

Near Field High Resolution Numerical Simulation of Tohoku 2011 Tsunami on Multi-core System

Shenyi Song

Graduate University of Chinese Academy of Science
Computer Network Information Center, Chinese Academy
of Science
Beijing, China

Aiyu Zhu

Institute of Geophysics, China Earthquake Administration
Beijing, China

Shuxia Zhang

Supercomputing Institute, University of Minnesota
Minneapolis, USA

Liang Zheng

Graduate University of Chinese Academy of Science
Beijing, China

David A. Yuen

Supercomputing Institute & Depart. Of Geology and
Geophysics, University of Minnesota
Minneapolis, USA

Zhonghua Lu

Computer Network Information Center, Chinese Academy
of Science
Beijing, China

Abstract—We use Finite Volume Method based on Shallow-water equation to simulate the Tohoku 2011 tsunami and parallelize the code using OpenMP. With the parallelization, we achieved over 80% of the potential speed-up on multi-core system. Using adaptive mesh refinement (AMR) to obtain high resolution result in important region, and the highest resolution near Sendai airport and Fukushima nuclear power plants is about 20 meters.

Keywords—Shallow-water equation; OpenMP; Adaptive Mesh Refinement

I. INTRODUCTION

On March 11 at 2:46pm JST, a massive 9.0-magnitude earthquake occurred at 38.322° N, 142.369° E, near the northeastern coast of Japan, creating extremely destructive tsunami waves which hit Japan just minutes after the earthquake. Some important facilities in Japan were damaged by the earthquake and tsunami.

We use GeoCLAW to calculate the tsunami. GeoCLAW is an open-source code for modeling tsunami propagation and inundation. For high resolution simulation, we parallelized GeoCLAW using OpenMP and achieved about 5X speedup on multi-core system. AMR is also used to obtain high resolution result in specified region.

II. SHALLOW-WATER EQUATIONS

Shallow-water equations are often used to modeling the tsunami. They are a set of hyperbolic partial differential equations that describe the flow below a pressure surface in a fluid. Shallow-water equations are depth-averaged 2D equations. Although tsunami waves have 3D flow variation, many people use 2D governing equations to solve large-scale problems. Shallow-water equations are as follows:

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) &= 0 \\ \frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}(hu^2 + \frac{1}{2}gh^2) + \frac{\partial}{\partial y}(huv) &= -gh \frac{\partial b}{\partial x} + S_{fx} \quad (1) \\ \frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}(hv^2 + \frac{1}{2}gh^2) &= -gh \frac{\partial b}{\partial y} + S_{fy} \end{aligned}$$

Where $h(x,y,t)$ is the depth of water, $u(x,y)$ and $v(x,y)$ are the depth-averaged velocities in the x and y directions. S_{fx} and S_{fy} are friction terms. Here we employ the commonly used empirical Manning formulae.

$$\begin{aligned} S_{fx} &= n^2 u \sqrt{u^2 + v^2} h^{-4/3} \\ S_{fy} &= n^2 v \sqrt{u^2 + v^2} h^{-4/3} \end{aligned} \quad (2)$$

Where n is the Manning coefficient—an estimated parameter reflecting the roughness of the bed. We have employed our simulations by using the GeoCLAW, an open source research code which uses high-resolution finite volume methods together with adaptive mesh refinement to tackle geophysical flow problems of the nature.[1] In particular, this code has recently been used together with the shallow water equations to model tsunamis, dam breaks, and storm surges.[2][3]

III. OPENMP PARALLELIZATION

GeoCLAW is a serial code, and it costs too much time on high resolution simulation, so we have to parallelize the code. We use Intel Vtune to analyze the computing flow structure of GeoClaw and identify the hotspot, which consumes most of the computing time.

Then the main work is parallelizing this subroutine. We use l8.msi.umn.edu, with dual Quad Core Intel Xeon W5590 3.33GHz, 24GB memory and enabling Hyper-Threading. It's a shared memory architecture.

OpenMP is usually used on shared memory system. And besides OpenMP, we use other method to optimize the code.

When an OpenMP job is running with multiple threads, there are many ways of pinning threads to physical cores. The performance for the same job can be very different from different mapping topology. The environment variable **KMP_AFFINITY** can determine the machine topology and assigns OpenMP threads to the cores based on their physical location in the machine.

The other effect we used is first-touch. It is used on the shared arrays by all threads. Before the calculation in OpenMP loop, we can set up another OpenMP loop and initialize the data in the shared arrays. If there are no parallelization for the initialization loop, the physical location of memory for the shared arrays is on one of the CPU. On which 8 physical cores are hosted, When more than 8 threads are used for calculation, memory congestion occurs as some threads need to access memory across QPI. When we parallelize the initialization loop, each thread will only access its own memory. Using the first touch will increase the performance about 20%.

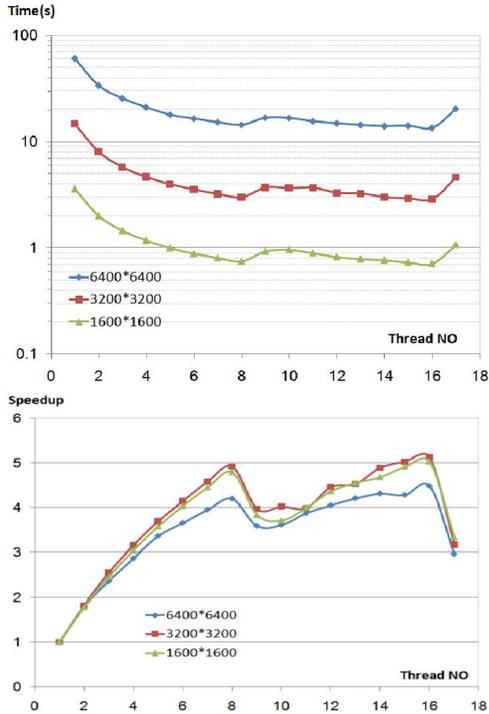


Fig.1 Computing time and speedup on l8.msi.umn.edu of three different size of mesh.

We got about 5X speedup on l8.msi.umn.edu, and the most large speedup is using 16 threads. The CPU on l8.msi.umn.edu is Intel Xeon W5590, which Intel Hyper-Threading Technology is used. So in this CPU, one physical core can simulate two logical threads. In the operation system, we can also see 16 logical cores. On l8.msi.umn.edu, the performance of 16 threads is a little higher than 8 threads.

IV. SIMULATION OF TOHOKU TSUNAMI

A. Earthquake and Tsunami Source

The initial condition of the problem is determined by the displacement or velocity field from the earthquake under sea floor. We used GPS data for constrain condition and calculate the elastic displacement instead of the Okada 1986 model, and get the vertical sea floor uplifting. The earthquake model data is added into the computation region in the first 30 seconds.

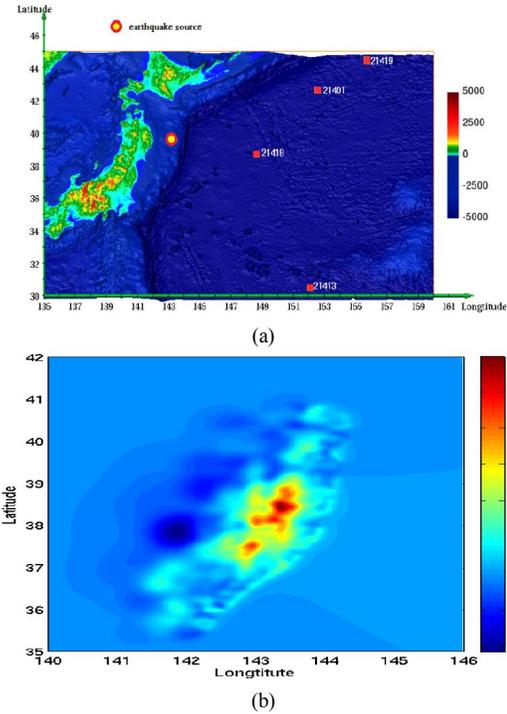
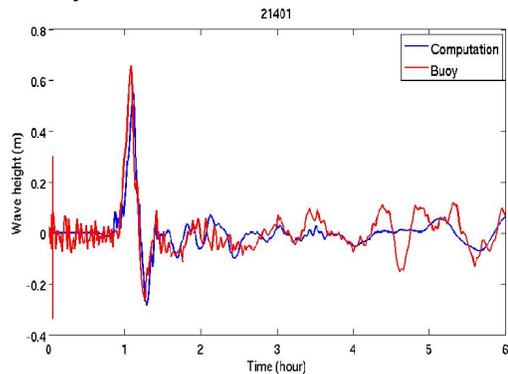


Fig.2 (a) The study region (135o E to 160o E, 30o N to 45o N) with a bathymetric map surrounding Japan. (b) We mark some DART buoys. And the GPS data we used as source

The best way to understand tsunami propagation is observing tsunami signals recorded by offshore buoys or by stations located close to the coasts. The 2011 Japan event was recorded by several stations. In this section we compare our results with the tsunami signal recorded by offshore buoys and coastal stations in the near field.



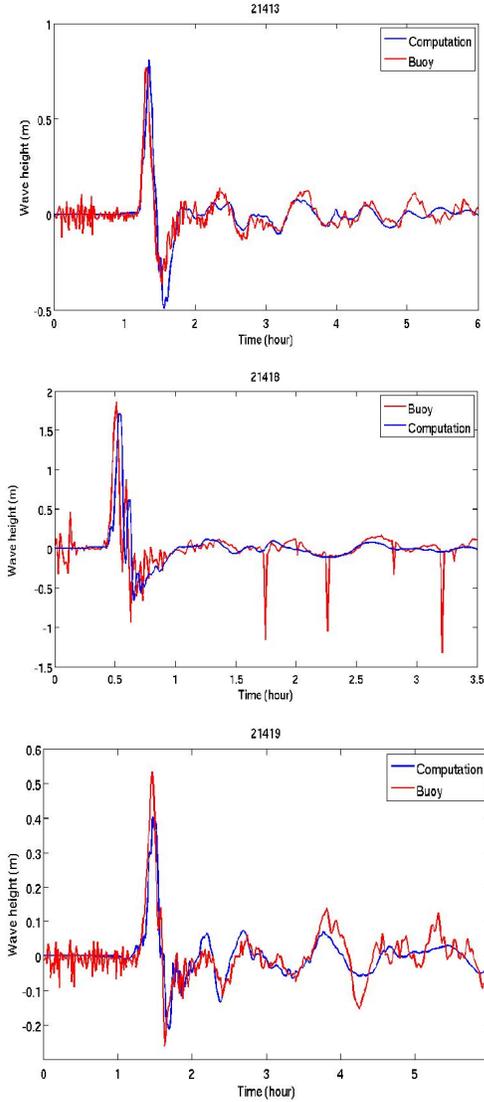
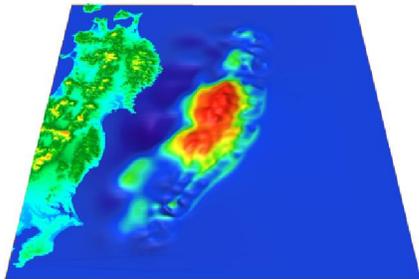
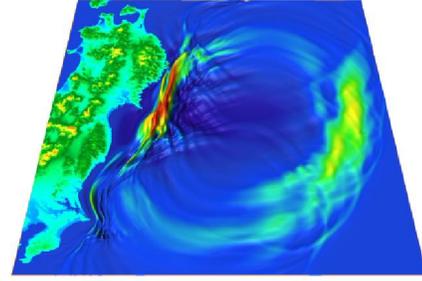


Fig.3 Comparison of the simulated wave height and arrival time with the observed at the Buoy stations. (the red point marked in Figure 2(a))

Figure 3 compares the simulation result and the observed record of the DART buoys. The results match very well to the observed records at both the wave height and arrival time.



(a)



(b)

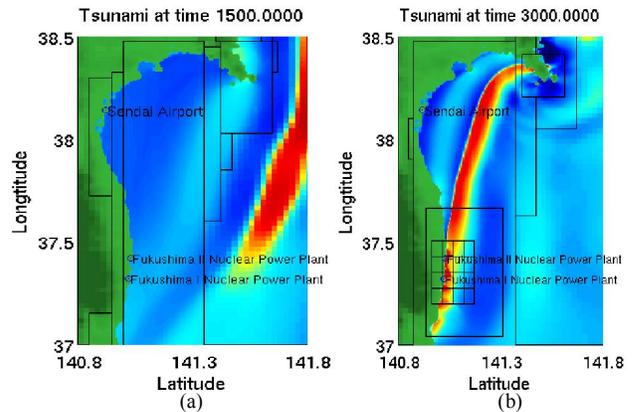
Fig.4 The wave propagation at (a) 1 minute, (b) 15 minutes after the earthquake.

The simulation results shown in Figure 4 are obtained with one-level fine grid mesh of 1600x1600. Although they match well with the observation at the buoy stations in the ocean, we can not use one level mesh to simulate the inundation of tsunami waves over Japan islands because the computational cost would be too high. One better way is use higher density in the important region such as the earthquake source and the seaside.

B. Adaptive Mesh Refinement

The use of Adaptive Mesh Refinement (AMR) becomes essential for the simulations on large-scale problems, especially on hyperbolic PDE problem.

AMR can be set up in GeoCLAW[4], and the regions can be configured before the computation starts, so the density near the seaside can be higher automatically. We set up two important regions, Sendai Airport and Fukushima Nuclear Power Plant. We use 4 level grids. The resolution of the first level is about 2.5km, and the refinement ratios at each level are 4, 4, 8, so the resolution of the finest grid is about 20m.(Figure 5)



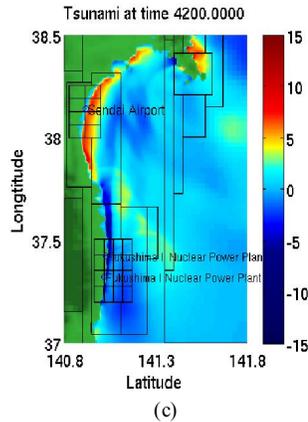


Fig.5 Use higher resolution near the seaside (600m), and when the tsunami comes near the seaside, uses the highest resolution (about 20m) near the beach of Sendai Airport and Fukushima Nuclear Power Plant.

V. DISCUSSION AND CONCLUSION

In this paper, we have showcased an success of applying well known knowledge and available technologies in shared-memory computing to an application scaling and speeding up the simulation with GeoClaw on multi-core system for high resolution simulation of Tohoku 2011 Tsunami Waves, starting from parallelizing the sequential code GeoCLAW - how we identified and widened the bottlenecks by parallelizing the most time consuming subroutine a and how we further improved the performance by setting optimal runtime environment variables. Over 75% of the potential speedup is achieved on l8.msi.umn.edu. The results of GPS source model has a generally agreement with the observed data, and paralleled GeoCLAW scales well. The parallel computing can accelerate computing and do higher resolution simulation.

In the future, we will try to do more parallelization on GeoCLAW and develop the GPU version. We can use new version to simulate world-wide shallow water wave propagation.

ACKNOWLEDGMENTS

We thank the Erik Sevre and Dave George for discussions. This research has been supported by the CMG program of National Science Foundation.

REFERENCE

- [1] Leveque, R. J., George, D. L. and M. J. Berger, Tsunami modeling with adaptively refined finite volume methods, Acta Numerica. Cambridge Univ. Press[J], 2011. PP123
- [2] R. J. LeVeque and D.L. George, High-Resolution Finite Volume Methods for the Shallow Water Equations with Topography and Dry-States. In P. L.
- [3] Leveque, R. J.; Berger, M. J.; Mandli, K. T., The GeoClaw Software for Geophysical Flows, <http://adsabs.harvard.edu/abs/2010AGUFMDI51A1855L>
- [4] Liu, C. Synolakis, and H. Yeh, editors, Advanced Numerical Models for Simulating Tsunami Waves and Runup, World Scientific [J], 2008. PP43-73.