MESIO: highly-parallel loader of unstructured meshes

Ondřej Meca[1], Ondřej Kozinski[1], Lubomír Říha[1], Tomáš Brzobohatý[1]

IT4Innovations, VSB – Technical University of Ostrava, Ostrava, Czech Republic
{ondrej.meca,ondrej.kozinski,lubomir.riha,tomas.brzobohaty@vsb.cz}

Unstructured meshes are commonly used in numerical methods such as the finite element method, finite volume method, etc. Decades of development has established a rich set of database formats used by both theoretical researchers and mainstream engineers from industrial practice. Formats were created with respect to the requirements of a given tool, and many of them were created before the extensive development and adoption of HPC technologies. Nowadays, it complicates their optimal usability from the HPC point of view since formats were optimized mainly for sequential tools and they are not prepared for reading by massive parallel libraries. At the same time, adoption of other (more HPC-centric) formats for mesh databases is rarefied, as many engineering professionals understandably prefer to use well known and tested sequential modeling tools for the most important part of the design pipeline: creation of high-quality numerical models.

Technically, parallel libraries usually require a particular data organization within a database. Mesh elements should already be decomposed into domains, and data should be arranged in a pattern that allows simple parallel reading with minimal communication and synchronization steps. Due to the availability of many mature tools for domain decomposition, if required, it is relatively easy to build a converter that loads an input database, prepares a mesh for a decomposer, calls the decomposer, and stores decomposed data in a selected parallel database. The disadvantage of such a converter is its sequential behavior during parsing and preparation of a mesh for a decomposer. Unfortunately, for this non-trivial task, there is no tool that is able to perform these two steps for a general database in parallel. It greatly slows down, and in some cases even inhibits (e.g., due to memory limits) the connection between tools that generate sequential mesh databases and parallel solvers. This is one of the key factors that contributes to the low interest in HPC resources from the mainstream engineering community, which desire to use mature (mostly sequential) tools for the preparation of models.

The key technical contribution of the MESIO tool is an algorithm that avoids all sequential parts of the above described conversion process. The algorithm is general enough to be able to load an unstructured mesh from an arbitrary database format. Using the library, one can obtain the fully parallel loader with the ability to prepare data for his/her favourite parallel decomposer (which provides a decomposition of a requested quality). Using the library, it is possible to completely skip conversion to a database suitable for a parallel solver since differences between loading the original (sequential) database and a parallel one are negligible.

The poster presents how to use the MESIO tool in various scenarios and demonstetes that it is possible to efficiently reconstruct meshes with thousands of million nodes and elements in several seconds on thousands of processors, even from databases that were not designed to be read in parallel.