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

Benchmark suite to evaluate performance and accuracy of the developments

D2.1

Work Package	WP 2: Systematic assessment of LBM covers and their extreme scale performance
Lead Author	G. Staffelbach(CERFACS)
Contributing Authors	Gabriel Staffelbach (CERFACS), Paul Pouech (CERFACS), Ondrej Vysocky (IT4I)
Reviewed By	Romain Cuidard (CS)
Due Date	01.10.2021
Date	30.09.2021X
Version	0.1

Dissemination Level

☒ 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)

The Scalable project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 956000.

Deliverable Information

Deliverable	Benchmark suite to evaluate performance and accuracy of the developments
Deliverable Type	
Deliverable Title	D2.1
Keywords	benchmark, performance assessment, energy efficiency assessment
Dissemination Level	Public
Work Package	WP 2: Systematic assessment of LBM covers and their extreme scale performance
Lead Partner	CERFACS
Lead Author	G. Staffelbach
Contributing Authors	Gabriel Staffelbach (CERFACS), Paul Pouech (CERFACS), Ondrej Vysocky (IT4I)
Reviewed By	Romain Cuidard (CS)
Due Date	01.10.2021
Planned Date	01.10.2021
Version	0.1
Final Version Date	30.09.2021X

Disclaimer:

The opinions of the authors expressed in this document do not necessarily reflect the official opinion of the SCALABLE partners nor of the European Commission.

D2.1

Gabriel Staffelbach (CERFACS), Paul Pouech (CERFACS), Ondrej Vysocky (IT4I)

October 15, 2021

Contents

List of Figures

List of Tables

The Scalable project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 956000.

Part 1
Benchmark suite description

This part constitutes the description of the four benchmark cases that will be used throughout SCALABLE. Two academic and two industrial cases were chosen. The acedemic cases offer the advantage of a known simple solution whereas the industrial uses cases are representative of the target applications of the end-user partners of the project.
Note that the final characteristics of the benchmarks (mesh size, refinement levels, resolution size) will be fixed for D2.2 as they condition the strong and weak scaling characteristics of the application on the target system.

1 Convected vortex

The idea of this test case is to convect a two dimensional isentropic vortex in a bi-periodic domain. An example of a convected vortex velocity field is displayed on Fig 1 in ProLB. This benchmark is too simple for extensive HPC use however offers the advantage of being simple enough to evaluate weak scaling performance and basic non regression for an analytical flow.
Such a vortex can be generated using the following function:

$$\psi(x,y) = \Gamma \exp\left(-\frac{(x-x_c)^2 + (y-y_c)^2}{2R_c^2}\right) \tag{1}$$

with Γ and R_c the circulation and vortex radius respectively. x_c and y_c correspond to the coordinates of the vortex center. Following this, both pressure and velocity local components can be expressed as follows:

$$u = U_0 + \frac{\partial\psi}{\partial y}, v = V_0 - \frac{\partial\psi}{\partial x} \text{ and } P - P_0 = -\frac{\rho\Gamma^2}{2R_c^2}\exp\left(-\frac{(x-x_c)^2 + (y-y_c)^2}{R_c^2}\right) \tag{2}$$

The values and significance of the different parameters of the aforementioned expressions (as well as some mesh specifics) are defined in the following table:

Convected Vortex Parameters	
Domain bounding box	-0.05 x 0.05 m
Resolution	64x64
Density	$\rho_0 = 1.1608\ Kg.m^3$
Temperature	300 K
Pressure	$P_0 = 1$ bar
Circulation	$\Gamma = 34.728\ m^2.s^{-1}$
Radius	$R_c = 0.005$ m
Convection Speed	$U_0 = 170\ m.s^{-1}; V_0 = 0$

Table 1: Characteristics of the CoVo use case

The Scalable project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 956000.

As in LBM the streaming operator is aligned with the grid, a vortex convection from west to east is too simple , only one direction is really tested. Therefore a more challenging test case is a vortex convected back with an angle, exactly *atan(1/10)/*π x 180 = 5,71059314 (after 10 revolutions, the vortex should be back in the center). The only added difference expected from the aligned use case is a small velocity component to the Y direction (south to north) which is specified in the following table:

Unaligned Convected Vortex	
Convection speed x	U_0=170 $m.s^{-1}$ (Mach 0.5)
Convection speed y	V_0=17 $m.s^{-1}$

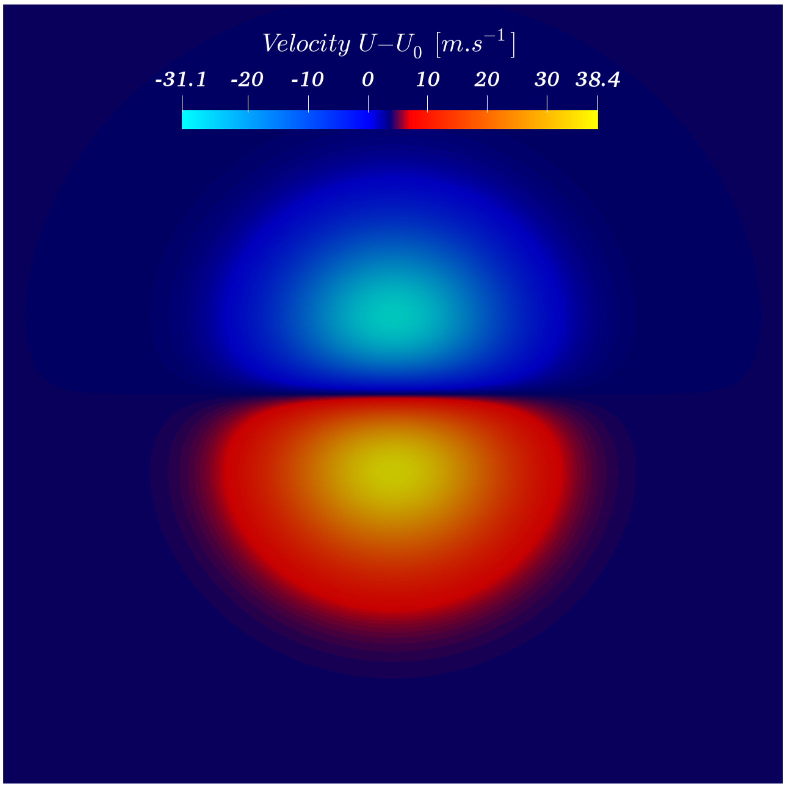

Figure 1: Longitudinal Velocity field of a convected vortex computed with ProLB

2 Turbulent flow between two parallel plates

In order to carefully assess turbulence modeling capabilities, it's important to start with cases where the analytical theory allows to be quantitative. The second academic test case is a classical turbulent flow in a channel composed of two parallel plates. This configuration is based on the Physics of Fluids article of Moser et al: *Robert D. Moser, John Kim, and Nagi N. Mansour. "Direct numerical simulation of turbulent channel flow up to* Re_θ= *590". In: Physics of Fluids 11.4 (1999), pp. 943–945. doi: 10.1063/1.869966. url: https://doi.org/10.1063/1.869966.* It consists of a three dimensional periodic channel with walls at the top and the bottom boundaries (as displayed on Fig 2. A forcing term is applied on the pressure gradient in the normal direction in order to create and maintain the flow direction. The mesh and physical parameters are given in table 2.

3 Lagoon Landing gear

The third test case, and the first industrial application, follows the previous one as it focuses on testing 3D turbulent flows capabilities but in a more complex situation: a turbulent flow around a full landing gear.

The Scalable project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 956000.

Turbulent flow between two parallel plates	
Shape LxHxW	πH/2 x H x 0.289πH/2
Dimensions LxHxW	0,314 x 0,2 x 0,09 m
Resolution Δx	0,001m
Resolution Δy	0,001m
Δy^+	5,44
Cells	5,652 Million tetrahedra (314x200x90)
Domain Height	H = 0.2m
Density	1.1608 $Kg.m^3$
Temperature	300 K
Pressure	10^5 Pa
Bulk Reynolds	Re = 10 000
Bulk Velocity	1,61268091 $m.s^{-1}$
Skin Friction coef.	5.908e^{-3}
Wall Shear stress	τ_w = 8,92e^{-3}
Forcing term	0,089179448 $kg.m^{-2}.s^{-2}$
Friction Reynolds	Re_τ= 544

Table 2: Characteristics of the turbulent channel use case

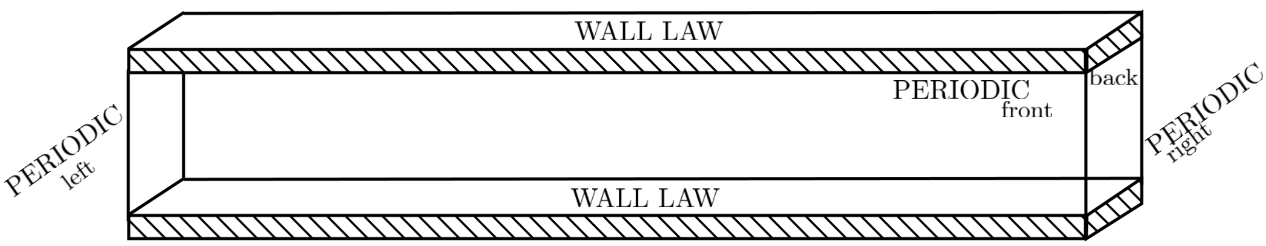

Figure 2: Standard configuration for the turbulent flow between parallel plates test case

The Scalable project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 956000.

Lagoon Parameters	
Reynolds number	$1.59e^6$
Reference Length	0.3 m
Pressure	99447 Pa
Temperature	293.15 K
Velocity	$78.99m.s^{-1}$
Minimal mesh size	$5e^{-4}$ m

Table 3: Characteristics of the Lagoon use case

Called ”lagoon”, this test case has been largely used to benchmark various industrial and research codes and consist in a simplified deployed front landing gear from an Airbus plane in final approach operating conditions before landing. The geometry of the Lagoon geometry is shown on Fig 3. This configuration produces an interesting non-trivial flow and allows to assess both the aerodynamic and acoustic fields and can be compared them to a large experimental and numerical database for validation. The physical and numerical parameters of the case are described in the following table:

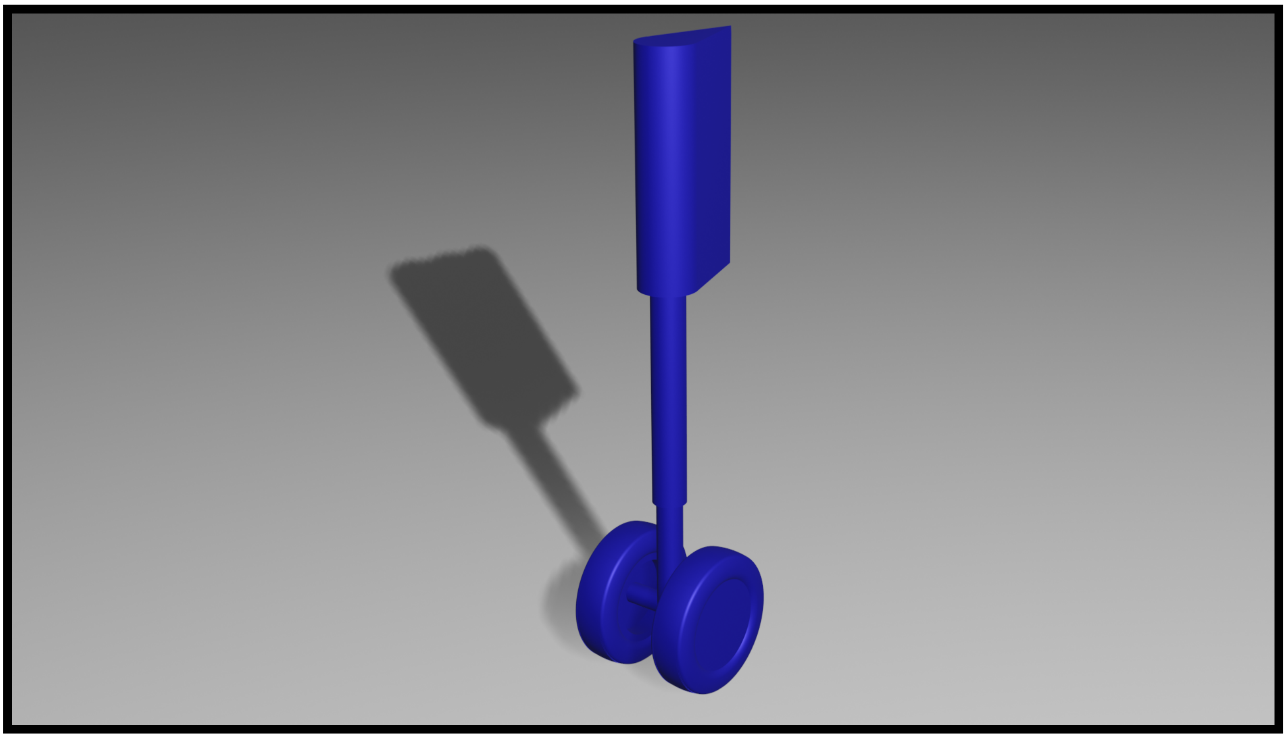

Figure 3: CAO of the Lagoon test case

The Scalable project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 956000.

Lagoon Parameters	
Reynolds number	$5.19e^6$
Reference Length	1.7 m
Pressure	101325 Pa
Temperature	293 K
Velocity	45.83 $m.s^{-1}$
Minimal mesh size	$25e^{-4}$ m

Table 4: Characteristics of the S2A use case

4 S2A flow around a car

The final test case consist in a modern sedan car placed inside a wind tunnel. The geometry of the car is displayed on Fig. 4 and parameters for the simulation are described in the following Table:

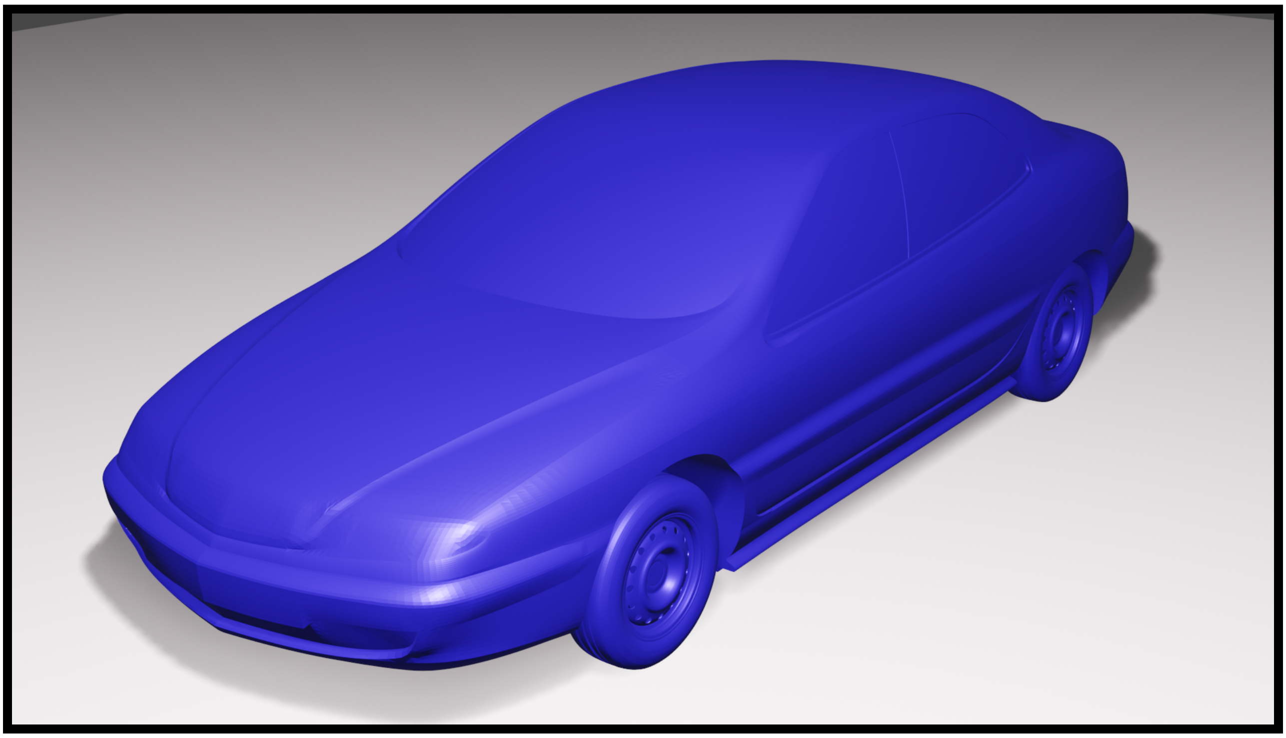

Figure 4: CAO of the S2A test case

The Scalable project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 956000.

Part 2
Performance tools and methodology

Under the SCALABLE project we will analyse the WaLBerla and ProLB applications behavior using various open-source as well as commercial tools [1] to identify performance bottlenecks of these applications. The performance analysis methodology of the POP-COE project will be used [https://pop-coe.eu/further-information/learning-material, https://pop-coe.eu/node/69].

The following list also contain tools for power consumption profile generation and dynamic hardware parameters tuning to improve energy efficiency of the applications. The most appropriate tool will be used depending on benching scenario (for example some tools do not offer GPU support and will not be therefore relevant for GPU profiling).

1. **Extrae**

   Extrae provides performance data collection mostly using the preload mechanism of the linker which enables omitting compilation hooks and executing the unmodified production binary. It intercepts the main parallel runtime environments (MPI, OpenMP, OmpSs, Pthreads, CUDA, OpenCL, SHMEM) and supports all major programming languages (C, C++, Fortran, Python, JAVA). Extrae also provides Basic analysis framework, that automatically generates POP metrics from the Extrea trace file, otherwise it must be evaluated manually.

2. **Paraver**

   Paraver is a trace-based performance analyser with great flexibility to explore and extract information from Extrae output files. Paraver provides two main visualisations: timelines that graphically display the evolution of the application over time, and tables (profiles and histograms) that provide statistical information. These two complementary views allow easy identification of computational inefficiencies such as load balancing issues, serialisations that limit scalability, cache and memory impact on the performance, and regions with generally low efficiency.

   Furthermore, Paraver contains analytic modules, for example the clustering tool for semi-automatic detection of the application structure, and the tracking tool to detect where to improve code to increase scalability.

3. **PyPOP**

   PyPOP is a python package for calculating POP metrics from application profiles, primarily designed for literate programming using Jupyter notebooks. PyPOP currently consumes Extrae output trace files only.

4. **Scalasca**

   Scalasca provides parallel application trace analysis that characterises its each execution's inefficiencies beyond those captured by call-path profiling tools. It detects wait states in communications and synchronisations, such as the "Late Sender" where receive operations are blocked, waiting for associated sends to be initiated, and the root cause of these. Also it highlights excess computation or imbalance. Contributions to the critical path of execution are quantified to single-out call-paths that are the best candidates for optimisation.

5. **Score-P**

   The Score-P instrumentation and measurement infrastructure, developed by a community including RWTH Aachen University, supports runtime summarisation and event trace collection for applications written in C, C++ and Fortran using MPI, OpenMP, Pthreads, SHMEM, CUDA and OpenCL. For scalability, event traces are analysed via a parallel replay with the same number of processes and threads as the measurement collection, generally re-using the same hardware resources.

6. **Cube**

   Cube, which is used as performance report explorer for Scalasca and Score-P, is a generic tool for displaying a multi-dimensional performance space consisting of the dimensions (i) performance metric, (ii) call path, and (iii) system resource. Each dimension can be represented as a tree, where non-leaf

[1] The listed commercial tools are available at IT4Innovations or FZJ cluster for the users. No license is necessary to buy under the SCALABLE project.

nodes of the tree can be collapsed or expanded to achieve the desired level of granularity. In addition, Cube can display multi-dimensional Cartesian process topologies.

7. **Vampir**

   Vampir, as well as Cube, is a GUI framework, that enables developers to quickly display and analyze arbitrary program behavior at any level of detail. The tool suite implements optimized event analysis algorithms and customizable displays that enable fast and interactive rendering of very complex performance monitoring data. The combined handling and visualization of instrumented and sampled event traces generated by Score-P or Scalasca enables an outstanding performance analysis capability of highly-parallel applications. Current developments also include the analysis of memory and I/O behavior that often impacts an application's performance.

8. **MAQAO**

   MAQAO (Modular Assembly Quality Analyzer and Optimizer) is a performance analysis and optimisation tool suite operating at binary level. It can run single and multi-node parallel applications. When compared to other tools it provides an advanced focus on core/node performance. Its main goal is to guide application developers along the optimization process through synthetic reports and hints. The tool mixes both dynamic and static analyses based on its ability to reconstruct high level structures such as functions and loops from an application binary. Since MAQAO operates at binary level, it is agnostic regarding the language used in the source code and does not require recompiling the application to perform analyses. At the moment, support for the Intel64, Xeon Phi and ARM architectures is implemented.

9. **MERIC**

   MERIC is a lightweight runtime system for parallel applications dynamic behavior detection and energy consumption tuning according the READEX approach. The library supports wide range of tunable hardware parameters, as well as various power monitoring systems.

   Open-source MERIC library is designed to minimize the energy consumption of the HPC infrastructure running a parallel application by dynamic tuning of a wide range of hardware knobs. It supports tuning of CPUs and GPUs of various vendors, as well as several power monitoring solutions. The library and associated tools perform detailed analyses of complex application behaviour, visualization of measured data, identification of the optimal hardware settings with respect to energy consumption and runtime, and dynamic tuning during the application runtime.

10. **RADAR visualizer**

    MERIC traces can be visualised using RADAR visualizer. Since MERIC executes parts of the application in various configurations when searching for optimal energy consumption settings, RADAR visualizer is designed to presents the application behavior in the default configuration and comparing it with every other evaluated configuration, which creates a unique insight to application behavior.

11. **ARM MAP**

    ARM MAP (previously known as Allinea MAP) from the ARM Forge tool suite for HPC software development is a parallel profiler that shows which lines of code took the most time and why. It supports both interactive and batch modes for gathering profile data, and supports MPI, OpenMP, and single-threaded programs. Syntax-highlighted source code with performance annotations enable you to drill down to the performance of a single line, and a rich set of zero-configuration metrics help you visualize memory usage, floating-point calculations, and MPI usage across processes.

12. **Intel Advisor**

    Intel Advisor a is tool for application behavior analysis at the core-level, focused on application vectorization, threading, and memory accesses. It is using knowledge of the application assembly code to generate roofline model that specifies a performance limitations of the workload caused by the memory bandwidth or the maximum performance of the floating-point units of the underlying hardware.

13. **Intel VTune**

    Intel VTune Profiler is a performance analysis tool at node-level for serial and multithreaded applications, that can be executed on a variety of hardware platforms (CPU, GPU, FPGA). The profiler also identifies application hotspots, I/O, resources utilization and boundedness, as well as parts of code for potential optimization. Moreover it reports hardware-related issues in the code such as data sharing, cache misses, or branch miss-prediction.

The Scalable project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 956000.

14. **Nvidia Nsight Systems**

    Nvidia Nsight Systems is a system wide performance analysis tool designed to visualize an application's algorithm and help single out performance optimisation opportunities using Nvidia hardware.

The Scalable project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 956000.