

Computational Steering of CFD Simulations inside Grid Computing Environments

Manuel J. García ¹, Juan Duque ², Pierre Boulanger ³

(1) Applied Mechanics Group, EAFIT University, Carrera 49 N-7 Sur-50, Medellín - Colombia
E-mail : mgarcia@eafit.edu.co

(2) Applied Mechanics Group, EAFIT University, Carrera 49 N-7 Sur-50, Medellín - Colombia
E-mail : jduquelo@eafit.edu.co

(3) AMMI Lab, University of Alberta, 221 Athabasca Hall, Edmonton, Alberta - Canada,
E-mail : pierreb@ualberta.ca

Short Abstract:

Simulation of complex phenomena it is a long process and has been traditionally made in batch mode. However, advances in computer processing and networking allow monitoring and altering simulation parameters of the computational process whilst it is running. This is called computational steering. By combining this capability with advanced communications tools it is possible for a group of scientists located across various continents to work collaboratively while visualising on-going simulations. These bring the possibility to share their experience and promote new ideas and solutions. In this paper, a collaborative computational steering environment specialized to solve CFD problems is presented.

Key words: Computational Steering, CFD, Collaborative Environment, High performance Computing, Virtual Wind Tunnel.

1- Computational Steering of CFD Simulations inside Grid Computing Environments

The requirement of studying fluid flow conditions around virtual prototypes is not new inside the engineering and academic communities: MIT's David OH with his Java Virtual wind Tunnel [OH1], NASA's Virtual Wind tunnel at AMES Research Center [BL1], CHAM (Concentration, Heat and Momentum) and more recently efforts from Rank and Wenisch et al. with their HPC (High Performance Computing) Implementation [RW1] [WV1], are just a few examples of the growing interest inside the community.

This paper presents an overview of the architecture used for the implementation of an Interactive Computational Fluid Dynamics (CFD) environment, intended for studies mainly related with shape optimization of the airflow around virtual prototypes. The idea behind this type of implementation is to constrain faster the design domain that a group of designers has to face at early stages of the product development, where highly accurate results are not required but fast estimation of the state variables are highly appreciated.

The structure of an automated server responsible for managing the CFD module and updating the simulation conditions, as they are modified by the user, is the core of the project and will be a matter of discussion. Data and instructions are collected by client applications that connect to a server (located on the same computer or on a High Performance Computing HPC facility) and this is in charge of managing the simulation and modify the parameters as information arrives.

2- A Client-Server Architecture

The goal of the environment is to develop interaction techniques for users on remote locations using a high speed network. As the users have different specifications of their available computational resources (operative system, computer capabilities, etc.) a Client/Server schema where only some specific data transactions and messages allowed becomes the natural choice for development. The main role of the server is to guarantee a smooth interaction between the users and the solver. The users are located at a client machine and the server and solver are both located on a remote machine. Three different process involved on the steering environment are identified:

1. Data I/O and Steering: This process is handled by the VWTClient (Paraview plugin application). The Client should be able to handle all the defined user data requests and inputs, guaranteeing a fluent interaction and allowing the user to perform the steering of the scenario he/she desires either defining from scratch a new scenario or using a previously defined one.

2. Data distribution and Simulation triggering: This process is handled by the VWTServer (ruby script) that is able to handle the user requests performed from remote locations. The Server is multi-user oriented and the interactions presented on this diagram are only performed with a single user (analogically, the same tasks are performed for the interaction with any user).

3. CFD scenario solving: After the developments of the VWT solver this has become a straightforward process when all the data is properly defined and set. The preprocessor/solver are programmed on C++ and can be used as independent tools when required

4- Conclusions

A collaborative workspace for CFD simulations/training represent significant opportunities to developing countries specially given the relatively low resources that have to be spent to collaborate with experts all across the globe. The present implementation uses grid infrastructure for the solver and clients distributed over the network. This architecture has provided useful information about the way users interact with simulation. Message passing time is key to guarantee stability and the depth of the interactive experience.

The Fixed-Grid preprocessor is a fast and reliable discretization method for 3D domains where the detail of the features of the object is not the main goal. The discretization is automatically calculated without requiring any user-based expertise or work. Domain decomposition allows this algorithm to be parallelizable and the computational load to be balanced between nodes. Defeaturing may present an advantage for both speed and for providing hints about what the shape might become during the CAE analysis. The solver developed uses the Finite Volume Method. Although it is stable and accurate for solving CFD phenomena, it has shown not to be the best choice in terms of minimizing solution time.

The same basic architecture can be extended to simulate different phenomenon apart from fluid mechanics. Finally, streaming video rendered on the Server machine seems to be a suitable way to avoid sending large data blocks over the network on clients which possess low computational resources and low bandwidth.

5- References

[BL1] Bryson S and Levit C. The virtual wind tunnel. volume 4, 1992.

[G1] Garcia M. Fixed Grid Finite Element Analysis in Structural Design and Optimisation. PhD thesis, Department of Aeronautical Engineering, The University of Sydney, March 1999.

[GD1] Garcia M, Duque J, Henao M, and Boulanger P. ParaVoxel: a domain decomposition based Fixed Grid preprocessor". submitted to International Journal of Computational Methods (IJCM)

[OH1] Oh David. The java virtual wind tunnel -a two dimensional computational fluid dynamics simulation-. Web, 2001. <http://raphael.mit.edu/Java/>

[RW1] Rank E, Wenisch P, Borrmann A, Duster A, And Treeck C. Computational steering: Towards advanced interactive high performance computing in engineering

sciences. In (WCCM8) June 30, 2008 Venice, Italy, June 2008.

[WV1] Wenisch P, van Treeck C., Borrmann A, Rank E, and Wenisch O. Computational steering on distributed systems: Indoor comfort simulations as a case study of interactive cfd on supercomputers. Int. J. Parallel Emerg. Distrib. Syst., 22(4):275{291, January 2007.