Enabling Very-Large Scale Earthquake Simulations on Parallel Machines

Yifeng Cui[1], Reagan Moore[1], Kim Olsen[2], Amit Chourasia[1], Philip Maechling[4],
Bernard Minster[3], Steven Day[2], Yuanfang Hu[1], Jing Zhu[1], Amitava Majumdar[1],
Thomas Jordan[2]

[1]San Diego Supercomputer Center, 9500 Gilman Drive, La Jolla, CA 92093-0505 USA,
{yfcui, moore, yhu, jzhu, majumdar}@sdsc.edu
[2]San Diego State University, 5500 Campanile Drive, San Diego, CA 92182 USA,
{kolsen, steven.day}@geology.sdsu.edu
[3]Scripps Institution of Oceanography, 9500 Gilman Drive, La Jolla, CA 92024 USA,
jbminster@ucsd.edu
[4]University of Southern California, Southern California Earthquake Center, Los Angeles,
CA 90089 USA, {maechlin, tjordan}@usc.edu

Abstract. The Southern California Earthquake Center initiated a major large-scale earthquake simulation called TeraShake. The simulations propagated seismic waves across a domain of 600x300x80 km at 200 meter resolution, some of the largest and most detailed earthquake simulations of the southern San Andreas fault. The output from a single simulation may be as large as 47 terabytes of data and 400,000 files. The execution of these large simulations requires high levels of expertise and resource coordination. We describe how we performed single-processor optimization of the application, optimization of the I/O handling, and the optimization of execution initialization. We also look at the challenges presented by run-time data archive management and visualization. The improvements made to the application as it was recently scaled up to 40k BlueGene processors have created a community code that can be used by the wider SCEC community to perform large scale earthquake simulations.

Keywords: parallel computing, scalability, earthquake simulation, data management, visualization, TeraShake

1   Introduction

The southern portion of the San Andreas Fault, between Cajon Creek and Bombay Beach in the state of California in the United States has not seen a major event since 1690, and has accumulated a slip deficit of 5-6 meters [13]. The potential for this portion of the fault to rupture in an earthquake as large as magnitude 7.7 is a major component of seismic hazard in southern California and northern Mexico. To gain insights into how intensely the region will shake during such an event, the Southern California Earthquake Center (SCEC) initiated a major large-scale earthquake simulation in 2004, called TeraShake [8][11]. The simulations used a 3,000 by 1,500

by 400 mesh, dividing the volume into 1.8 billion cubes with a spatial resolution of 200 meters and a maximum frequency of 0.5 hertz, some of the largest and most detailed earthquake simulations of this region. TeraShake runs used up to 2048 processors on the NSF funded TeraGrid [12], and in some cases produced 47 TB of time-varying volumetric data outputs for a single run. The outputs were then registered in a digital library, managed by San Diego Supercomputer Center's Storage Resource Broker (SRB) [10], with a second copy archived into SDSC's High Performance Storage System.

The TeraShake-2 simulations added a physics-based dynamic rupture component to the simulation, which was run at a very high 100 meter resolution, to create the earthquake source description for the San Andreas Fault. This is more physically realistic than the kinematic source description previously used in TeraShake-1. The resulting seismic wave propagation gave a more realistic picture of the strong ground motions that may occur in the event of such an earthquake, which can be especially intense in sediment-filled basins such as the Los Angeles area.

In this paper, we look at challenges we faced porting the application, optimizing the application performance, optimizing the I/O handling, and optimizing the run initialization (Section 2 and 3). Section 4 discusses challenges for large-scale executions. We also discuss the challenges for data archive and management (Section 5), as well as the expertise required for analyzing the results (Section 6). Final section briefly discusses the results of our simulations and draws some conclusions.

2   Challenges for Porting and Optimization

To compute the propagation of the seismic waves that travel along complex paths from a fault rupture across an entire domain, the anelastic wave model (AWM), developed by Kim Olsen et al. [2][4][6][7][8], was picked as the primary model for the SCEC TeraShake simulation. The AWM uses a structured 3D grid with fourth-order staggered-grid finite differences for velocity and stress. One of the significant advantages of the code is the use of Perfectly Matched Layers absorbing boundary conditions on the sides and bottom of the grid, and a zero-stress free surface boundary condition at the top [7]. The AWM code is written in Fortran 90. Message passing is done with MPI using domain decomposition. I/O is done using MPI-I/O so that all processors write velocity output to a single file. The code was extensively validated for a wide range of problems, from simple point sources in a half-space to dipping propagating faults in 3D crustal models [7].

The computational challenges in porting the AWM code to the TeraGrid were two fold. First we identified and fixed the bugs related to Message Passing Interface (MPI) and MPI-IO that caused the code to hang on the target platforms. We found that the original design of the MPI-IO data type in the code that represents count blocks was defined at each time step, which caused a memory leak problem. Our improved version of the code defined indexed data type once only at the initialization phase, and effectively set new views by each task of a file group to obtain efficient MPI-IO performance. For MPI-IO optimization, we modified the collective writes

from using an individual file pointer to using an explicit offset, which not only made large output writing possible, but also greatly improved the I/O performance.
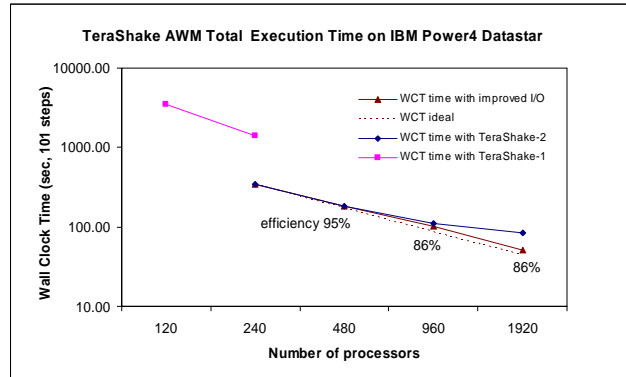

Fig. 1. Strong scaling of AWM with a parallel efficiency of 86% on 1920 processors.

The second effort was to enhance and integrate features necessary for large-scale simulations. Previous simulations of the Southern California region with the AWM code were only tested up to a 592 x 592 x 592 mesh. For the TeraShake case with 1.8 billion mesh points, new problems emerged in managing memory requirements for problem initialization. The code was enhanced from 32-bit to 64-bit for managing 1.8 billion mesh points. To improve the code performance, we profiled the execution time of each part of the code, and identified its performance bottleneck. We optimized cache performance and reduced instruction counts. Some of the very time-consuming functions were in-lined, which immediately saved more than 50% of the initialization time. The reduction of the required memory size and tuning of data operations were necessary steps to scale the code up to the TeraShake scale.

As part of the TeraShake-2 effort, we integrated a new dynamic rupture component into the AWM. This new feature models slip dynamically on the fault surface to generate a more realistic source than the kinematic source description used in TeraShake-1.

The TeraShake simulation poses significant challenges for I/O handling. In the heavy-I/O case, the I/O takes 46% of the total elapsed time on 240 processors of the 10 teraflops TeraGrid Power4 p655 Datastar at SDSC, and the performance saturates quickly as the number of processors increases to more than a few hundred. To improve the disk write performance, we carefully calculated the runtime memory utilization of writes, and accumulated output data in a memory buffer until it reached an optimized size before writing the data to disk.. This optimization through best tradeoff between I/O performance and memory overhead alone reduced the I/O time by a factor of 10 or more, resulting in a very small fraction of the surface velocity write time compared to the total elapsed time.

The simulation algorithm showed very good scaling as a function of the number of processors. The integrated AWM code scales up to 2,048 processors on Datastar. Figure 1 illustrates the significant improvement of scaling after I/O tuning. The figure

also shows the improvement of single CPU performance using machine-specific aggressive optimization flags. The overall performance optimization of the code forms the basis for a parallel efficiency of 96% on 40,960 Blue Gene/L processors at IBM TJ Watson, the latest achievement for petascale earthquake simulation.

3   Challenges for Initialization

AWM initialization presented a significant challenge as we scaled up to TeraShake problem size. Originally the AWM didn't separate the mesh generator processing from the finite difference solver. This made it difficult to scale the code up to a large problem size. While allocated memory is about 1 GB per processor for the finite difference solver, the mesh generation processing performed during the initialization stage required much more memory. Tests performed on the 32-way 256 GB memory Datastar p690 used around 230 GB of memory for initialization. Note that while TeraShake-1 used an extended kinematic source defined at 18,886 points, TeraShake-2 used dynamic sources which were defined at 89,095 points. Memory required per processor using the dynamic source exceeds 4 GB, far beyond the limit of the memory available per processor on both target machines TeraGrid IA-64 and Datastar.

To reduce the memory requirements, we deallocated the arrays not being actively used, and reused existing arrays. More importantly, we separated the source and mesh initialization step from the main production run, so that a pre-processing step is performed to prepare sub-domain velocity model and source partition. With this strategy, the production run only reads in source and mesh input data needed by each processor. This means the production runs with dynamic sources required only the memory size associated with the point source, which reduced the memory requirement by a factor of 8.

To improve disk read performance, we optimized the code by reading data in bulk. We aggressively read data, with data retrieval going beyond the disk attached to the local processor. We calculated the actual location of the data and then assigned the read to the corresponding processors, which improved the disk read performance by a factor of 10.

The final production initialization for TeraShake-1 used a 3-D crustal structure based on the SCEC Community Velocity Model Version 3.0. The source model is based on that inferred for the 2002 Denali Earthquake (M7.9), and some modifications were made in order to apply it to the southern San Andreas fault [8].

4   Challenges for Executions

The large TeraShake simulations were expected to take multiple days to complete. As we prepared the code for use in this simulation, we recognized that the foremost needs were the capabilities of checkpoint and restart which were not available in the original code. We integrated and validated these capabilities, partly prepared by Bernard Minster's group at Scripps Institution of Oceanography. Subsequently, we added

more checkpoints/restart features for the initialization partition, as well as for the dynamic rupture mode. To prepare for post-processing visualizations, we separated the writes of volume velocity data output from writes of velocity surface data output. The latter was output at each time step. To track and verify the integrity of the simulation data collections, we generated MD5 checksums in parallel at each processor, for each mesh sub-array in core memory. The parallelized MD5 approach substantially decreased the time needed to checksum several Terabytes of data.
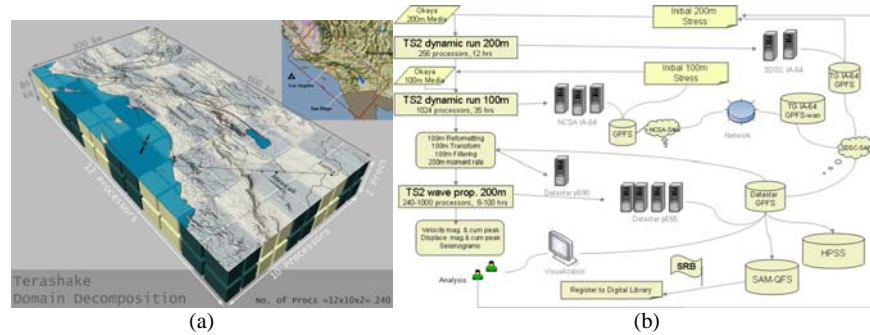

(a)   (b)

Fig. 2. (a) TeraShake domain decomposition on SDSC IBM Power4 DataStar p655 nodes. The rectangle in red on the top right inset shows the simulation region that is 600km long and 300km wide. Domain decomposition of the region onto 240 processors is shown in the center. (b) SCEC seismic wave simulation data flow.

The TeraShake runs required a powerful computational infrastructure as well as an efficient and large scale data handling system. We used multiple TeraGrid computers for the production runs at different stages of the project. The data-intensive TeraShake1.2 simulation, which generated 47 TB volume outputs, used 18,000 CPU hours on 240 Datastar processors (Fig. 2a). The optimal processor configuration was a trade-off between computational and I/O demands. Volume data was generated at each 10th to 100th time step for the run. Surface data were archived for every time step. Checkpoint files were created at each 1000th step in case restarts were required due to reconfigurations or eventual run failures. The model computed 22,728 time steps of 0.011 second duration for the first 250 seconds of the earthquake scenario.

The TeraShake-2 dynamic rupture simulations used a mesh size of 2992 x 800 x 400 cells at 100m resolution, after the appropriate dynamic parameters were determined from several coarser-grid simulations with 200 m cells. Figure 2b shows the execution and dataflow between compute resources and archiving storage system. The 200m resolution runs were conducted on 256 TeraGrid IA-64 processors at SDSC. The 100m resolution runs were conducted on 1024 TeraGrid IA-64 processors at the National Center for Supercomputing Applications. The TeraShake-2 wave propagation runs were executed on up to 2000 processors of Datastar, determined to be the most efficient available processor. The simulation output data were written to the DataStar GPFS parallel disk cache, archived on the Sun Sam-QFS file system, and registered into the SCEC Community Digital Library supported by the SDSC SRB.

5 Challenges for Data Archive and Management

The data management was highly constrained by the massive scale of the simulation. The output from the seismic wave propagation was migrated onto both a Sun Sam-QFS file system and the IBM High Performance Storage System (HPSS) archive as the run progressed – and moving it fast enough at sustained data transfer rate over 120 MB/sec to keep up with the 10 terabytes per day of simulation output.

The TeraShake simulations have generated hundreds of terabytes of output and more than one million files, with 90,000 - 120,000 files per simulation. Each simulation is organized as a separate sub-collection in the SRB data grid. The sub-collections are published through the SCEC community digital library. The files are labeled with metadata attributes which defined the time steps in the simulation, the velocity component, the size of the file, the creation date, the grid spacing, and the number of cells, etc [5]. All files registered into the data grid can be accessed by their logical file name, independently of whether the data were on parallel file system GPFS, Sam-QFS, or the HPSS archive. General properties of the simulation such as the source characterization are associated as metadata for the simulation collection. Integrity information is associated with each file (MD5 checksum) as well as existence of replicas. Since even tape archives are subject to data corruption, selected files are replicated onto either multiple storage media or multiple storage systems.
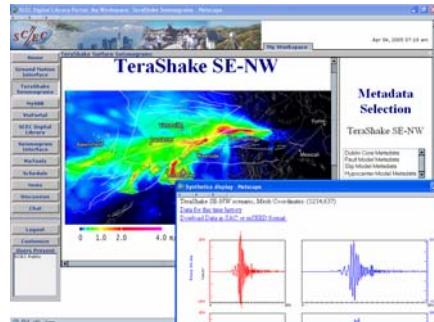

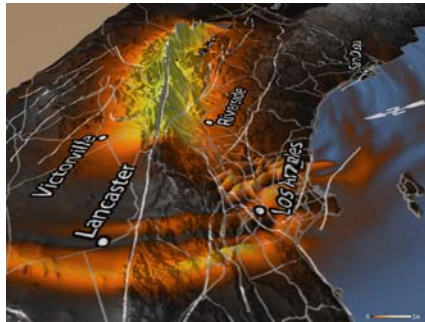**Fig. 3.** User interaction with the TeraShake Surface Seismograms portlet at the SCECLib Portal.

Fig. 4. TeraShake Surface rendering of displacement magnitude with topographic deformation

The SCEC digital library includes the digital entities (simulation output, observational data, and visualizations), metadata about each digital entity, and services that can be used to access and display selected data sets. The services have been integrated through the SCEC portal into seismic-oriented interaction environments [5][9]. A researcher can then select an earthquake simulation scenario and select a location on the surface, by pointing and clicking over the interactive cumulative peak velocity map, or interact with the full service resolution data amounting to one terabyte (Fig. 3).

6   Challenges for Analysis of Results

Verification of the simulation progress at runtime and thereafter seismological assessment of data computed was a major concern for the success of the TeraShake project. Visualization techniques helped solve this problem by rendering the output data during the simulation run. Animations of these renderings were instantly made available for analysis. SDSC's volume rendering tool Vista, based on the Scalable Visualization Toolkit (SVT), was used for visualizations. Vista employs ray casting for performing volumetric rendering. Surface data have been visualized with different variables (velocities and displacements) and data ranges in multiple modes. The resulting animations have proven valuable not only to domain scientists but also to a broader audience by providing an intuitive way to understand the TeraShake simulation results. Visualizations alone have consumed more than 40,000 CPU hours on Datastar and IA-64 at SDSC. Upwards of 100 visualization runs were performed, with each run utilizing 8 to 256 processors in a distributed manner. The results have produced over 130,000 images [1] (Fig. 4 shows an example).

Scientists want to conduct hands on analysis in an attempt to gain a better understanding of output data. The size of TeraShake data poses a significant problem for accessibility and analysis. We developed a web front end where scientists can download the data and are able to create custom visualizations over the web directly from surface data. The portal uses LAMP (Linux, Apache, MySQL, PHP) and Java technology for web middle-ware and on the back-end compute side relies on specialized programs to fetch data from the archive, visualize, composite, annotate and make it available to client browser.

7   Summary

The TeraShake simulation was one of the early projects at SCEC targeting capability computing, and the code accuracy has been extensively verified for anelastic wave propagation and dynamic fault rupture [3]. A major result of the simulation was the identification of the critical role a sedimentary waveguide along the southern border of the San Bernardino and San Gabriel Mountains has in channeling seismic energy into the heavily populated San Gabriel and Los Angeles basin areas. The simulations have considerable implications for seismic hazards in southern California and northern Mexico.

The TeraShake simulations demonstrated that optimization and enhancement of major applications codes are essential for using large resources (number of processors, number of CPU-hours, terabytes of data produced). TeraShake also showed that multiple types of resources are needed for large problems: initialization, run-time execution, analysis resources, and long-term data collection management.

The improvements made to the TeraShake AWM have created a community code that can be used by the wider SCEC community to perform large scale earthquake simulations. The TeraShake code is already being integrated for use in other SCEC Projects such as the SCEC Earthworks Science Gateway.

SCEC has identified a PetaShake platform for petascale simulations of dynamic ruptures and ground motions with outer/inner scale ratios as high as $10^{4.5}$. Excellent scalability of TeraShake AWM shown on 40k BG/L processors has demonstrated an important step toward petascale earthquake rupture and wave propagation computing.

References

1. Chourasia. A., Cutchin. S. M., Olsen. K.B., Minster. B., Day. S., Cui. Y., Maechling. P., Moore. R., Jordan. T.: Insights gained through visualization for large earthquake simulations, submitted to Computer Graphics and Application Journal (2006).
2. Cui, Y., Olsen, K., Hu, Y., Day, S., Dalguer, L., Minster, B., Moore, R., Zhu, J., Maechling, P., Jordan, T.: Optimization and Scalability of A Large-Scale Earthquake Simulation Application. Eos, Trans, AGU 87(52), Fall Meet. Suppl. (2006), Abstract S41C-1351.
3. Dalguer, L.A. and Day, S.: Staggered-grid split-node method for spontaneous rupture simulation, J. Geophys. Res. (2007), accepted.
4. Day, S.M. and Bradley, C.: Memory-efficient simulation of an-elastic wave propagation, Bull. Seis. Soc. Am. 91 (2001) 520-531
5. Faerman, M., Moore, R., Cui, Y., Hu, Y., Zhu, J., Minister, B. and Maechling, P.: Managing Large Scale Data for Earthquake Simulations, Journal of Grid Computing, Springer-Verlag DOI 10.1007/s10723-007-9072-x (2007)
6. Olsen, Kim B.: Simulation of three-dimensional wave propagation in the Salt Lake Basin, Ph.D. thesis, The University of Utah, (1994)
7. Olsen, K.B., Day, S.M., and Bradley, C.R.: Estimation of Q for long-period (>2 s) waves in the Los Angeles Basin, Bull. Seis. Soc. Am. 93 (2003) 627-638
8. Olsen, K., Day, S.M., Minster, J.B., Cui, Y., Chourasia, A., Faerman, M., Moore, R., Maechling, P. and Jordan, T.: Strong Shaking in Los Angeles Expected from Southern San Andreas Earthquake, Geophysical Research Letters, Vol 33 (2006), 1-4
9. Olsen, K., Zhu, J. and Talley, J.: Dynamic User Interface for Cross-plot, Filtering and Upload/Download of Time Series Data. Eos, Trans, AGU 87(52), Fall Meet. Suppl. (2006), Abstract IN51B-0814
10. Moore, R., Rajasekar, A., Wan, M.: Data Grids, Digital Libraries and Persistent Archives: An Integrated Approach to Publishing, Sharing and Archiving Data. Special Issue of the Proceedings of the IEEE on Grid Computing, Vol. 93, No.3 (2005) 578-588
11. SCEC/CME Web Site: http://www.scec.org/cme.
12. TeraGrid Website: http://teragrid.org/about/
13. Weldon, R., K. Scharer, T. Furnal and Biasi, G.: Wrightwood and the earthquake cycle: What a long recurrence record tells us about how faults work, Geol. Seismol. Am. Today, 14 (2004) 4-10