6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
11-15 June 2018, Glasgow, UK

MASSIVELY-PARALLEL MULTI-GPU SIMULATIONS FOR FAST AND ACCURATE AUTOMOTIVE AERODYNAMICS

CHRISTOPH A. NIEDERMEIER[1], CHRISTIAN F. JANSSEN[2] AND THOMAS INDINGER[1]

[1] FluiDyna GmbH
Edisonstraße 3, 85716 Unterschleißheim
{christoph.niedermeier,thomas.indinger}@fluidyna.de

[2] Altair Engineering GmbH
Friedensallee 27, 22765 Hamburg, Germany
christian.janssen@altair.com

Key words: LBM, GPU, Automotive Aerodynamics, Multi-GPU

Abstract. In this paper, the scaling properties of the commercial Computational Fluid Dynamics (CFD) solver ultraFluidX are analyzed. The solver is based on the Lattice Boltzmann Method (LBM), is explicit in time, inherently transient and only requires next-neighbor information on the computational grid. ultraFluidX thus highly benefits from the computational power of Graphics Processing Units (GPUs). Additionally, the solver can make use of multiple GPUs on multiple compute nodes through an efficient implementation based on CUDA-aware MPI. The capabilities of the solver have already been demonstrated to various automotive OEMs around the world through several validation projects. This paper focuses on the scalability of ultraFluidX on single- and multi-node configurations in large-scale multi-GPU environments. Weak and strong scaling results for two selected test cases are reported and demonstrate that multi-GPU flow solvers have the potential to deliver high-fidelity results overnight.

1 INTRODUCTION

Aerodynamic properties of vehicles play a crucial role in the automotive design process. The required sophisticated level of knowledge about the flow field around the vehicle under consideration can only be achieved through wind tunnel experiments or highly-resolved numerical simulations. The flow field in such a case is very complex and unsteady, and the corresponding CFD simulations have to be fully transient and highly resolved to capture all relevant physical effects. GPU-based CFD paves the way for a wide use of such high-fidelity simulations for automotive aerodynamics and gives a boost to green computing using more energy-efficient hardware. ultraFluidX achieves unprecedented turnaround times of just a few hours for transient flow simulations of fully detailed production-level passenger and heavy-duty vehicles.

2 ULTRAFLUIDX

The current contribution is based on the commercial GPU-based CFD solver ultraFluidX. ultraFluidX is developed by FluiDyna GmbH and is exclusively distributed by Altair Engineering. Main motivation for the code development is to introduce higher fidelity models, fully transient analyses, small turnaround times and low total cost to automotive technical and business drivers. ultraFluidX is based on the Lattice Boltzmann Method and a recent, highly accurate LBM collision operator based on Cumulants [1, 2, 3]. The LBM collision model is accompanied by an LBM-consistent Smagorinsky LES model [4] an a wall modeling technique based on the work of [5]. The Cartesian LBM base mesh is combined with an octree-based grid refinement. Based on a relatively coarse Cartesian background grid, the mesh is consecutively refined towards the obstacle surface. Between each refinement level, the mesh spacing is reduced by a factor of 2. The first refinement levels typically are defined via axis-aligned bounding boxes, as can be seen in Figure 1 (left). Higher refinement levels closer to the geometry are either based on custom user-defined volumes or are defined via geometry offsets. Kinks and thin parts are fully resolved in the volumetric preprocessing. Moreover, the appropriate physical modeling is selected automatically inside tight gaps and cavities. Representative production-level volume meshes are generated within the order of $\mathcal{O}(1h)$.

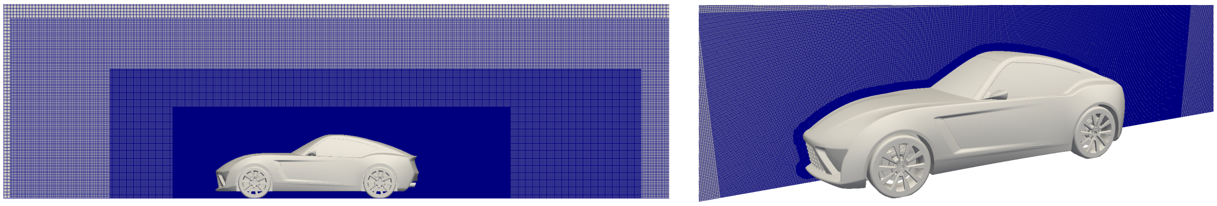


Figure 1: Details of the ultraFluidX grids.

In addition, the code contains several enhanced features for automotive applications. These features typically come with additional computational overhead, which has to be considered in the performance evaluation and in comparison to other available software packages. These features include, but are not limited to, porous media zones, support for moving floors (single- and five-belt systems), static floors with boundary layer suction, rotating wheel models (straightforward wall-velocity boundary conditions or more advanced MRF and overset grid approaches) and automatic convergence detection. The code also features advanced post-processing capabilities, such as window averaging, spatial drag/lift contributions, probe output and section cut output.

3 PERFORMANCE ANALYSIS

In this section, several weak and strong scaling results from single- and multi-node configurations are presented. First, a rather academic test case is analyzed. Then, a realistic production case including extensive grid refinement and additional automotive features that were discussed in the previous section is investigated. In particular, the case contains porous media, rotating wheels modeled with a wall-velocity boundary condition and a belt system. The cases were run on three different test beds, two local machines with NVIDIA Tesla P100/V100 GPUs and a massively parallel supercomputer with NVIDIA Tesla P100s distributed among multiple nodes, as detailed in Table 1. The local P100 machine from Supermicro is equipped with ten P100 PCIe GPUs. The machine features a single PCIe root complex and, hence, allows for very efficient peer-to-peer communication between the GPUs. The second, massively parallel test bed consists of four nodes of the TSUBAME 3.0 supercomputer. Each of these nodes features four P100 GPUs with SXM2 and first-generation NVLink, while the nodes are connected to each other via Intel Omni-Path. Moreover, the NVIDIA DGX-1 machine based on the Volta GPU architecture is tested. This machine features eight V100 GPUs and an even faster second-generation NVLink GPU interconnect.

Table 1: Details of the three test beds.

	P100 PCIe (local)	**P100 SXM2 (non-local)**	**V100 (local)**
Model/Make	SYS-4028GR-TR2	TSUBAME 3.0	DGX-1
GPU Type	Tesla P100	Tesla P100	Tesla V100
GPU Architecture	NVIDIA Pascal	NVIDIA Pascal	NVIDIA Volta
SP Performance	9.3 TFLOPS	10.6 TFLOPS	15.7 TFLOPS
GPU Memory	16 GB HBM2	16 GB HBM2	16 GB HBM2
Memory Bandwidth	732 GB/s	732 GB/s	900 GB/s
Interconnect	PCIe 3.0 x16	NVLink 1.0	NVLink 2.0
GPUs per Node	10	4	8
Node Count	1	4	1

3.1 Empty wind tunnel (uniform grid, no refinement)

At first, an empty wind tunnel case is used to evaluate the single-GPU performance of ultraFluidX on the different hardware configurations, before turning to multi-GPU simulations. The rectangular domain has an aspect ratio of 6:3:2 with a constant-velocity inlet, a non-reflecting outlet and slip walls in between. The case was run on a single GPU, using eight successively refined, uniform grids with a total of 0.3 to 36 million voxels (refined in steps of two). The resulting single-GPU performance averaged over 10,000 iterations is plotted in Figure 2. Significant differences between the Pascal and the Volta GPU architectures can be observed. The maximum performance of the V100 is approximately 1.9 times higher than the maximum performance of the P100 (PCIe). Among the GPUs with Pascal architecture, the slightly higher theoretical peak performance of the SXM2 models can also clearly be observed. Moreover, and more importantly, the single-GPU performance highly depends on the GPU utilization. For less than one million voxels per GPU, the single-GPU performance drops below 80 % of the maximum performance. A similar trend can be seen for increasing voxel counts per GPU: again, the performance is reduced. For, e.g., 36 M voxels per GPU, a performance of 85 % to 90 % of the maximum performance is found.

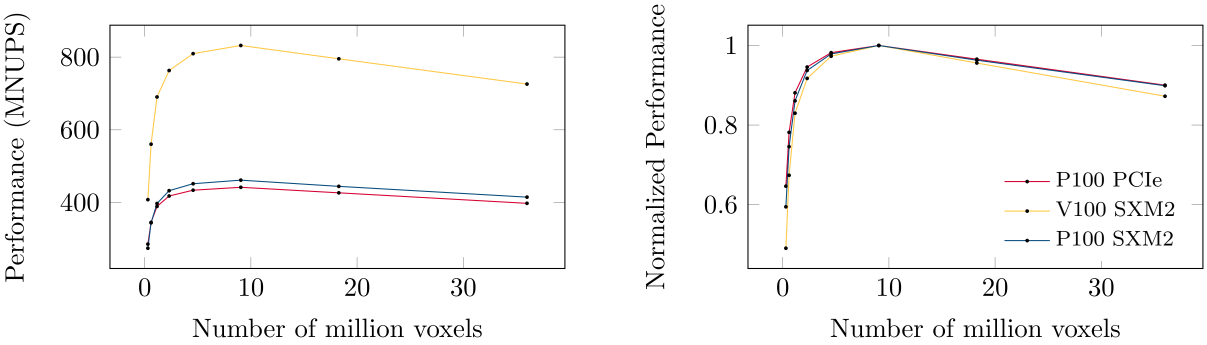
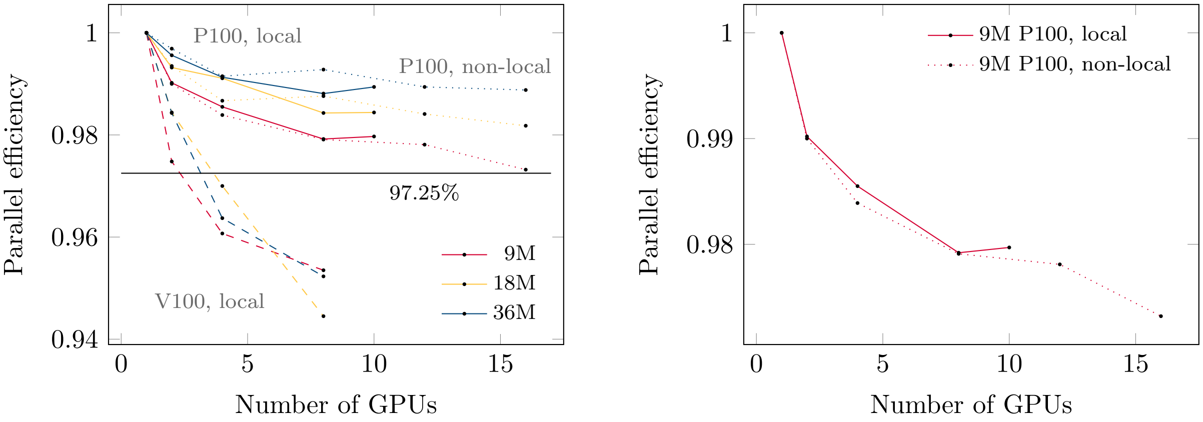


Figure 2: Single-GPU performance for different test case sizes.

Then, weak scaling tests were conducted on all three hardware configurations and using all three grids (9 M, 18 M, 36 M). In order to do so, the initial rectangular domain was extended downstream depending on the number of GPUs to keep the aspect ratio of 6:3:2 constant per GPU device. Excellent parallel efficiency is found for these tests (see Figure 3a), above 98 % for one to eight P100 GPUs in local and non-local setups and above 95 % for up to eight V100s in the DGX-1. Moreover, no significant differences between the fully local P100 setups and the distributed multi-node setup is found (Figure 3b). Even for the simulations with 16 P100s in a total of four different compute nodes, the parallel efficiency is above 97 %.



(a) Overview over all three test beds.

(b) Close-up: local vs. non-local P100s.

Figure 3: Weak scaling results for the empty channel case: parallel efficiency.

After these initial weak scaling runs, a set of strong scaling tests is run. Such tests are much more relevant for engineering applications, as they directly reflect the impact of the parallelization on the time-to-solution of the problem under consideration. By adding more and more GPUs, the individual GPU workload is successively reduced in order to evaluate if and how the overall simulation times can be reduced by using more computational resources. The resulting parallel efficiency for the empty wind tunnel case is depicted in Figure 4. The parallel efficiency for the 9 M case drops to 45 % for eight GPUs. On the other hand, the parallel efficiency for the 36 M case is above 100 % for the simulations on two and four GPUs. This is caused by the fact that the individual GPU performance changes as well in strong scaling tests, as it scales with the individual GPU work load, independent on the additional communication and parallelization overheads.

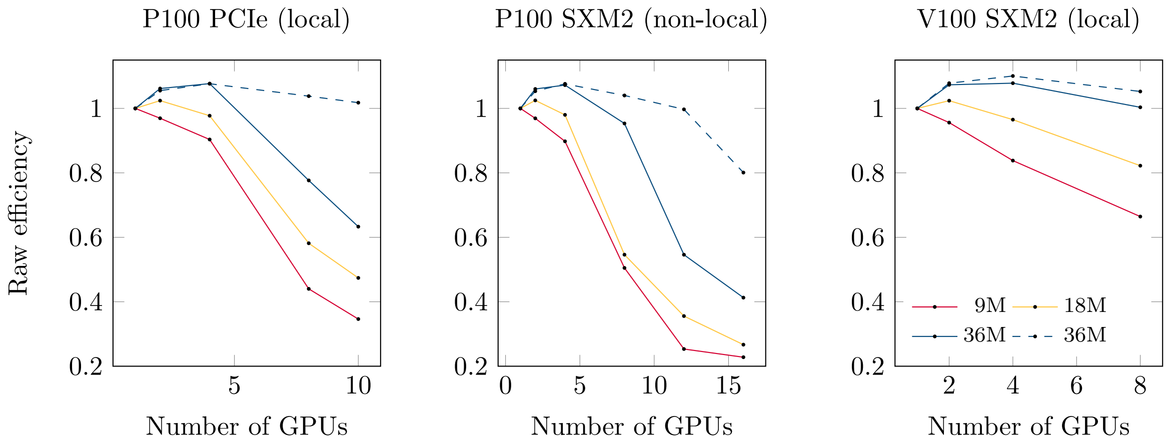


Figure 4: Strong scaling results for the empty channel for three selected grid sizes: parallel efficiency in relation to the single-GPU run (raw efficiency).

In order to asses the direct impact of the parallelization and communication overhead, the GPU performance has to be examined in relation to the raw GPU performance for the corresponding GPU utilization (voxels/GPU device). This *effective efficiency* is depicted in Figure 5. No overshoots above 100 % can be found, the performance degradation for high GPU numbers is also less pronounced, and, most importantly, the scaling for one to four GPUs shows a parallel efficiency of above 95 %.

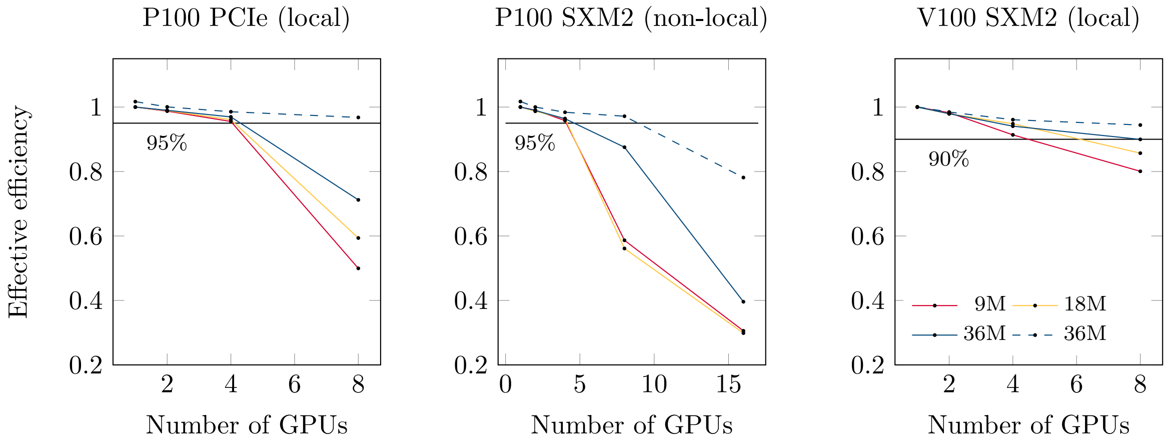


Figure 5: Strong scaling results for the empty channel for three selected grid sizes: parallel efficiency in relation to the single-GPU run with the same workload (voxels/GPU, effective efficiency).

Apart from efficiency considerations, strong scaling tests immediately reveal the potential of further accelerating the simulation under considerations, i.e., reducing the time-to-solution. In Figure 6, the total performance of the solver for the empty channel case is plotted. Again, good strong scaling properties can be observed. Moreover, it can be observed that the scaling strongly depends on the test case configuration and, more specifically, the ratio of computational load to communication overhead. Exemplarily, we compare the baseline 36 M case with an aspect ratio of 6:3:2 with a stretched channel case consisting of 36 M voxels as well, but with an aspect ratio of 24:3:2. The latter has a significantly reduced communication overhead because of the lower surface-to-volume ratio per subdomain. Therefore, while the compact channel case shows a clear performance saturation on the Pascal GPUs above four GPUs, the stretched channel case shows much better scaling (dashed line in Figures 4 – 6). On the DGX-1 with Volta GPU architecture, the difference between the two setups is significantly less pronounced due to the higher bandwidth of the second-generation NVLink GPU interconnect.

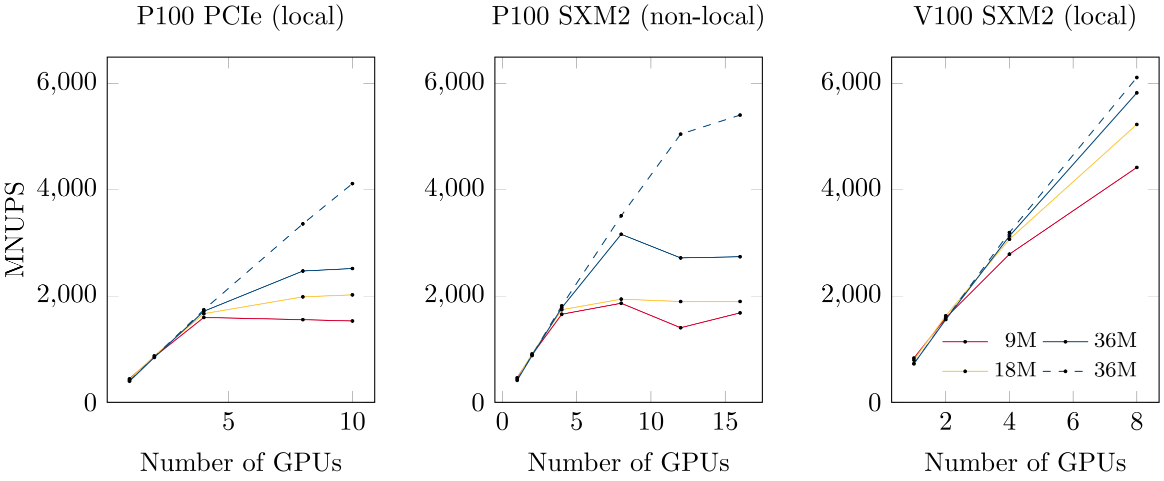


Figure 6: Strong scaling results for the empty channel case: total performance for two different test case configurations with an aspect ratio of 6:3:2 (solid lines) and, exemplarily, an aspect ratio of 24:3:2 (dashed line).

3.2 Realistic wind tunnel with a generic car geometry

The scaling analysis of the channel flow simulations showed excellent scaling and a high code performance. For production cases, several computationally intensive features are added to the simulation setup that, in addition, add load imbalances. To assess the impact of the advanced automotive features on the scaling behavior of ultraFluidX, a generic roadster geometry (Figure 7) is used in the following. Four grid resolutions with 36 M, 71 M, 143 M, and 286 M voxels were selected for the analysis. All cases have six grid refinement levels in addition to the coarse background mesh, where the majority of voxels resides in levels three to six (approx. 12 %, 30 %, 34 % and 10 % of the total voxels, respectively). On the highest refinement level, the chosen resolution per case results in voxel sizes of 3.1 mm, 2.5 mm, 2.0 mm and 1.6 mm, where the corresponding time step sizes are 5.3 ns, 4.2 ns, 3.3 ns and 2.5 ns. For these cases, the performance is averaged over 100 iterations, which is sufficient because of the larger compute time per iteration compared to the empty wind tunnel case.

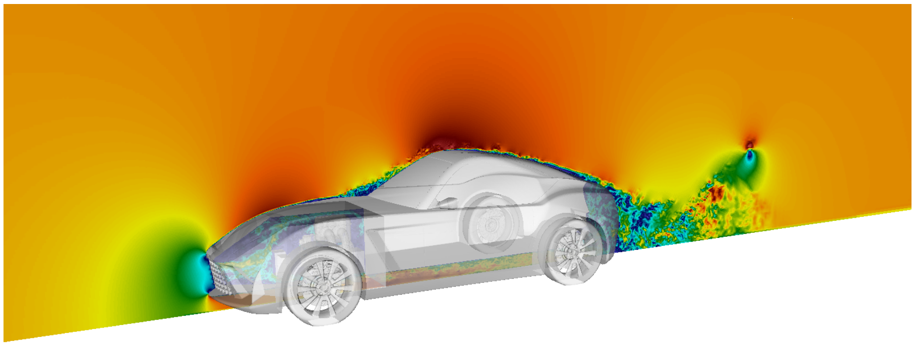


Figure 7: Generic roadster geometry in the numerical wind tunnel. The section cut shows the instantaneous velocity magnitude in the y-normal symmetry plane of the wind tunnel.

The strong scaling results are depicted in Figure 8. This time, again the total performance is shown, as it corresponds to the most relevant property of such a simulation: the acceleration and reduction in time-to-solution. Exemplarily, the 143 M voxel grid is discussed here. More than 80 % strong scaling efficiency is observed for all configurations when switching from four to eight GPUs. This is independent of the GPU architecture (Pascal vs. Volta) or the parallelization (single-GPU vs. multi-GPU). Overall, only approx. 2.6 hours of compute time would be necessary to simulate one second of physical

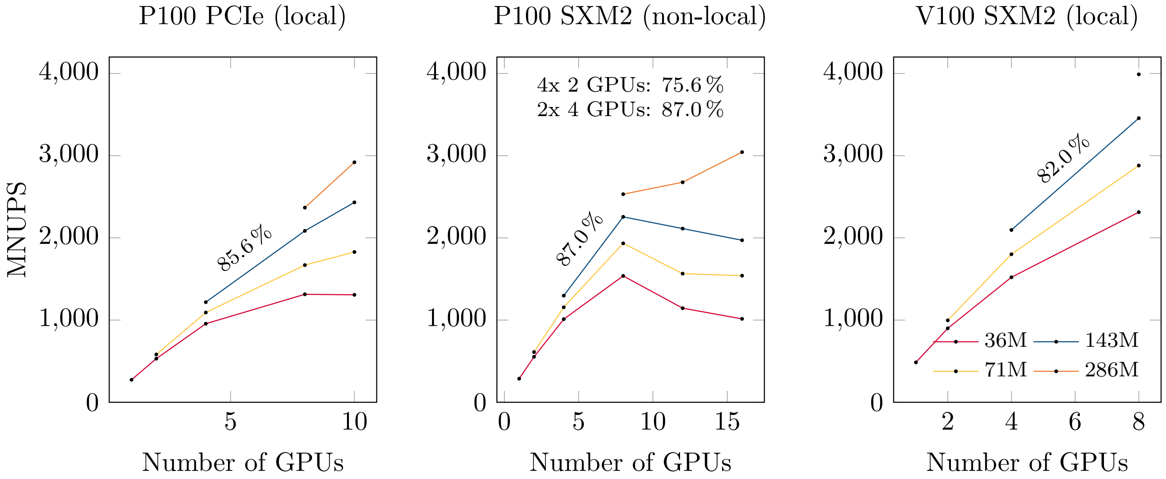


Figure 8: Strong scaling results for the roadster case: total performance and parallel efficiency (for the scaling from four to eight GPUs).

time at the real Mach number of the case (e.g., for aeroacoustic analyses). If only the aerodynamics are of interest, a further reduction of the time-to-solution by a factor of two and more is possible through the usage of an elevated Mach number, a common technique in LBM simulations.

4 SUMMARY AND CONCLUSIONS

This paper is concerned with the scaling analysis of the massively-parallel multi-GPU solver ultraFluidX. ultraFluidX can achieve turnaround times of just a few hours for simulations of fully detailed production-level passenger and heavy-duty vehicles. As far as scaling efficiency is concerned, ultraFluidX shows a very good weak scaling behavior with a parallel efficiency above 95 % for an empty wind tunnel case. Even for a realistic automotive production case, ultraFluidX scales very well up to eight GPUs on all tested hardware configurations, no matter if in a shared-memory system or a massively-parallel multi-node configuration. The strong scaling efficiency is found to be above 80 % for realistic grid resolutions and hardware configurations. Selected applications of ultraFluidX will be presented at the ECFD conference, including passenger vehicles and full truck geometries [6]. The resulting turnaround times are very competitive and allow to embed even highly-resolved, high-fidelity CFD simulations into a simulation-based design cycle.

REFERENCES

[1] Martin Geier, Martin Schönherr, Andrea Pasquali, and Manfred Krafczyk. The cumulant lattice Boltzmann equation in three dimensions: Theory and validation. *Computers & Mathematics with Applications*, 70(4):507–547, 2015.

[2] Martin Geier, Andrea Pasquali, and Martin Schönherr. Parametrization of the cumulant lattice Boltzmann method for fourth order accurate diffusion part I: Derivation and validation. *Journal of Computational Physics*, 348:862–888, 2017.

[3] Martin Geier, Andrea Pasquali, and Martin Schönherr. Parametrization of the cumulant lattice Boltzmann method for fourth order accurate diffusion Part II: Application to flow around a sphere at drag crisis. *Journal of Computational Physics*, 348:889–898, 2017.

[4] Orestis Malaspinas and Pierre Sagaut. Consistent subgrid scale modelling for lattice Boltzmann methods. *Journal of Fluid Mechanics*, 700:514–542, 2012.

[5] Orestis Malaspinas and Pierre Sagaut. Wall model for large-eddy simulation based on the lattice Boltzmann method. *Journal of Computational Physics*, 275:25–40, 2014.

[6] Christoph A. Niedermeier, Hanna Ketterle, and Thomas Indinger. Evaluation of the Aerodynamic Drag Prediction Capability of different CFD Tools for Long-Distance Haulage Vehicles with Comparison to On-Road Tests. In *SAE 2017 Commercial Vehicle Engineering Congress*.