TTK is Getting MPI-Ready

Eve Le Guillou\textsuperscript{1,2}, Michael Will\textsuperscript{3}, Pierre Guillou\textsuperscript{4}, Jonas Lukasczyk\textsuperscript{3}, Pierre Fortin\textsuperscript{2}, Christoph Garth\textsuperscript{3}, and Julien Tierny\textsuperscript{1}

\textsuperscript{1}CNRS, LIP6 – Sorbonne Universités, UPMC, CNRS – France
\textsuperscript{2}CRIStAL – Inria, UMR 9189 - CRIStAL - Centre de Recherche en Informatique Signal et Automatique de Lille – France
\textsuperscript{3}RPTU Kaiserslautern-Landau – Allemagne
\textsuperscript{4}Laboratoire d’Informatique de Paris 6 (LIP6) – Sorbonne Université, Centre National de la Recherche Scientifique : UMR7606 – France

Résumé

This system paper documents the technical foundations for the extension of the Topology ToolKit (TTK) to distributed-memory parallelism with the Message Passing Interface (MPI). While several recent papers introduced topology-based approaches for distributed-memory environments, these were reporting experiments obtained with tailored, mono-algorithm implementations. In contrast, we describe in this paper a versatile approach (supporting both triangulated domains and regular grids) for the support of topological analysis pipelines, i.e., a sequence of topological algorithms interacting together, possibly on distinct numbers of processes. While developing this extension, we faced several algorithmic and software engineering challenges, which we document in this paper. Specifically, we describe an MPI extension of TTK’s data structure for triangulation representation and traversal, a central component to the global performance and generality of TTK’s topological implementations. We also introduce an intermediate interface between TTK and MPI, both at the global pipeline level, and at the fine-grain algorithmic level. We provide a taxonomy for the distributed-memory topological algorithms supported by TTK, depending on their communication needs and provide examples of hybrid MPI+thread parallelizations. Detailed performance analyses show that parallel efficiencies range from 20\% to 80\% (depending on the algorithms), and that the MPI-specific preconditioning introduced by our framework induces a negligible computation time overhead. We illustrate the new distributed-memory capabilities of TTK with an example of advanced analysis pipeline, combining multiple algorithms, run on the largest publicly available dataset we have found (120 billion vertices) on a standard cluster with 64 nodes (for a total of 1536 cores). Finally, we provide a roadmap for the completion of TTK’s MPI extension, along with generic recommendations for each algorithm communication category.