Hybrid Parallel Simulations of Fluid Flows in Complex Geometries: Application to the Human Lungs

Mathias J. Krause*, Thomas Gengenbach, and Vincent Heuveline

Karlsruhe Institute of Technology (KIT)
Engineering Mathematics and Computing Lab (EMCL)
Institute for Applied and Numerical Mathematics 4
Fritz-Erler-Str. 23, Bldg. 01.86
76133 Karlsruhe, Germany
{mathias.krause,thomas.gengenbach,vincent.heuveline}@kit.edu
http://www.united-airways.eu, http://openlb.org
http://numhpc.math.kit.edu, http://emcl.kit.edu

Abstract. In this paper a hybrid parallel strategy dedicated to the simulations of fluid flows in complex geometries by means of Lattice Boltzmann methods (LBM) is introduced. The approach allows coping with platforms sharing both the properties of shared and distributed architectures and relies on spatial domain decomposition where each subdomain represents a basic block entity which is solved on a symmetric multi-processing (SMP) system. Main emphasis is placed on testing its realization and studying its efficiency on a realistic fluid flow problem with a highly complex geometry. Therefore, as a suitable problem the simulation of the expiration in the human lung, whose functionality is described by a dedicated two-scale model, is considered.

Keywords: Numerical Simulation, Lattice Boltzmann, High Performance Computing, Computational Fluid Dynamics, Human Lungs, Modeling, Respiratory System.

1 Introduction

The numerical simulation of the full human respiratory system corresponds to one of the *Grand Challenges* in scientific computing nowadays. The main difficulties are not only related to the highly complex multiphysics phenomenology involving multi-scale features but also to the complex geometry. Many other problems especially in the field of biotechnology and medical sciences are also characterized by underlying highly complex computational domains. Furthermore, related applications like the analysis of possible implications due to surgeries in advance or cognition-guided surgeries require to consider patient-individual geometries. A widely automated preprocessing as well as efficient

* Corresponding author.

M.R. Guarracino et al. (Eds.): Euro-Par 2010 Workshops, LNCS 6586, pp. 209–216, 2011.
© Springer-Verlag Berlin Heidelberg 2011

parallelization strategies are both necessary conditions aiming to enable real-time simulations. In this context, it is of great importance to take advantage of nowadays available hardware architectures like IBM Cell processors, Graphic Processing Units (GPUs), multi-core processors and especially hybrid high performance technologies that blur the line of separation between architectures with shared and distributed memory.

The goal of this work is to present a highly efficient parallelization strategy dedicated to the numerical simulation of fluid flows in complex geometries which takes great advantage of today's typical hybrid high performance computers. The approach is based on LBM which have evolved to a mature tool in computational fluid dynamics (CFD) and related topics in the landscape of both commercial and academic software in the last decade (cf. literature e.g. [1,2]). The simplicity of the core algorithms as well as the locality properties resulting from the underlying kinetic approach lead to methods which are very attractive in the context of parallel and high performance computing (HPC). The LBM, which are employed here, are discretization strategies for families of BGK-Boltzmann equations [3] which are related to the Navier-Stokes equations as shown by Saint-Raymond in [4]. Junk and Klar [5] interpret particular LB schemes of the presented type directly as Chorin-type projection methods for incompressible Navier-Stokes equations.

In [6,7], Krause et al. propose a hybrid parallelization strategy for LB simulations. In the following, this idea is picked up and extended for simulations with underlying complex computational domains. Its realization is tested for the simulation of an expiration in a human lung. The underlying two-scale model [8] is based on the coupling of a bronchiole model with a model describing the incompressible Newtonian fluid flow in the upper part of the human lungs which can be resolved using standard medical imaging like computer tomography (CT) or Magnetic Resonance Imaging (MRI).

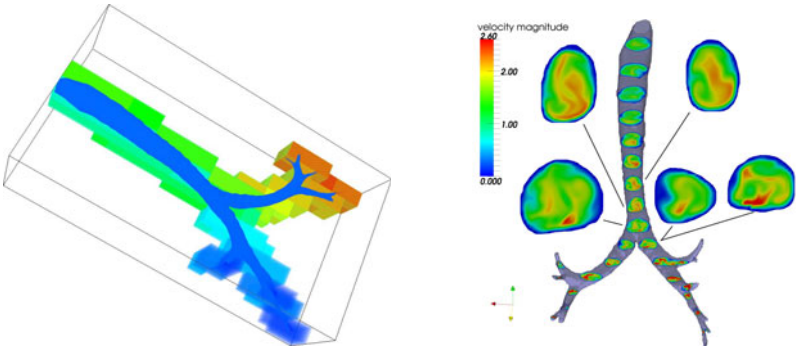
The paper is organized as follows: In Section 2, the hybrid parallelization strategy is explained. A test case is formulated in Section 3 which relies on a patient-specific lung geometry. Details concerning the discretization method are stated. Finally, the measured performance results on the *Jülich Research on Peta-flop Architectures (JUROPA)*[1] supercomputer are presented and analyzed.

2 Hybrid Parallelization Strategy for LBM

The most time demanding steps in LB simulations are usually the collision and the streaming operations. Since the collision step is purely local and the streaming step only requires data of the neighboring nodes, parallelization by domain partitioning leads to low communication costs and is therefore very efficient [9,10,11].

The hybrid parallelization strategy proposes to partition the data of a considered discrete position space Ω_h, which is a uniform mesh with spacing $h > 0$,

[1] http://www.fz-juelich.de/jsc/juropa



(a) The geometry of a human lung split into $n = 139$ cuboids which are assigned to processing units. Different colors represent different processing units. Each cuboid comprise about 50^3 computing grid points which are shared by cores within the same shared memory area.

(b) The simulated flow field for an expiration in the human lungs in various planes. The intersections are colored according to the computed velocity magnitude in m/s. Cf. [8]

Fig. 1. Partition of the geometry data (a) and resulting flow field (b)

according to their geometrical origin into $n \in \mathbb{N}$ disjoint, preferably cube-shaped sub-lattices Ω_h^k $(k = 0, 1, ..., n-1)$ of almost similar sizes.

This becomes feasible by extending Ω_h to a cuboid-shaped lattice $\widetilde{\Omega}_h$ through the introduction of ghost cells. Then, $\widetilde{\Omega}_h$ is split into $m \in \mathbb{N}$ disjoint, cuboid-shaped extended sub-lattices $\widetilde{\Omega}_h^l$ $(l = 0, 1, ..., m-1)$ of approximately similar size and as cube-shaped as possible. Afterwards, all those extended sub-lattices $\widetilde{\Omega}_h^l$ which consist solely of ghost cells are neglected. The number of the remaining extended sub-lattices $\widetilde{\Omega}_h^l$ $(l_0, l_1, ..., l_{n-1})$ defines n. Finally, one gets for each $k \in \{0, 1, ..., n-1\}$ the wanted Ω_h^k as a subspace of $\widetilde{\Omega}_h^{l_k}$ by neglecting the existing ghost cells.

For $p \in \mathbb{N}$ the number of available processing units (PUs) of a considered hybrid high performance computer which communicates by means of a network, an even load balance for complex geometries in particular will be assured if the domain Ω_h is partitioned in a sufficiently large number $n \in \mathbb{N}$ of sub-lattices. Then, several of the sub-lattices Ω_h^k $(k = 0, 1, ..., n-1)$ can be assigned to each of the available PUs. In order to find an optimal distribution where the communication costs between the PUs are kept low, a sophisticated graph-based partitioning algorithm can be applied. Then, each PU in turn, distributes its load for any of its sub-lattices Ω_h^k among the available threads sharing the local memory of the PU. The sub-lattice Ω_h^k is always a uniform mesh. Hence, a dedicated parallelization paradigm for the shared memory part of the platform can be applied in a way that takes advantage of the regular structure by means of efficiency.

3 Numerical Experiments

The aim of this section is to illustrate the proposed hybrid parallelization approach considering a realistic problem with an underlying complex geometry, namely the expiration in a human lung. Thereunto, a prototypical problem for the sub-model, describing the flow in the upper lungs, of the two-scale model introduced in [8] is formulated. Then, the applied numerical methods in order to solve this problem are stated. Finally, the performance results are presented and discussed. All numerical experiments are performed with the open source software package *OpenLB*[2]. In this library, the previously described hybrid parallelization concept has been realized through the use of OpenMP and MPI. A detailed description of both the implementation and performance tests for a simple underlying geometry are provided in detail in [6,7]. The obtained numerical results, see Figure 1b, for this simulation set-up are presented in detail in [8].

3.1 Case Study Description and Discretization Issues

An expiration for a realistic everyday situation is to be simulated. Thereby, the flow rate is fixed to be $Q = 150$ ml/s which is typically observed for adult males in situations of resting or sitting awake [12]. The air is considered at normal conditions, i.e. $1,013$ hPa and $20\,^{\circ}$C. Thus, its kinematic viscosity is $\nu = 1.4 \cdot 10^{-5}$ m^2/s.

In order to set up flow simulations, it is necessary to determine a couple of parameters, hence the velocity profiles can be set at the considered bronchi $i \in I$ in accordance with Poiseuille's law. The area of the boundary at the trachea is found to be approximately $A \approx 0.9065$ cm^2. To reach the desired flow rate of $Q = 150$ ml/s a mean speed of $U_{mean}(Q) \approx 1.65$ m/s is needed. This mean speed is considered to be the characteristic macroscopic velocity. The characteristic macroscopic length $L = 0.015$ m is fixed to be the diameter of A. With it, one obtains the Reynolds number $Re \approx 1768$. As a suitable test geometry, the lung of a middle-aged male human is considered. The computational domain Ω is extracted from CT data with a resolution of 0.4 mm. To solve the fluid flow problem in the upper part of the human lungs Ω a standard $D3Q19$ LB model with BGK collision operator [1,2] is applied. This discretization approach requires the computational domain to be a uniform, hence, a voxel mesh Ω_h. The patient specific mesh is created in a preprocessing step [13] which leads to $h = 0.23$ mm$/N$ for three different refinement levels $N = 1, 2, 3$. For $N = 3$ one obtains approximately 57.7 million fluid voxels, which results in about 1.1 billion unknowns for the $D3Q19$ LB scheme.

3.2 Performance Results

According to the concept presented in Section 2, the considered lattices Ω_h with $h = 0.23$ mm$/N$ and $N = 1, 2, 3$ need to be partitioned. Thereto, at first

[2] http://www.openlb.org

all Ω_h are extended to cuboid-shaped lattices $\widetilde{\Omega}_h$ through the introduction of ghost cells. An overview concerning the dimensions of the different considered extended lattices $\widetilde{\Omega}_h$ as well as its corresponding numbers of fluid, non-boundary and boundary cells is given in Table 1. In Figure 1a the obtained remaining extended sub-lattices obtained for $m = 1,000$ but also for the geometry of the upper human lungs are visualized.

Table 1. The considered extended discrete lattices $\widetilde{\Omega}_h$ representing the geometry of human lungs for different discretization parameters $h = 0.23$ mm$/N$ whereby $N = 1, 2, 3$ is the corresponding refinement level

refinement level N	dimension of $\widetilde{\Omega}_h$	fluid cells	non-boundary / fluid cells	boundary/ fluid cells
1	$452 \times 234 \times 859$	$2,327,357$	87.8%	12.2%
2	$900 \times 464 \times 1714$	$17,465,152$	93.6%	6.4%
3	$1350 \times 696 \times 2571$	$57,675,848$	95.7%	4.3%

Then, the three considered extended lattices $\widetilde{\Omega}_h$ ($h = 0.23$ mm$/N$ with $N = 1, 2, 3$) are split into $m = 1,200,\ 12,000,\ 24,000,\ 48,000$ disjoint extended sub-lattices $\widetilde{\Omega}_h^l$ ($l = 0, 1, ..., m-1$). Afterwards, all sub-lattices which just consist of ghost cells are neglected, leaving a certain number $n \leq m$ of extended sub-lattices $\widetilde{\Omega}_h^{l_k}$ ($k = 0, 1, ..., n-1$). This leads to twelve different test cases which are listed in Table 2. There, for each of the test cases ($h = 0.23$ mm$/N$ with $N = 1, 2, 3$) the obtained number n of remaining lattices $\widetilde{\Omega}_h^{l_k}$ ($k = 0, 1, ..., n-1$) as well as the minimal and maximal number of cells in the respective remaining lattices $\widetilde{\Omega}_h^{l_k}$ are specified. According to the presented strategy, all remaining extended sub-lattices $\widetilde{\Omega}_h^{l_k}$ ($k = 0, 1, ..., n-1$) are distributed among the available PUs. In the presented case, they are spread by building blocks with respect to the numbering.

The twelve test cases are executed on the supercomputer *JUROPA*. Thereby, in every case a different number of cores p, but always with just one core per node, is employed. The C++ source code is compiled with the Intel compiler using optimization level 3. In order to keep the total execution times low, the number of the time steps to be performed is always set to 100. The time in seconds measured for these 100 steps obtained employing p processes, respectively nodes, is captured by the variable t_p. In order to compare performances of LB implementations obtained on different computers or obtained for different implementation techniques, the measuring unit *million fluid-lattice-site updates per second* MLUP/s is frequently introduced, e.g. in [14]. In the following, this terminus is extended to also enable comparisons of LB code executed in parallel by introducing the measuring unit MLUP/ps, which stands for *million fluid-lattice-site updates per process and second*. The performance results obtained for the test cases are measured in this unit and captured by the variable P_{LB} which is defined according to $P_{LB} := N_c/(10^4 t_p p)$ whereby N_c denotes the number of fluid cells.

Table 2. The table shows all considered test cases and provides for each of the cases ($h = 0.23$ mm$/N$ with $N = 1, 2, 3$) the initial number m of sub-lattices $\widetilde{\Omega}_h^l$ ($l = 0, 1, ..., m-1$), the remaining number n of sub-lattices $\widetilde{\Omega}_h^{l_k}$ ($k = 0, 1, ..., n-1$) as well as the minimal and maximal number of cells in the respective remaining lattices $\widetilde{\Omega}_h^{l_k}$

case name	refinement level N	initial cuboids m	remaining cuboids n	max. cell cuboid	min. cell cuboid
N1m1200	1	$1,200$	152	73,710	77,142
N1m12000	1	$12,000$	744	6,859	8,000
N1m24000	1	$24,000$	$1,347$	3,375	4,096
N1m48000	1	$48,000$	$2,258$	1,728	2,028
N2m1200	2	$1,200$	161	581,256	602,454
N2m12000	2	$12,000$	741	57,798	62,400
N2m24000	2	$24,000$	$1,310$	28,830	30,752
N2m48000	2	$48,000$	$2,254$	14,440	15,625
N3m1200	3	$1,200$	158	2,000,349	2,026,056
N3m12000	3	$12,000$	734	198,476	205,320
N3m24000	3	$24,000$	$1,296$	97,336	103,776
N3m48000	3	$48,000$	$2,210$	47,952	52,022

In Figure 2 the measured performances P_{LB} are plotted as a function of the employed number of MPI-processes $p = 2^0, 2^1, ..., 2^8$. Considering the obtained graphs of all test cases, a general observation made is a slight decline of P_{LB} for increasing p. This is in accordance with the obtained performance results for the problem with an underlying simple geometry presented in [6,7].

For all considered test cases ($m = 1,200,\ 12,000,\ 24,000,\ 48,000$ and $p = 2^0, 2^1, ..., 2^8$) it is observed that the greater the refinement level N is chosen the better are the obtained performances P_{LB}. The best value $P_{LB} \approx 2.14$ MLUP/ps is measured for $m = 12,000$ and $N = 3$ on $p = 8$ nodes. This characteristics can be explained by the fact that the computational costs for a boundary cell are usually higher than for a non-boundary cell and that the ratio of boundary cells to fluid cells increases for smaller problem sizes as stated in Table 1.

Further, it is observed that for the cases where the number of initial cuboids m is the smallest, i.e. $m = 1,200$, in general the obtained performances P_{LB} are the smallest. In these cases, the number of obtained sub-lattices ranges from $n = 152$ to $n = 161$, while for the other cases it is much greater ($n \geq 734$). This leads to a relative bad load balance especially where $p \geq 64$. However, for a fixed problem size, which is characterized by N, the overhead due to communication costs will be greater if m and n increases. Thus, increasing m, it is expected that the positive effect of a better load balance is compensated by additional costs due to increasing effort for the three steps *blocking*, *communicating* and *writing to ghost cells* needed for the MPI-based implementation [7]. This is indeed observed in the test results. By trend, in the case where $N = 1$ the best result is obtained for $m = 12,000$ and for the cases $N = 2, 3$ for $m = 24,000$. Yet, it is to be noted

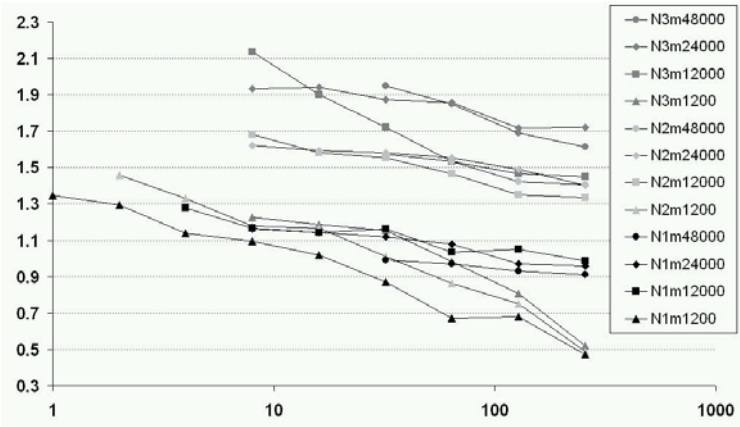


Fig. 2. The graph shows the obtained values for P_{LB} in MLSUP/ps (million fluid-lattice-site updates per process and second) as a function of the number of employed MPI-processes p for 100 times steps of an $D3Q19$ LB algorithm obtained on the *JUROPA*. The underlying computational domain is obtained by CT scans of the upper human lungs.

that the measured performances P_{LB} are relatively close to each other in the majority of the considered cases with $m \geq 12,000$. In particular, comparing the results obtained for $m = 24,000$ to those for $m = 48,000$, no significant drop of the performances P_{LB} is observed.

4 Conclusions

A hybrid parallelization strategy dedicated for LB simulations of fluid flow problems with underlying complex geometries is introduced. The approach is based on a partition of a domain into blocks which are distributed across the nodes of a cluster. Its realization takes advantage of the paradigms MPI and OpenMP and is part of the open source software *OpenLB*. It is tested for a realistic problem with an underlying highly complex computational domain, namely the upper part of a human lung extracted from CT data.

The obtained parallel performance results confirm the efficiency of the underlying approach in terms of the computing time that can almost be halved if the number of employed nodes is doubled. Thereunto, it is important to ensure a good load balance, which is accomplished by splitting the computational domain in a sufficient large number of sub-domains. The associated overhead is observed to be rather small. It is expected that applying a sophisticated graph-based partitioning algorithm would lead to a reduction of this overhead since relative expensive network communication could be replaced by memory operations. With the obtained progress in terms of efficiency of the hybrid parallelization, the simulation of a complete breathing cycle in the complete human lungs becomes technically feasible.

References

1. Chopard, B., Droz, M.: Cellular automata modeling of physical systems. Cambridge University Press, Cambridge (1998)
2. Hänel, D.: Molekulare Gasdynamik. Springer, Heidelberg (2004)
3. Krause, M.J.: Fluid Flow Simulation and Optimisation with Lattice Boltzmann Methods on High Performance Computers: Application to the Human Respiratory System. Ph.D. thesis, Karlsruhe Institute for Technology (KIT), Universität Karlsruhe (TH), Kaiserstrae 12, 76131 Karlsruhe, Germany (2010), http://digbib.ubka.uni-karlsruhe.de/volltexte/1000019768
4. Saint-Raymond, L.: From the BGK model to the Navier-Stokes equations. Annales Scientifiques de l'École Normale Supérieure 36, 271–317 (2003), http://www.sciencedirect.com/science/article/B6VKH-48HS9DK-5/2/4b7102c9ed9f501112dc9b08b7c9ae3d
5. Junk, M., Klar, A.: Discretizations for the Incompressible Navier-Stokes Equations Based on the Lattice Boltzmann Method. SIAM J. Sci. Comput. 22, 1–19 (2000)
6. Heuveline, V., Krause, M.J., Latt, J.: Towards a hybrid parallelization of lattice Boltzmann methods. Computers & Mathematics with Applications 58, 1071 (2009)
7. Heuveline, V., Krause, M.J.: OpenLB: Towards an Efficient Parallel Open Source Library for Lattice Boltzmann Fluid Flow Simulations. In: International Workshop on State-of-the-Art in Scientific and Parallel Computing. PARA, vol. 9 (2010) (accepted)
8. Gengenbach, T., Heuveline, V., Krause, M.: Numerical Simulation of the Human Lung: A Two-scale Approach. In: BMT 2010 - Reguläre Beiträge (BMT 2010 Reguläre Beiträge), Rostock-Warnemünde, Germany (2010)
9. Massaioli, F., Amati, G. Achieving high performance in a LBM code using OpenMP. In EWOMP 2002, The Fourth European Workshop on OpenMP (2002)
10. Pohl, T., et al.: Performance Evaluation of Parallel Large-Scale Lattice Boltzmann Applications on Three Supercomputing Architectures, p. 21 (2004)
11. Zeiser, T., Götz, J., Stürmer, M.: On performance and accuracy of lattice Boltzmann approaches for single phase flow in porous media: A toy became an accepted tool - how to maintain its features despite more and mor complex (physical) models and changing trends in high performance computing! In: Resch, M., Shokin, Y., Shokina, N., et al. (eds.) Proceedings of 3rd Russian-German Workshop on High Performance Computing, Novosibirsk, July 2007. Springer, Heidelberg (2008)
12. Valentin, J.: Basic Anatomical and Physiological Data for Use in Radiological Protection: Reference Values - ICRP Publication 89. Annals of the ICRP 32, 1–277 (2002), http://www.ingentaconnect.com/content/els/01466453/2002/00000032/00000003/art00002
13. Gengenbach, T.: Modellierung und numerische Simulation der menschlichen Atemwege auf Hochleistungsrechnern. Master's thesis, Universitaet Karlsruhe (TH) (2009)
14. Wellein, G., Zeiser, T., Hager, G., Donath, S.: On the single processor performance of simple lattice Boltzmann kernels. Computers & Fluids 35, 910 (2006)