Project 4: Shallow Water Simulation

Motivation

'Tsunami' is becoming a common household word after two major disasters in the far-east within resent years. Tsunami warning systems are as a consequence becoming increasingly interesting and at the foundation of any such warning system is a shallow water simulation. In the shallow water model, any sea of water is modeled only in two dimensions, with the height of the water at any point simply a parameter of a 2d surface point. To model this we need only the height, and the momentum in the x and y planes.

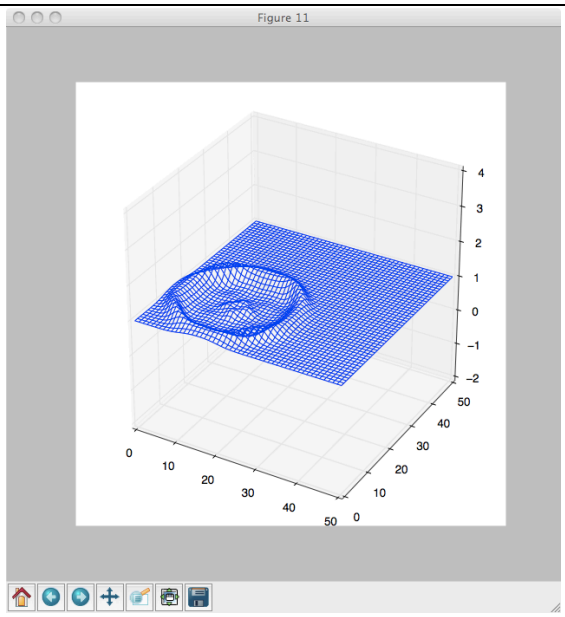

Figure 1 Example of shallow water visualization.

You can read more on the model here: http://www.mathworks.com/moler/exm/chapters/water.pdf

Programming Task

The solution should be parallelized implemented using PuPyMPI, and may be based on the sequential version available. The code should be run on from one through at least 256 CPUs. A sequence 1,2,4,8,16, 32, 64, 128, and 256 is enough. You are allowed to increase the workload linearly with the number of CPUs in order to obtain good performance, i.e. every time you double the number of CPUs you also double the number of modeled grid points.

Report

Your report, which should be submitted through Absalon, should be no longer than 2 pages in total,. You should explain how you have transformed the sequential code and how this has influenced your performance. You should provide experiment results with the 3 provided benchmark sizes, perhaps with problem scaling as described above, where you compare the performance of the original version to your parallel solution. The results should be presented as an easy the read graph, which includes the absolute and relative before and after transformation of the code.

You should run the jobs on Manjula

The sequential runtimes are:

N	Run1	Run2	Run3	Average
1000	5,55	5,57	5,59	5,57
2000	19,35	19,19	19,30	19,28
4000	89,38	89,65	89,65	89,56