Speeding up LIGGGHTS using a MPI/OpenMP hybrid parallelization and the road towards adaptive time stepping

R. Berger[*,°], C. Kloss[+], S. Pirker[°]

[°] Department on Particulate Flow Modelling
Johannes Kepler University Linz
Altenbergerstrasse 69, 4040 Linz
e-mail: richard.berger@jku.at, web page: http://www.particulate-flow.at

[+] DCS Computing GmbH
Altenbergerstr. 66a - Science Park, 4040 Linz, Austria

ABSTRACT

DEM has become a viable tool for simulating a variety of industrial processes. As simulations become larger and more complex, DEM codes such as LIGGGHTS [1] must find new ways to deliver needed performance to complete these simulations in reasonable time.

LIGGGHTS uses domain decomposition and message-passing (MPI) to scale across many hundreds of processing cores. Because load-imbalance is a common problem in granular simulations, it added the ability to dynamically adjust subdomain boundaries along coordinate axes. However in many simulation domains heterogeneous particle distributions make finding good domain decompositions not trivial and is limited.

To better tackle load-imbalance a hybrid parallelization using both MPI and OpenMP parallelization has been developed. MPI domain decomposition is still used to generate multiple subdomains, allowing to distribute the workload among multiple compute nodes. Each node can then be fully utilized using OpenMP threads. Threaded versions of all major computational steps, including particle-particle and particle-wall interactions were created and achieve load balancing through various means. E.g., through the usage of the RCB partitioning algorithm [2] implemented in existing libraries.

The benefit of the hybrid approach is the added flexibility of multiple layers of decomposition. One mapping to the physical layout of the computing hardware, another focusing on balancing the workload among compute resources. So far these improvements have led to better performance of about 44% compared to a MPI-only parallelization in typical test cases.

Despite these successes, improvements achieved through better parallelization and load-balancing can only help up to a certain point. Finite memory bandwidth and synchronization overheads limit overall speedup of any parallelization. It is therefore necessary to keep investing in finding better, more intelligent algorithms for core problems to secure further advances.

One of the areas currently under investigation is the introduction of adaptive time stepping in our code. The main goal is to allow certain areas of a simulation domain to be simulated with different time steps than others, while generating qualitatively equivalent results as simulations with a homogenous time step. We present preliminary work of solving cases which dynamically freeze resting particles and exclude them from expensive computations such as pair-wise force computations and integration. Dormant particles are also dynamically reactivated in order to retain interesting physical phenomena such as shockwave propagation. Adding such capabilities introduces numerous side effects and is in conflict with other existing algorithms and data structures. But these are important prerequisites in order to come closer to our final goal. We report the progress made and further outline the next steps of adding these dynamic runtime optimizations.

REFERENCES

[1] C. Kloss, C. Goniva, A. Hager, S. Amberger, S. Pirker, *"Models, algorithms and validation for opensource DEM and CFD-DEM"*, Progress in Computational Fluid Dynamics, An International Journal 12 (2) (2012) 140-152.

[2] M. Berger, S. Bokhari, *"A partitioning strategy for nonuniform problems on multiprocessors"*, IEEE Transactions on Computers C-36 (5) (1987) 570-580