MPI-optimisation of the CFD software package MGLET based on grid-locality

Dr.-Ing. Yoshiyuki Sakai†

3rd July 2019

† Chair of Hydromechanics, Technical University of Munich, 80333 Munich, Germany.
Email: `yoshiyuki.sakai@tum.de`

Background

MGLET is our in-house CFD code that has been primary developed and maintained at the Chair of Hydromechanics (Prof. M. Manhart). The code is currently being used by several research groups: At Chair of Hydromechanics of Technical University of Munich, for instance, turbulent flow through complex geometries, flow in porous media, and fibre suspensions in fluid media have been investigated using MGLET. The groups of Prof. Helge Andersson and Prof. Bjørnar Pettersen (both NTNU Trondheim) use the code to predict and analyse bluff-body flows primarily using direct numerical simulation (DNS) and immersed boundary method (IBM). At the Institute for Atmospheric Physics (DLR Oberpfaffenhofen), aircraft wake vortices are investigated including their interaction with atmospheric boundary layers and ground effects. These applications demonstrate the code's excellent numerical efficiency and adaptability to the diverse hydrodynamic problems. Currently, MGLET exhibits a sufficient parallel efficiency of $\approx 50\%$ up to a problem size of ≈ 17 billion discrete cells distributed over approximately 33000 parallel processes.

MGLET employs a finite-volume method to solve the divergence form of the incompressible Navier-Stokes equations for the primitive variables (i.e. three velocity components and pressure), in order to simulate complex flow phenomena within an arbitrarily shaped domain. The code is capable of performing DNS and LES (Large-Eddy Simulation) of complex turbulent flows, which can be optionally coupled with transport of some scalar quantities.

A conventional domain decomposition is adopted for parallelisation, which is combined with a local grid refinement strategy that is done by adding grid boxes with finer resolutions in an octree-like, hierarchical and overlapping manner. This means one parent grid box can have maximum 8 child grid boxes in 3D. Henceforth we refer to those grid boxes or subdomains as grids, whereas the degree of grid refinement as the grid levels. Such grid arrangement allows us to distribute grid cells in the simulation domain in a optimal way with little management overhead. This leads to two distinct types of grid-wise communications, namely: *intra-level* communication between the boundaries of neighbouring grids, and *inter-level* boundary and volume communication between adjacent grid levels. The grids are then distributed among parallel processes in such a way that the number of grid cells for each process becomes as homogeneous as possible, in order to balance the memory consumption and the computational effort.

In this project, we aim to improve the grid distribution algorithm in such a way that the grid locality is also taken into account, in order to maximise the performance in the NUMA (non-uniform memory access) architecture where the memory access speed depends on the relative location of processors in the network. For instance, SuperMUC Phase 2 utilises Haswell processor with 28 cores per node, and also consists of 512 nodes per island (14336 cores per island). The memory bandwidth within a node is at 137 GByte/s. All computing nodes within an individual island are connected via a fully non-blocking Infiniband network (FDR14 at 55 Gbps). Above the island level, the pruned interconnect enables a bi-directional bi-section bandwidth ratio of 4:1

(intra-island / inter-island). Such non-uniformity of access speed motivates us to distribute the grids that perform frequent communications as close as possible in the relative location in the network. We expect that this modification will improve MGLET's parallel-scalablity even further.

Tasks

- implement a new grid distribution algorithm
- validate the simulation results with the original version
- perform parallel scaling evaluation to measure the improvement in various grid configuration

Required skills

- basic knowledge in UNIX/Linux system
- experience in scripting languages (e.g. MATLAB, Python, R)
- sufficient communication skill in English
- patience to work with research code

Recommended skills

- basic knowledge in compiled languages (e.g. FORTRAN, C/C++)
- basic knowledge in fluid dynamics and/or CFD
- experience in CFD application (e.g. OpenFOAM, Fluent)

Benefits for applicant

- practical experience in high-performance computing
- interaction with HPC experts at LRZ
- practical experience in CFD
- in-depth supervision in scientific computing and turbulence research
- ideal preparation for Master's thesis

Contact

Dr.-Ing. Yoshiyuki Sakai
Associate Professorship of Hydromechanics
Technical University of Munich (TUM)
Arcisstr. 21, 80333 Munich, Germany
Tel: +49 (0)89 289 25278
Email: yoshiyuki.sakai@tum.de