International Conference on Hydroinformatics

2014

Software Tools For Large Scale Interactive Hydrodynamic Modeling

Gennadii Donchyts

Fedor Baart

Arthur van Dam

Bert Jagers

Sander van der Pijl

11th International Conference on Hydroinformatics
HIC 2014, New York City, USA

SOFTWARE TOOLS FOR LARGE SCALE INTERACTIVE HYDRODYNAMIC MODELING

GENNADII DONCHYTS (1, 2), FEDOR BAART (1,2), ARTHUR VAN DAM (1), BERT JAGERS (1), SANDER VAN DER PIJL (1)

(1): Deltares, P.O. Box 177, 2600 MH, Delft, the Netherlands

(2): Civil Engineering and Geosciences, Delft University of Technology, Stevinweg 1, 2628 CN, Delft, the Netherlands

Developing easy-to-use software that combines components for simultaneous *visualization*, *simulation* and *interaction* is a great challenge. Mainly, because it involves a number of disciplines, like computational fluid dynamics, computer graphics, high-performance computing. One of the main characteristics of an interactive model is that it should provide immediate feedback to the user, for example respond to changes in model state or view settings. Features involving interaction during simulation are usually available for models with a relatively small number of computational cells and are used mainly for demonstration and educational purposes. The reason for that is that the time required to compute a single time step and render model results become significant when comparing to a simple model. It would be useful if interactive modeling would also work for models typically used in consultancy projects involving large scale simulations. This results in a number of technical challenges related to the combination of the model itself and the visualization tools (scalability, implementation of an appropriate API for control and access to the internal state). While model parallelization is increasingly addressed by the environmental modeling community, little effort has been spent on developing a high-performance interactive environment.

What can we learn from the other domains where visualizations plays crucial role, such as 3D animation, gaming, virtual globes (Autodesk 3ds Max, Google Earth) that also focus on efficient interaction with 3D environments? In these domains high efficiency is usually achieved by the use of computer graphics algorithms such as surface simplification depending on current view, distance to objects, and efficient caching of the aggregated representation of object meshes. We investigate how these algorithms can be re-used in the context of interactive hydrodynamic modeling without significant changes to the model code and allowing model operation on both multi-core CPU personal computers and high-performance computer clusters.

INTRODUCTION

The goal of this work is to develop ***an interactive large scale water simulation software*** that works on parallel architectures and is suitable for both desktop (multi-core CPU and GPU) and HPC configurations. The software will be based on existing state-of-the-art hydrodynamic simulation code D-Flow FM Kernkamp *et al.* [1]. The model code was recently parallelized to utilize distributed architectures using MPI (van der Pijl *et al.* [2]). However, the deployment and efficient use of model in parallel model still requires significant efforts for setup and use, for example, model domain should be split into a number of sub-domains to run them on different cluster nodes, boundary conditions for every sub-domain have to be defined correctly, and so on. Additionally to that, the model provides a very limited visualization possibilities for parallel configurations. The aim of this work is to enable both visualization and interaction for the parallel version of D-Flow FM model as it was shown in Donchyts *et al.*, [3]. This presentation will be based on preliminary results of the research and demonstrate a prototype software that allows simultaneous visualization of the parallelized version of the D-Flow FM model results, it will also allow interaction with the model variables (like bathymetry, water level) in real-time during the run on parallel hardware.

During the work we also plan to test a number of existing software components that provide scientific visualization functionality and parallel and distributed rendering components, such as VTK (Schroeder, [4]) and ParaView (Cedilnik, [5]). It worth mentioning that the effective use of resources available in modern computer hardware was mentioned as one of the top research problems in scientific visualization in Johnson, [6]. Recent advances in processor technology (CPU and GPU) are not yet fully utilized in the general purpose scientific visualization libraries like VTK, used as the main library in ParaView. However, there are efforts to extend it for an effective use in parallel architectures (Moreland, [7]). Furthermore, popularity of the High-Performance Computing (HPC) in the last decade brings additional challenges, like development of the software infrastructures that shield developers of the algorithm from the complexities and enable efficient use of available resources (Childs, [8]).

PARALLELIZED INTERACTIVE WATER SIMULATION MODEL

In parallel architectures that involve visualization and interaction a communication between different components has to be done efficiently. We aim to use modern message-based client / server communication library called ØMQ (Hintjens, [9]). The library provides different transport protocols for communication in both distributed (TCP) and single computer (IPC / inter-process or ITC / inter-thread) configurations.

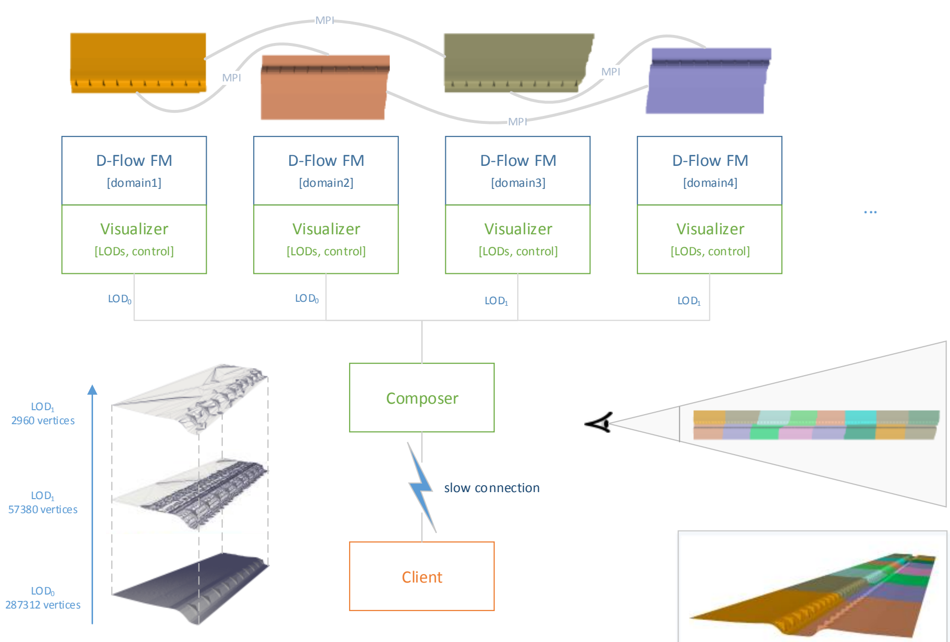

Figure 1. Example of a convergence curve with oscillations

To achieve optimal interaction and rendering performance it is intended to use level of detail algorithms (LOD) suitable for the visualization of large terrains (Losasso, [10], Hoppe, [11]). For the visualization and interaction with water surface we will investigate applicability of the algorithms discussed in Bruneton, [12].

The choice of the LOD for different model domains is based on current 3D view on the client (view frustum in computer graphics) and changes automatically during 3D scene navigation, see Figure 1. Depending on current LOD, a corresponding coarsened mesh will be selected on the simulation nodes, thus limiting amount of data that needs to be transferred to the client using slow connection. Additionally to the surface (water or bathymetry) meshes the software will allow visualization of the scalar or vector variables. The final scene will be rendered on the client.

REFERENCES

[1] H. W. J. Kernkamp, A. Dam, G. S. Stelling, and E. D. Goede, “Efficient scheme for the shallow water equations on unstructured grids with application to the Continental Shelf,” *Ocean Dynamics*, vol. 61. pp. 1175–1188, 2011.

[2] A. van der Pijl, S.P., Kernkamp, H. W. J., Borsboom, M., Genseberger, M., van Dam, “Parallelization of a shallow-water solver on unstructured meshes (D-Flow FM).,” in *Proceedings of the 22nd International Conference on Domain Decomposition Methods.*

[3] G. Donchyts, F. Baart, and H. A. van Dam, “The Joy of Interactive Modelling,” in *European Geoscience Union*, 2013.

[4] W. J. Schroeder and K. M. Martin, “The visualization toolkit,” in *Visualization Handbook*, 2005, pp. 593–614.

[5] A. Cedilnik, B. Geveci, and K. Moreland, “Remote large data visualization in the paraview framework,” *Proc. 6th …*, 2006.

[6] C. Johnson, “Top scientific visualization research problems,” *IEEE Comput. Graph. Appl.*, vol. 24, no. 4, pp. 13–17, Jul. 2004.

[7] K. Moreland and U. Ayachit, “Dax toolkit: A proposed framework for data analysis and visualization at extreme scale,” *... Vis. …*, 2011.

[8] H. Childs, B. Geveci, W. Schroeder, J. Meredith, K. Moreland, C. Sewell, T. Kuhlen, and E. W. Bethel, “Research Challenges for Visualization Software,” *Computer (Long. Beach. Calif).*, vol. 46, no. 5, pp. 34–42, May 2013.

[9] P. Hintjens, *ZeroMQ: Messaging for Many Applications*. 2013.

[10] F. Losasso, J. O. Talton, N. Kwatra, and R. Fedkiw, “Two-Way Coupled SPH and Particle Level Set Fluid Simulation,” *Vis. Comput. Graph. IEEE Trans.*, vol. 14, no. 4, pp. 797–804, Jul. 2008.

[11] H. Hoppe, “Smooth view-dependent level-of-detail control and its application to terrain rendering,” *Vis. Proc.*, 1998.

[12] E. Bruneton, “Real- time Realistic Ocean Lighting using Seamless Transitions from Geometry to BRDF,” *Comput. Graph. Forum*, 2010.