International Journal of Research in Advanced Engineering and Technology
ISSN: 2455-0876; Impact Factor: RJIF 5.44
www.engineeringresearchjournal.com
Volume 3; Issue 1; January 2017; Page No. 54-56

A study on parallel and distributed processing in structural engineering and its applications

Vikas Kumar Pandey, Bipin Kumar Singh

Assistant Professor in Civil Engineering Department in NIMS University, Shobhanagar, Jaipur, Rajasthan, India

Abstract

Parallel processing has provided viable solution to many computational intensive numerical problems of science and engineering. Multiprocessor machines, consisting of hundreds to thousands processing elements interconnected and cooperating with each other for solving complex problems, are developed by number of countries. Efforts are still in progress to develop hardware, software, and computational tools for availing better computational power at lower cost. Generally various research organizations can access these facilities for developing their applications, but not many scientists and engineers can utilize these facilities due to number of complexities. The present study highlights the parallel and distributed processing in structural engineering and its applications.

Keywords: Parallel, Distributed, Engineering

1. Introduction

Substructure technique is inherently suitable for coarse grain implementation of structural analysis over distributed processing. In the implementation, generation of stiffness matrix and load vector for different substructures can be easily distributed to different computers and after computation of boundary displacements, calculation of internal displacements and element stresses of each substructure can be carried out in parallel without loss of any accuracy in final results.

Calculation of substructure stiffness matrix and load vector is quite time consuming. This is due to renumbering of internal and interface nodes for eliminating internal DOF using static condensation. It is clear from different applications that as the ratio of number of internal nodes to interface or boundary nodes increases, time to compute substructure stiffness matrix and load vector increases.

The computational efficiency depends on size of problem i.e. total number of unknowns. When distributed processing is implemented for small size problem, communication time gives more overhead to computational time and therefore computational efficiency is not that significant in such cases.

In distributed implementation on Web Dedip static load balancing i.e. distribution of computational load in beginning depending on computational power of computer is very simple. But dynamic load balancing i.e. modification of computational load among various computers during the process is difficult and hence it is not recommended here.

For distributed static finite element analysis, computational efficiency of 80 to 90% has been observed with different number of computers. Dynamic analysis of structure for calculation of natural frequency can be implemented as easily as static analysis over distributed computing environment using substructure technique and Web Dedip.

For the calculation of substructure stiffness and mass matrices either static condensation or dynamic condensation technique can be used. When regular plane frames are divided into more than three substructures static condensation gives natural frequency of first three modes almost similar to that of considering entire structure.

For distributed dynamic analysis an efficiency of about 60 - 70% was observed for the problem of 300 storied plane frame. The Web Dedip environment, which was developed originally at Space Application Centre, ISRO (Indian Space Research Organization) Ahmedabad primarily for image processing applications has been tuned in the present work to deal with distributed data processing. It has been shown that using Local Area Network, as available in most of the organizations, and Web Dedip environment structural engineer can implement his application over distributed computing environment.

Structural engineer without bothering much about technicalities of parallel processing has to simply identify the part of applications, which can run on different computers m parallel and accordingly, divide the application into small tasks. The programs are to be prepared for these tasks, and taking advantage of no message passing functions the configuration of application is to be done using Web Dedip.

The potential of using Web Dedip along with LAN for distributed structural engineering applications was explored in the present work through wide variety of large size problems.

In the beginning, feasibility of Web Dedip environment was studied through static analysis of microwave tower. Substructure technique was used to distribute the computations among various computers and finally the result of entire structure was obtained. After gaining confidence in use of Web Dedip, static finite element analysis was implemented over distributed computing environment.

Initially, small size problems of beam under pure bending and deep beam were tested using substructure technique. In substructure technique advantage of banded matrices was taken to reduce computational load and communication time. Subsequently, large size problems of a square plate with circular hole subjected to in plane forces (27980 DOF), annular plate (17175 DOF) and skew plate (27663 DOF) subjected to transverse loading were implemented over different number of computers of LAN. The results of analysis were verified with those available in literature and speedup was calculated. Also the effect of ratio of number of internal

nodes to interface nodes was studied with one large size problem.
Finally, nonlinear analysis which is quite time consuming process because of its iterative nature was attempted to have advantage of distributed processing.

1.1 Research Study

A problem of geometrical nonlinear analysis of clamped plate subjected to uniformly distributed load (20165 DOF) was implemented over different number of computers and computational efficiency was observed. For distributed implementation two approaches were used. In first approach the calculation of unbalanced load vector, linear, nonlinear and initial stress stiffness matrices was distributed to different computers.
While in other approach the calculation of tangent stiffness matrix and load vector for different substructures was distributed to different computers. In all applications discussed above, computer programs for different tasks were prepared using VC++ and configuration of application was done through Web Dedip. Most of the applications were implemented over network of Pentium IV computers having 256 MB RAM and running on WINDOWS-XP operating system at 1.8 GHz speed and which were connected through 100 Mbps ethernet network.
Using Local Area Network and Web Dedip environment, a wide variety of applications of structural engineering has been implemented on distributed computing environment. The applications included in the present work covers wide spectrum of structural engineering like analysis of skeletal and continuum structures in static, dynamic, linear and nonlinear domains in addition to problems of laminated composites and optimization.
Application of substructure technique in some of the difficult areas such as dynamic and nonlinear analysis of large size problems has been successfully demonstrated. For dynamic analysis it is shown that static condensation technique for substructure stiffness and mass matrices is adequate if structure is divided into more than three substructures.
Various alternative strategies for development of distributed application using Web Dedip have been discussed to give clear idea of different possibilities in structural analysis with a mention of advantages and limitations of each alternative. This may guide the future research work m proper direction.
Distributed dynamic analysis was implemented for plane frames in this study using direct stiffness approach. Distributed dynamic analysis of continuum structures using finite element method may be thought of for future application.
Dynamic and nonlinear finite element analysis of laminated composite plates may be attempted as an extension of present static linear analysis of laminated composite plates.
Material nonlinearity is another field where distributed processing may be as effective as in problems of geometrical nonlinearity and hence may be explored. Pre-processor may be developed for menu driven input and post-processor for graphical output to make the application more user friendly and attractive. In the present study substructurmg technique was used. The use of hierarchical substructuring particularly for very large structures may prove more beneficial.

1.2 Significance of the study

In parallel algorithms, independent computations are performed in parallel (i.e. executed concurrently). To achieve this parallelism, the algorithm is divided into a collection of independent tasks which can be executed in parallel and which communicate with each other during the execution of the algorithms.
Parallel algorithms can be characterized by the following three factors:

- Maximum amount of computation should be performed by a typical task before communication with other tasks or process;
- Inter-process communication topology, which is the geometric layout of the network of task modules;
- Executive control to schedule, enforce the interactions among the different task modules and ensure the correctness of the parallel algorithm.

The design of a parallel algorithm must deal with a host of complex problems, including data manipulation, storage allocation, memory interference, and in the case of parallel processors, inter-processor communication.
In general, the parallel numerical algorithms fall into two categories: reformulation (or restructuring) of serial algorithms into concurrent algorithms, and algorithms developed especially for parallel machines. Generally parallel numerical algorithms belong to the first category, i.e. decomposition of serial algorithms into concurrent tasks.
The second category includes the algorithms whose development was spurred by performance criteria for parallel processing which is referred to as uniquely parallel. There are two approaches for design of parallel algorithms: Divide and conquer, and reordering. The divide and conquer approach involves breaking a task up into smaller subtasks that can be treated independently.
The degree of independence of these tasks is a measure of the effectiveness of the numerical algorithm, since it determines the amount and frequency of communication and synchronization. Reordering refers to restructuring the computational domain and/or the sequence of operations in order to allow concurrent computations. For example, the assembly strategy, node-by-node or element-by-element assembly, may change the degree of parallelism that can be achieved in the solution of the resulting algebraic equations.

2. Conclusion

Several attempts have been made in recent years to exploit the potential of parallel processing machines in the solution of various structural analysis and design problems. These include finite element computations on SIMD vector computers, shared memory multiprocessors, and message-passing multi-computers.
For time dependent or transient problems, several parallel temporal integration techniques have been proposed for structural dynamics problems. Explicit schemes are well suited to both vector and parallel processing. This is especially true when a lumped mass matrix is used. A number of special strategies can be used to increase the degree of parallelism and/or vectorization in finite element computations.
These strategies are applications of the principle of divide and conquer, based on breaking a large (and/or complex) problem into a number of smaller (and/or simpler) subproblems which may be solved independently on distinct processors. The

degree of independence of the subproblems is a measure of the effectiveness of the algorithm since it determines the amount of frequency of communication and synchronization.

The computational strategies developed can be classified into four major categories and possible combinations: Domainwise strategies, Operator-splitting techniques, Column-wise (or row-wise) algorithms; and Node- or Element-wise algorithms. The last two categories of numerical algorithms allow only small granularity of the parallel tasks. On the other hand, the first two categories allow large granularity, which can improve the performance of the numerical algorithms.

3. References

1. Buyya R. High Performance Cluster Computing (Architecture and System), Prentice Hall Inc., New York, 1999.
2. Wilkinson B, Allen M. Parallel Programming Techniques and Applications using Networked Workstations and Parallel Computers, Prentice Hall, New Jersey, 1999.
3. Rajaraman V. Supercomputers, Wiley Eastern limited, 1993.
4. Adeli H. Parallel Processing in Computational Mechanics, Mareel Dekker Inc., New York, 1992.
5. Noor AK. New Computing Systems and Future High-Performance Computing Environment and Their Impact on Structural Analysis and Design, Computers and Structures, 1997; 64(1-4):1-30.
6. Topping BHV. Parallel and Distributed Processing for Computational Mechanics, Systems and Tools, Saxe-Coburg Publications, 1999.
7. Hwang K. Advanced Computer Architecture: Parallelism, Scalability & Programming, McGraw Hill Inc., New York, 1993.
8. Adeli H. Parallel and Distributed Processing in Structural Engineering, Proceeding of American Society of Civil Engineers, 1988.
9. Topping BHV, Khan AI. Parallel Finite Element Computations, Saxe-Coburg Publications, 1996.
10. Crichlow JM. An Introduction to Distributed and Parallel Computing, Prentice Hall of India Private Ltd., New Delhi, 1999.
11. litku S, Lesting J, Salama M. The Impact of Distributed Computing on Education, Computers and Structures, 1982; 15(2):149-156.
12. Adeli H, Vishnubhotla P. Parallel Processing, Microcomputers in Civil Engineering. 1987; 2:257-269.
13. Rao AR, Umesha PK, Loganathan K, Raman NV. Parallel Processing in Structural Analysis: A State-of-the-Art, Journal of Structural Engineering. 1990; 17(l):19-28.