



Large-scale CFD Applications (Weather Prediction) on TSUBAME 2

Takayuki Aoki

***Global Scientific Information and Computing Center
Tokyo Institute of Technology***

Supercomputer in the world



2010 November

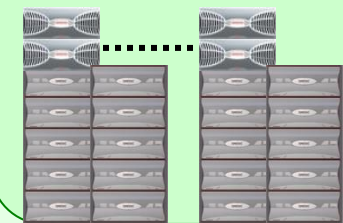
Rank	Site	Computer/Year Vendor	Cores	R _{max}	R _{peak}	Power
1	National Supercomputing Center in Tianjin China	Tianhe-1A - NUDT YH Cluster, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C / 2010 NUDT	186368	2566.00	4701.00	4040.00
2	DOE/SC/Oak Ridge National Laboratory United States	Jaguar - Cray XT5-HE Opteron 6-core 2.6 GHz / 2009 Cray Inc.	224162	1759.00	2331.00	6950.60
3	National Supercomputing Centre in Shenzhen (NSCS) China	Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU / 2010 Dawning	120640	1271.00	2984.30	2580.00
4	GSIC Center, Tokyo Institute of Technology Japan	TSUBAME 2.0 - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows / 2010 NEC/HP	73278	1192.00	2287.63	1398.61
5	DOE/SC/LBNL/NERSC United States	Hopper - Cray XE6 12-core 2.1 GHz / 2010 Cray Inc.	153408	1054.00	1288.63	2910.00

SUBAME2.0 System Overview (2.4 Pflops/15PB)

Petascale Storage : Total **7.13PB**(Lustre + Accelerated NFS Home)

Lustre Partition **5.93PB**

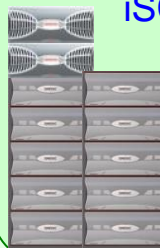
x5



MDS,OSS
 HP DL360 G6 30nodes
 Storage
 DDN SFA10000x5
 (10 enclosures x5)
 Lustre (5 Filesystems)
 OSS: 20 OST: 5.9PB
 MDS: 10 MDT: 30TB

OSS x20 MDS x10

Home NFS/
iSCSI



Storage Server
 HP DL380 G6 4nodes
 BlueArc Mercury 100 x2
 Storage
 DDN SFA10000 x1
 (10 enclosures x1)

NFS,CIFS x4 NFS,CIFS,iSCSI
 Accelerationx2

Tape System
 Sun SL8500 8PB

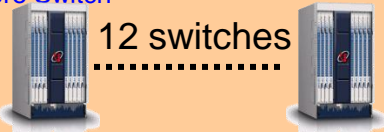
SuperTitenet

SuperSinet3

Node Interconnect: **Optical, Full Bisection, Non-Blocking, Dual-Rail QDR**

Core Switch

12 switches



Voltaire Grid Director 4700
 IB QDR: 324ports

Edge Switch

179 switches



Voltaire Grid Director 4036
 IB QDR : 36 ports

Edge Switch(w/10GbE)

6 switches



Voltaire Grid Director 4036E
 IB QDR:34ports
 10GbE: 2ports

Mgmt Servers

Compute Nodes : **2.4PFlops**(CPU+GPU)

1.69TFlops(CPU)

"Thin" Nodes



1408nodes
(32node x44 Racks)

NEW DESIGN Hewlett Packard CPU+GPU
 High BW Compute Nodes x 1408
 Intel Westmere-EP 2.93GHz
 (TB 3.196GHz) 12Cores/node
 Mem:55.8GB(=52GiB)or 103GB(=96GiB)
 GPU NVIDIA M2050 515GFlops,
 3GPUs/node
 SSD 60GB x 2 120GB *55.8GB node
 120GB x 2 240GB *103GB node
 OS: Suse Linux Enterprise + Windows

HPC
4224 NVIDIA "Fermi" GPUs
 Memory Total : 80.55TB
 SSD Total : 173.88TB

"Medium" Nodes



24 nodes

HP 4Socket Server
 24nodes
 CPU Nehalem-EX 2.0GHz
 32Cores/node
 Mem:137GB(=128GiB)
 SSD 120GB x 4 480GB
 OS: Suse Linux Enterprise

6.14TFLOPS

"Fat" Nodes



10 nodes

HP 4Socket Server 10nodes
 CPU Nehalem-EX 2.0GHz
 32Core/node
 Mem:274GB(=256GiB)x8
 549GB(=512GiB) x2
 SSD 120GB x 4 480GB
 OS: Suse Linux Enterprise

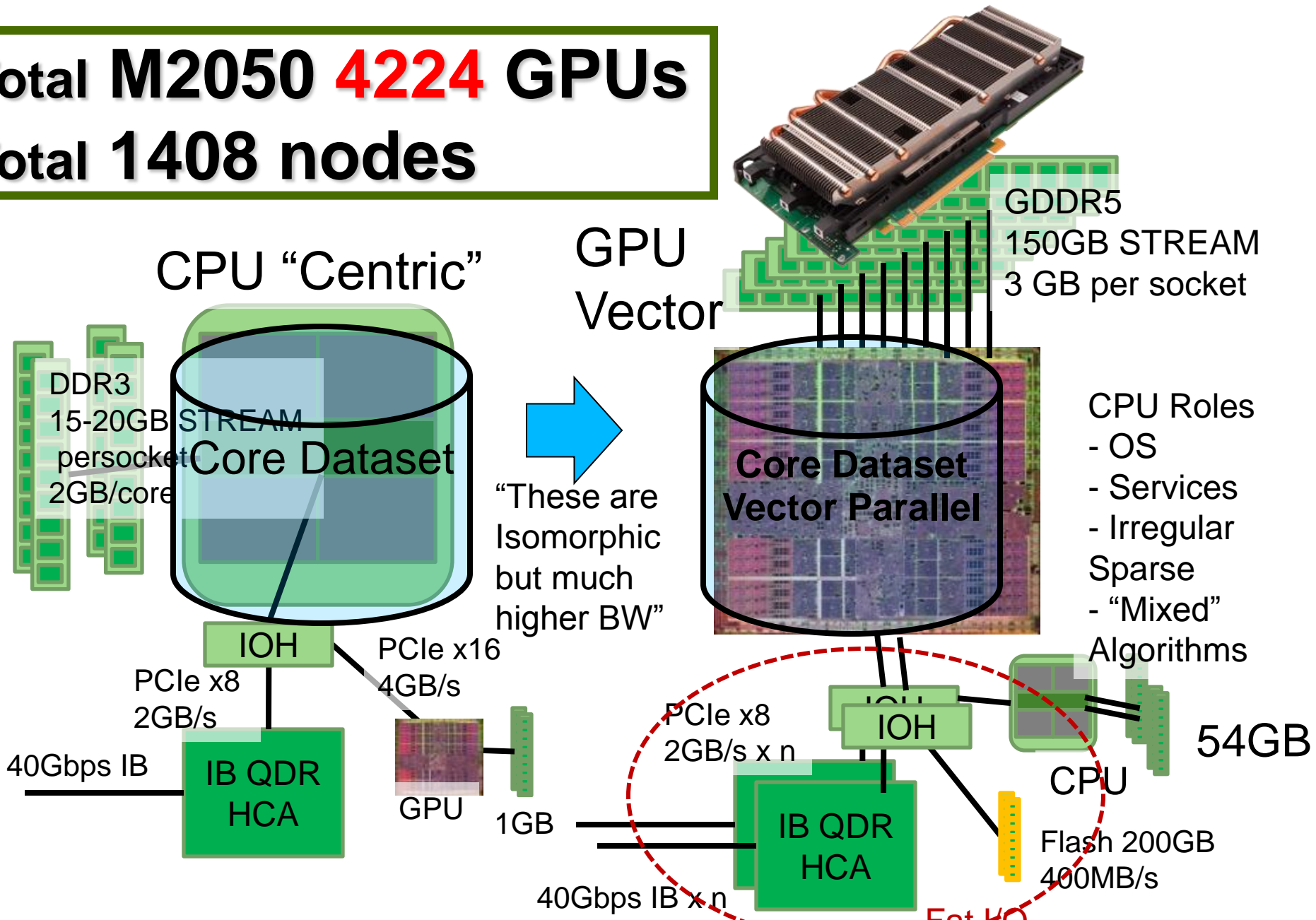
2.56TFLOPS

PCI-E gen2 x16 x2slot/node

GSIC:NVIDIA Tesla S1070GPU (34 units)

TSUBAME 2.0: GPU Centric Nodes

Total M2050 **4224** GPUs
Total 1408 nodes





GPU Applications of our group

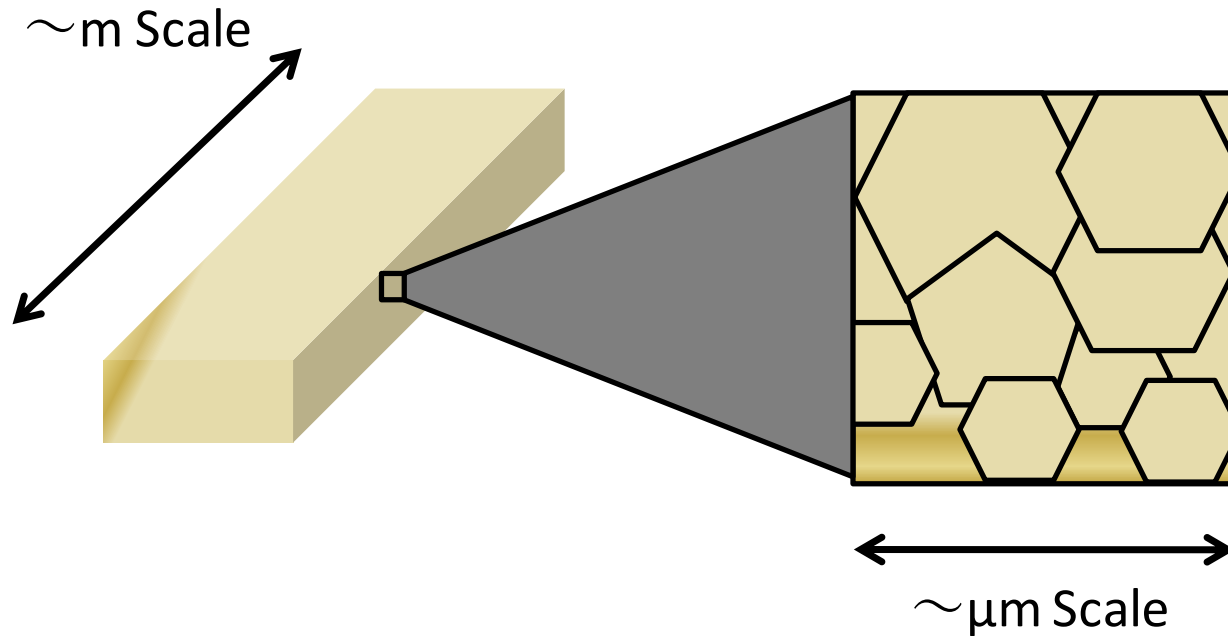
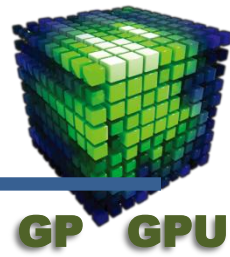
- Higher-order Compressible Flow
- Lattice Boltzmann for Pulmonary Airflow
- FDTD for Electromagnetic wave Propagation
- Large-Eddy Simulation for Turbulence Flow
- Real-time TSUNAMI Simulation
- Dendrite Solidification based on Phase Field Model
- Numerical Weather Prediction
- Two-Phase Flow Simulations



GPU Applications of our group

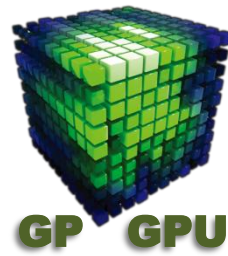
- Higher-order Compressible Flow
- Lattice Boltzmann for Pulmonary Airflow
- FDTD for Electromagnetic wave Propagation
- Large-Eddy Simulation for Turbulence Flow
- Real-time TSUNAMI Simulation
- Dendrite Solidification based on Phase Field Model
- Numerical Weather Prediction
- Two-Phase Flow Simulations

Material Compound



Meso-scale Analysis for Solidification Process

Phase Field Model

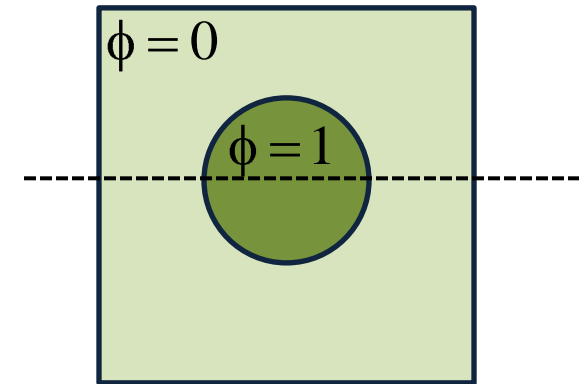


- Non-equilibrium Statistical Physics

- Phase Field Model ϕ

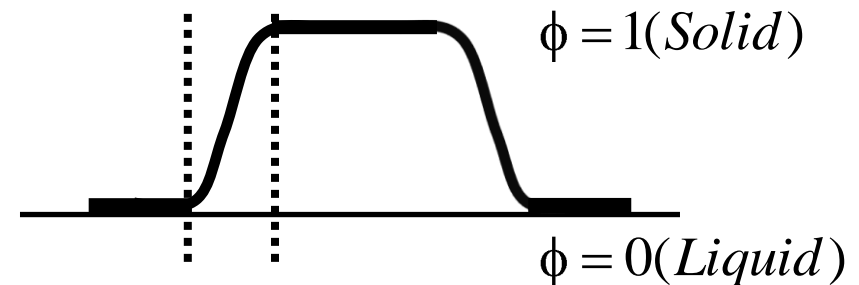
- Introduction of index function

$\phi = 0$	$\phi = 1$
liquid	solid



- diffusive surface

- stabilization of surface energy



- High computational cost of non-linear term



Phase Field Equation

Allen-Cahn Equation

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

$$\beta = -\frac{15L}{2W} \frac{T - T_m}{T_m} \phi \left(-\phi \right) \quad \varepsilon = \bar{\varepsilon} \left(1 - 3\gamma + 4\gamma \frac{\phi_x^4 + \phi_y^4 + \phi_z^4}{|\nabla \phi|^4} \right)$$

Thermal Conduction

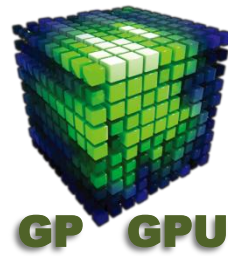
$$\frac{\partial T}{\partial t} = k \nabla^2 T + \frac{L}{C} 30\phi^2 \left(-\phi \right) \frac{\partial \phi}{\partial t}$$

↑ Introduction of non-isotropic surface energy

Phase Field	$0 < \phi < 1$
Liquid	$\phi = 0$
Solid	$\phi = 1$

Second-order Finite Difference Method and 1st-order Euler Time Integration

Numerical Stencil Access

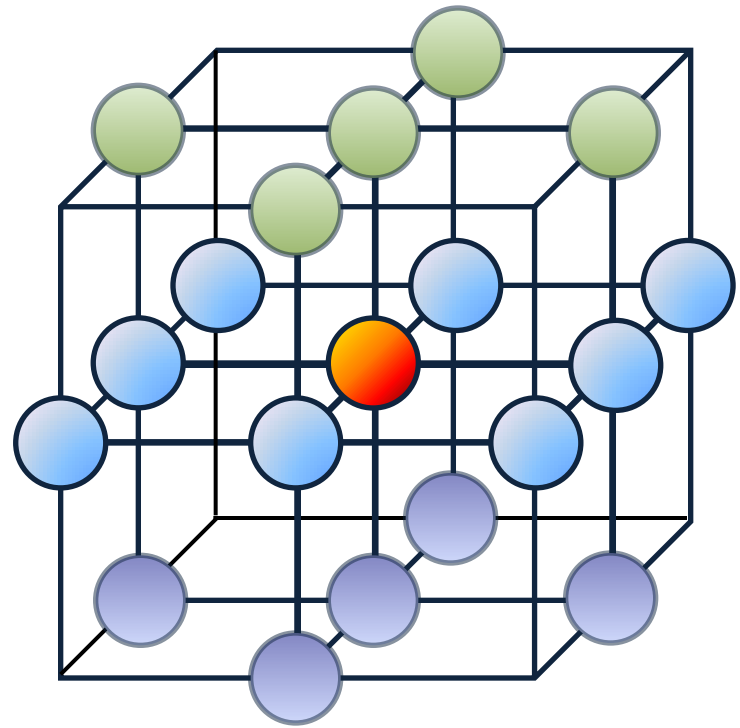


Phase Field ϕ

Temperature T

19 points to solve $\phi_{i,j,k}$

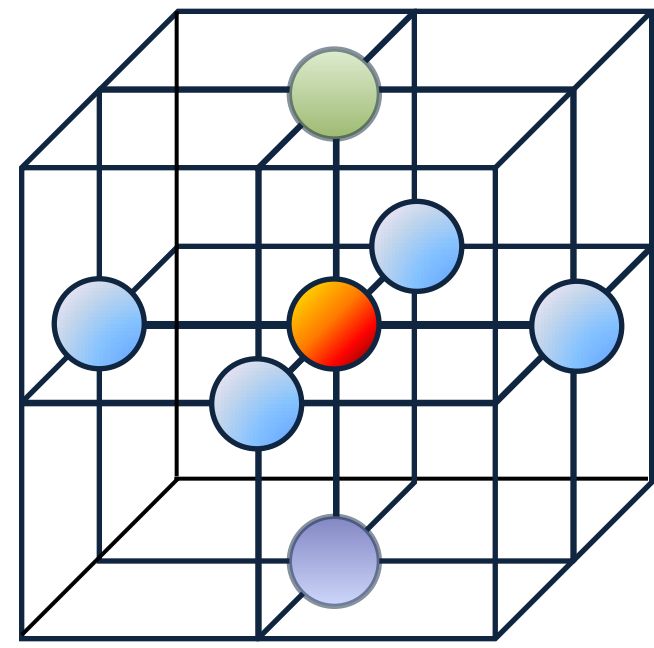
7 points to solve $T_{i,j,k}$



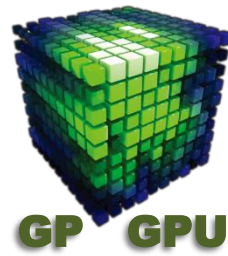
$z = k - 1$

$z = k$

$z = k + 1$



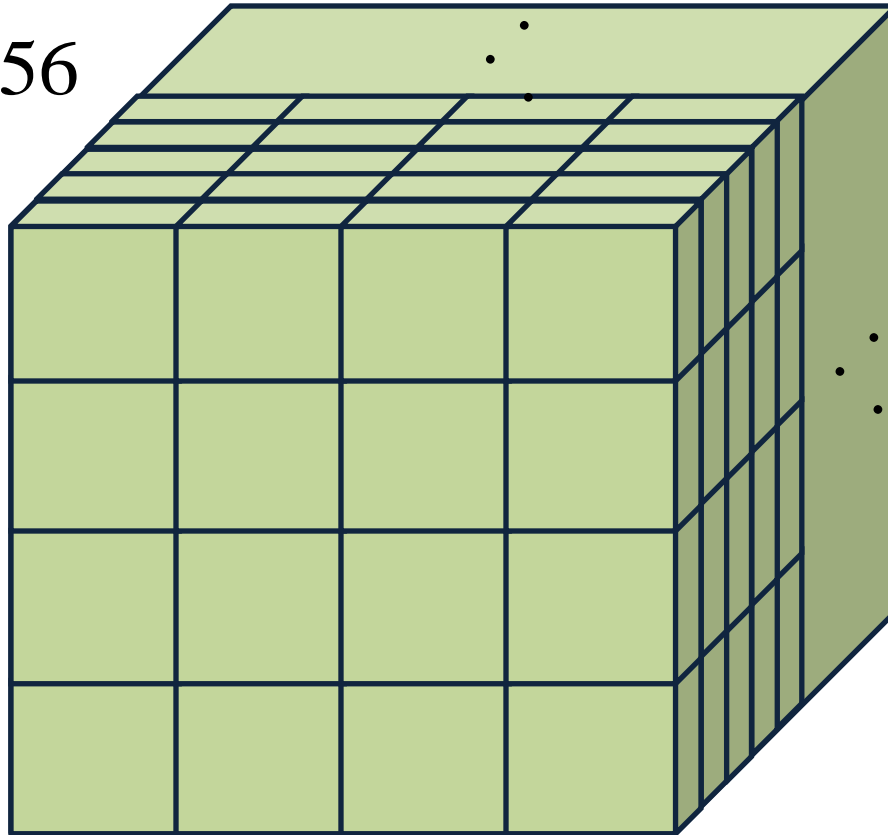
Thread Assignment



$$nx = 256$$

$$ny = 256$$

$$nz = 256$$



$$blockDimx = 64$$

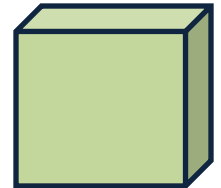
$$blockDim.y = 4$$

$$blockDimz = 1$$



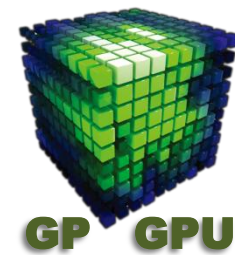
$$ny = 4 \quad nx = 64$$

$$nz = 64$$



64 threads in the x-direction for Coalesced memory access

Sweep of 1 thread in a block

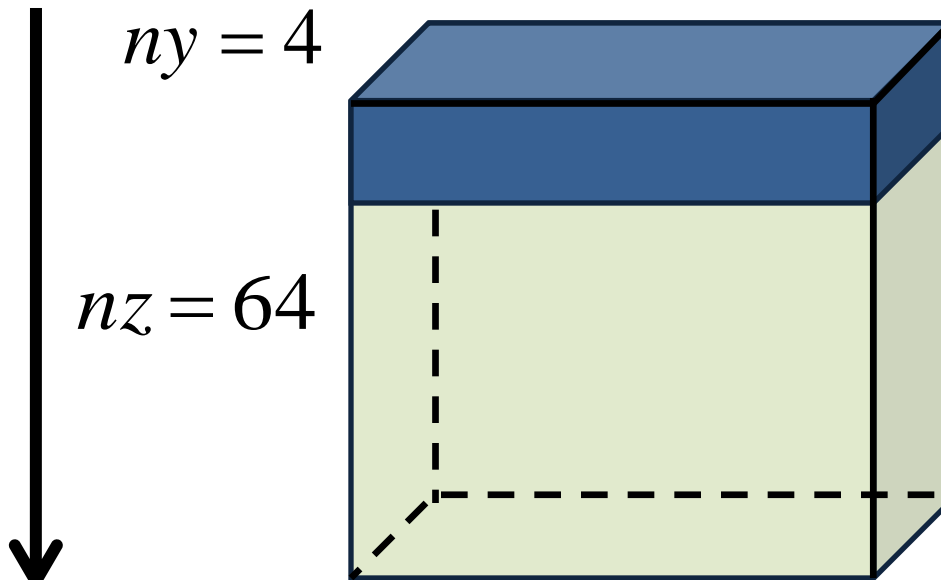


$nx = 64$

$ny = 4$

$nz = 64$

Z-axis



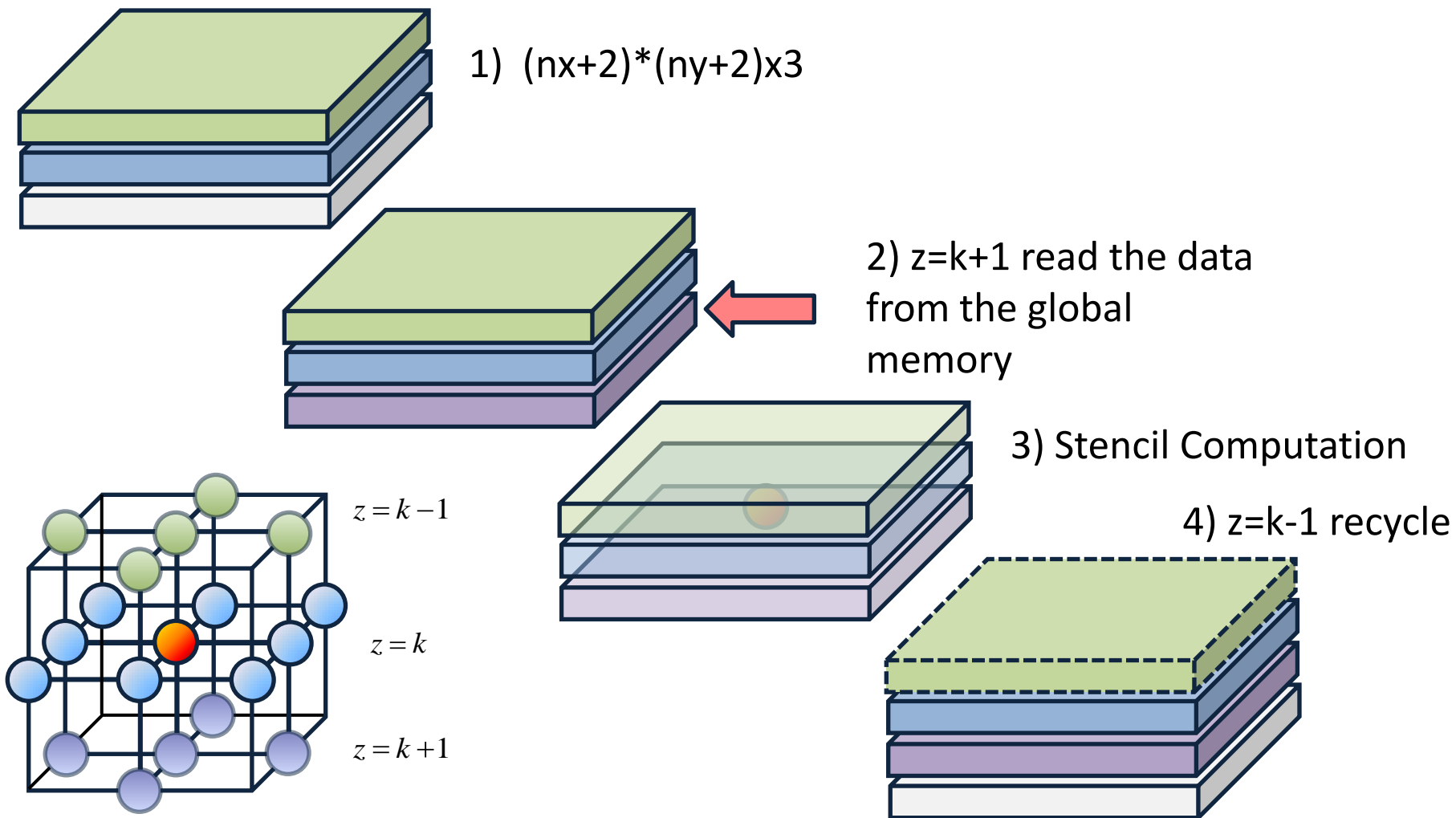
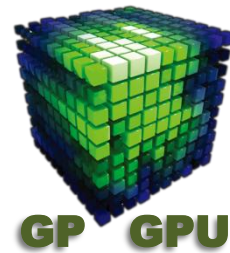
$threadDimx = 64$

$threadDim.y = 4$

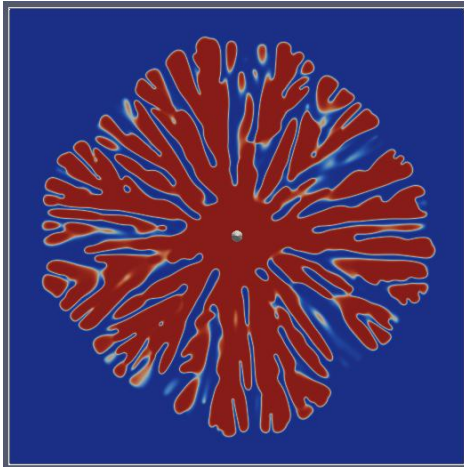
$threadDimz = 1$

Marching in the z-directional direction

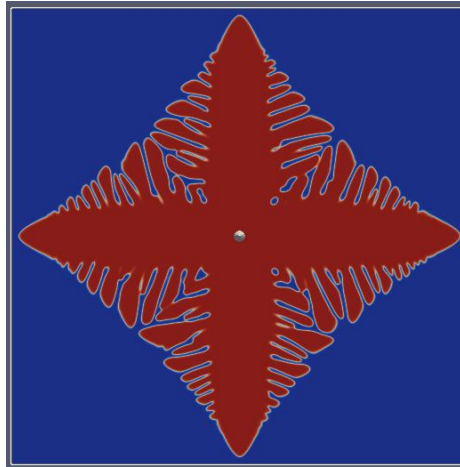
Recycle of Shared Memory



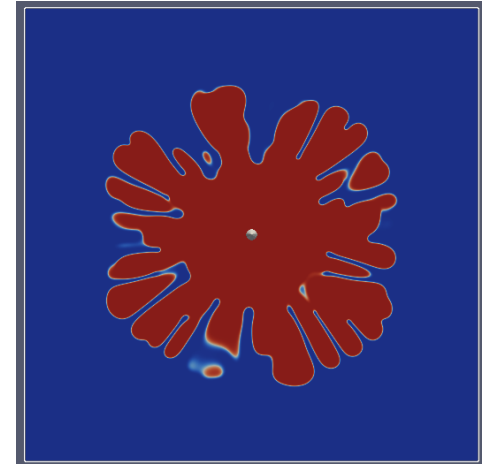
Dependence on Parameters



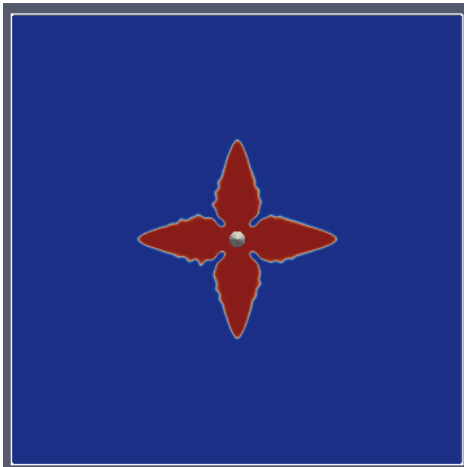
$\gamma = 0.015, A=0.01$



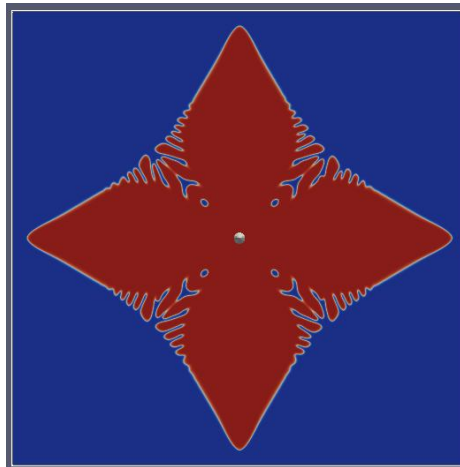
$\gamma = 0.075, A=0.01$



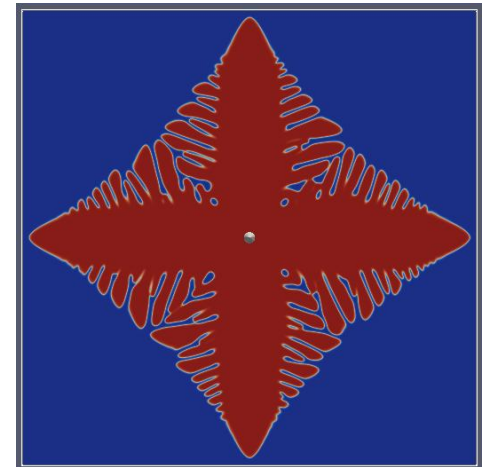
$\gamma = 0.015, A=0.01, \Delta t=\text{half}$



$\gamma = 0.1, A=0.01$



$\gamma = 0.075, A=0.01$



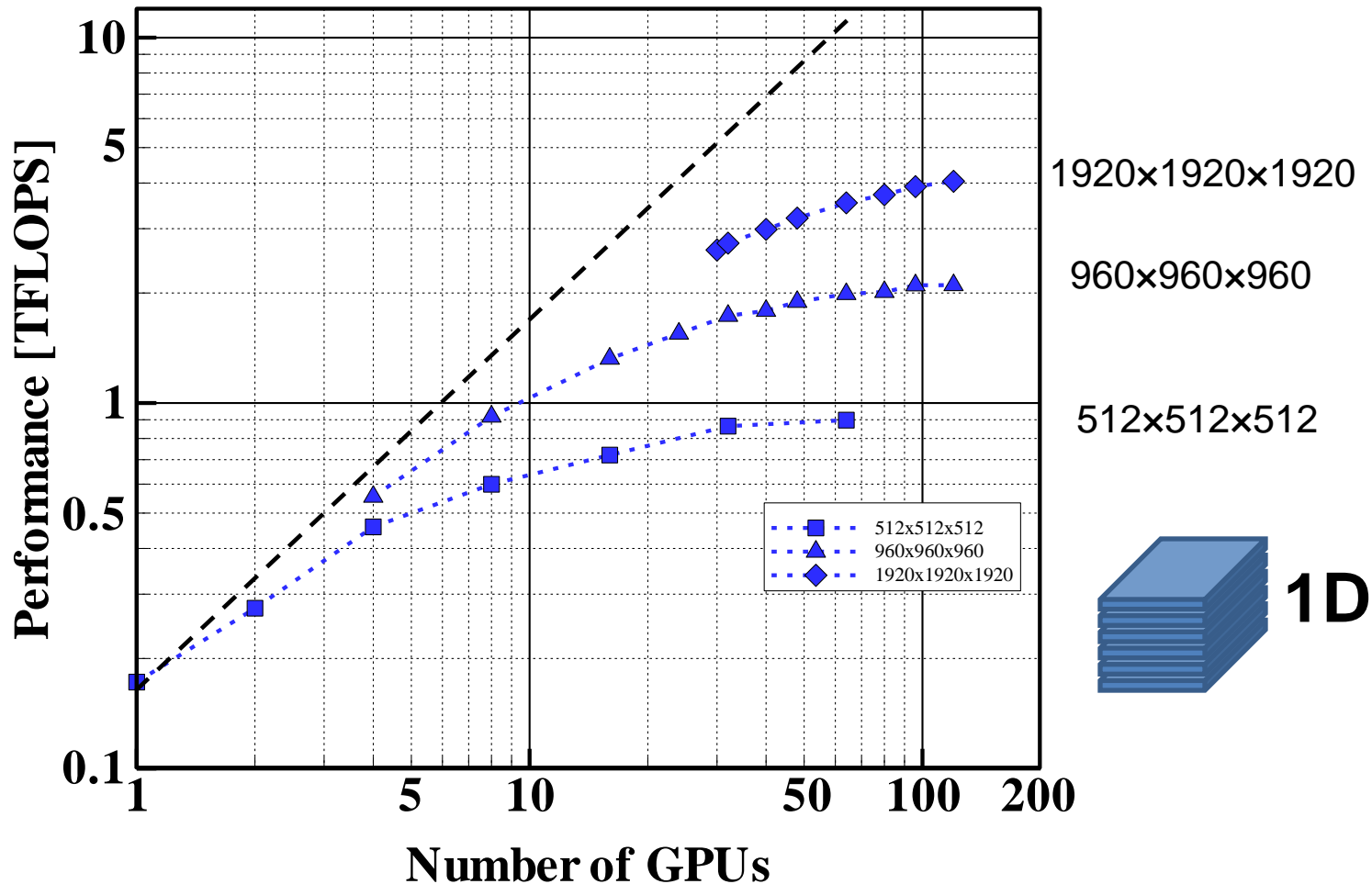
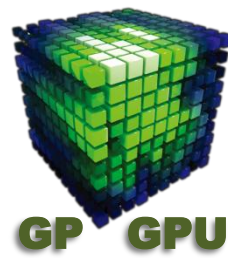
$\gamma = 0.075, A=0.005$

Dendrite Solidification for Pure Metal

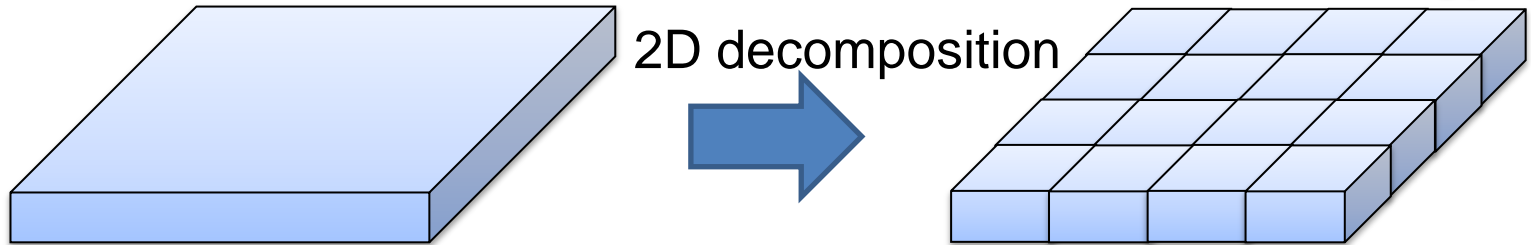
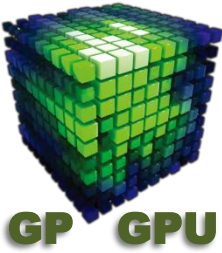
The background of the slide is a grayscale micrograph of a metal surface. It displays a complex, textured morphology characteristic of dendritic solidification. The surface is covered with a dense network of interconnected, branching structures that resemble a tree or coral. These structures are darker in color, contrasting with the lighter, smoother background. The overall appearance is highly irregular and porous, typical of a solidified metal surface under non-equilibrium conditions.

Multi-GPU Performance

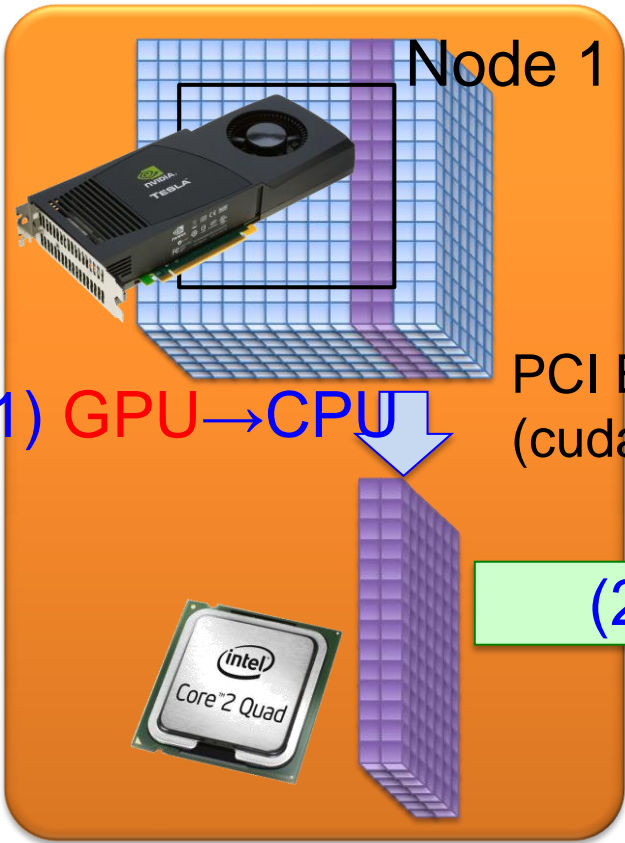
w/o overlapping



Multi-GPU : Domain decomposition



GPU



Node 1

PCI Express
(cudaMemcpy)

CPU

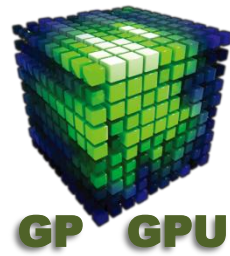
MPI

Node 2

(3) CPU -> GPU

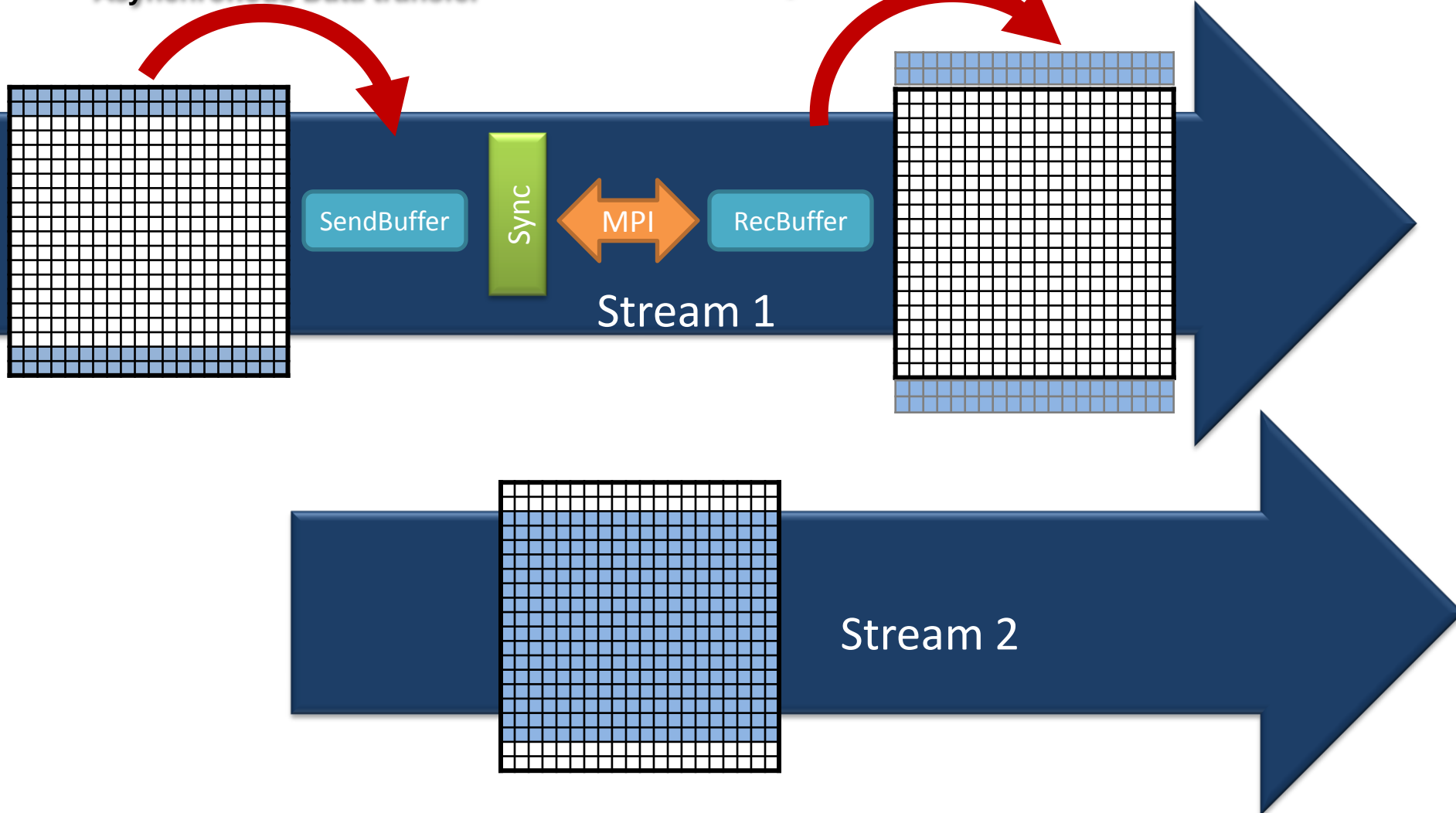


Overlapping between Computation and Communication



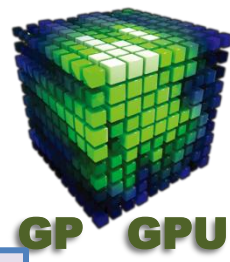
Asynchronous Data transfer

Asynchronous Data transfer



Multi-GPU Performance

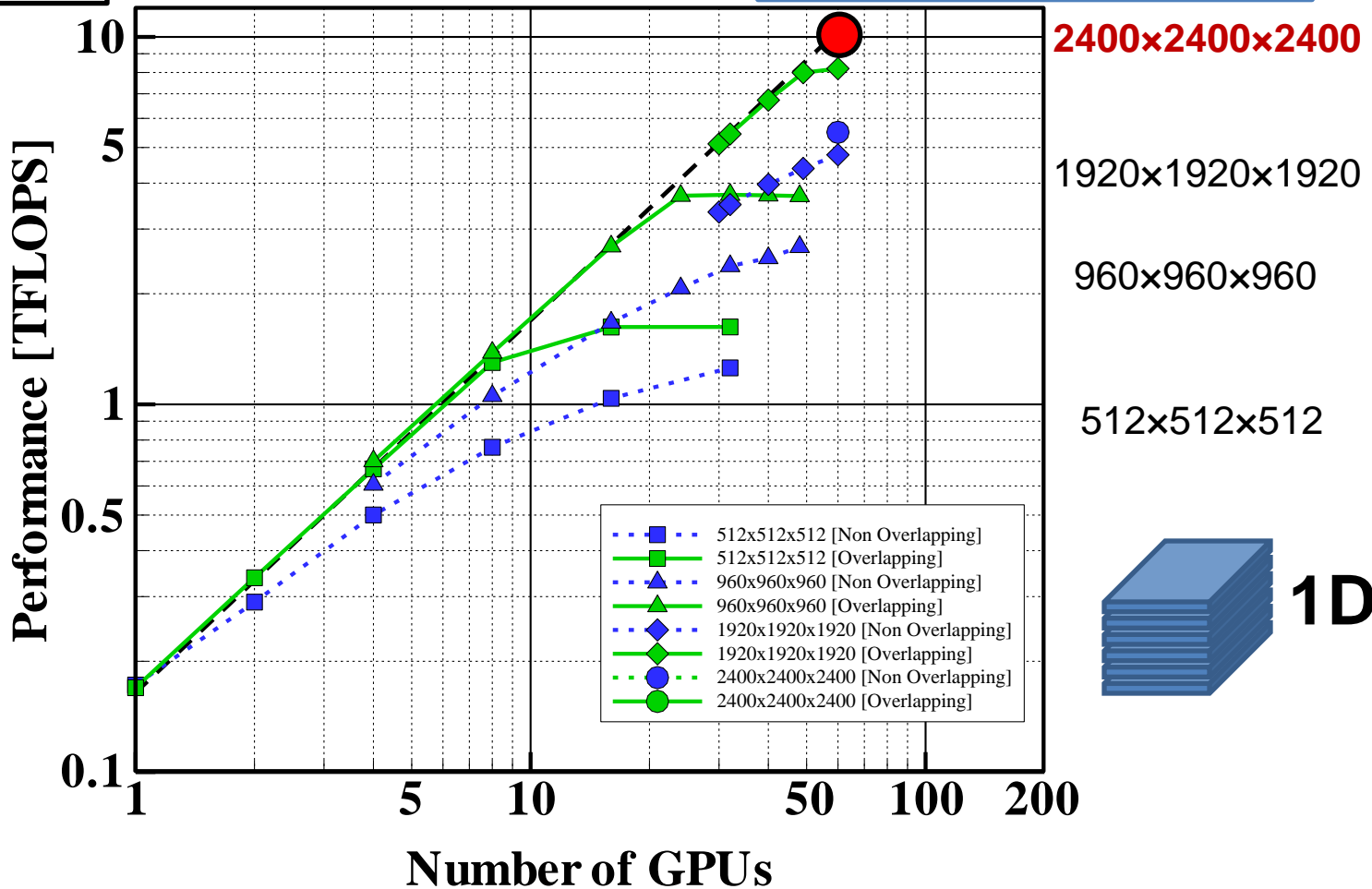
w/o overlapping



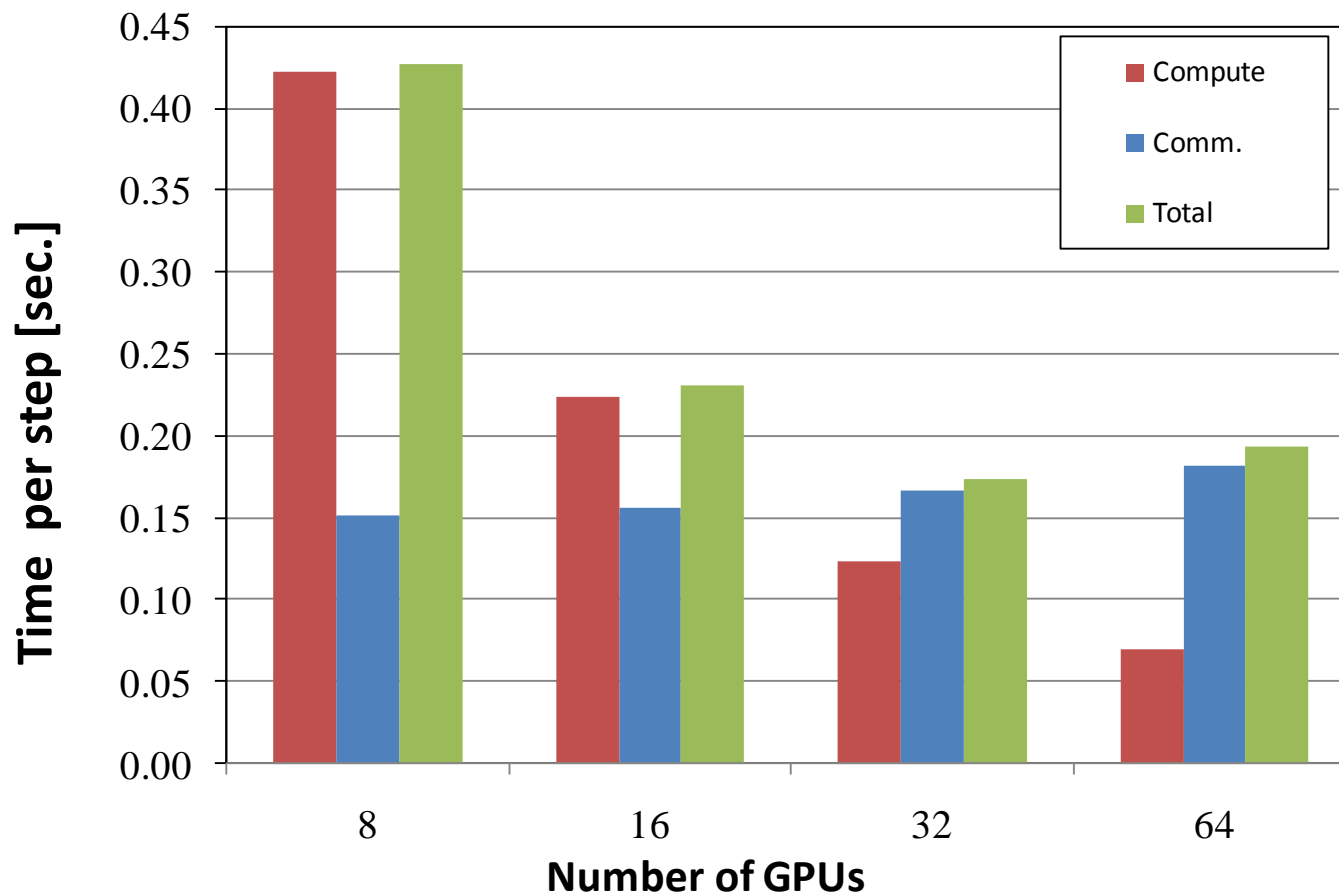
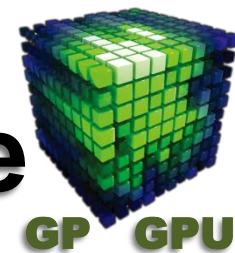
GP GPU

1 GPU/node

10TFLOPS : 60GPU



Breakdown of Overlapping Case

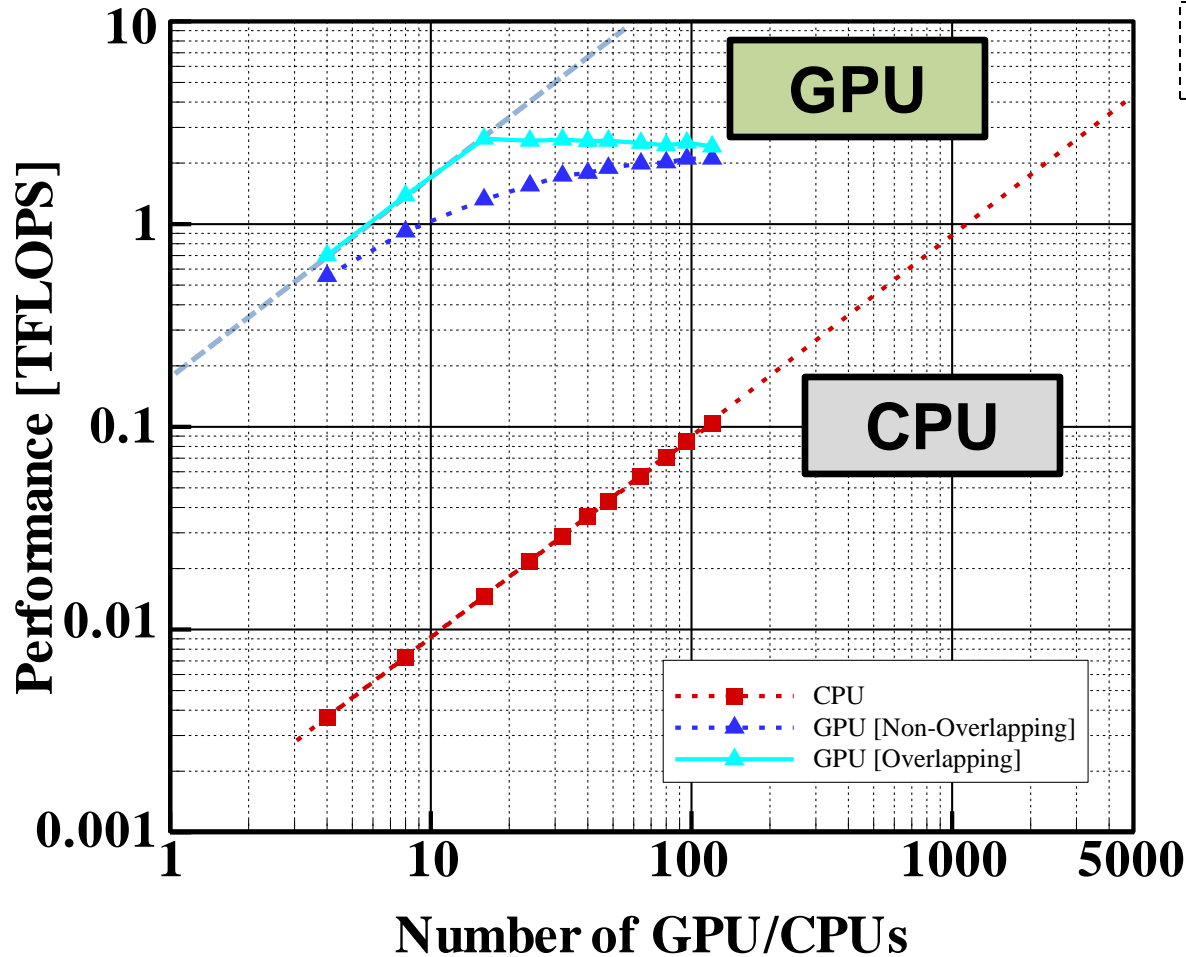
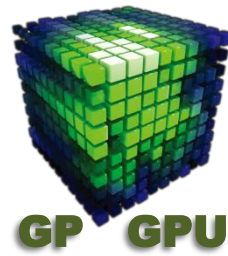


Grid Number:
1024x1024x1024

Machine:
TSUBAME
Tesla S1070
Infiniband
(hpc1tes2 Queue)

- Computational time becomes short for more GPU numbers
- Communication time is almost same
- The communication time can not be hidden for more than 32 GPUs

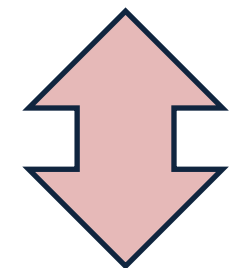
Direct Comparison : CPU/GPU



960³ mesh

Tesla S1070

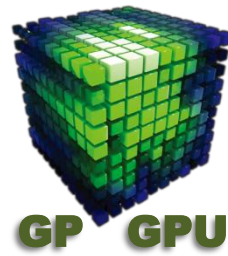
60 GPU



10000 CPU
core

TSUBAME 1.0

Weather Prediction



Collaboration: Japan Meteorological Agency

Meso-scale Atmosphere Model:

Cloud Resolving Non-hydrostatic model

Compressible equation taking consideration of sound waves.

Meso-scale

2000 km

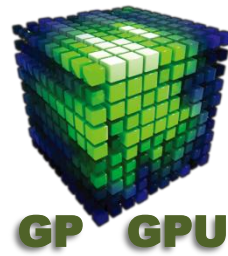
Typhoon

a few km

Tornado, Down burst
Heavy Rain



Atmosphere Model



Dynamical Process:

Full 3-D Navior-Stokes Equation

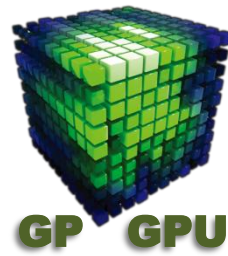
$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P - 2\boldsymbol{\Omega} \times \mathbf{u} - \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}) + \mathbf{g} + \mathbf{F}$$

Physical Process:

Cloud Physics, Moist, Solar Radiation, Condensation,
Latent heat release, Chemical Process, Boundary Layer

So called “Parameterization” including many empirical rules.

WRF GPU Computing



■ WRF (Weather Research and Forecast)

Community Code developed by NCAR, NCEP, OU, NOAA/FSL, AFWA

WSM5 (WRF Single Moment 5-tracer) Microphysics*

Represents condensation, precipitation and thermodynamic effects of latent heat release

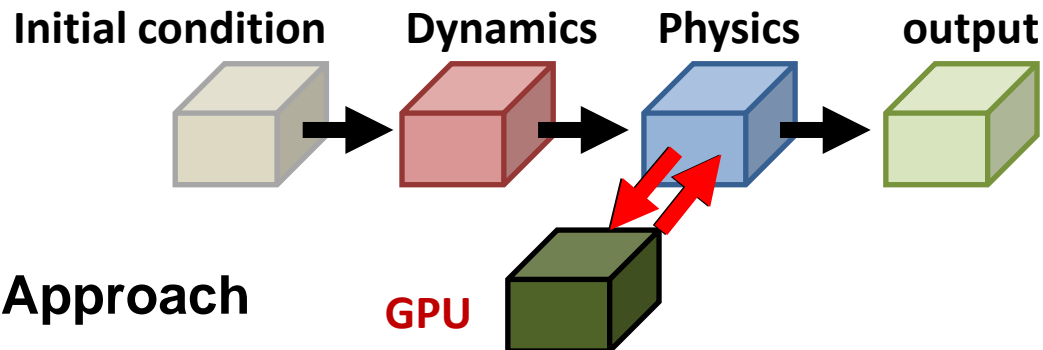
1 % of lines of code, 25 % of elapsed time

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

WRF-Chem**

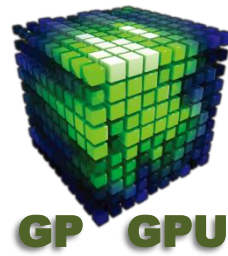
provides the capability to simulate chemistry and aerosols from cloud scales to regional

⇒ x 8.5 increase



Accelerator Approach

Full GPU Implementation

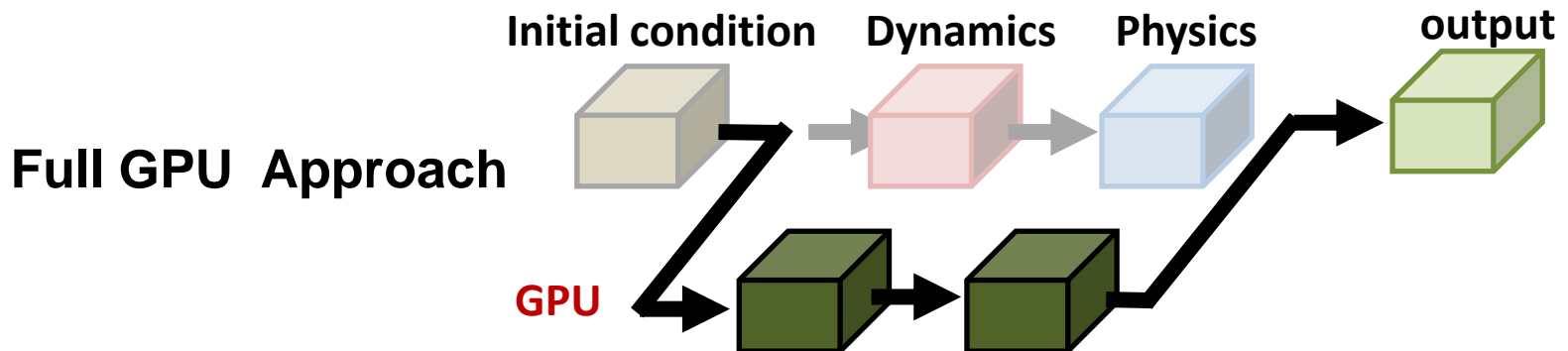


■ ASUCA Production Code

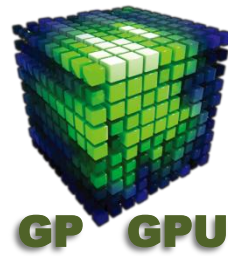
- ✓ A next-generation high resolution weather simulation code that is being developed by Japan Meteorological Agency (JMA)
- ✓ ASUCA succeeds the JMA-NHM as an operational non-hydrostatic regional model at JMA

■ Similar Structure as WRF

- ✓ HEVI (Horizontally explicit Vertical implicit) scheme
- ✓ Dynamical Core uses a numerical scheme with 3rd-order accuracy in time and space
 - Flux-form non-hydrostatic compressible equation
 - Generalized coordinate



Entire Porting Fortran to CUDA



■ Rewrite from Scratch

```
Program init
implicit none

integer i
integer a(10)

do i = 1, 10
  a(i) = i
end do

end program init
```

Fortran

✓ **Original code at JMA**

```
#include <iostream>

int main()
{
  int i;
  int a[10];
  for(i=0;i<10;i++){
    a[i] = i + 1;
  }
}
```

C/C++

✓ **Changing array order**

```
#include <cuda.h>

__global__ void init(int *a){
  a[threadIdx.x] =
  threadIdx.x+1;
}

int main()
{
  int i;
  int *a;
  cudaMalloc(&a,sizeof(int)*
  10);
  init(1,10)>>>(a);
  cudaFree(a);
}
```

CUDA

✓ **GPU code**

z,x,y (k,i,j)-ordering

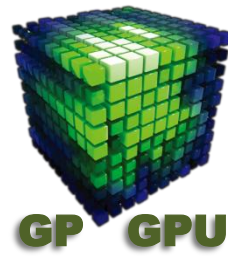
x,z,y (i,k,j)-ordering

x,z,y (i,k,j)-ordering

■ 1 Year by a Ph.D student

Introducing many optimizations, overlapping the computation with the communication, kernel fuse, reordering kernel execution

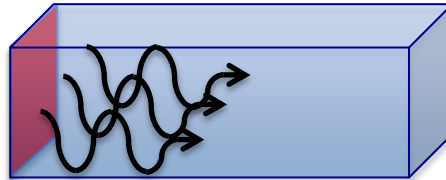
Implementation : Advection



Thread

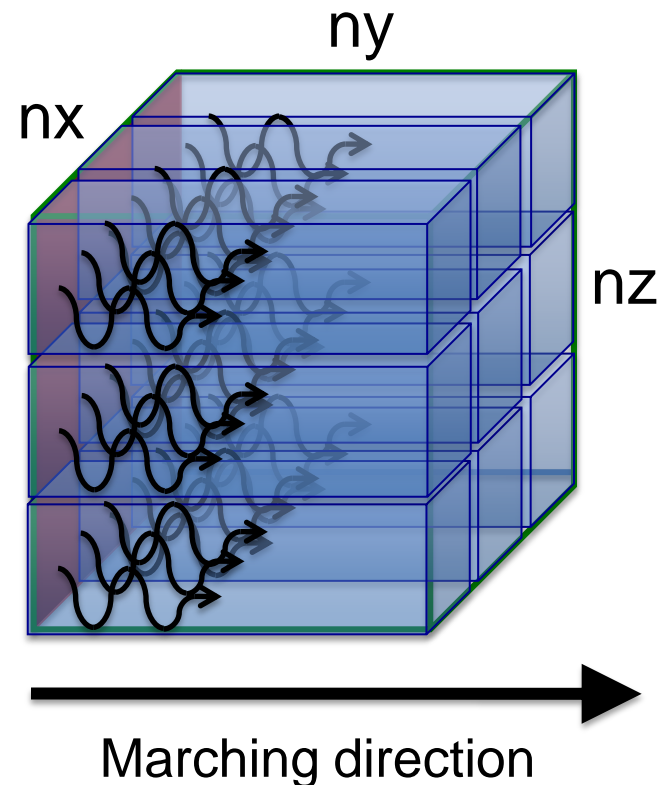


Block



64 x 4 threads (2D) in a block

- Each thread specifies a (x, z) point, marching in y
 - ✓ Improve data transfer performance using domain decomposition



Using Shared Memory



- Shared Memory (SMem) = Software Managed Cache
- Read a 2D sub-domain from VRAM into SMem
- Advection : 12-point stencil
 - ✓ Store the xz-slice in $(64 + 3) \times (4 + 3)$ SMem

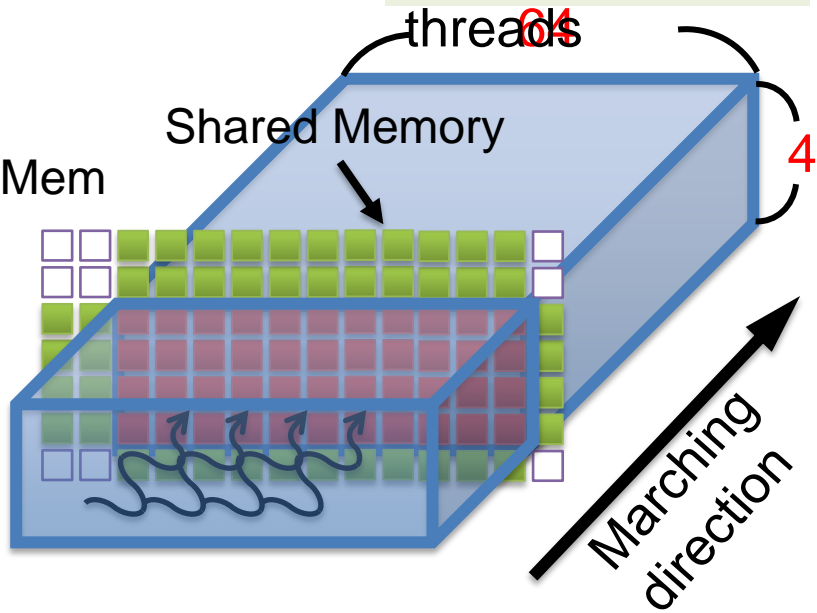
1 Block
= 64 x 4
threads

Access GMem directly : 4 + 4 read,
1write



Using SMem : ~1

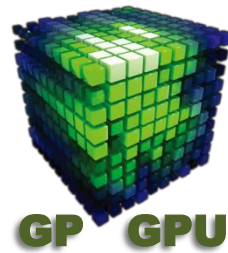
read, 1write



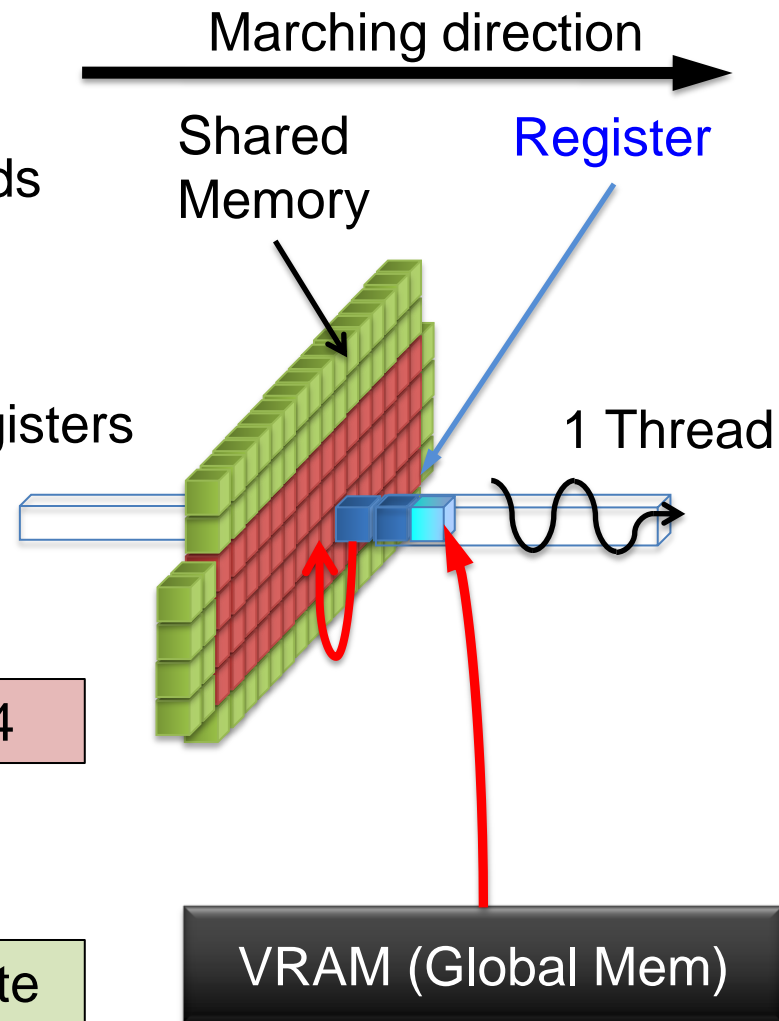
- 2D sub-domain
- Halo
- Not in use

	Shared Memory	VRAM (Global Memory)
Access speed	~ 2 cycle	400-600 cycle
Capacity	16 kByte/Block	2 GByte (Total)

Using Registers in marching direction



- Register
 - ✓ Access speed : 1 cycle
 - ✓ used for data not shared among threads
- Advection : 12-point stencil
 - ✓ Each thread keeps 4 y-elements in registers
 - ✓ Elements are reuse

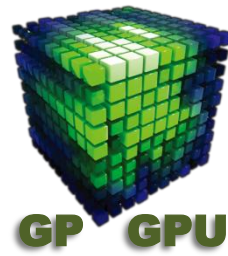


Access GMem directly : 4 + 4 + 4
read, 1write



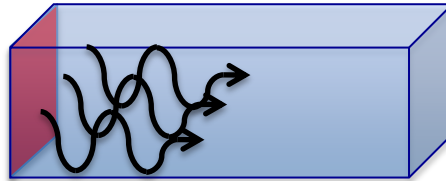
Using SMem and Registers : ~1 read, 1write

Implementation : 1D Helmholtz equation



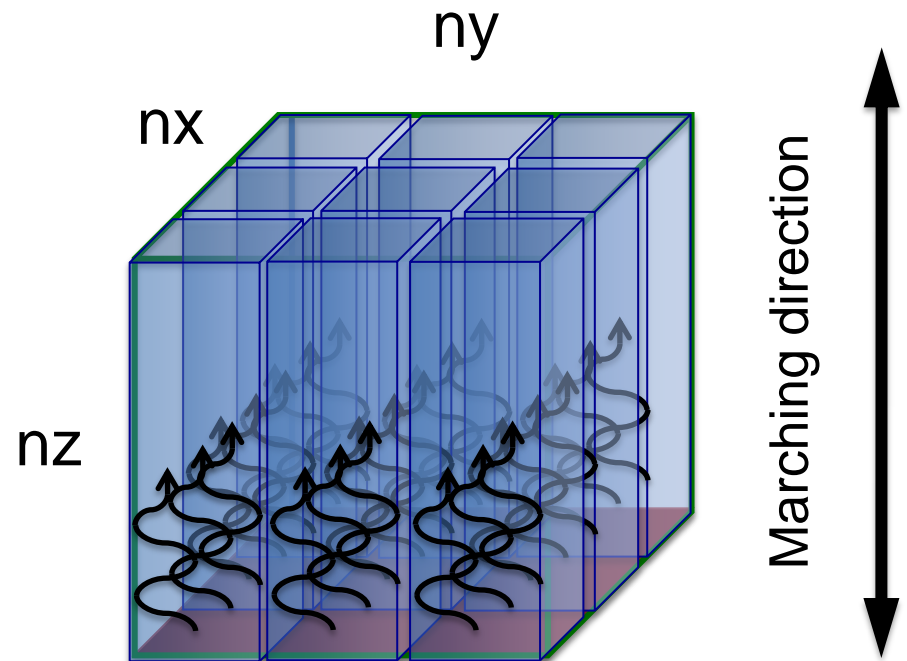
Thread 

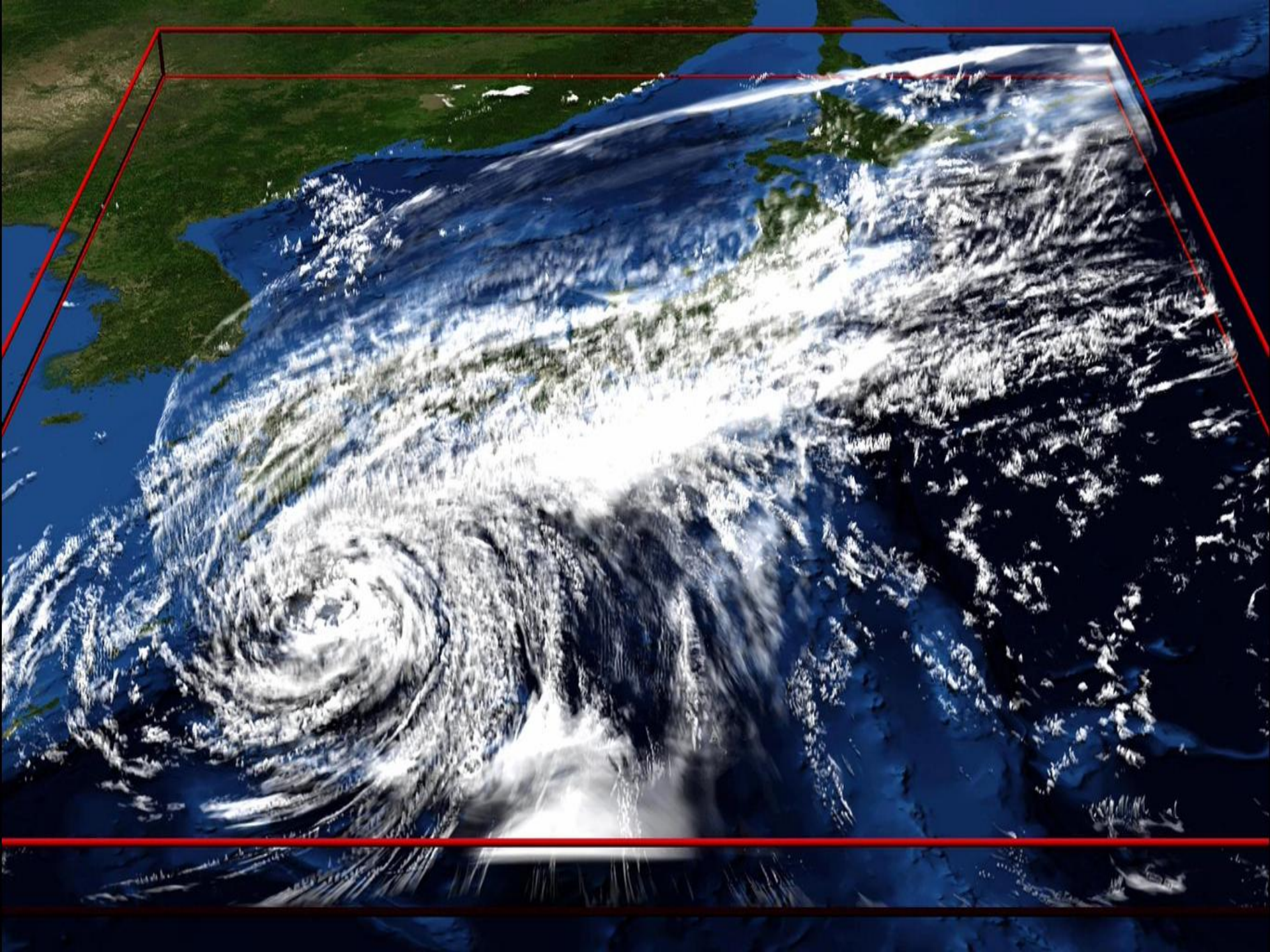
Block



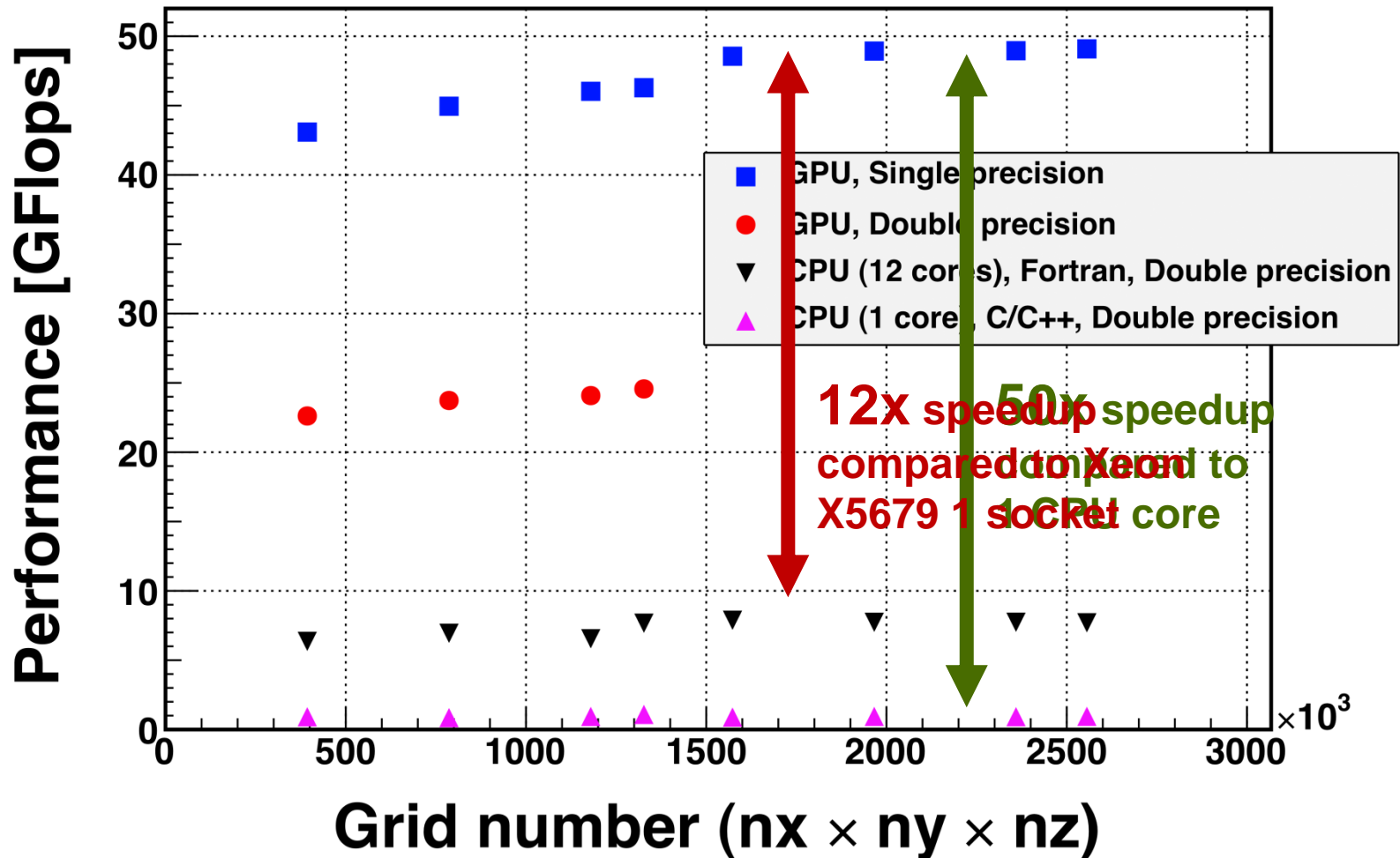
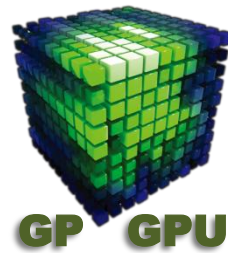
64 x 4 threads (2D) in a block

- 1D Helmholtz equation
 - ✓ Element in k depends on elements in $k \pm 1$
 - ⇒ marching in z direction





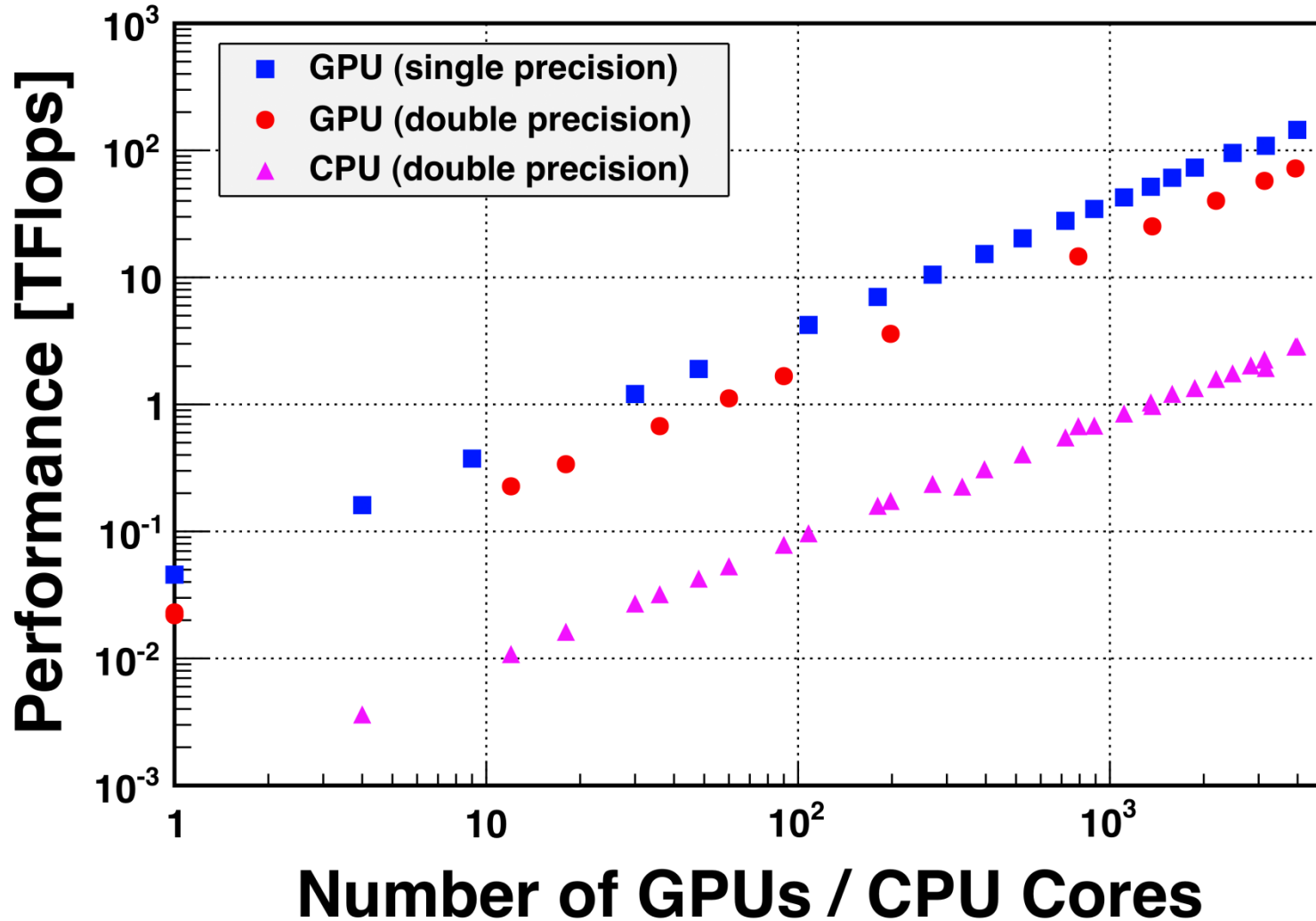
TSUBAME 2.0 (1 GPU)





GP GPU

TSUBAME 2.0 Weak Scaling

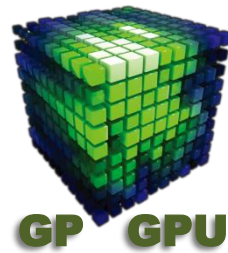


145.0 Tflops
Single precision

76.1 Tflops
Double precision

Fermi core Tesla
M2050
3990

Two-Phase Flows



Mesh Method
different from
SPH

High accuracy

- 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

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

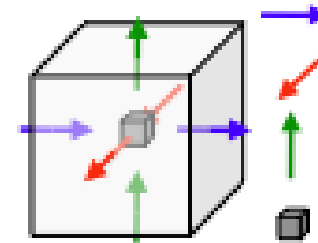
VOF continuum eq.

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

Level-Set
re-initialization

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

Staggered variable position



\mathbf{u} : velocity on x-direction

\mathbf{v} : velocity on y-direction

\mathbf{w} : velocity on z-direction

p, ϕ, ψ, ρ : scalar variables

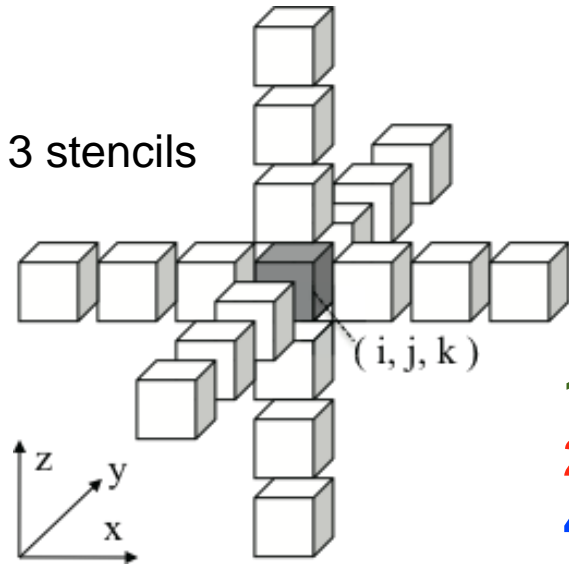


GP GPU

3D Advection Equation

Advection equation

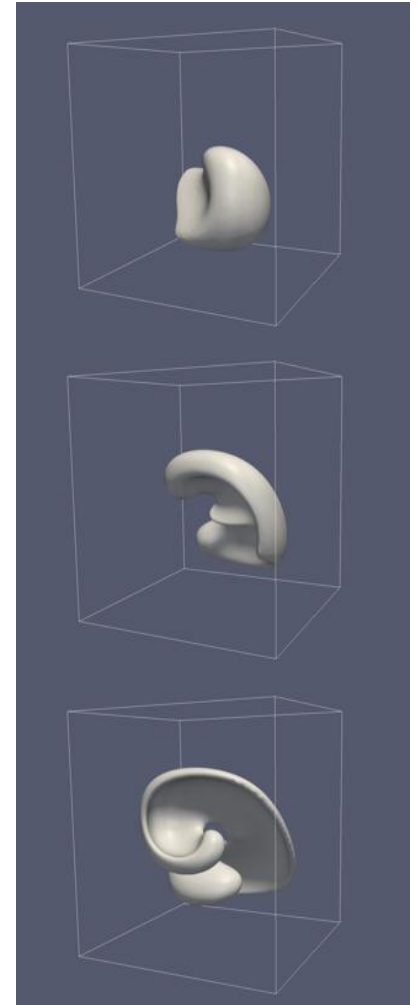
$$\frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla f = 0$$

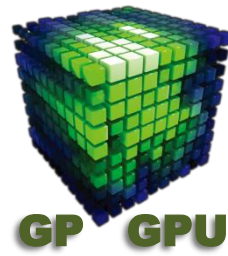


19 input values/cell
259 flop/cell (5th-WENO)
49 flop/cell (5th-up FD)

Discretization: Space : 5th-WENO
Time : 3rd TVD Runge-Kutta

312 GFlops (1GPU:GTX285)





Level-Set method (LSM)

The Level-Set methods (LSM) use the signed distance function to capture the interface. The interface is represented by the zero-level set (zero-contour).

ϕ : Level-Set function(distance function)

H : Heaviside function

$$\begin{cases} H(\phi) = \frac{1}{2} & \phi > \varepsilon \\ H(\phi) = \frac{1}{2} \left(\frac{\phi}{\varepsilon} + \frac{1}{\pi} \sin \left(\frac{\pi\phi}{\varepsilon} \right) \right) & |\phi| \leq \varepsilon \\ H(\phi) = -\frac{1}{2} & \phi < -\varepsilon \end{cases}$$

Re-initialization for Level-Set function

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

Advantage : Curvature calculation, Interface boundary

Drawback : Volume conservation

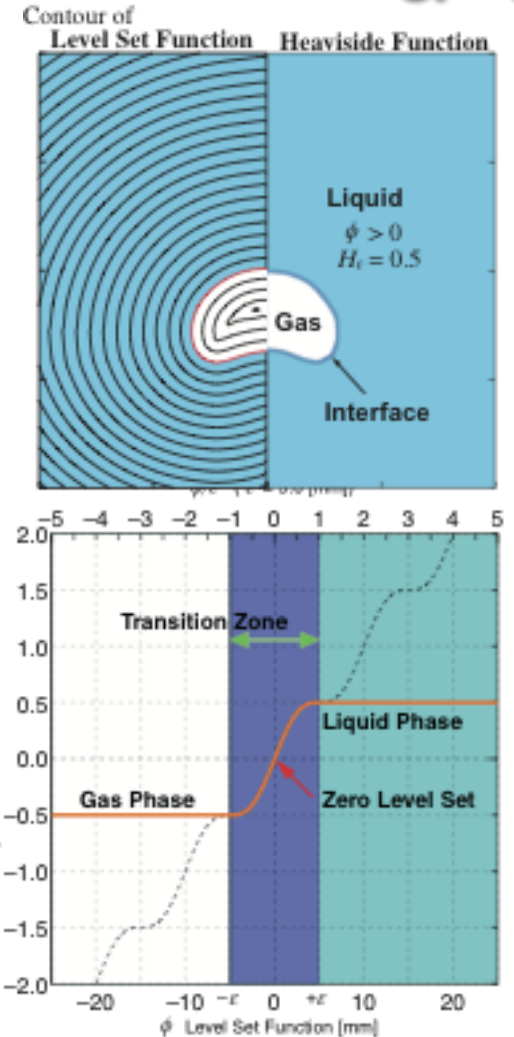
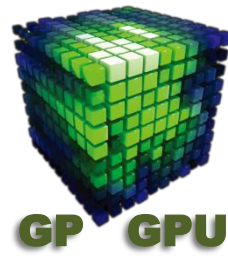


Fig. Takehiro Himeno, et. Al., JSME, 65-635,B(1999),pp2333-2340

Continuous Surface Force (CSF) model by Brackbill, Kothe and Zemach (1991)



The interfacial surface force is transformed to a volume force in the region near the interface via a delta function

Surface tension force

Curvature

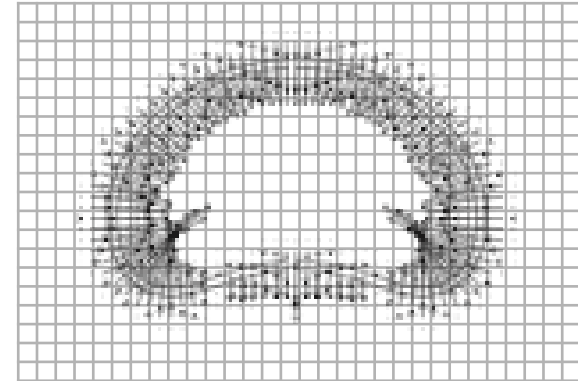
↓

$$\mathbf{F}_S = \sigma \kappa \mathbf{n}$$

← Normal vector

$$\kappa = -\nabla \cdot \mathbf{n} = -\nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}$$

$$\mathbf{F}_S = \sigma \kappa \delta(\phi) \nabla \phi$$

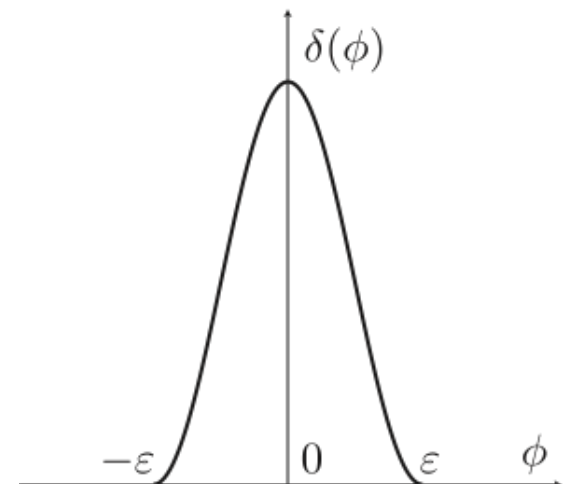


Surface tension represented by volume force

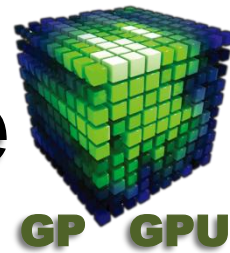
Approximate delta function

$$\delta(\phi) = \frac{\partial H(\phi)}{\partial \phi} = \frac{1}{2} \left(\frac{1}{\varepsilon} + \frac{1}{\varepsilon} \cos \left(\frac{\pi \phi}{\varepsilon} \right) \right)$$

$$\int_{-\varepsilon}^{\varepsilon} \delta(\phi) d\phi = 1$$



Anti-diffusive Interface Capture



GP GPU

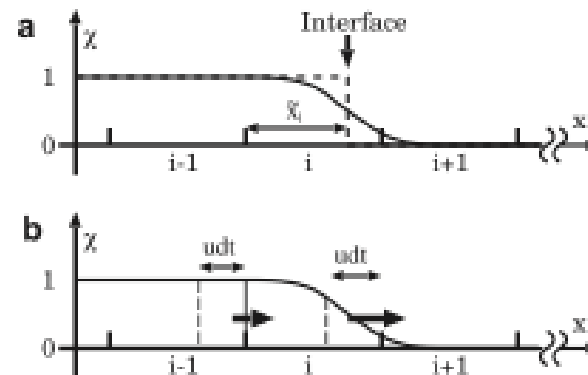
THINC (tangent of hyperbola for interface capturing) Scheme

[Xiao, etal, Int. J. Numer. Meth. Fluid. 48(2005)1023]

- VOF(volume of fluid) type interface capturing method
- Flux from tangent of hyperbola function
- Semi-Lagrangian time integration

$$F_i(x) = \frac{1}{2} \left(1 + \alpha \tanh \left(\beta \left(\frac{x - x_{i-1/2}}{\Delta x} - \tilde{x}_i \right) \right) \right)$$

$$\alpha = \begin{cases} 1 & (\text{if } n_x > 0) \\ -1 & (\text{if } n_x \leq 0) \end{cases}$$



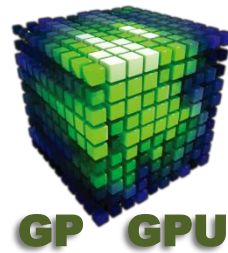
- 1D implementation can be applied to 2D & 3D → Simple

$$Fl_{x,i+1/2} = - \int_{x_{i+1/2}}^{x_{i+1/2} - u_{i+1/2} \Delta t} F_{up}(x) dx \quad up = \begin{cases} i & (\text{if } u_{i+1/2} > 0) \\ i+1 & (\text{if } u_{i+1/2} \leq 0) \end{cases}$$

- Finite Volume like usage
 - * THINC is the method how to compute flux

→ 3 kernel (x, y, z) can be fused to 1 kernel. Merit in memory R/W

Sparse Matrix Solver

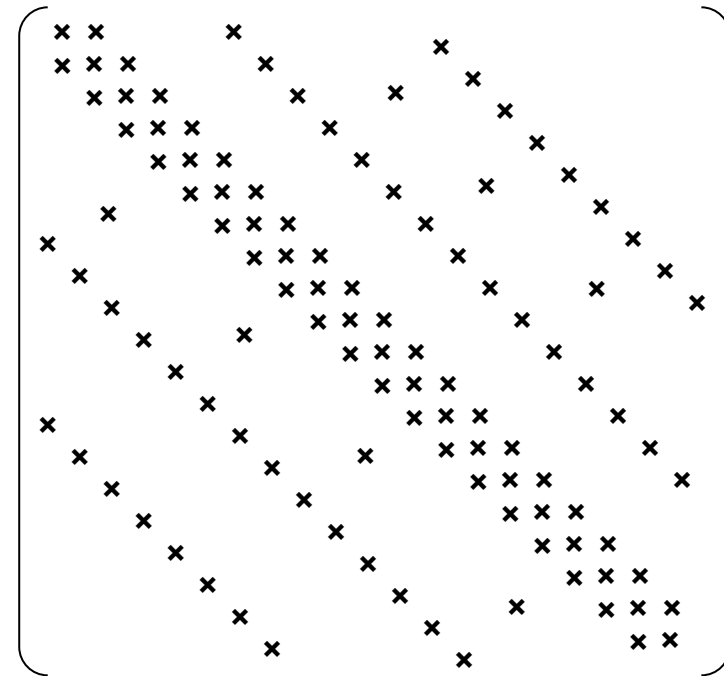


$$\mathbf{Ax} = \mathbf{b} \quad \text{for} \quad \nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = \frac{\nabla \cdot \mathbf{u}}{\Delta t}$$

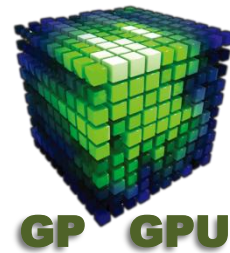
Krylov sub-space methods:
CG, BiCGStab, GMRes, , ,

Pre-conditioner:
Incomplete Cholesky,
ILU, MG, AMG,
Block Diagonal Jacobi

Non-zero Packing:
CRS \rightarrow ELL, JDL



BiCGStab + AMG



Collaboration with
Mizuho Information & Research Institute

Set $k = 0$ $r_0 = p_0 = M^{-1} \mathbf{b} - A\mathbf{x}_0$

for $k = 0; k < N; k++;$

$$\alpha_k = \frac{\langle \mathbf{r}_k, \mathbf{r}_k \rangle}{\langle \mathbf{r}_k, M^{-1} A \mathbf{p}_k \rangle} \quad \mathbf{q}_k = \mathbf{r}_k - \alpha_k M^{-1} A \mathbf{p}_k \quad \omega_k = \frac{\langle \mathbf{q}_k, M^{-1} A \mathbf{q}_k \rangle}{\langle M^{-1} A \mathbf{q}_k, M^{-1} A \mathbf{q}_k \rangle}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k + \omega_k \mathbf{q}_k$$

$$\mathbf{r}_{k+1} = \mathbf{q}_k - \omega_k M^{-1} A \mathbf{q}_k$$

if $\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle \leq \varepsilon^2 \langle \mathbf{b}, \mathbf{b} \rangle$ exit;

$$\beta_k = \frac{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle}{\omega_k \langle \mathbf{r}_k, M^{-1} A \mathbf{p}_k \rangle}$$

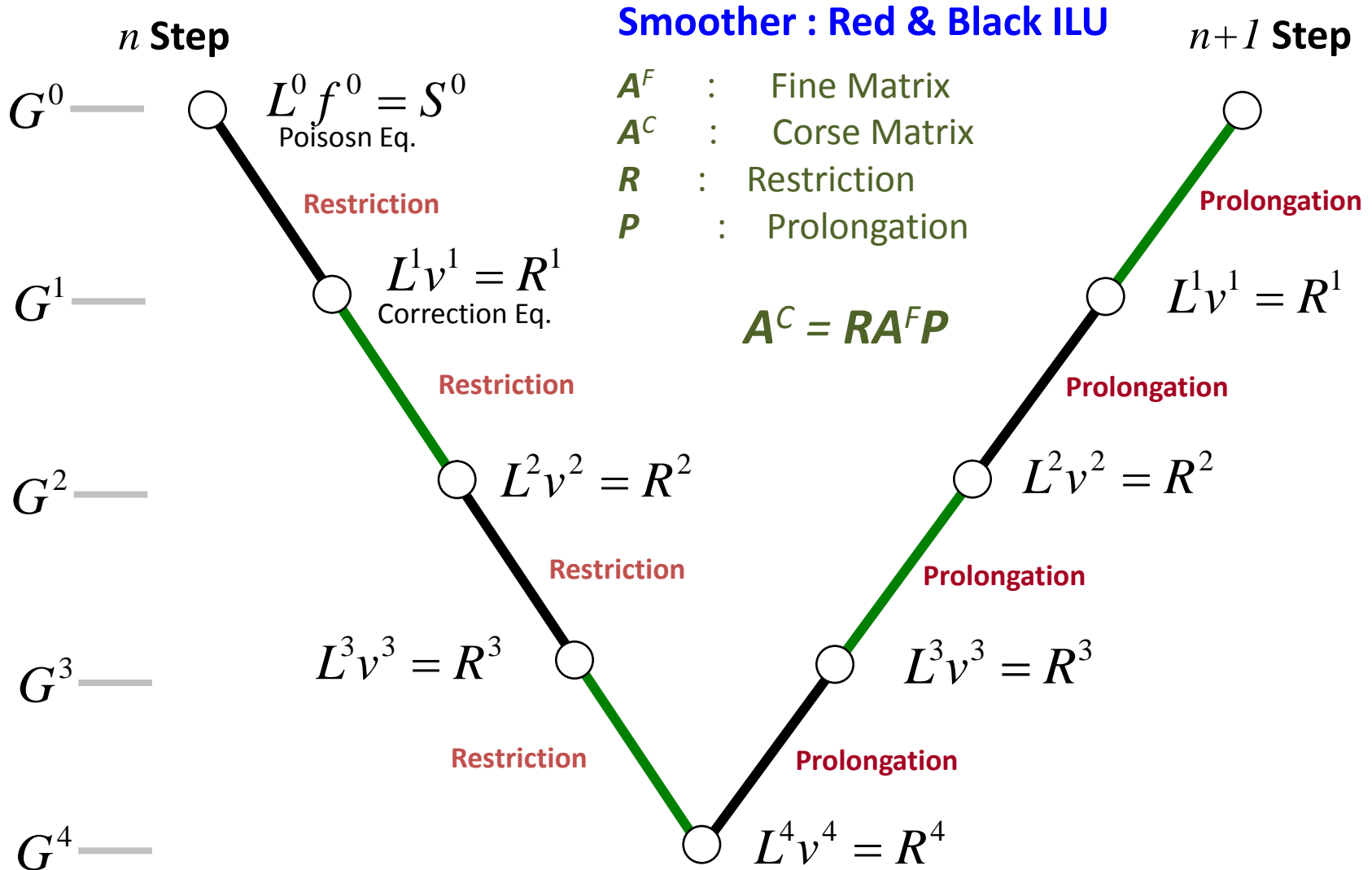
$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{r}_k - \omega_k M^{-1} A \mathbf{p}_k$$

loop end

AMG V-Cycle

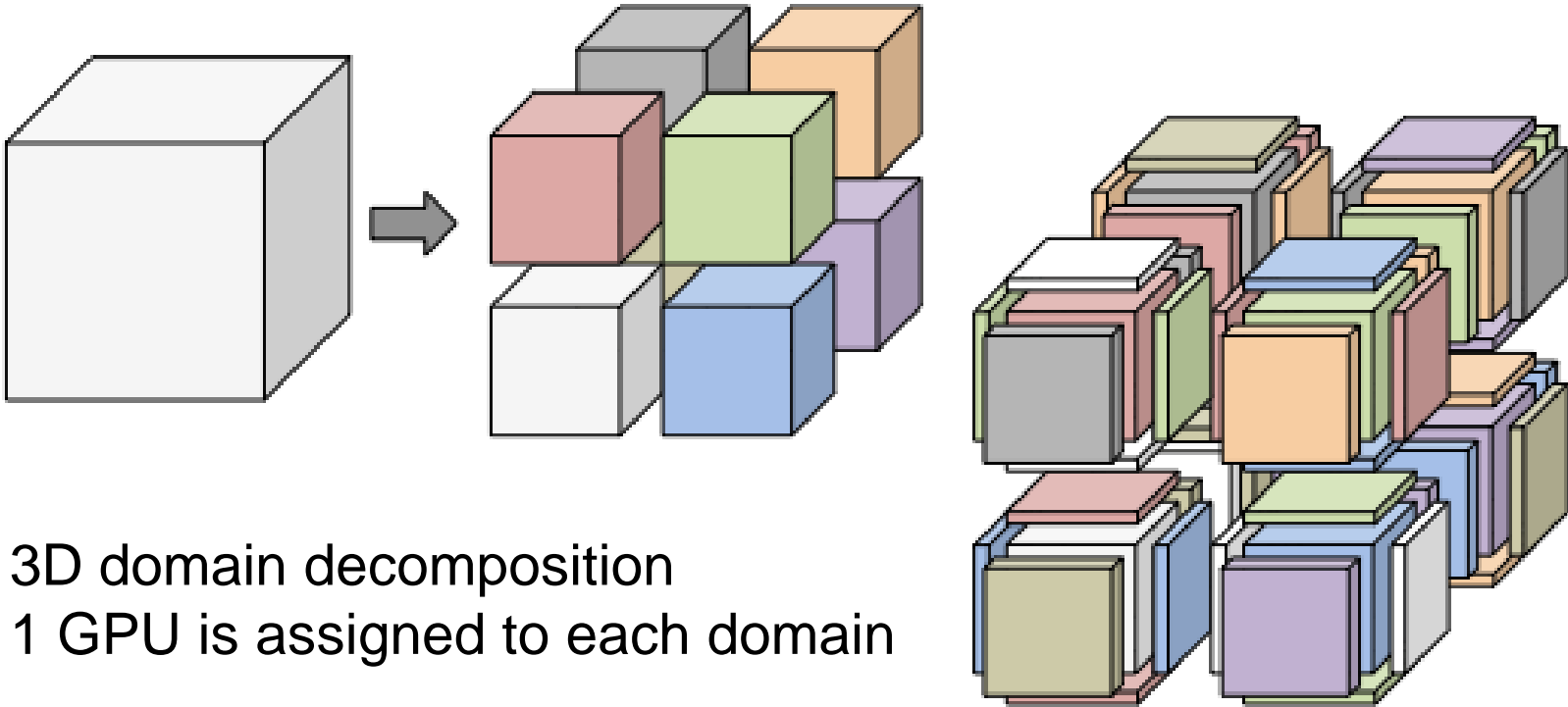


GP GPU





Multi-Dimensional Domain Decomposition



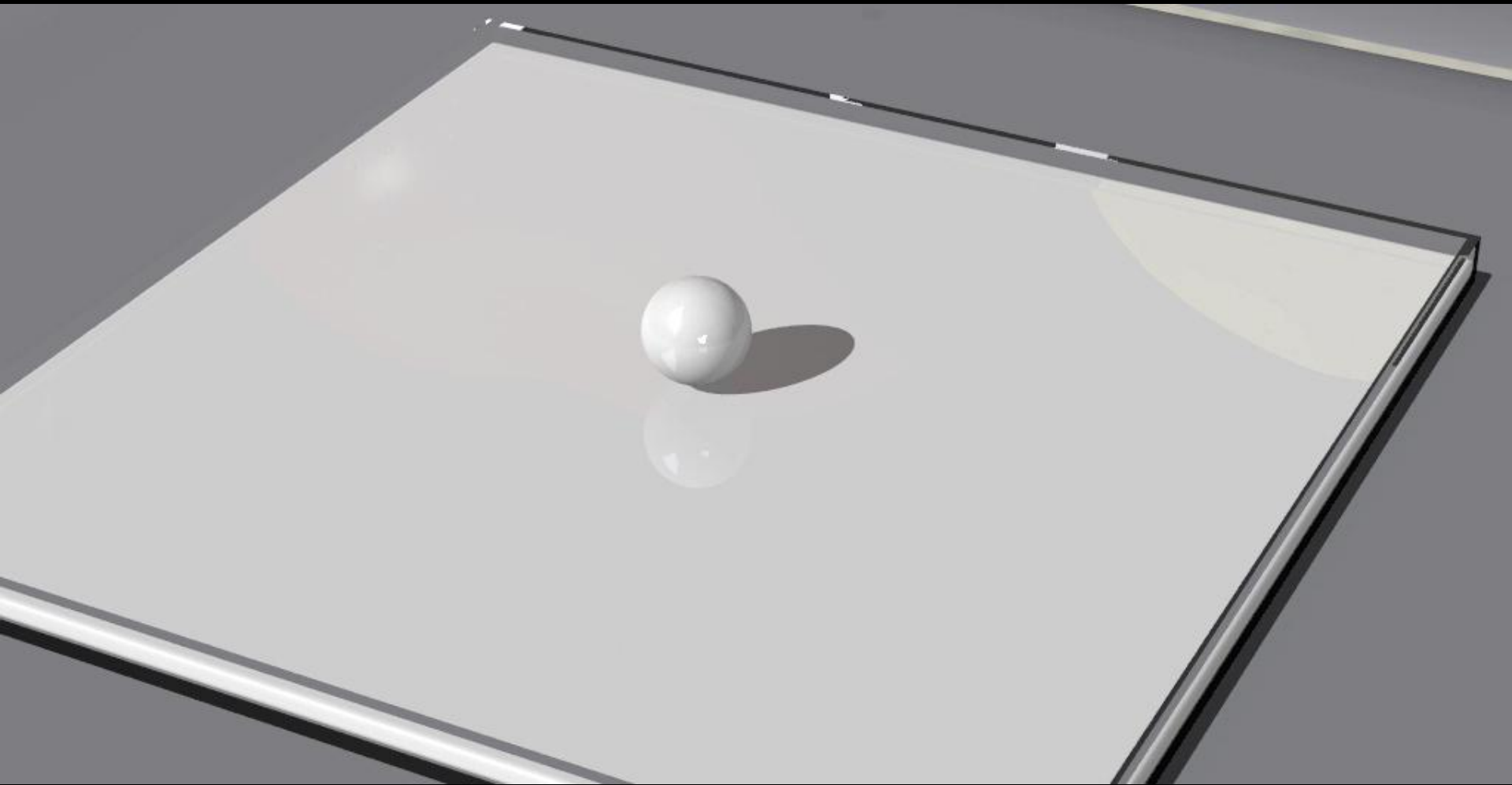
- 3D domain decomposition
- 1 GPU is assigned to each domain

- Communication buffer for each face
- Host buffer & Device buffer

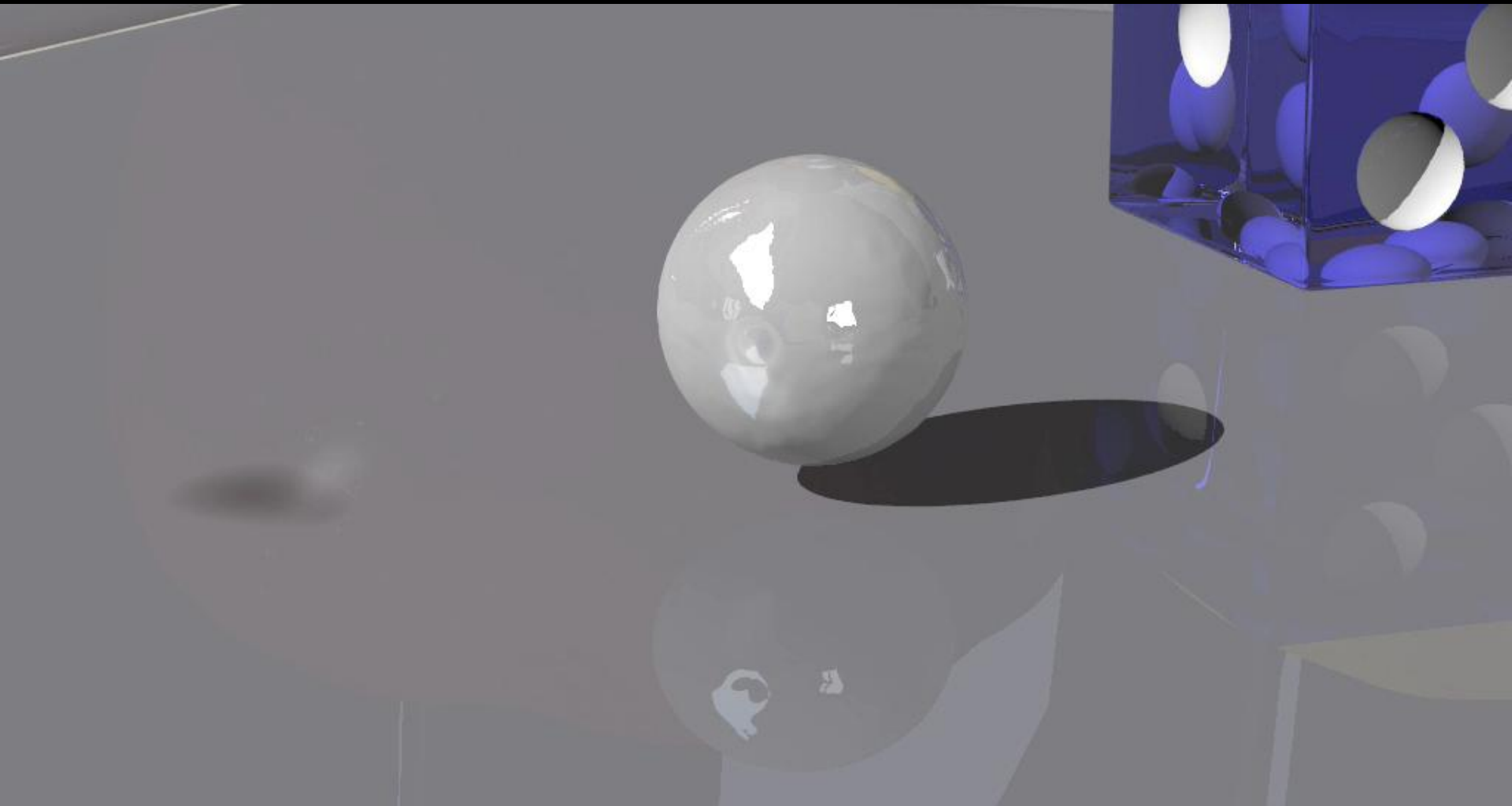


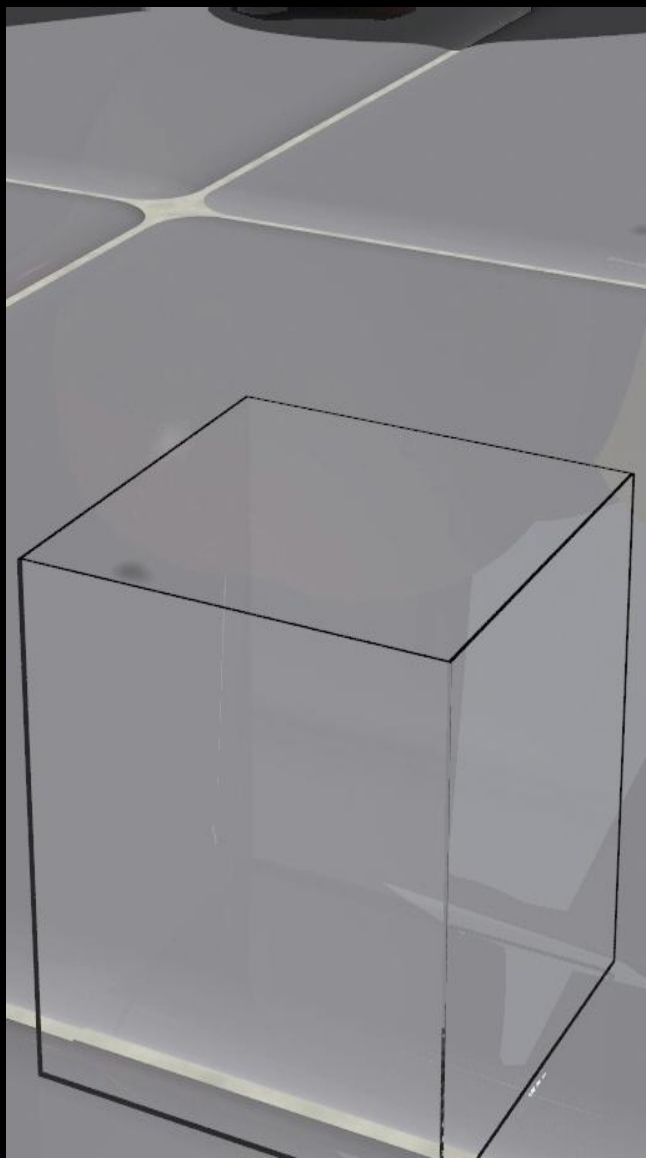


Milk Crown

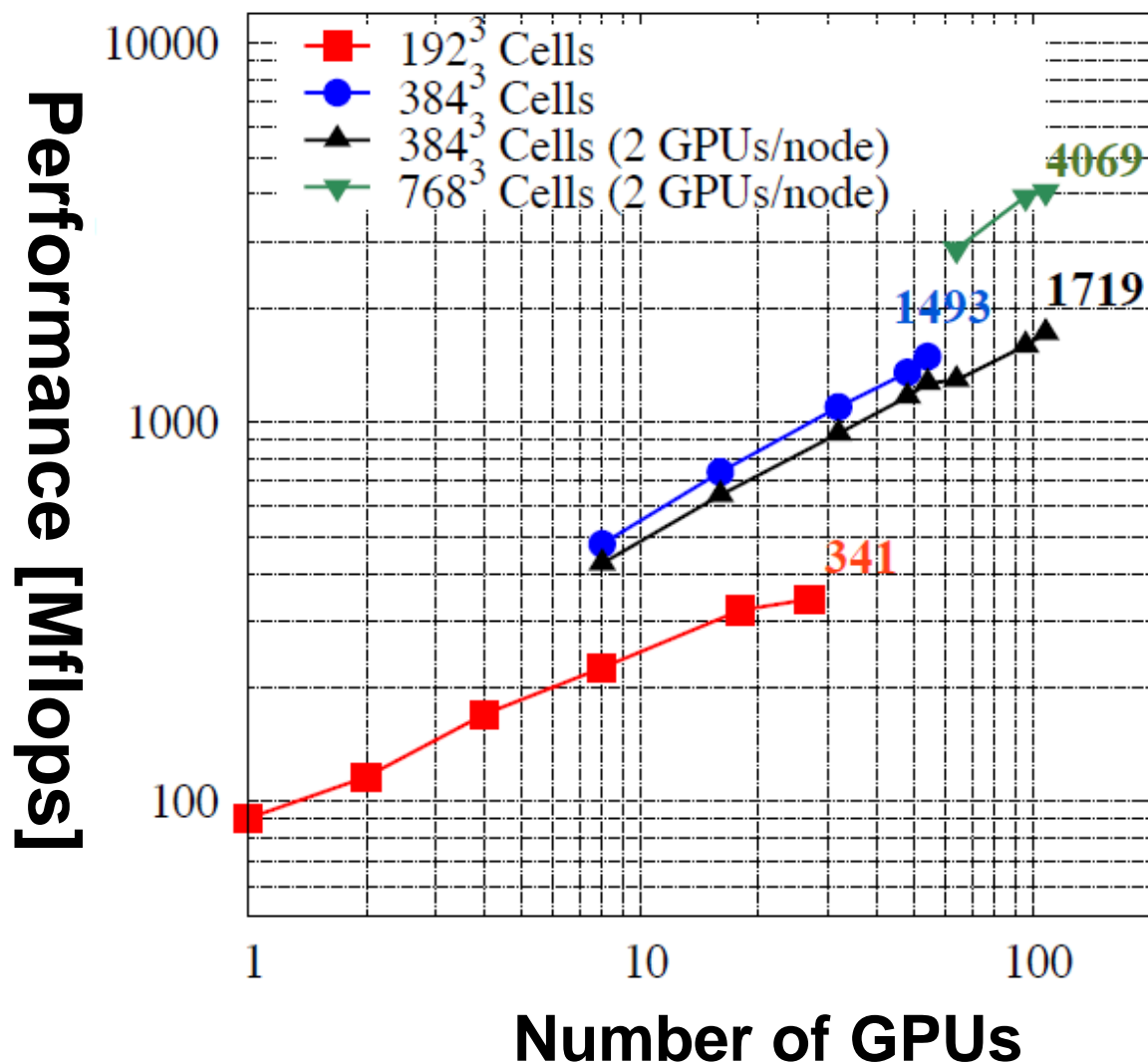
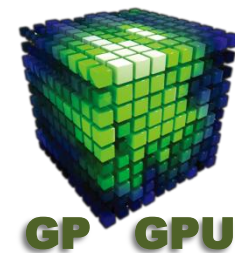


Drop on dry floor

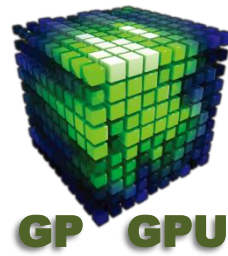




Multi-GPU Performance on TSUBAME 1.2



Multi-GPU Summary



Some CFD applications show good strong scalability up to 32 GPUs in the case of TSUBAME.

The balance between computation and communication performance becomes bad because of the high GPU performance.

In order to achieve high performance for multi-GPU application, the overlapping technique between computation and communication is very important.

Be careful for GPU-to-CPU data transfer (cudaMemcpy) and CPU-to-CPU data transfer (MPI library).



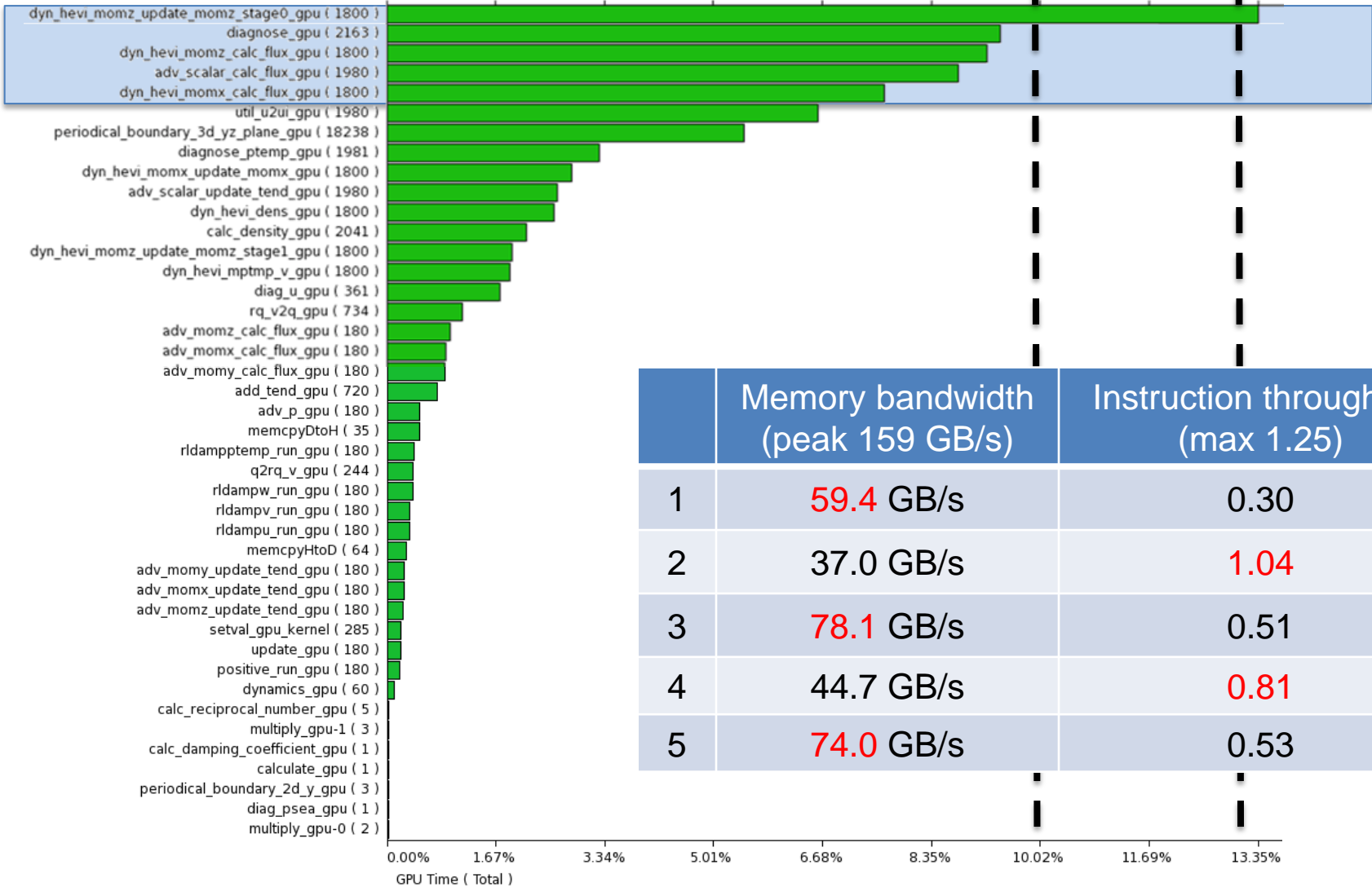
Thank you
for your kind attention

Profile of functions in ASUCA



GTX 285

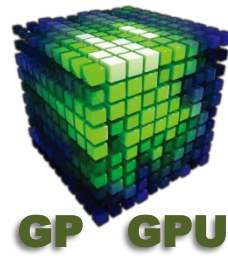
0 % GPU Time (Total) 10 % 13 %



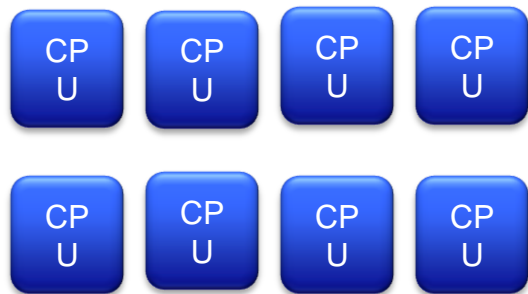
	Memory bandwidth (peak 159 GB/s)	Instruction throughput (max 1.25)
1	59.4 GB/s	0.30
2	37.0 GB/s	1.04
3	78.1 GB/s	0.51
4	44.7 GB/s	0.81
5	74.0 GB/s	0.53

TSUBAME 1.2 node detail

SunFire X4600



Dual core Opteron 2.4GHz



Memory 32GB~128GB

DDR2

Telsa
S1070



102
GB/s
VRAM



PCI Express 1.0 x8 2



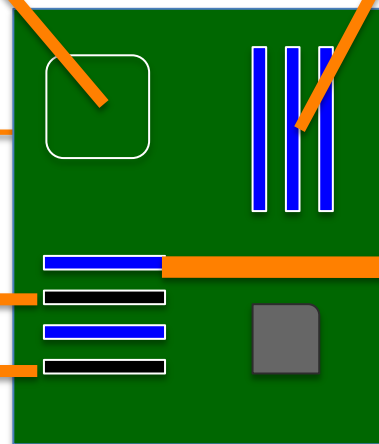
102
GB/s
VRAM



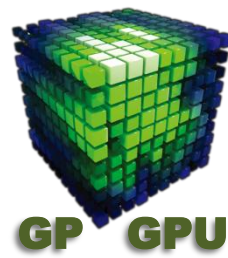
1000BASE-T
0.125 GB/s

InfiniBand SDR x2

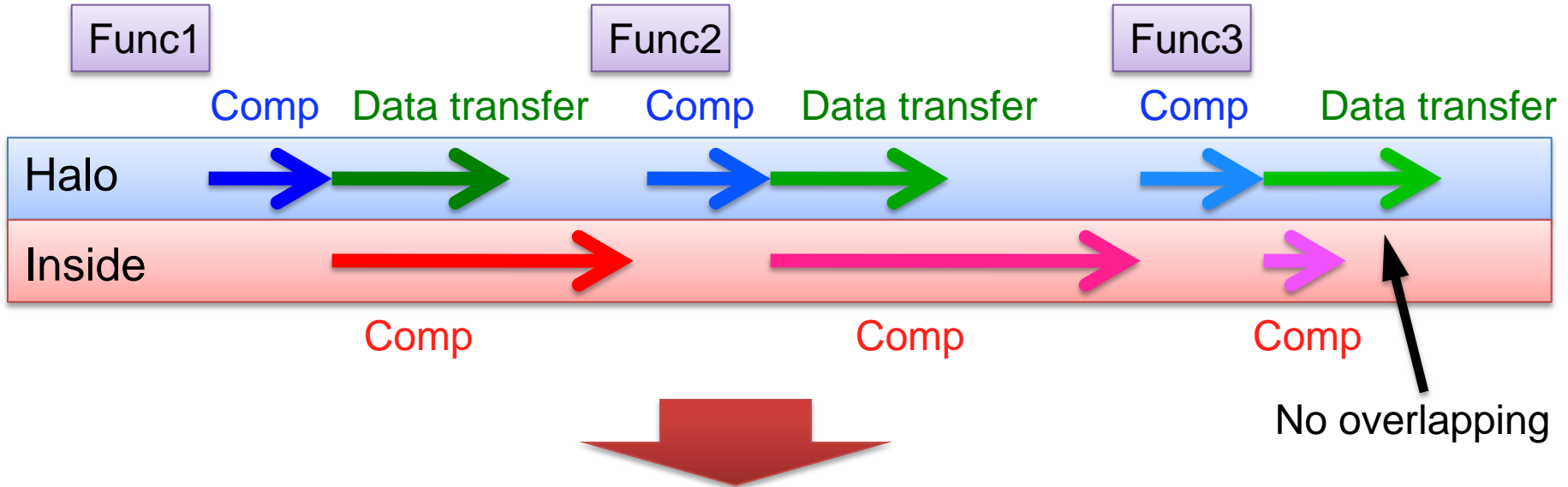
1 GB/s x2



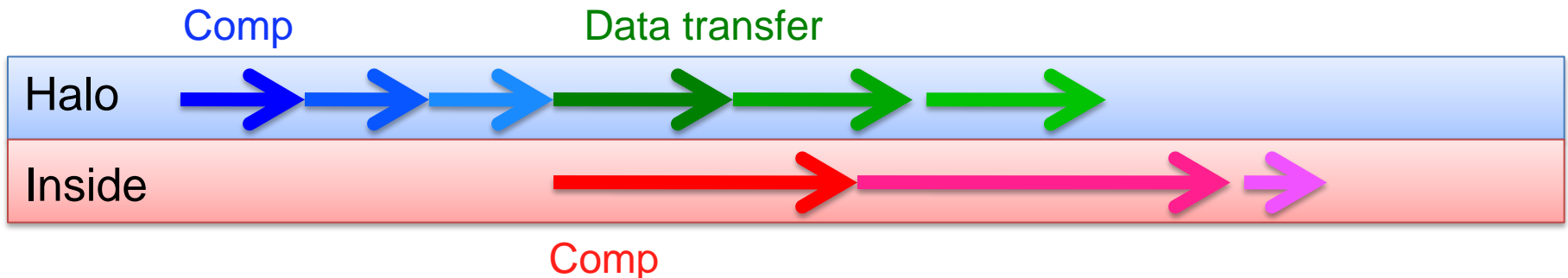
Re-ordering the communication and computation



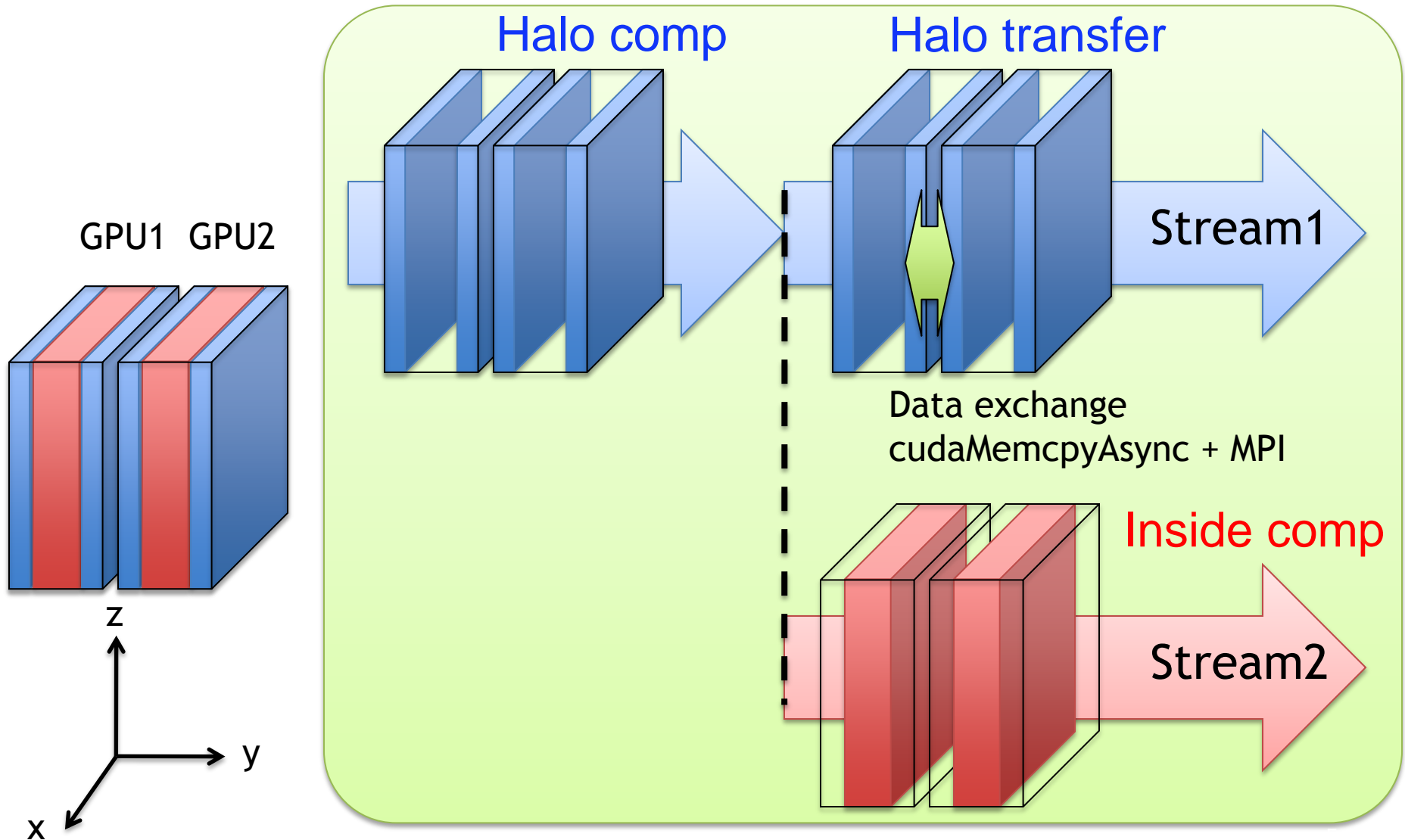
■ Overlapping comm and comp in each function



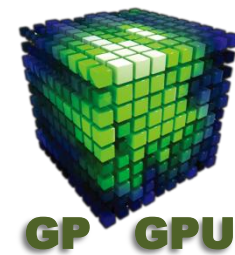
■ Overlapping overall comms and comps in functions



Overlapping communication with computation



Implementation : Advection



Thread

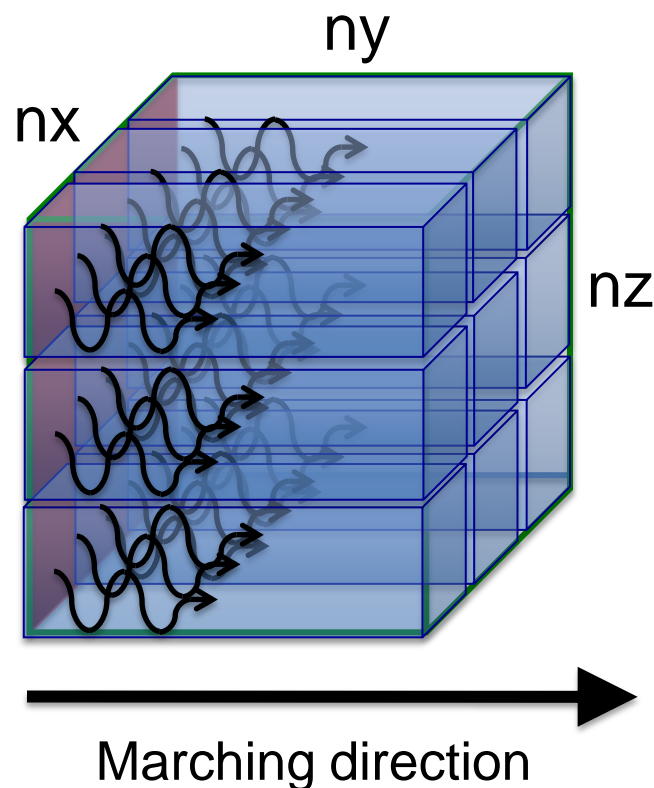


Block

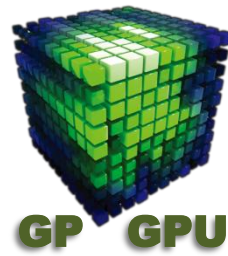


64 x 4 threads (2D) in a block

- Each thread specifies a (x, z) point, marching in y
 - ✓ Improve data transfer performance using domain decomposition



Using Shared Memory



- Shared Memory (SMem) = Software Managed Cache
- Read a 2D sub-domain from VRAM into SMem
- Advection : 12-point stencil
 - ✓ Store the xz-slice in $(64 + 3) \times (4 + 3)$ SMem

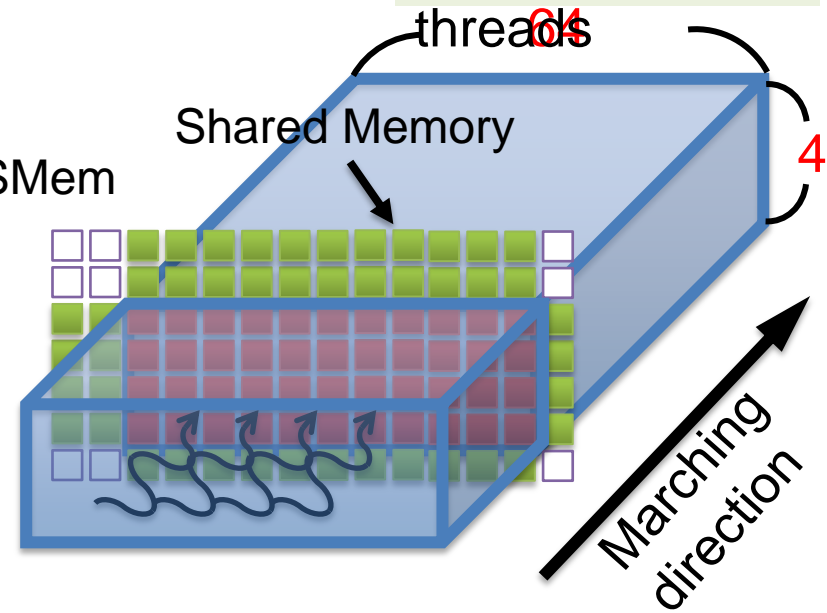
1 Block
= 64 x 4
threads

Access GMem directly : 4 + 4 read,
1write



Using SMem : ~1

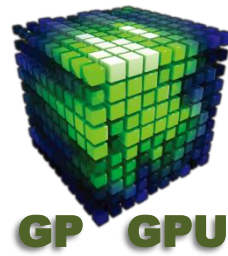
read, 1write



- 2D sub-domain
- Halo
- Not in use

	Shared Memory	VRAM (Global Memory)
Access speed	~ 2 cycle	400-600 cycle
Capacity	16 kByte/Block	2 GByte (Total)

Using Registers in marching direction

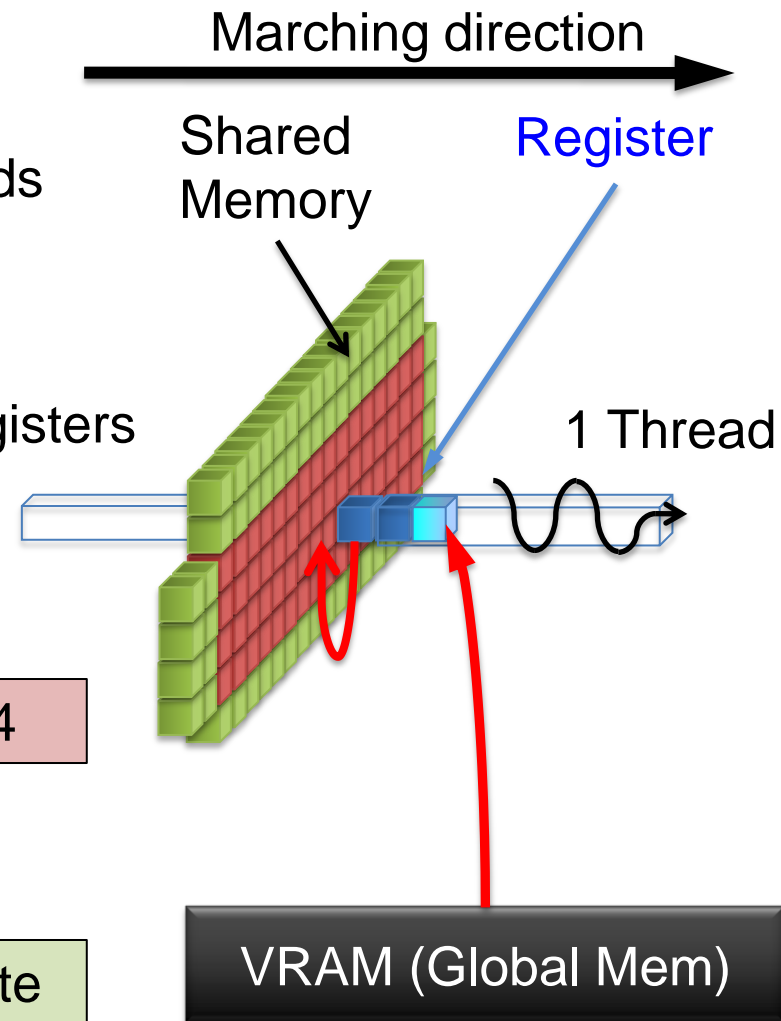


- Register
 - ✓ Access speed : 1 cycle
 - ✓ used for data not shared among threads
- Advection : 12-point stencil
 - ✓ Each thread keeps 4 y-elements in registers
 - ✓ Elements are reuse

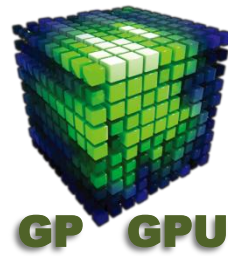
Access GMem directly : 4 + 4 + 4
read, 1write



Using SMem and Registers : ~1 read, 1write

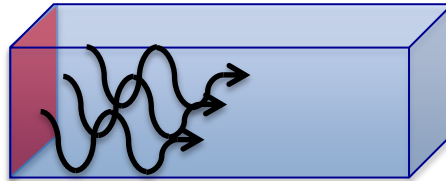


Implementation : 1D Helmholtz equation



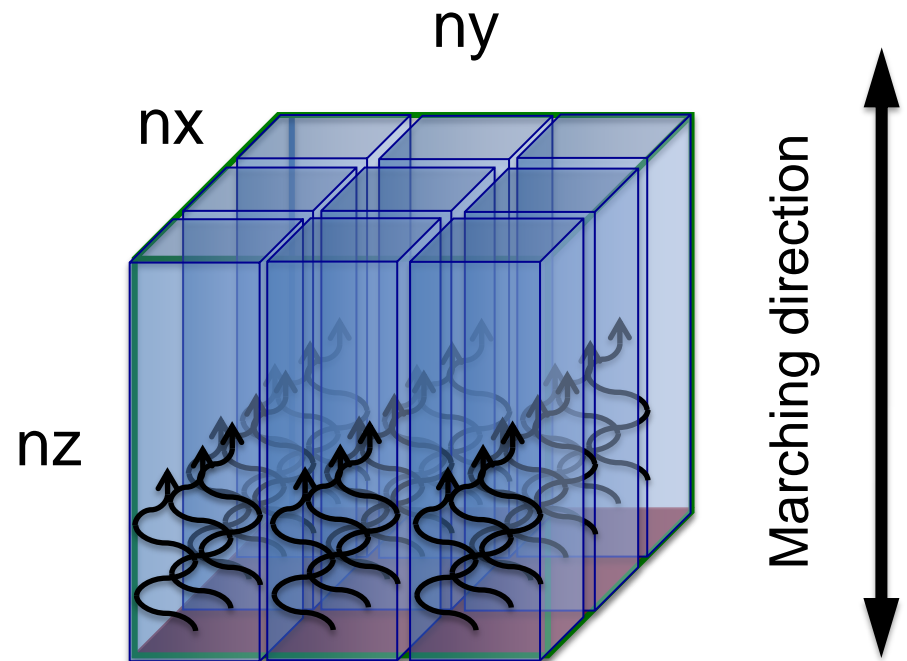
Thread 

Block

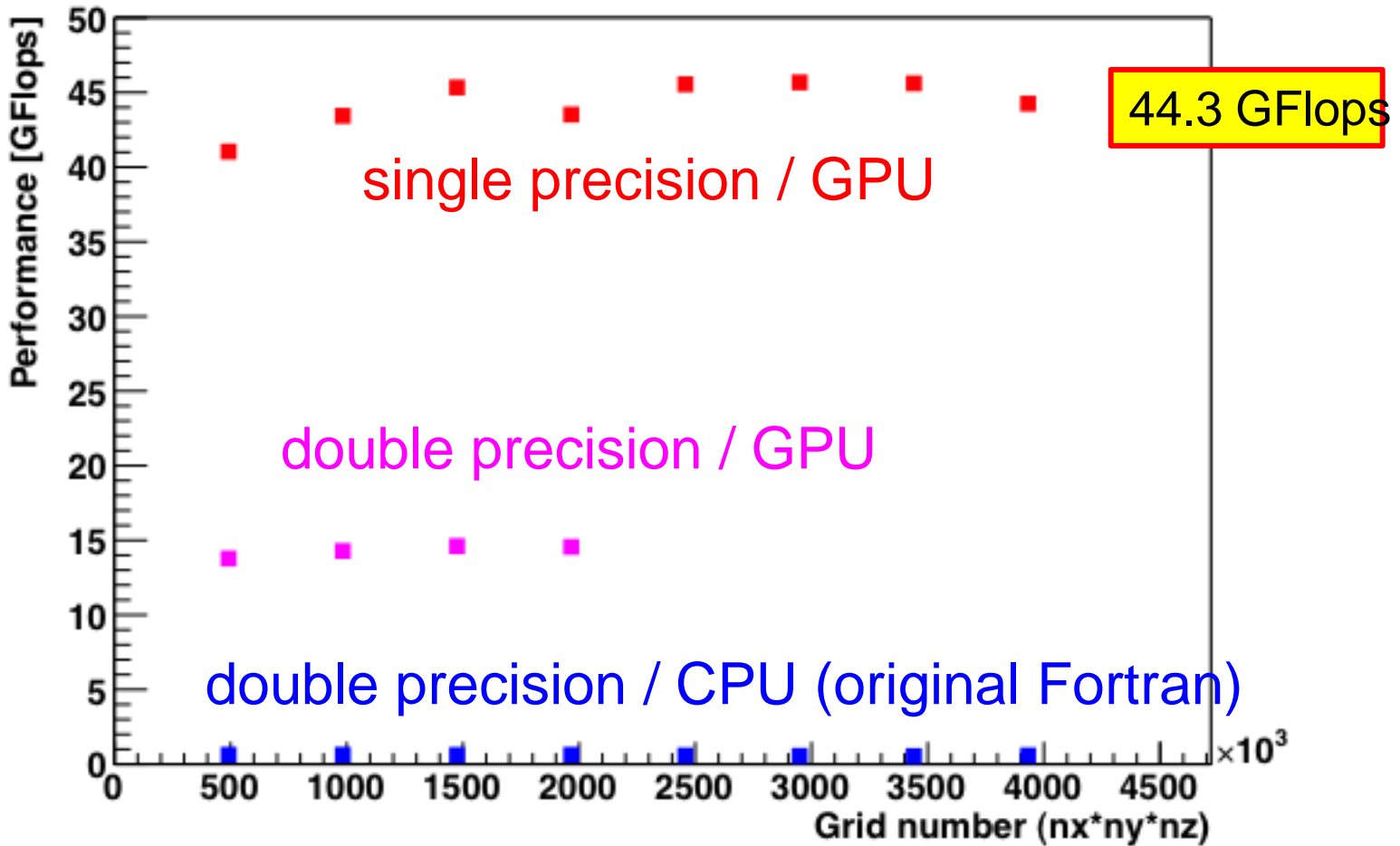
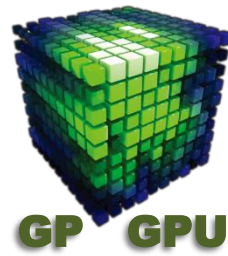


64 x 4 threads (2D) in a block

- 1D Helmholtz equation
 - ✓ Element in k depends on elements in $k \pm 1$
 - ⇒ marching in z direction



Single GPU Performance



Mountain Wave Test
NVIDIA Tesla S1070 card

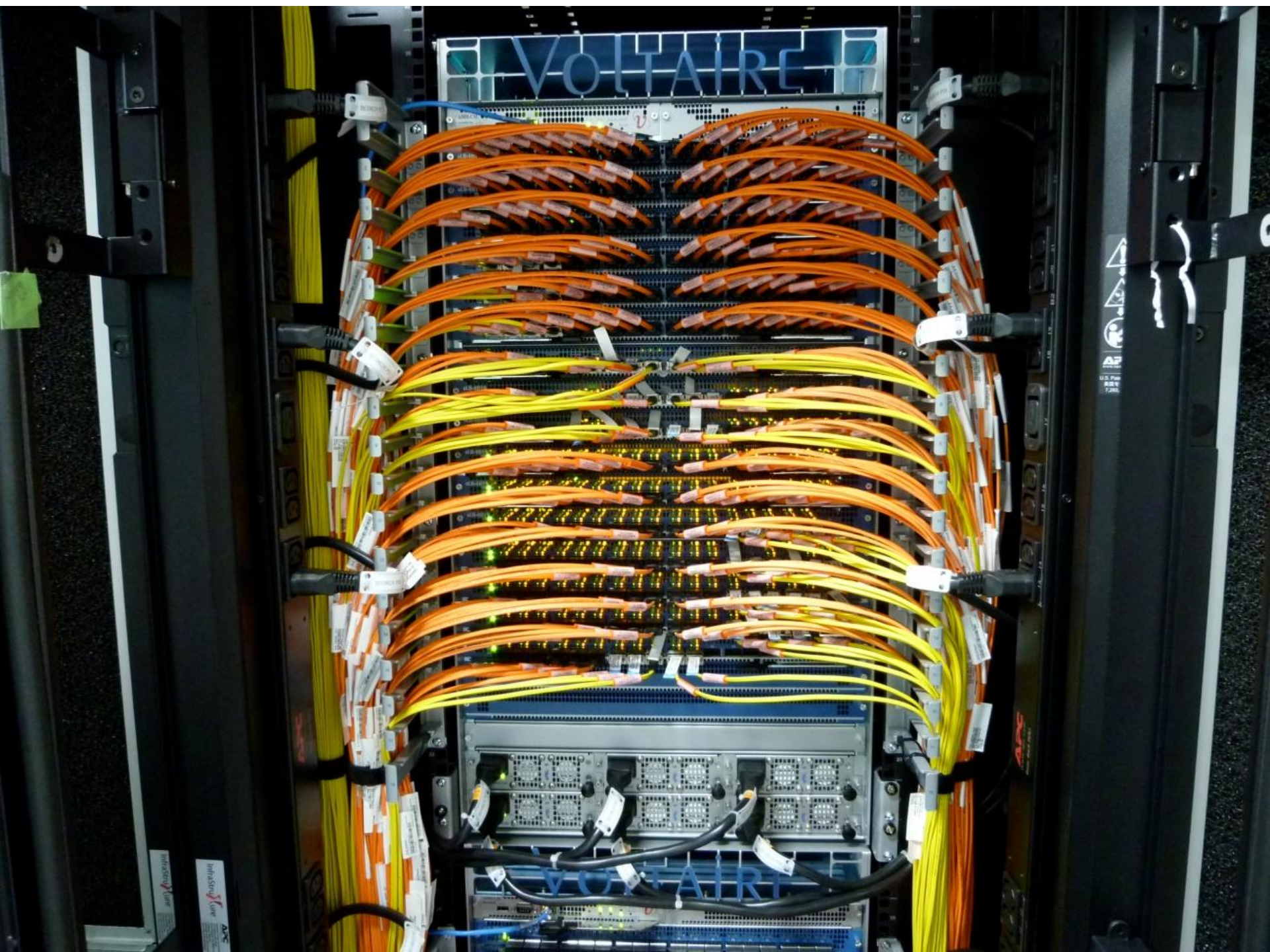
↑
320 x 256 x 64

TSUBAME 2.0



GPU M2050





VOLTAIRE

VOLTAIRE

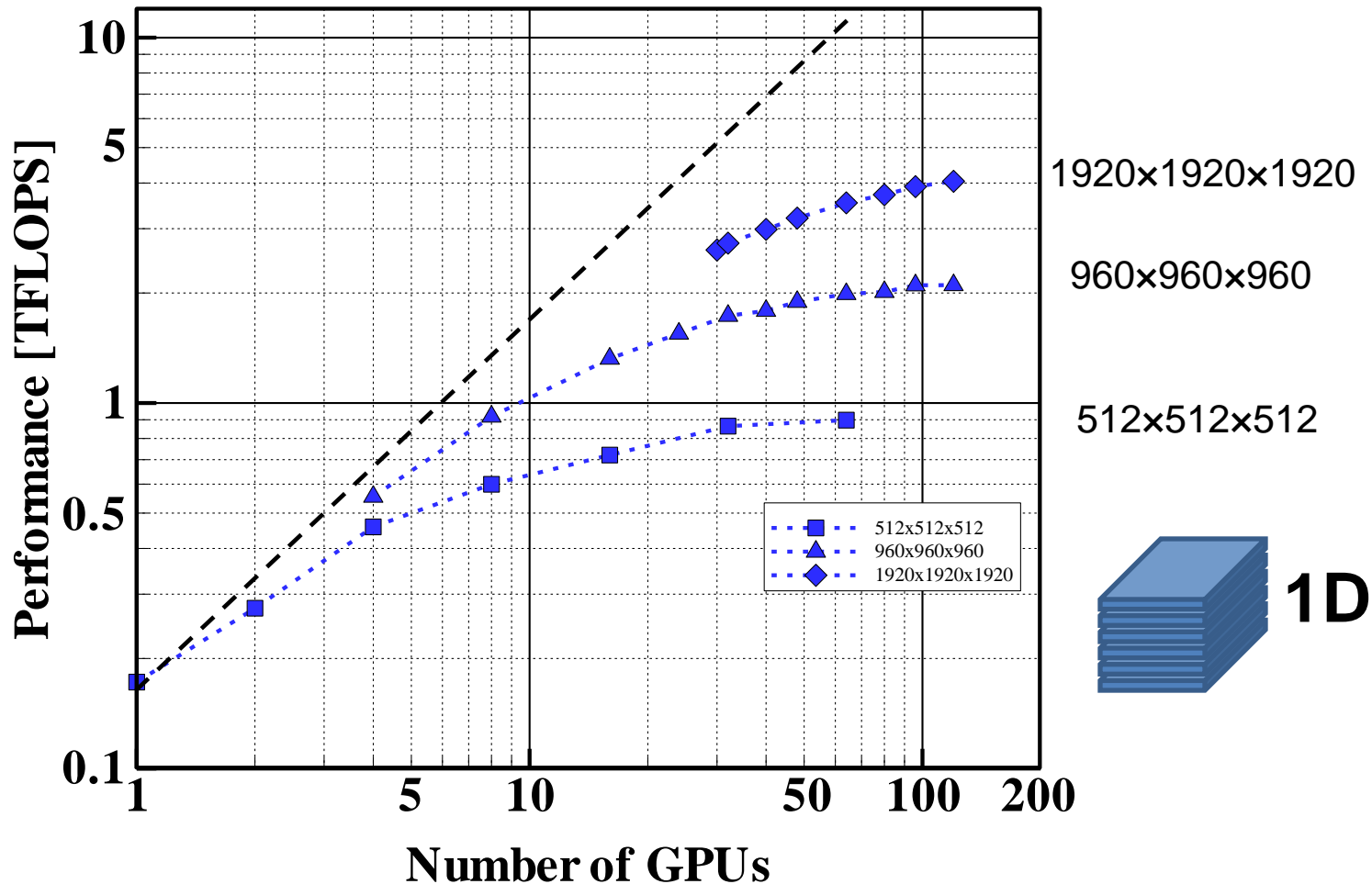
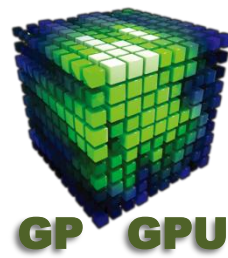
AP
U.S. For
RBY
7.05

Infocore
View

Infocore
View
APC

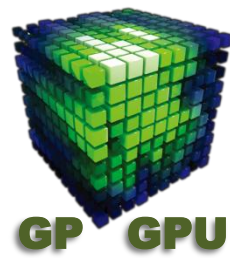
Multi-GPU Performance

w/o overlapping

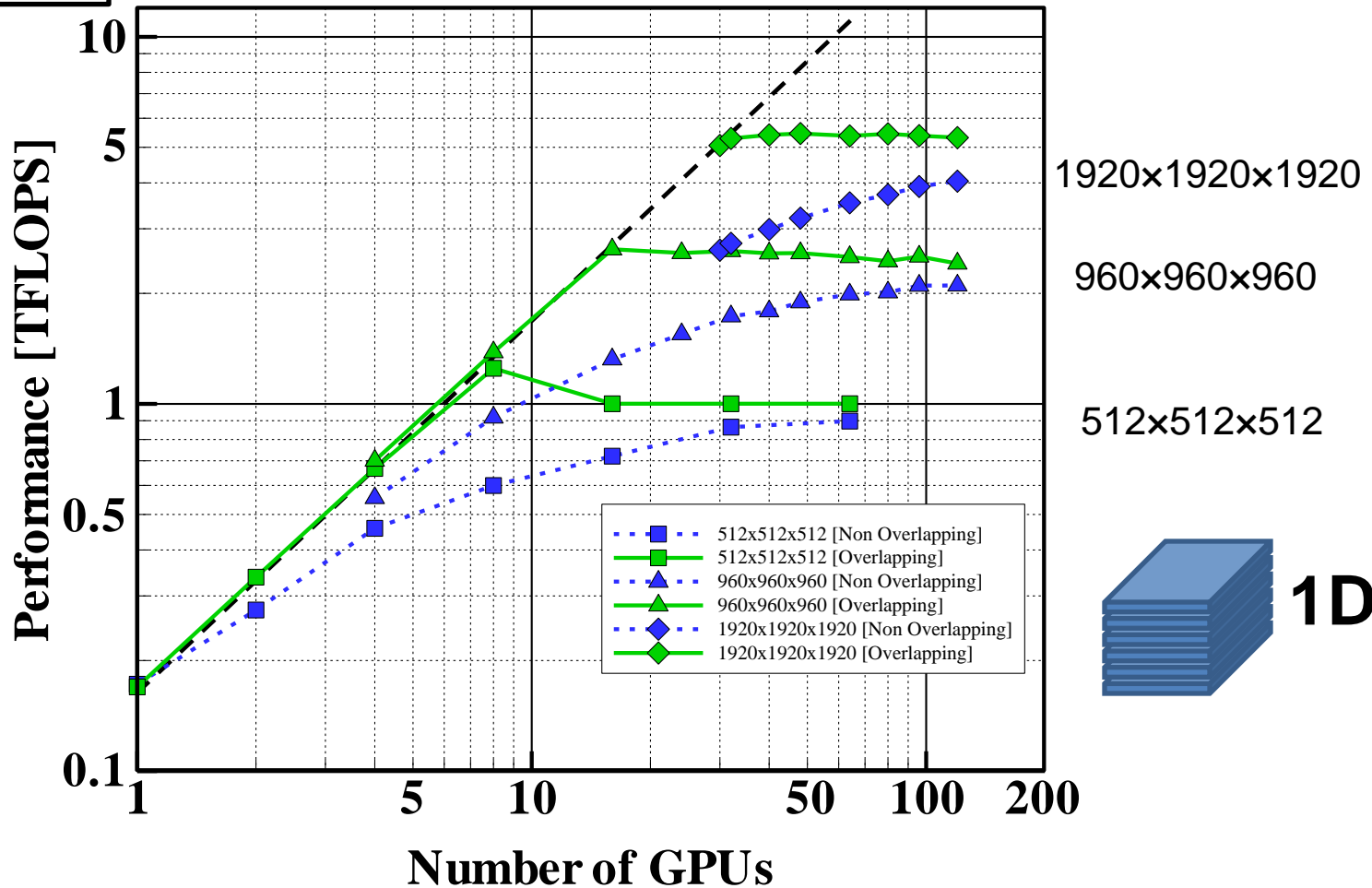


Multi-GPU Performance

w/o overlapping



2 GPU/node



Thread Assignment

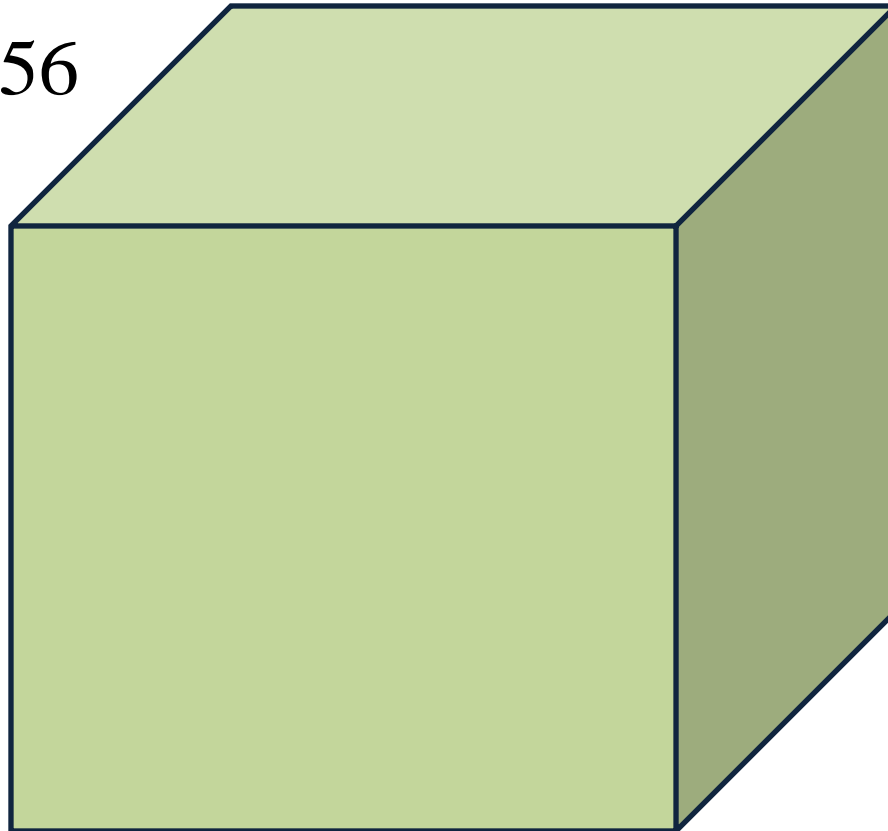


GP GPU

$$nx = 256$$

Mesh number
256x256x256

$$ny = 256$$



$$nz = 256$$

Thread Assignment

