BENCHMARK STUDY OF A 3D PARALLEL CODE FOR THE PROPAGATION OF LARGE SUBDUCTION EARTHQUAKES

Mario Chavez[1,2], Eduardo Cabrera[3], Raúl Madariaga[2], Narciso Perea[1], Charles Moulinec[4], David Emerson[4], Mike Ashworth[4], Alejandro Salazar[3]

[1] Institute of Engineering, UNAM, C.U., 04510, Mexico DF, Mexico
[2] Laboratoire de Géologie CNRS-ENS, 24 Rue Lhomond, Paris, France
[3] DGSCA, UNAM, C.U., 04510, Mexico DF, Mexico
[4] STFC Daresbury Laboratory, Warrington WA4 4AD, UK

chavez@servidor.unam.mx, eccf@super.unam.mx, raul.madariaga@ens.fr
narpere@iingen.unam.mx, c.moulinec@dl.ac.uk, d.r.emerson@dl.ac.uk,
m.ashworth@dl.ac.uk, alejandro@labvis.unam.mx

Abstract. Benchmark studies were carried out on a recently optimized parallel 3D seismic wave propagation code that uses finite differences on a staggered grid with 2nd order operators in time and 4th order in space. Three dual-core supercomputer platforms were used to run the parallel program using MPI. Efficiencies of 0.91 and 0.48 with 1024 cores were obtained on HECToR (UK) and KanBalam (Mexico), and 0.66 with 8192 cores on HECToR. The 3D velocity field pattern from a simulation of the 1985 Mexico earthquake (that caused the loss of up to 30000 people and about 7 billion US dollars) which has reasonable agreement with the available observations, shows coherent, well developed surface waves propagating towards Mexico City.

Key words: Benchmark, modeling, finite difference, earthquakes, parallel computing.

1 Introduction

Realistic 3D modeling of the propagation of large subduction earthquakes, such as the 1985 Mexico earthquake (Fig. 1), poses both a numerical and a computational challenge, particularly because it requires enormous amounts of memory and storage, as well as an intensive use of computing resources. As the recurrence time estimated for this highly destructive type of event in Mexico is only a few decades, there is a seismological, engineering and socio-economical interest in modeling them by using parallel computing [1].
In this paper, we present the results from benchmark studies performed on a recently optimized parallel 3D wave propagation staggered-grid finite difference code, using the Message Passing Interface (MPI). The code was run

on three dual-core platforms, i.e.: KanBalam (KB, Mexico, [2]), HPCx (UK, [3]) and HECToR (UK, [4]). Characteristics of the three systems are shown in Table 1. In section 2, a synthesis of the 3D wave propagation problem and the code are presented; a description of the strategy followed for the data parallelism of the problem and the MPI implementation are discussed in section 3. The benchmark experiment performed on the code and its main conclusions are addressed in section 4 and in section 5, the results obtained for the modeling of the seismic wave propagation of the Mexico 1985 Ms 8.1 subduction earthquake are given.

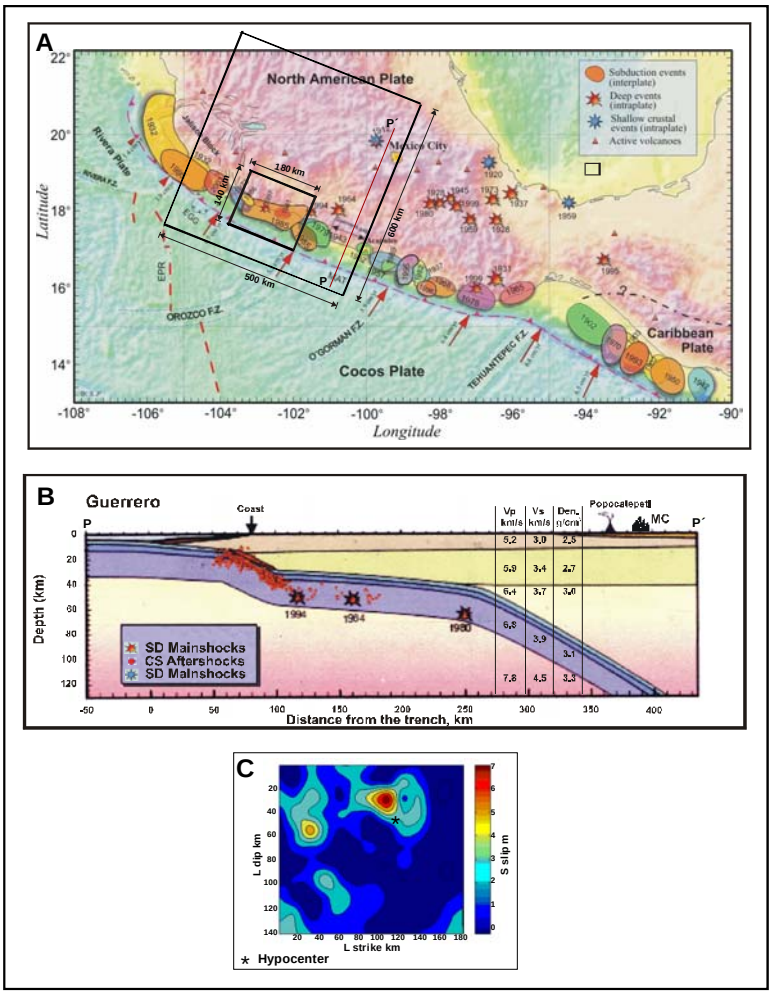

Fig. 1. A) Inner rectangle is the rupture area of the 19/09/1985 Ms 8.1 earthquake on surface projection of the 500x600x124 km earth crust volume 3DFD discretization; B) profile P-P´; C) Kinematic slip distribution of the rupture of the 1985 earthquake [1].

Table 1. Characteristics of the 3 Supercomputer platforms used in the benchmark study.

Platform	HPCx	KB	HECToR
Processor	IBM PowerPC 5 1.5GHz dual core	AMD Opteron 2.6GHz dual core	AMD Opteron 2.8GHz dual core
Cache	L1 data 32KB and L1 instr 64KB per core L2 1.9MB shared L3 128 MB shared	L1 instr and data 64KB per core L2 1MB shared	L1 instr and data 64KB per core L2 1MB shared
FPU's	2 FMA	1Mult, 1Add	1Mult, 1Add
Peak performance/core	6 GFlop/s	5.2 GFlop/s	5.6 GFlop/s
Cores	2560	1368	11328
Peak Perf	15.4 TFLOP/s	7.12 TFLOP/s	63.4 TFLOP/s
Linpack	12.9 TFLOP/s	5.2 TFLOP/s	54.6 TFLOP/s
Interconnect	IBM High performance switch	Infiniband Voltaire switch 4x, fat tree topology	Cray SeaStar2 3D toroidal topology
Bandwidth	4GB/s	1.25 GB/s	7.6 GB/s
latency	5.5 μs	13 μs	5.5 μs
File system	GPFS	Lustre	Lustre

2 3D Wave propagation modeling and its algorithm

The 3D velocity-stress form of the elastic wave equation, consists of nine coupled, first order partial differential hyperbolic equations for the three particle velocity vector components and the six independent stress tensor components [1, 5].

The finite difference staggered algorithm applied to the mentioned equations is an explicit scheme which is second-order accurate in time and fourth-order accurate in space. Staggered grid storage allows the partial derivatives to be approximated by centered finite differences without doubling the spatial extent of the operators, thus providing more accuracy. The discretization of the 3D spatial grid is such that $x_i=x_0+(i\text{-}1)h_x$, $y_j=y_0+(j\text{-}1)h_y$, and $z_k=z_0+(k\text{-}1)h_z$ for i=1, 2, 3, … I, j= 1, 2, 3, … ,J, and k=1, 2, 3, … ,K, respectively. Here x_0, y_0, z_0 are the minimum grid values and h_x, h_y, h_z give the distance between points in the three coordinate directions. The time discretization is defined by $t_l=t_0+(l\text{-}1)h_t$ for l=1, 2, 3,…,L. Here t_0 is the minimum time and h_t is the time increment.

3 Parallel implementation of the 3DFD algorithm

We use 3D data parallelism for efficiency. The domain was decomposed into small subdomains and distributed among a number of processors, using

simple partitioning to give an equal number of grid points to each processor [1]. This approach is appropriate for the 3DFD wave propagation code, as large problems are too big to fit on a single processor [1].
The Message Passing Interface (MPI) was used to parallelize the 3DFD code [1]. In particular, MPI_Bcast, MPI_Cart_Shift and MPI_SendRecv instructions were used; the first two to communicate the geometry and physical properties of the problem, before starting the wave propagation loop, and the last to update the velocities and stresses calculated at each time step. The nature of the chosen 3DFD staggered scheme precluded the efficient application of overlapping MPI_Cart_Shift, MPI_SendRecv operations with computations.

Parallel I/O from MPI-2 was used in the code to read the earth model data by all processors and to write the velocity seismograms by the processors corresponding to the free surface of the physical domain [1], which is only a small percentage of the total number of processors. As this type of parallel I/O is machine independent, it fitted the benchmark experiment performed on the three platforms.

4 Benchmark experiment

As mentioned above the code was run on three dual-core platforms, i.e.: KanBalam (KB, Mexico, [2]), HPCx (UK, [3]) and HECToR (UK, [4]).

The actual size of the problem is 500 x 600 x 124 km (Fig 1), and its physical properties are also shown in the Fig. We used spatial discretizations hx = hy = hz, of: 1.0, 0.500, 0.250 and 0.125 km (to include thinner surficial geologic layers in the Z direction) and the associated time discretizations were 0.03, 0.02, 0.01 and 0.005 s, respectively (to comply with the Courant-Friedrich-Lewy condition). Therefore, Nx=500, 1000, 2000, 4000; Ny=600, 1200, 2400, 4800 and Nz=124, 248, 496, 992 are, the model size in the X, Y and Z directions, respectively (notice that Nz is about 0.25 of Nx and Ny). The number of time steps, Nt, used for the experiment was 4000.
Speedup, Sp, and efficiency, E, among others, are the most important metrics to characterize the performance of parallel programs. Theoretically, speedup is limited by Amdahl's law [6], however there are other factors to be taken into account, such as: communications costs, type of decomposition and its resultant load balance, I/O and others [1]. Sp and E, disregarding those factors, can be expressed by:

$$Sp \equiv mT_1(n/m)/T_m(n/m), E \equiv T_1(n/m)/T_m(n) \quad (1)$$

for a scaled-size problem n (weak scaling), and for a fixed-size problem (strong scaling)

$$Sp \equiv T_1/T_m \quad , E \equiv T_1/T_m\, m \quad (2)$$

where T_1 is the serial time execution and T_m is the parallel time execution on m processors.

If the communications costs and the 3D decomposition are taken into account, the expression for *Sp* is:

$$Sp \equiv \frac{A\Gamma R^3}{A\Gamma R^3 / m + 24\left(\iota + 4\beta R^2 / m^{2/3}\right)}, \qquad (3)$$

where the cost of performing a finite difference calculation on $mx \times my \times mz$, *m,* processors is $A\Gamma R^3 / m$; A is the number of floating operations in the finite difference scheme (velocity-stress consists of nine coupled variables); Γ is the computation time per flop; *R* is equal to *Nx* x *Ny* x *Nz* ; ι is the latency and β is the inverse of bandwidth [1]. This scheme requires the communication of two neighbouring planes in the 3D decomposition [1].

This benchmark study consisted of both scaled-size (weak scaling) and fixed-size (strong scaling) problems. In the former, the number of processors (m) utilized for KB and HECToR varied from 1 - 8192 and for the latter, 64 and 128 processors were used on the three platforms. For both type of problems, and whenever it was possible, experiments with one or two cores were performed, for KB, HECToR, and HPCx platforms.

The results of the two studies are synthesized in Table 2 and Fig. 2. From the results of the weak-scaling problems, it can be concluded that when large amounts of cores (1024 for KB) and (8192 for HECToR), with respect to the total number available in the tested platform, *Sp* and *E* decrease considerably, to 492 and 0.48 and 5375 and 0.66, for KB and HECToR, respectively. We think that this behavior is due to the very large number of communications demanded among the processors by the 3DFD algorithm [1]. This observation is more noticeable for the dual-core results, due to, among other factors, the fact that they are competing for the cache memory available and the links to the interconnect, and that this is stressed when thousands of them are used. The opposite behavior of *Sp* and *E* is observed when only tens, hundreds (for KB) or up to 1024 cores are used for HECToR, Table 2, Fig 2.

From the results for the strong-scaling problem shown in Table 2, it can be concluded that for the three platforms, the observed *Sp* and *E* are very poor, particularly when the two cores were used,. The “best” results were obtained for HECToR, followed by KB and HPCx. Given that the mentioned observation is valid for the three platforms, we can conclude that the 3DFD code tested is ill suited for strong-scaling problems.

5 Seismological results for the 19/09/1985 Mexico's Ms 8.1 subduction earthquake

Herewith we present examples of the type of results that for the 1985 Mexico earthquake (Fig. 1) were obtained on the KB system with the parallel MPI implementation of the 3DFD code. At the top of Fig 3, the 3D low frequency velocity field patterns in the X direction, and the seismograms obtained at observational points, in the so-called near (Caleta) and far fields (Mexico City),

Table 2. Scaled and fixed –size* models: *mi* (i = x, y, z) processors used in each axis (*mz* was fixed to 4 because *Nz* is about one fourth of *Nx* and *Ny*), timings, speedup, efficiency and memory per subdomain (Mps) obtained for KB, HECToR and HPCx. The total run time of KB of 37600 s was used to compute *Sp* and *E* for the (*) cases.

Size model and spatial step (dh km)	m	mx	my	mz	Cores per chip used	Total run time (s)	Speedup (Sp)	Efficiency (E)	Mps (GB)
500x600x124 (1) **KB**	1	1	1	1	1	13002	1	1	1.9
1000x1200x248 (0.5) **KB**	16	1	4	4	1	6920	30	1.9	0.97
1000x1200x248 (0.5) **KB**	16	1	4	4	2	11362	18	1.14	0.97
2000x2400x496 (0.25) **KB**	128	4	8	4	2	15439	108	0.84	0.97
4000x4800x992 (0.125) **KB**	1024	16	16	4	2	27033	492	0.48	0.97
500x600x124 (1) **HECToR**	1	1	1	1	1	11022	1	1	1.9
1000x1200x248 (0.5) **HECToR**	16	1	4	4	1	6404	28	1.7	0.97
1000x1200x248 (0.5) **HECToR**	16	1	4	4	2	10583	17	1.04	0.97
2000x2400x496 (0.25) **HECToR**	128	4	8	4	1	6840	207	1.6	0.97
2000x2400x496 (0.25) **HECToR**	128	4	8	4	2	11083	127	0.99	0.97
4000x4800x992 (0.125) **HECToR**	1024	16	16	4	1	7200	1568	1.53	0.97
4000x4800x992 (0.125) **HECToR**	1024	16	16	4	2	12160	928	0.91	0.97
8000x9600x1984 (0.0625) **HECToR**	8192	32	32	8	2	16800	5375	0.66	0.97
1000x1200x248 (0.5) **KB***	1	1	1	1	1	37600	1	1	14.3
1000x1200x248 (0.5) **KB***	64	4	4	4	1	2699	13.9	0.22	0.242
1000x1200x248 (0.5) **KB***	64	4	4	4	2	3597	10.5	0.16	0.242
1000x1200x248 (0.5) **KB***	128	4	8	4	1	1681	22.4	0.18	0.121
1000x1200x248 (0.5) **KB***	128	4	8	4	2	2236	16.8	0.13	0.121
1000x1200x248 (0.5) **HECToR***	64	4	4	4	1	1898	19.8	0.31	0.242
1000x1200x248 (0.5) **HECToR***	64	4	4	4	2	2910	12.9	0.20	0.242
1000x1200x248 (0.5) **HECToR***	128	4	8	4	1	878	42.8	0.33	0.121
1000x1200x248 (0.5) **HECToR***	128	4	8	4	2	1420	26.5	0.21	0.121
1000x1200x248 (0.5) **HPCx***	64	4	4	4	2	4080	9.2	0.14	0.242
1000x1200x248 (0.5) **HPCx***	128	4	8	4	2	2100	17.9	0.14	0.121

of the wave propagation pattern for times equal to 49.2 and 136.8 s. The complexity of the propagation pattern at t = 49.2 s, when the seismic source is still rupturing, is contrasted by the one for t = 136.8 s, in which packages of coherent, well developed surface waves, are propagating towards Mexico City Finally, at the bottom of Fig. 3 we show the observed and synthetic (for a

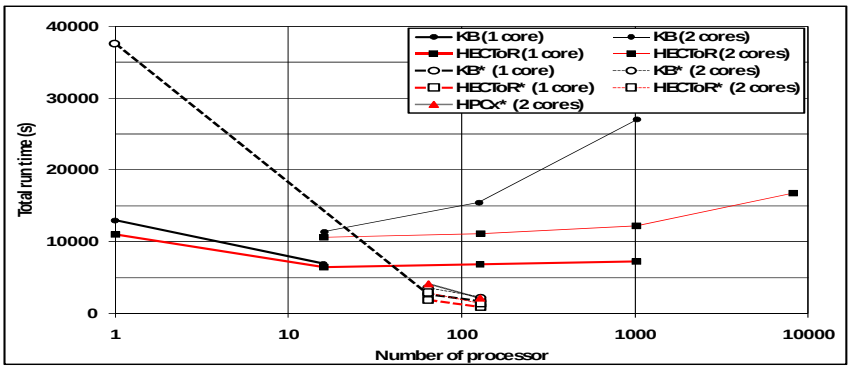

Fig. 2 Execution time vs number of processors for the three platforms for Scaled and fixed –size* models of Table 2.

spatial discretization dh = 0.125 km) low frequency, north-south velocity seismograms of the 19/09/1985 Ms 8.1 Mexico earthquake, and their corresponding Fourier amplitude spectra for the firm soil Tacubaya site in Mexico City, i.e. at a far field observational site. Notice in Fig. 3, that the agreement between the observed and the synthetic velocity seismogram is reasonable both in the time and in the frequency domain.

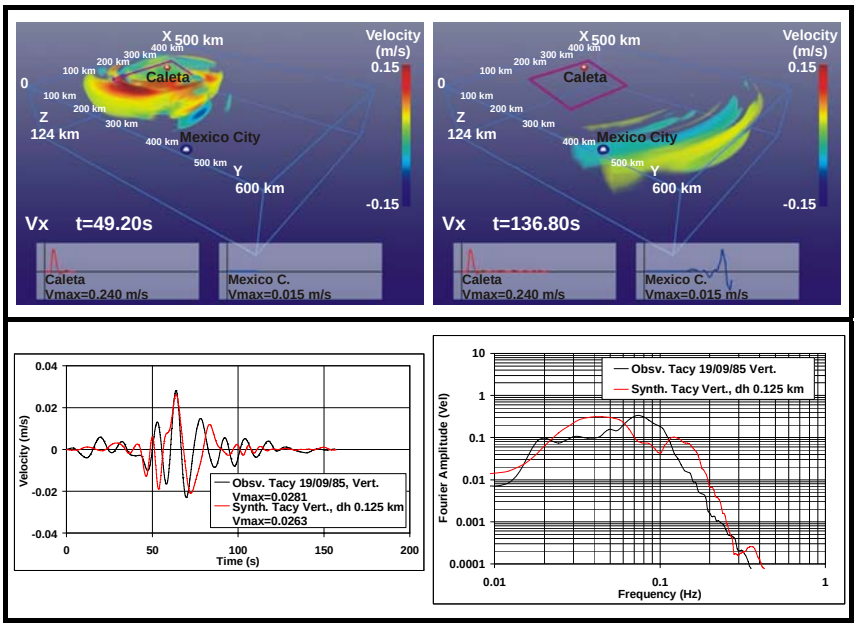

Fig. 3. Top) 3D Snapshots of the velocity wavefield in the X direction of propagation for t = 49.2 and 136.8 s for the 1985 Mexico earthquake; Bottom) Left side observed and synthetic seismograms at Mexico City, right side Fourier amplitude spectra.

7 Conclusions

Benchmark studies were carried out on a recently optimized seismic wave propagation 3D, parallel MPI finite difference code that uses 2^{nd} order operators in time and 4^{th} order in space on a staggered grid, 3DFD. Three dual-core supercomputer platforms were used to test the program. Efficiencies of 0.91 and 0.48 with 1024 cores were obtained for the HECToR (UK) and KanBalam (Mexico) machines, and of 0.66 for 8192 cores for HECToR. In order to improve its performance, probably, Non-blocking MPI communications should be incorporated in a future version of the code. The agreement between the observed and the synthetic velocity seismograms obtained with 3DFD and a dh = 0.125 km [1], is reasonable, both in time and in frequency domains. The 3D velocity field patterns from a simulation of the 1985 Mexico earthquake (which caused the loss of up to 30,000 people and about 7 billion US dollars), show large amplitude, coherent, well developed surface waves, propagating towards Mexico City.

Acknowledgments

We would like to thank the support of Genevieve Lucet, José Luis Gordillo, Hector Cuevas, the supercomputing staff and Marco Ambriz, of DGSCA, and the Institute of Engineering, UNAM, respectively. We acknowledge DGSCA, UNAM for the support to use KanBalam, as well as the STFC Daresbury Laboratory to use HECToR and HPCx. The authors also acknowledge support from the Scientific Computing Advanced Training (SCAT) project through EuropeAid contract II-0537-FC-FA (http://www.scat-alfa.eu).

References

[1] Cabrera E., M. Chavez, R. Madariaga, N. Perea, M. Frisenda. 3D Parallel Elastodynamic Modeling of Large Subduction Earthquakes. F. Capello et al. (eds): Euro PVM/MPI 2007, LNCS 4757, pp. 373-380, 2007, Springer-Verlag Berlin Heidelberg 2007.
[2] http://www.super.unam.mx/index.php?op=eqhw
[3] HPCx Home Page http://www.hpcx.ac.uk/
[4] HECToR Home Page http://www.hector.ac.uk/
[5] S. E. Minkoff. Spatial Parallelism of a 3D Finite Difference Velocity-Stress Elastic Wave Propagation code. SIAM J. Sci. Comput. Vol 24, No 1, 2002, pp 1-19.
[6] G. Amdahl, Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities, in Conference Proceedings, AFIPS, 1967, pp. 483-485.