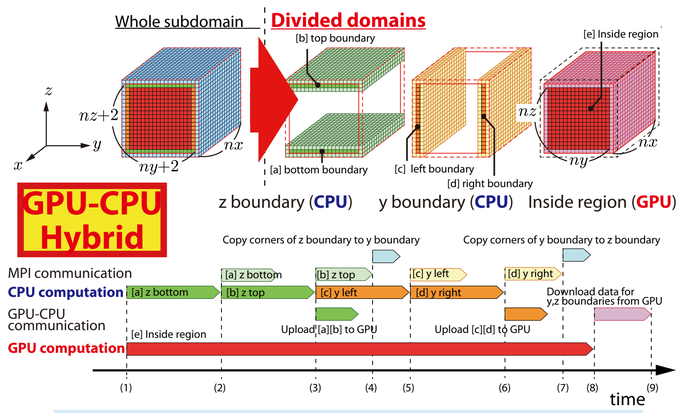
Peta-scale GPU Applications on TSUBAME

TOKYO TECH Pursuing Excellence

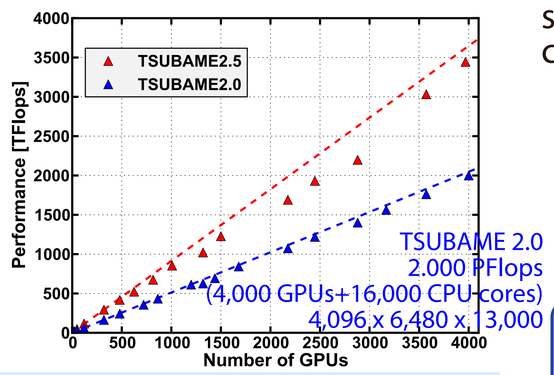
Phase-field simulation

The mechanical properties of metal materials largely depend on their intrinsic internal microstructures. The phase-field simulation is the most powerful method known to simulate the micro-scale dendritic growth during solidification in a binary alloy.



Scheme of the GPU-CPU Hybrid method

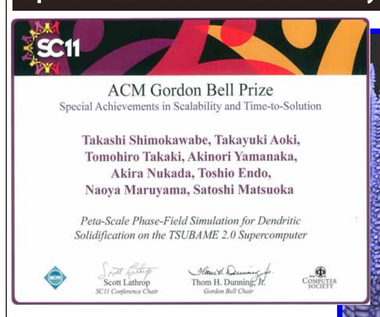
TSUBAME 2.5
3.406 PFlops (3,968 GPUs+15,872 CPU cores)
4,096 x 5,022 x 16,640

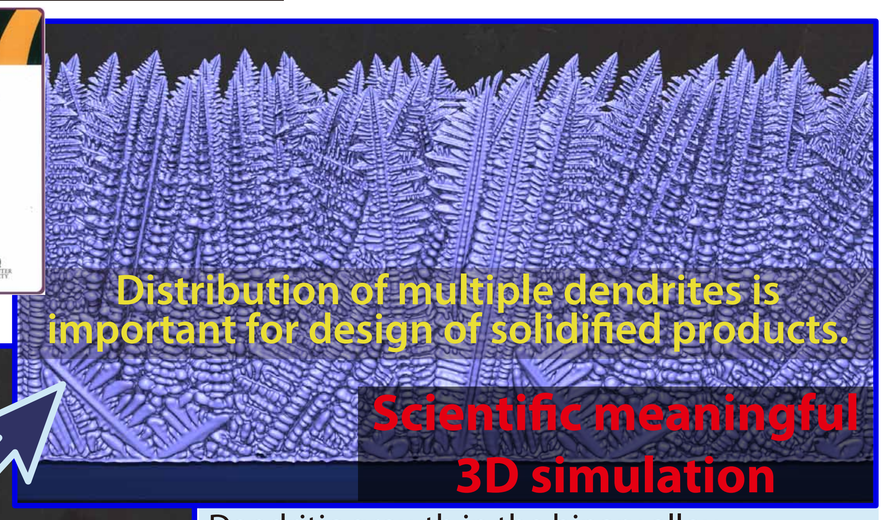


TSUBAME 2.0
2.000 PFlops
(4,000 GPUs+16,000 CPU-cores)
4,096 x 6,480 x 13,000

Weak scaling in single precision
Mesh size of a subdomain (1GPU + 4 CPU cores): 4096 x 162 x 130

2011 ACM Gordon Bell Prize
Special Achievements in Scalability and Time-to-Solution

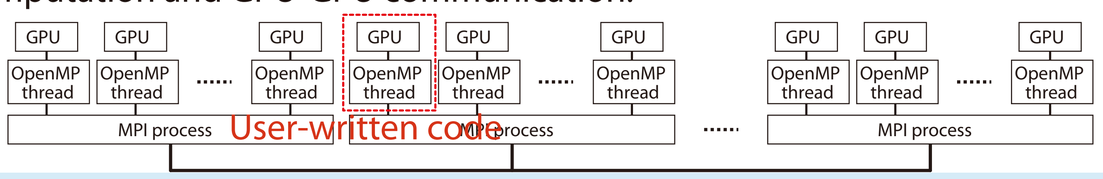




Dendritic growth in the binary alloy solidification with 4096 × 1024 × 4096 (768 GPUs of TSUBAME2.0)

Stencil comp. framework

A high-productivity framework for multi-GPU computation of mesh-based applications is proposed. Our framework automatically translates user-written functions that update a grid point and generates both GPU and CPU code. In order to execute user's code on multiple GPUs, the framework parallelizes this code by using MPI and OpenMP. The programmers write user's code just in the C++ language and can develop program code optimized for GPU supercomputers without introducing complicated optimizations for GPU computation and GPU-GPU communication.



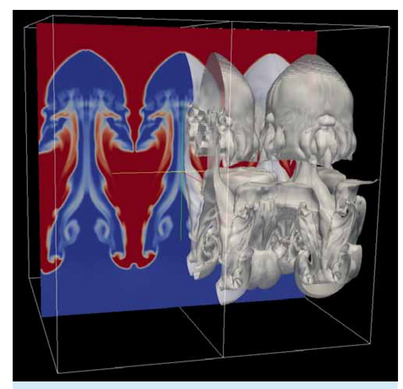
User-written code for a single GPU is parallelized by usiing MPI and OpenMP.

User-written function that update a grid point

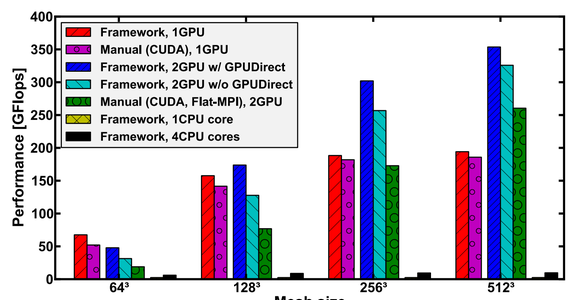
```
struct Diffusion3d {
  __host__ __device__
  void operator()(const ArrayIndex3D &idx,
    float ce, float cw, float cn, float cs,
    float ct, float cb, float cc,
    const float *f, float *fn) {
    fn[idx.ix()] = cc*f[idx.ix()]
    +ce*f[idx.ix<1,0,0>()]+cw*f[idx.ix<-1,0,0>()]
    +cn*f[idx.ix<0,1,0>()]+cs*f[idx.ix<0,-1,0>()]
    +ct*f[idx.ix<0,0,1>()]+cb*f[idx.ix<0,0,-1>()];
  }
};
```

User-written function is run over the grids

```
Loop3D loop3d(nx+2*mgnx, mgnx, mgnx,
              ny+2*mgny, mgny, mgny,
              nz+2*mgnz, mgnz, mgnz);
loop3d.run(Diffusion3d(), ce, cw, cn, cs,
           ct, cb, cc, f, fn);
```



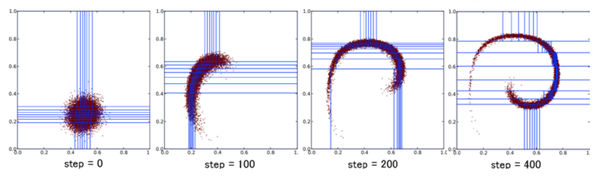
Simulation results of the Rayleigh-Taylor instability using our framework.



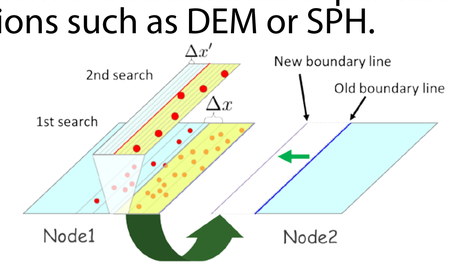
Performance of diffusion computation is obtained by the proposed framework and manual implementation. The framework can utilize GPUDirect for two-GPU computation within a node, which improves the peformance.

Particle simulation

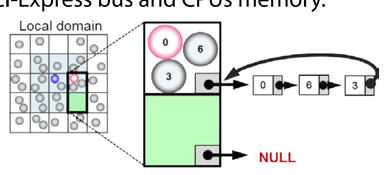
The distinct element method (DEM) is used for numerical simulations of granular mechanics. Each particle collides with the contacting particles. In order to bring the simulation closer to the real phenomena for the purpose of quantitative studies, it is necessary to execute large-scale DEM simulations on modern high-performance supercomputers. In this study, we propose an efficient method to realize the dynamic load balance of particle simulations based on short-range interactions such as DEM or SPH.



- By applying the slice-grid method to our particle simulation, we maintain the same number of particles in each domain.
- Frequency of de-fragmentation of GPU memory is optimized.



- The data of the particles moving to the neighbor subdomains are copied through PCI-Express bus and CPUs memory.

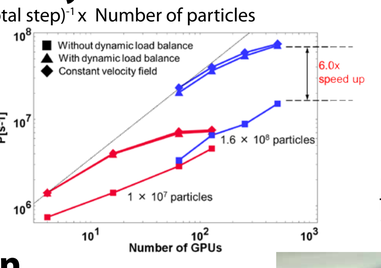


- We introduce the linked-list method for the neighbor particle list to save the memory drastically.

Performance scalability on TSUBAME 2.0

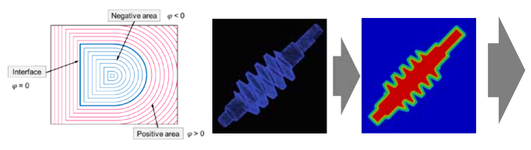
P = (Total computational time/Total step)$^{-1}$ x Number of particles

- Number of particles : 1.6×10^8
- Velocity field : Rotational velocity field
- Time integration : 4th Runge-Kutta method
- Initial condition : Gaussian distribution



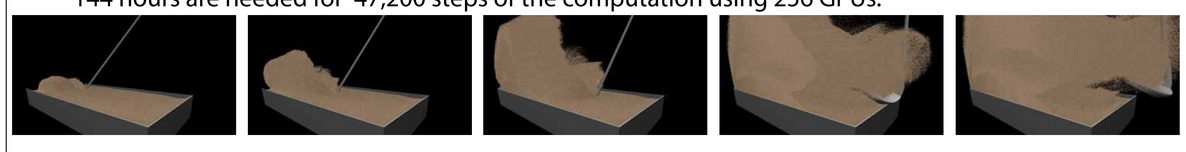
Convey simulation

- 64 GPUs are used.
- 4 million particles are used.
- Signed distance function is generated from CAD data and the zero iso-surface represent the CAD polygons.



Demonstration of 130 million particles golf-bunker shot

- Domain decomposition is executed every 10 steps.
- 144 hours are needed for 47,200 steps of the computation using 256 GPUs.



LES wind simulation

The lattice Boltzmann method (LBM) is a class of CFD methods that solve the discrete-velocity Boltzmann equation. LBM continuously accesses memory with a simple algorithm and is suitable for large-scale computations including complicated objects.

LES lattice Boltzmann method

$$f_i(x + c_i\Delta t, t + \Delta t) = f_i(x,t) - \frac{1}{\tau}(f_i(x,t) - f_i^{eq}(x,t))$$

Relaxation time and eddy viscosity

$$\tau = \frac{3\nu_*}{c^2\Delta t} + \frac{1}{2} \qquad \nu_* = \nu_0 + \nu_{SGS}$$

Coherent-structure Smagorinsky model

$$\nu_{SGS} = C|F_{CS}|^{\frac{3}{2}}\Delta^2|S| \qquad F_{CS} = \frac{Q}{E}$$

local memoty access

$$Q = -\frac{1}{2}\frac{\partial\bar{u}_j}{\partial x_i}\frac{\partial\bar{u}_i}{\partial x_j} \qquad E = \frac{1}{2}\left(\frac{\partial\bar{u}_j}{\partial x_i}\right)^2$$

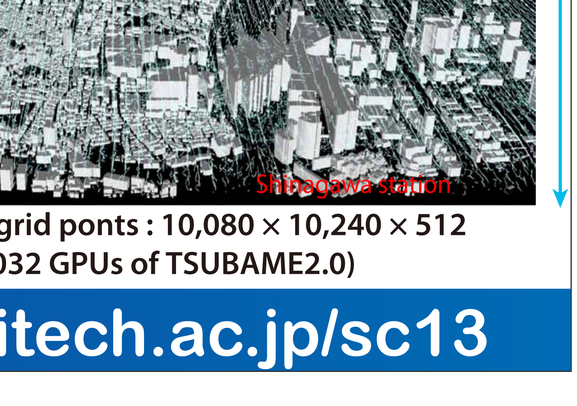
Discrete velocity

D3Q19 model

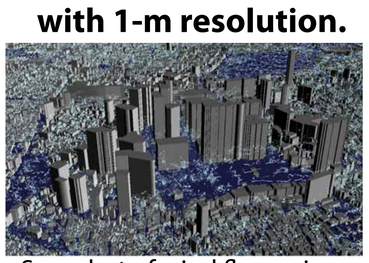
Overlapping technique

Weak scaling in single precision.

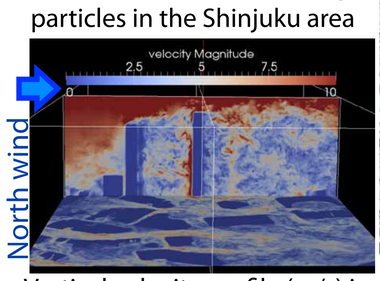
Mesh size of a subdomain: 192 x 256 x 256/GPU



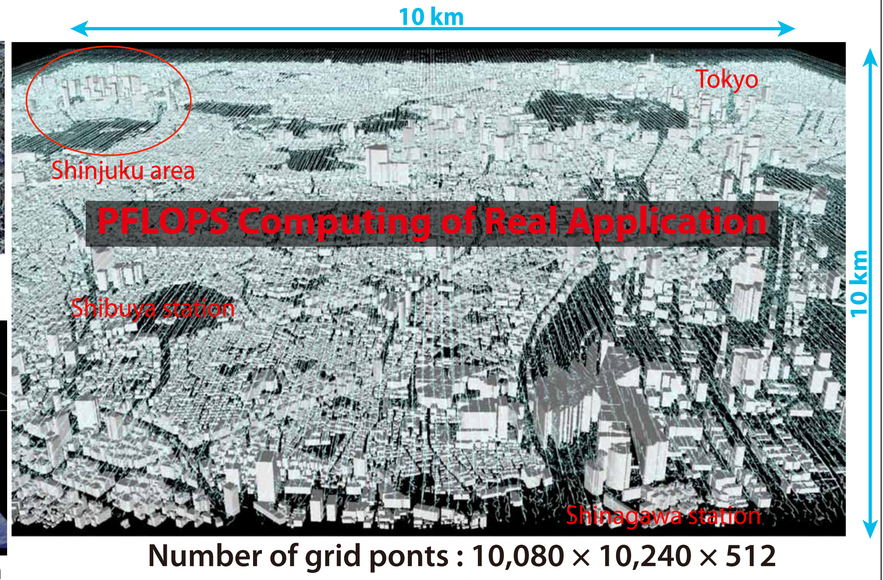
Large-scale LES wind simulation for 10 km x 10 km area in metropolitan Tokyo with 1-m resolution.



Snapshot of wind flow using particles in the Shinjuku area



Vertical velocity profile (m/s) in the Shinjuku area (north is left).



Number of grid ponts : 10,080 × 10,240 × 512
(4,032 GPUs of TSUBAME2.0)

http://www.gsic.titech.ac.jp/sc13