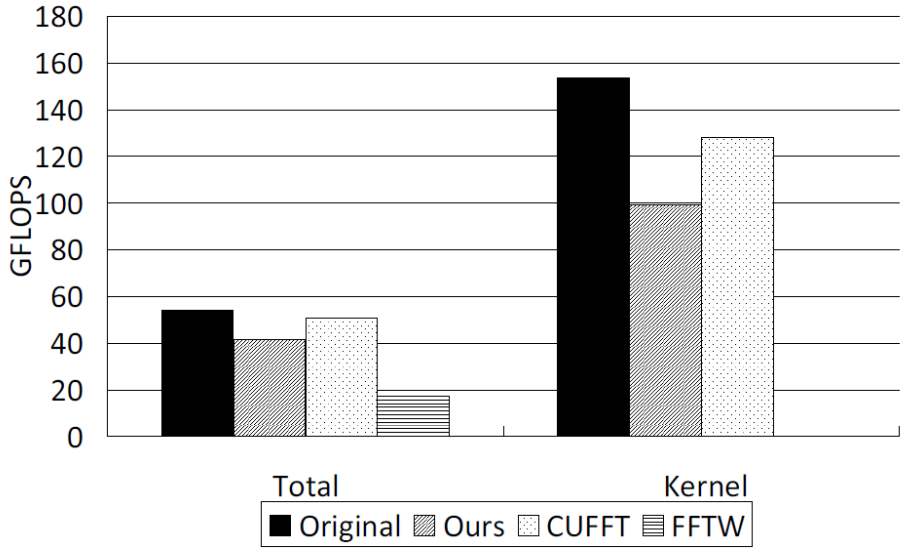
- Error detection in CUDA global memory + Checkpoint/Restart
- Works with existing NVIDIA CUDA GPUs

Lightweight Error Detection

- *Cross Parity* for 128B blocks of data
  - Detects a single-bit error in a 4B word
  - Detects a two-bit error in a 128B block
- No on-the-fly correction → Rollback upon error

Exploit the latency hiding capability of GPUs for data correctness

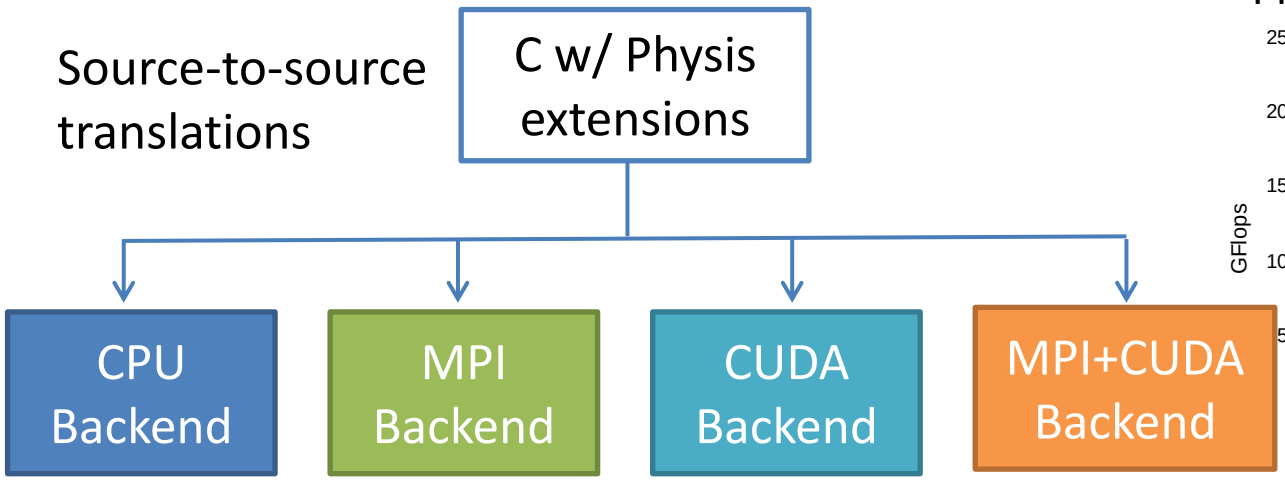




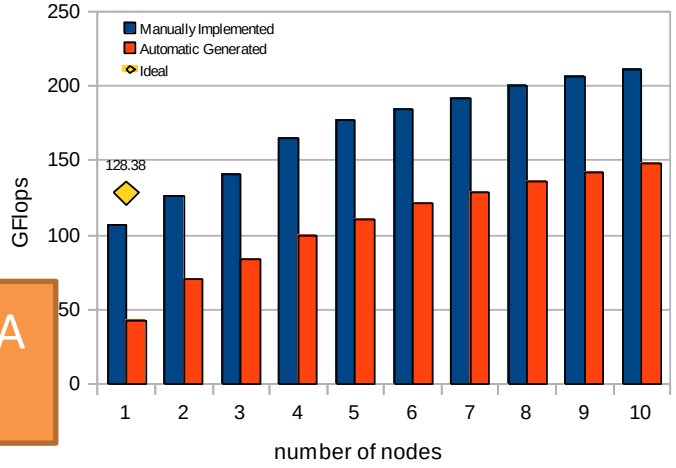
Tokyo Tech CCOE
4. GPU Higher-Level Programming Model

Physis: A Domain Specific Framework for Large-Scale GPU Clusters

- Portable programming environment for stencil computing on structured grids
  - Implicitly parallel
  - Distributed shared memory
- Provides concise, declarative abstractions for stencil computing



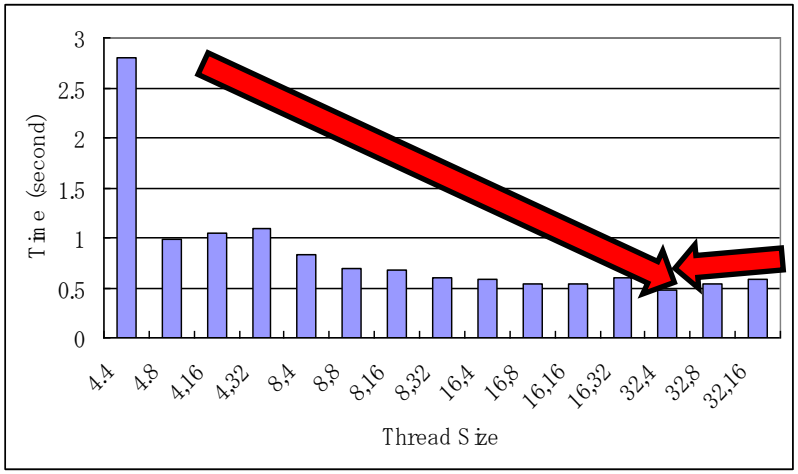
Preliminary Performance Results



Towards Auto-Tuning of Stencil Computation on GPU-accelerated Clusters

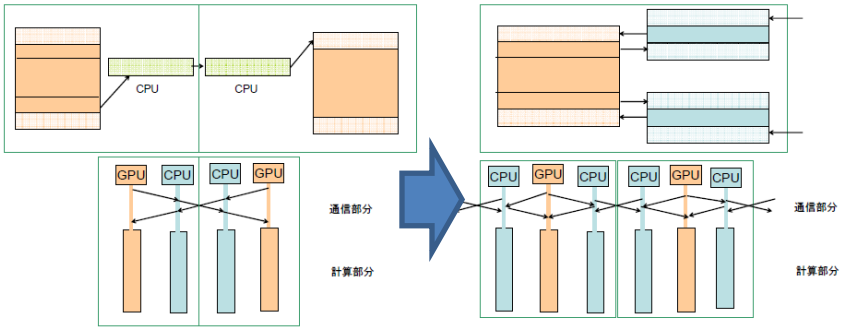
- Programming on GPU clusters gets harder because of
  - Hybrid programming with CUDA/MPI/threads
  - # of tuning parameters is increasing
  - Hiding latency

→ Auto-tuning is more important



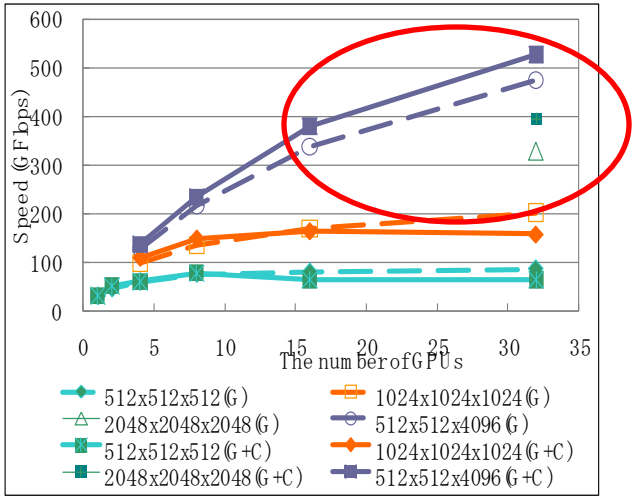
We find the optimal parameters, such as block size, by auto-tuning

→ x60 faster than 1 CPU core



- 3-HOP communication has been required, for GPU<->CPU<->CPU<->GPU

→ 1-HOP method is proposed, and auto-tune between communication methods



Up to 20% speed-up with 1-HOP method

526GFlops is achieved with 32GPUs on TSUBAME 1.2

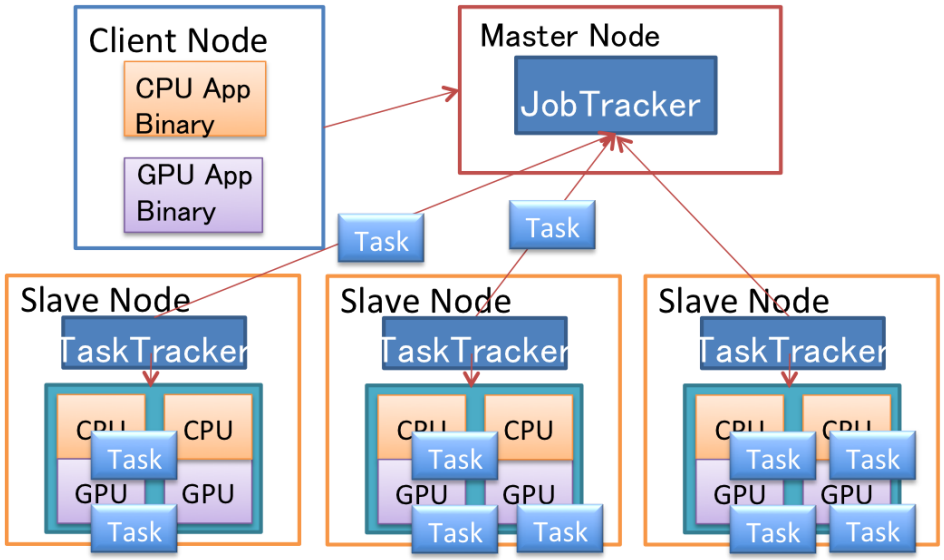
→ x1170 faster than 1 CPU core

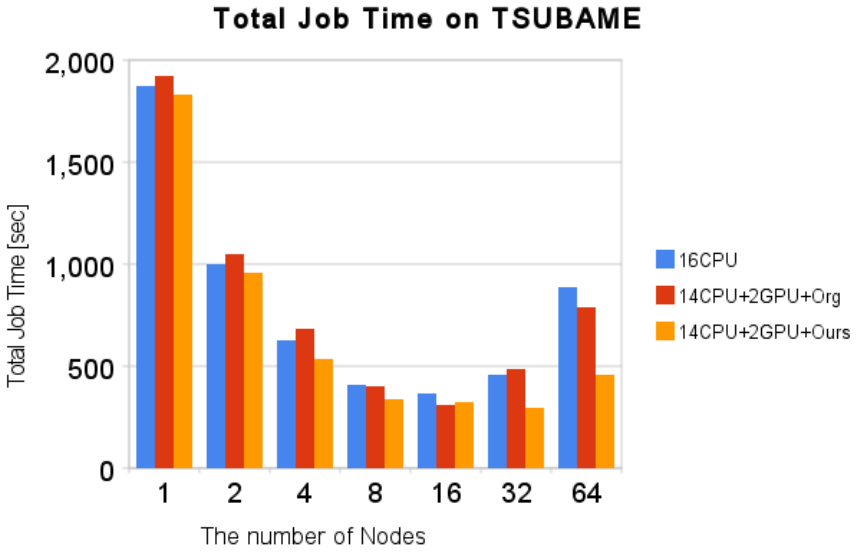
MapReduce for GPU-based Heterogeneous Clusters [MAPRED'2010]

Problem :

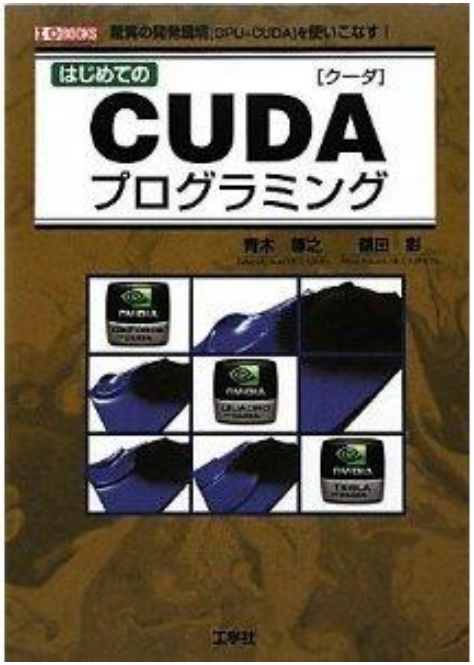
- Map Task Scheduling for efficient execution
  - Depends on **running task characteristics** and **underlying environments**

- Hybrid Map Task Scheduling
  - Automatically detects map task characteristics by monitoring
  - Scheduling map tasks to minimize overall MapReduce job execution time





1.93 times faster than the Hadoop original scheduling



Aoki and Nukada
"The First CUDA Programming" Textbook

Matsuoka, Endo, Nukada, Aoki et. al.
Special Issue IPSJ Magazine 2009,
"Accelerators, Reborn"

TSUBAME E-Science Journal
(Vol.2 at SC10)

GPU Computing Consortium

Established Sep. 1, 2009 in GSIC
607 Members as of Sep. 2010
8 CUDA Lectures (Hands-on)
1 International Workshop
1 Symposium (Oct., 2010)
3 Seminars by World-famous