MPI parallelization and performance aspects of a graph based LB flow solver

L. Abrahamyan[1*], J. Bernsdorf[2], T. Zeiser[3], P. Lammers[4], A. Hoekstra[1] and P. Sloot[1]

[1] *Section Computational Science, University of Amsterdam*
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
{labraham,alfons,sloot}@science.uva.nl
[2] *C&C Research Laboratories, NEC Europe Ltd., Rathausallee 10, D-53757 St.Augustin, Germany*
bernsdorf@ccrl-nece.de
[3] *Regionales Rechenzentrum Erlangen, University of Erlangen-Nuremberg,*
Martensstrasse 1, D-91058 Erlangen, Germany
thomas.zeiser@rrze.uni-erlangen.de
[4] *HLRS, Nobelstrasse 19, D-70569 Stuttgart, Germany*
lammers@hlrs.de

We propose an efficient parallelization strategy for a graph based lattice Boltzmann implementation and present performance results for a variety of complex geometries.

A special focus is the partitioning and memory/load balancing strategy for geometries with a high solid fraction and/or complex topology such as porous media, fissured rocks and geometries from medical applications. The topology of the lattice nodes representing the fluid fraction of the computational domain is mapped on a graph. Graph decomposition is performed with both the multilevel recursive-bisection and multilevel k-way schemes based on Kernighan-Lin and modified Fiduccia-Mattheyses partitioning algorithms. We use the METIS library [**?**, **?**] as efficient implementation of graph partitioning algorithms and study the scalability of our parallel lattice Boltzmann solver when different graph partitioning methods are used.

Most HPC systems available today use microprocessors with several levels of cache to bridge the gap between slow memory and the fast CPU. The performance of computational kernels therefore significantly depends on the data layout and thus efficient cache utilization.

For parallel codes, the impact of loop structures and data layout on the communication patterns between different processors (e.g. size and number of individual messages, necessity of temporary copies, ability to overlap computation and communication) adds further complexity.

As extension of pervious work [**?**] which was focused on the single processor performance, preliminary performance results and optimization strategies of the new parallel code will be presented for a variety of platforms ranging from commodity-off-the-shelf PCs to specialized vector systems.

References

[1] G. Karypis and V. Kumar *J. Parallel Distrib. Comput.*, **48(1)**, 96 (1998);

[2] A. Abou-Rjeili and G. Karypis *IEEE Int. Par. & Distr. Proc. Symp. (IPDPS)*, **(in press)**.

[3] G. Wellein, T. Zeiser, S. Donath, G. Hager *Comp. Fluids*, **(in press)**.

[*] Lilit Abrahamyan acknowledges support from the Dutch National Science Foundation, NWO, Token 2000 Distributed Interactive Medical Exploratory for 3D Medical Images (DIME) project (634.000.024) and HPC-Europa Transnational Access program.