Efficient solvers for coupled problems in respiratory mechanics

This project is about performance improvements for simulations of the human lung. The computational model is a highly resolved three-dimensional finite element discretization of all relevant physical processes, namely fluid flow in flexible airways (fluid-structure interaction), lung tissue inflation and volume constraints that link the air volume flow between airway and lung tissue. This highly nonlinear equation is solved by a monolithic approach in each time step. The realization of the model combines extensively validated modules for fluid and structural dynamics with solver components from the Trilinos package, including algebraic multigrid methods and parallel linear algebra.

In this project, improved preconditions for coupled block systems in multiphysics scenarios have been developed. In addition, the performance analysis showed that the modular approach, while ensuring correctness, involved a relatively high overhead in terms of setup of matrix structures in each nonlinear iteration step with expensive communication and index translation. Therefore, new algorithms have been developed that significantly reduce setup costs by re-using the matrices from the first time step with opportunistic combine strategies that keep the modular concept - for example, different applications that need to change the connectivity structures in each step can still use the same structural code. The developments have resulted in a code that runs up to twice as fast on simulations that take a few weeks on several hundreds of processor cores.

The code Indexa is a testbed for the development of our next-generation of fluid dynamics and multiphysics application codes. It discretizes the partial differential equations using high-order discontinuous Galerkin methods. In this method, integrals of weak forms are computed on elements and the element surfaces (faces) using fast tensorial quadrature. These fast integration kernels are used for operator evaluation and replace the (sparse) matrix-vector products in iterative solvers, including multigrid solvers. On x86 CPUs, the kernels are explicitly vectorized over several elements and faces by intrinsics.

Even though integration increases the operation count as compared to a sparse-matrix vector products, the final application performance has been shown to increase in almost all cases due to a reduced memory transfer. In terms of memory access requirements, this can be thought of as going from a sparse matrix kernel to a stencil representation, albeit with more complicated mathematical operations and somewhat different access patterns. The goal of the code analysis is to explore thread parallelism possibilities, vectorization options, and design decisions in terms of future throughput-oriented hardware.

Dr. Martin Kronbichler and Prof. Wolfgang A. Wall. TUM

Porting to Multi-GPUs system: