Effective Dynamic Load Balance using Space-Filling Curves for Large-scale SPH Simulations on GPU-rich Supercomputers

Dr. Satori Tsuzuki
(Japan Agency for Marine-Earth Science and Technology)

Prof. Takayuki Aoki
(Tokyo Institute of Technology)

Smoothed Particle Hydrodynamics

- ✓ SPH (Smoothed Particle Hydrodynamics) is a short-range interaction based particle method for simulations of incompressible flows.
- ✓ In SPH, physical properties are considered to be distributed in the neighboring area in the range of an effective radius of smoothing functions *W*.

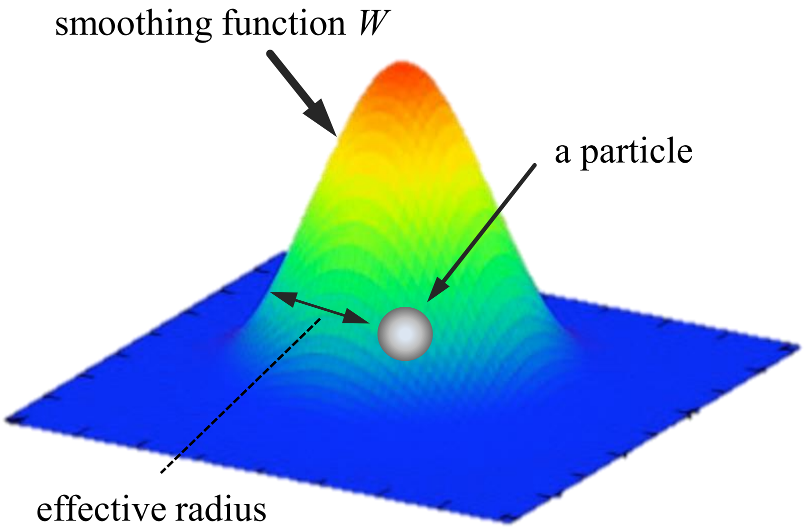
A physical value $\phi(\boldsymbol{x}) = \int_{r \in W} \phi(\boldsymbol{r}) W(\boldsymbol{x} - \boldsymbol{r}) d\boldsymbol{r}$

Normalization $\int W(\boldsymbol{x} - \boldsymbol{r}) d\boldsymbol{r} = 1$

Discretization

$$\phi(\boldsymbol{x}) = \sum_j m_j \frac{\phi_j}{\rho_j} W(\boldsymbol{x} - \boldsymbol{x}_j)$$

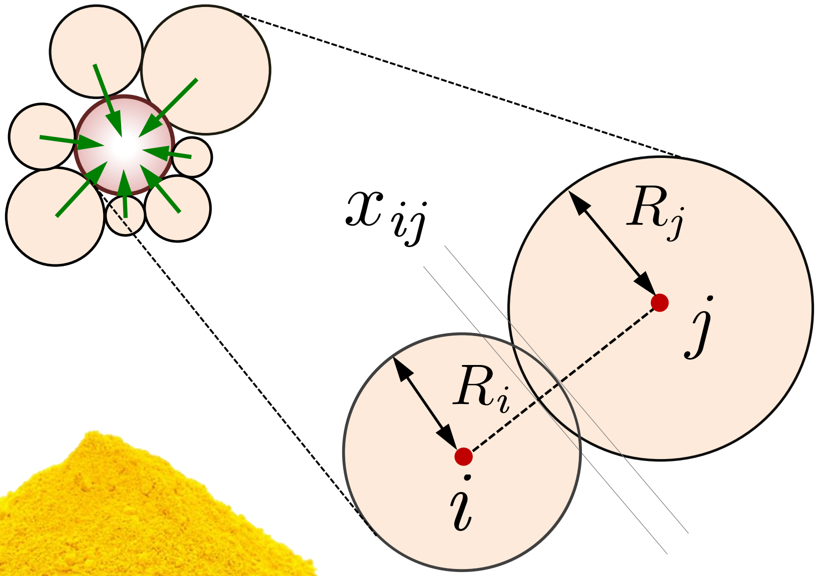
m_j : mass ρ_j : density



Discrete Element Method

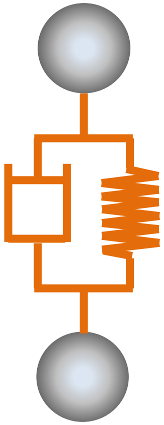
- **A short-range interaction based particle method for granular materials**

  Interactions among particles are calculated only when they contact.

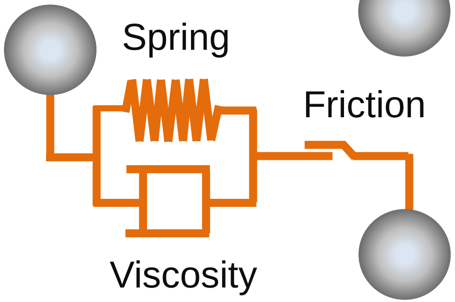


$$F_{ij} = -kx_{ij} - \gamma\dot{x}_{ij}$$

Normal direction



Tangential direction



Short-range Interaction

- **Interactions among only particles in the neighboring cells**

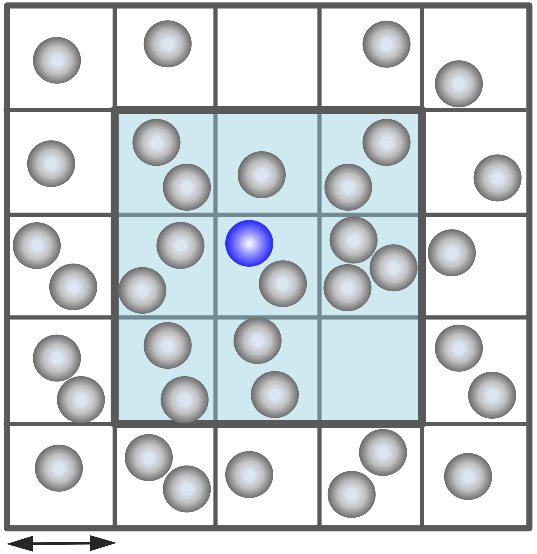
Neighbor-particle List Method

✓ Interactions among only particles which exist in neighboring cells are calculated.

✓ **Neighbor-particle list** is often coupled with **linked-list technique** to reduce the amount of memory consumption.

✓ It becomes possible to reduce the interaction's costs from $O(N^2)$ to $O(N)$.

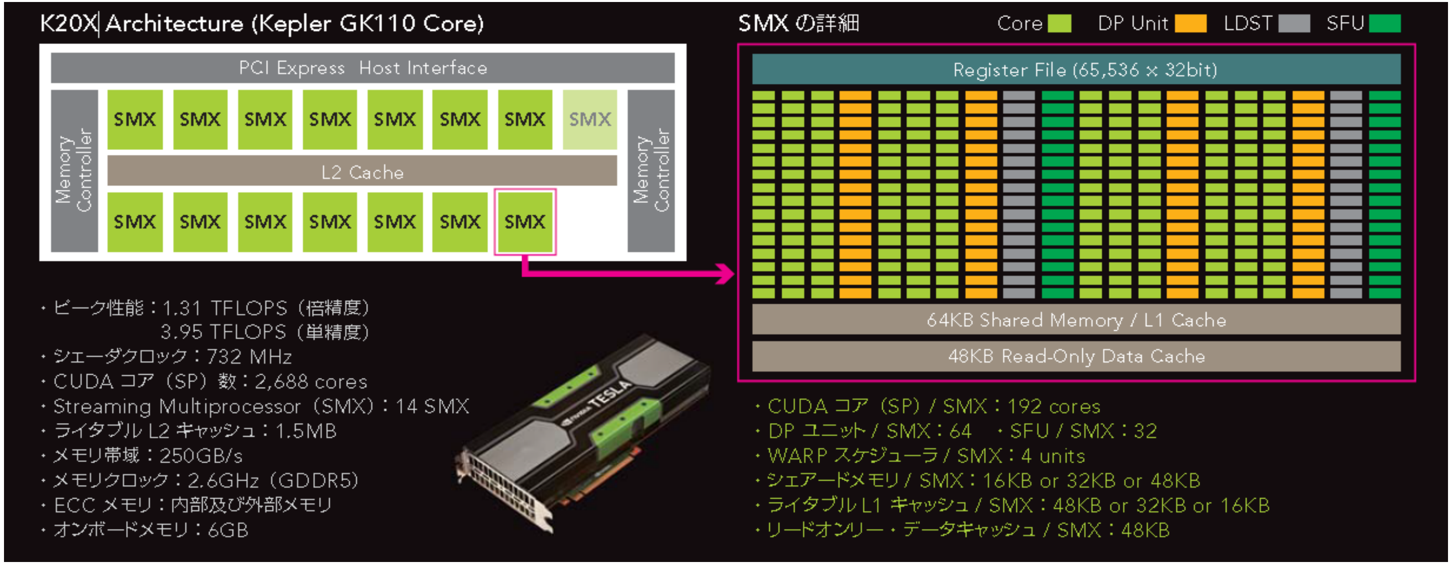
A computational domain



Computational costs is **directly proportional to the number of particles.**

GPU Computing

■ **GPU (Graphics Processing Units)**



■ **Multi-GPU computing**

Effective methods for multi-GPU computing should be considered.

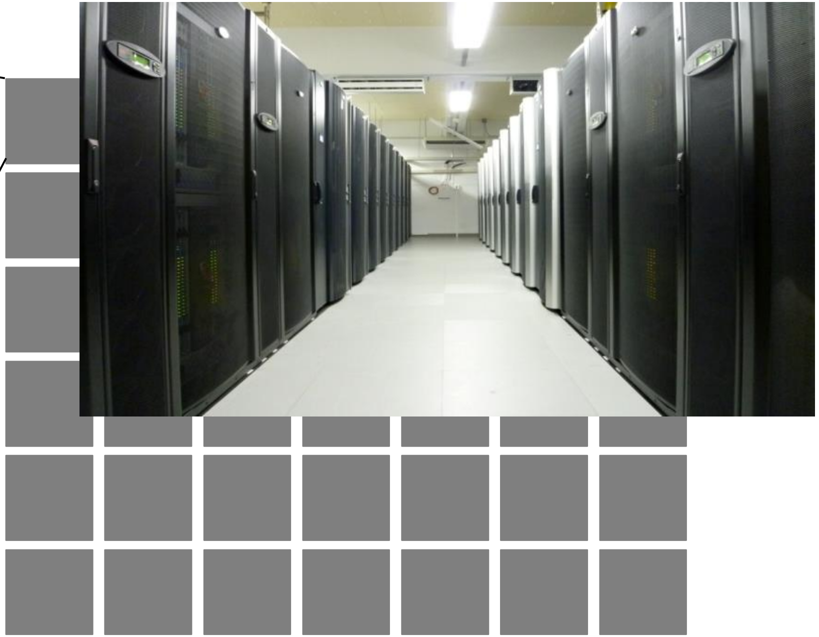
TSUBAME 2.5

Performance: 224.7 TFLOPS (CPU) ※ Turbo boost
5.562 PFLOPS (GPU)
Total: **5.7** PFLOPS (double precision)
17.1 PFLOPS (single precision)
Memory: 116 TB

Rack (30 nodes)
Performance: **122** TFLOPS
Memory: 2.28 TB

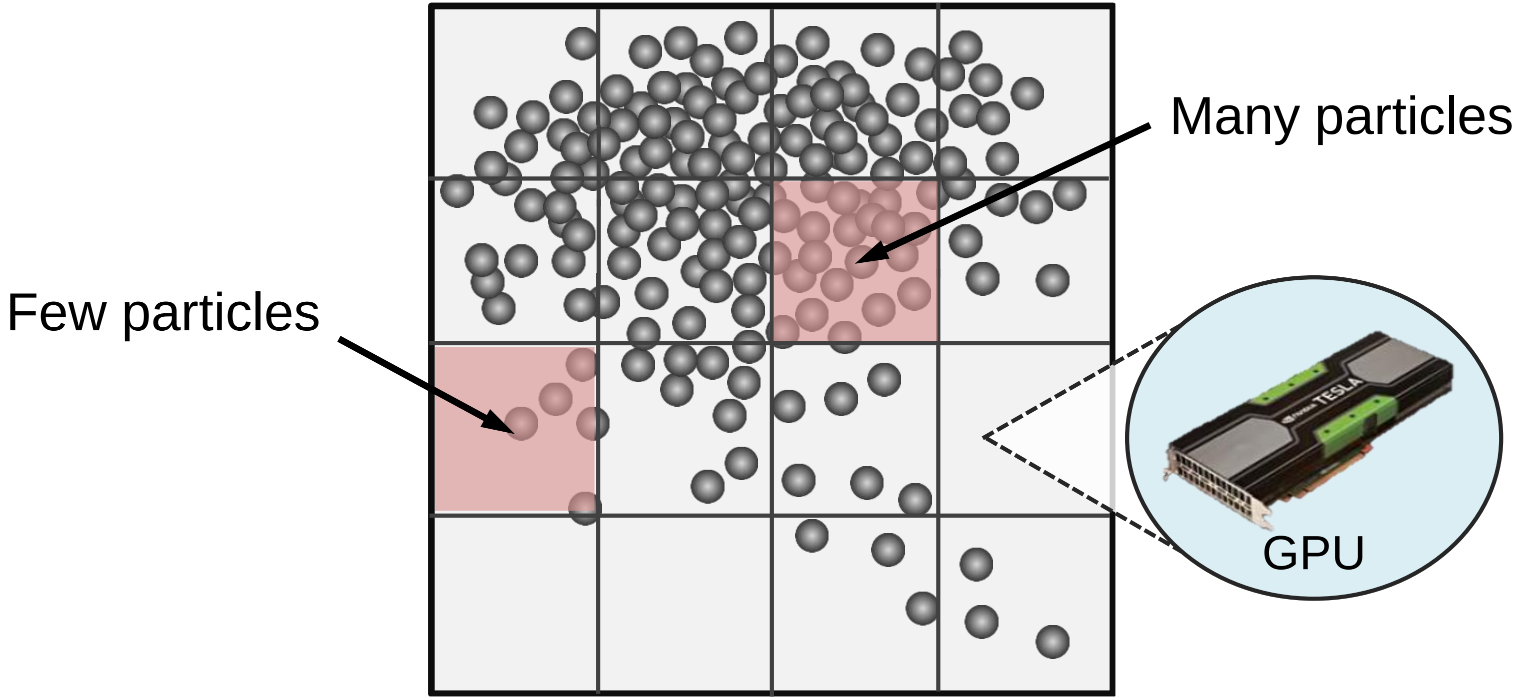
Compute Node
(2 CPUs, 3 GPUs)

Performance: **4.08** TFLOPS
Memory: 58.0GB(CPU)
+**18** GB(GPU)



Multi-GPU Computing

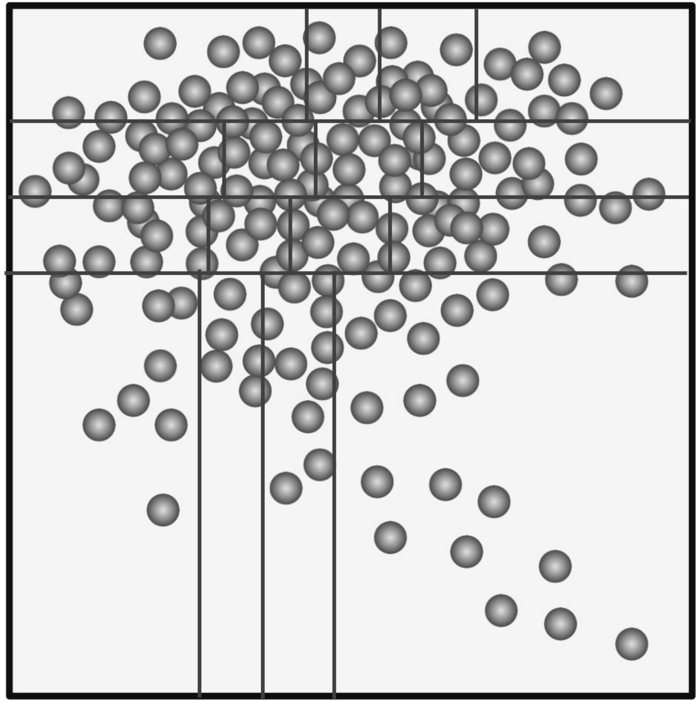
- **A load imbalance problem among subdomains**



Dynamic Load Balance

- **2-dimensional slice-grid method (2012)**

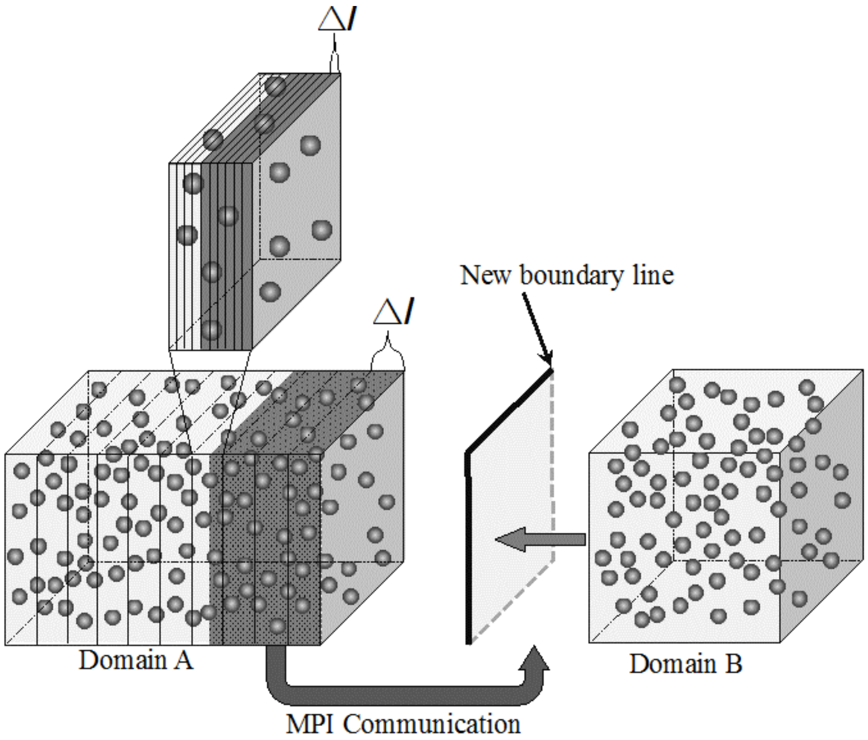
1. Boundary shift for **vertical** direction



2. Boundary shift for **horizontal** direction

Dynamic Load Balance

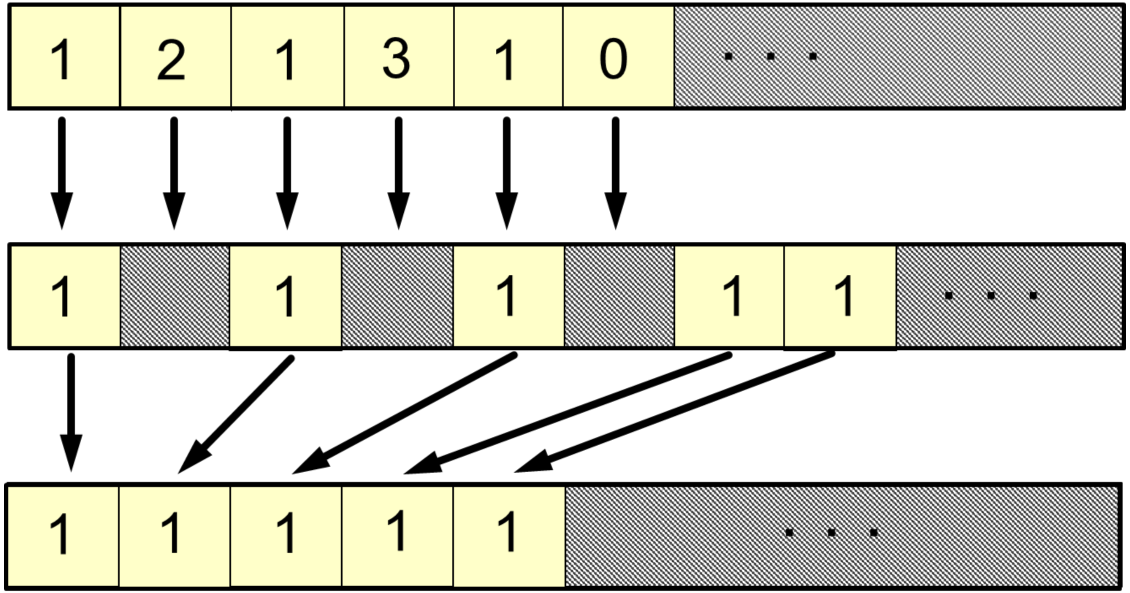
- Particle counting and boundary shift on GPU



Re-numbering

- **De-fragmentation of GPU memory**

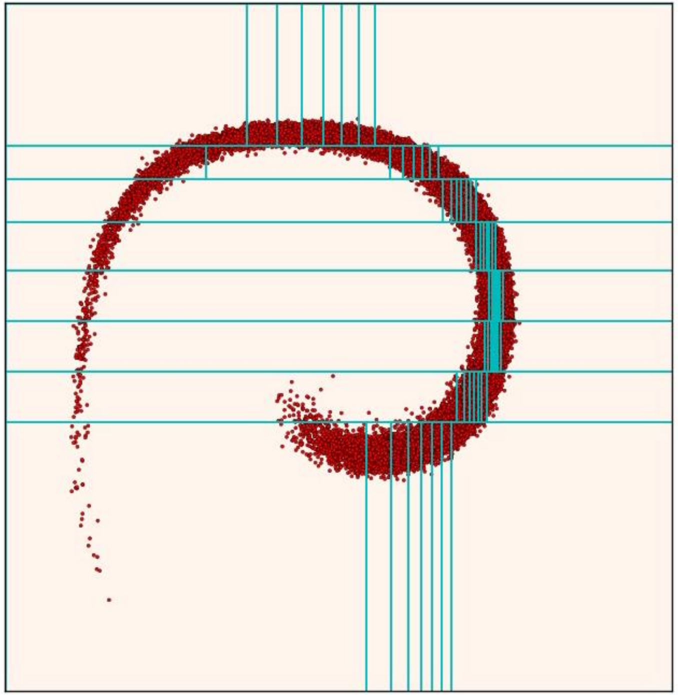
1. Inflows/outflows of particles
2. Re-numbering of particle data



✓ Each Number represents the domain ID.

Verification

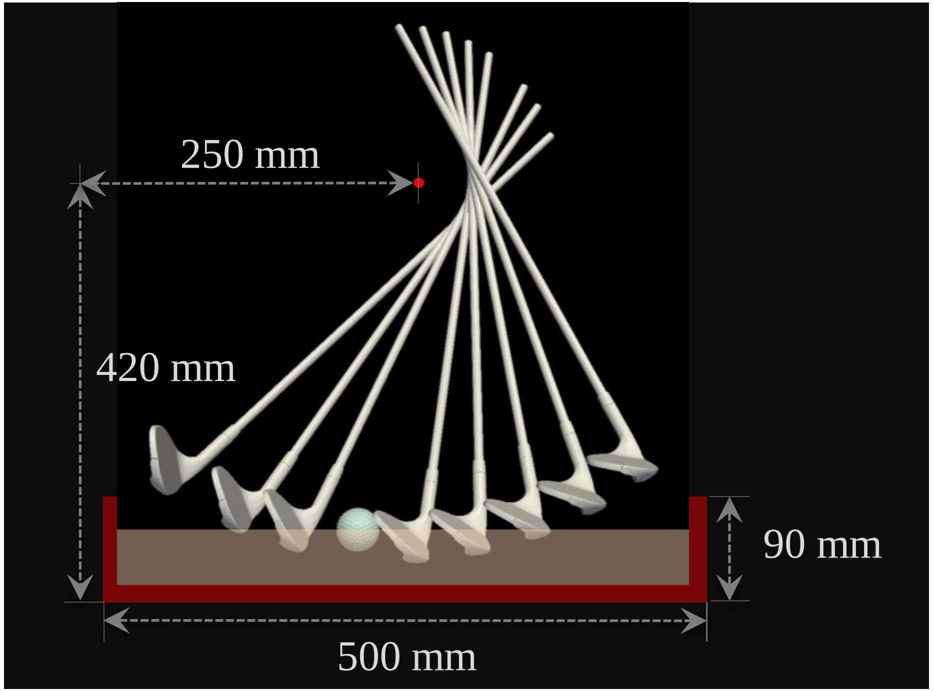
- The passive particles under vortex velocity field using 64 GPUs



An agitation simulation
using 64 GPUs (2013)

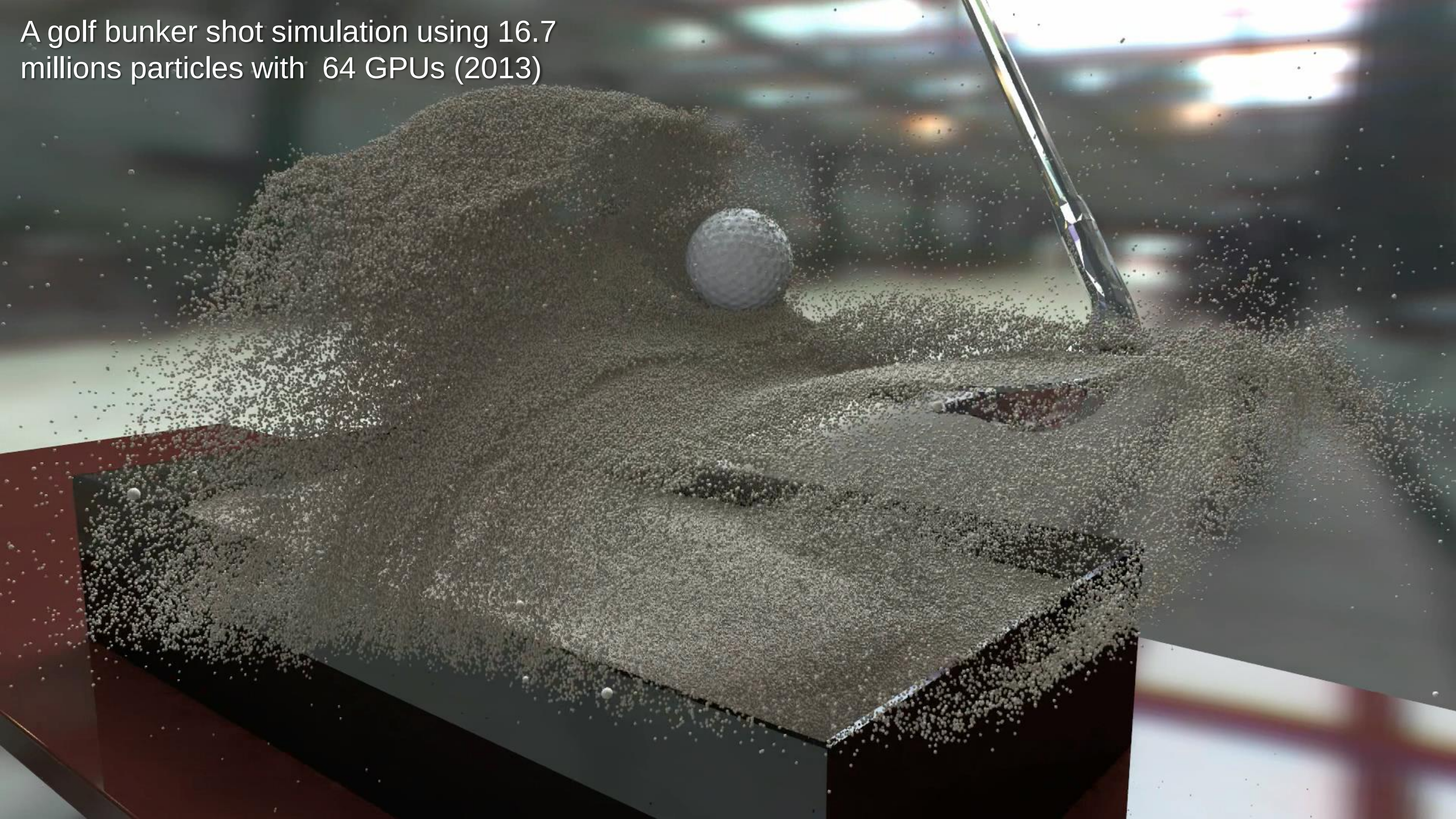
A Golf Bunker Shot Simulation

An arrangement of golf bunker shot simulation



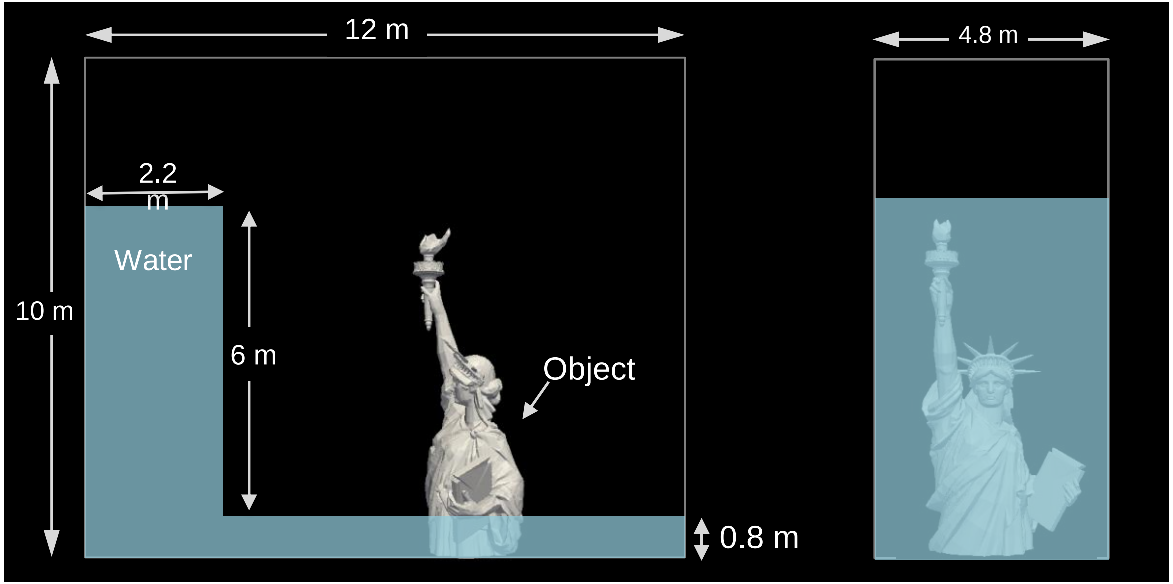
An example of physical conditions

Proper time	5.0×10^{-6}	[s]
Radius	0.4	[mm]
Mass	5.09×10^{-7}	[kg]
Young's module	2.8	[GPa]
Poisson ratio	0.17	
Friction co-efficient	0.3	
Number of particles	1.67×10^{7}	
Time steps	104000	


A golf bunker shot simulation using 16.7
millions particles with 64 GPUs (2013)

A Dam Break Simulation

- **Arrangement and physical conditions of incompressible flows**



Tab. 1: Conditions

g	[m/s^2]	9.8
ν	[m^2/s]	1.0×10^{-6}
ρ	[kg/m^3]	1.0×10^{3}
v_{max}	[m/s]	12
l_0	[m]	0.0125
C_h		2.6
γ		2
C_T		0.5
C_M	[m/s]	2/15


A dam break simulation using
72 M particles with 80 GPUs (2014)

Strong Scalability

- **Computational conditions**

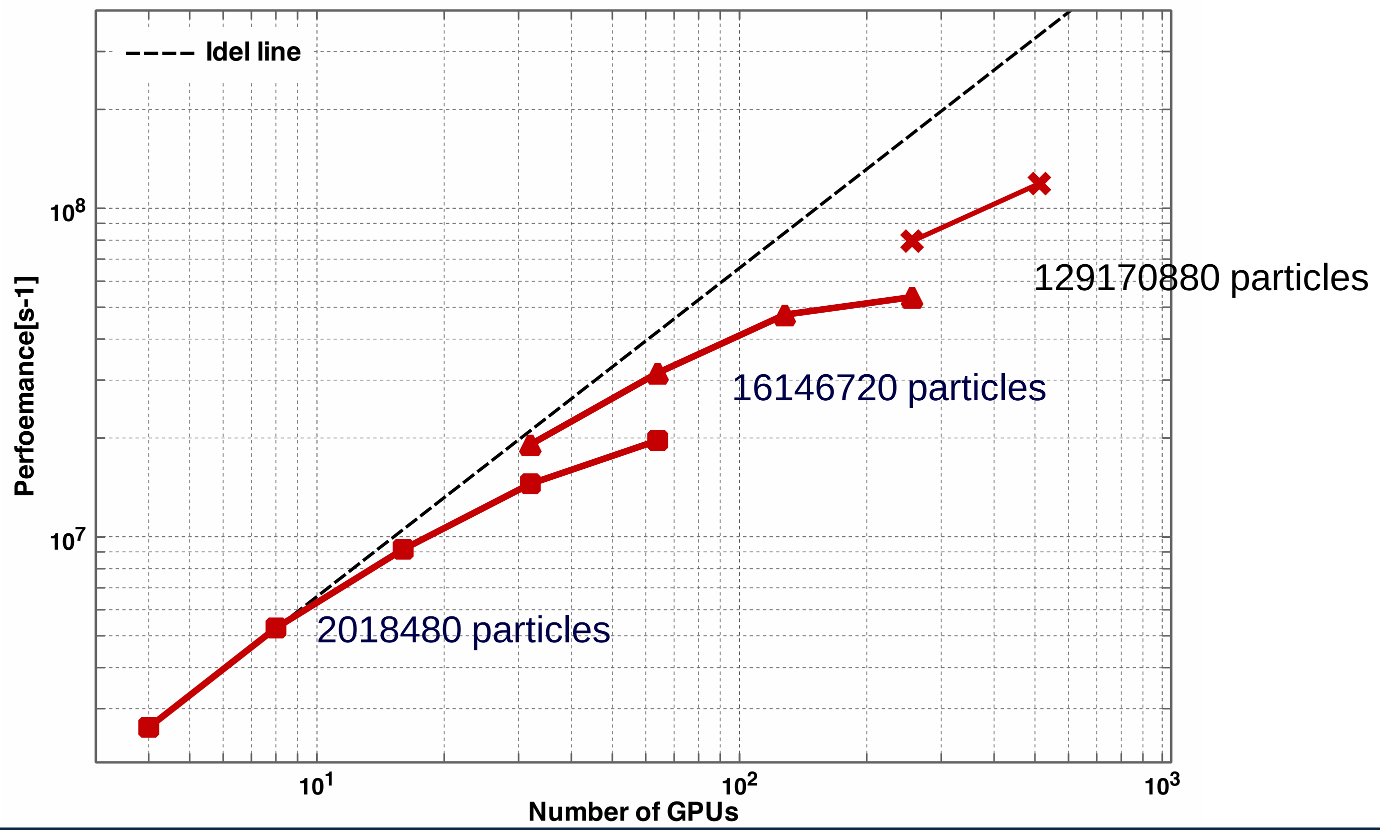
Number of particles	:	2×10^6, 1.6×10^7, 1.29×10^8 particles
Decomposition method	:	Slice-grid method with dynamic load balance
Time integration	:	2nd order Runnge-Kutta method
Problem	:	Agitation simulation

Allocate single GPU per each subdomains.

Define performance P as follows:

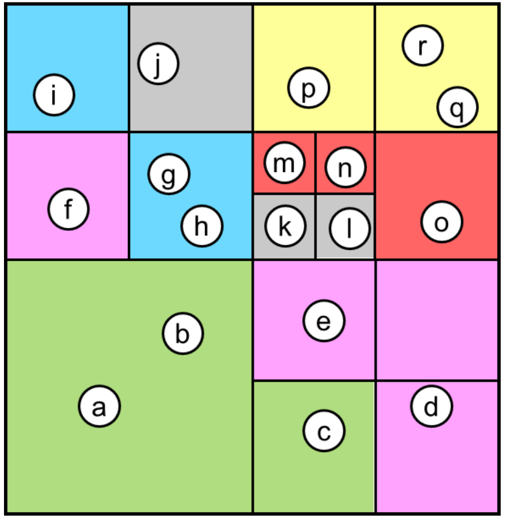
$$P = (\text{Computational time/steps})^{-1} \times \text{Number of particles}$$

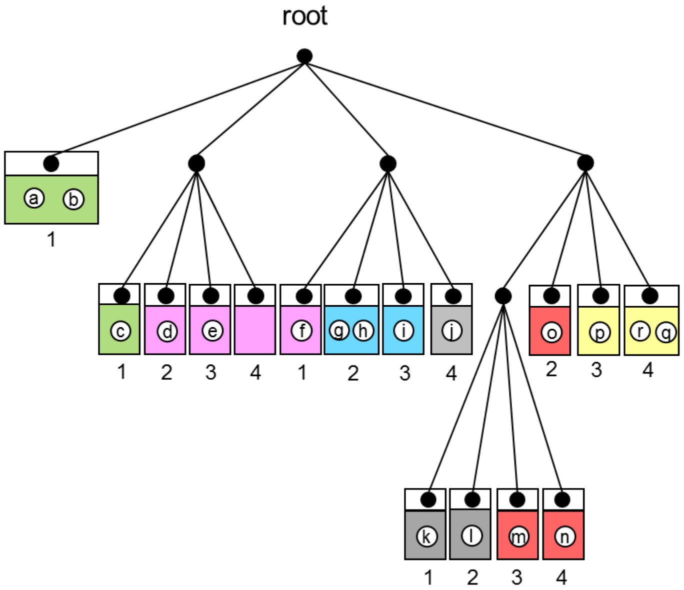
Strong Scalability



Space-Filling Curve

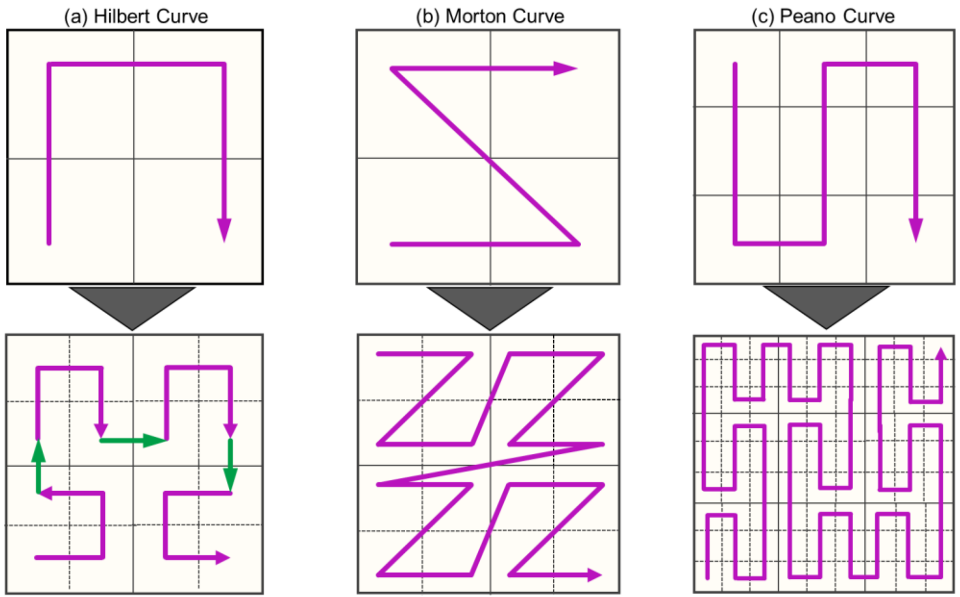
- ✓ A space-filling curve enables us to fill a multi-dimensional space with a continuous curve.
- ✓ A computational domain becomes recursively divided by a tree, and leafs are connected by using a space-filling curve so that each bundle of leafs has same number of particles.



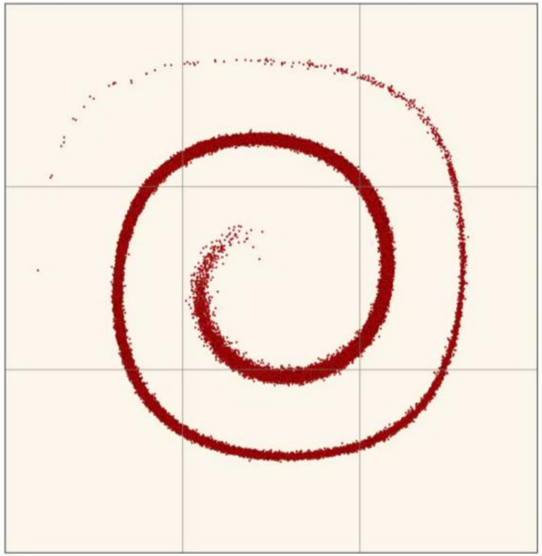


Space-Filling Curve

- ✓ A space-filling curve enables us to fill a multi-dimensional space with a continuous curve.
- ✓ A computational domain becomes recursively divided by a tree, and leafs are connected by using a space-filling curve so that each bundle of leafs has same number of particles.



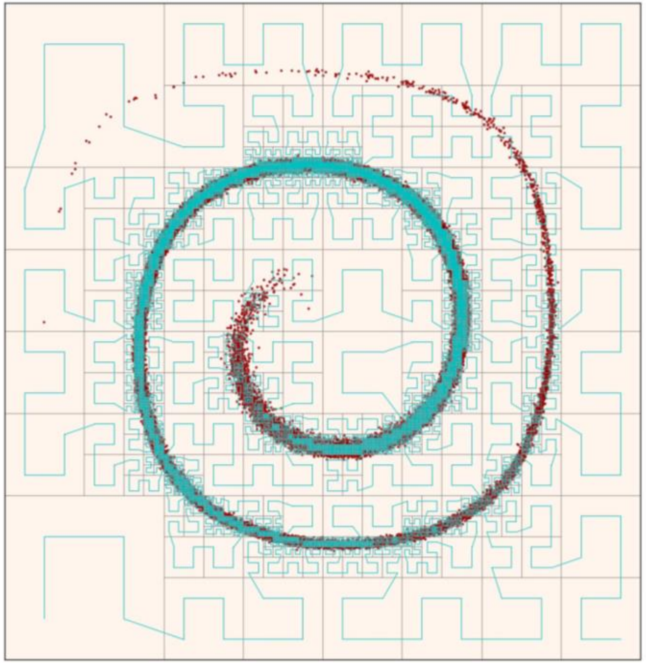
A recursive tree construction



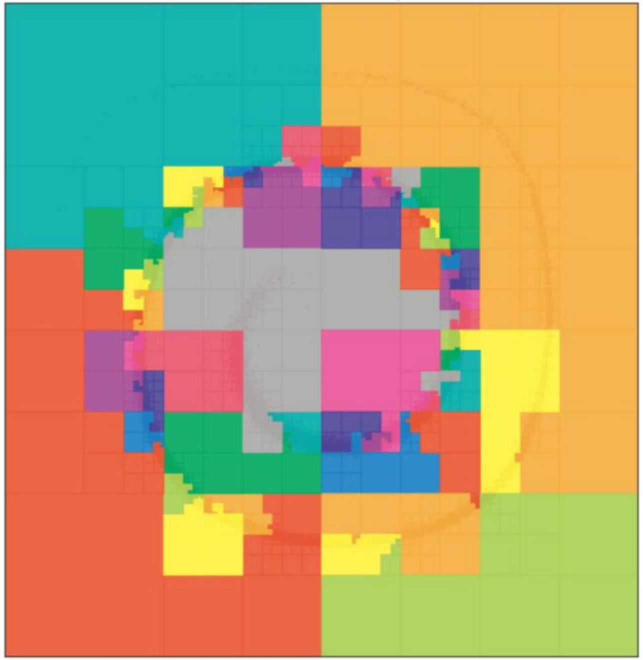
Hilbert Curve

- A computational domain becomes recursively divided into 2x2 sub-leafs by a quad-tree.
- The Hilbert curve shows good connectivity among sub-leafs.

A connection of leafs



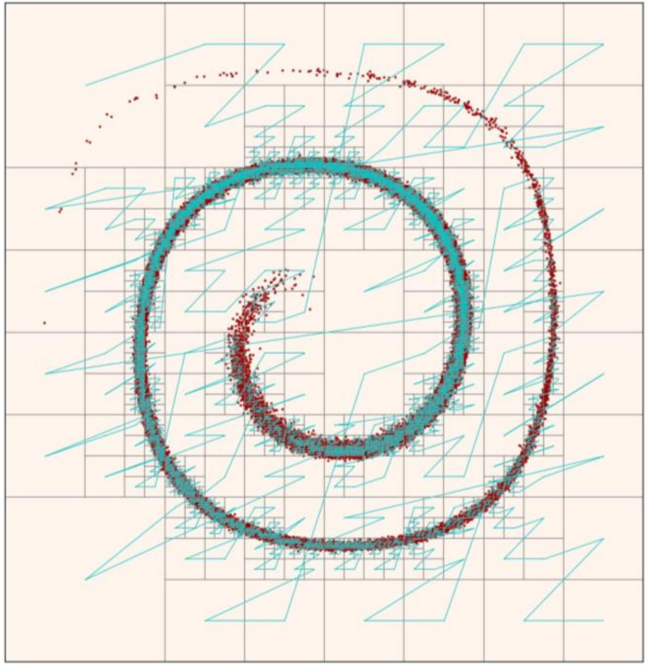
A domain decomposition



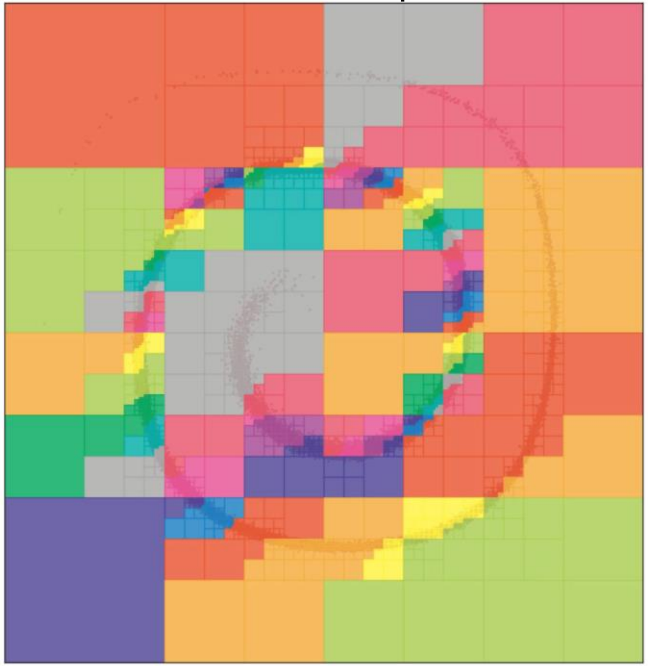
Morton Curve

- A computational domain becomes recursively divided into 2x2 sub-leafs by a quad-tree.
- The Morton curve (Z-curve) often causes large jumps between sequential leafs.

A connection of leafs



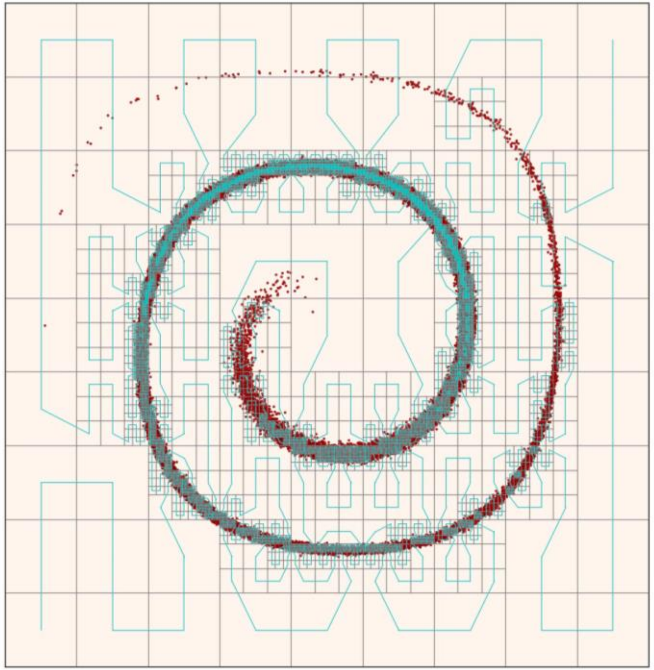
A domain decomposition



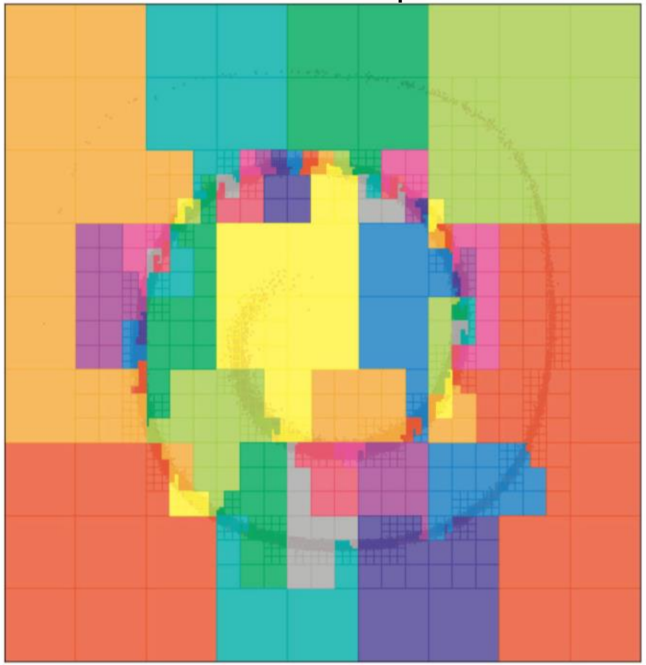
Peano Curve

- A computational domain becomes recursively divided into 3x3 sub-leafs by a nona-tree.
- The Peano curve shows high locality because it divides the leaf into 3x3 sub-leafs.

A connection of leafs



A domain decomposition



Weak Scalability

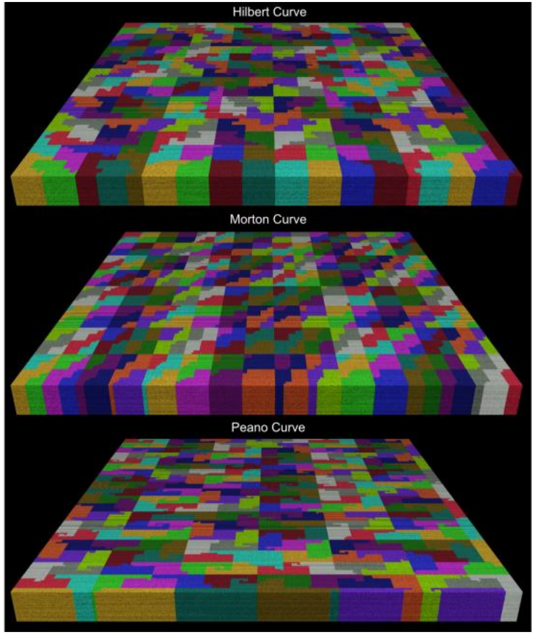
■ Computational conditions

Conditions	Values
particles / (GPU)	441.535
total number of GPUs	256
total number of particles	1.7 M (million), 4 M, 111 M particles
problem size / (GPU)	9.0 m × 8.0 m × 10 m
Supercomputer	TSUBAME2.5 (Tokyo Tech, JAPAN)
CPU	XeonX5670, 2Socket (12Core)
GPU	NVIDIA Tesla K20X GPUs

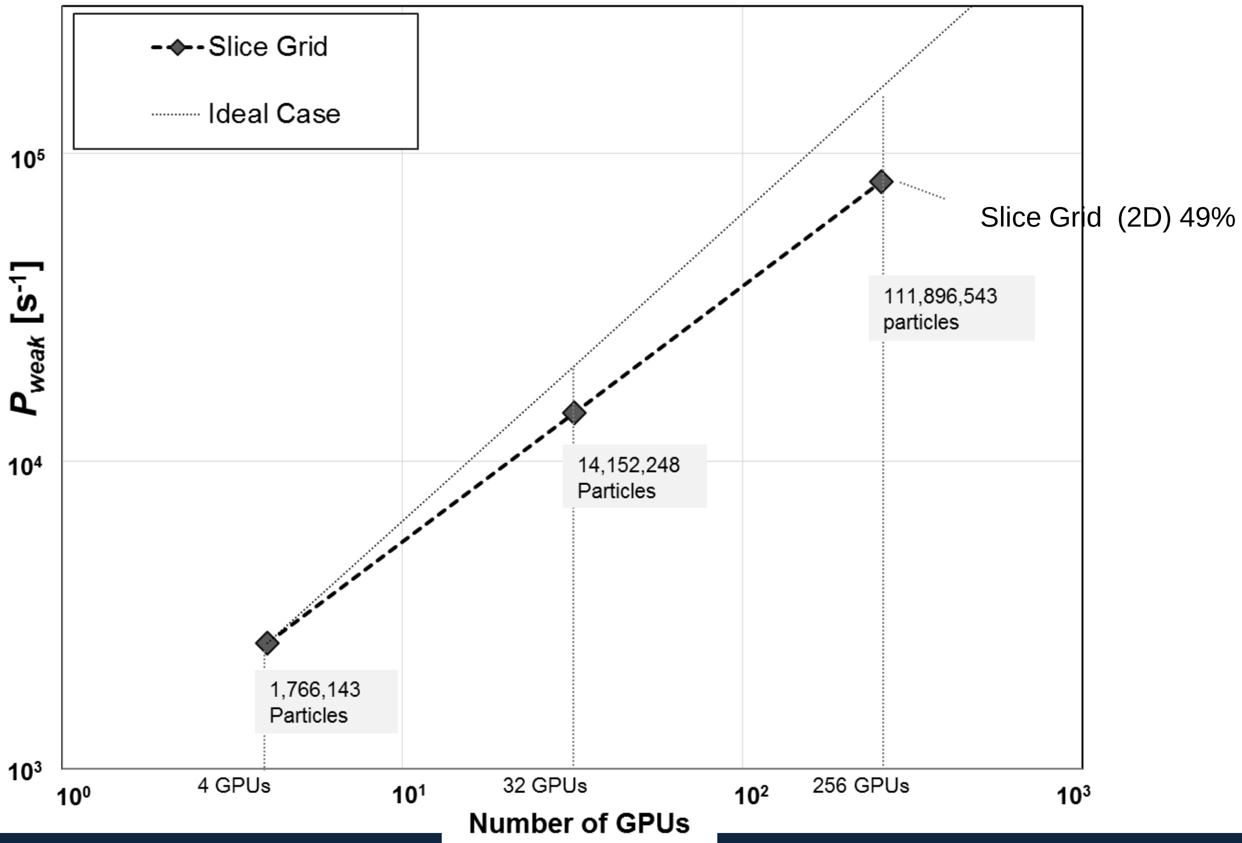
- Allocate single GPU per each subdomains.
- Define performance P as follows:

$$\boxed{P = (\text{computational time/steps})^{-1} \times \text{number of particles}}$$

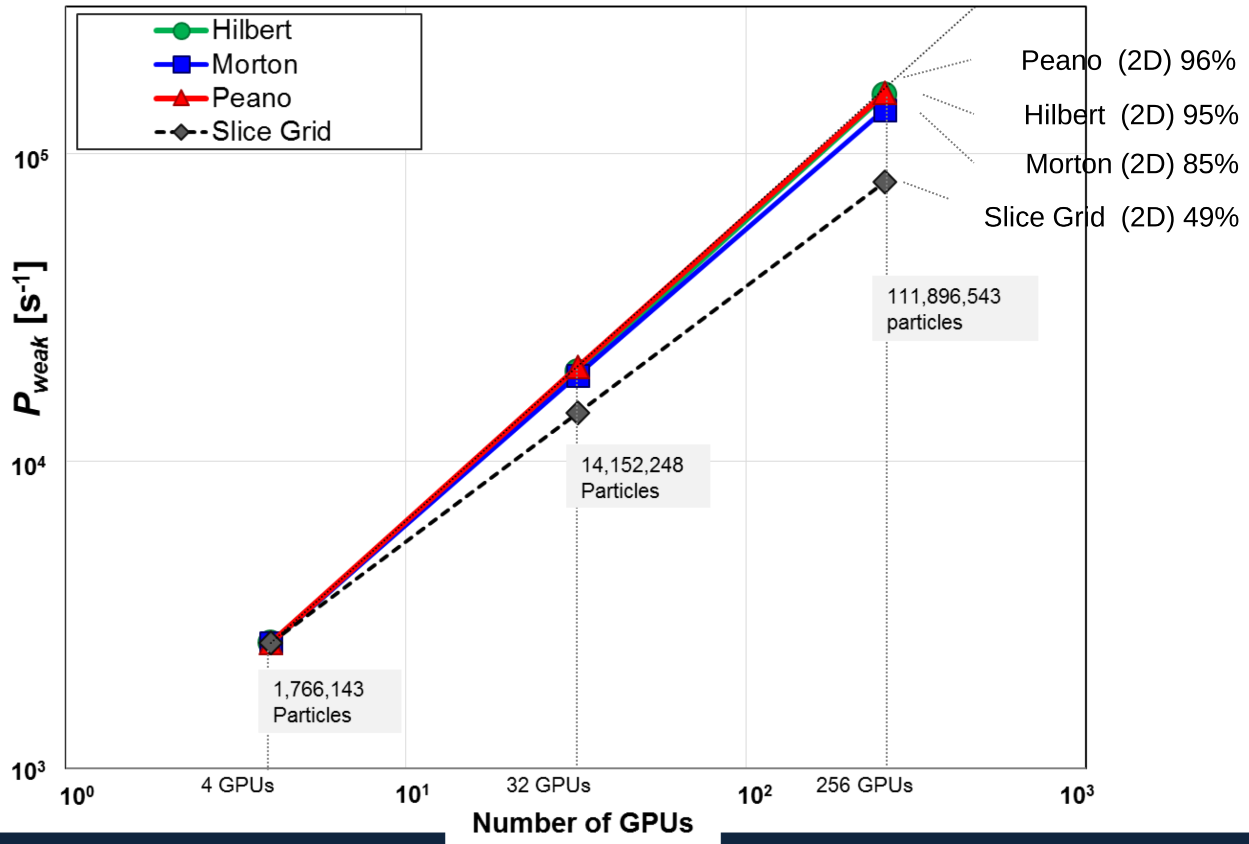
- Snapshots of the initial domain decomposition



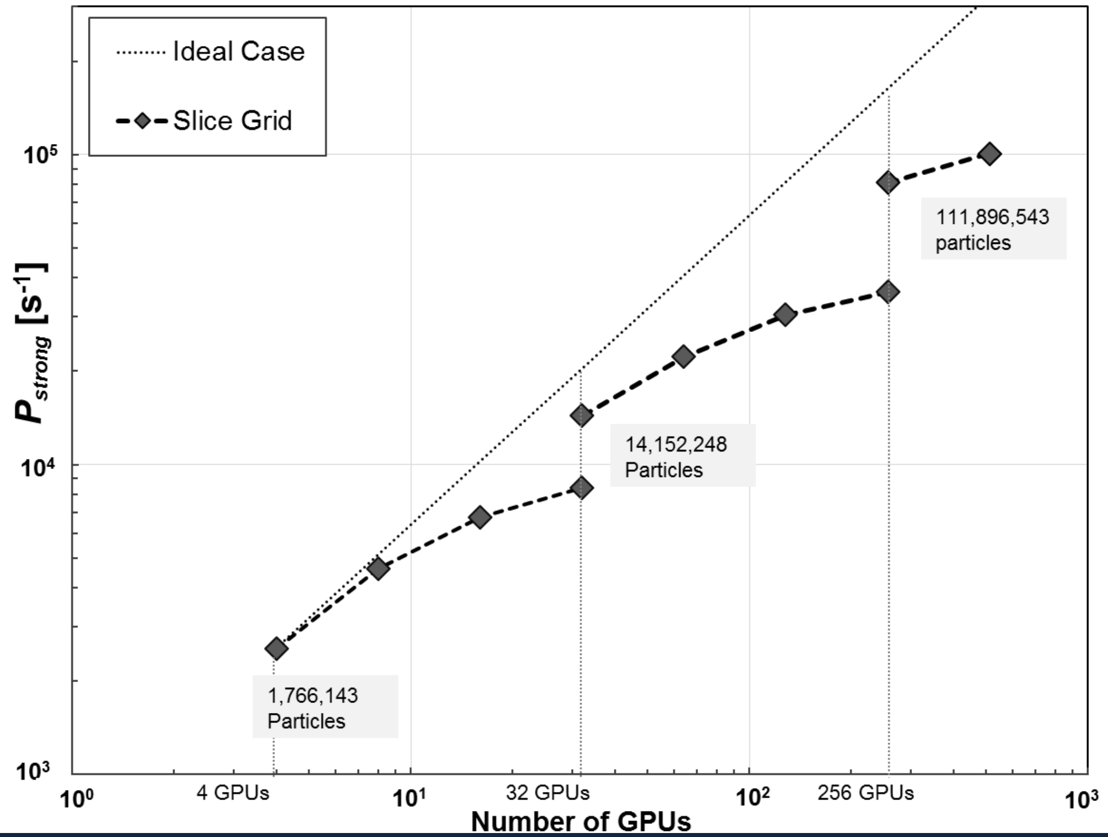
Weak Scalability



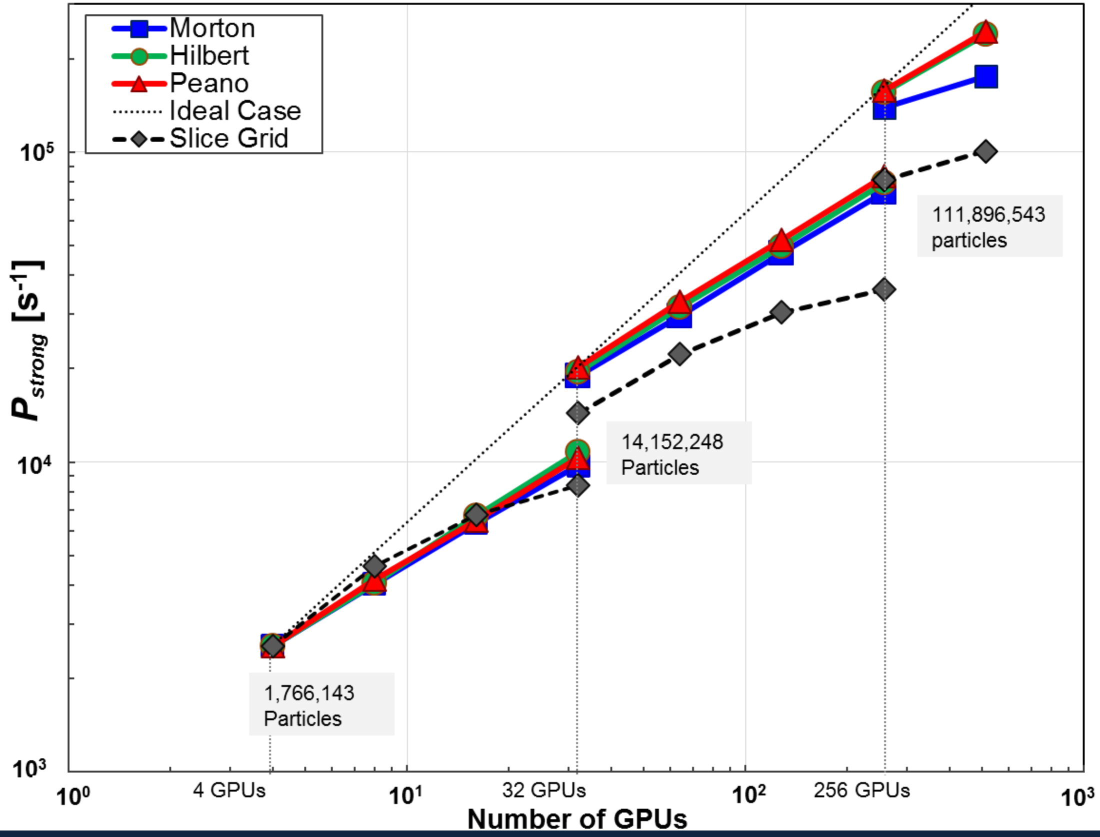
Weak Scalability



Strong Scalability

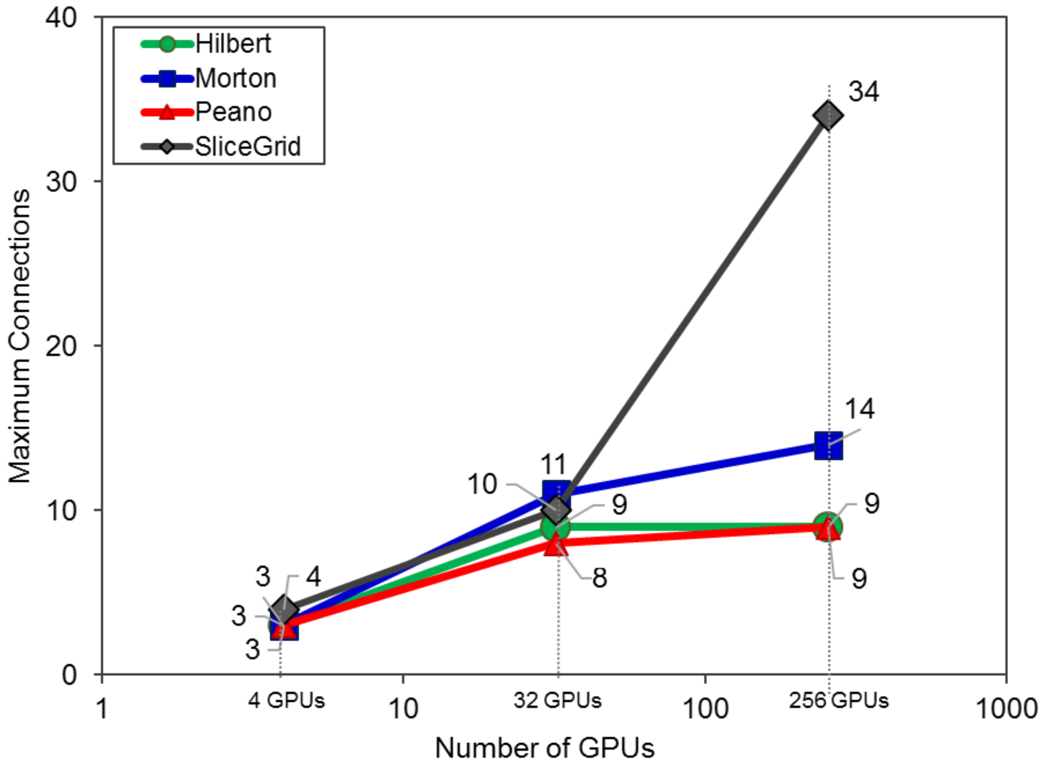


Strong Scalability

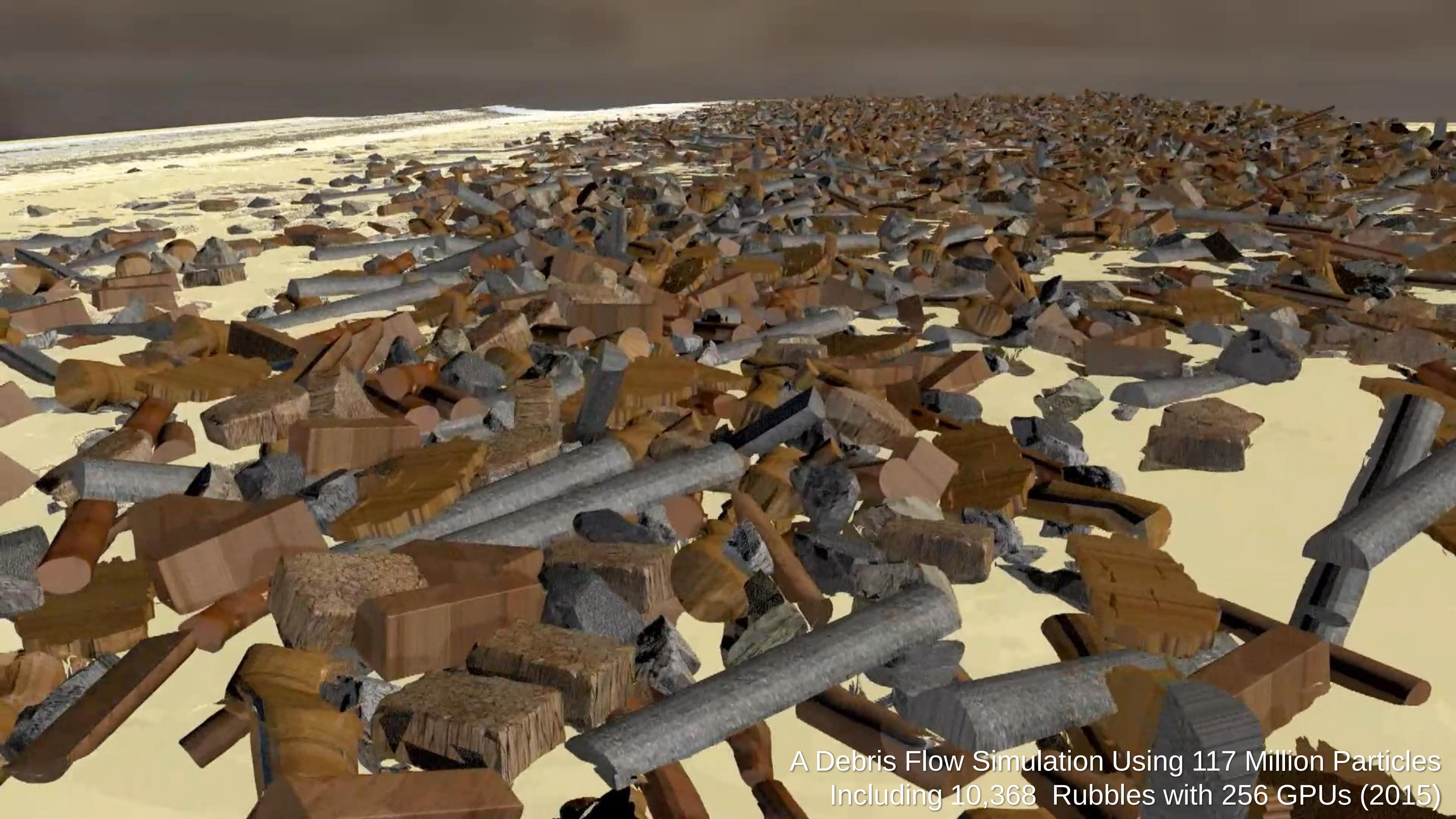


Connectivity of Sub-domains

- Maximum number of connections among neighboring subdomains.



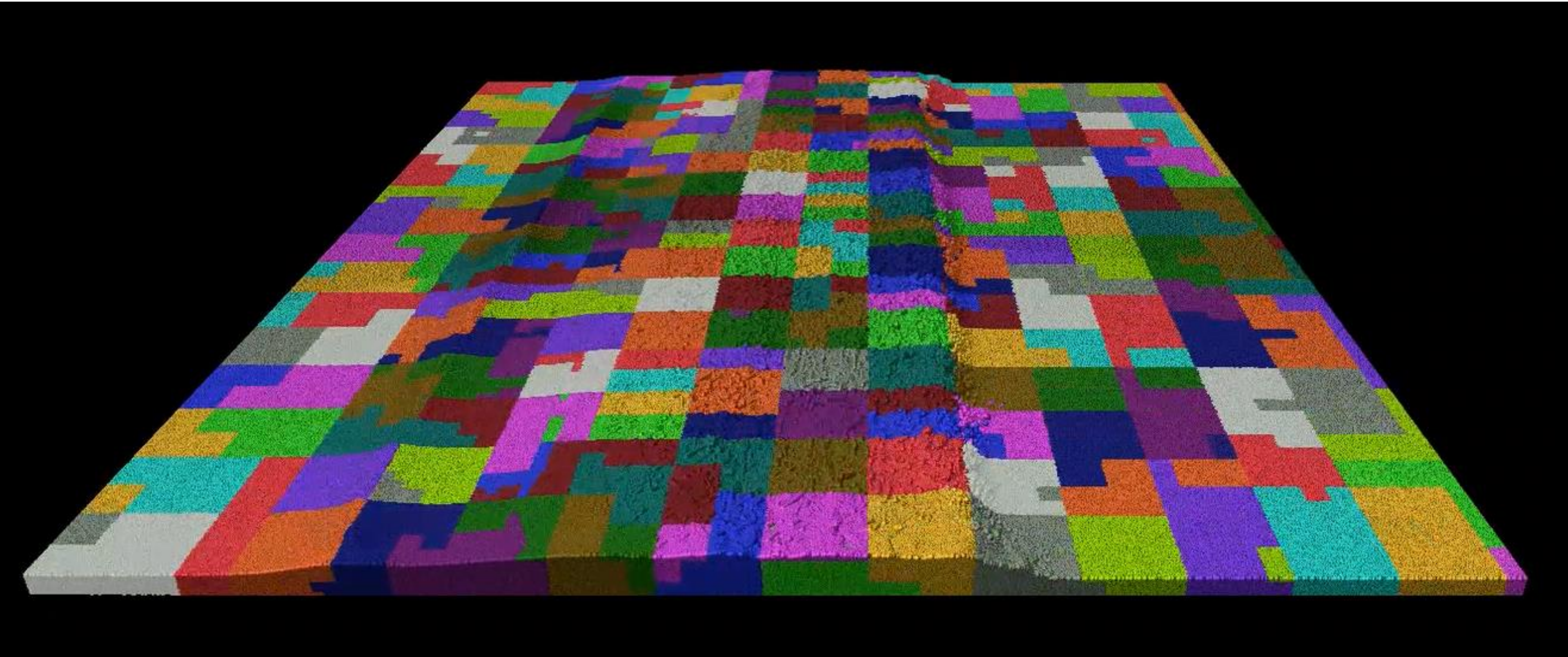
A Tsunami Simulation Using 20 Million Particles Including 945 Rubbles with 256 GPUs (2015)



A Debris Flow Simulation Using 117 Million Particles
Including 10,368 Rubbles with 256 GPUs (2015)

Dynamic Load Balance

- Dynamic load balance with Hilbert Curves (All the 117M particles are visualized.)



Conclusion

We succeeded in realizing large-scale (DEM/SPH) simulations with using from 10^7 to 10^8 particles by applying effective methods of dynamic load balance among GPUs based on the slice-grid method / space-filling curves on the TSUBAME2.5 supercomputer.

- ✓ The weak and strong scalability of a test case SPH simulation are examined on the multi-GPU system. It is found that the performance keeps improving in proportion to the number of GPUs when using space-filling curves.
- ✓ A realistic large-scale tsunami simulation with 7 buildings and floating driftwoods was successfully carried out running on 256 GPUs on TSUBAME 2.5 at Tokyo-Tech.
- ✓ A debris flow simulation using 117 Million particles interacting with 10,638 floating rubbles was successfully executed running on 256 GPUs on TSUBAME 2.5 at Tokyo-Tech.