Parallelizing flood model for Linux clusters with MPI

V. D. Tran[1], L. Hluchy[1], D. Froehlich[2], and W. Castaings[2]

[1] Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 845 07 Bratislava, Slovakia
viet.ui@savba.sk
[2] 303 Frenchmans Bluff Drive, Cary, North Carolina 27513-5662, USA
Froehlich@pbworld.com
[3] LMC-IMAG, Domaine Universitaire BP 53
38041 Grenoble Cedex 9, France
william.castaings@inrialpes.fr

Abstract. This paper focuses on parallelization process of DaveF, a new two-dimensional depth-averaged flow and sediment transport model that allows breach development and the resulting flood wave to be simulated simultaneously. Problems encountered during parallelization and techniques used to solve them are described. The experimental results with different input data on different machines are also included.

1 Introduction

Over the past few years, floods have caused widespread damages throughout the world. Most of the continents were heavily threatened. Therefore, modeling and simulation of floods in order to forecast and to make necessary prevention is very important. The kernel of flood simulation is a numerical modeling, which requires an appropriate physical model and robust numerical schemes for a good representation of reality.

Simulating river floods is an extremely computation-intensive undertaking. Several days of CPU-time may be needed to simulate floods along large sections of rivers. For critical situations, e.g. when an advancing flood is simulated in order to predict which areas will be threatened so that necessary prevention measures can be implemented in time, long computation times are unacceptable. Therefore, using HPCN (High Performance Computing and Networking) platforms to reduce the computational time of flood simulation is imperative. The HPCN versions of hydraulic models not only reduce computation times but also allow simulation of large scale problems, and consequently provide more reliable results.

Generally, the process of HPCN implementation consists of the following steps:

- Analyzing computational approaches used in the models: the methods of discretization (finite elements, finite differences, finite volumes), the algorithms (Newton iteration, frontal solution methods)