Large-scale granular simulations using Dynamic load balance on a GPU supercomputer

Satori Tsuzuki†, Takayuki Aoki††

AOKI Lab.

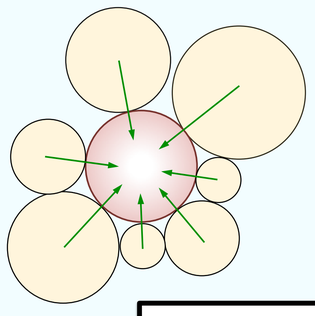
†Department of Energy Science, Tokyo Institute of Technology, ††Global Scientific Information and Computing Center, Tokyo Institute of Technology

Background

Discrete Element Method (DEM) is used for numerical simulations of granular mechanics [1]. Each particle collides with the contacting particles.

Real granular phenomena often consist of more than billions of stones or particles. Due to computational resources, a single particle represents more than 10 thousands particles in the previous simulations. 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.

We have developed a DEM simulation code for a supercomputer with a large number of GPUs. Spatial domain decomposition is a reasonable way for the DEM algorithm and suitable for computers with multiple GPU nodes. However, in the static domain decomposition, the spatial particle distribution changes in time and the computational load for each domain becomes quit unequal.



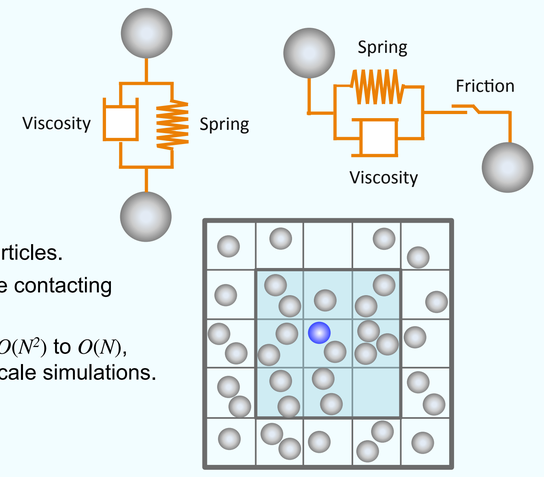
Discrete Element Method

- ✓ The particle interaction is modeled as a spring force proportional to the penetration depth of the contacting two particles[1]. The friction in the tangential direction is also taken into account. The force of the i-particle among the contacting N particles is described as follows,

$$m_i \ddot{x}_i = \sum_{j \neq i}^{N} (-k x_{ij} - \gamma \dot{x}_{ij}) \qquad (1)$$

i-particle experiences a summation of the forces from the contacting N particles.

- ✓ The neighbor-particle list is commonly used to reduce the cost to count the contacting particles and kept on the virtual mesh on the computational domain.
- ✓ By applying neighbor-particle list, we reduce the computational cost from $O(N^2)$ to $O(N)$, however the required memory often becomes a severe problem in large-scale simulations.



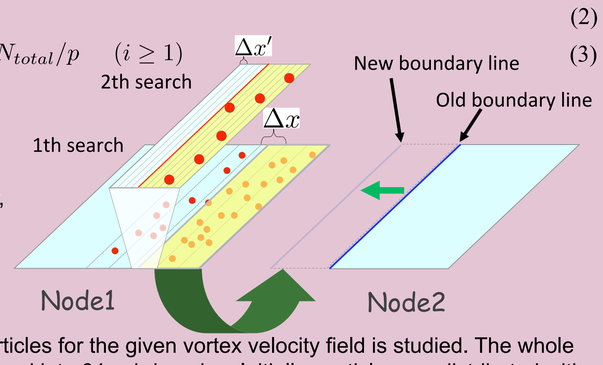
Dynamic Load Balance

- The whole computational domain is decomposed into several subdomains (local domain) and a GPU is assigned to compute particles in the subdomain.
- The 2-dimensional slice-grid method is introduced to maintain equal load balance among GPUs. Unlike the hierarchical based algorithm such as Orthogonal Recursive Bisection [2]. the horizontal and vertical boundary lines are shifted in turn
- The number of particles moving to neighbor subdomains across the boundary are counted by the following process iteratively:

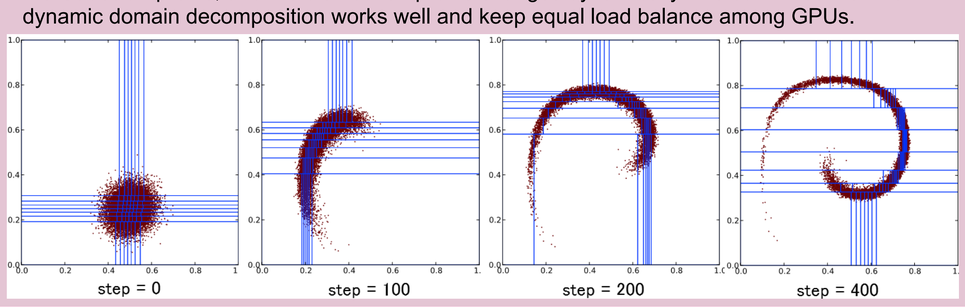
$$\Delta N_0 = N_0 - N_{total}/p \qquad (2)$$

$$\Delta N_i = \Delta N_{i-1} + N_i - N_{total}/p \quad (i \geq 1) \qquad (3)$$

- We divide the subdomain into a proper space Δx and count the number of particles within Δx.
- Such data as the position, velocity, mass, radius and so on of the particles moving to the neighbor subdomains are copied through PCI-Express bus

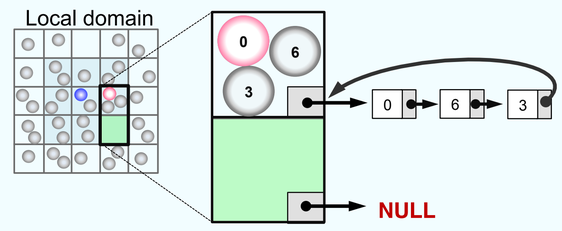


- A benchmark of passive scalar particles for the given vortex velocity field is studied. The whole computational domain is decomposed into 64 subdomains. Initially, particles are distributed with the Gaussian profile, and the distribution profile changes dynamically. It is confirmed that the dynamic domain decomposition works well and keep equal load balance among GPUs.



Linked-list Method

- ✓ A memory pointer is added to each particle data to have a reference to the next particle existing in the virtual mesh sequentially. [3].
- ✓ We can reduce the memory usage for the neighbor-particle list to 1/8.



Objectives

We propose an efficient method to realize large-scale particle simulations based on short-range interactions such as DEM on GPU supercomputers by using the follows:

- **Dynamic load balance among GPUs**
  By applying the slice-grid method to our particle simulation, we maintain the same number of particles in each domain.
- **De-fragmentation of GPU memory**
  Frequency of the memory de-fragmentation process is determined by taking into consideration of the trade-off of the data communication between CPU and GPU.
- **Linked-list method**
  We introduce the linked-list method for the neighbor particle list to save the memory drastically.

TSUBAME2.5 Overview

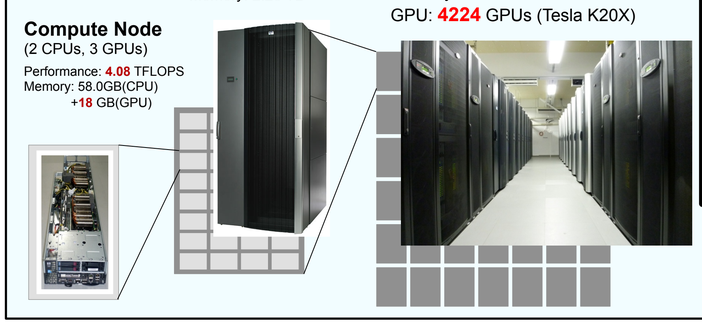
- ➢ TSUBAME is a supercomputer deployed by Global Scientific Information and Computing Center (GSIC) at Tokyo Institute of Technology, Japan.

Performance: 224.7 TFLOPS (CPU)
※Turbo boost
5.562 PFLOPS (GPU)

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

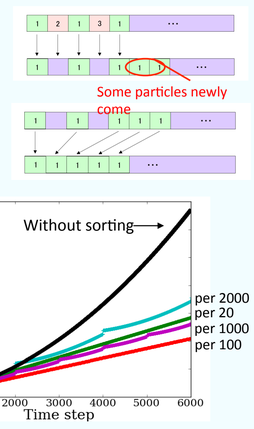
Total: **5.7** PFLOPS (double precision)
17.1 PFLOPS (single precision)
Memory: 116 TB
GPU: **4224** GPUs (Tesla K20X)

Compute Node
(2 CPUs, 3 GPUs)
Performance: **4.08** TFLOPS
Memory: 58.0GB(CPU)
+**18** GB(GPU)



De-fragmentation

- ✓ Fragmentation of GPU memory degrades the memory access and memory usage.
- ✓ De-fragmentation revives the performance to linear increase.
- ✓ The frequency of de-fragmentation is optimized.



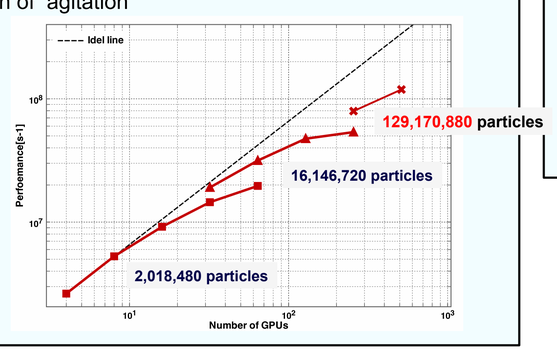
Performance scalability on TSUBAME 2.5

The strong and weak scaling for the computation of agitation simulation are measured on TSUBAME 2.5.

Problem	: Agitation simulation
Number of particles	: 2×10^6, 1.6×10^7, 1.29×10^8
Time integration	: 2nd order Runnge-Kutta
Decomposition	: 2D slice grid method
GPUs	: 512 GPUs

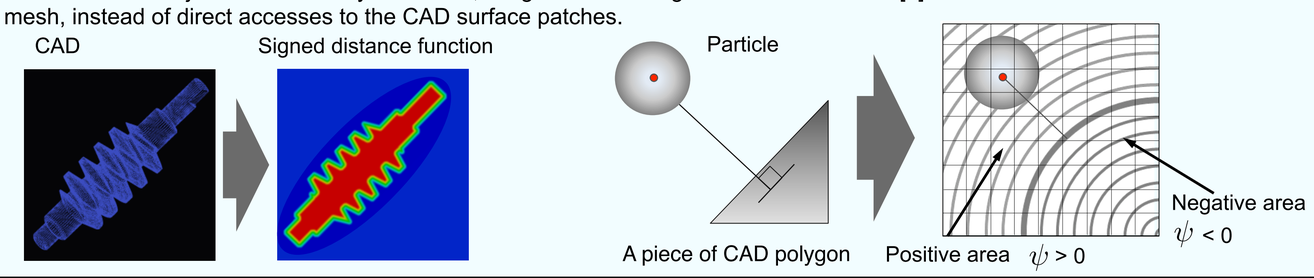
Allocate single GPU per each subdomain.
Define performance P as follows:

P = (Total computational time / Total step)$^{-1}$ × Particle number



Level-Set method

- ✓ To interact with objects described by CAD data, we generate the signed-distance function[4] on a uniform mesh, instead of direct accesses to the CAD surface patches.

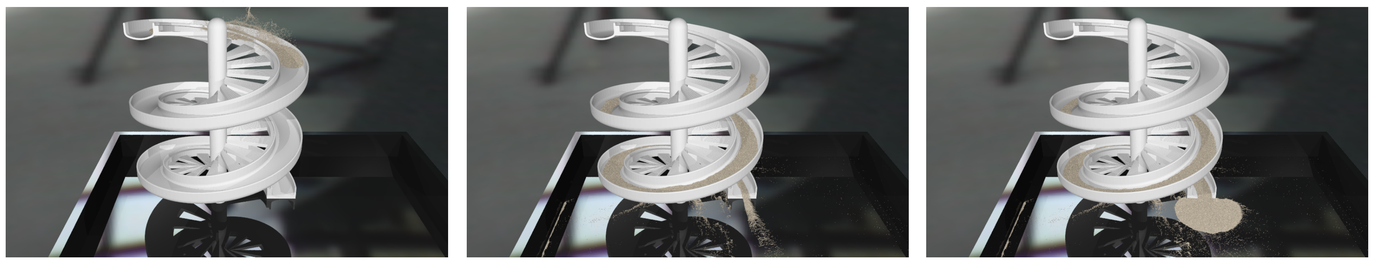


Agitation/Convey analysis on TSUBAME 2.5

- ✓ Simulations with 4M particles run on 64 GPUs for 200,000 time steps.

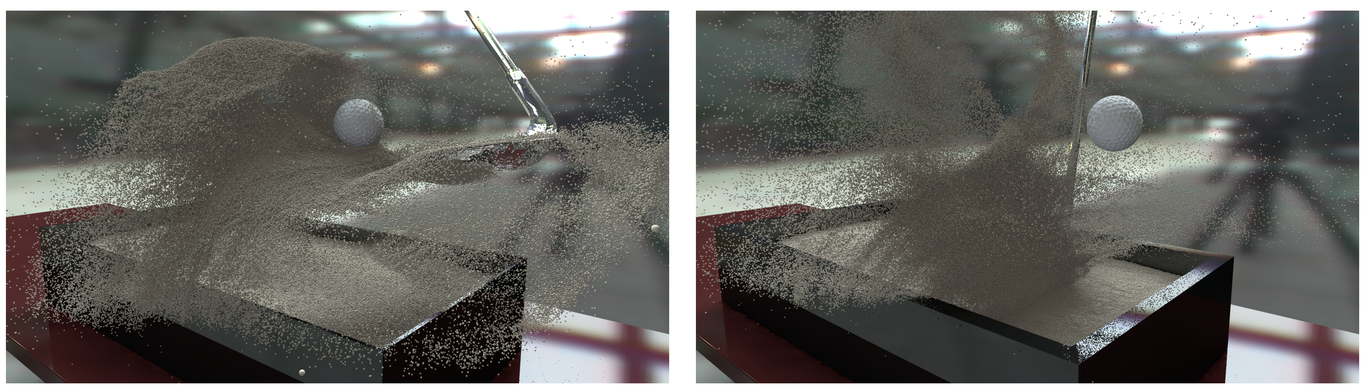
Spiral Slider simulation

- ✓ 4 million particles with 32 GPUs are used for the 201000 computational steps.



Demonstration of golf bunker shot

- ✓ It takes 144 hours to run the computation of 130M particles for 47,200 time steps on 256 GPUs.
- ✓ Simulations using 16.7M particles for various swing trajectories are carried out on 64 GPUs.



On-going studies

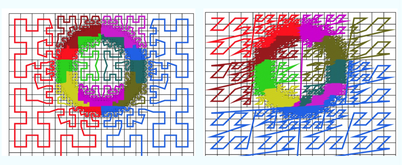
DEM

- Discussion of the computational performance of our developed simulation code by measuring the weak and strong scalability for the test-case simulation with using from 0.1 to 1.0 billon particles on multi-GPU (NVIDIA K20X) on TSUBAME2.5.
- Reconstruction of our simulation code to change its dataset to store particles from the Array of Structure (AOS) to the Structure of Array (SOA) to get high efficiency of the coalesced access on GPU.
- Optimization of our simulation code by using the On-Chip shared memory on GPU to store the particles which are in the same CUDA block to reduce the memory access to the VRAM on GPU.

SPH

- Development of the simulation code of SPH using multi-GPUs with dynamic load balance based on slice-grid method and discuss the appropriate way to sum up forces from neighboring particles and the way of node-to-node communication among GPUs on the account that a particle interacts with more than 100 particles in the SPH whereas a particle interacts with less than 10 particles in DEM.
- Execution of the large-scale SPH simulations applied to several practical multiphase flow problem such as dam break and bubbly flows on TSUBAME2.5.
- Measurement of the weak and strong scalability for the SPH simulation with using from 0.1 to 1.0 billon particles with multi-GPU on TSUBAME2.5.
- Discussion of the strong and weak point of the other dynamic load balance algorithm : Hilbert and Morton curve algorithm and compare the computational performance of the three.

Hilbert curve Morton curve



References

[1] P. A. Cundall, O. D. L. Strack, "A discrete numerical model for granular assembles," *Geotechnique* 29 (1979) 47-65.
[2] T. Hamada, T. Narumi, R. Yokota, K. Yasuoka, K. Nitadori, and M. Taiji, "42 tflops hierarchical n-body simulations on gpus with applications in both astrophysics and turbulence," *in Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, ser. SC '09. New York, NY, USA: ACM, 2009, pp. 62:1–62:12.
[3] G. S. Grest, B. Du¨nweg, and K. Kremer, "Vectorized link cell Fortran code for molecular dynamics simulations for a large number of particles," *Computer Physics Communications*, vol. 55, pp. 269–285, Oct. 1989.
[4] J. A. Bærentzen and H. Aanæs, "Computing discrete signed distance fields from triangle meshes," *Informatics and Mathematical Modelling*, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, Tech. Rep., 2002.

Acknowledgement

This research was supported in part by KAKENHI, Grant-in-Aid for Scientific Research (S) 26220002 from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan, in part by the Japan Science and Technology Agency (JST) Core Research of Evolutional Science and Technology (CREST) research program "Highly Productive, High Performance Application Frameworks for Post Petascale Computing", and in part by "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures (JHPCN)" and "High Performance Computing Infrastructure (HPCI)" in Japan. The authors thank the Global Scientific Information and Computing Center, Tokyo Institute of Technology for use of the resources of the TSUBAME2.5 supercomputer.