Multi-Node approach for setting a full micromagnetic framework - PETASPIN

Anna Giordano[1], Stefano Chiappini[2], Alfio Messina[2], Giulio Siracusano[3], Giovanni Finocchio[1]

[1] Mathematical and Computer Sciences, Physical Sciences and Earth Sciences, University of Messina, Italy.
[2] Istituto Nazionale di Geofisica e Vulcanologia, Italy.
[3] University of Catania, Italy.

Nowadays, state-of-the-art numerical solvers leverage on high performance computing (HPC) and parallelization capabilities as achieved through GPUs (Graphics Processing Unit), where popular hybrid programming with the Message Passing Interface (MPI) [1] and the Compute Unified Device Architecture (CUDA) [2] are among the dominant technological solutions. The full micromagnetic framework is an ideal test-bed for the development of software for multi-node architectures. In this approach, magnetization dynamics is calculated by solving Landau–Lifshitz–Gilbert-Slonczevski (LLGS) equation, [3] which models precessional motion of magnetization in the magnetic materials. The micromagnetic contributions give rise to magnetic fields that can be local (magnetocrystalline anisotropy), short range (exchange, Dzyaloshinskii Moriya interactions [4]) and long range (magnetostatic) non-local terms [5]. In a finite difference time domain solver, the computation of local and short non-local field terms on very large structures can be scaled without great effort up to the assignment of one discrete computational cell to one thread. The main simulation limit is the total amount of memory installed on the used GPU, as the greater number of cells is considered the more amount of memory is allocated. So, when large devices need to be simulated, a single GPU might not be capable of completing the calculations. To overcome this issue, we modify the solver code in order to run on a multi-node infrastructure. We present our computational framework (based on MVAPICH [6] middleware), named PETASPIN and extension of GPMagnet [7,8], that is an integrated hardware/software solution for application in massive micromagnetic computations. Our solver splits the entire dataset across all the available nodes, then performs the computation of the external magnetic field for every partial dataset (no need to have information from neighbor cells) inside every node. For the demagnetizing contribution (Fourier Transform), the relevant part of the dataset are transferred upon need from one node to another by means of MPI APIs, through a dedicated 56 Gbit/s Infiniband network. In this way we are able to process a large dataset that would not completely fit into a single GPU. To give an example, on our hardware setup we can process a maximum of 8 millions of cells/GPU, when dealing with devices with a larger amount of cells we are forced to use more than one GPU. The only drawback of this approach is a time delay introduced in the global process due to data transfer across the nodes with MPI interface, as the Infiniband link is significantly slower the internal bus. This approach can be used to study within a full micromagnetic framework devices for unconventional computing such as skyrmion based shuffler. [9]

[1] MPI Rel. 3 Guide [Online]. Available http://mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf
[2] NVIDIA CUDA C Programming Guide. (Online) http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf
[3] J.C. Slonczewski, J. Magn. Magn. Mater. 159 (1996) L1–L7.
[4] G. Siracusano et al, Phys. Rev. Lett. 117, 087204, 2016.
[5] G. Bertotti - Hysteresis in Magnetism, Academic Press, eBook ISBN 9780080534374, 1998
[6] MVAPICH-GDR [Online]. Available: http://mvapich.cse.ohio-state.edu/userguide/gdr/
[7] L. Lopez-Diaz et al., J. Phys. D, Appl. Phys., vol. 45, no. 32, p. 323001, 2012.
[8] A. Giordano et al, J. Appl. Phys. 111, 07D112, 2012.
[9] D. Pinna et al, preprint at arXiv:1701.07750, 2017.