

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

Satori Tsuzuki, Takayuki Aoki

Abstract—Billion particles are required to describe granular phenomena by using particle simulations based on Discrete Element Method (DEM), which computes contact interactions among particles. Multiple GPUs on TSUBAME 2.5 in Tokyo Tech are used to boost the simulation and are assigned to each domain decomposed in space. Since the particle distribution changes in time and space, dynamic domain decomposition is required for large-scale DEM particle simulations. We have introduced a two-dimensional slice-grid method to keep the same number of particles for each domain. Due to particles across the sub-domain boundary, the memory used for living particles is fragmented and the optimum frequency of de-fragmentation on CPU is studied by taking account for data transfer cost through the PCI-Express bus. Our dynamic load balance works well with good scalability in proportion to the GPU number. We demonstrate several DEM simulations running on TSUBAME 2.5.

Keywords—Particle method, multi-GPU computing, Dynamic Load Balance.

I. BACKGROUND

Discrete Element Method (DEM) is used for numerical simulations of granular mechanics. 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 quite unequal.

We propose an efficient method to realize large-scale particle simulations based on short-range interactions such as DEM or SPH on GPU supercomputers by using dynamic load balance among GPUs.

II. DISCRETE ELEMENT METHOD

The particle interaction is modeled as a spring force proportional to the penetration depth of the contacting two particles.

Satori Tsuzuki is with the Department of Energy Science, Tokyo Institute of Technology, Japan, E-mail: tsuzuki@sim.gsic.titech.ac.jp

Takayuki Aoki is with the Global Scientific Information and Computing center (GSIC), Tokyo Institute of Technology, Japan, E-mail: taoki@gsic.titech.ac.jp

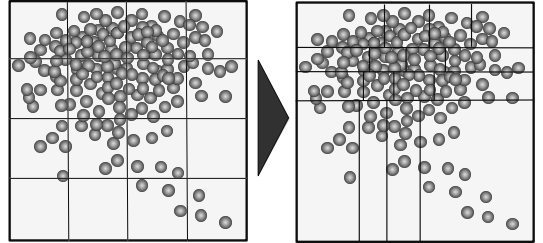


Fig. 1. Outline of dynamic load balance based on 2-dimensional slice-grid method.

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,

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

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 using 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.

Linked-list technique should be introduced to save the memory use[1]. A memory pointer is added to each particle data to have a reference to the next particle existing in the virtual mesh sequentially and we can reduce to 1/8.

To interact with objects described by CAD data, we generate the signed-distance function on a uniform mesh, instead of direct accesses to the CAD surface patches[2].

III. DYNAMIC LOAD BALANCE AMONG GPUS

The whole computational domain is decomposed into several subdomains 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[3], the horizontal and vertical boundary lines are shifted in turn in Fig.1. The number of particles moving to neighbor subdomains across the boundary are counted by the

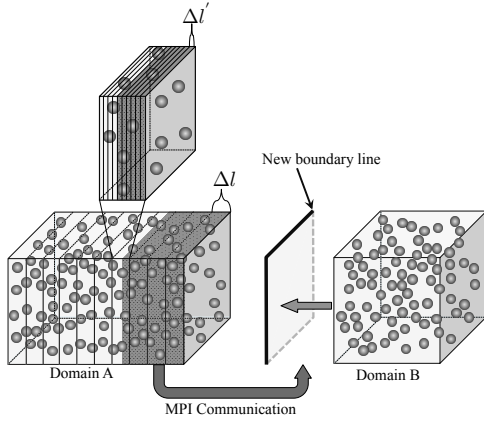


Fig. 2. Counting particles on GPU and boundary shift.

following process iteratively:

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

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

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. Figure. 2 shows the process of counting particles and boundary shift on GPU.

Fragmentation of GPU memory degrades the memory access and memory usage. De-fragmentation revives the performance to linear increase by re-numbering as shown in Fig.3. The frequency of de-fragmentation should be optimized.

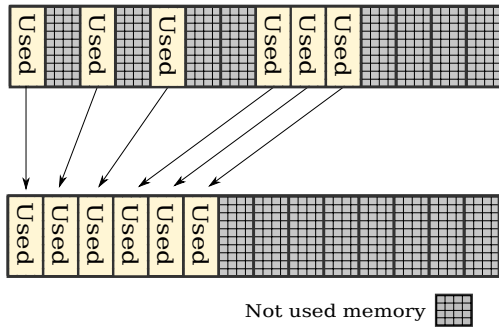


Fig. 3. Re-numbering of GPU memory.

IV. APPLICATION TO THE PRACTICAL PROBLEM

We demonstrated several DEM simulations: a golf bunker shot, a conveyor with screw, an agitation analysis, a spiral slider, were carried out on TSUBAME 2.5 in Tokyo Tech.

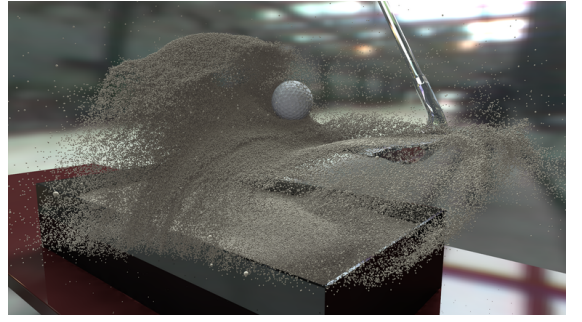


Fig. 4. A golf bunker shot simulation with 16 million particles.

V. STRONG SCALABILITY ON TSUBAME2.5

The strong scalability for the test-case simulation with using from 0.1 to 1.0 billion particles on multi-GPU (NVIDIA K20X) on TSUBAME2.5 was measured.

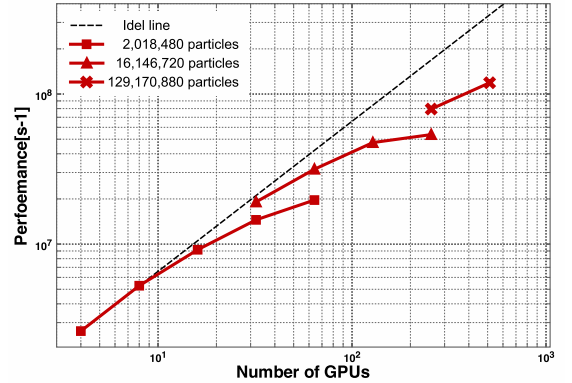


Fig. 5. Strong scalability of DEM simulation on TSUBAME 2.5.

VI. CONCLUSION

Our DEM simulation code works well with good scalability in proportion to the GPU number by introducing the dynamic load balance based on the 2-dimensional slice-grid method. Large-scale DEM simulations for the practical problems are carried out on TSUBAME supercomputer.

REFERENCES

- [1] G. S. Grest, B. Dünweg, 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.
- [2] 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.
- [3] 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. [Online]. Available: <http://doi.acm.org/10.1145/1654059.1654123>