Project Title	SCAlable LAttice Boltzmann Leaps to Exascale
Project Acronym	SCALABLE
Grant Agreement No.	956000
Start Date of Project	01.01.2021
Duration of Project	36 Months
Project Website	www.scalable-hpc.eu

D3.1 – Description of synthetic structured/unstructured data organization

Work Package	**WP 3.1, Choice of a structured/unstructured data organization**
Lead Author (Org)	**Raphael KUATE (CS GROUP)**
Contributing Author(s) (Org)	**Romain CUIDARD (CS GROUP)**
Reviewed by	**Romain CUIDARD (CS GROUP), Julien BILLIERE (CS GROUP)**
Approved by	**Management Board**
Due Date	**01.01.2022**
Date	**17.01.2022**
Version	**V1.0**

Dissemination Level

X	PU: Public
	PP: Restricted to other programme participants (including the Commission)
	RE: Restricted to a group specified by the consortium (including the Commission)
	CO: Confidential, only for members of the consortium (including the Commission)

Versioning and contribution history

Version	Date	Author	Notes
0.1	15.11.2021	Raphael KUATE (CS GROUP)	Creation
0.2	17.12.2021	Romain CUIDARD (CS GROUP)	Reviewed version
0.3	23.12.2021	Julien BILLIERE (CS GROUP)	Final version before approbation by the Management Board
1.0	17.01.2022	Raphael KUATE (CS GROUP)	Final version with integration of MB modifications

Table of Contents

List of Figures

List of Tables

TERMINOLOGY

Terminology/Acronym	Description
DoA	Description of Action
EC	European Commission
GA	Grant Agreement to the project
KPI	Key Performance Indicator
SCALABLE	SCAlable LAttice Boltzmann Leaps to Exascale
HPC	High-Performance Computing
LBM	Lattice Boltzmann Methods
CFD	Computational Fluid Dynamics

Executive Summary

The main objective of SCALABLE for CS GROUP is the improvement of LaBS deployment in bigger clusters of thousands of cores, achieved by a transfer of performance technology from WaLBerla. Therefore, in this deliverable 3.1, which sums up work conducted under Task 3.1, we focus on the choice of a suitable structured/unstructured data organization for LaBS. We thus present prospective work comparing the two codes LaBS and WaLBerla in many aspects and show that on industrial applications, the structured block data organization used in WaLBerla may not improve LaBS scalability. However, the domain decomposition resulting from the structured block data organization may improve data exchange enhancement in two aspects: a scheduled communication between blocks lying in the same process, and a reduction of the size of the surfaces involved in communication between blocks of different processes. These two points will, respectively, be the future work for the next tasks: Task 3.2 “Development of appropriate scheduling”, and Task 3.3 “Development of appropriate load-balancing".

1 Introduction

1.1 Context

Lattice Boltzmann methods (LBM) are nowadays trustworthy alternatives to conventional CFD methods, since it has been already shown in several engineering applications that they are faster than Navier-Stokes approaches in comparable scenarios. LBM methods can handle complex geometries and a wide range of Multiphysics applications that are of high industrial relevance. The main distinguishing feature of the LBM is its algorithmic locality stemming from an explicit time step. Thus, the LBM is especially well-suited to exploit advanced supercomputer architectures through vectorization, accelerators, and massive parallelization.

WaLBerla is one of the most advanced LBM research codes in the public domain. Its superb performance and unlimited scalability have been demonstrated, reaching more than a trillion lattice cells already on Peta scale systems. WaLBerla performance excels in academic use cases because of its carefully designed implicit blocks data structures. However, WaLBerla is not compliant with industrial applications due to lack of complex geometry engine and user friendliness for non-HPC experts.

The CFD software LaBS is an industrial LBM code with capabilities at a proven high level of maturity, but with high scalability performance in improvement. Therefore, in the context of EuroHPC, SCALABLE will transfer the performance technology from WaLBerla to LaBS. This collaboration will deliver improved scalability for LaBS to be prepared for the upcoming European Exascale systems.

1.2 Objective

The main objective of SCALABLE for CS GROUP is the improvement of LaBS deployment in bigger clusters of thousands of cores. The current usage context of LaBS, depending on the use case, is below a thousand of cores. In the current document, we will focus on data/algorithmic differences between LaBS and WaLBerla, in order to explore what data structure can be compatible with our performance target objective for LaBS. The remainder of this document is organized as follows. In the second section, we present the differences between LaBS and WaLBerla meshes and simulations on industrial tests cases. Then the third section focuses on main LBM simulations steps comparisons between LaBS and WaLBerla on academic test cases. The fourth section examine domain decomposition differences and point out some perspectives of LaBS development for the next work packages' tasks T3.2 and T3.3.

2 Industrial use cases comparisons

2.1 Existing data structure

The data structure in WaLBerla[1] employs structured blocks of cells of fixed size in each dimension of the space [1] [2] [3] [4]. The simulation domain is initially subdivided into equally sized blocks. Each initial block can be further subdivided into eight equally sized smaller blocks, and recursively until the local level of refinement is reached within each block. The size of blocks is a parameter which can be handle by the user, a size of $32 \times 32 \times 32$ is generally used. The default handling of levels of refinement in WaLBerla uses axis aligned bounding boxes (AABB). In industrial applications, non-meshed holes, internal solid boundaries and location of levels of refinement are widely described by mesh files. Thus, for industrial use cases, some adaptive works have been done in the meshing algorithm of WaLBerla for the efficient handling of complex geometries: the using of shell meshes files for internal boundaries and level of refinement description. The resulting domain partitioning in WaLBerla geometrically represents a forest of octrees with each initial block being the root of one octree. However, the holes of the domain not containing any initial block are not always emptied and are sometimes meshed, particularly in complex geometries. The case of which is inefficient for computations. Thus, another change has been made within the meshing algorithm of WaLBerla, introducing an un-refinement procedure in order to lightweight holes which should not have been meshed.
In LaBS[2], the data structure uses an unstructured cell design, thus allowing very flexible data refinement than a block structured cell design (WaLBerla). The main constraint on refinement levels in LaBS is a minimal number of consecutive cells of the same level or the minimal number of cells between successive resolution domain, for numerical scheme conveniences. The common constraint on refinement levels both in LaBS and WaLBerla is the so-called 2:1 balanced refinement constraint.

2.2 Prospective work

In order to explore different data structure improvement possibilities for LaBS, we have meshed complex geometries with both LaBS and WaLBerla for some industrial use cases with the same parameters: the exact internal holes location and levels of refinement. Since these cases needs mesh refinement, another constraint in WaLBerla is the number of halo/ghost layer for communication between blocks. In case of coarse-to-fine communication step, at least four ghost layers are required [4]. So, all WaLBerla computations done use four ghost layers.

2.2.1 Industrial test case Lagoon

The mesh has the following properties

- finest mesh size: 0.0005
- number of levels of refinement: 10

[1] https://www.walberla.net
[2] http://www.prolb-cfd.com

2.2.1.1 Mesh size

LaBS mesh

- number of internal nodes: 61 465 328
- number of equivalent finest nodes: 43 523 886

WaLBerla mesh

- size of blocks: 32 × 32 × 32
  - number of blocks: 94 344
  - number of cells: 3 091 464 192
- size of blocks: 16 × 16 × 16
  - number of blocks: 648 547
  - number of cells: 2 656 448 512
- size of blocks: 16 × 16 × 16 with un-refinement of holes and handling of levels of refinement using mesh files
  - number of blocks: 409 203
  - number of cells: 1 676 095 488
  - number of fluid cells: 1 594 992 209

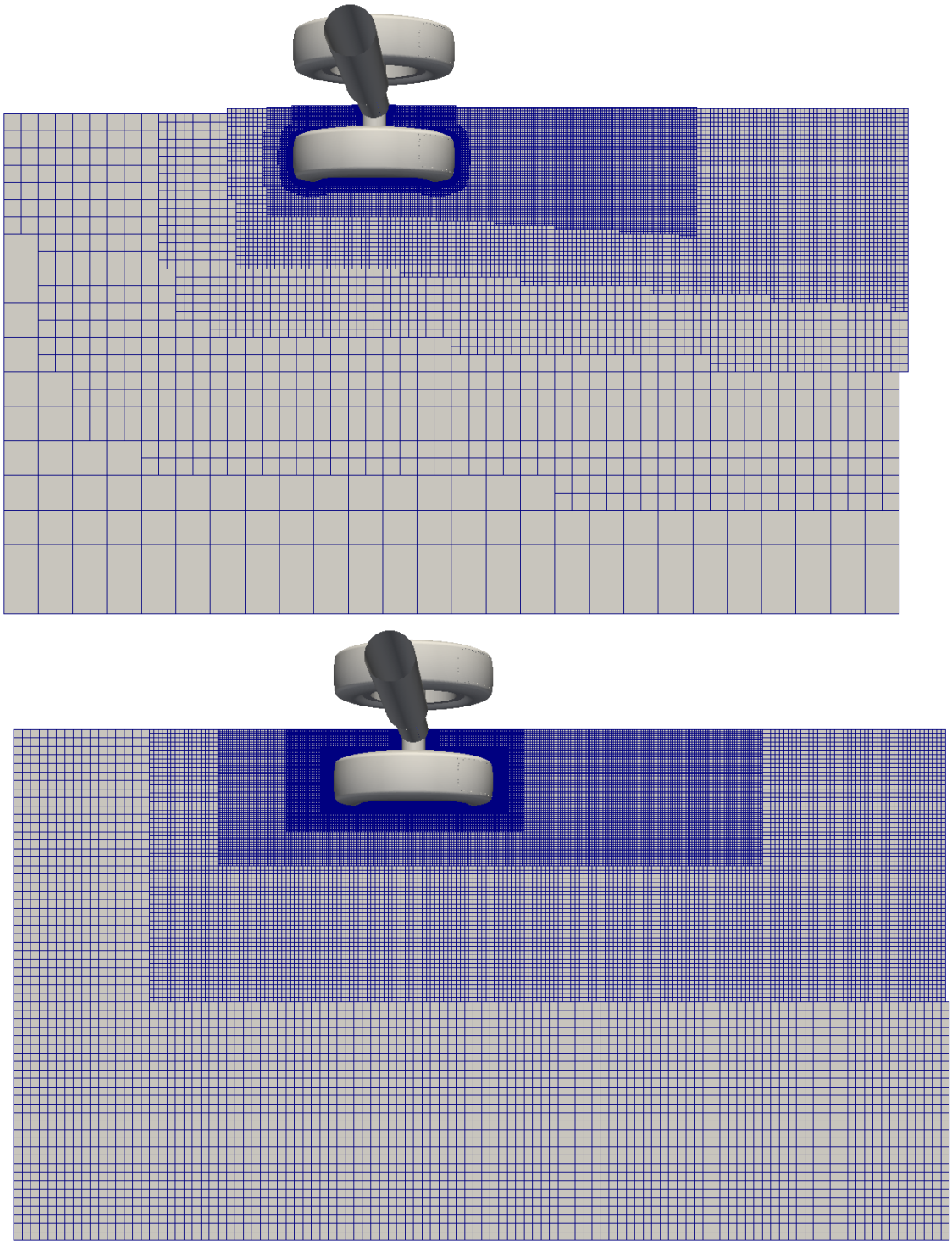

Figure 1 – Mesh of test case Lagoon. Slice around an excluded mesh area. Top picture: LaBS mesh. Bottom picture: WaLBerla mesh with un-refinement of holes, the finest mesh size area is extended.

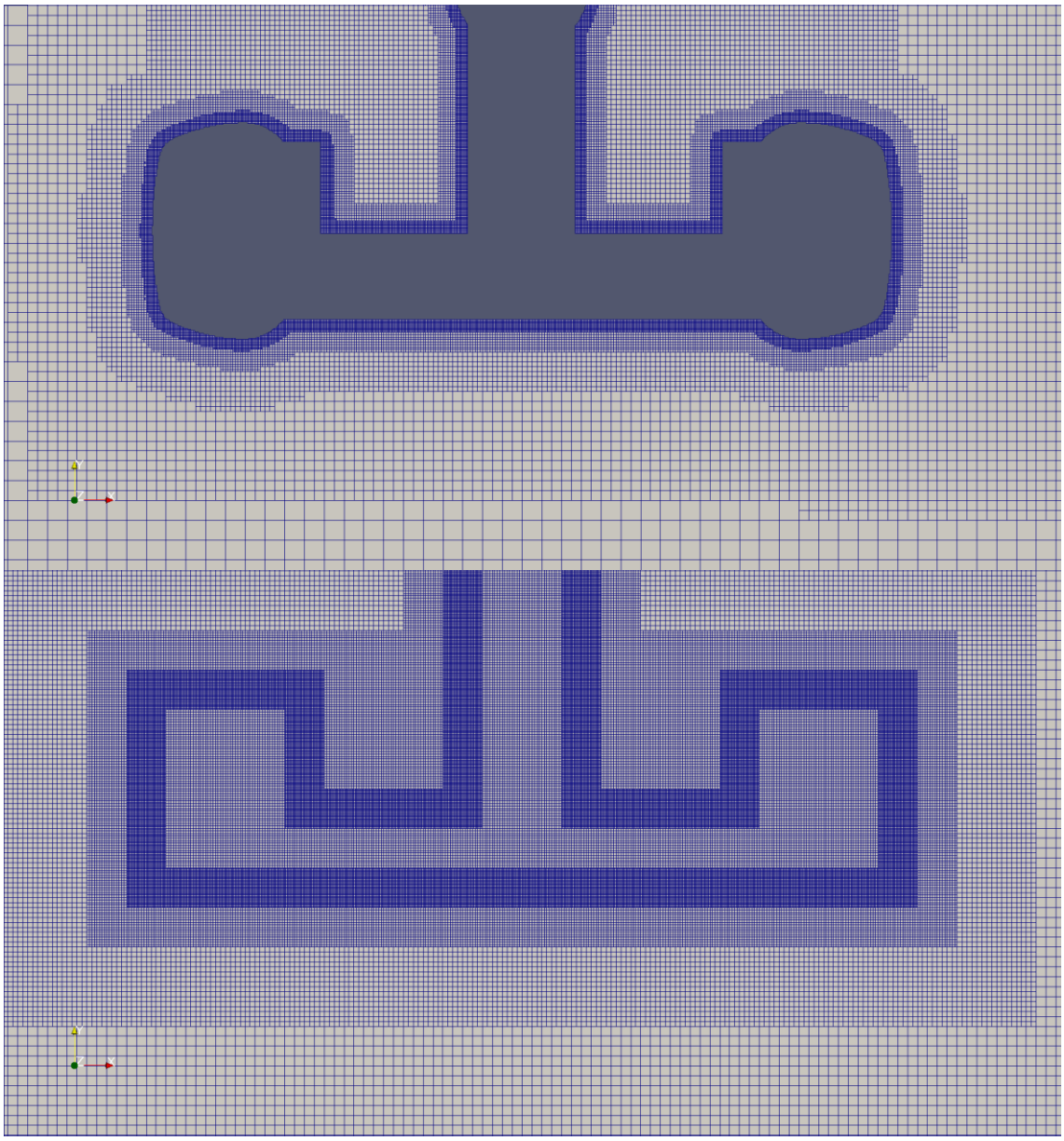

Figure 2 – Mesh of test case Lagoon. Slice around a mesh hole. Top picture: LaBS mesh. Bottom picture: WaLBerla mesh with un-refinement of area excluded by LaBS. The unstructured cell design (LaBS) allows flexible data refinement than a block structured cell design (WaLBerla)

Comparisons

On industrial cases, we observe that the number of nodes growths with the size of blocks. For the Lagoon test case, blocks of size 32 × 32 × 32 can produce up to three times more nodes than blocks of size 16 × 16 × 16. For equivalent meshes i.e., identical geometry, mesh size, levels and location of refinement area. We have noted that LaBS reduces WaLBerla mesh size for about 60% on many industrial test cases. On some cases like the test case Lagoon of which the critical mesh area is not centered within the simulation domain, the reduction of the mesh size in LaBS compared to WaLBerla can grow up to 96%.
From these observations the simulations with WaLBerla on industrial cases are done using blocks of size 16 × 16 × 16.

2.2.1.2 Simulations

For a complete view of the differences between LaBS and WaLBerla on industrial cases, we have simulated an LBM model on the previous geometries. The D3Q19 stencil has been used, with the SRT collision model, in order to be as close as possible to the standard LaBS model, even if the LaBS DRT collision model needs more computations. Default WaLBerla boundary conditions are used. The following results summarize the solver performances of the two codes. We use the MFLUPS (million fluid lattice cell updates per second) for performances comparison.

$$\mathrm{MFLUPS} = \frac{\#\mathrm{nodes} \times \mathrm{N_{iter}}}{\mathrm{T} \times 10^6}$$

with T being the elapsed time for $\mathrm{N_{iter}}$ iterations on #nodes fluid nodes. The mesh being non uniform, one has

$$\mathrm{MFLUPS} = \frac{N_{iter} \times \sum_{k=0}^{level\ max} \#nk \times 2^k}{\mathrm{T} \times 10^6}$$

with $\#nk$ being the number of nodes of level k. Level k = 0 corresponds to the coarse refinement level.

Simulation parameters

- number of cores: 448
- number of iterations: 10 000
- number of levels of refinement: 10

LaBS simulations

- number of internal nodes: 61 465 328
- number of finest equivalent nodes: 43 523 886
- elapsed time: 1 516.3 s
- MFLUPS: 14 696.451

WaLBerla simulations

- number of cells: 1 676 095 488
- number of fluid cells: 1 594 992 209
- elapsed time: ~ 7 days
- MFLUPS: 388.068

Comparisons

One can observe that the elapsed time of WaLBerla simulations can be reduced by the LaBS simulations up to 99%, on equivalent geometries. Even if the heavy mesh obtained from

WaLBerla can explain these differences, one must notice also that LaBS improves MFLUPS compared to WaLBerla, even if LaBS physics solved is more complex and cost a lot than the physics solved in WaLBerla. We have observed on many industrial cases, that LaBS scales up the MFLUPS about more than 25 times compared to WaLBerla. This scaling is about 37 on the test case Lagoon.

2.2.2 Industrial test case S2A

We present the summary of the test case S2A, the overall result behavior between WaLBerla and LaBS being closed to the test case Lagoon.

Simulation parameters

- number of cores: 480
- number of iterations: 220 000
- number of levels of refinement: 10

2.2.2.1 LaBS Simulations

- number of internal nodes: 62 481 806
- number of finest equivalent nodes: 32 308 869
- elapsed time: 27 483.6 s
- MFLUPS: 132 416.095

2.2.2.2 WaLBerla Simulations

- size of blocks: 16 × 16 × 16 with un-refinement of holes and handling of levels of refinement using mesh files
- number of cells: 194 129 920
- number of fluid cells: 148 069 698
- elapsed time: ~ 152 days
- MFLUPS: 709.018 (estimated on 5910 iterations)

2.2.3 Perspectives

The previous simulations comparisons between LaBS and WaLBerla on industrial test cases show that the use of block structured cell data design for LaBS as done in WaLBerla may not improve LaBS efficiency.

3 Computation steps on uniform mesh: structured blocks VS unstructured cell design

In order to examine the differences of behavior of the data structures during simulations, we have performed computations on a simple test case. The case of which the computation domain is uniformly meshed, such that the codes LaBS and WaLBerla use approximately the same number of nodes/cells.

3.1 Main LBM steps

Since the physics solved is more complex in LaBS, the comparison is focused on the collision and stream steps of the LBM. However, the collision step of the LBM is performed in LaBS in two sub-steps: *collide* and *macroscopic*, where in WaLBerla one has only one step. The following table summarizes the results obtained for computations using D3Q19 stencil, with the DRT scheme in LaBS and the SRT scheme in WaLBerla. The times units in the two last rows of each table (collide, stream) are in nanoseconds / (node*time step)

Table 1 – comparison of main LBM steps between WaLBerla and LaBS

	32 cores			*64 cores*		
	LaBS	*WaLBerla*	*gain*	*LaBS*	*WaLBerla*	*gain*
nodes /proc	99 266	101 306		1 000 000	1 000 000	
Collide	102.6+40.9	83.27	-18.84%	101.0+40.4	85.11	-15.75%
Stream	100.1	112.14	10.73%	101	106.24	5.77%
	128 cores			*256 cores*		
	LaBS	*WaLBerla*	*gain*	*LaBS*	*WaLBerla*	*gain*
nodes /proc	500 000	500 000		250 000	250 000	
Collide	101.1+40.3	84.11	-16.80%	102.7+40.6	84.09	-18.12%
Stream	100.2	104.93	4.50%	100.3	104.91	4.39%

The table above shows that we have equivalent computational times on the main LBM steps using structured block data in WaLBerla or unstructured cell design in LaBS. The differences in the collide step is explained by its subdivision in two sub-steps in LaBS and the fact that the DRT scheme of LaBS cost more than the SRT scheme used in WaLBerla.

3.1.1 Perspectives

The comparison of main LBM steps of simulation does not show major differences in computation costs within the block structured block data of WaLBerla and the unstructured data cells design of LaBS.

3.2 Data exchange amount

Once the calculations are done within a timestep of the LBM, one of the critical steps is the data exchange enhancement among the processes, in large scale computations scheduled on

a cluster. When the number of processes for a fixed mesh size increases, the amount of data exchanged increases also, since the boundary surfaces between sub-domains lying within each process increases. This overhead, if uncontrolled, may limit the code scalability. We have observed that the amount of data exchanged grows within a factor of 1.6 when the number of MPI process doubles, both for LaBS and WaLBerla, following the function:

$$x \mapsto \frac{C}{1.6^{\frac{\log\left(\frac{x}{16}\right)}{\log 2}-1}},\ x \geq 32$$

Where C denotes a constant.

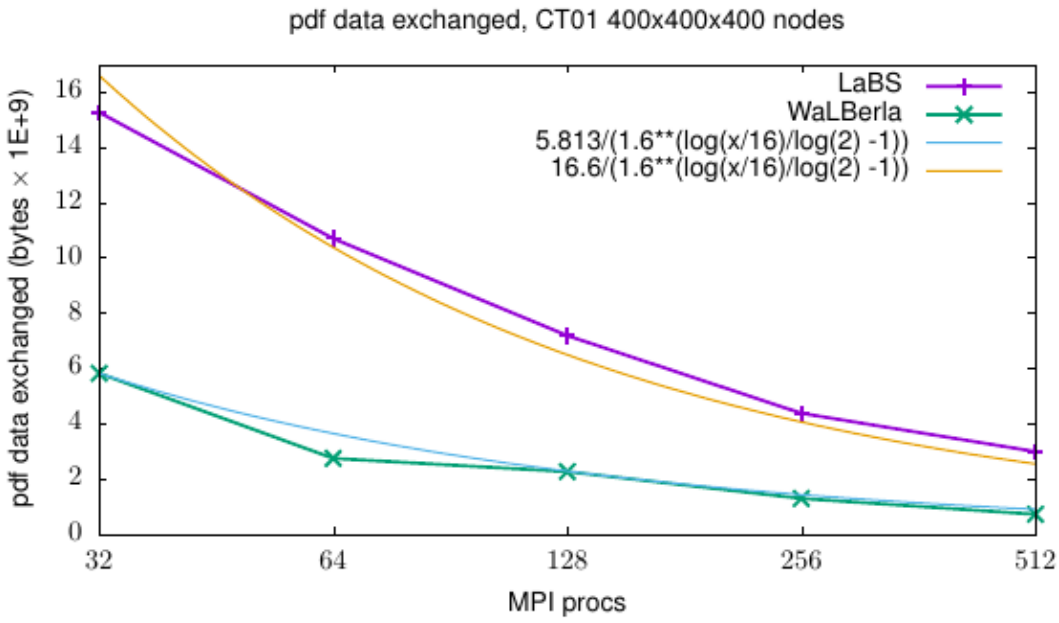

Figure 3 – Comparison of exchanged data per process on a fixed geometry. Uniform cube test case with 400x400x400 nodes.

The previous picture show that the data exchange amount scalability is exactly de same in both WaLBerla and LaBS. However, the bottom curve shows that within the optimized communication mode in WaLBerla, the amount of data exchanged is widely reduced.

3.2.1 Improving LaBS data exchange amount

We have investigated the differences in data exchange amount, and it appears that LaBS was not optimizing the particle distribution function (***pdf***) data exchanged. So, these optimizations have been added to LaBS and the following figure show that the amount of data exchanged is of the same order of magnitude in LaBS and in WaLBerla.

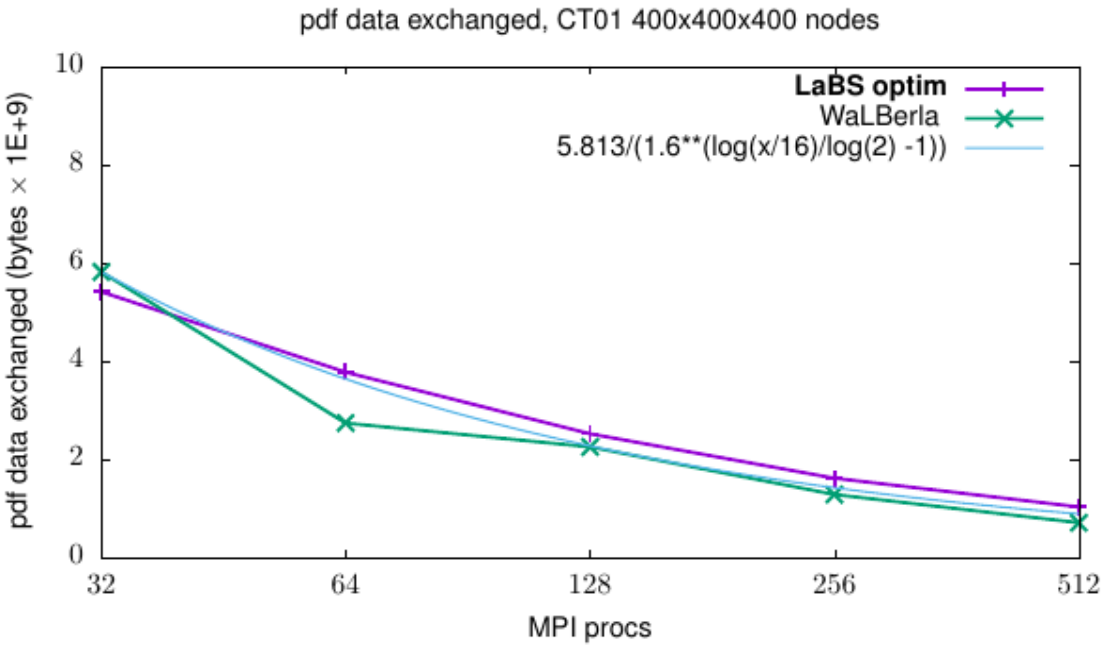

Figure 4 – Comparison of exchanged data per process on a fixed geometry. Uniform cube test case with 400x400x400 nodes, the optimization of amount of data exchanged for pdf values has been added in LaBS.

4 Domain decomposition

From the previous comparisons on data exchange amount, one can observe that WaLBerla amount of data exchanged remains slightly lesser than those exchanged in LaBS. The remaining difference may lie in the way that sub-domains involved within each processor are partitioned, as shown in the figure bellow. On average, the structured block data of WaLBerla optimizes the data exchange surface between sub-domains. On the other hand, the domain decomposition among processes in LaBS focuses only on load-balancing of nodes, the case of which is optimal for calculations on industrial cases within complex geometries but may not be optimal for data exchange enhancement.

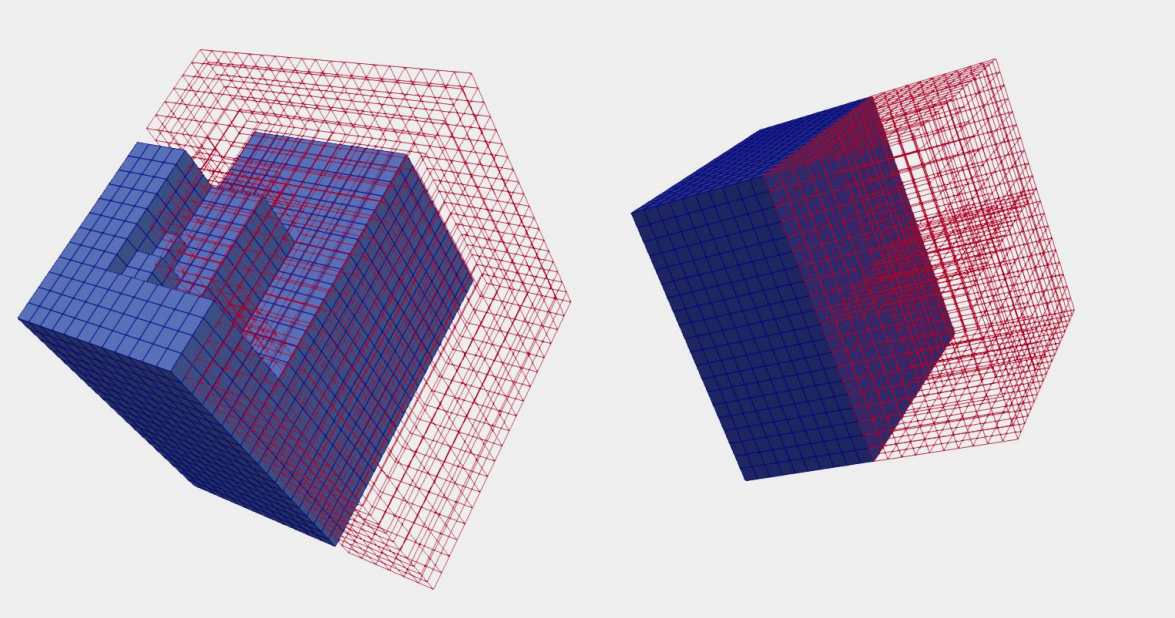

Figure 5 – Domain decomposition on between two processes. Left picture LaBS approach, right picture WaLBerla unstructured block approach.

We have compared the average number per processor of nodes in LaBS exchanging *pdf* data with distant nodes and the number of similar cells in WaLBerla. For these comparisons we used a light mesh of the industrial test case Lagoon with 7 levels of refinement and a reduced simulation domain. The domain is meshed with WaLBerla and the LaBS mesh reproduces approximatively the Waberla mesh i.e., its blocks for any level of refinement. We observed that for a fixed size mesh, this average number of cells sending *pdf* data does not vary a lot in WaLBerla (between 2361.25 and 2365.75), but the similar number of nodes decreases with the number of processes in LaBS (between 66289 and 16383), as the following picture shows.

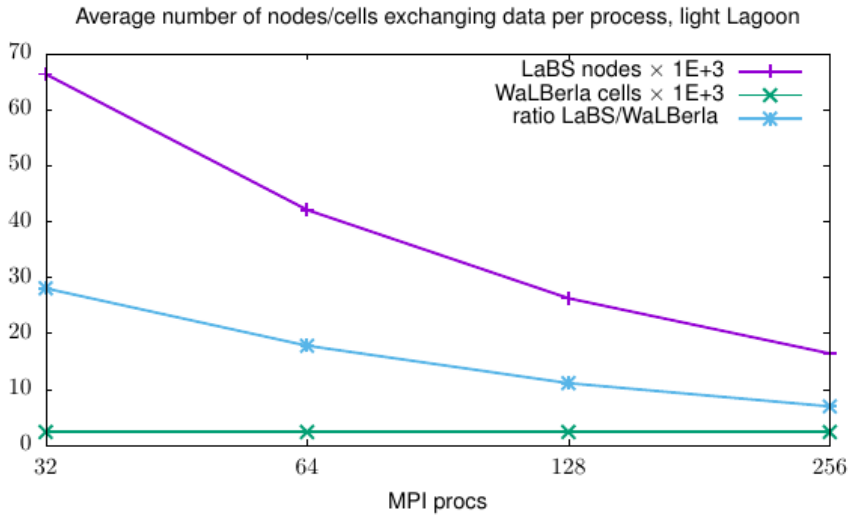

Figure 6 – Average cells/nodes sending *pdf* data per process. Light test case Lagoon, 7 levels of refinement and reduced simulation domain. Quasi-identical LaBS (11 470 216 nodes) and WaLBerla (10 350 592 cells) meshes.

The ratio between LaBS average number of nodes per process sending data and similar cells in WaLBerla on this test case varies from about 7 to about 30 on quasi-identical meshes. This

may seem huge however, LaBS “equivalent” mesh really uses fewer nodes and reduces the WaLBerla mesh size for more than 60%. The case of which the observed ratio may not really impact simulations within the same scale.

4.1.1 Perspectives

The previous observed differences in domain decomposition between LaBS and WaLBerla can be detailed in two aspects. First a revisiting of the load balancing algorithm of LaBS, w.r.t. the minimization of date exchange surface between different processes. Secondly, the data lying inside the same process but belonging to different levels of refinement are grouped into families int LaBS, thus, their management if scheduled as WaLBerla blocks, which are independent in terms of simulation steps, may be handled independently and therefore improve data exchange enhancements in LaBS. The furure work within the next tasks will then be organized as follows:

- **Task 3.2: Development of appropriate scheduling**
    - Scheduling of independently organized families, in terms of simulation steps, among LaBS data belonging to the same process.
- **Task 3.3: Development of appropriate load-balancing**
    - Improvement of LaBS load balancing algorithm w.r.t. the minimization of exchange surfaces between data lying into different processes.

5 Conclusion

The objective of this task 3.1 was scheduled to define the choice of a structured/unstructured data organization for LaBS possibly using one of the following approaches**:**

1. **Only large, structured blocks like in waLBerla, used sparsely at transitions and boundaries**
2. **Only smaller structured blocks, more suitable for thin layers**
3. **Combination of unstructured data like in LaBS, with larger blocks in fluid core, needing the management two different kinds of data storage**

We have investigated various aspects of the differences between LaBS and WaLBerla. In terms of mesh size and simulation MFLUPS on industrial tests cases, we have shown that no improvement of LaBS scalability may happen using unstructured block data organization. On the other hand, we have also examined the main simulation steps differences on academic tests cases, without transitions or any complex data resulting of geometries. Once more, we do not observe significant advantages of using any block structured data organization. So, none of the three previously proposed data organization appears being reliable for any improvement of LaBS scalability. However, our investigations on the data exchange step of the LBM simulation show that one may expect some improvements in LaBS, as described in the perspectives of the previous section.

6 Bibliography

[1] C. Feichtinger, S. Donath, H. Köstler, J. Götz and U. Rüde, "WaLBerla: HPC software design for computational engineering simulations," *Journal of Computational Science,* vol. 2, pp. 105-112, 2011.

[2] C. Godenschwager, F. Schornbaum, M. Bauer, H. Köstler and U. Rüde, "A Framework for Hybrid Parallel Flow Simulations with a Trillion Cells in Complex Geometries," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, New York, NY, USA, 2013.

[3] F. Schornbaum, "Block-Structured Adaptive Mesh Refinement for Simulations on Extreme-Scale Supercomputers," 2018.

[4] F. Schornbaum and U. Rüde, "Massively Parallel Algorithms for the Lattice Boltzmann Method on NonUniform Grids," *SIAM J. Sci. Comput.,* vol. 38, 2016.