

# Parallelization of GeoClaw Code for Modeling Geophysical Flows with Adaptive Mesh Refinement on Many-core Systems

Shuxia Zhang<sup>1</sup>

1. Supercomputing Institute  
University of Minnesota  
Minnesota, USA  
szhang@msi.umn.edu

Shenyi Song<sup>2</sup>

2. Computer Network Information Center  
China Academy of Science  
Beijing, China  
songsy@sccas.cn

Aiyu Zhu<sup>3</sup>

3. Institute of Geophysics  
China Earthquake Administration  
Beijing, China  
aiyuzhu@yahoo.com.cn

David A. Yuen<sup>4</sup>

4. Supercomputing Institute &  
Dept. of Geology and Geophysics  
University of Minnesota  
Minnesota, USA  
daveyuen@gmail.com

**Abstract**— We have parallelized GeoClaw using OpenMP to meet the urgent need of simulating tsunami waves at near-shore from Tohoku 2011 at a grid distance around 100 meters because of no available parallel version yet by then. It took us some efforts to obtain the correct results and speed-up in the computation. With this OpenMP parallelization, we achieved over 75% of the potential speed-up on an eight core Dell Precision T7500 workstation. Recently we obtained an unreleased OpenMP version of GeoClaw from David George, who developed GeoClaw as part of his Ph.D thesis. In this paper, we will show the tactics used for merging the two OpenMP codes and for speeding up the calculation with adaptive mesh refinement (AMR). We will also show the simulated inundation of the Tohoku 2011 Tsunami waves onto the Sendai airport and Fukushima Nuclear Power Plants, over which the finest grid distance of 20 meters is achieved through 4-level AMR. Our eventual goal of this work is to scale GeoClaw over at least several tens of cores so that it can meet the urgent need for solving real-world geophysical flow problems on many-core systems, which we believe is the most efficient and feasible means for AMR in the near future.

**Keywords**—GeoClaw; OpenMP; Adaptive Mesh Refinement; Inundation of tsunami waves; Real-world Geophysical Flows.

## I. INTRODUCTION

GeoClaw is a state-of-the-art finite-volume code and freely available open-source software for modeling tsunami propagation and inundation, and other flooding problems using adaptive finite volume methods [1]. It is a very user friendly package and has been used for high-resolution tsunami simulations [2], dam breaks, storm surges and debris flows, and other geophysical applications [3] [4]. But the calculation with GeoClaw was done in a sequential manner [5] when we started deploying GeoClaw in the beginning of February of 2011. We did not pay attention to its performance until late March, 2011 when we started the

simulations of the destructive 77-foot tsunami [6] with grid distance around 100 meters. With a series of tactics employed, we were able to achieve over 75% of the potential speed-up with GeoClaw for a test case with 6400x6400 grid mesh on a Dell Precision T7500 workstation at University of Minnesota, Supercomputing Institute (UMSI). The tactics employed include the use of profiling tool for hotspot analysis, OpenMP parallelization as well as the use of hyper-threading and control of thread-affinity [7].

One thing we would like to point out here is that we used a multiple-stage working strategy for parallelizing GeoClaw with OpenMP for the objectives to meet the urgent need and get GeoClaw scale well on large many-core systems for a long run. In the first stage we parallelized GeoClaw on one-level grid, which allowed us to present the simulation of the near field characteristics of Tohoku 2011 Tsunami Waves at the research exhibition hosted by MSI [8]. The next stage of our original plan was to parallelize GeoClaw with the multiple-level AMR enabled.

Meeting with Dr. David George, who is developer of GeoClaw, in Minneapolis on April 15, 2011 changed our working plan. He told us that they have been working on parallelizing GeoClaw at University of Washington although they did not list that in the GeoClaw webpage. Later he gave us an unreleased version of their OpenMP development named UWA\_GeoClaw in this paper to distinguish it from the version developed at University of Minnesota Supercomputing Institute, named as UMSI\_GeoClaw simply for the convenience of description. We are very grateful to David George for allowing us to use UWA\_GeoClaw. We found that a lot of the work that we planned to do for AMR has been done in UWA\_GeoClaw. But what we have implemented in UMSI\_GeoClaw has not carried out in

UWA\_GeoClaw. Merging the two into one code would provide the potential for a better performance than either of them due to their complementary characteristics and hence provide a valuable means for geoscientists to solve challenging geophysical flow problems on many-core systems.

## II. NUMERICAL METHODS AND ADAPTIVE MESH REFINEMENT

GeoClaw solves the nonlinear free-surface shallow water equations subject to the hyperbolic conservation laws using finite-volume wave-propagation algorithms developed by Randall LeVeque[3], which use Riemann solvers for accurately handling transoceanic propagation as well as inundation of tsunami waves in the presence of realistic bottom bathymetry or topographic feature [9]. This is accomplished by using adaptive mesh refinement. The first level consists of equal-spaced coarse grids covering the entire domain. Rectangular cartesian sub-grids of higher level refinement are inserted to track moving waves and inundation at the shoreline. Up to six levels of sub-grid refinement may be used. At any given time in the calculation, a particular level of refinement may have numerous disjoint grids associated with it.

The GeoClaw code consists of a series of standard routines under the amrclaw directory, which are for and/or associated with adaptive mesh refinement and many specialized routines under the geoclaw directory for solving geophysical flow problems[5]. Python scripts are provided in GeoClaw for setting inputs and parameters controlling the simulation. For example, user can specify the refinement ratios between particular levels - allowing a large degree of refinement even for a small number of levels. The user can also impose a grid refinement over a specific region over a particular time range, which is very useful for forecasting inundations at extremely high resolution.

## III. PARALLELIZATION OF GEOCLAW WITH OPENMP

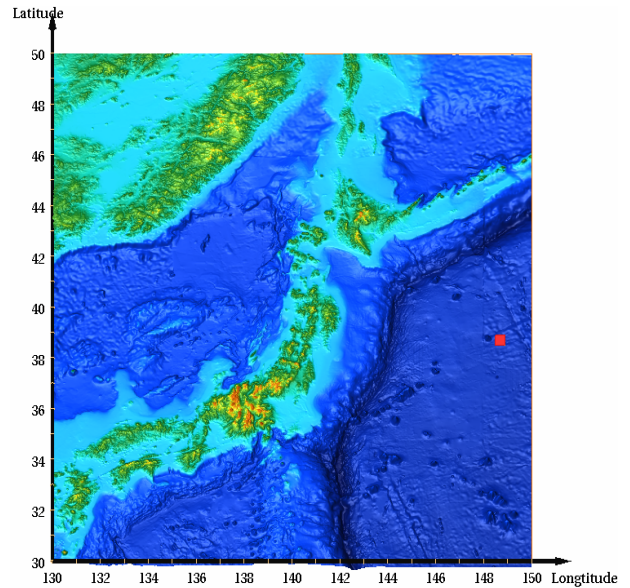
### A. Model setup

Figure 1 shows the studied region over which GeoClaw has been employed for the purposes of both evaluating performance described in this section and demonstrating a success of real-world application, which will be described in section IV. Two major inputs are needed for running GeoClaw: (1) a model of the impulsive disturbance that displaces the water or triggers the tsunami and (2) bathymetric data for the region of study interest. The impulsive disturbance is based upon the focal model proposed by Chen Ji at UCSB [10] for the faulting mechanics associated with the magnitude 9.0 earthquake, which is translated into an impulsive disturbance or seafloor deformation using the Okada's elastic dislocation model [11] because the UCSB focal model produces a better fit to

the observations compared to other focal models [12]. We downloaded the topographic data with 1 minute resolution from database of National Geophysical DATA Center [13].

It is essential to maintain the computing results accurate while try to speed and scale up the calculation. We constantly monitor the difference between the results compiled with intel **ifort** with `-O3` optimization and those compiled with GNU **gfortran** with `-O0` after the simulation runs 300 time steps. Basically, no difference can be observed by naked eyes based on the graphic comparison with Matlab.

A timer is set inside the Tick routine, reporting the time consumed for each time step. The simulation of each test case ran for ten time steps. The average of the last five steps is reported as the wall time per *time* step or  $T_n$ , where n is the number of threads used. Speed-up is measured by the ratio of  $T_1$  to  $T_n$  with  $T_1$  being the wall time per time step obtained with 1 thread. Please note all the calculations shown in this paper are done on a Dell Precision T7500 workstation, which has eight physical cores and supports hyper-threading. Intel® compiler and VTune™ Amplifier XE 2011 (Vtune) [14] are available on this machine, which provide a useful tool for us to identify quickly the hotspot and analyze the performance under the run-time environment of various interests.



**Figure 1. The region being studied (Longitude:130° to 150°, Latitude: 30° to 50° in Latitude) with a bathymetric map surrounding Japanese Islands.** The red square near the right edge marks the DART buoy 21418. The observed wave heights and arrival time at a few DART buoy stations including 21418 are used for comparing numerical simulations [12].

### B. Complementary characteristics of UMSI\_GeoClaw and UWA\_GeoClaw

Figure 2 and Figure 3 show the profiling results of running GeoClaw for test cases with one-level grid-mesh and

three levels of grid-meshes, respectively. In the former case, the STEP2 routine is the hot spot. It consumes almost 90% of the total CPU time. In contrast, the same routine consumes 60% of the total CPU time for the 3-levels of grid, whose major inputs are shown in Table 1. The significant amount of difference is attributed to the additional CPU time spending on the operations for and associated with the adaptive mesh refinement. Obviously, parallelization on STEP2 routine alone is not sufficient for achieving a big number of speed-up when AMR is being used intensively. For the test case mentioned above, it cannot exceed a factor of 2.5. Further parallelization is needed to scale up the calculation for fully using the intended functionality of adaptive mesh refinement.

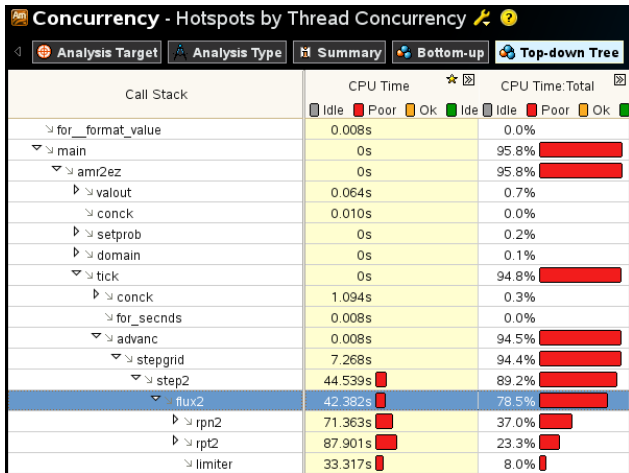


Figure 2. Profiling results of running sequential Geoclaw for one-level of grid-mesh 1200x1200 covering the studied region shown in Figure 1.

Table 1: Important parameters set in setrun.py

Control Variables	Values
clawdata.xlower	140
clawdata.xupper	146
clawdata.ylower	35
clawdata.yupper	41
clawdata.mx	500
clawdata.my	500
mxnest	3
clawdata.inratx	[2,4,4]
clawdata.inraty	[2,4,4]
clawdata.inratt	[2,4,4]
clawdata.cfl_desired	0.75
geodata.regions.append*	

\*A sub-grid with a refinement ratio of 2 is enforced over the region covering the epicenter of 9.0 earthquake for better resolving impulsive disturbance, by assigning Array [2, 2, 0., 50., 140.5, 144.5, 36.5, 39.5] to geodata.regions.append.

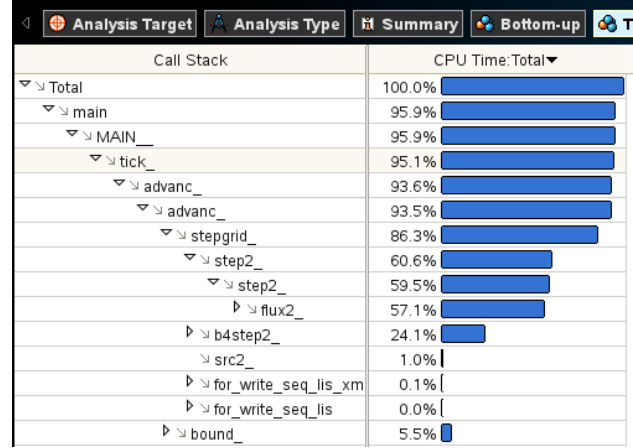


Figure 3. Profiling results of running Geoclaw with one thread for three-levels of grid meshes, whose control parameters are shown in Table 1.

The good news is that a lot of the work that we planned to do for AMR has been done in UWA\_GeoClaw [15], where the routines for and associated with AMR have been parallelized with OpenMP for individual grid patches at a given AMR level. (A patch is a rectangular Cartesian grid). Some of the routines in geoclaw directory are also parallelized although the routine **step2** is sequential.

Very impressive speed-up is achieved with UWA\_GeoClaw for the test case shown in Figure 3 although its one thread job runs 10% slower compared to the serial code. For example, a speed-up of 3.7 is obtained for a run with 4 threads (Figure 4). The speed-up is 7 for a run with 8 threads.

Before taking action to further parallelize the code, it is critical to obtain a clear picture about which of the existing parallel regions accounts for how much of the speed up achieved with the UWA\_GeoClaw code. To address this question, we turned off the parallelization for the routines in the geoclaw directory, reran the test and compared the results with those obtained with the parallelization on. We found the performance difference is negligible, indicating the parallelization on the routines in the geoclaw directory does not contribute much, where routine **step2** is not parallelized in UWA\_GeoClaw.

Our experience with UMSI\_GeoClaw, in which parallelized routine **step2** yields significant speed-up, strongly suggests the use of the parallelized routinestep2 can enhance the performance of UWA\_GeoClaw. Hence, merging the two into one code should yield better performance than either of them due to their complementary characteristics.

### C. Merging of UMSI\_GeoClaw with UWA\_GeoClaw through nested parallelism

We have merged UMSI\_GeoClaw with UWA\_GeoClaw through a nested parallelism. Concretely, we combined the amrclaw directory from UWA\_GeoClaw and the geoclaw directory from UMSI\_GeoClaw. Three run-time variables, OMP\_NUM\_THREADS, NUM\_THREADS, OMP\_NESTED are essential for enabling the nested parallel calculation. Intel compiler (11.1) is used, which supports the nested parallelism of OpenMP.

The first level parallel region starts in routine **advanc** for AMR operations in the amrclaw directory, subject to the threads specified by OMP\_NUM\_THREADS. The second or the nested parallel region is in step2 routine for wave sweeps in the geoclaw directory, subject to the value of NUM\_THREADS.

Figure 4 shows the profiling results for the same test case with 3-levels AMR as shown in Figure 3, but using 4 threads, where nested parallel is not on, i.e. step2 itself runs sequentially.

As comparison, Figure 5 shows the same test as Figure 4, but the nested parallel is enabled with 2 threads dedicated to the computing in routine **step2**.

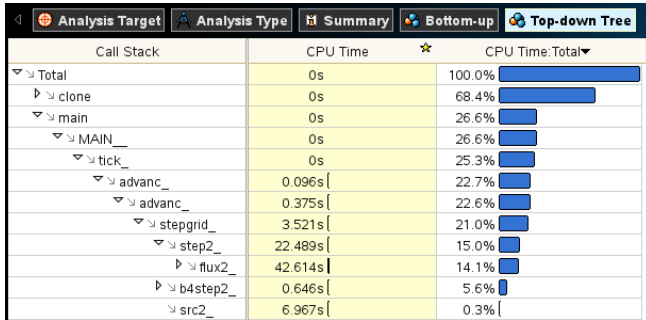


Figure 4. The profiling results for the test case as shown in Figure 3 using 4 threads for 3-levels AMR, and sequential step2 routine, or without enabling nested parallel.

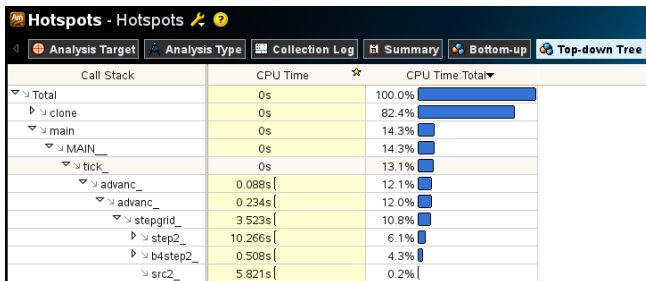


Figure 5. The profiling results for the same test as shown in Figure 4, except 2 threads dedicated to the computing in step2 routine.

One can observe that compared to Figure 4, the nested parallel reduces the wall-clock time in routine steps from 22.5s to 10.3s (Figure 5). Consequently the speed-up is increased to 6.5 from 3.7, which is achieved by applying 4 threads for AMR operations relative to the results with 1 thread shown in Figure 3.

Figure 6 shows an interesting result, or additional speed-up attributed to the use of hyper threading, which is known as Intel® HT Technology, enabling multiple threads to run on each core and using processor resources more efficiently[14]. Compared to the results shown in Figure 4, where all the 8 physical cores are used for the calculation, using 16 threads does enhance the parallel computing to 88.4% (Figure 6) from 82.4% (Figure 5).

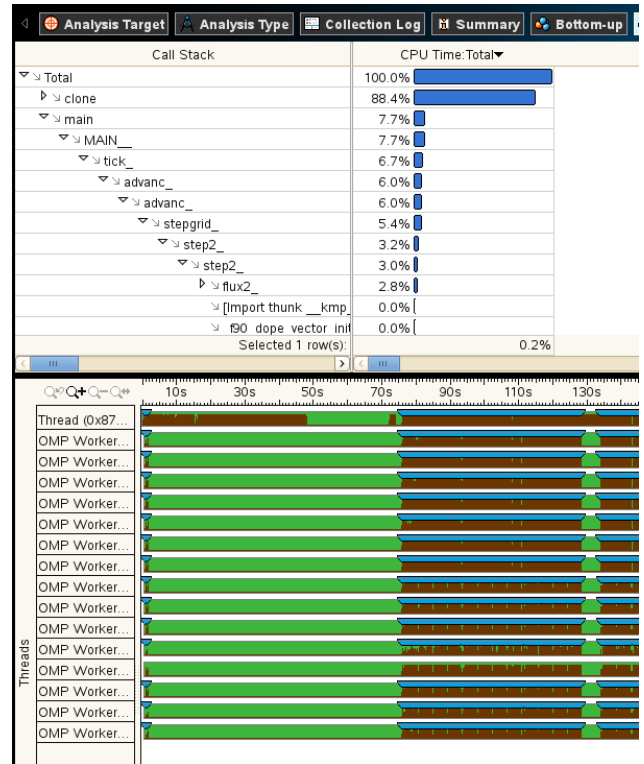


Figure 6. The profiling results for the test case as shown in Figure 3 using 8 threads for AMR, and 2 nested threads for step2 routine.

## IV. INUNDATION SIMULATION OF TSUNAMI WAVES FROM THE TOHOKU 2011 EARTHQUAKE

As a proof of combining the well-known knowledge with OpenMP parallelization and SMP computing technologies, we would show some simulation results obtained by applying the newest version of OpenMP GeoClaw to modeling the Tohoku 2011 tsunami waves and their inundation to Sendai airport and nuclear plants, subject to the focal mechanism model (Figure 7) proposed by Guangfu Shao, Xiangyu Li, Chen Ji, and Takahiro Maeda [10]. They used the hypocenter of the Japan Meteorological Agency

(Lon.=142.860 deg.; Lat.=38.10 deg.) and the fault plane of strike at198.00 deg. and dip angle of10.00 deg.

the inundation of tsunami waves over Japanese islands because the computational cost would be too high.

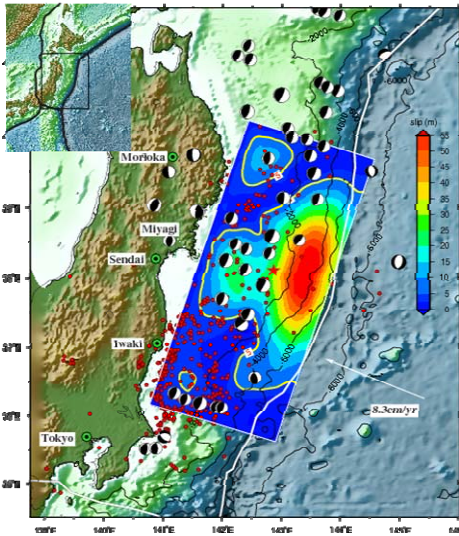


Figure 7. Preliminary result of the Mar 11, 2011, 9.1 Honshu Earthquake, cited from reference [xx].

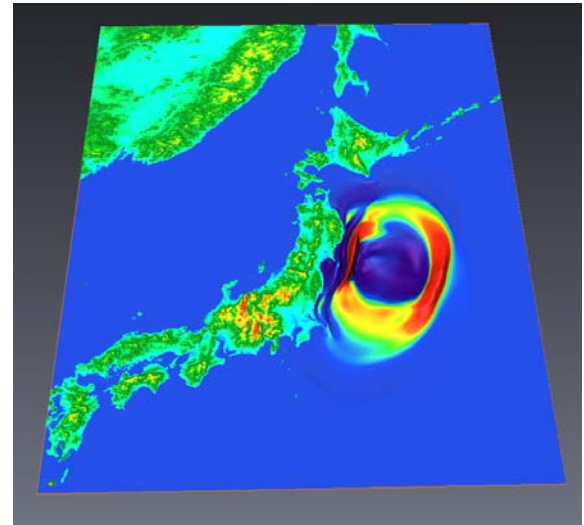


Figure 9. A snapshot of simulated wave propagation at 0.5 hours after the initial impulsive disturbance. The wave heights marked with red and deep blue color are more than 2 m above and less than 1.8m below the sea-level respectively

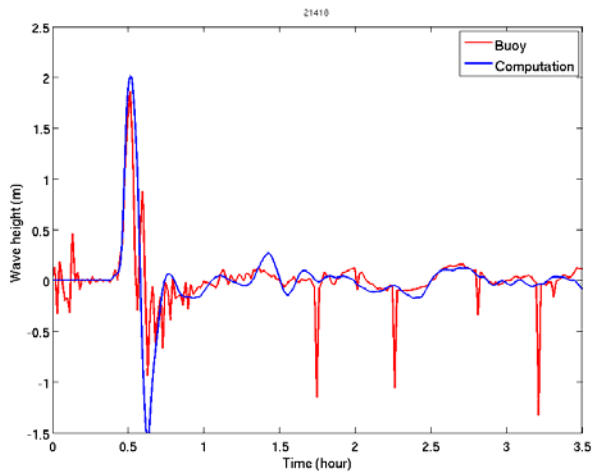


Figure 8. Comparison of the simulated wave height and arrival time with the observed at the Buoy station 21418 (the red square marked in Figure 1).

Figure 8 compares the simulated wave-height as a function time to the observed records at DART buoy 21418 station (the red square marked in Figure 1). We can see the simulation results match very well to the observation at both the wave height and arrival time. Figure 9 shows a snapshot of wave propagation at the time of 0.5 hours after the initial impulsive disturbance.

The simulation results shown in Figure 8 and Figure 9 are obtained with one-level fine grid mesh of 1200x1200. Although they match well with the observations at the buoy stations in the ocean, one cannot use this model to simulate

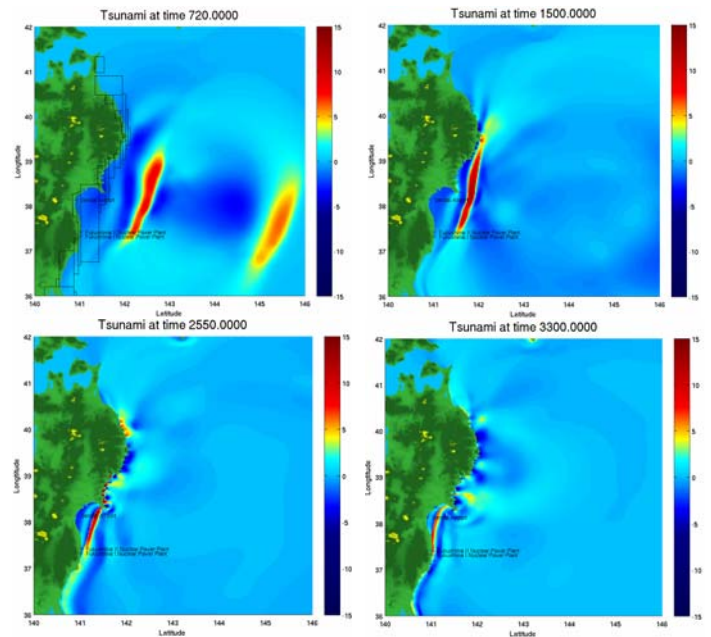


Figure 10. The position of the big tsunami waves at the simulation time of 20 minutes (upper left), 25 minutes (upper right), 42 minutes (lower left) and 55 minutes (lower right). Sendai airport and Fukushima nuclear power plants are marked in these figures.

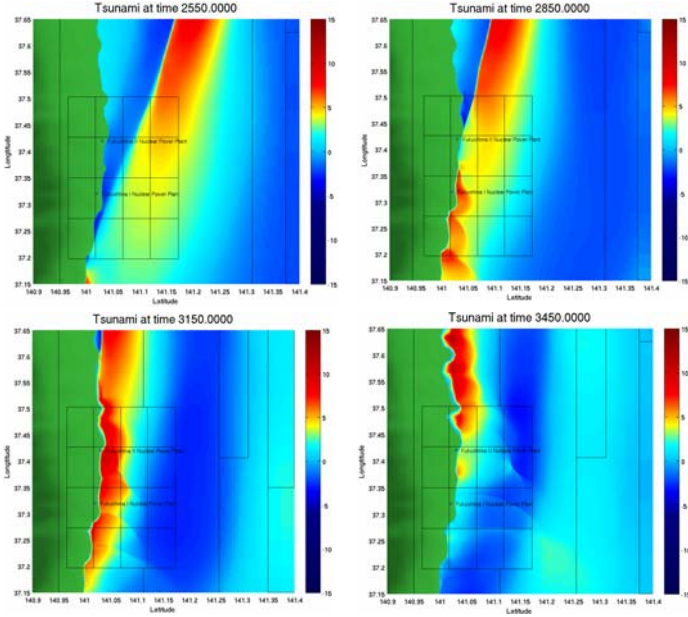


Figure 11. Inundation process of tsunami waves toward to Fukushima nuclear plants at the simulation time of 42.5 minutes (top left), 47.5 minutes (top right), 52.5 minutes (lower left) and 57.7 minutes (lower right). The finest sub-grid distance is around 20 meter around the nuclear plants.

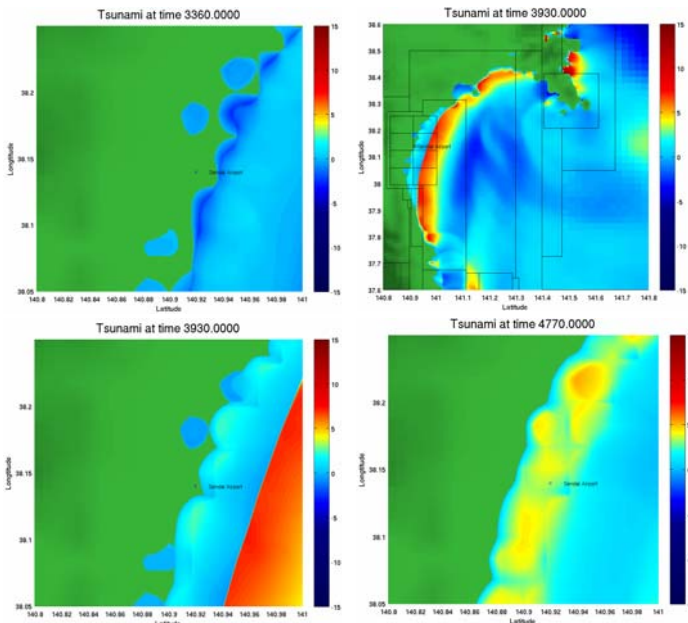


Figure 12. Inundation process of tsunami waves over Sendai airport at simulation time of 56 minutes (top left), 65.5 minutes (upper right and lower left) and 79.5 minutes (top right). The finest sub-grid distance is around 20 meter around the airport. The sub-grid nesting is shown in the upper right panel.

The use of multi-level AMR becomes practically essential for simulating the inundation process of tsunami waves through automatically imposing the sub-grid only at the locations when it is necessary. Even though, the computing cost increases dramatically as more and more levels of sub-grids are imposed.

Figure 10 shows the position of the big tsunami waves at the simulation time of 20 minutes (top left), 25 minutes (top right) when the tsunami waves arrived to Japanese islands, 42 minutes (lower left) when tsunami waves approached to the nuclear power plants and 55 minutes (lower right) or a few minutes before the tsunami waves overwhelmed the Sendai airport. Since the one-minute resolution of the topographic data used, this model could not catch well the actual inundation process of the tsunami waves. Nevertheless, this model does predict reasonably well about the arrival time of the tsunami waves [16],[17]. Please note the simulation of a 40-minute tsunami travel from its source to the nuclear power plants has taken more than two hours of wall clock time with all the computing power available on the eight-core Dell Precision T7500.

Figure 11 shows the simulated inundation of tsunami waves over land approaching Fukushima nuclear plants. The 5<sup>th</sup> level of sub-grid patch covering the nuclear plant area with a grid distance about 20 meters is enforced so that the inundation process can be well resolved in the simulation.

What shown in Figure 12 is the simulated inundation of tsunami waves over Sendai airport, over which the 4<sup>th</sup> level of sub-grid patch with a grid distance about 20 meters is also imposed. Our simulation has produced a very vivid picture of the inundation process of tsunami waves over the Sendai airport.

## V. DISCUSSION AND CONCLUSIONS

The central work reported in this paper is the merge of UMSI\_GeoClaw with UWA\_GeoClaw because they both were parallelized with OpenMP, but were developed independently. We have shown that the parallel implementation in UWA\_GeoClaw is very good for speeding up the calculations for multi-level adaptive mesh refinement [15] whereas UMSI\_GeoClaw speeds up well the calculation associated with sweeping wave propagation in the geoclaw directory [7]. We have merged the two together by using routines in the amrclaw directory from UWA\_GeoClaw and the routines in geoclaw directory from UMSI\_GeoClaw. A nested parallelism is used for the merged code, which shows speed-up at both levels of parallel regions. It can also benefit from Intel's hyper-threading technology (see description for Figure 6).

As an application proof, we have showed some results obtained by applying the newest version of OpenMP GeoClaw to modeling the Tohoku 2011 tsunami waves and

the their inudation to the Sendai airport and Fukushima nuclear power plants. Although the simulation results are not intended for comparing to the real observations because of the poor resolution in the topographic representation, this model does predict reasonably well about the arrival time of the tsunami waves. However, it is worthwhile to point out that the simulation of a 40-minute tsunami travel from its source to the nuclear power plants has taken more than two hours of wall clock time with all the computing power available on the eight core Dell Precision T7500 workstation. That indicates the need of using large many core systems to finish a forecasting before the event takes place.

Because of the horrifying 3/11 Earthquake and Tohoku tsunami and the extraordinary amount of data and observations coming from the well-covered GPS, volcano and earthquake monitoring facilities in the surrounding region, research activities related to tsunamis at the tectonically active regions has become intensive. GeoClaw will become an important package for the environmental sciences because it can solve many different problems related to hyperbolic equations [4]. The OpenMP GeoClaw code comes out timely. Its availability will help many researchers run simulations of geophysical flows more efficiently since almost every desktop and Laptops nowadays are made of multi-core architectures.

On the other hand, more and more larger-size many-core systems have become available [18], leading by SGI and Intel. Better use of these large SMP machines poses great challenges for researchers to rethink or redesign application models. That was another motivation for us to use OpenMP to parallelize GeoClaw. Although a lot of researchers have shown outstanding performance with the use of GPU for general purpose computing, it will take some years to become capable of solving real-world AMP applications like GeoClaw. MPI has been available for almost two decades, but it has not shown an efficient way of dealing with AMR in GeoClaw application either. Hence we believe that getting GeoClaw scaling well on many-core systems is the most efficient and feasible means for meeting the urgent application challenges.

#### ACKNOWLEDGMENT

We thank Dave George for giving us their unreleased Open MP version of GeoClaw. Without it we could have not finished this manuscript. We thank Erik O.D Sevre, Dave L. George and Kyle Mandli for stimulating discussions. Participation of Shenyi Song and Aiyu Zhu in this research has been supported by the CMG program and Geophysics programs of the National Science Foundation.

#### REFERENCES

- [1] R. J. LeVeque, M. J. Berger, et. al., Clawpack Software 4.5.1, <http://www.clawpack.org>, downloaded at Feb 1, 2011.
- [2] Leveque, R. J., George, D. L. and M. J. Berger, Tsunami modeling with adaptively refined finite volume methods, Acta Numerica, 123 pages, DOI: 10.1017/S0962492904, Cambridge Univ. Press, 2011.
- [3] 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. Liu, C. Synolakis, and H. Yeh, editors, *Advanced Numerical Models for Simulating Tsunami Waves and Runup*, volume 10 of Advances in Coastal and Ocean Engineering, pages 43-73. World Scientific, 2008.
- [4] Leveque, R. J.; Berger, M. J.; Mandli, K. T., The GeoClaw Software for Geophysical Flows, <http://adsabs.harvard.edu/abs/2010AGUFMD151A1855L>
- [5] GeoClaw - 4.6 documentation, <http://kingkong.amath.washington.edu/clawpack/doc/geoclaw.html>
- [6] "Japan tsunami was at least 23-metres high: report" ABS-CBN Interactive. <http://www.abs-cbnnews.com/global>
- [7] Shuxia Zhang, Aiyu Zhu, Shenyi Song and David Yuen, Use of Many-Core Architectures for High-Resolution Simulation of Tohoku 2011 Tsunami Waves, submitted to Supercomputing Conference, 2011.
- [8] Shenyi Song, Aiyu Zhu, Shuxia Zhang, Erik Sevre, David Yuen, Near field high resolution simulation of Tohoku 2011 Tsunami Waves, to appear at the Research Exhibition hosted by University of Minnesota Supercomputing Institute on April, 25, 2011. <https://www.msi.umn.edu/events/researchexhibition2011.html>
- [9] George, D.L., Augmented Riemann Solvers for the Shallow Water Equations over Variable Topography with Steady States and Inundation, *J. Comp. Phys.*, 227(6):3089-3113, 2008.
- [10] Guangfu Shao, Xiangyu Li, Chen Ji, Takahiro Maeda, Preliminary Result of the Mar 11, 2011 Mw 9.1 Honshu Earthquake [http://www.geol.ucsb.edu/faculty/ji/big\\_earthquakes/2011/03/0311\\_v2/Honshu\\_2.html](http://www.geol.ucsb.edu/faculty/ji/big_earthquakes/2011/03/0311_v2/Honshu_2.html)
- [11] Y. Okada, Surface deformation due to shear and tensile faults in a half-space. *Bull Seism Soc Am* 75:1135–1154, 1985.
- [12] Zhu Aiyu, Song Shengyi, Zheng Liang, Zhang Shuxia, Zhang Dongning and Yuen Dave. Source mechanism analysis of Tohoku 2011 Tsunami Waves, *in preparation*, 2011
- [13] Download Topographic data from NOAA data center: [http://www.ngdc.noaa.gov/mgg/gdas/gd\\_designagrid.html](http://www.ngdc.noaa.gov/mgg/gdas/gd_designagrid.html)
- [14] Intel® Hyper-Threading Technology (Intel® HT Technology), <http://www.intel.com/technology/platform-technology/hyper-threading/index.htm>
- [15] A comparison of patch-based and loop level openMP for AMR in GeoClaw by Berger, Mandli, and May. In preparation, 2011
- [16] Timeline of the Fukushima I nuclear accidents. [http://en.wikipedia.org/wiki/Timeline\\_of\\_the\\_Fukushima\\_I\\_nuclear\\_accidents](http://en.wikipedia.org/wiki/Timeline_of_the_Fukushima_I_nuclear_accidents)
- [17] 2011 Tōhoku earthquake and tsunami [http://en.wikipedia.org/wiki/2011\\_T%C5%8Dhoku\\_earthquake\\_and\\_tsunami](http://en.wikipedia.org/wiki/2011_T%C5%8Dhoku_earthquake_and_tsunami)
- [18] Koronis, a massive global share memory system. <https://www.msi.umn.edu/hardware/koronis/>

