Development of a Scalable Code Framework with Decentralised Domain Management for Numerical Computation on modern HPC Clusters

Name: Christoph Ertl
E-Mail: christoph.ertl@tum.de
Supervisor: Prof. Dr. rer. nat. Ernst Rank
Chair for Computation in Engineering
Started: 08/2016
☒ ongoing ☐ finished

Background
One of the main challenges in numerical computing on modern high performance clusters for the simulation of real world phenomena is the efficient management of the simulation domain. This includes the subdivision of the domain, its distribution and the continued balancing of workload per computational resource during runtime — all while minimising the communication between the participating units.

Classical approaches either store the complete domain topology with each unit, or use a dedicated subset of units for bookkeeping. This entails a significant memory requirement and costly global communication overhead to ensure consistency.

Decentral Domain Management
This work addresses the aforementioned shortcomings by employing a decentral approach to domain management. The essential idea is to limit the domain view of each participating unit to their direct neighbours. Transfer of data and updates of topology are only realised between them. Global updates are not necessary. Since there is an upper bound to the amount of neighbours each subdomain can have, regardless of total domain size, the approach promises to scale even when computing on the largest clusters.

Load-balancing however becomes more involved as the mainly used strategy Space-Filling-Curves rely on complete topological knowledge. Therefore, a heuristic concept is proposed that determines the best targets for the transfer of sub-domains to computing units, weighting an optimised balance in terms of computational work with minimising communication cost when applying stencil based computational kernels.
Initial Results

The decentral management facilities are being implemented on top of a code framework for fluid flow simulations specifically designed for massive parallel computers featuring a dedicated hierarchic data structure [1].

So far all facilities necessary to communicate topology alterations to neighbouring units are in place [2]. These include refinement, coarsening and transfer of subdomains to different units. A test case, where a cubic domain of varying resolutions has been completely refined – increasing the resolution by a factor eight – has shown the viability of the idea.

In Fig. 1 the results for the decentral management are illustrated. The time it takes to update the topology information is lowered when more processes are used. The sub-domains per process decrease, so is the communication effort of the direct communication between affected processes.

Fig. 1: Domain update times with decentral management.

Fig. 2: Domain update times using central management.

The central approach shown in Fig. 2 struggles as expected. Communication effort per process decreases likewise, the number of update queries to the central management system rise however, leading to the increase in time for the domain update.

References
