The first real-scale DEM simulation of a sandbox experiment using 2.4 billion particles

M. Furuichi
Department of Mathematical Science and Advanced Technology, Japan Agency for Marine-Earth Science and Technology (JAMSTEC)
m-furuic@jamstec.go.jp

D. Nishiura
Department of Mathematical Science and Advanced Technology, Japan Agency for Marine-Earth Science and Technology (JAMSTEC)
nishiura@jamstec.go.jp

M. Asai
Department of Civil Engineering, Kyushu University
Japan
asai@doc.kyushu-u.ac.jp

T. Hori
R&D Center for Earthquake and Tsunam, Japan Agency for Marine-Earth Science and Technology (JAMSTEC)
horit@jamstec.go.jp

ABSTRACT

A novel implementation of the Discrete Element Method (DEM) for a large parallel computer system is presented to simulate a sandbox experiment with realistic particle sizes. To save memory in the pairwise tangential forces and halve the arithmetic costs, interactions are calculated using the action-reaction law. An iterative load-balancer the flexible 2D orthogonal domain decomposition is applied to overcome the load-imbalance problem caused by the Lagrangian nature of DEM. An overlapping communication technique combined with cell-ordering with space-filling curves is also applied to hide the overhead cost because of the MPI communication tasks. We verify our complex parallel implementation with the action-reaction law via a reproducibility test. The parallel scaling test shows good strong, and weak scalabilities up to 2.4 billion particles on the Earth Simulator and the K computer. The world's first real-scaled numerical sandbox simulation successfully captures the characteristics of real observations.

KEYWORDS

Discrete Element Method, Dynamic load-balancing, Overlapping communication, Space filling curve, Thrust formation

1 INTRODUCTION

The Discrete Element Method (DEM) has been widely accepted as an effective numerical approach for granular problems associated with fields, including engineering, geophysics and non-linear physics. For example, our targeted DEM problem is a sandbox experiment that is a scaled analog model of thrust formations in accretionary prisms. For such a problem, it is essential to have many particles for a qualitative and quantitative understanding of granular dynamics. However, the efficient parallel implementation of DEM is still a challenge. Thus, we developed new DEM codes designed for HPC systems to overcome the degradation of parallel performance due to the load imbalance and communication costs by utilizing an iterative dynamic load balancer [1] and overlapping communication techniques with space-filling curve (SPC).

2 Parallel implementation of DEM

To parallelize the DEM code for HPC systems, hierarchal stages of optimization, such as distributed-memory, shared-memory and cache levels, should be used. For distributed-memory tuning with the MPI, workload imbalance problems arise for moving particles. We proposed a new dynamic load-balancing method designed for particle simulations with short-range interactions such as the DEM [1]. Our method is based on the flexible 2D orthogonal grid domain decomposition [2] to manage the imbalances in the execution time between MPI procs. The sub-domains of the MPI procs are iteratively updated within the framework of an iterative non-linear solver. Our Smoothed particle hydrodynamics (SPH) code utilizing this load-balancer shows strong and weak parallel scalings upto 3.5B (billion) particles on the K computer and Earth Simulator (ES). The iterative load-balancer has been previouly introduced [1], and it is suitable for our target DEM simulation because it can handle different computational costs for individual particles depending on the number of contacting particles.

The shared-memory implementation is also based on earlier studies [3,4]. We applied preconditioning to generate the list of potentially contacting particle pairs via a sorting technique [3]. We used the action-reaction law to calculate the interactions because it outperformed an alternative method of calculating the interaction twice in elapsed time [4].

The DEM model utilizes normal and tangential forces with a frictional contact. The parallel implementation of the normal force is straightforward and similar to other particle simulation methods such as SPH and molecular dynamics. However, the implementation of tangential forces differs because the interactions of each particle pairs should be stored and synchronized via MPI to obtain the cumulative tangential displacement. We can save memory by using the action-reaction law to store the pair forces. However, the parallelization of DEM utilizing the action-reaction law requires complex programming. For example, a cell's particle order is stored with the initial particle IDs in the halo region that are defined by the initial particle positions to ensure that consistent relations are maintained between the action and reaction. To verify such a detailed implementation, a strict debugging test is presented in Sec 4.

3 Overlapping commutation via SPC

Parallel optimization can further be achieved using overlapping communication techniques. Each MPI process holds a grid-based subdomain and adjacent halo cells, as shown in Fig. 1(a). By dividing the subdomain into two parts, the outermost subdomain cells and the remaining inner cells, we can overlap the inner cells computations with the data exchange for the halo region after updating the outermost subdomain cells. To optimize the memory access, particles are ordered in the arrays with the halo, outermost and inner cells. We also reorder the cell labels with the SPC (Hilbert curve), as shown by the inner arrows in Fig. 1(a) for cache-aware data access for the inner domain computation [5].

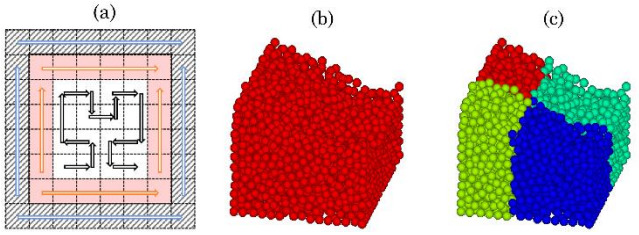

Figure 1:(a) Halo (striped), outer most (pink) and inner cells (white) in the MPI subdomain. Results after pushing left wall from uniformly distributed particles with (b) one and (c) four processes. Different colors identify different MPI processes.

4 RESULTS

To verify our parallel implementation of the distributed-memory DEM code, we performed the reproducibility tests shown in Figs. 1 (b) and (c). In these tests, the shared-memory parallelization was switched off to prevent changing the order of summation. The two solutions from the same initial setup but performed with a different number of MPI processes successfully showed an identical solution.

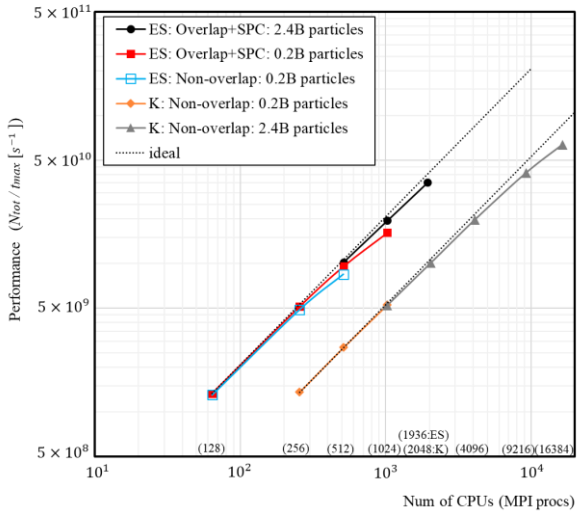

Figure 2: Strong/weak scalability of DEM on ES and K-computer.

The scaling test with/without the overlapping technique is shown in Fig. 2. We observe good, weak and strong scaling behavior upto 2B particles and 1,936 CPUs in the ES (the maximum was 2,048 CPUs). The performance improvement using the overlapping technique is also shown for the 0.2B particles cases. The performance improvement on the ADB (cache) hit ratio was 1.24% with reordering via SPC. Parallel scaling was also observed by K-computer upto 16,384 CPUs although the reordering scheme was unavailable owing to its slow integer computation.

With the developed DEM code, a real-size-scale sandbox simulation with 2.4B particles (100 μm particle size and 1 m × 1 m × 0.02 m domain size) was performed using ES and K computer. The load imbalance (t_max- t_min)/t_ave observed around at the time step in Fig. 3 was 7.4%.

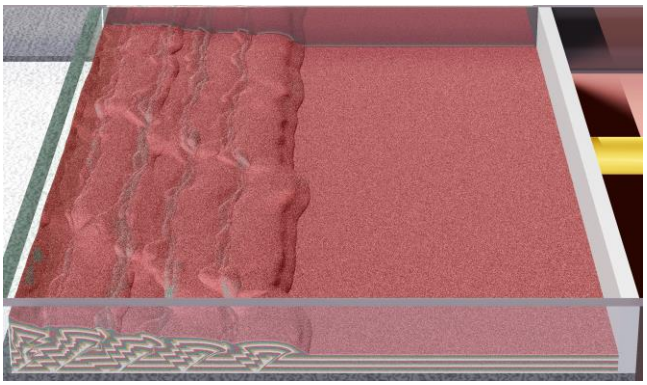

Figure 3: Snapshot of sandbox simulation with 2.4B particles.

5 CONCLUSIONS

We developed a parallelized DEM code with a dynamic load balancer and overlapping communication techniques. Our code is available for different supercomputer systems, including the ones with heterogeneous architectures [1]. The performance and verification tests show good parallel scaling and credibility of our code. The real-scale numerical sandbox experiment shows similar results to the analog experiment, which analyzes the granular mechanisms in the spontaneous generation of mountain belts uplifted by thrust formation.

ACKNOWLEDGMENTS

This study was supported by the Post-K project of (MEXT) using the K computer at the RIKEN (Project ID: hp160221), Earth Simulator project of JAMSTEC (Project name: Development of Advanced Simulation Methods for Solid Earth Simulations) and a Grant-in-Aid (Project ID:JP15K17754) from the Japan Society for the Promotion of Science.

REFERENCES

[1] M. Furuichi and D. Nishiura, 2017, *Phys. Commun.,* Vol 219, 135-148 DOI: 10.1016/j.cpc.2017.05.015
[2] J. Makino, 2004. A Fast Parallel Treecode with GRAPE. *PASJ*. 56, 521–531
[3] D. Nishiura, H. Sakaguchi, 2011, *J. Comput. Phys.,* 230, 5, 1923-1938, Doi:10.1016/j.jcp.2010.11.040
[4] D. Nishiura, M. Furuichi and H. Sakaguchi, 2015. *Phys. Commun.* Vol 194, 18-32 DOI:10.1016/j.cpc.2015.04.006
[5] S. Tsuzuki and T. Aoki. 2016. *Proceedings of the 7th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems ScalA*. 16, 1–8.