VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Toward Scalable Voxelization of Meshes with High Growth Rate

SC22 Dallas, TX | hpc accelerates.

Markéta Hrabánková | Ondřej Meca | Tomáš Brzobohatý | Lubomír Říha | Milan Jaroš | Petr Strakoš

MOTIVATION

The poster presents a scalable approach that converts the results of large-scale Computational Fluid Dynamics (CFD) simulations into a volumetric representation used by volume rendering-based visualization. Even if this functionality is provided by standard post-processing tools, its efficient parallelization requires an appropriate load-balancing. Unfortunately, load-balancing according to the number of cells or cell volumes does not scale for unstructured meshes with high growth rates typical in CFD. In the poster, we show that with an appropriate redistribution of data among available resources, it can operate in just several seconds with significantly improved scalability.

OpenFOAM

MESIO | OpenVDB | blender

Scalability tests of voxelization were performed on a Siemens electric motor model. The use case was prepared in ANSYS and computed by OpenFOAM. The model has 120 million cells. The results were stored in 1024 directories in the OpenFOAM parallel format. The total database size was 41GB.

STEP 01

CFD

Many engineers use numerical methods to simulate physical phenomena, such as Computational Fluid Dynamics (CFD). The simulation process is computationally demanding as high-precision simulations require detailed and extensive models. It is, consequently, common to use HPC infrastructure. The utilization of HPC infrastructures can significantly speed up the whole visualization process.

STEP 02

DATA

The ever-increasing precision enables the simulation of more complex problems in more detail. Still, it also leads to a large amount of resulting data that needs to be post-processed and visualized. Finite volume method (FVM) results are typically stored as values together with unstructured mesh databases that are supported by standard visualization tools. We can divide databases into sequential and parallel, where parallel databases have cells decomposed into domains that can be processed separately if required. Sequential databases are without any special cell ordering allowing direct parallel processing.

STEP 03

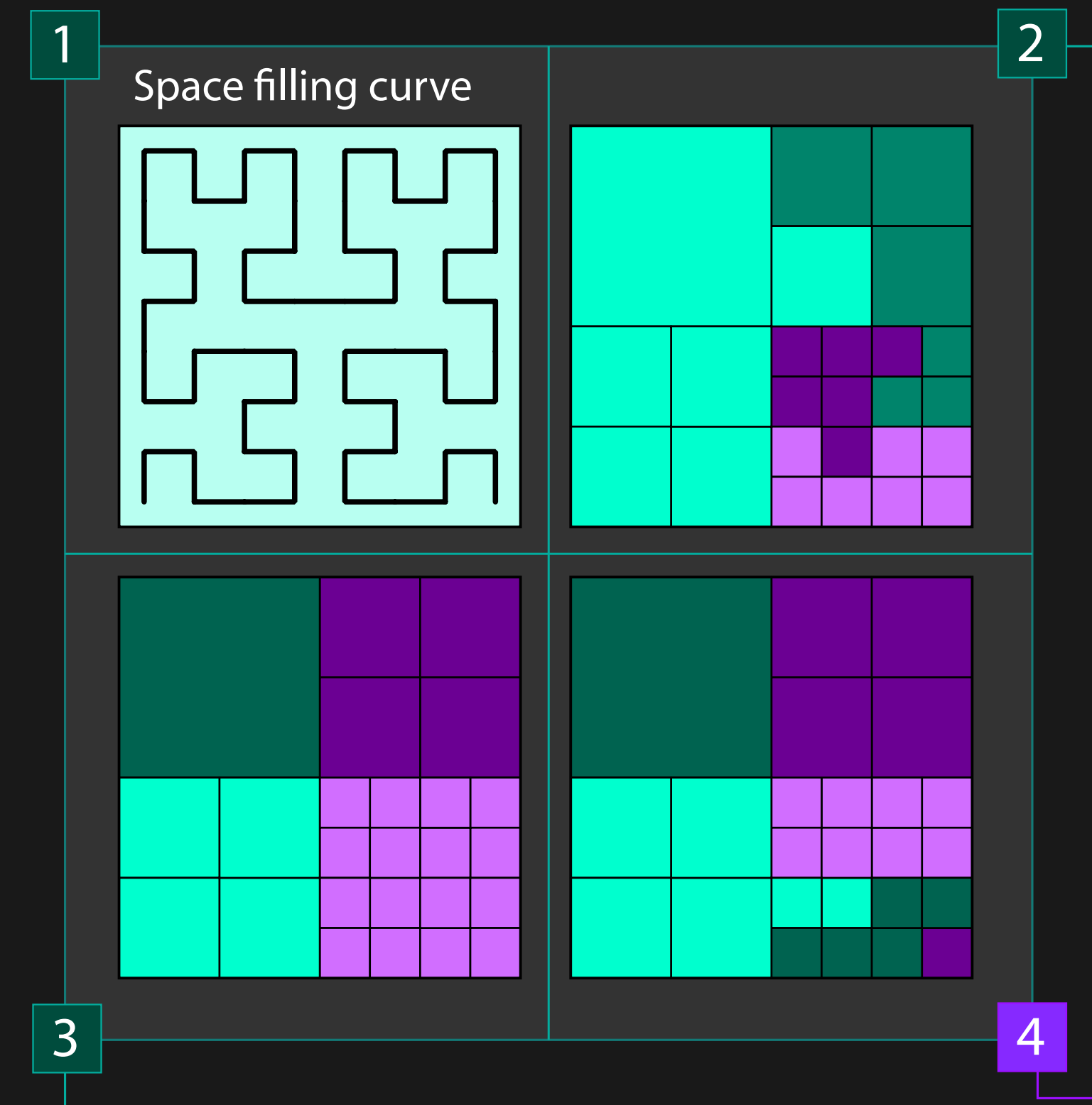
LOAD

Parallel databases can be straightforwardly loaded in parallel (each domain by a separated loader). Sequential databases are usually loaded by a single process only. Then, cells can be redistributed into workers allowing their parallel processing. Our tool, MESIO provides highly parallel loading of both sequential and parallel databases. Loading is based on domain decomposition techniques utilizing Hilbert's space-filling curve (SFC) and parallel histogram sorting. It assures good load-balancing together with a reasonable small number of neighbouring processes for each MPI process.

STEP 04

FILTERS

For the visualization, users typically apply a sequence of filters (e.g., computing streamlines, iso-surfaces, etc.). In this poster, we focus on voxelization (the process of re-sampling the original data to a structured, regular grid of voxels). Applying this filter in parallel is challenging, especially for meshes with significantly different cell sizes, which is common in CFD (high growth rate). The simple load-balancing according to the number of cells or cell volumes does not work since the complexity of the voxelization is dependent on both the number of cells and the domain volume. A hybrid domain decomposition must be used.



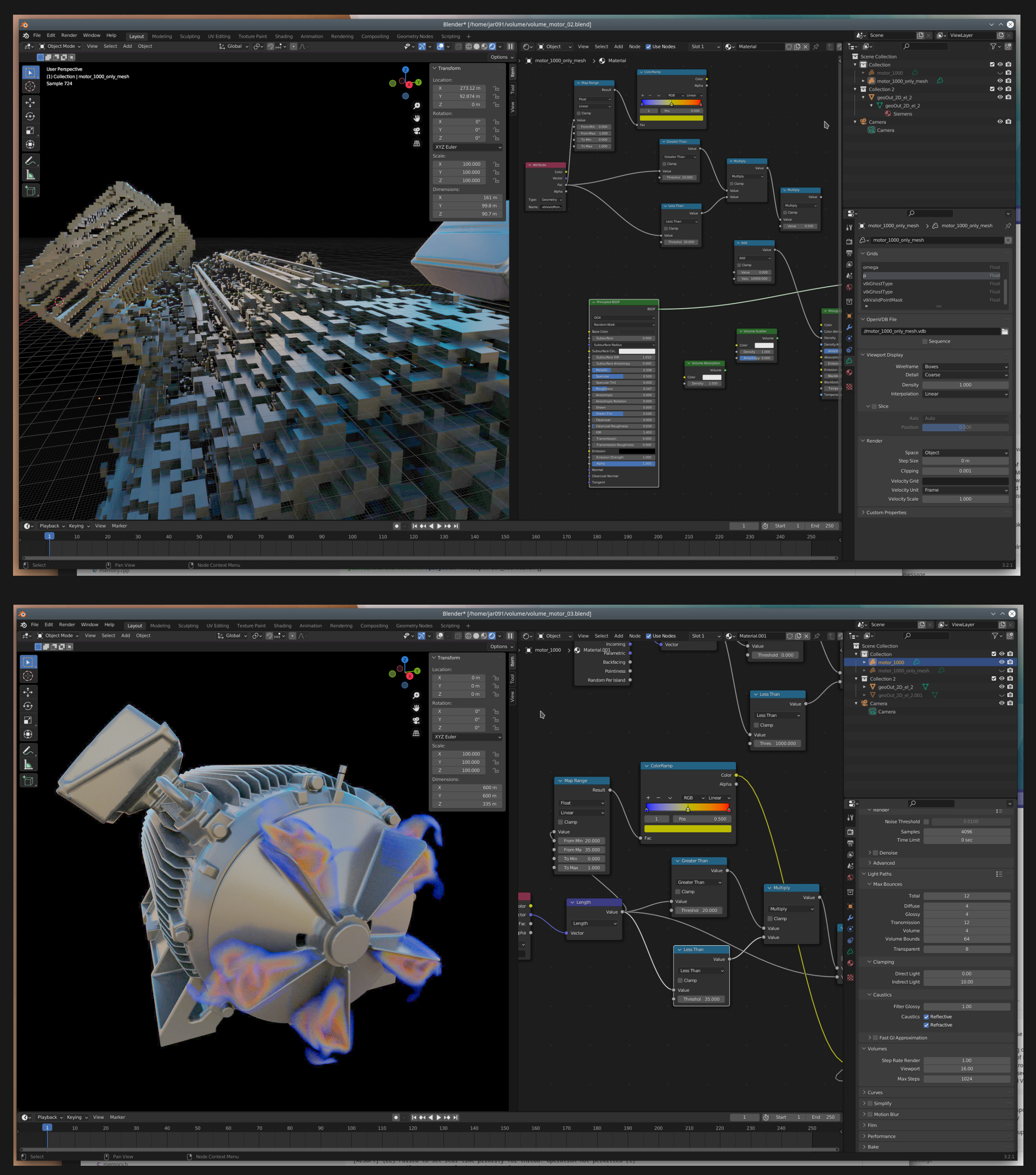
The balancing respecting the number of elements and domain volume has been implemented in our tool MESIO. Balancing domains according to both criteria and scalable redistribution and mesh processing allow for voxelized unstructured meshes with significantly different cell sizes in several seconds.

Domain decomposition with geometry volume balancing

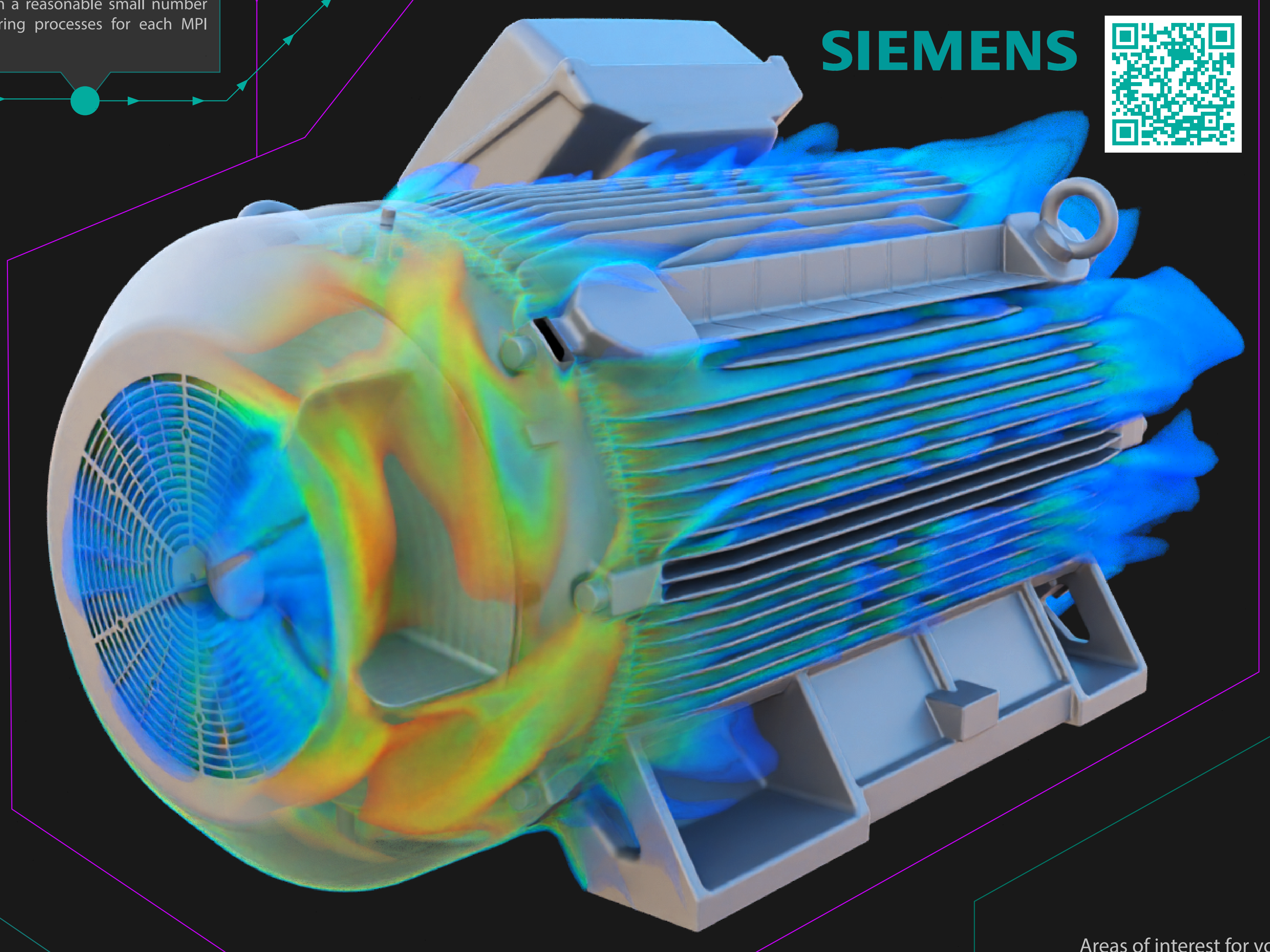
Domain decomposition with cell count balancing

Hybrid domain decomposition, geometry volume and cell count balancing

Blender workflow for volume rendering



To take full advantage of the HPC cluster, we extended the Blender Cycles renderer to support OpenMP, MPI and CUDA Unified Memory (UM). It supports rendering massive scenes on the GPU. Blender supports reading the OpenVDB format and converting to NanoVDB structures for efficient GPU rendering. Using remote interactive rendering, data can be displayed, coloured or filtered using Blender's tools such as the Material/Shader editor.



STEP 05

VISUALIZATION

Voxelized mesh can be stored, e.g., to OpenVDB database that can be imported into, e.g., publicly available tool Blender for cinematic-style visualization using volume rendering as implemented in Blender Cycles path-tracing renderer.

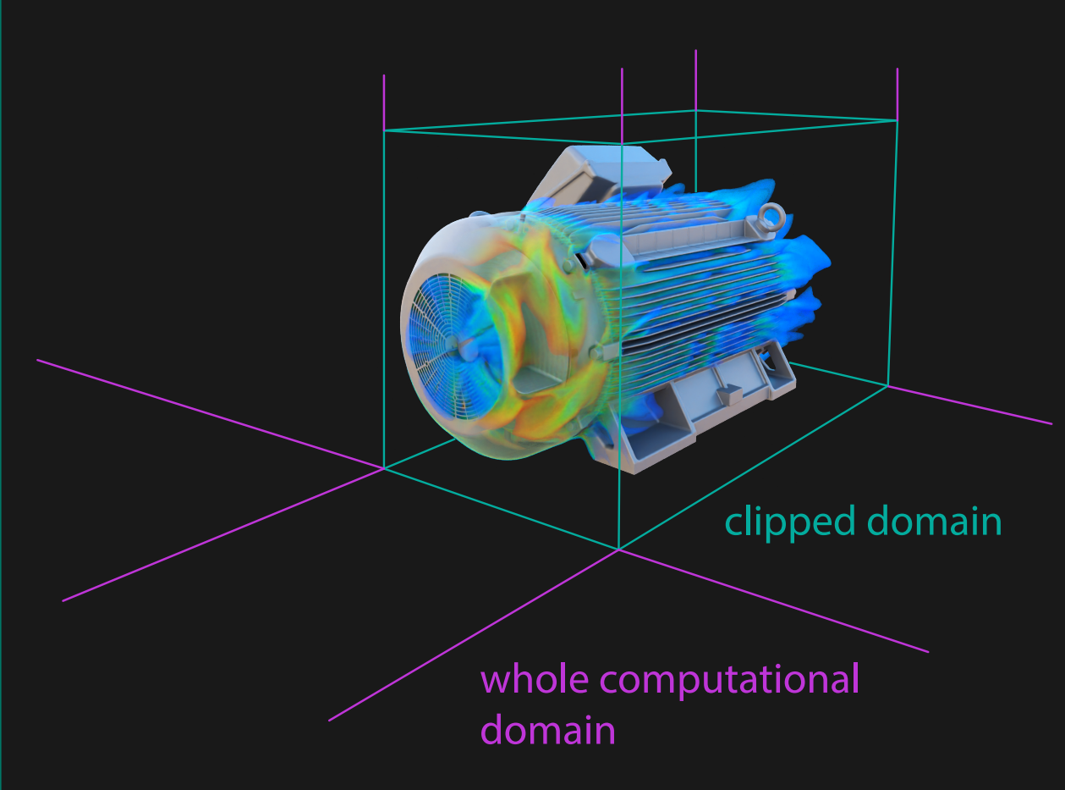
At IT4Innnovations, with the utilization of HPC infrastructure, we use Blender for interactive GPU accelerated rendering of high-resolution sparse volumes.

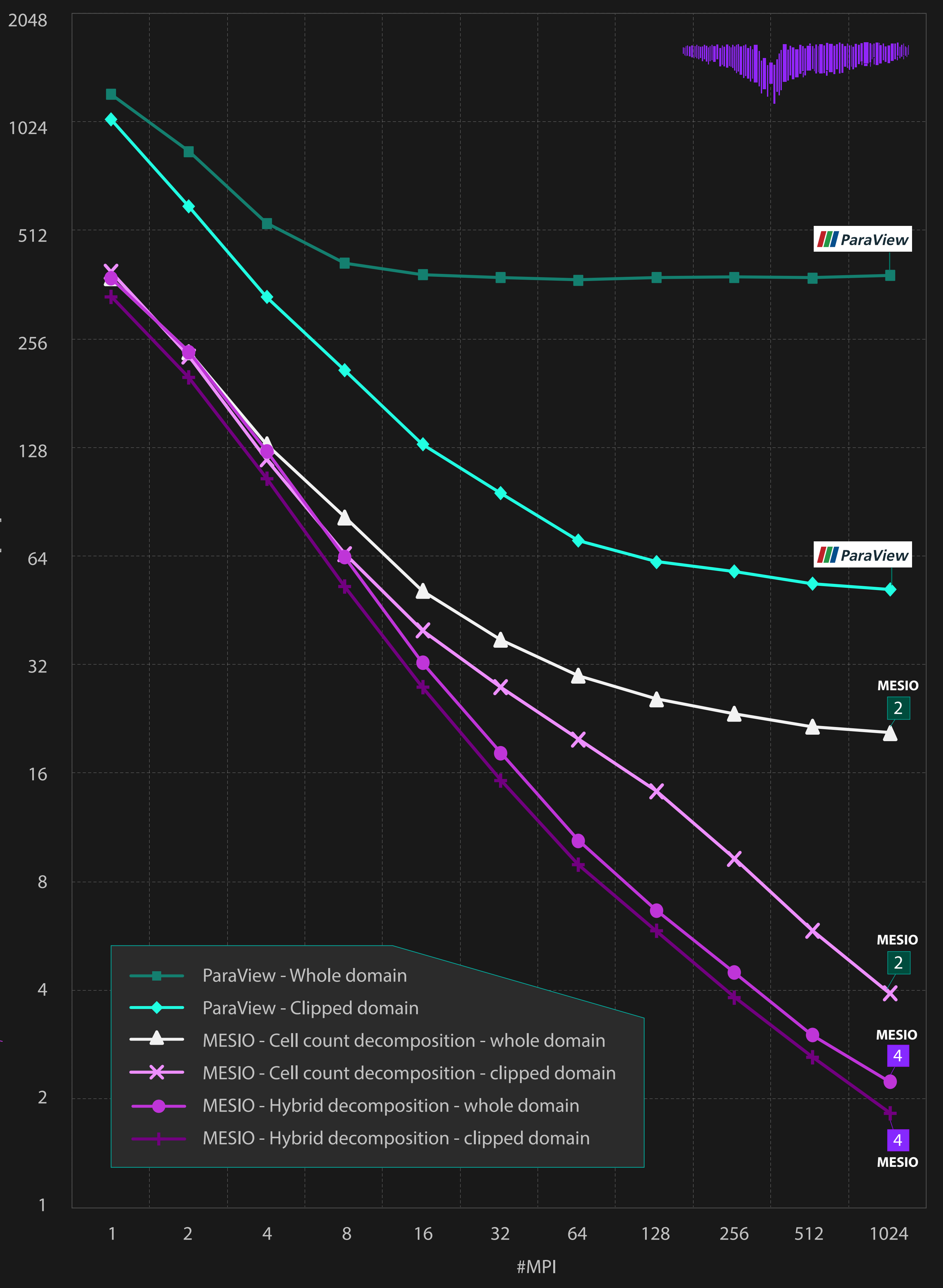
Voxelization Filter Application - Parallel Scalability

We compare our approach with the commonly used open-source tool ParaView which OpenFOAM developers recommend to visualize the CFD results. It supports parallel data processing of ParaView databases. Chart show performance of voxelization (filter resampleToImage in ParaView). We ran tests with two settings: (a) voxelization of the whole domain and (b) voxelization of the clipped domain. Both domains were resampled to 1000^3 voxels. In the case of the clipped domain, only a box around the electric motor was selected. It cuts off the largest cells that are far from the electric motor.

As can be seen in the graph, for meshes with a high growth rate, voxelization stops scale early. In the case of ParaView, only 16 processes can be utilized effectively. The scalability of the clipped domain is much better. In the case of the hybrid domain decomposition, the scalability of both whole and clipped domains is the same (absolute times are better for clipped domains since the smaller number of cells is processed).

Areas of interest for voxelization





This work was supported by the Doctoral grant competition VSB-Technical University of Ostrava, reg. no. CZ.02.2.69/0.0/0.0/19_073/0016945 within the Operational Programme Research, Development and Education, under project DGS/TEAM/2020-008 "Development of a tool for scientific data processing and visualization in VR with multi-user support" and by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140).