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

Application performance, accuracy and energy efficiency

D2.2

Work Package	WP 2: Systematic assessment of LBM covers and their extreme scale performance
Lead Author	G. Staffelbach(CERFACS)
Contributing Authors	Radim Vavrik (IT4I), Ondrej Vysocky (IT4I), Lubomir Riha (IT4I), Paul Poeuch (CERFACS), Markus Holzer (CERFACS), Gabriel Staffelbach (CERFACS)
Reviewed By	Romain Cuidard (CSGROUP)
Due Date	30.09.2021
Date	16.12.2021
Version	1.0

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	Application performance, accuracy and energy efficiency
Deliverable Type	
Deliverable Title	D2.2
Keywords	benchmark, performance assessment, energy efficiency assessment, READEX
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	Radim Vavrik (IT4I), Ondrej Vysocky (IT4I), Lubomir Riha (IT4I), Paul Poeuch (CERFACS), Markus Holzer (CERFACS), Gabriel Staffelbach (CERFACS)
Reviewed By	Romain Cuidard (CSGROUP)
Due Date	30.09.2021
Planned Date	1.12.2021
Version	1.0
Final Version Date	16.12.2021

Contents

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

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.

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

The test cases were executed at a non-accelerated part of the IT4Innovations system Barbora equipped with two Intel Xeon Gold 6240 CPUs (codename Cascade lake) per node. Each CPU has 18 cores (hyper-threading is disabled) designed to work at 150 W TDP. The CPU can reach up to 3.9 GHz CPU core turbo frequency when a maximum of two cores are active, 3.3 GHz when all cores execute SSE instructions, while the nominal frequency is 2.6 GHz. Since Nehalem architecture the Intel company has been using 'uncore' to refer to the frequency of subsystems in the physical processor package that are shared by multiple processor cores e.g., last level cache, on-chip ring interconnect or the integrated memory controllers, which in overall occupies approximately 30 % of a chip area [3]. The Barbora's CPUs can scale the uncore frequency in the range of 1.2–2.4 GHz. The CPU has six memory channels, and can provide up to 131 GB/s memory bandwidth. Each node has 192 GB RAM memory.

Barbora's computational nodes accommodate on board the Atos|Bull High Definition Energy Efficient Monitoring (HDEEM) system [2], that reads power consumption from the system hardware sensors and stores the data to an integrated memory. The sensor, that monitors the consumption of the whole node, provides 1000 power samples per second, and the rest of sensors, that monitors the node sub-units, use 100 samples per second. Both, aggregated values and power samples can be read from the user-space. Since the Sandy Bridge generation, Intel processors integrate a Running Average Power Limit (RAPL) hardware power controller, that provides power measurement and mechanism to limit power consumption of several domains [1].

For this deliverable LaBS gcc 8.3.0 and OpenMPI 4.0.2 where used and for WaLBerla gcc 9.3.0 and OpenMPI 4.0.3. Both were compiled without threading or accelerator support.

For tracing and the performance analysis we used Extrae 3.8.3 and Paraver 4.9.2 tools, both developed at Barcelona Supercomputing Center.

In the following section, numerical parameters for the use case is provided. For more details on the use cases please refer to deliverable D2.1. A repository containing the raw input files is available at IT4I upon request.

1 WaLBerla

1.1 COVO

The simulation of the convection of a compressible and isentropic vortex (COVO) is done on a two dimensional domain which is periodic in x and y-direction. Since WaLBerla is a three dimensional framework a layer of thickens one in z-direction exists. The numerical parameters of the COVO base base case are displayed in Table 1.

LBM scheme	D2Q9 Cumualant collision operator
Number of cells	$384 \times 384 \times 1$
Block size	$64 \times 64 \times 1$
Number of timesteps	5 001
VTK writing	every 1 000 timesteps

Table 1: WaLBerla COVO: Numerical Parameters

This use case is too small to benefit from HPC acceleration but is a well known non regression and accuracy test and will be used as such throughout the project.

1.2 Turbulent Channel

The simulation of the Turbulent Channel is done on a three dimensional domain which is periodic in x and z-direction. In y-direction NoSlip boundary conditions are applied. The numerical parameters of the Turbulent Channel base case are displayed in Table 2.

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

LBM scheme	D3Q19 SRT collision operator with Smagorinsky subgridscale model
Number of cells	$1720 \times 1080 \times 480$
Number of refinement levels	1
Number of timesteps	101
VTK writing	complete fields are not written
Extraction of turbulent characteristics	Every 50 timesteps

Table 2: WaLBerla Turbulent Channel: Numerical Parameters

1.3 Lagoon

The simulation of the Lagoon test case is done on a three dimensional domain and uses NoSlip boundary conditions in y and z-direction. In x-direction an inflow boundary condition is used on the eastern wall and an outflow boundary condition on the western wall. The numerical parameters of the Lagoon base case are displayed in Table 3.

LBM scheme	D3Q27 Cumulant collision operator
Streaming pattern	Esoteric Twist
Number of cells	$256 \times 576 \times 224$
Block size	$64 \times 32 \times 32$
Number of refinement levels	No mesh refinement is used
Number of timesteps	101
VTK writing	100 timesteps

Table 3: WaLBerla Lagoon: Numerical Parameters

1.4 S2A

The simulation of the S2A test case is done on a three dimensional domain uses NoSlip boundary conditions in y and z-direction. In x-direction an inflow boundary condition is used on the eastern wall and an outflow boundary condition on the western wall. The numerical parameters of the S2A base case are displayed in Table 4.

LBM scheme	D3Q27 Cumulant collision operator
Streaming pattern	Esoteric Twist
Number of cells	$1088 \times 224 \times 160$
Block size	$64 \times 32 \times 32$
Number of refinement levels	No mesh refinement is used
Number of timesteps	101
VTK writing	100 timesteps

Table 4: WaLBerla S2A: Numerical Parameters

2 LaBS

2.1 COVO

The convective vortex test case is computed on a bi-periodic 2D domain with an homogeneous mesh. The case numerical parameters are displayed on Table 5.

LBM scheme	D3Q19HRR
Number of cells	400x400x1
Cell size	$2.5e^{-4}$m
Number of iterations	97590

Table 5: LaBS COVO: Numerical Parameters

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

2.2 Turbulent Channel

The Turbulent Channel test case is computed on a three-dimensional domain with a velocity imposed at the inlet and a pressure imposed at the outlet. Top and bottom boundary conditions are wall laws. Turbulent Channel LaBS numerical parameters are displayed on Table 6.

LBM scheme	D3Q19DRT
Number of primary cells	628x400x150
Primary Cell size	$5e^{-4}$m
Refinement level	Top Wall: 1 ; Bottom Wall: 1
Minimal cell size	$2.5e^{-4}$
Final number of cells	92 221 800
Number of iterations	40 000

Table 6: LaBS Turbulent Channel: Numerical Parameters

2.3 Lagoon

The Lagoon test case is computed on three-dimensional domain with wall laws boundary conditions imposed on the lagoon walls. A frictionless boundary condition is imposed on the ceiling of the fluid domain. On the left side of the domain, a velocity is imposed to ensure an inflow boundary condition. All other boundaries are defined as an outlet with the same imposed pressure. Absorbing region are placed at all the inlet and outlet boundary condition to ensure non-reflectivity. The numerical parameters of the S2A base case are displayed in Table 7.

LBM scheme	D3Q19DRT
Model	Athermal LES-SISM
Primary Cell size	0.256m
Refinement levels	Lagoon walls: 1 Lagoon wake: 5 Fluid domain: 10
Minimal Cell size	$5e^{-4}$m
Final number of cells	61 465 328
Number of iterations	1000

Table 7: LaBS Lagoon: Numerical Parameters

2.4 S2A

The S2A car test case is computed on a three-dimensional domain with a fixed velocity imposed at the inlet and pressure imposed at the outlet. Two absorbing regions are placed at the inlet and outlet of the domain to insure non-reflectivity. All the car surfaces boundary conditions are wall laws as well as the ceiling of the overall fluid domain and the surface on which the car lies on. The numerical parameters of the S2A base case are displayed in Table 8.

LBM scheme	D3Q19DRT
Model	Athermal LES-SISM
Primary Cell size	0.64m
Refinement levels	S2A walls: 1 S2A wake: 4 Fluid domain: 9
Minimal Cell size	$25e^{-4}$m
Final number of cells	62 621 959
Number of iterations	520 000

Table 8: LaBS S2A: Numerical Parameters

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, accuracy and efficiency assessment

In order to evaluate scalability and efficiency of particular performance aspects in the above-mentioned parallel benchmark applications, the SCALABLE project will use a performance model and analysis methodology developed within the Horizon 2020 POP Center of Excellence.

Power consumption analysis follows the performance analysis. It focus on two basic ways how to look at the application behavior. First, static CPU frequencies tuning is performed to identify possible energy savings and its impact on application performance. Second, we present power consumption timeline of each use case, based on HDEEM power samples, which gives an idea about dynamic behavior of the application, and the difference in between these test cases. More dynamism, means that the static configuration does not fit the application, and under-exploit the possible energy savings, or leads to higher performance degradation.

3 Performance model

Attempting to optimise performance of a parallel code can be a daunting task, and often it is difficult to know where to start. For example, we might ask if the way computational work is divided is a problem? Or perhaps the chosen communication scheme is inefficient? Or does something else impact performance? To help address this issue, POP has defined a methodology for analysis of parallel codes to provide a quantitative way of measuring relative impact of the different factors inherent in parallelisation. This section introduces these metrics, explains their meaning, and provides insight into the thinking behind them.

A feature of the methodology is that it uses a hierarchy of metrics, each metric reflecting a common cause of inefficiency in parallel programs. These metrics then allow comparison of parallel performance (e.g. over a range of thread/process counts, across different machines, or at different stages of optimisation and tuning) to identify which characteristics of the code contribute to inefficiency.

The first step to calculating these metrics is to use a suitable tool (e.g. Extrae) to generate trace data whilst the code is executed. The traces contain information about the state of the code at a particular time (e.g. it is in a communication routine or doing useful computation) and also contains values from processor hardware counters (e.g. number of instructions executed, number of cycles).

The metrics are then calculated as efficiencies between 0 and 1, with higher numbers being better. In general, we regard efficiencies above 0.8 as acceptable, whereas lower values indicate performance issues that need to be explored in detail.

The approach outlined here is applicable to various parallelism paradigms, however for simplicity the POP metrics presented here are couched in terms of a distributed-memory message-passing environment (e.g. MPI). For this the following values are calculated for each process from the trace data: time doing useful computation, time in communication, number of instructions & cycles during useful computation. Useful computation excludes time within the overheads of parallelism.

At the top of the hierarchy is **Global Efficiency (GE)**, which we use to judge overall quality of parallelisation. Typically, inefficiencies in parallel code have two main sources:

- Overheads imposed by the parallel nature of a code
- Poor scaling of computation with increasing numbers of processes

and to reflect this we define two sub-metrics to measure these two inefficiencies. These are **Parallel Efficiency** and **Computation Efficiency**, and our top-level GE metric is the product of these two sub-metrics:

$$\text{GE} = \text{Parallel Efficiency} * \text{Computation Efficiency} \tag{1}$$

Parallel Efficiency (PE) reveals the inefficiency in splitting computation over processes and then communicating data between processes. As with GE, PE is a compound metric whose components reflects two important factors in achieving good parallel performance in code:

- Ensuring even distribution of computational work across processes
- Minimising time communicating data between processes

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

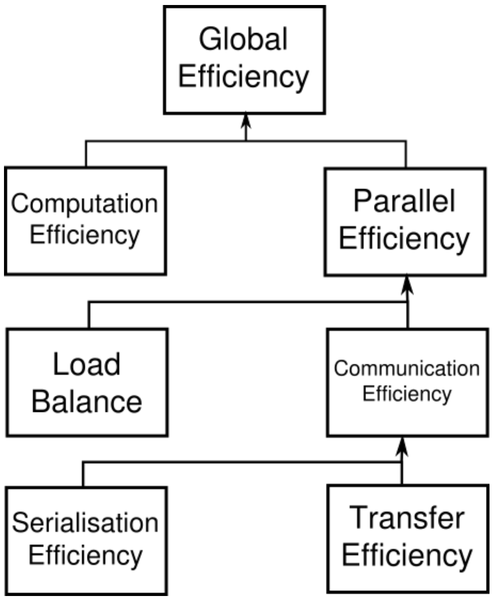


Figure 1: POP metrics hierarchy

These are measured with **Load Balance Efficiency** and **Communication Efficiency**, and PE is defined as the product of these two sub-metrics:

$$\text{PE} = \text{Load Balance} * \text{Communication Efficiency} \tag{2}$$

Load Balance (LB) is computed as the ratio between average useful computation time (across all processes) and maximum useful computation time (also across all processes):

$$\text{LB} = \text{average computation time} \;/\; \text{maximum computation time} \tag{3}$$

Communication Efficiency (CommE) is the maximum across all processes of the ratio between useful computation time and total runtime:

$$\text{CommE} = \text{maximum computation time} \;/\; \text{total runtime} \tag{4}$$

CommE identifies when code is inefficient because it spends a large amount of time communicating rather than performing useful computations. CommE is composed of two additional metrics that reflect two causes of excessive time within communication:

- Processes waiting at communication points for other processes to arrive (i.e. serialisation)
- Processes transferring large amounts of data relative to the network capacity

These are measured using Serialisation Efficiency and Transfer Efficiency. In obtaining these two sub-metrics we first calculate (using the Dimemas simulator) how the code would behave if run on an idealised network where transmission of data takes zero time.

Serialisation Efficiency (SerE) describes any loss of efficiency due to dependencies between processes causing alternating processes to wait:

$$\text{SerE} = \text{maximum computation time on ideal network} \;/\; \text{total runtime on ideal network} \tag{5}$$

Transfer Efficiency (TE) measures inefficiencies due to time in data transfer:

$$\text{TE} = \text{total runtime on ideal network} \;/\; \text{total runtime on real network} \tag{6}$$

These two sub-metrics combine to give **Communication Efficiency**:

$$\text{CommE} = \text{Serialisation Efficiency} * \text{Transfer Efficiency} \tag{7}$$

The final metric in the hierarchy is **Computation Efficiency (CompE)**, which are ratios of total time in useful computation summed over all processes. For strong scaling (i.e. problem size is constant) it is the

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

ratio of total time in useful computation for a reference case (e.g. on 1 process or 1 compute node) to the total time as the number of processes (or nodes) is increased. For CompE to have a value of 1 this time must remain constant regardless of the number of processes.

Insight into possible causes of poor computation scaling can be investigated using metrics devised from processor hardware counter data. Two causes of poor computational scaling are:

- Dividing work over additional processes increases the total computation required
- Using additional processes leads to contention for shared resources

and we investigate these using **Instruction Scaling** and **Instructions Per Cycle (IPC) Scaling**.

Instruction Scaling is the ratio of total number of useful instructions for a reference case (e.g. 1 processor) compared to values when increasing the numbers of processes. A decrease in Instruction Efficiency corresponds to an increase in the total number of instructions required to solve a computational problem.

IPC Scaling compares IPC to the reference, where lower values indicate that rate of computation has slowed. Typical causes for this include decreasing cache hit rate and exhaustion of memory bandwidth, these can leave processes stalled and waiting for data.

4 Performance assessment

This section describes the results obtained by tracing the test cases. The basic analysis of performance metrics should provide a baseline for any following analysis performed after particular optimizations or application updates. Except for the COVO test cases, we present an overview of the application structure, strong scaling charts and the evaluation of the metrics using the performance model described in the previous section.

All the following test cases contain three standard phases - Initialization, main Iterative, and Finalization phase. We omit the Finalization phase in this assessment since its duration is usually very short and it affects the application scaling negligibly.

The strong scaling charts include also three regions that represent a percentage of the ideal (linear) scaling: Green is 100–80 %, orange means 80–0 %, and red indicates speedup less than 1, i.e. the slowdown.

In general, the performance metrics presented in the heat-map tables indicate the area of inefficiency, but finding the root cause of the particular issue requires deeper analysis that is out of the scope of this deliverable.

In this section, when referring to average frequency, the CPU core frequency is meant. Under the POP performance analysis the CPU uncore frequency is not monitored.

4.1 WaLBerla

4.1.1 COVO

Since the size of the problem in the COVO test case is very small, less than 1 second for 5001 time steps in the Iterative phase of plain runs, the runtimes of traced runs are affected by the very low granularity of computations and thus very high tracing overhead. The application structure in Figure 2 also confirms that most of the time is spent in non-blocking communication. Figure 3 shows that the duration of the vast majority of computation bursts is less than 100 μs, despite most of the useful runtime being spent in 0.5–5 s long bursts. The traced run on 1 node (only 36 processes) is 65x slower than a plain run. Figure 4 illustrates in detail the granularity of computations and the time spent in MPI communication in just 2 time steps of the Iterative phase.

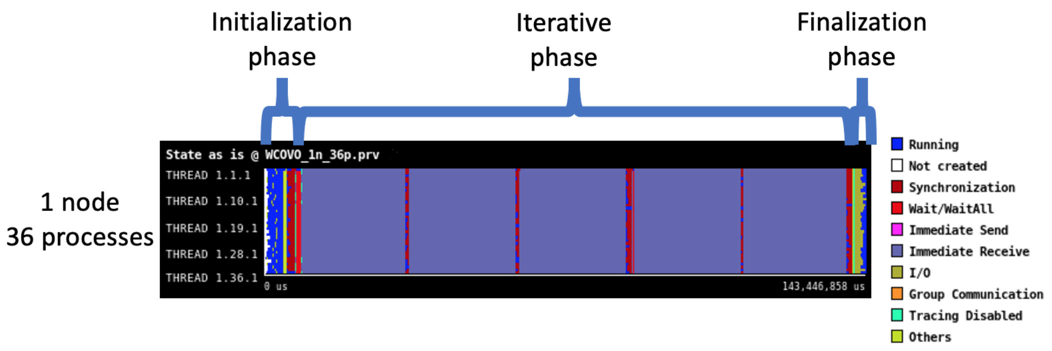


Figure 2: WaLBerla COVO: Application structure.

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

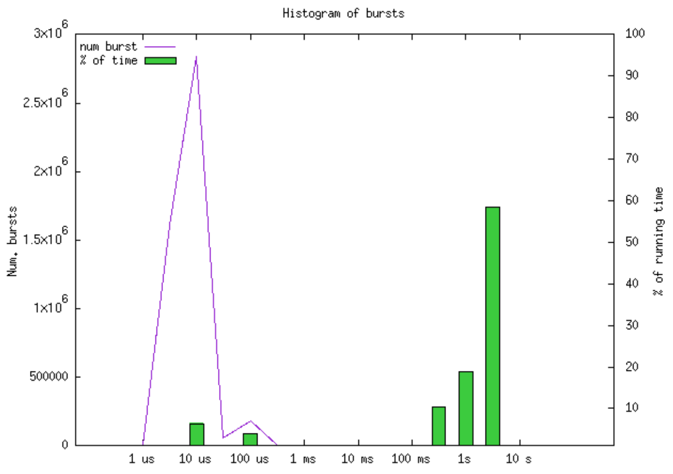


Figure 3: WaLBerla COVO: Granularity of computation bursts in the whole application.

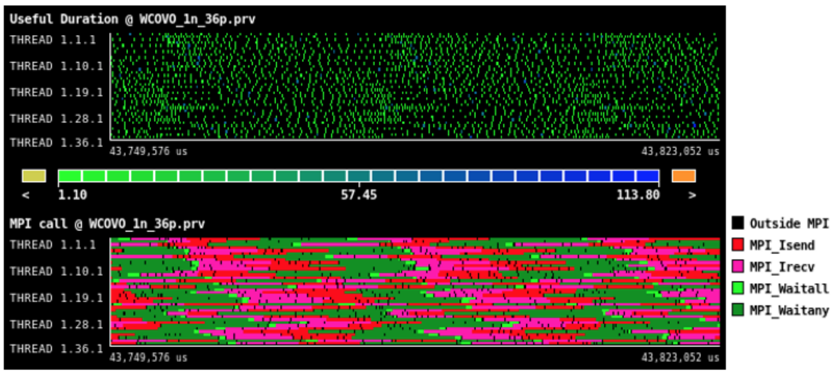


Figure 4: WaLBerla COVO: Detail of useful computations (top) and MPI calls (bottom) in 2 time steps.

Evaluation of the performance metrics is not presented in this case as it would lead to imprecise performance characteristics. Nevertheless, Table 9 shows very good weak scaling of the Iterative phase in the plain runs.

Number of processes	36	72	144	288	576	1152
Elapsed time plain [s]	2.2	2.3	2.3	2.3	2.5	3.4
Elapsed time plain Iterative phase [s]	0.6	0.7	0.7	0.7	0.8	0.9

Table 9: WaLBerla COVO: Weak scaling of plain runs.

4.1.2 Turbulent Channel

The structure of the Turbulent Channel test case in the context of strong scaling is shown in Figure 5. The Running state, which means doing useful computation, is prevailing till 180 processes. Then the computation granularity is reduced so that it is dominated by communication.

The initialization phase does not scale (Figure 6) and the runs with 180 and 360 processes exhibit increased time spent in synchronization state because of load imbalance, bad instruction scaling and mainly significant frequency degradation (Figure 7). Table 10 shows very low average frequencies that could be caused by e.g. waiting for resources, system interruptions, or the process synchronization. The reported frequency values are derived from clock cycles related to a particular application thread. In case that the thread is suspended by the system, the frequency can be artificially lowered and fall below a native core frequency.

Figure 8 shows the perfect scaling of the Iterative phase. We can expect that this use case will scale on even more processes. All the performance efficiencies (Figure 9) are very good with only slight degradation of IPC. In fact, the average IPC is quite low (Figure 11), less than 0.5, being a good candidate for further analysis.

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

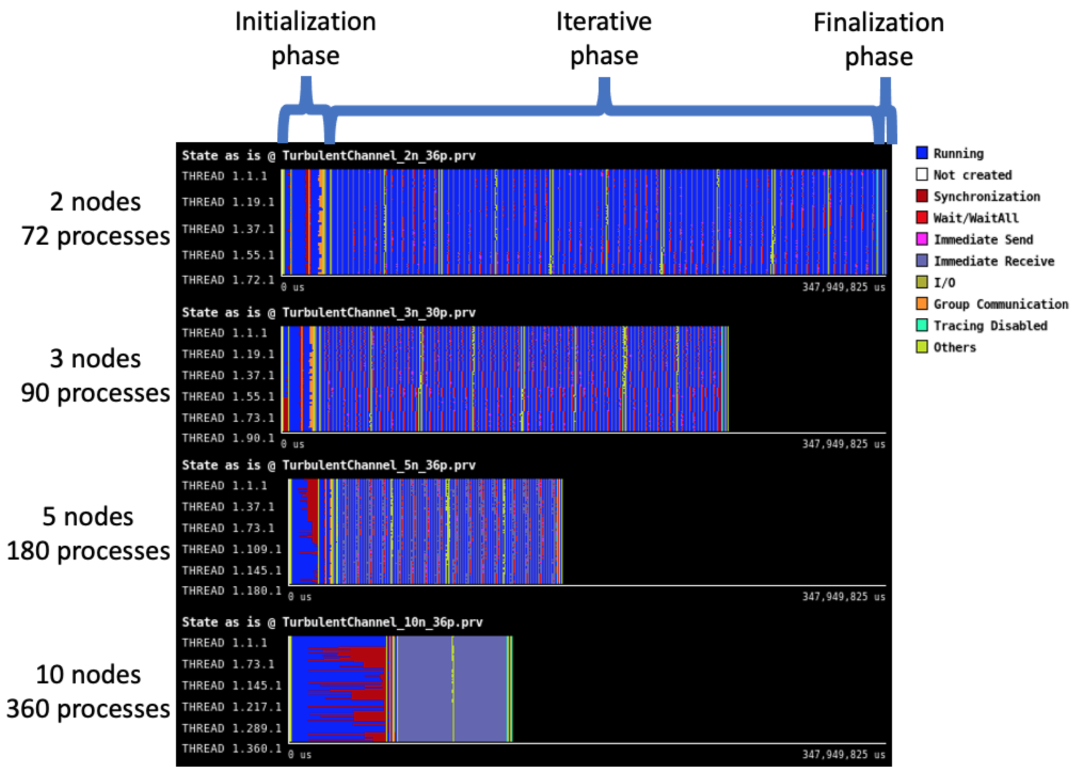


Figure 5: WaLBerla Turbulent Channel: Application structure.

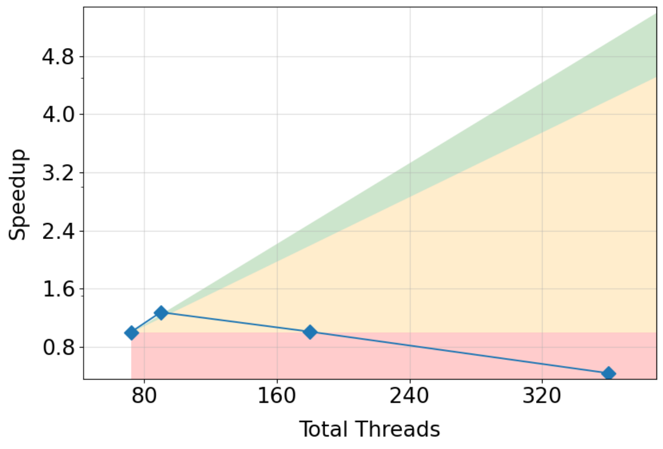


Figure 6: WaLBerla Turbulent Channel Initialization phase: Strong scaling.

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

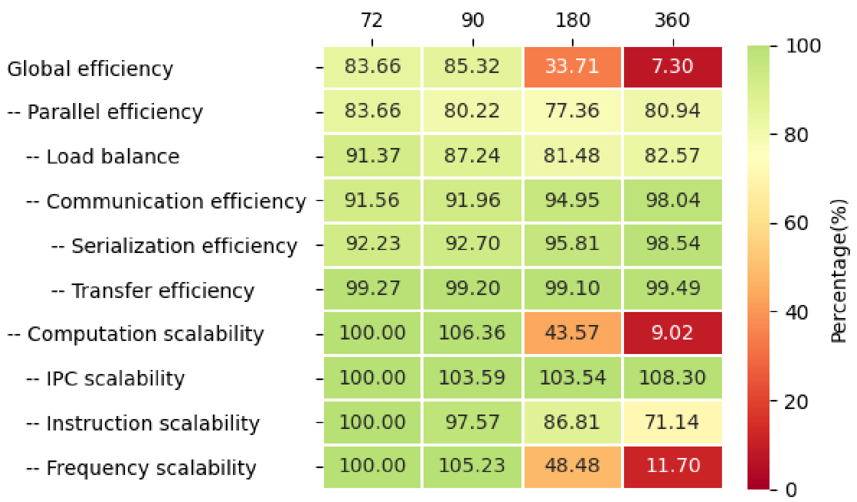


Figure 7: WaLBerla Turbulent Channel Initialization phase: Efficiency metrics.

Number of processes	72	90	180	360
Elapsed time [s]	27.12	21.28	26.93	62.18
Efficiency	1.00	1.02	0.40	0.09
Speedup	1.00	1.27	1.01	0.44
Average IPC	1.58	1.63	1.63	1.71
Average frequency [GHz]	2.49	2.62	1.21	0.29

Table 10: WaLBerla Turbulent Channel Initialization phase: Overview of the Speedup, IPC and Frequency.

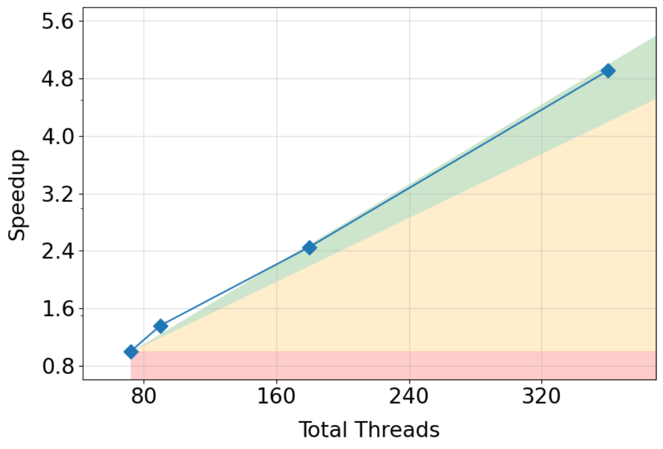
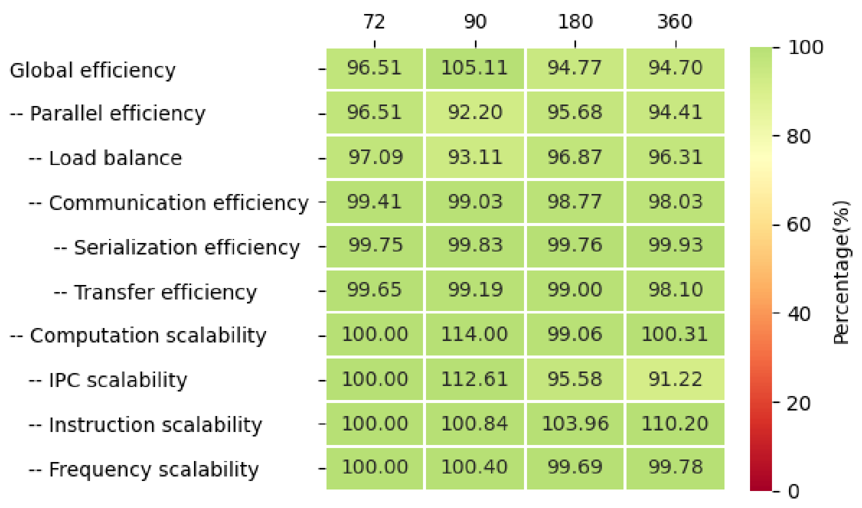


Figure 8: WaLBerla Turbulent Channel Iterative phase: Strong scaling.

Number of processes	72	90	180	360
Elapsed time [s]	314.99	231.37	128.31	64.20
Efficiency	1.00	1.09	0.98	0.98
Speedup	1.00	1.36	2.45	4.91
Average IPC	0.43	0.48	0.41	0.39
Average frequency [GHz]	3.13	3.14	3.12	3.12

Table 11: WaLBerla Turbulent Channel Iterative phase: Overview of the Speedup, IPC and Frequency.

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



	72	90	180	360
Global efficiency	96.51	105.11	94.77	94.70
-- Parallel efficiency	96.51	92.20	95.68	94.41
-- Load balance	97.09	93.11	96.87	96.31
-- Communication efficiency	99.41	99.03	98.77	98.03
-- Serialization efficiency	99.75	99.83	99.76	99.93
-- Transfer efficiency	99.65	99.19	99.00	98.10
-- Computation scalability	100.00	114.00	99.06	100.31
-- IPC scalability	100.00	112.61	95.58	91.22
-- Instruction scalability	100.00	100.84	103.96	110.20
-- Frequency scalability	100.00	100.40	99.69	99.78

Figure 9: WaLBerla Turbulent Channel Iterative phase: Efficiency metrics.

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

4.1.3 Lagoon

The structure of the Lagoon test case (Figure 10) reveals a growing fraction of communication and also significant I/O sections bounding the Iterative phase. During the tracing process, we have experienced high variability in the duration of the I/O sections using the shared Lustre parallel file system. This indicates that the application could be sensitive to I/O workload in shared environments.

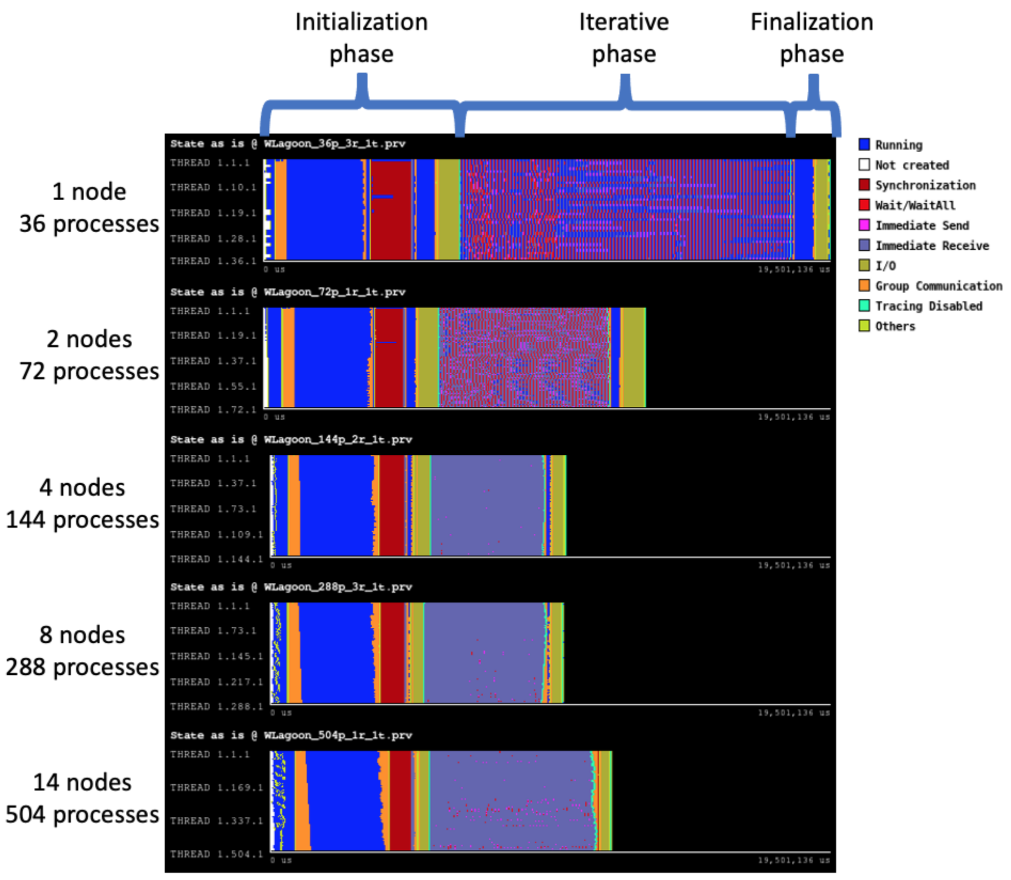


Figure 10: WaLBerla Lagoon: Application structure.

The duration of the Initialization phase is almost constant with scale (Figure 11). This is caused mainly by very poor instruction scaling, but also by constant load imbalance and a small portion of communication serialization (Figure 12). Table 12 shows good values of both IPC and frequency.

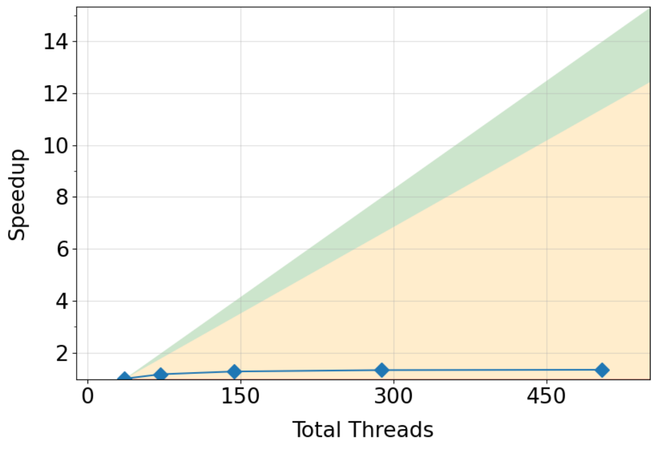
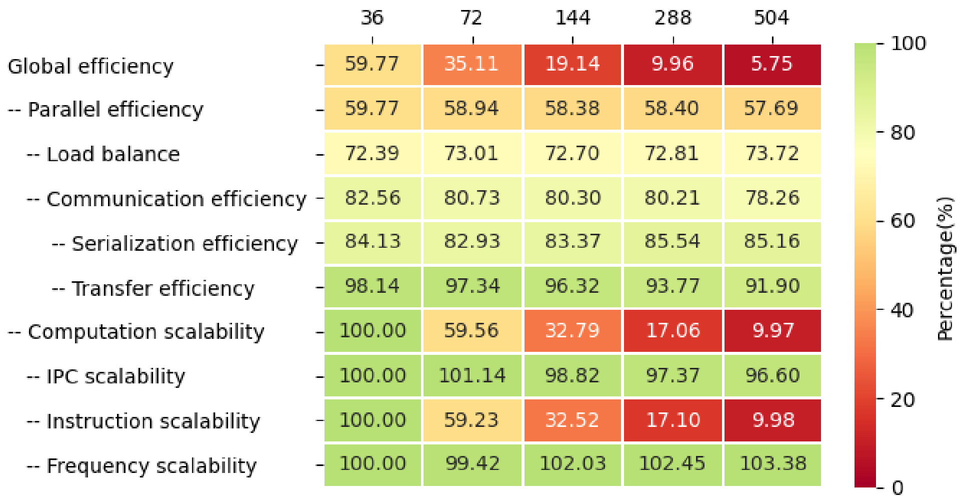


Figure 11: WaLBerla Lagoon Initialization phase: Strong scaling.

The Iterative phase of the Lagoon test case seems to scale well only to 72 processes (Figure 13). But this is not the case for plain runs, as shown in Table 13. The growing difference between plain and traced elapsed

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



	36	72	144	288	504
Global efficiency	59.77	35.11	19.14	9.96	5.75
-- Parallel efficiency	59.77	58.94	58.38	58.40	57.69
-- Load balance	72.39	73.01	72.70	72.81	73.72
-- Communication efficiency	82.56	80.73	80.30	80.21	78.26
-- Serialization efficiency	84.13	82.93	83.37	85.54	85.16
-- Transfer efficiency	98.14	97.34	96.32	93.77	91.90
-- Computation scalability	100.00	59.56	32.79	17.06	9.97
-- IPC scalability	100.00	101.14	98.82	97.37	96.60
-- Instruction scalability	100.00	59.23	32.52	17.10	9.98
-- Frequency scalability	100.00	99.42	102.03	102.45	103.38

Figure 12: WaLBerla Lagoon Initialization phase: Efficiency metrics.

Number of processes	36	72	144	288	504
Elapsed time [s]	6.34	5.40	4.95	4.76	4.71
Efficiency	1.00	0.59	0.32	0.17	0.10
Speedup	1.00	1.17	1.28	1.33	1.35
Average IPC	1.93	1.96	1.91	1.88	1.87
Average frequency [GHz]	3.09	3.07	3.15	3.16	3.19

Table 12: WaLBerla Lagoon Initialization phase: Overview of the Speedup, IPC and Frequency.

times - the tracing overhead - is suggesting that the computation granularity is too low. This is also indicated by the decrease of the transfer efficiency shown in Figure 14, which means the communication is getting to be a dominant component. The low granularity of this test case with the given problem size would most probably cause scaling degradation for runs with more than 504 processes even without tracing overhead. Though, for the used number of processes, the plain runs show very good scalability. Besides the transfer efficiency, the table also contains a temporal drop in load balance for 144 and 288 that is not clear and needs further investigation.

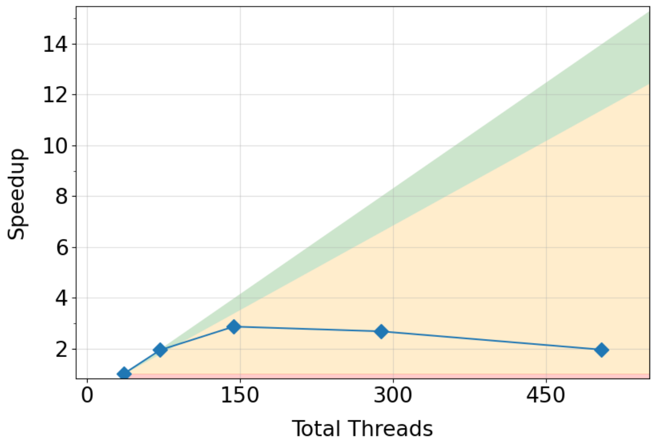


Figure 13: WaLBerla Lagoon Iterative phase: Strong scaling.

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

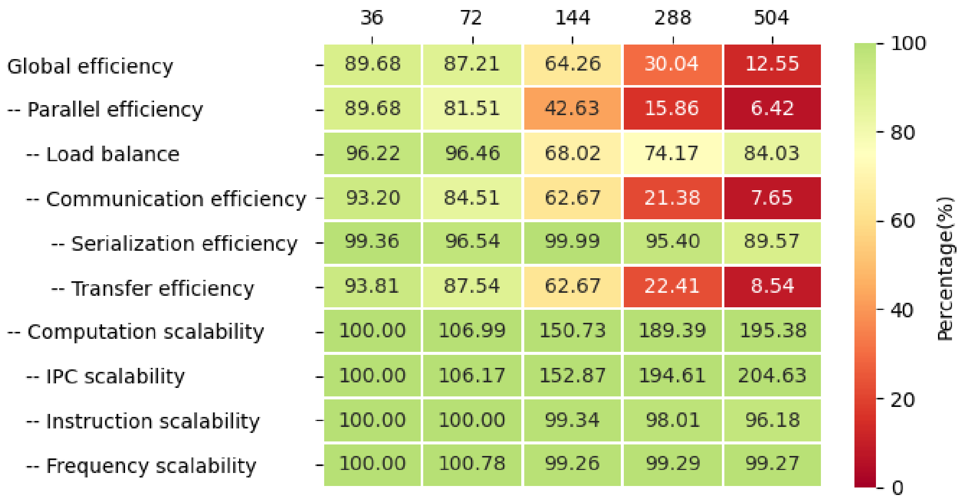


Figure 14: WaLBerla Lagoon Iterative phase: Efficiency metrics.

Number of processes	36	72	144	288	504
Elapsed time plain [s]	11.4	5.8	3	1.7	0.9
Elapsed time [s]	11.43	5.88	3.99	4.27	5.83
Efficiency	1.00	0.97	0.72	0.33	0.14
Speedup	1.00	1.94	2.87	2.68	1.96
Average IPC	0.46	0.48	0.70	0.89	0.93
Average frequency [GHz]	2.70	2.73	2.68	2.69	2.68

Table 13: WaLBerla Lagoon Iterative phase: Overview of the Speedup, IPC and Frequency.

4.1.4 S2A

The structure of the S2A test case (Figure 15) is quite similar to the Lagoon case. We have observed the high sensitivity for the I/O workload of the file system too. Again, the Initialization phase does not scale as evidenced in Figure 16. The reasons are the same - instruction scaling, load balance, and serialization to a lesser extent (Figure 17). The values of IPC (Table 14) are slightly better than the Initialization phase of Lagoon when exceeding 2 in all runs.

Number of processes	36	72	144	288	576	590
Elapsed time [s]	8.15	6.92	6.55	6.00	6.00	6.38
Efficiency	1.00	0.59	0.31	0.17	0.08	0.08
Speedup	1.00	1.18	1.24	1.36	1.36	1.28
Average IPC	2.06	2.11	2.08	2.06	2.05	2.05
Average frequency [GHz]	3.07	3.05	3.11	3.13	3.14	3.14

Table 14: WaLBerla S2A Initialization phase: Overview of the Speedup, IPC and Frequency.

The Iterative phase of S2A does not scale perfectly even for the plain runs as reported in Table 14, but the scaling degradation presented in Figure 18 is again skewed by the lower granularity thus higher tracing overhead for the large number of processes. Nevertheless, Figure 19 identifies a severe issue in load balance. The computation workload distribution efficiency goes from quite low 60.5 % down to only 23.2 %.

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

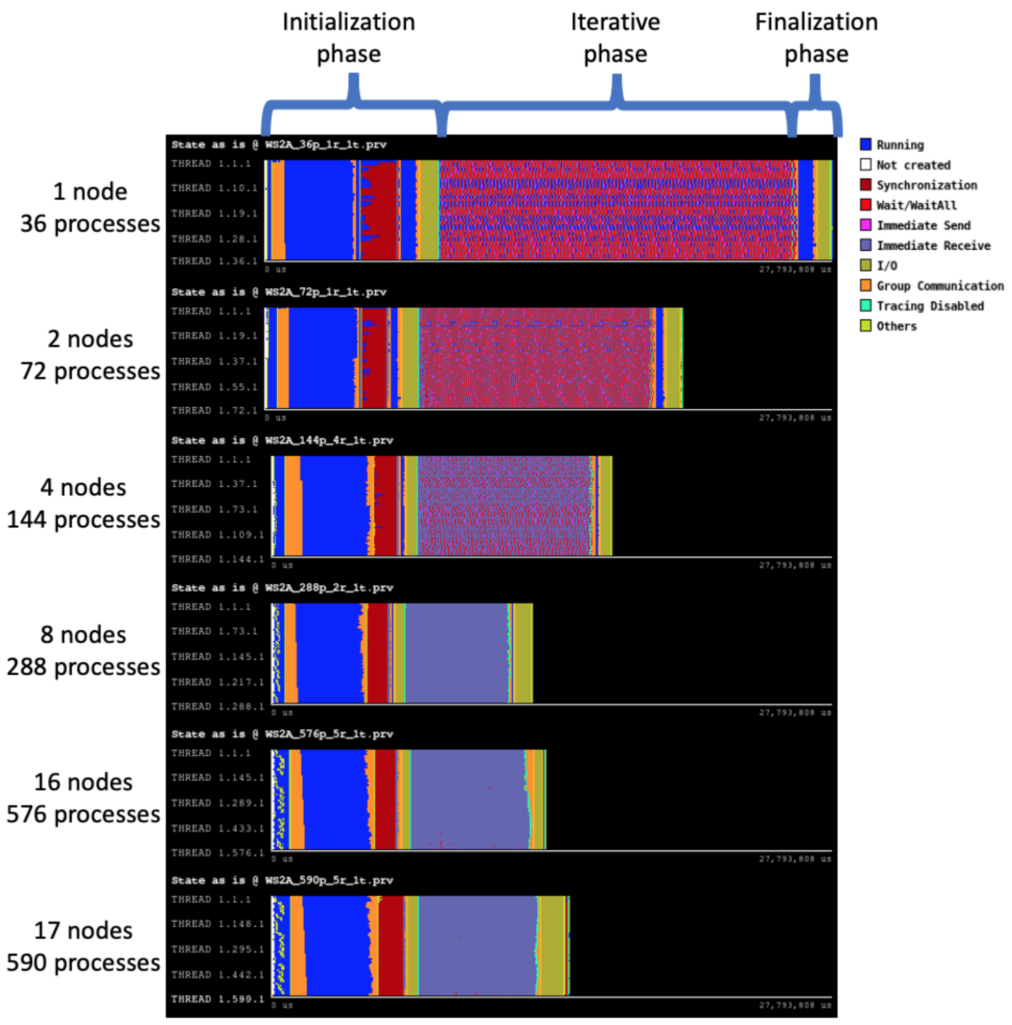


Figure 15: WaLBerla S2A: Application structure.

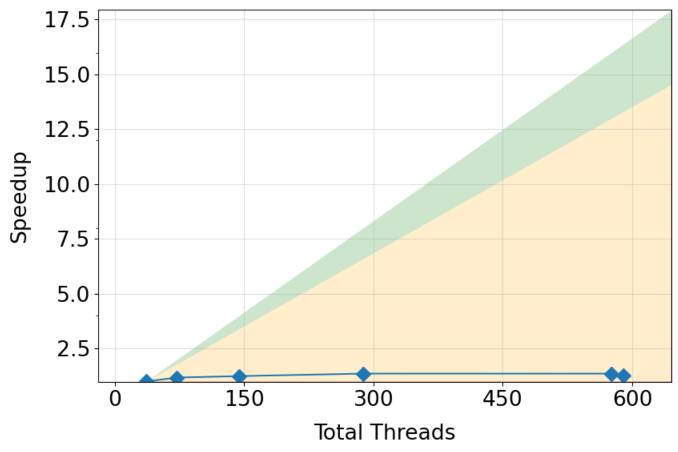
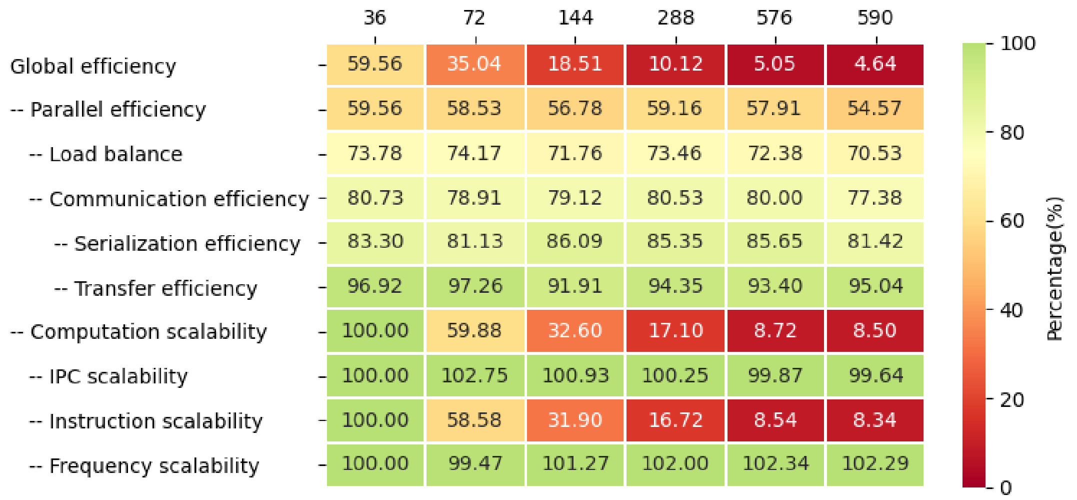


Figure 16: WaLBerla S2A Initialization phase: Strong scaling.

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



	36	72	144	288	576	590
Global efficiency	59.56	35.04	18.51	10.12	5.05	4.64
-- Parallel efficiency	59.56	58.53	56.78	59.16	57.91	54.57
-- Load balance	73.78	74.17	71.76	73.46	72.38	70.53
-- Communication efficiency	80.73	78.91	79.12	80.53	80.00	77.38
-- Serialization efficiency	83.30	81.13	86.09	85.35	85.65	81.42
-- Transfer efficiency	96.92	97.26	91.91	94.35	93.40	95.04
-- Computation scalability	100.00	59.88	32.60	17.10	8.72	8.50
-- IPC scalability	100.00	102.75	100.93	100.25	99.87	99.64
-- Instruction scalability	100.00	58.58	31.90	16.72	8.54	8.34
-- Frequency scalability	100.00	99.47	101.27	102.00	102.34	102.29

Figure 17: WaLBerla S2A Initialization phase: Efficiency metrics.

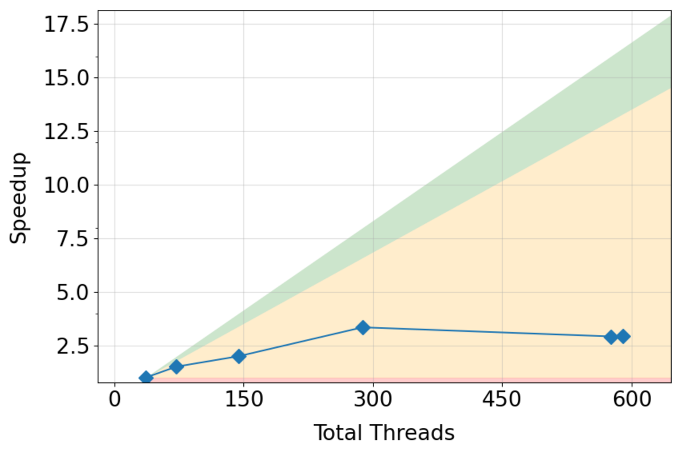
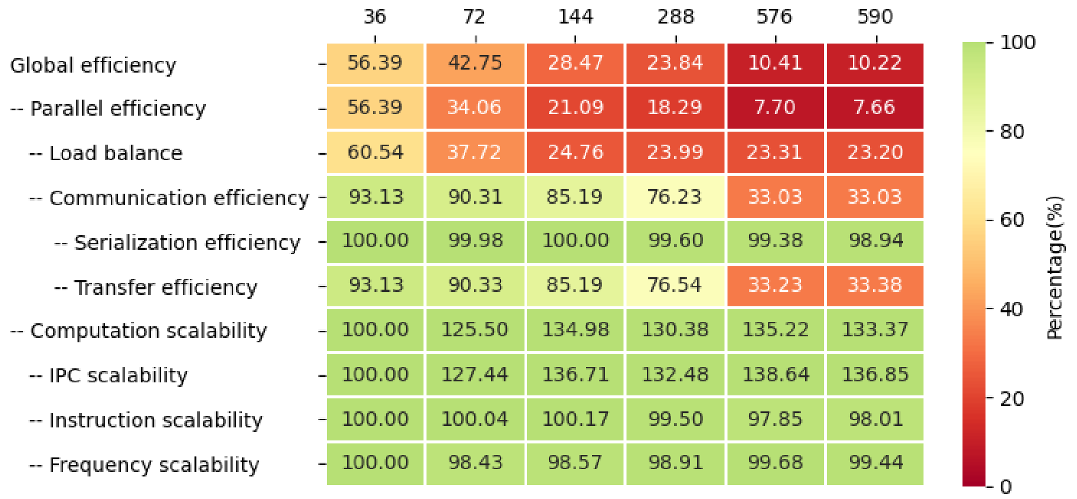


Figure 18: WaLBerla S2A Iterative phase: Strong scaling.



	36	72	144	288	576	590
Global efficiency	56.39	42.75	28.47	23.84	10.41	10.22
-- Parallel efficiency	56.39	34.06	21.09	18.29	7.70	7.66
-- Load balance	60.54	37.72	24.76	23.99	23.31	23.20
-- Communication efficiency	93.13	90.31	85.19	76.23	33.03	33.03
-- Serialization efficiency	100.00	99.98	100.00	99.60	99.38	98.94
-- Transfer efficiency	93.13	90.33	85.19	76.54	33.23	33.38
-- Computation scalability	100.00	125.50	134.98	130.38	135.22	133.37
-- IPC scalability	100.00	127.44	136.71	132.48	138.64	136.85
-- Instruction scalability	100.00	100.04	100.17	99.50	97.85	98.01
-- Frequency scalability	100.00	98.43	98.57	98.91	99.68	99.44

Figure 19: WaLBerla S2A Iterative phase: Efficiency metrics.

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

Number of processes	36	72	144	288	576	590
Elapsed time plain [s]	17.1	10.7	7.6	4.3	2.2	2.1
Elapsed time [s]	17.67	11.65	8.75	5.22	5.98	5.95
Efficiency	1.00	0.76	0.50	0.42	0.18	0.18
Speedup	1.00	1.52	2.02	3.38	2.95	2.97
Average IPC	0.75	0.96	1.03	1.00	1.05	1.03
Average frequency [GHz]	2.80	2.75	2.76	2.77	2.79	2.78

Table 15: WaLBerla S2A Iterative phase: Overview of the Speedup, IPC and Frequency.

4.2 LaBS

4.2.1 COVO

As the LaBS COVO test case is designed in a similar manner as the WaLBerla COVO case, the traced runtime is again affected by the very high tracing overhead caused by the low granularity (Figure 21), despite very long total elapsed time. Figure 22 shows that there are few longer computation bursts taking approx. 2000 μs in the Iterative phase, but still the most of them are very short, less than 10 μs, causing the overhead. The traced run on 1 node (only 36 processes) is 5.9x slower than a plain run. Evaluation of the performance metrics is therefore again omitted since it would lead to imprecise performance characteristics. The structure in Figure 20 also confirms that the tracing of the Finalization phase was intentionally disabled as we omit its analysis.

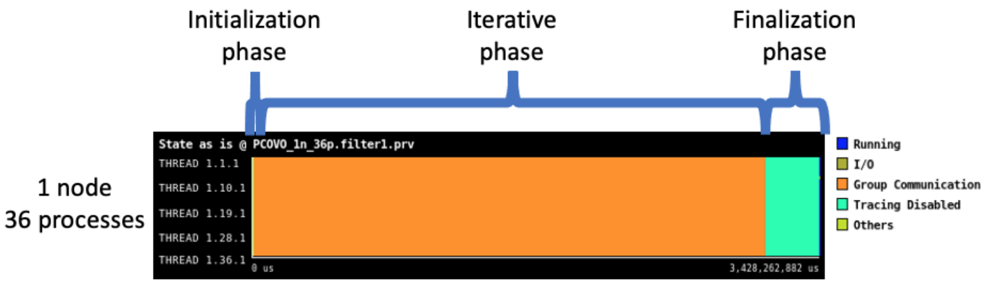


Figure 20: LaBS COVO: Application structure.

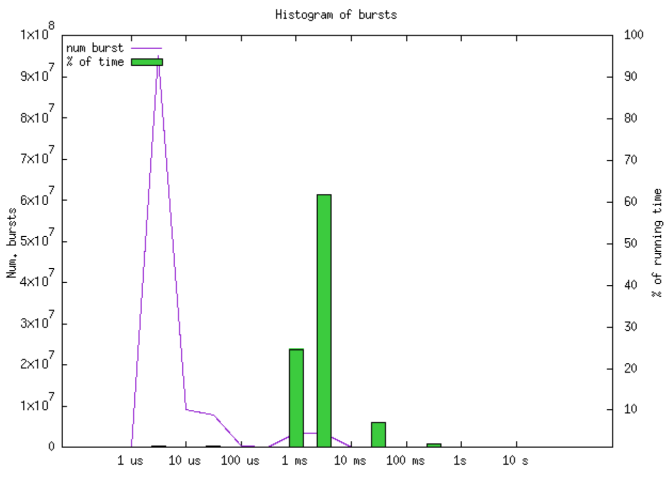


Figure 21: LaBS COVO: Granularity of computation bursts in the whole application.

Table 16 suggests decent weak scaling of the Iterative phase in the plain runs. Moreover, it also reveals a non-iterative phase with a significant duration that is even growing with scale.

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

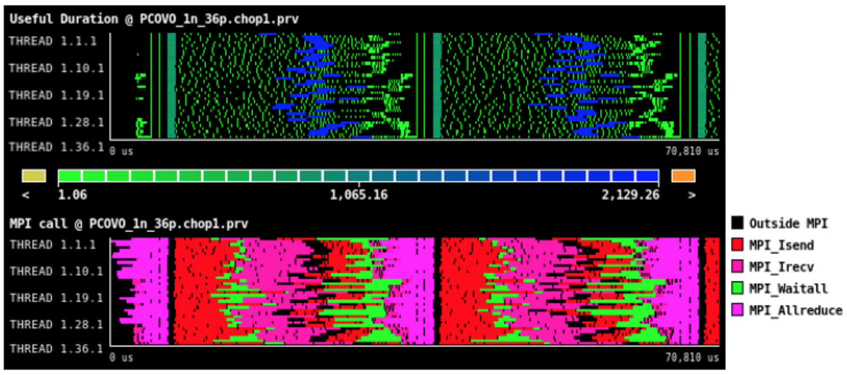


Figure 22: LaBS COVO: Detail of useful computations and MPI calls in 2 iterations.

Number of processes	36	72	144	288	576	1152
Elapsed time plain [s]	581.7	786	1257.7	2235.4	4167.3	7598.6
Elapsed time plain Iterative phase [s]	336.6	342.1	363.5	385	407.9	442.1

Table 16: LaBS COVO: Weak scaling of plain runs.

4.2.2 Turbulent Channel

The structure of the Turbulent Channel test case depicts Figure 23, which shows relatively long Initialization phase, a short Iterative phase given by the limited number of time steps defined, and a constant Finalization phase that was not traced.

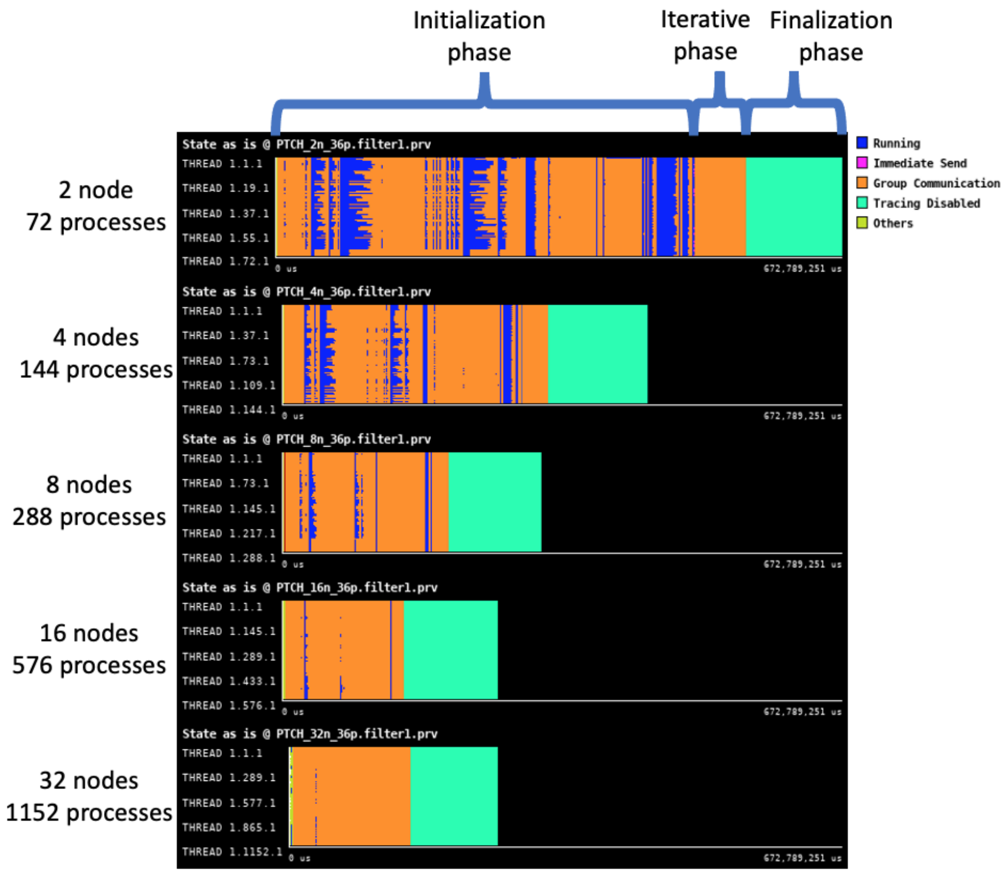


Figure 23: LaBS Turbulent Channel: Application structure.

The Initialization phase stops scaling after 288 processes (Figure 24). There are multiple factors limiting the scaling (Figure 25), including the load balance degrading from 82 % to 60 %, serialization efficiency oscillating around 80 %, and for the largest run on 1152 processes also the transfer efficiency and instruction scaling

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

drops. The average IPC and frequency values are also slightly decreasing with scale (Table 17).

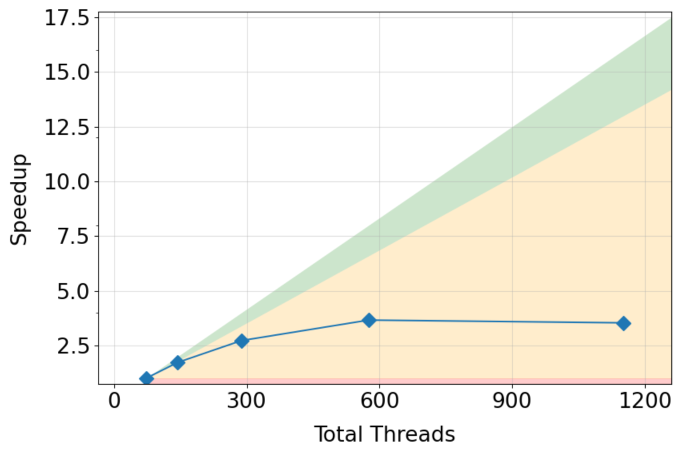
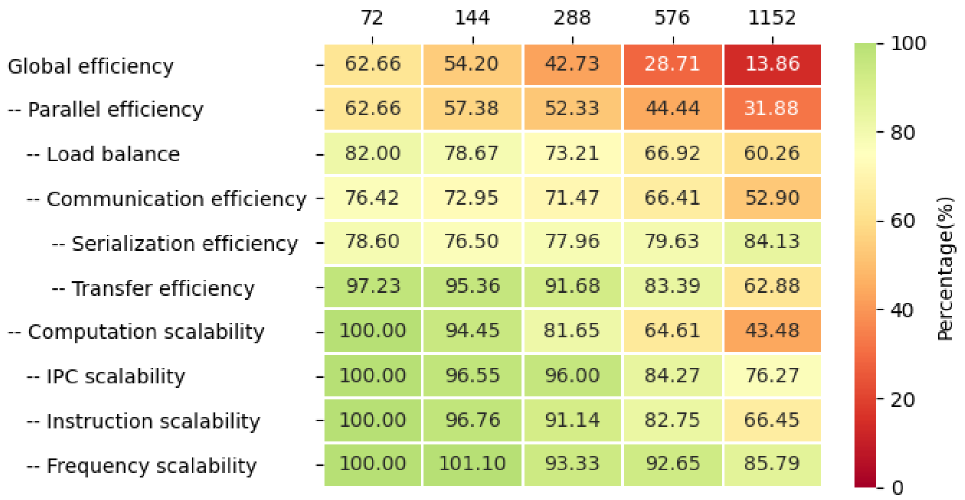


Figure 24: LaBS Turbulent Channel Initialization phase: Strong Scaling.



	72	144	288	576	1152
Global efficiency	62.66	54.20	42.73	28.71	13.86
-- Parallel efficiency	62.66	57.38	52.33	44.44	31.88
-- Load balance	82.00	78.67	73.21	66.92	60.26
-- Communication efficiency	76.42	72.95	71.47	66.41	52.90
-- Serialization efficiency	78.60	76.50	77.96	79.63	84.13
-- Transfer efficiency	97.23	95.36	91.68	83.39	62.88
-- Computation scalability	100.00	94.45	81.65	64.61	43.48
-- IPC scalability	100.00	96.55	96.00	84.27	76.27
-- Instruction scalability	100.00	96.76	91.14	82.75	66.45
-- Frequency scalability	100.00	101.10	93.33	92.65	85.79

Figure 25: LaBS Turbulent Channel Initialization phase: Efficiency metrics.

Number of processes	72	144	288	576	1152
Elapsed time [s]	498.30	288.07	182.71	135.94	140.80
Efficiency	1.00	0.86	0.68	0.46	0.22
Speedup	1.00	1.73	2.73	3.67	3.54
Average IPC	1.05	1.01	1.01	0.88	0.80
Average frequency [GHz]	3.17	3.20	2.96	2.94	2.72

Table 17: LaBS Turbulent Channel Initialization phase: Overview of the Speedup, IPC and Frequency.

The Iterative phase scales well up to 288 processes (Figure 26), then the scaling degrades mainly due to transfer efficiency and for larger scales partially by decreasing the serial efficiency (Figure 27). Both average IPC and frequency were almost constant with good values around 1.25 and 3.17 respectively as reported in Table 18. For this analysis, the whole Iterative phase of 100 time steps was used.

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

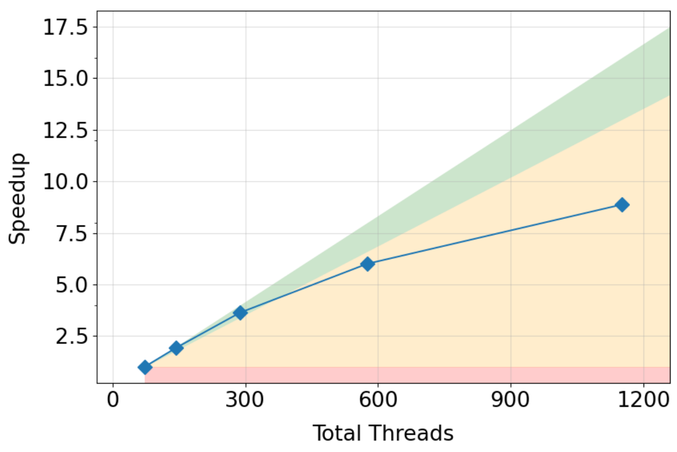
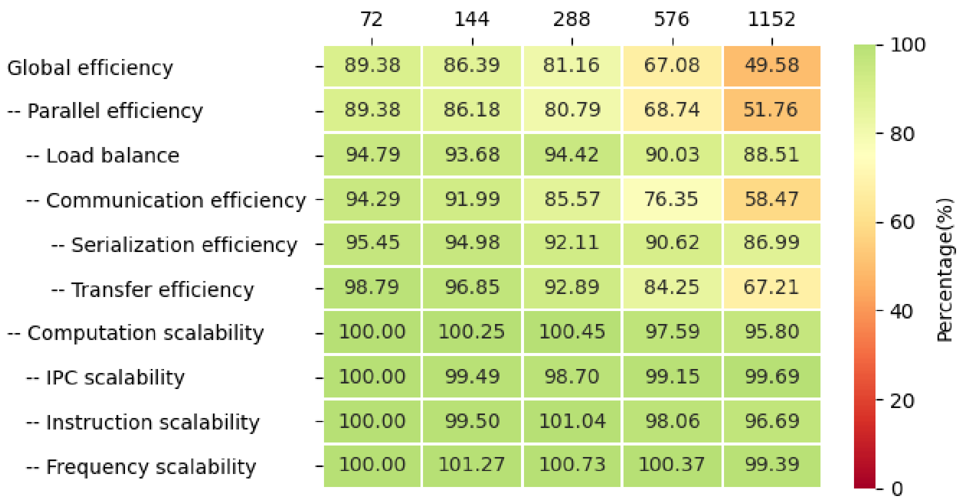


Figure 26: LaBS Turbulent Channel Iterative phase: Strong Scaling.



	72	144	288	576	1152
Global efficiency	89.38	86.39	81.16	67.08	49.58
-- Parallel efficiency	89.38	86.18	80.79	68.74	51.76
-- Load balance	94.79	93.68	94.42	90.03	88.51
-- Communication efficiency	94.29	91.99	85.57	76.35	58.47
-- Serialization efficiency	95.45	94.98	92.11	90.62	86.99
-- Transfer efficiency	98.79	96.85	92.89	84.25	67.21
-- Computation scalability	100.00	100.25	100.45	97.59	95.80
-- IPC scalability	100.00	99.49	98.70	99.15	99.69
-- Instruction scalability	100.00	99.50	101.04	98.06	96.69
-- Frequency scalability	100.00	101.27	100.73	100.37	99.39

Figure 27: LaBS Turbulent Channel Iterative phase: Efficiency metrics.

Number of processes	72	144	288	576	1152
Elapsed time [s]	59.57	30.81	16.40	9.92	6.71
Efficiency	1.00	0.97	0.91	0.75	0.55
Speedup	1.00	1.93	3.63	6.00	8.88
Average IPC	1.26	1.25	1.24	1.25	1.25
Average frequency [GHz]	3.16	3.20	3.18	3.17	3.14

Table 18: LaBS Turbulent Channel Iterative phase: Overview of the Speedup, IPC and Frequency.

4.2.3 Lagoon

Due to the computational complexity and the size of the problem in the Lagoon test case, the tracing produced a large amount of data. In order to handle such large trace files, we have focused the performance analysis only on the Initialization phase and 20 iterations out of the 1000 performed in the Iterative phase. The application structure in Figure 28 shows the Initialization phase more than twice as long as the Iterative one, and again the disabled tracing of the short Finalization phase. Most of the communication time is spent in group communication.

The Initialization phase does not scale well with more than 144 processes (Figure 29). There are multiple reasons, as indicated by metrics in Figure 30. The low load balance efficiency that holds around 70 % for all runs is the main limiting factor for the smaller runs. With growing scale, the serialization efficiency and instruction scaling is degrading down to 54 % and 50 % respectively. There is also a significant drop from 83.6 % to less than 60 % of the transfer efficiency on 1152 processes. Table 19 shows quite high average

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

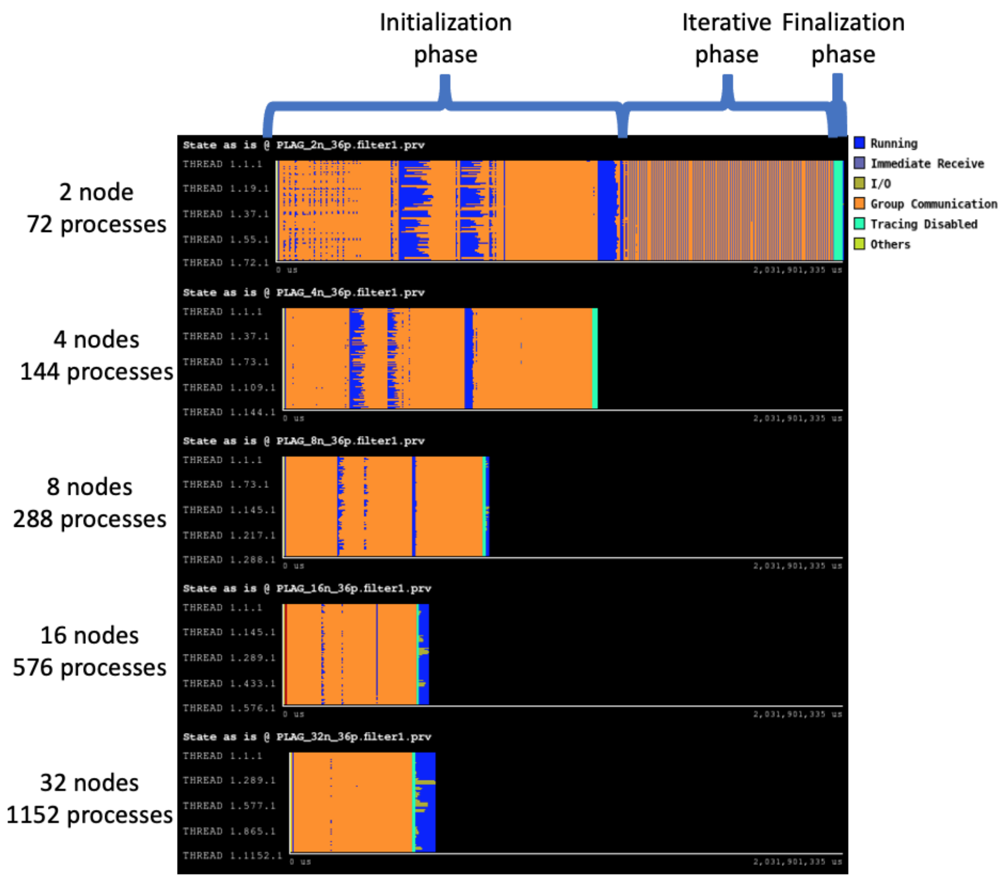


Figure 28: LaBS Lagoon: Application structure.

frequencies, up to 3.24 GHz, and slightly increasing IPC.

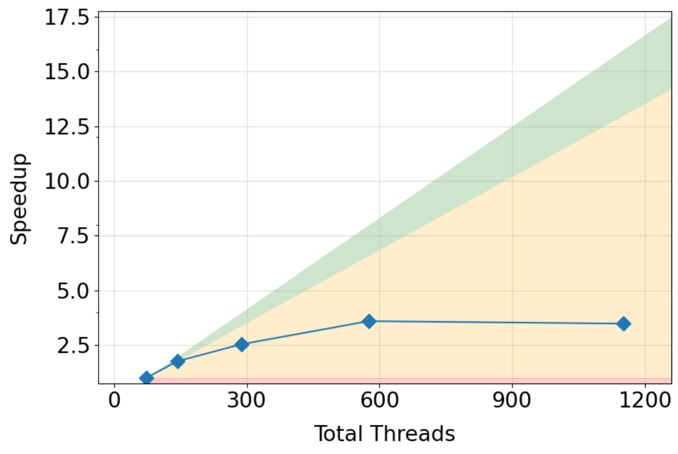


Figure 29: LaBS Lagoon Initialization phase: Strong Scaling.

Number of processes	72	144	288	576	1152	1152
Elapsed time [s]	1259.75	711.49	495.45	350.02	361.40	331.12
Efficiency	1.00	0.89	0.64	0.45	0.22	0.15
Speedup	1.00	1.77	2.54	3.60	3.49	4.75
Average IPC	0.91	0.92	0.93	0.95	1.03	1.19
Average frequency [GHz]	3.24	3.23	3.20	3.16	3.12	3.04

Table 19: LaBS Lagoon Initialization phase: Overview of the Speedup, IPC and Frequency.

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

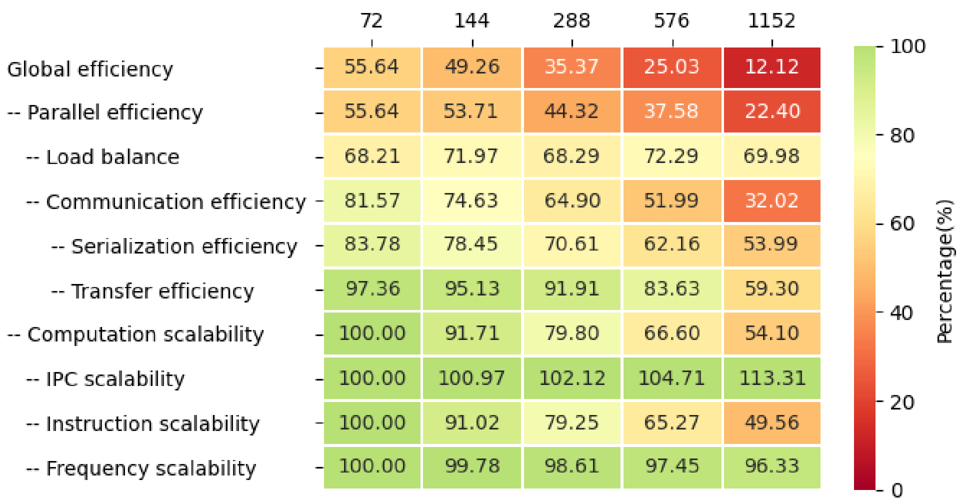


Figure 30: LaBS Lagoon Initialization phase: Efficiency metrics.

The very good scaling of the Iterative phase (Figure 31) on the beginning of the scale is then disrupted by the increasing load imbalance, see Figure 32. The communication efficiency is decreased by not negligible level of serialization in all runs, but the largest run is affected by the drop of the transfer efficiency from 87 % to 73 %. Table 20 reports slightly increasing IPC and decreasing though still high frequency at the same time.

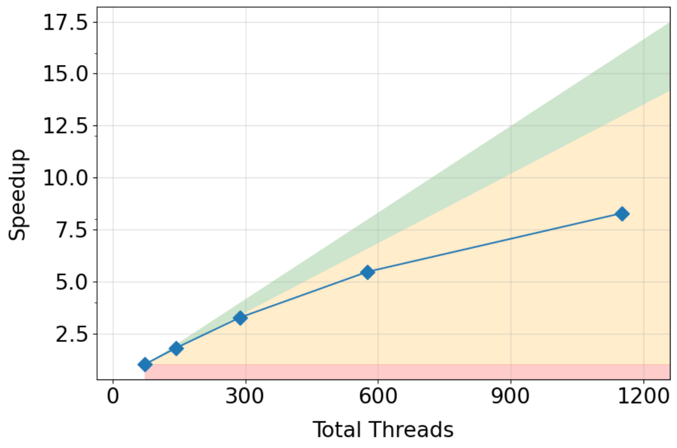
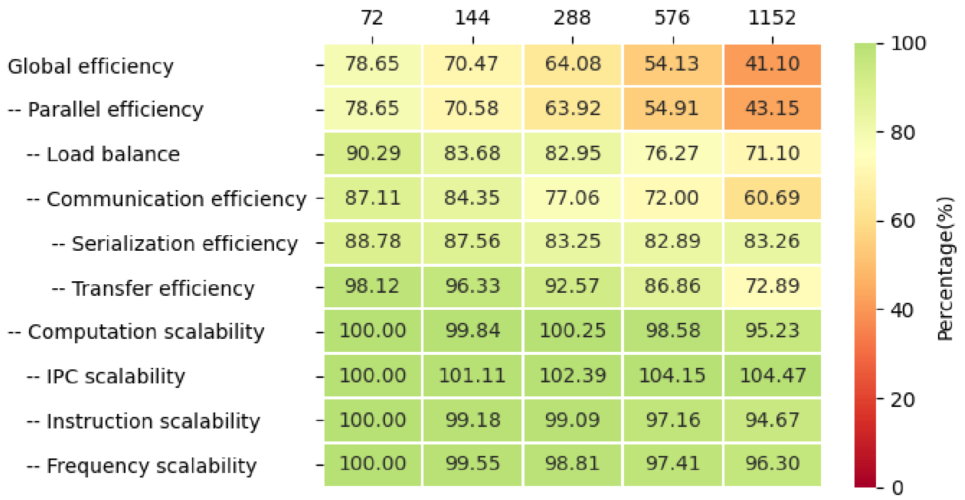


Figure 31: LaBS Lagoon Iterative phase: Strong Scaling.

Number of processes	72	144	288	576	1152
Elapsed time plain [s]	750.5	405.8	215.4	120.1	69.4
Elapsed time [s]	754.3	411.2	229.6	133.5	89.6
Elapsed time 20 iter. [s]	14.85	8.29	4.56	2.70	1.78
Efficiency	1.00	0.90	0.81	0.69	0.52
Speedup	1.00	1.79	3.26	5.51	8.36
Average IPC	1.09	1.10	1.11	1.13	1.14
Average frequency [GHz]	3.26	3.24	3.22	3.17	3.14

Table 20: LaBS Lagoon Iterative phase: Overview of the Speedup, IPC and Frequency.

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



	72	144	288	576	1152
Global efficiency	78.65	70.47	64.08	54.13	41.10
-- Parallel efficiency	78.65	70.58	63.92	54.91	43.15
-- Load balance	90.29	83.68	82.95	76.27	71.10
-- Communication efficiency	87.11	84.35	77.06	72.00	60.69
-- Serialization efficiency	88.78	87.56	83.25	82.89	83.26
-- Transfer efficiency	98.12	96.33	92.57	86.86	72.89
-- Computation scalability	100.00	99.84	100.25	98.58	95.23
-- IPC scalability	100.00	101.11	102.39	104.15	104.47
-- Instruction scalability	100.00	99.18	99.09	97.16	94.67
-- Frequency scalability	100.00	99.55	98.81	97.41	96.30

Figure 32: LaBS Lagoon Iterative phase: Efficiency metrics.

4.2.4 S2A

The structure of the S2A test case, depicted in Figure 33, is quite similar to the Lagoon test case. Again, the Initialization phase is a little longer than the Iterative one in this problem configuration. It seems there is a larger portion of the non-blocking point to point communication over the group communication in the Iterative phase.

Again, The Initialization phase does not scale (Figure 34), the causes are just like the Lagoon Initialization, but even more severe for the most metrics (Figure 35). The load balance goes down to 52.5 %, serialization efficiency degradation stops at 63.7 %, and instruction scaling at less than 37 %. There is also the drop of transfer efficiency from 85 % to 68 % in the largest run. The average IPC is quite constant around 1.19 with an average frequency high above 3 GHz for most of the runs, see Table 21.

Number of processes	36	72	144	288	576	1152
Elapsed time [s]	1571.35	1012.75	628.83	408.35	294.68	331.12
Efficiency	1.00	0.78	0.62	0.48	0.33	0.15
Speedup	1.00	1.55	2.50	3.85	5.33	4.75
Average IPC	1.19	1.19	1.17	1.16	1.23	1.19
Average frequency [GHz]	3.25	3.17	3.22	3.20	2.99	3.04

Table 21: LaBS S2A Initialization phase: Overview of the Speedup, IPC and Frequency.

Number of processes	36	72	144	288	576	1152
Elapsed time plain [s]	1036.4	567.7	311.4	164.2	93.7	56.7
Elapsed time [s]	1034.9	580.7	319.6	174.4	105.6	96.2
Elapsed time 20 iter. [s]	20.17	11.41	6.28	3.46	2.01	1.80
Efficiency	1.00	0.88	0.80	0.73	0.63	0.35
Speedup	1.00	1.77	3.21	5.82	10.02	11.19
Average IPC	1.04	1.07	1.07	1.08	1.11	1.08
Average frequency [GHz]	3.27	3.20	3.25	3.20	3.04	3.06

Table 22: LaBS S2A Iterative phase: Overview of the Speedup, IPC and Frequency.

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

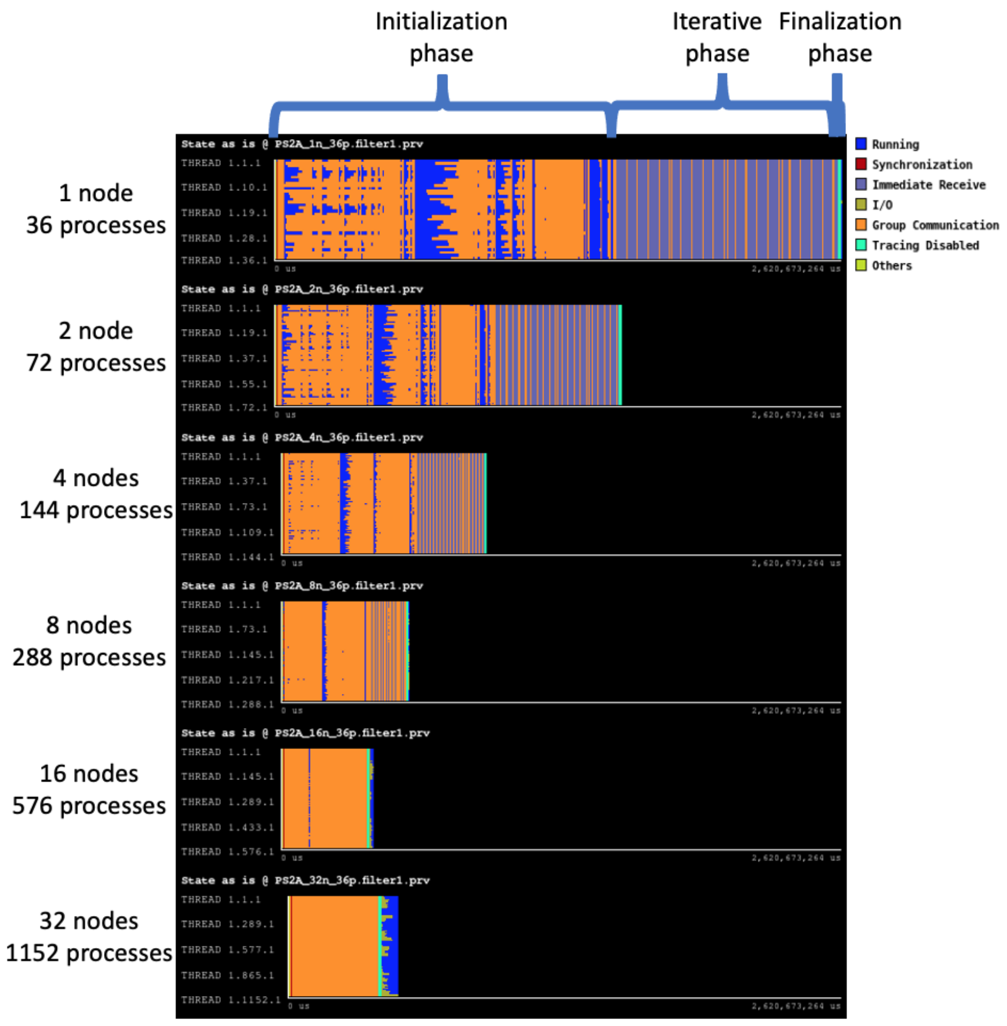


Figure 33: LaBS S2A: Application structure.

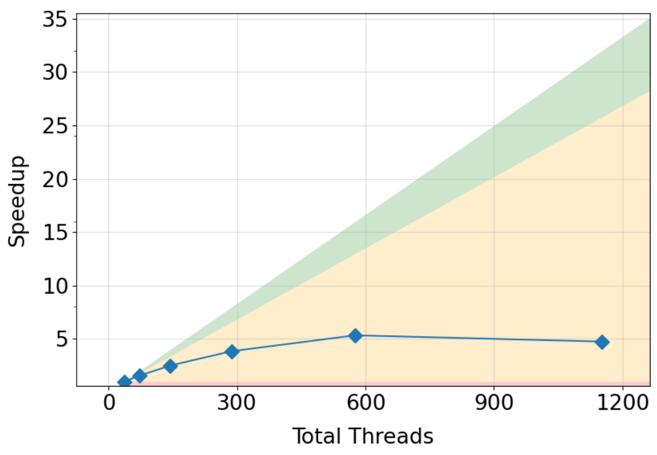
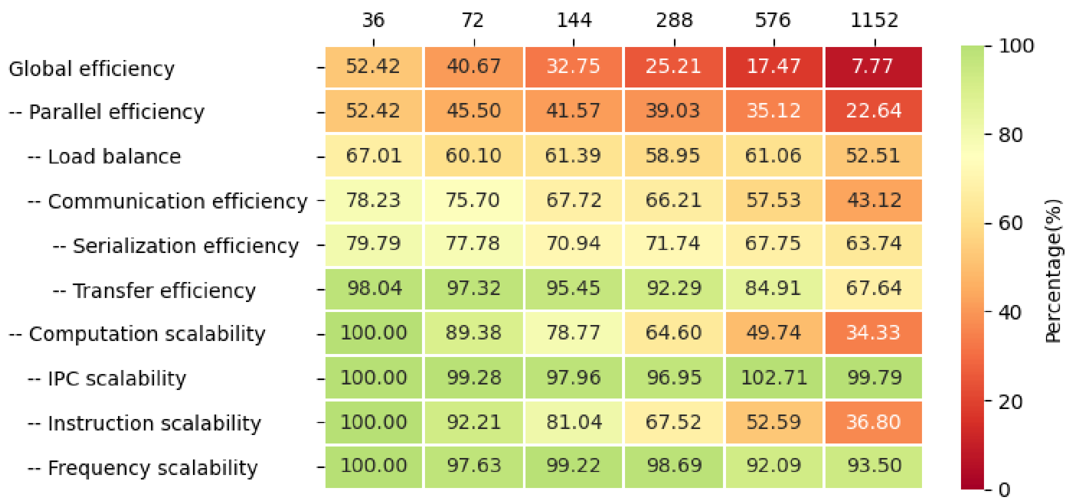


Figure 34: LaBS S2A Initialization phase: Strong Scaling.

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



	36	72	144	288	576	1152
Global efficiency	52.42	40.67	32.75	25.21	17.47	7.77
-- Parallel efficiency	52.42	45.50	41.57	39.03	35.12	22.64
-- Load balance	67.01	60.10	61.39	58.95	61.06	52.51
-- Communication efficiency	78.23	75.70	67.72	66.21	57.53	43.12
-- Serialization efficiency	79.79	77.78	70.94	71.74	67.75	63.74
-- Transfer efficiency	98.04	97.32	95.45	92.29	84.91	67.64
-- Computation scalability	100.00	89.38	78.77	64.60	49.74	34.33
-- IPC scalability	100.00	99.28	97.96	96.95	102.71	99.79
-- Instruction scalability	100.00	92.21	81.04	67.52	52.59	36.80
-- Frequency scalability	100.00	97.63	99.22	98.69	92.09	93.50

Figure 35: LaBS S2A Initialization phase: Efficiency metrics.

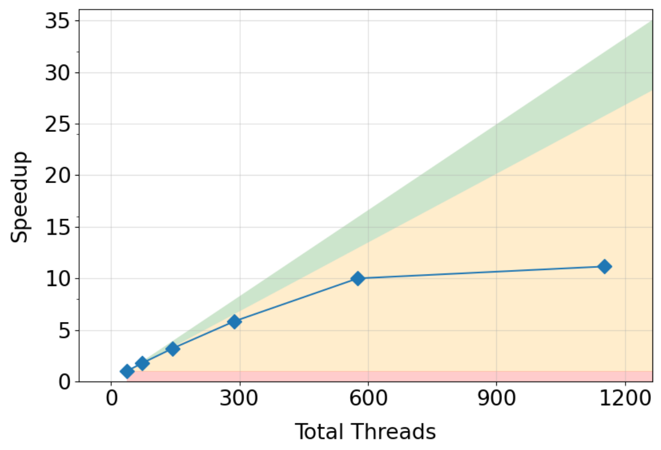
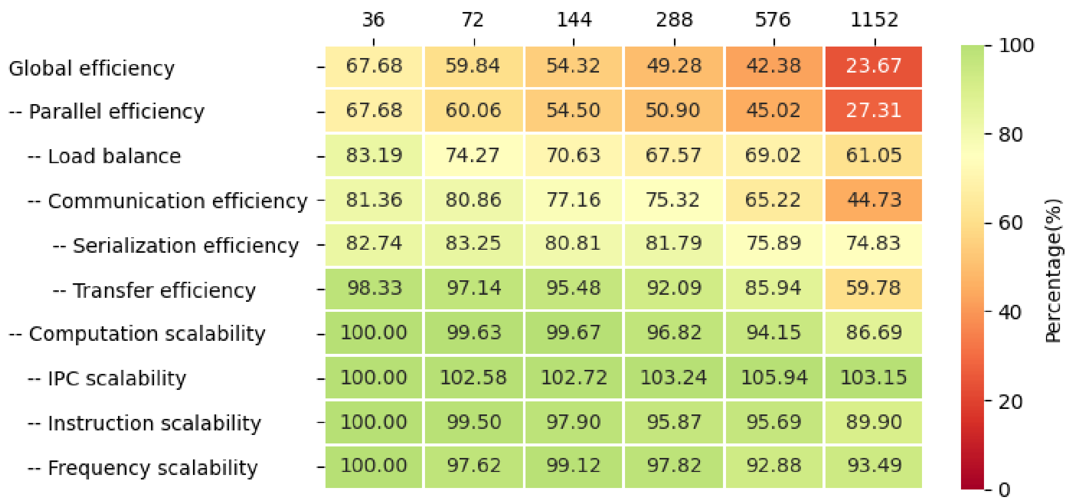


Figure 36: LaBS S2A Iterative phase: Strong scaling.



	36	72	144	288	576	1152
Global efficiency	67.68	59.84	54.32	49.28	42.38	23.67
-- Parallel efficiency	67.68	60.06	54.50	50.90	45.02	27.31
-- Load balance	83.19	74.27	70.63	67.57	69.02	61.05
-- Communication efficiency	81.36	80.86	77.16	75.32	65.22	44.73
-- Serialization efficiency	82.74	83.25	80.81	81.79	75.89	74.83
-- Transfer efficiency	98.33	97.14	95.48	92.09	85.94	59.78
-- Computation scalability	100.00	99.63	99.67	96.82	94.15	86.69
-- IPC scalability	100.00	102.58	102.72	103.24	105.94	103.15
-- Instruction scalability	100.00	99.50	97.90	95.87	95.69	89.90
-- Frequency scalability	100.00	97.62	99.12	97.82	92.88	93.49

Figure 37: LaBS S2A Iterative phase: Efficiency metrics.

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

4.3 Conclusion

In this section, we have presented a basic analysis of the 8 selected test cases. Despite we have shown the performance characteristics of both the Initialization and Iterative phase of the test cases, the priority in optimizations implementation is usually the Iterative phase as it is the most demanding stage in the real-world simulation runs. Therefore, a deep analysis that is necessary to identify the real causes of inefficiencies should be targeted mainly to the iterative solvers.

The main performance issues identified by the performance model were often bad transfer efficiency, load imbalance, and bad serialization efficiency.

In general, a degradation of transfer efficiency in the largest runs with hundreds of processes, meaning an increased ratio of time spent in sending MPI messages, might be an indication of low granularity of computations. This could potentially hide other performance issues that would probably disrupt scalability for larger runs with even more processes.

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

5 Power consumption analysis and static optimisation

For this deliverable we have executed static application tuning of CPU core (CF) and uncore (UCF) frequencies, that influence power consumption of the CPU, to identify available energy savings with and without performance degradation constraint. The static tuning in this context means that one specific configuration of the CPU frequencies it set at the beginning of the application run, and remains unchanged to the end of the execution. Such static configuration is not optimal for each part of the application, which means that the static configuration cannot reach maximum available savings. The dynamic tuning, switching to the configuration that fits the executed region of the application, of the WaLBerla and LaBS will be performed in a future deliverable.

To present the applications' behavior in details, we executed exhaustive state-space-search, not using an algorithm directly converging from the initial to the optimal configuration. For the Turbulent Channel test case, we have the whole state-space using 0.2 GHz step for both CPU core and uncore frequency. Based on the Turbulent Channel results for the Lagoon and S2A test cases we have reduced the state-space, omitting the configurations of the lowest core and lowest uncore frequencies, that cause performance penalty around 30 % and more, which we consider to be non-acceptable.

We are using the HDEEM as well as RAPL energy consumption measurements for energy-efficiency analysis. The HDEEM measurement provides higher precision, and most importantly it gives the power consumption of the whole compute node (blade), while Intel RAPL monitors the power consumption of the CPU itself only. If the energy consumption analysis accepts (limited) performance degradation, ignoring the power consumption of the non-monitored components leads to optimal configuration identification, which may result in longer runtime and overall energy consumption higher, in compare to case when all components are monitored for their power consumption. From this point of view, the HDEEM values are more representative than values measured by RAPL, however to make the results comparable to other state-of the-art researches, we also present the RAPL (energy consumption of Package + DRAM power domains) measurements.

The power monitoring systems evaluates energy consumption based on power samples the power monitoring system took in time, working at a specific sampling frequency. Both HDEEM blade power domain and RAPL work at 1 kHz frequency, while HDEEM not only provides the energy consumption, but moreover the raw power samples. Consumed energy (in Joules) is calculated from power samples (in Watts), that are measured at a sampling frequency (in Hertz) as shown in equation

$$Energy(t) = \int_0^t Power(x)\,\mathrm{d}x \approx \frac{\sum_{i=0}^{n} PowerSample_i}{SamplingFrequency} \tag{8}$$

The following subsections also present power consumption timeline, based on HDEEM power samples taken during the test case execution. Since WaLBerla as well as LaBS have been analysed statically, in this deliverable we do not correlate these timelines to specific regions in the application. The power consumption timelines should give us rough idea about the application dynamic behavior.

The CPU parameters tuning as well as energy consumption measurement was performed by MERIC tools [4], without the instrumentation of the WaLBerla, and LaBS respectively.

5.1 WaLBerla

Lagoon and S2A test cases were executed using 288 MPI processes (8 fully occupied nodes). The Turbulent Channel, which has specific restrictions to number of processes, that can be used to execute, used 180 processes (5 fully occupied nodes). The number of processes were selected with respect to meaningful execution time of each test case, since each test case were executed several times in the limited or full range of available CPU core and uncore frequencies. When choosing these amounts of processes, we also took into account the scalability of these test cases, and avoided the configurations at the scalability end.

5.1.1 Turbulent Channel

For the Turbulent Channel test case we have analysed application run when using 180 processes. When not considering performance penalty caused by the reducing the CPU frequencies, up to 19 % overall HDEEM energy savings (CF 2.3 GHz, UCF 1.8 GHz) causing performance degradation of 8 % (in this configuration 26.3 % RAPL savings). However, configuration of 2.3 GHz CF, 2 GHz UCF reduces HDEEM consumption about 18 %, while extends the runtime about 4.5 % only, and reaches 23.8 % RAPL savings. In the full state-space the RAPL energy savings may reach up to 28 % (CF 1.3 GHz, UCF 1.6 GHz) causing performance degradation 40 %. In this configuration the HDEEM savings are just 14 %.

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

When choosing a configuration with respect to performance the CF 2.3 GHz, UCF 2.2 GHz cause 1.9 % slowdown, while brings 17 % HDEEM energy savings, 21.8 % RAPL savings respectively.

core [GHz] / uncore [GHz]	1.0	1.3	1.7	1.9	2.1	2.3	2.5	2.6	turbo
1.2	−67.34	−60.47	−64.78	−55.66	−51.79	−46.24	−36.04	−36.04	−36.47
1.4	−58.68	−49.47	−40.47	−36.74	−38.02	−35.62	−29.2	−29.2	−20.14
1.6	−52.52	−40.48	−31.66	−32.07	−27.4	−28.22	−10.6	−10.6	−10.54
1.8	−44.02	−34.48	−29.5	−20.99	−23.6	−7.98	−10.84	−10.84	−4.81
2	−45.07	−37.16	−22.91	−20.36	−16.09	−4.53	−3.42	−3.42	−2.72
2.2	−45.48	−34.04	−22.94	−22.02	−17.05	−1.86	−1.2	−1.2	−0.63
2.4	−45.18	−31.06	−21.96	−20.04	−20.47	0.58	−1.15	−1.15	−1.73

Table 23: WaLBerla Turbulent Channel: Impact of the static tuning to overall runtime [%].

core [GHz] / uncore [GHz]	1.0	1.3	1.7	1.9	2.1	2.3	2.5	2.6	turbo
1.2	6.63	7.16	0.63	2.32	1.91	1.01	1.16	0.21	−14.09
1.4	9.13	11.34	11.3	11.03	8.72	6.65	10.61	4.56	−7.85
1.6	10.3	14.08	14.23	12.43	12.98	12.05	13.2	13.44	−3.5
1.8	12.36	14.94	13.88	16.9	13.12	18.97	15.21	12.5	−0.85
2	7.98	10.56	14.33	13.89	14.26	18.02	13.68	14.46	1.7
2.2	3.34	7.99	10.28	9.72	11.02	17.12	14.29	12.78	3.45
2.4	−1.36	5.28	6.91	5.2	5.07	14.25	12.17	8.42	5.33

Table 24: WaLBerla Turbulent Channel: HDEEM energy savings in [%].

core [GHz] / uncore [GHz]	1.0	1.3	1.7	1.9	2.1	2.3	2.5	2.6	turbo
1.2	26.1	24.33	18.48	18.48	15.84	13.5	11.07	9.61	−8.95
1.4	27.14	26.99	25.31	25.31	21.15	17.6	18.76	13.24	0.35
1.6	27.04	28	26.51	26.51	23.58	22.64	20.69	19.45	4.44
1.8	27.81	27.51	25.4	25.48	22.64	26.34	21.42	18.27	7.95
2	22.2	23.16	24	24	21.85	23.87	19.12	18.99	5.84
2.2	16	18.66	18.87	16.77	18.05	21.82	17.77	15.97	2.71
2.4	9.57	14.53	14.27	14.27	11.35	17.51	14.35	10.5	−0.4

Table 25: WaLBerla Turbulent Channel: RAPL energy savings in [%].

From the Figure 38 of power consumption timeline we can identify a low-power initialization phase and an iterative solver with power oscillation according compute and memory access intensity. From this point of view this test case does not show much dynamism that can be exploited by dynamic hardware parameters switching. Figure 39 shows the CPUs power consumption during the iterations, and Figure 40 focuses on a single iteration. From this we can see not only that one CPU consumes most of the time more power than the other one, but also that scheme is different. It may indicate that the workload of the CPUs is not the same, which imply that the optimal configuration of each CPU might be different. This behavior has a potential for additional energy savings when performing the dynamic tuning, and should be investigated further during the upcoming phases of the SCALABLE project.

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

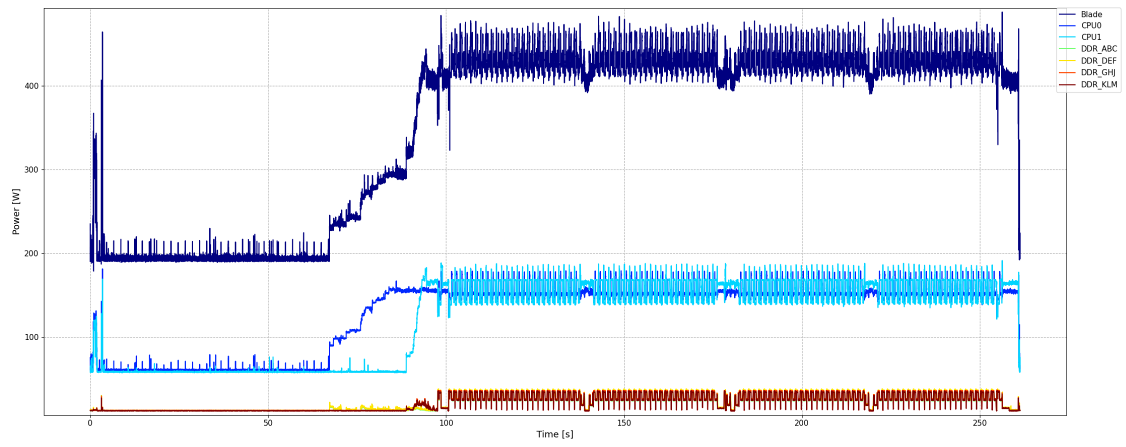


Figure 38: WaLBerla Turbulent Channel: Power consumption of the blade, CPUs and memory channels.

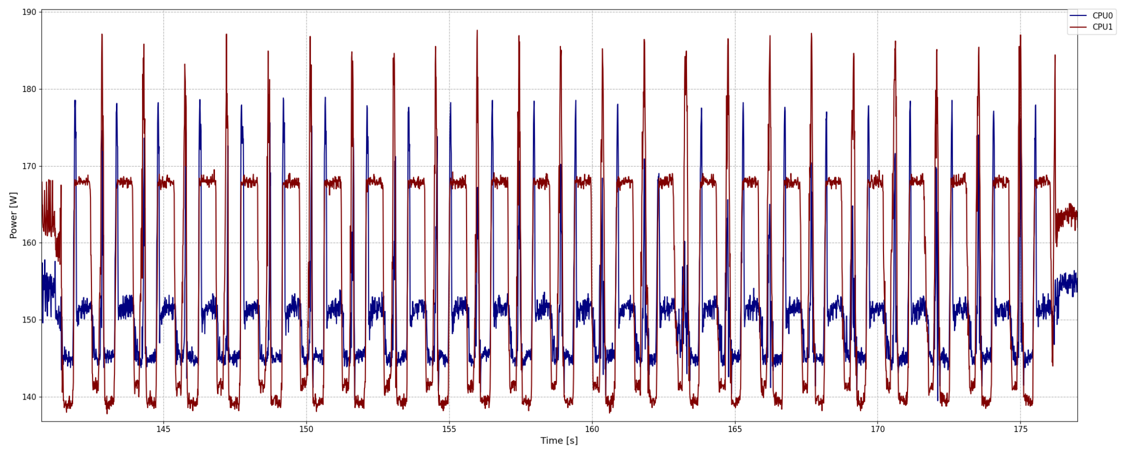


Figure 39: WaLBerla Turbulent Channel: CPUs power consumption during the iterative solver execution.

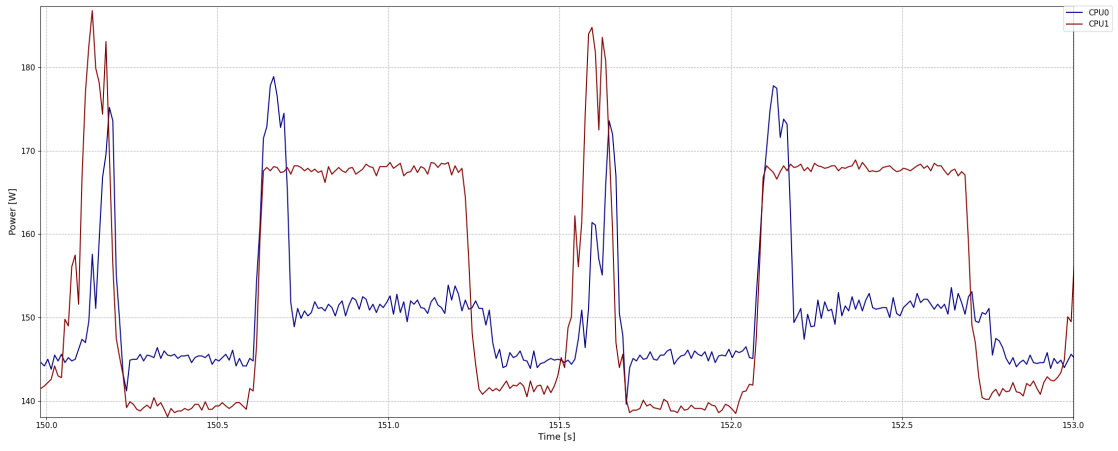


Figure 40: WaLBerla Turbulent Channel: Power consumption of the CPUs, detail at one iteration of the iterative solver.

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

5.1.2 Lagoon

In the space of evaluated configurations 13.6 % HDEEM savings can be reached (CF 2.1 GHz, UCF 1.6 GHz), in the same configuration RAPL savings is 22.5 %, however the performance drops about over 30 %. The static tuning unfortunately does not provide a possibility to get major energy savings without a performance penalty, only 2.3 % HDEEM, 3 % RAPL savings interchanged with 2 % performance degradation.

From the power consumption timeline (Figure 41) we can see much more dynamism during the initialization phase in compare to Turbulent Channel test case. In this test case execution the initialization phase is already longer than the solver execution time.

core [GHz] / uncore [GHz]	1.9	2.1	2.3	2.5	2.6	turbo
1.6	−38.57	−30.37	−24.25	−18.23	−16.17	−4.8
1.8	−35.92	−27.98	−22.16	−15.83	−12.65	−2.04
2	−33.47	−26.18	−19.97	−14.26	−11.33	−1.21
2.2	−32.79	−24.04	−17.78	−12.34	−9.53	−0.2
2.4	−30.96	−22.88	−16.5	−10.87	−8.16	−0.12

Table 26: WaLBerla Lagoon: Impact of the static tuning to overall runtime [%].

core [GHz] / uncore [GHz]	1.9	2.1	2.3	2.5	2.6	turbo
1.6	12.65	13.66	13.52	11.93	10.5	1.05
1.8	12.43	13.54	13.12	12.13	11.37	2.33
2	10.98	11.63	11.85	10.56	9.63	1.35
2.2	7.79	9.93	10.21	8.86	8.04	0.72
2.4	5.36	7.06	7.37	6.57	5.7	−0.52

Table 27: WaLBerla Lagoon: HDEEM energy savings in [%].

core [GHz] / uncore [GHz]	1.9	2.1	2.3	2.5	2.6	turbo
1.6	21.98	22.52	20.04	17.46	14.87	2.24
1.8	21.13	21.66	19	16.53	15.03	3.08
2	18.75	19.12	16.7	14.78	13.26	1.97
2.2	15.4	15.48	14.66	12.9	11.28	1
2.4	11.74	11.83	11.5	9.48	8.65	−0.31

Table 28: WaLBerla Lagoon: RAPL energy savings in [%].

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

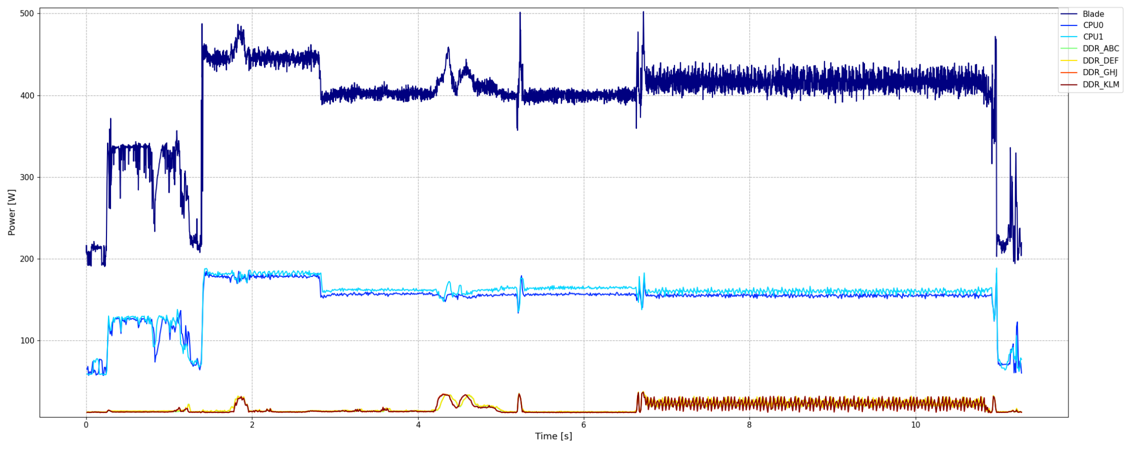


Figure 41: WaLBerla Lagoon: Power consumption of the blade, CPUs and memory channels.

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

5.1.3 S2A

The S2A test case shows behavior similar to Lagoon test case. Also in this case it is not possible to reach energy savings without performance degradation when doing the static tuning. The Lagoon performance aware optimal configuration (turbo CF and 1.8 GHz UCF) results in 1 % performance degradation, with negligible HDEEM savings and 0.4 % RAPL savings. 10 % performance degradation limit gives a space to reach 10.8 % HDEEM savings in 2.6 GHz CF, 1.8 GHz UCF configuration, resulting in 15 % RAPL savings.

When focusing on energy savings only 17 % HDEEM savings is reachable (2.1 GHz CF, 1.6 GHz UCF) which extend the runtime about 24 %. In this configuration RAPL savings are 26 %. Another 1.2 % RAPL savings are reachable in the evaluated state-space, however with major performance degradation of 31 %.

core [GHz] / uncore [GHz]	1.9	2.1	2.3	2.5	2.6	turbo
1.6	−31.13	−23.95	−18.54	−13.41	−11.62	−2.07
1.8	−29.68	−23.48	−17.13	−12.41	−10.38	−1.14
2	−28.65	−22.28	−16.64	−11.18	−9.72	−0.93
2.2	−28.01	−21.14	−15.44	−10.75	−9.33	−0.71
2.4	−27.76	−20.63	−15.7	−10.64	−8.7	0.26

Table 29: WaLBerla S2A: Impact of the static tuning to overall runtime [%].

core [GHz] / uncore [GHz]	1.9	2.1	2.3	2.5	2.6	turbo
1.6	17.15	17.01	15.57	13.58	11.39	−0.06
1.8	16.23	15.79	14.85	12.85	10.84	0.09
2	14.07	14.04	12.71	11.1	8.63	−0.21
2.2	11.31	11.59	10.62	8.59	6.27	−0.49
2.4	7.75	8.58	6.89	5.4	3.57	0.08

Table 30: WaLBerla S2A: HDEEM energy savings in [%].

core [GHz] / uncore [GHz]	1.9	2.1	2.3	2.5	2.6	turbo
1.6	27.29	26.04	22.44	18.87	16.11	0.16
1.8	25.85	24.2	21.23	17.75	14.99	0.43
2	22.98	21.6	18.46	15.44	12.6	−0.06
2.2	19.55	18.32	15.58	12.26	9.36	−0.29
2.4	15.12	14.11	11.45	8.36	5.95	0.23

Table 31: WaLBerla S2A: RAPL energy savings in [%].

The power consumption timeline, depicted in Figure 42, shows the initialization phase, that is similar to power consumption of the Lagoon test case. The major difference is visible during the following stage, when the power consumption of the CPUs persists at the same level as during the previous stage (approximately 160 W), while the Lagoon solver part shows about 10 W drop. Another difference in these two power timelines is power consumption fluctuations during the solver iterations. In this test case the the CPUs power consumption is stable, but the memories power consumption has a saw shape, which is opposite behavior to Lagoon solver power consumption.

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

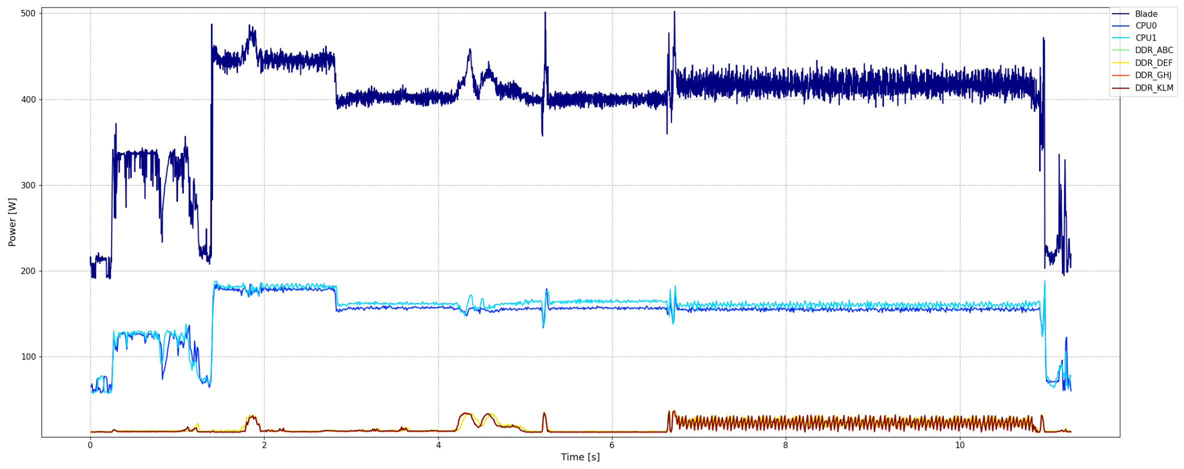


Figure 42: WaLBerla S2A: Power consumption of the blade, CPUs and memory channels.

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

5.2 LaBS

The LaBS energy-efficiency analysis were execute using 288 processes (8 fully occupied nodes). Also in this case the test cases' scalability and execution time were considered when choosing this configuration for the analysis.

5.2.1 Turbulent Channel

As well as in case of the WaLBerla also for the LaBS Turbulent Channel test case we have evaluated CF and UCF configurations within the full state-space and based on these measurements we have reduced the state space to configurations with reasonable performance degradation.

In this case the static tuning brings several configurations in which performance degradation is similar to HDEEM savings, so it is possible to reach 11.2 % HDEEM savings in a configuration (2.6 GHz CF, 2.2 GHz UCF) causing runtime extension 10.5 % (15.5 % RAPL savings), or 24 % HDEEM savings with 24.9 % longer runtime in 1.9 GHz CF, 1.8 GHz UCF (34 % RAPL savings), but unfortunately there is no such configuration with a trade-off bellow 10 %.

core [GHz] / uncore [GHz]	1.0	1.3	1.7	1.9	2.1	2.3	2.5	2.6	turbo
1.2	−82.69	−59.5	−46.03	−39.69	−34.48	−32.33	−28.33	−27.28	−21.19
1.4	−73.49	−53.02	−37.11	−34.18	−25.99	−25.18	−21.39	−19.59	−13.57
1.6	−70.49	−48.63	−33.39	−27.91	−23.11	−19.67	−16.58	−15.44	−7.93
1.8	−69.51	−48.13	−30.62	−24.88	−23.66	−17.18	−16.65	−13.3	−2.03
2	−67.67	−46.6	−29.31	−23.21	−20.09	−16.16	−12.88	−12.12	0.02
2.2	−67.15	−45.07	−28.1	−23.92	−18.21	−15.11	−11.68	−10.54	0.18
2.4	−67.59	−44.25	−27.08	−23.85	−17.39	−13.83	−10.83	−10.97	0.54

Table 32: LaBS Turbulent Channel: Impact of the static tuning to overall runtime [%].

core [GHz] / uncore [GHz]	1.0	1.3	1.7	1.9	2.1	2.3	2.5	2.6	turbo
1.2	17.31	22.74	21.01	20.7	19.79	16.55	13.35	10.59	−10.9
1.4	19.54	24.21	23.98	22.23	22.98	19.11	15.94	13.78	−7.65
1.6	19.02	24.62	24.39	23.86	23.04	20.8	17.52	14.88	−3.3
1.8	17.46	22.96	24.24	24.08	21.01	20.87	15.86	14.69	0.81
2	14.94	20.65	22.13	22.45	20.52	18.86	15.82	12.93	1.56
2.2	11.15	18	19.87	18.85	18.85	16.74	13.9	11.17	0.86
2.4	6.49	14.44	16.8	15.53	15.91	14.15	11.27	7.62	0.55

Table 33: LaBS Turbulent Channel: HDEEM energy savings in [%].

core [GHz] / uncore [GHz]	1.0	1.3	1.7	1.9	2.1	2.3	2.5	2.6	turbo
1.2	39.33	39.77	36.03	34.01	30.95	26.59	22.09	17.35	−8.93
1.4	39.72	39.86	37.46	34.66	32.85	28.33	23.75	20.19	−3.62
1.6	38.03	39.49	37.12	35.03	32.07	29.44	24.25	20.92	−2.49
1.8	35.54	37.3	36.57	34.09	30.01	29	22.27	20.01	1.38
2	32.24	34.11	32.93	31.26	28.78	26.04	21.14	17.77	1.8
2.2	28.17	30.31	29.49	27.26	26.11	22.55	18.54	15.51	0.94
2.4	22.15	26.77	25.64	23.46	22.36	19.23	15.4	11.13	0.53

Table 34: LaBS Turbulent Channel: RAPL energy savings in [%].

Figure 43 presents the LaBS Turbulent Channel power consumption timeline, which shows dynamic behavior during the initialisation phase, but stable power consumption of the final execution part, where memories seems to be without active usage.

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

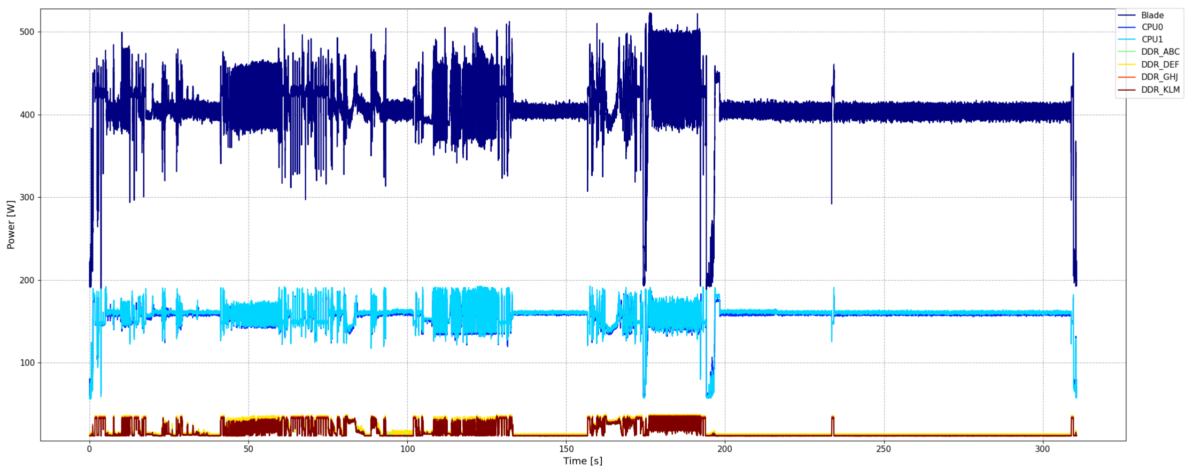


Figure 43: LaBS Turbulent Channel: Power consumption of the blade, CPUs and memory channels.

Figure 44 and Figure 45 shows power consumption of a single CPU and a single memory respectively, in a detail with focus on one of parts of the execution, that shows high and frequent variety in power consumption of both power domains. Such part of the application has high requirements to CPU as well as to memory. The change in the hardware utilization is too fast, to exploit such application dynamism. Most probably it is not possible to get energy savings by tuning the CF and UCF during this application region, however still some energy savings should be reachable.

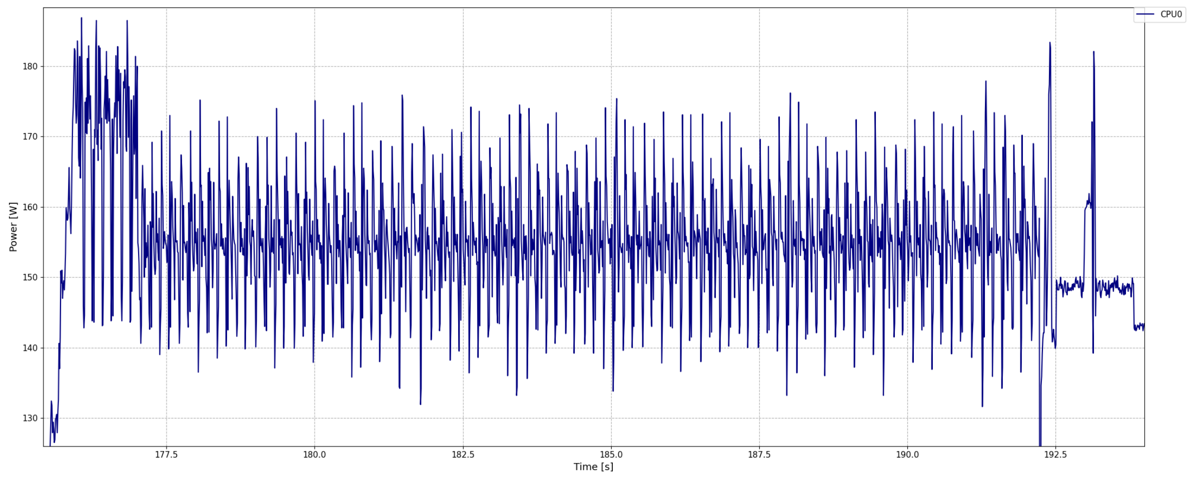


Figure 44: LaBS Turbulent Channel: Power consumption of a single CPU – detail at a specific part of the execution.

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

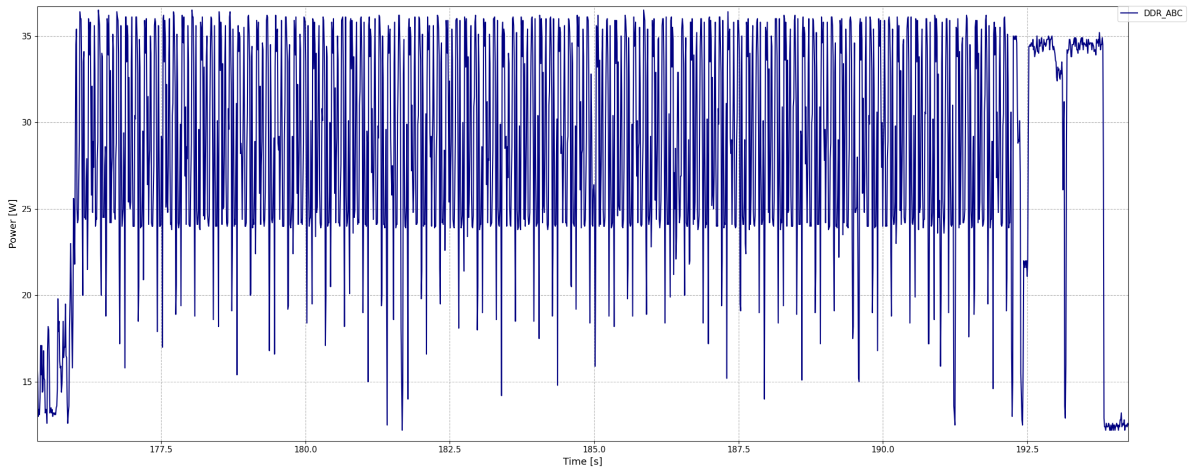


Figure 45: LaBS Turbulent Channel: Power consumption of a memory channel – detail at a specific part of the execution.

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

5.2.2 Lagoon

In case of the Lagoon, we are able to get over 10.7 % HDEEM savings just by reducing the UCF to 1.8 GHz, which extends the runtime about 0.8 %. In this configuration the RAPL savings are 5.2 %. The 2.6 GHz CF and 2.0 GHz UCF leads to runtime longer about 10 %, however energy consumption drops about 18 % HDEEM and 22 % RAPL. In the evaluated state-space the energy-consumption-optimal configuration is the 2.3 GHz CF, 1.6 GHz UCF that brings 21.2 % HDEEM and 29.4 % RAPL savings. This configuration causes 21.2 % performance degradation.

core [GHz] / uncore [GHz]	2.1	2.3	2.5	2.6	turbo
1.6	−27.39	−21.27	−21.27	−13.4	−2.55
1.8	−25.8	−18.8	−18.8	−11.21	−0.84
2	−26.19	−17.21	−17.21	−9.93	−0.18
2.2	−22.26	−15.72	−15.72	−8.62	−0.09
2.4	−21.17	−15.32	−15.32	−8	−0.08

Table 35: LaBS Lagoon: Impact of the static tuning to overall runtime [%].

core [GHz] / uncore [GHz]	2.1	2.3	2.5	2.6	turbo
1.6	21.66	21.26	19.61	19.61	10.71
1.8	21.14	21.17	19.58	19.58	10.75
2	18.05	19.62	18.16	18.16	8.56
2.2	17.22	17.52	16.08	16.08	5.85
2.4	14.19	14.25	13.16	13.16	3.29

Table 36: LaBS Lagoon: HDEEM energy savings in [%].

core [GHz] / uncore [GHz]	2.1	2.3	2.5	2.6	turbo
1.6	30.75	29.38	29.38	25.02	5.72
1.8	29.74	28.87	28.87	24.46	5.21
2	26.15	26.95	26.95	22.19	3.27
2.2	24.09	23.8	23.8	19.42	1.26
2.4	20.79	19.58	19.58	15.89	−0.16

Table 37: LaBS Lagoon: RAPL energy savings in [%].

Figure 46 of the Lagoon power consumption timeline looks different to Turbulent Channel one's, however main traits remains the same – since the final part of the execution is shorter in this case, significant part of the execution time has high variability in power consumption, another words variability in requirements to underlying hardware. Based on the reached static savings, we may assume that this region is possible to tune without a performance penalty.

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

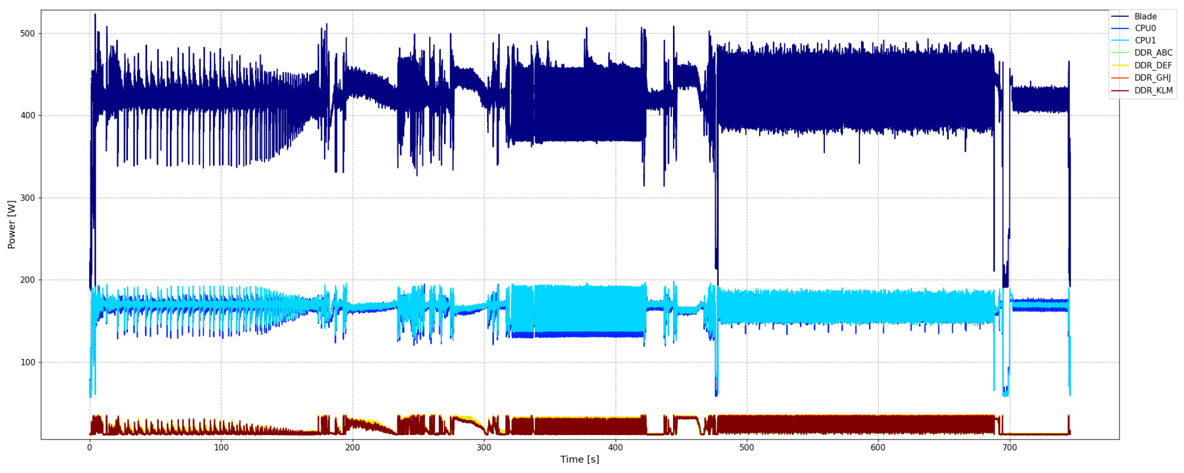


Figure 46: LaBS Lagoon: Power consumption of the blade, CPUs and memory channels.

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

5.2.3 S2A

In case of the LaBS S2A execution it is possible to save 5.5 % HDEEM energy consumption (6.5 % RAPL) if the uncore frequency is fixed to 1.8 GHz, which results in a run longer about 0.7 %. When CF turbo frequency is disabled, 15–22.3 % HDEEM savings (18.5–28.1 % RAPL) are reachable with trade-off 5.5–9.2 % runtime degradation, according the specified UCF from range 2.4–1.8 GHz. Maximum available energy savings in the evaluated state-space is in the configuration of the lowest frequencies (2.1 GHz CF, 1.6 GHz UCF). This configuration cause 18.6 % performance degradation, however brings 27.9 % HDEEM, 37.1 % RAPL savings respectively.

core [GHz] / uncore [GHz]	2.1	2.3	2.5	2.6	turbo
1.6	−18.57	−13.83	−11.24	−9.17	−1.83
1.8	−16.76	−12.18	−8.72	−7.26	−0.72
2	−15.36	−10.97	−8.06	−6.42	−0.24
2.2	−14.84	−9.78	−7.13	−5.76	0.08
2.4	−14.16	−9.22	−6.6	−5.46	−0.12

Table 38: LaBS S2A: Impact of the static tuning to overall runtime [%].

core [GHz] / uncore [GHz]	2.1	2.3	2.5	2.6	turbo
1.6	27.9	27.05	24.08	22.74	6.41
1.8	27.35	26.44	24.06	22.37	5.48
2	25.6	24.71	21.98	20.41	3.47
2.2	22.92	22.57	19.68	17.99	1.62
2.4	19.83	19.5	16.66	14.92	−0.14

Table 39: LaBS S2A: HDEEM energy savings in [%].

core [GHz] / uncore [GHz]	2.1	2.3	2.5	2.6	turbo
1.6	37.18	34.76	30.92	29.13	7.8
1.8	35.96	33.72	30.58	28.18	6.49
2	33.34	31.48	27.86	25.47	4.11
2.2	30.26	28.8	24.6	22.29	1.86
2.4	26.23	24.89	20.86	18.5	−0.16

Table 40: LaBS S2A: RAPL energy savings in [%].

The power consumption timeline is very similar to the progression of the Lagoon test case, with difference in length of the regions of similar behavior. Notable difference is in the power consumption of the first part of the execution initialization, which is not so variable in this case.

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

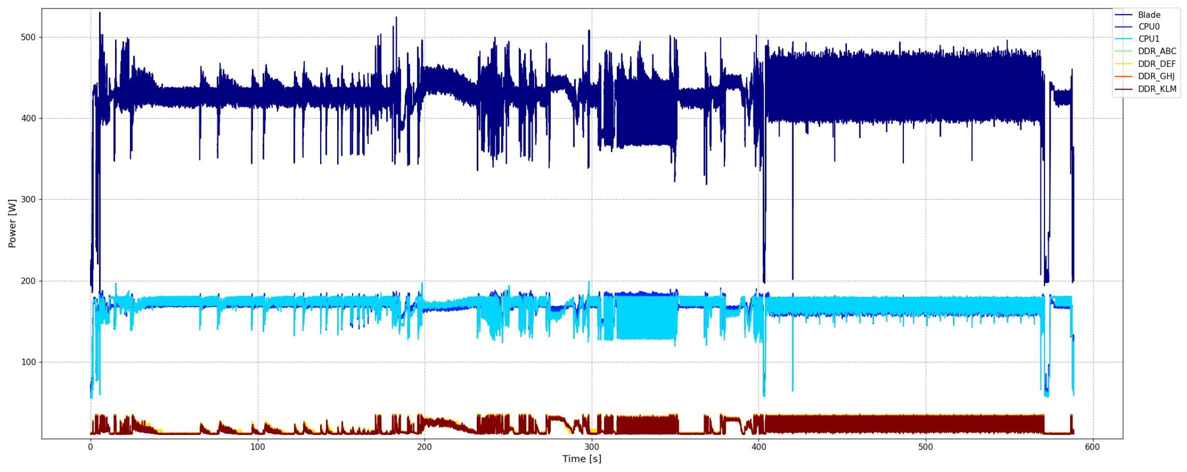


Figure 47: LaBS S2A: Power consumption of the blade, CPUs and memory channels.

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

5.3 Conclusion

The presented results show us that it is possible to reach major energy savings by using static CPU frequencies tuning, at least 13.6 % HDEEM savings for WaLBerla, and 21.2 % for LaBS respectively. In most of these cases the major energy savings are available only if a performance degradation is accepted, as show Table 41 and Table 42.

The applications have dynamic behavior, as visible from the power consumption timelines, so it is hard to identify a single configuration, that should fit the whole runtime. Moreover, the timelines of the test cases, are similar, but not the same, most probably also by executing various functions of the applications. Due to that it is hard to identify single configuration, that would fit all the test cases. We have manually selected configuration 2.6 GHz CF, 1.8 GHz UCF for WaLBerla, and 2.6 GHz CF, 2.2 GHz UCF for LaBS, which are the configurations that should bring energy savings and less or equal performance degradation in each test case.

It is important to say, that if the number of iterations is increased, the ratio between solver and initialization phase will change, so the optimal static CPU frequencies configuration identified by our analysis, may not correspond to optimal configuration of such execution. This problem is handled by dynamic tuning, that identifies optimal configuration of each region of the application, and may easily prevent the performance degradation. Impact of the dynamic tuning will be evaluated in a future SCALABLE deliverable.

		-2 % limit	**-5 % limit**	**-10 % limit**	**HDEEM optimal**	**RAPL optimal**	**WaLBerla optimal**
TBC	**time [%]**	-1.9	-4.5	-8	-8	-40	-10.8
	HDEEM [%]	17.1	18	19	19	14	12.5
	RAPL [%]	21.8	23.8	26.3	26.3	28	18.3
	CF; UCF [GHz]	2.3; 2.2	2.3; 2.0	2.3; 1.8	2.3; 1.8	1.3; 1.6	2.6; 1.8
Lagoon	**time [%]**	-2	-	-9.5	-30.3	-30.3	-12.6
	HDEEM [%]	2.3	-	8	13.6	13.6	11.4
	RAPL [%]	3	-	11.2	22.5	22.5	15
	CF; UCF [GHz]	-; 1.8	-	2.6; 2.2	2.1; 1.6	2.1; 1.6	2.6; 1.8
S2A	**time [%]**	-	-	-10.3	-24	-31.1	-10.3
	HDEEM [%]	-	-	10.8	17	17.1	10.8
	RAPL [%]	-	-	15	26	27.2	15
	CF; UCF [GHz]	-	-	2.6; 1.8	2.1; 1.6	1.9; 1.6	2.6; 1.8

Table 41: WaLBerla energy savings for various performance degradation trade-offs, and without any runtime limit.

		-2 % limit	**-5 % limit**	**-10 % limit**	**HDEEM optimal**	**RAPL optimal**	**LaBS optimal**
TBC	**time [%]**	-	-	-10.5	-24.9	-53	-10.5
	HDEEM [%]	-	-	11.2	24	24.2	11.2
	RAPL [%]	-	-	15.5	34	39.8	15.5
	CF; UCF [GHz]	-	-	2.6; 2.2	1.9, 1.8	1.3; 1.4	2.6; 2.2
Lagoon	**time [%]**	-0.8	-	-10	-21.2	-21.2	-8.6
	HDEEM [%]	10.7	-	18.1	21.2	21.2	16.1
	RAPL [%]	5.2	-	22.1	29.3	29.3	19.4
	CF; UCF [GHz]	-; 1.8	-	2.6; 2	2.3; 1.6	2.3; 1.6	2.6; 2.2
S2A	**time [%]**	-0.7	-5.7	-8.7	-18.6	-18.6	-5.7
	HDEEM [%]	5.5	18	24	27.9	27.9	18
	RAPL [%]	6.5	22.2	30.5	37.1	37.1	22.2
	CF; UCF [GHz]	-; 1.8	2.6; 2.2	2.5; 1.8	2.1; 1.6	2.1; 1.6	2.6; 2.2

Table 42: LaBS energy savings for various performance degradation trade-offs, and without any runtime limit.

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

References

[1] Corey Gough, Ian Steiner, and Winston Saunders. "CPU Power Management." In: *Energy Efficient Servers: Blueprints for Data Center Optimization*. Berkeley, CA: Apress, 2015, pp. 21–70. ISBN: 978-1-4302-6638-9. DOI: 10.1007/978-1-4302-6638-9_2. URL: https://doi.org/10.1007/978-1-4302-6638-9_2.

[2] Daniel Hackenberg et al. "HDEEM: High Definition Energy Efficiency Monitoring." In: *Energy Efficient Supercomputing Workshop (E2SC)*. Nov. 2014, pp. 1–10. DOI: 10.1109/E2SC.2014.13.

[3] David L. Hill et al. "The uncore: A modular approach to feeding the high-performance cores." eng. In: *Intel Technology Journal* 14.2 (3 2010), pp. 30–49.

[4] Ondrej Vysocky et al. "MERIC and RADAR Generator: Tools for Energy Evaluation and Runtime Tuning of HPC Applications." In: *High Performance Computing in Science and Engineering*. Ed. by Tomas Kozubek et al. Cham: Springer International Publishing, 2018, pp. 144–159. ISBN: 978-3-319-97136-0.

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