Hybrid Parallelization of the CFD Code PANTA

Hans Thermann[1], Bernd Wickerath[1], Daniel Grates[1], Stephan Schmidt[1], Dieter an Mey[2]

1. Institute for Jet Propulsion and Turbomachinery, Aachen University of Technology
2. Computing Center, Aachen University of Technology

1 Introduction

PANTA is a well vectorized 3D Navier-Stokes solver that is extensively used in the modeling of turbomachinery [2,3].
As the underlying geometry provides a natural decomposition of the computational domain it is well suited for parallelization.

The program structure depicted in Fig. 1 contains a couple of nested loops which might be exploited for parallelization. The numbers in parenthesis indicate the typical iteration count of the corresponding loops.

- Preparation, intput

- Computation
- S loop over footsteps (~1200)
- S loop over linearized Newton steps (1-4)
- (MPI) loop over blade rows (1-7)
- MPI loop over blade channels (1-5)
- (MPI/OMP) loop over geom. blocks (1-4)
- MPI/OMP loop over zones (1-80)
- S iteration loop (10-12)
- V/OMP lin. equ. solver etc

- Postprocessing, output

Fig. 1: PANTA code structure

In order to use a parallel machine with many processors, multiple levels of parallelization have to be combined. The current version exploits only three levels.
The most time-consuming routines had been well vectorized before, but they also have a high potential to be parallelized at loop level on a shared-memory system. As the program has many time-consuming routines, it is not possible to parallelize all of them by hand, auto-parallelization is inevitable.
Combining automatic and explicit parallelization with OpenMP is also feasible, but an integration of explicit parallelization with OpenMP directives and auto-parallelization with the support of standardised directives would be very helpful.
Because nested parallelism with OpenMP is not yet supported by the majority of current production (pre) compilers, MPI has to be used for the coarse-grained parallelization levels, if OpenMP is employed at a lower level.

2 Objectives for the Participation in OMPlab

The alternative parallelization using OpenMP for one of those loops (loop over the zones), which have been parallelized with MPI so far, is currently under way. The usage of shared memory at this level will save a lot of memory consumption. We also want to investigate and compare the performance and the parallelization effort of both approaches. As the Fujitsu compiler, which will be available in the lab session, is the first commercial compiler supporting nested parallelism, this will be a nice opportunity to carry out such experiments.

3 References

1. Institute for Jet Propulsion and Turbomachinery, Aachen University of Technology (RWTH) http://www.ist.rwth-aachen.de_[/en/forschung/instationaere_li.html]
2. Computing Center, Aachen University of Technology (RWTH) http://www.rz.rwth-aachen.de
3. Volmar, T., Brouillet, B., Gallus, H.E., Benetschik, H.: “Time Accurate 3D Navier-Stokes Analysis of a 1½ Stage Axial Flow Turbine”, AIAA 98-3247, 1998.
4. D. an Mey, S. Schmidt : “From a Vector Computer to an SMP-Cluster - Hybrid Parallelization of the CFD Code PANTA”, 2nd European Workshop on OpenMP, Edinburgh, 2000