Parallelizing XDEM: Load-Balancing policies and efficiency, a study

PARTICLES 2017

Alban Rousset[†*], Xavier Besseron[†] and Bernhard Peters[†]

[*] Faculty of Science, Technology and Communication (FSTC)

University of Luxembourg (UL)

Campus Kirchberg, 6 Richard Coudenhove-Kalergi, 1359 Luxembourg
e-mail: firstname.lastname@uni.lu

ABSTRACT

Computational simulation like eXtended Discrete Element Method (XDEM) [1] has become an indispensable tool for researchers to explore systems before or without having recourse to real experiments. To improve results accuracy of a simulation more particles are needed. Increasing the size or the accuracy of a model increases the amount of computation needed to perform the simulation, requiring the use of parallel systems when it becomes too large in terms of memory or time consuming.

In such a parallel simulation, the execution can only be as fast as the slowest process, the fastest processes being waiting for the slowest at the synchronization points. Thus, in order to minimize this waiting time, it is critical to properly balance the load between all the processes. To address this issue, numerous Load-Balancing/Partitioning mechanisms have been proposed in the literature to distribute (and re-distribute) the workload between the processes.

In XDEM, the simulation domain is geometrically decomposed in regular fixed-size cells that are used to distribute the workload between the processes. The role of the partitioning algorithm is to distribute the cells among all the processes in order to balance the workload. To accomplish this task, the partitioning algorithm relies on a computing/communication cost that has been estimated for each cell. A proper estimation of these costs is fundamental to obtain pertinent results during this phase.

The study in the work is twofold. First, we integrate five partitioning algorithms (ORB, RCB, RIB, k-way and PhG) in the XDEM framework [1]. Most of these algorithms are implemented within the Zoltan library [2], a parallel framework for partitioning and ordering problems. Secondly, we propose different policies to estimate the computing cost and communication cost of the different cells composing the simulation domain. Then, we present an experimental evaluation and a performance comparison of these partitioning algorithms and cost-estimation policies on a large scale parallel execution of XDEM running on the HPC platform of the University of Luxembourg. Finally, after explaining the pros and cons of each partitioning algorithms and cost-estimation policies, we discuss on the best choices to adopt depending on the simulation case.

REFERENCES

[1] Bernhard P. , *The extended discrete element method (XDEM) for multi-physics applications*, Scholarly Journal of Engineering Research, (2013).

[2] E. G. Boman and U. V. Catalyurek and C. Chevalier and K. D. Devine, *The Zoltan and Isorropia Parallel Toolkits for Combinatorial Scientific Computing: Partitioning, Ordering, and Coloring*, Int. J. Scientific Programming, (2012)