Advance in Technology, Engineering and Compuring – A Multinational Colloquium – 2017
Editor(s) Vinod P., Sunil Jacob, Sheeja Janaradhanan and Ratish Menon
doi:10.3850/978-981-11-0744-3_17-27-cd

Parallel Computing for Earthquake Analysis using Opensees

Chandana M.[a], Unni Kartha G.[b] and Mahesh C.[c]

Department of Civil Engineering and [c]Department of Computer Science and Engineering, Federal Institute of Science and Technology, Angamaly, Kochi, Kerala 683577, India. [a]chandumanat@gmail.com, [b]unnikartha@gmail.com, [c]mahesh.fisat@gmail.com.

Finite Element Method is the most widely used numerical technique to predict the approximate response of a structure under various loading conditions. Predicting the response of a structure to seismic loading is computationally very intensive and time consuming. Parallel finite element method is one solution to such situations where the computation is efficiently distributed among multiple cores available in modern supercomputers. OpenSees is an open source software with capabilities for performing parallel finite element method, developed by Pacific Earthquake Engineering Research Centre (PEER), USA specifically for carrying out earthquake engineering simulations. In this paper, a finite element model of a structure is developed and is analysed for different earthquake input motions and the response is studied using the parallel capabilities of the software. The computational advantage of using parallel software is studied.

1. Introduction

Performance based methodologies in structural engineering have increased the need for high fidelity computation and analysis of structural responses. Earthquakes are one of the most severe natural hazards affecting people around the world and the structural dynamic response under seismic loading are nonlinear functions of many factors. Dynamic analysis for simple structures can be carried out manually, but human minds have difficulties in grasping the behaviour of its complex surroundings and structures in one operation. Thus the process of dividing a system into individual components or elements, whose behaviour can be readily understood and then rebuilding the original structure from these divided elements, to study its behaviour, is a better way to proceed for its analysis. Hence, finite element method is the most widely used numerical technique to predict the approximate response of a structure under various loading conditions. Also, predicting the response of a structure to seismic loading is computationally very intensive and time consuming. Parallel finite element method overcomes this limitation, where the computation is efficiently distributed among multiple cores in parallel computers. Parallel computation is able to give the most significant result in terms of saving time, modeling, analysing and designing. The software OpenSees is a powerful tool for performing parallel finite element analysis.

2. Parallel Analysis in Opensees

In each finite element analysis, an analysis is used to construct four main types of objects, which are the main abstractions in OpenSees framework. They are, a set of modules to perform creation of the finite element model, specification of an analysis procedure, selection of quantities to be monitored during the analysis, and the output of results. They are: Model Builder, Analysis, Recorder and Domain object.

The various steps for modeling are as shown below:

- (i) The spacial dimension and the degree of freedom at each node are defined.
- (ii) Various variables required for the modeling and analysis are created and are assigned values.
- (iii) The model is created by assigning nodal coordinate values.
- (iv) The boundary constraints are added.
- (v) The geometric transformation from the local coordinate system to global coordinate system is done.
- (vi) The various type of elements in the model is defined.
- (vii) The nodal mass is calculated.
- (viii) The load pattern object which is associated with the timeseries, load and constraints is created.

An Analysis object is a composition of objects from other classes, each of which is responsible for performing an important operation in determining the state of the finite element model. The Analysis class has various subclasses to carry out a range of analysis procedures for static and transient structural analysis. The most important step in the Analysis class is analyze(), which advances the state of the domain for one or more load steps. When analyze() is called for static analysis, it loops over a specified number of steps, forming and solving the equations and then records the solution, once the convergence of the solution is satisfied. The only difference in the code for static analysis and a transient analysis is that in transient analysis, the integrator is given a time increment.

The first step in the parallel analysis in OpenSees is to set the unique process id and the number of processes in the system. The command getNP returns the total number of processes and the command getPID returns the unique process number, through the value returned in [expr [getNP] - 1]. After this, the analysis or the task is split and distributed among all the NP - 1 processors for parallel analysis. Each processor, when it starts will determine its unique id and the number of process involved in the computation. The recorder objects are used to monitor the user defined parameters in the model during the analysis. The Node recorder records the displacement at the nodes and the time argument will place the pseudo time as the first entry in the output line.

2.1 Message Passing Interface

The Message Passing Interface (MPI) allows parallel programs to be written using a distributed memory parallel programming model. In message passing, a program is considered as a collection of processes, in which each process has direct access only to its local memory. Only the memory associated with a particular process may be accessed and hence the computations can only be performed by a process on the data stored in its own memory. In order to perform large computations and for using more than one process to perform the computation, the copy of the data must be sent to the processor which requires it. This is referred to as message passing. The running program is viewed as a collection of independent communicating processes. In MPI, each process executes in its own address space, and is able to communicate with other processes. The processes communicate using send and receive pairs.

3. Numerical Experiment

Parallel computing can be described as using more than one computer, or a computer with more than one processor, to solve a problem. The dynamic analysis of a structure subjected to multiple earthquake excitations is a very time consuming and tiring process. It requires computation on a very large data set. Parallel computing helps in solving such large complex problems and gives a faster ouput. The interpreter OpenSeesMP is used to conduct parallel dynamic analysis of a frame. It has the ability to perform parallelization based on the processor id on which it runs and the number of processes involved in the parallel simulation. The program runs on a user specified number of processes and each process will use the same input file. The message passing interface implementation OpenMPI, which uses shared memory for message transfer is used for this numerical study. The modeling and analysis are contained in the files model.tcl and analysis.tcl respectively, in the program. The analysis.tcl contains the procedures doGravity and doDynamic for static and dynamic analysis. The list of input ground motions to run in the program are saved in a text file.

3.1 Model

A nonlinear elastic section model of the portal frame as shown in the Fig. 1 is developed for the analysis. The frame model is subjected to 60 earthquakes from the NGA West 2 PEER ground motion database and is analysed to study its response. The analysis is performed in the high performance computing facility available at the Centre for High Performance Computing (CHPC), FISAT, with a 32 core and a 62GB RAM server. The program is built in the machine from the source code for Ubuntu operating system. The properties of the frame are:

- Column size : 0.3 x 0.4m
- Beam size : 0.3 x 0.3m
- Beams element: nonlinear beamcolumn element.
- Columns element: nonlinear beamcolumn element.
- Section: Elastic section.
- Supports: Fixed.
- Damping: 5% Rayliegh damping.
- Concrete: M25 grade concrete.

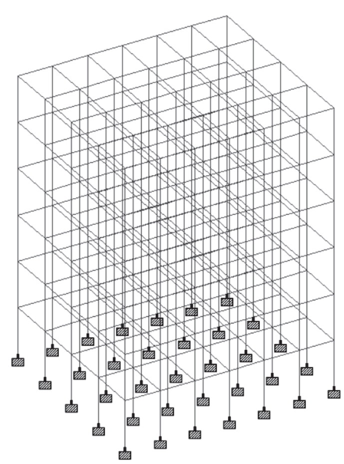

Figure 1. The Frame Model

3.2 Analysis

The Analysis objects perform the analysis. The analysis moves the model along from a state at time t to a state at time t + dt. In OpenSees, each of the analysis objects is composed of several component objects, which defines the type of analysis and how the analysis is performed. The procedure for the finite element analysis in OpenSees is as follows:

(i) The solver object BandGeneral, used to construct the system of equations is defined.

(ii) ConstraintHandler object which determines how the constraint equations regarding the boundary conditions or imposed displacement are enforced in the analysis are defined.

(iii) The DOF Numberer object, RCM is constructed. They determine the mapping between equation numbers and degrees of freedom.

(iv) The KrylovNewton and Newton method of solution of the system of equation is defined for static and transient analysis respectively.

(v) The predictive step for the time t+dt is determined using the LoadControl integrator command for static analysis and using Newmark for dynamic analysis.

(vi) The convergence of the solution of the equation $K\Delta U=R$ is tested using NormDispIncr.

(vii) The type of analysis (static and transient) to be performed is defined.

(viii) The analysis is carried out by applying the load using the command analyze.

3.3 Results

The response of the structure, with time evolution is studied. The computation time taken for the analysis, when using different number of processes is recorded. The Fig. 2 implies that the computation time reduces with increase in number of processes for the analysis. The displacement at the node 245 or the top-most node in the frame is recorded for 60 input ground motions and the maximum displacement for all the 60 input motions is shown in the Fig. 3. The displacement for the input motions 1, 10 and 60 are plotted in the Fig. 4, Fig. 6 and Fig. 8 respectively.

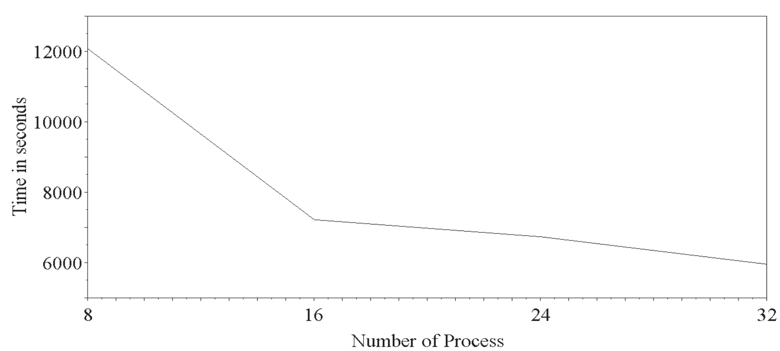

Figure 2. Computation time versus number of processes

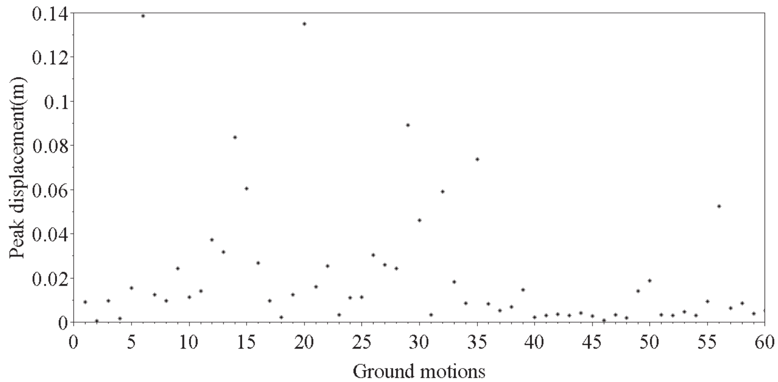

Figure 3. Peak displacement for each input ground motion

The drift between the nodes are recorded for all the input ground motions. The drift between the nodes 1 which is fixed and 6 which is above the node 1 and also between 6 and 11 are plotted below for the input ground motions 1, 10 and 60 in the Fig. 5, Fig. 7 and Fig. 9 respectively.

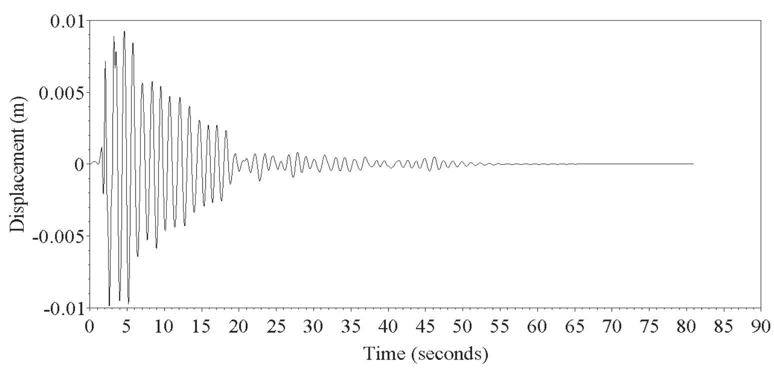

Figure 4. The time history displacement plot of the input ground motion data 1

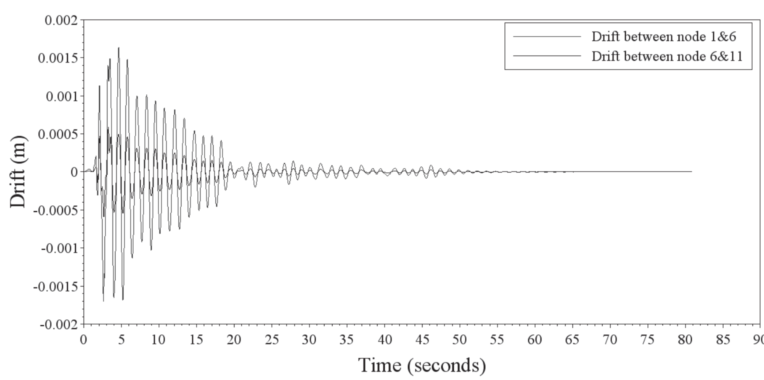

Figure 5. The time history dift plot of the input ground motion data 1

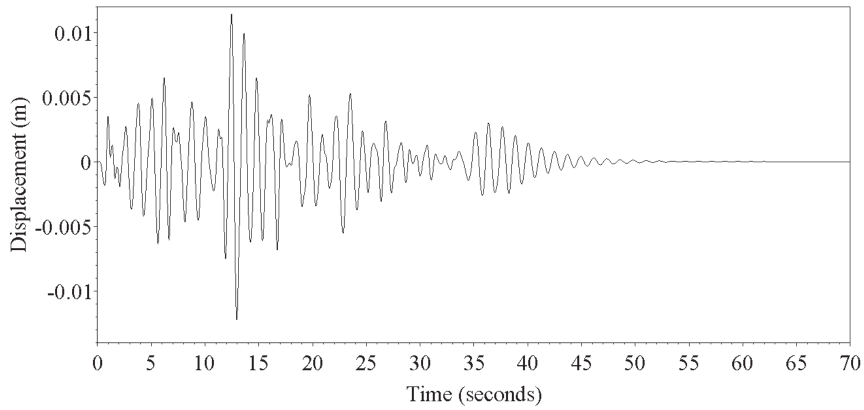

Figure 6. The time history displacement plot of the input ground motion data 10

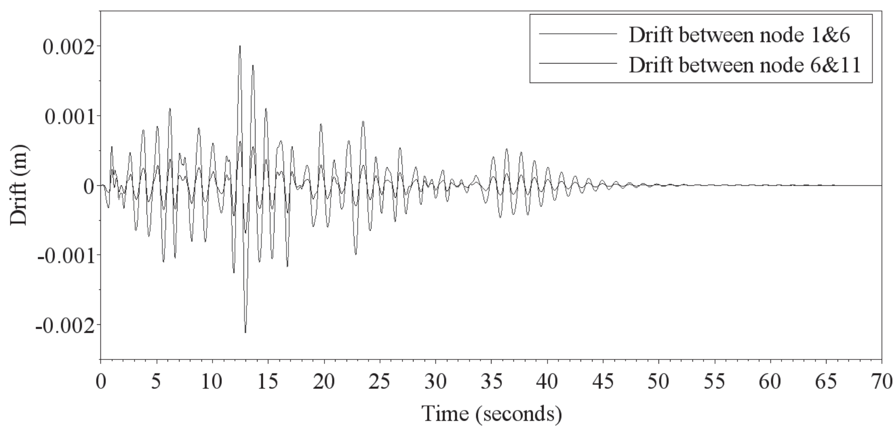

Figure 7. The time history drift plot of the input ground motion data 10

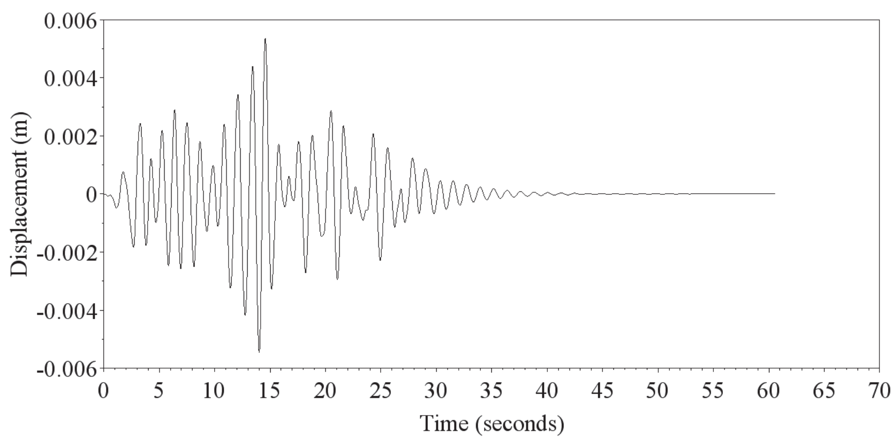

Figure 8. The time history displacement plot of the input ground motion data 60

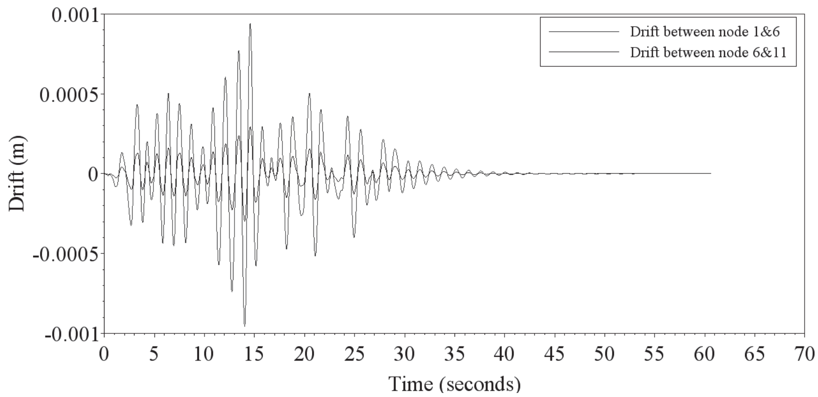

Figure 9. The time history drift plot of the input ground motion data 60

4. Conclusion

Earthquake engineering analysis and simulations demand high computational requirements. Finite element parallel analysis reduces the computation time and it arranges memory by distributing tasks to each processor for any number of processes. Large finite element simulations of problems in almost all fields of engineering requires parallel algorithms to work on clusters with thousands of processor cores. In parallel computation, for a large complex problem to scale to a large number of processors, the computation time must scale linearly with respect to the number of processors and the problem size. The task size of a problem depends on the parallel machine. Hence, a solution that works good in one machine may not work good in another machine. This paper describes a 3D finite element model developed using OpenSees, a software framework for seismic response simulation of structures. OpenSees is a powerful tool for performing finite element analysis and can generate models faster and more accurately. Dynamic analysis of the frame model, subjected to 60 input ground motions is carried out within less time. Hence, it is a major contribution to the field of science and research, to study the behaviour of a model under different loading conditions simultaneously.

References

1. Frank McKenna et al, (2010), Nonlinear Finite-Element Analysis Software Architecture Using Object Composition, *Journal of Computing In Civil Engineering, ASCE, January/February.*
2. Klaus-Jurgen Bathe, Ekkehard Ramm, Edward L Wilson (1975), Finite Element Formulations For Large Deformation Dynamic Analysis*, International Journal For Numerical Methods In Engineering, Vol. 9, 353-386.*
3. P.G. Gavali And M.S. Shah (2008), Earthquake Simulations Of Large Scale Structures Using Opensees Software On Grid And High Performance Computing In India*, The 14th World Conference on Earthquake Engineering October 12-17, Beijing, China.*

4. Sergey Choporov (2013), Parallel Computing Technologies in the Finite Element Method, *Third International Conference HPC-UA Ukraine, October*.
5. Silvia Mazzoni, Frank McKenna, Michael H. Scott, Gregory L. Fenves, et al. OpenSees Command Language Manual.
6. Frank McKenna, UC Berkeley, OpenSees Workshop, http://opensees.berkeley.edu.
7. Frank McKenna, UC Berkeley, OpenSees Parallel Workshop, http://opensees.berkeley.edu.
8. O.C.Zienkiewicz, R.L.Taylor, J.Z.Zhu. The Finite Element Method Its Basis and Fundamentals, *Elsevier, Sixth Edition.*
9. Robert D Cook, David S Malkus, Michael E Plesha. (2004). Concept and Applications of Finite Element Method, *Wiley, Fourth Edition.*
10. NGA West 2 PEER ground motion database.