# SCALABLE LATTICE BOLTZMANN LEAPS TO EXASCALE

Radim Vavrik, Ondrej Vysocky, Kristian Kadlubiak, Lubomir Riha
IT4Innovations, VŠB - Technical University of Ostrava
Ostrava, Czech Republic

Jayesh Badwaik
Jülich Supercomputing Center
Germany

Romain Cuidard, Denis Ricot
CS Group
Le Plessis-Robinson, France

Gabriel Staffelbach, Markus Holzer
CERFACS
Toulouse, France

Philipp Suffa
University of Erlangen-Nuremberg
Erlangen, Germany

## KEYWORDS

LBM, code generation, GPU portability, C++ 17, technology transfer

**ACM Reference Format:**
Radim Vavrik, Ondrej Vysocky, Kristian Kadlubiak, Lubomir Riha, Jayesh Badwaik, Romain Cuidard, Denis Ricot, Gabriel Staffelbach, Markus Holzer, and Philipp Suffa. 2023. SCALABLE LATTICE BOLTZMANN LEAPS TO EXASCALE. In *Proceedings of Supercomputing 2023 (SC2023).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 INTRODUCTION

The primary goal of the EuroHPC JU project SCALABLE is to improve an industrial Lattice Boltzmann Method (LBM)-based computational fluid dynamics (CFD) solver to be capable of exploiting current and future extreme scale architectures, while preserving its accessibility from both the end-user and software developer's point of view. This is accomplished by transferring technology and knowledge between an academic code waLBerla and an industrial code (LaBS). This poster briefly introduces the characteristics of both software packages and the technology transfer involved in the process with the resulting improvements both on CPU and GPU as well as the interest directed to energy efficiency.

Lattice Boltzmann methods are trustworthy alternatives to conventional CFD, showing roughly an order of magnitude performance advantage over Navier-Stokes approaches in comparable scenarios. The SCALABLE brings together the developers of waLBerla and LaBS to improve both solvers in terms of portability (e.g. targeting GPUs), energy efficiency scenarios, and transferring techniques between the two to achieve high performance, scalability, and energy efficiency breaking the silos between the worlds of scientific computing and physical flow modeling.


*SC2023, Nov 2023, Denver, US*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

## 2 EXCHANGE OF TECHNOLOGIES BETWEEN LABS AND WALBERLA

The main target of the collaboration is to get LaBS's accuracy on complex geometry with the good performance and scalability demonstrated by waLBerla. To do so, developers of both codes have exchanged ideas and strategies, and this has lead to new optimizations of the solver. First, adding the portability to NVIDIA GPUs and secondly, improving parallel efficiency on classical HPC clusters.

### 2.1 GPU portability

waLBerla uses code generation to tackle LBM execution on GPU accelerated clusters. Motivated by the performance observed with waLBerla, LaBS decided to push towards a GPU version. However, to maintain as much as possible a single source base for both CPU and GPU, LaBS chose to implement C++17 standard parallelism. Combined with the NVC++ compiler from NVIDIA, this allows to have a single CPU and GPU code. Two major optimization on the LaBS GPU prototype are the results of an exchange with waLBerla:

- The **merging of functions**. We observe that we have to reduce the number of calls to the GPU to increase the performance. So we have merged Propagate + Macroscopic and Gardient + Collide steps.
- The **reduction of memory footprint**. Among the quantities analyzed using NVIDIA profiler tools, one can denote the size of memory and number of registers used by each GPU thread, the occupancy percentage of the GPU card and the size of the grid used by the GPU kernels. So we replaced many static arrays with C++ macros.

This work, detailed in [1], had lead to an improvement on simple test case from 28 MLUPs to 609 MLUPs and on specific cases even 2550 MLUPs bridging the performance gap between LaBS and waLBerla.

### 2.2 Data exchange reduction

Another interesting exchange lead to the realization that data exchanges in LaBS was ~3-4x higher than waLBerla although they use similar solving methods. This lead to the optimization of the particle distribution function (PDF) data exchanges (see figure 1).

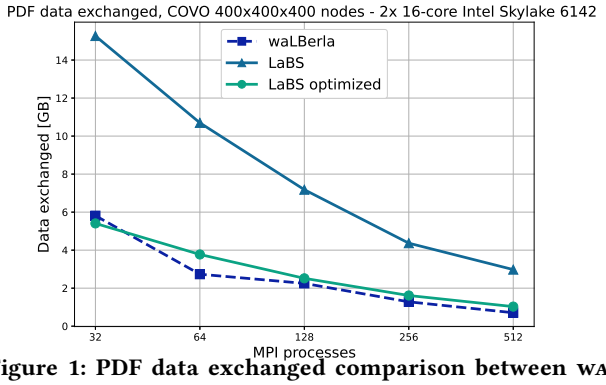


**Figure 1: PDF data exchanged comparison between waLBerla versus LaBS before optimization and after.**

### 2.3 Exporting waLBerla code generation

Building upon the waLBerla expertise using code generation, we engaged in an interesting experiment: waLBerla implemented the LaBS HRR collision model and resulting generated kernel was converted and integrated into the LaBS architecture.

This experiment has the following goals:

- For waLBerla, it allows the direct comparison with other classical collision models.
- For LaBS, it allows a direct comparison of the HRR performance and results using a hand made implementation and an automatically optimized one using the code generation.

Results are pending, as the kernel conversion to LaBS is on-going and is expected by SC23.

## 3 WALBERLA PERFORMANCE PORTABILITY ON EUROHPC JU SYSTEM: LUMI-G

In the SCALABLE project waLBerla was ported to support AMD GPUs using the ROCm framework and the code generation pipeline for highly efficient compute kernels. This effort can be demonstrated firstly on the LUMI supercomputer using up to 4096 MI250x GPUs (2x GCD per GPU). The results of a weak-scaling experiment on the Lagoon test case are shown in figure 2 using a uniform mesh with $512^3$ cells on each GCD. On 4096 GPUs we reach a scaling efficiency of 88 % using the AMD ROCm RDMA (GPU direct communication).

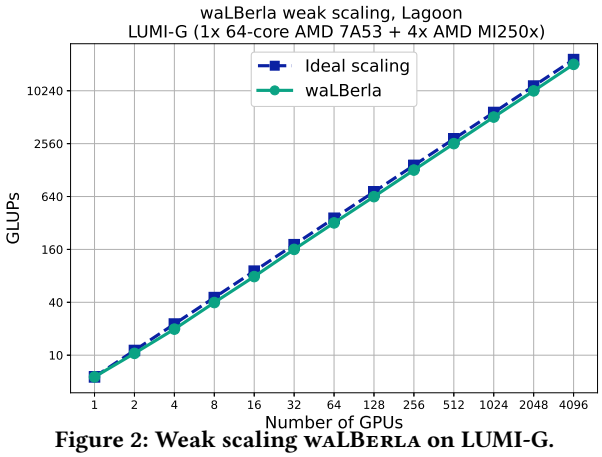


**Figure 2: Weak scaling waLBerla on LUMI-G.**

## 4 ENERGY EFFICIENCY AND PERFORMANCE

Most hardware has baseline settings defined by the vendor for a specific performance versus energy consumption. However all applications do not use the hardware in the same manner and it is possible to tune certain component's behavior to match the specific requirements of a running application, e.g. exploiting the dynamic behavior of different steps of a code. In SCALABLE, using the MERIC runtime system [3] implemented in Horizon 2020 READEX [2], we have successfully demonstrated improved energy consumption of LaBS on a non-accelerated system by up to 12 % without performance penalty using the dynamic tuning of CPU core and uncore frequencies. Further more, the accelerated version of waLBerla using the static tuning of GPU streaming multiprocessor frequency has been tested on NVIDIA A100 GPUs reducing energy consumption by almost 20 % with a 2 % impact on performance.

## 5 CONCLUSIONS

SCALABLE illustrates the synergy between academic and industrial application developers. Using academic codes as light house applications to single out successful porting and optimization strategies leads to concrete day to day improvements for applied tools used in industry. Here, data exchanges have been reduced by a factor 3 in the industrial code, leading to reduced bandwidth requirements for multi-core systems. Also, this code is now able to use NVIDIA GPUs using C++ 17 and NVC++ with comparable kernel performance with the automatic code generated code. This automatic code generator is now being used to improve CPU kernel performance as well by integrating the ad-hoc generated code from waLBerla into LaBS. waLBerla continues to pave the way with the first results on the EuroHPC JU system LUMI-G using AMD MI250x GPUs showing excellent scaling up to 4096 GPUs for now. This work will continue until the end of the project with a showcase application leveraging all improvement on these codes on a contra rotated open rotor Z08-AIPX7 case. Finally, using hardware tuning, we have improved the performance/energy consumption ratio by 10 to 20 % on CPU and GPU.

## 6 ACKNOWLEDGMENTS

The SCALABLE project (2021-2023) has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 956000. The JU receives support from the European Union's Horizon 2020 research and innovation program and France, Germany, Czech Republic. This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic (ID: MC2103) and through the e-INFRA CZ (ID:90254).

## REFERENCES

[1] Raphael Kuate. 2023. D3.3 implementation and final report about scheduling and load-balancing. (2023). https://zenodo.org/record/8153995.

[2] Joseph Schuchart et al. 2017. The readex formalism for automatic tuning for energy efficiency. *Computing*, 99, 8, (Aug. 2017), 727–745. DOI: 10.1007/s00607-016-0532-7.

[3] O. Vysocky, M. Beseda, L. Riha, J. Zapletal, V. Nikl, M. Lysaght, and V. Kannan. 2017. Evaluation of the HPC applications dynamic behavior in terms of energy consumption. In *Proceedings of the Fifth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*, 1–19. DOI: 10.4203/ccp.111.3.