8th. World Congress on Computational Mechanics (WCCM8)
5th. European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 – July 5, 2008
Venice, Italy

EVOLUTION OF A CFD CODE'S TOOL CHAIN TOWARDS HPC APPLICATIONS

***Yvan Fournier[1], Jérôme Bonelle[2] and Sofiane Benhamadouche[3]**

[1] EDF R&D,
Fluid Dynamics, Power Generation and Environment
6 quai Watier – BP 49
78401 Chatou Cedex
France
yvan.fournier@edf.fr

[2] EDF R&D,
Fluid Dynamics, Power Generation and Environment
6 quai Watier – BP 49
78401 Chatou Cedex
France
jerome.bonelle@edf.fr

[3] EDF R&D,
Fluid Dynamics, Power Generation and Environment
6 quai Watier – BP 49
78401 Chatou Cedex
France
sofiane.benhamadouche@edf.fr

Key Words: *High Performance computing, Computational Fluid Dynamics.*

ABSTRACT

EDF has a long history of in-house CFD codes, of which *Code_Saturne*[1] and NEPTUNE_CFD[2] represent the current generation. Development on *Code_Saturne*, a co-located finite volume solver using generic polyhedral meshes for incompressible turbulent flows, started in 1998. This code has been released under the GPL licence since early 2007. Work on the multiphase flow solver NEPTUNE_CFD started in 2001, as an EDF/CEA collaboration, and is based on the same architecture.

The base architecture, while rather recent, was thus initiated at a time (1998) when vector machines and Fortran77 were still the norm at EDF, though domain-splitting type parallelism was anticipated. Thus, care was taken not to depend on algorithms whose parallelization would be an issue. As memory use was a big potential stumbling block, the code was separated in two executables: a kernel, built around the minimal necessary mesh connectivity information, and a pre-post-processor, which handled conversion of external meshing tool output to the internal format, and conversion from internal data to visualization tool input. This pre-post-processor also handled concatenation and conformal joining of non-conforming meshes.

The first parallel version of *Code_Saturne*, V1.1 was only released in 2002. It already included full distributed parallelism through MPI for the code's kernel, but pre and post processing went through a serial phase, limiting possible mesh sizes. NEPTUNE_CFD was parallelized in 2003, sharing much of *Code_Saturne'*s infrastructure.

Since then, work has been ongoing to further parallelize the tool chain, starting with the generation of post-processing output, such that *Code_Saturne*'s recently released version 1.3 still requires a serial pre-processing phase, but all post-processing output is

done in parallel. We did not initially use MPI-IO, the choices being to avoid memory bottlenecks first, performance bottlenecks second, while remaining portable to the still numerous machines offering only MPI 1.2 functionality, but we are now starting to use it as an option in the next version's development.

Generation of ghost cells has also been rewritten and migrated from the serial preprocessor to the parallel kernel. Overall reduction of the preprocessor's functionality has allowed for various optimizations, leading to a 40% peak memory reduction compared to the prior version of the code, but this is only one of the major steps towards a fully parallel preprocessing stage. At least some domain splitting functionality will be migrated from the preprocessor to the kernel. Most importantly, concatenation and conformal joining of initially non-conforming meshes, one of our preprocessor's main assets, will be re-implemented around the parallel kernel, effectively removing the preprocessor memory bottleneck.

FVM ("Finite Volume Mesh"), the library developed as the basis for preprocessor functionality parallelization also includes fully distributed parallel point cloud in mesh localization, allowing for highly automatic mesh to mesh interpolation with no memory bottleneck. This is the basis for new approaches currently tested, such as chimera-type meshes, and zonal RANS-LES coupling, with partial overlap allowing for good transmission of turbulent structures from one mesh to the other. Single-phase and two-phase flow coupling should also be possible using a similar functionality (with "physical" issues to be dealt with).

Research issues include multigrid solvers, parallel repartitioning and improving load balancing, communication schemes for IO or point location type operations which may use specific temporary data distributions, and finally multilevel parallelism.

As will be detailed in this roadmap, much work remains to be done, but *Code_Saturne* and NEPTUNE_CFD are well on their way to enabling routine CFD calculations on very large meshes, much beyond our recent 100-million tetrahedral, 8192 processor run of *Code_Saturne* on EDF's IBM Blue Gene/L for a fuel assembly model mixing grid.

This work has been achieved partly in the framework of the development of *Code_Saturne* funded by EDF and partly in the framework of the NEPTUNE project, financially supported by CEA (Commissariat à l'Énergie Atomique), EDF (Électricité de France), IRSN (Institut de Radioprotection et de Sûreté Nucléaire) and AREVA-NP.

REFERENCES

[1] F. Archambeau, N. Mechitoua, M. Sakiz, "*Code_Saturne*: a Finite Volume Code for the Computation of Turbulent Incompressible Flows – Industrial Applications", *International Journal on Finite Volumes*., Vol. **1**, 2004.

[2] N. Méchitoua, M. Boucker, J. Laviéville, J. Hérard, S. Pigny, and G. Serre. "An unstructured finite volume solver for two-phase water-vapour flows based on an elliptic oriented fractional step method", *10th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH 10)*, October 2003