Journal of Mechanical Science and Technology (KSME Int. J.), Vol. 19, No. 11, pp. 2007~2015, 2005

Parallel Process System and its Application to Steam Generator Structural Analysis

Yoon-Suk Chang, Han-Ok Ko, Jae-Boong Choi, Young-Jin Kim*
School of Mechanical Engineering, Sungkyunkwan University,
300 Chunchun-dong, Jangan-gu, Suwon, Kyonggi-do 440-746, Korea
Shinobu Yoshimura
Institute of Environmental Studies, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

A large-scale analysis to evaluate complex material and structural behaviors is one of interesting topic in diverse engineering and scientific fields. Also, the utilization of massively parallel processors has been a recent trend of high performance computing. The objective of this paper is to introduce a parallel process system which consists of general purpose finite element analysis solver as well as parallelized PC cluster. The later was constructed using eight processing elements and the former was developed adopting both hierarchical domain decomposition method and balancing domain decomposition method. Then, to verify the efficiency of the established system, it was applied for structural analysis of steam generator in nuclear power plant. Since the prototypal evaluation results agreed well to the corresponding reference solutions it is believed that, after reinforcement of PC cluster by increasing number of processing elements, the promising parallel process system can be utilized as a useful tool for advanced structural integrity evaluation.

Key Words : ADVENTURE_Solid, Balancing Domain Decomposition, Hierarchical Domain Decomposition, Parallel Process System, PC Cluster, Steam Generator

1. Introduction

A lot of general-purpose finite element method (FEM) and alternatives have been developed in the last three decades to evaluate mechanical and physical phenomena quantitatively. Recent progress of computational mechanics promoted the wide utilization of them in various fields and extended the practical environment from single processing element (PE) to several kinds of massively parallel processors (Yagawa et al., 1994; Ha et al., 2004). Also, in these days, the high performance scientific computing is being more and more important and indispensable trend. Because there are increasing demand for precise simulation of material and structural behaviors, a large-scale analysis is required to evaluate larger models with complex geometry than those usually adopted in the past.

As a typical case, with regard to nuclear power plant, aging issues and maintenance obsolescence of major components have to be solved for continued safe operation beyond the original licensed period (Kim et al., 2000). It will allow maintaining economic electric generating capacity that does not produce greenhouse gases or other pollutants, and is cheaper than building new generating capacity as well. In order to achieve these advantages, structural integrity evaluation is prerequisite to ensure that the effects of aging can be

---

* Corresponding Author,
E-mail : yjkim50@skku.edu
TEL : +82-31-290-5274; **FAX** : +82-31-290-5276
School of Mechanical Engineering, Sungkyunkwan University, 300 Chunchun-dong, Jangan-gu, Suwon, Kyonggi-do 440-746, Korea. (Manuscript **Received** April 13, 2005; **Revised** October 10, 2005)

adequately managed (Chang et al., 2001). Besides, the major components have been conservatively designed based on the results of repetitive two dimensional and partial three dimensional finite element analyses. So, if exactness and cost effectiveness are guaranteed, it is desirable to perform three dimensional large-scale analyses using full model considering realistic operating conditions.

The objective of this paper is to introduce a promising parallel process system which consists of general purpose finite element analysis solver as well as parallelized PC cluster. The later is constructed using eight PEs under Linux environment by incorporating a communication library. The former is developed adopting two types of representative algorithms : the one is hierarchical domain decomposition method (HDDM) and the other is balancing domain decomposition method (BDDM). To verify the efficiency of the parallel process system, it is applied for structural analysis of steam generator in nuclear power plant. Then, the analysis results are compared to the corresponding reference solutions. The details described in the remaining sections are as follows. In Section 2, configuration and characteristics of parallel process system are presented. In Section 3, backgrounds and algorithms of finite element analysis solver are reviewed. In Section 4, numerical experiments using the parallel process system and those results are illustrated and compared. Finally, in Section 5, concluding remarks are summarized.

2. Configuration of Parallel Process System

2.1 High performance PC cluster

The construction of parallel process system begins with selecting a suitable communication library which is responsible for control a distributed memory environment. The general function of it is to restrict the indiscriminate access by users while facilitate the maintenance and amendment of the code by administrative manager. Also, the library has technical functions such as initialization of parallel processor, sending and receiving messages among processors and identification of program type etc. In this research, the well known LAM/MPI (Massage Passing Interface) was adopted as the optimum communication library (William et al., 1999). It was contrived to operate even on heterogeneous clusters and, with TCP/IP, LAM imposes virtually no communication overhead.

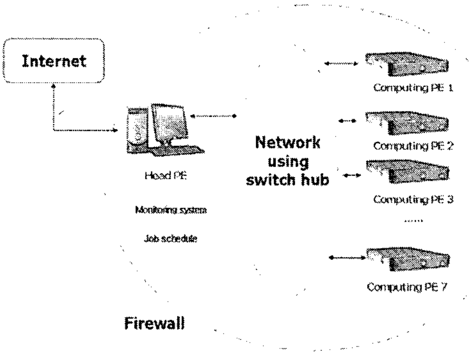
As a second stage, a Beowulf style PC cluster was built for large-scale analysis which can be defined as a cluster of commodity PEs. It works together through high speed network and allows the system to be viewed as a unified parallel computer system. Initially, two kinds of PC cluster were considered : the one is diskless cluster and the other is disk-based cluster. In case of diskless cluster, a hard disk (HDD) and operating system (OS) are installed only in a head PE. It is good for managing data and expansion of cluster although the assigned network loads are relatively high. In case of disk-based cluster, the HDD and OS are installed in each computer, and only small amounts of directories are shared. In this research, the disk-based cluster was selected since it has a benefit to reduce the network loads. Fig. 1 (a) shows a schematic diagram for configuration of PC cluster in which the relationship among head PE, hub and computing PEs are illustrated. Fig. 1(b) and Table 1 represent, respectively, the real arrangement and characteristics for PC cluster composed of eight PEs with equal specifications.

2.2 Features of ADVENTURE program

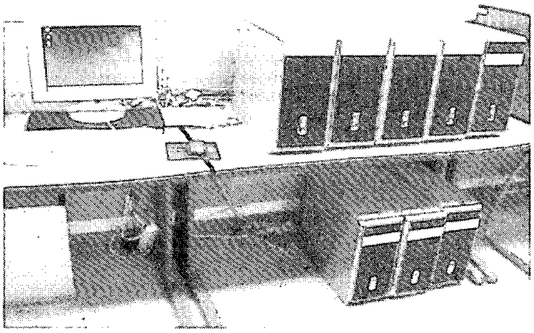
The ADVENTURE program consists of pre-, main- and post-processing parts that can be mounted on various kinds of parallel and distributed environments (Yoshimura et al., 2002). It employs massively parallel algorithms in order to handle a huge-scale efficiently. The program with standardized I/O format and libraries adopts module-based architecture to attain flexibility, portability, extensibility and maintainability of it. Each portion of the program requiring different knowledge, algorithm and programming style is packed into an independent module. Fig. 2 shows the schematics of the nineteen ADVENTURE

Table 1 Characteristics of PC cluster

Hardware		Software	
CPU	Pentium IV 2.8	OS	Redhat Linux 9.0
Memory	512 MByte×2 (PC3200)	Parallel library	LAM/MPI
HDD	120 GByte (7,200 rpm)	Services	NFS, rsh, ssh
Network	100 Mbps NIC	—	—
Hub	100 Mbps 8-port switch hub	—	—



(a) Schematic diagram



(b) Real arrangement

Fig. 1 Configuration of PC cluster

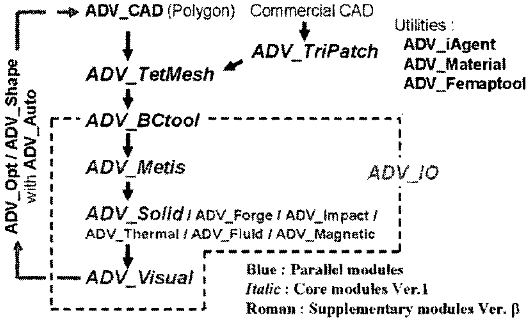


Fig. 2 Schematics of ADVENTURE modules

modules and their execution process.

The pre-processing modules include the surface patch generator which converts geometry model data into a collection of triangular surface patch data named as ADVENTURE_TriPatch, a tetrahedral mesh generator (Yagawa et al., 1995 ; Yoshimura et al., 1999) (ADVENTURE_TetMesh), an attachment tool of boundary conditions and material properties onto the mesh (ADVENTURE_BCtool) and a domain decomposer of a finite element model (ADVENTURE_Metis). The main processing modules, i.e., solvers include an implicit elastic-plastic analysis module named as ADVENTURE_Solid (Miyamura et al., 2002) which enables large-deformation and implicit dynamic analyses. A thermal conductive analysis module (ADVENTURE_Thermal), a thermal-fluid analysis module (ADVENTURE_Fluid), a magnetic analysis module (ADVENTURE_Magnetic), an explicit impact analysis module (ADVENTURE_Impact) and a rigid plastic analysis module (ADVENTURE_Forge) are also incorporated. The post processing module named as ADVENTURE_Visual is for parallel visualization of analysis results (Shoui et al., 2000). The common functions related to finite elements were coded as class libraries and named as libFEM.

3. Algorithm of Finite Element Analysis Solver

3.1 Overview of ADVENTURE_Solid

In domain decomposition method (DDM), an analysis model, i.e., a finite element mesh inclusive of boundary conditions and material properties is subdivided into a number of subdomains. One of the key technologies of the ADVENTURE_Solid is a HDDM which enables parallel finite element calculations on various kinds of computing environments. Basically, in the HDDM, force equivalence and continuity conditions among subdomains are satisfied through iterative calculations such as the conjugate gradient (CG) method. Therefore, it is indispensable to reduce the number of iterations by adopting some appropriate preconditioning technique especially for solving large-scale ill-conditioned problems. The Neumann-Neumann algorithm (N-N) (DeRoeck et al., 1991) is known as efficient domain decomposition preconditioner for structured subdomains. However, its convergence deteriorates with the increasing number of subdomains due to lack of a coarse space problem which takes care of global propagation of error. The BDDM based N-N algorithm proposed by Mandel (1993) shows that the equilibrium conditions for singular problems on subdomains result in simple and natural construction of a coarse space problem. It has been applied to solve various phenomena (Le Tallec et al., 1996) and there have been also several researches on parallelization of the BDDM while those were for medium scale ones up to one million degrees of freedoms (DOFs). In addition, it is necessary to consider the capacity of parallelized process system to get solution for large-scale problems as well. The features of HDDM and BDDM are recalled in the following subsections since those are adopted for numerical experiments in this paper.

3.2 Hierarchical domain decomposition method

The HDDM employs a hierarchical technique to implement the DDM on various parallel processors, in which a group of PEs is subdivided into the following two subgroups : one parent PE and many children PEs. At the same time, the analysis model is subdivided into a number of subdomains of which number can be much larger than that of the children PEs. The main role of parent PE is to manage all children PEs as well as to store mesh data of subdomains and iterate loops of the CG method. Each child PE performs finite element calculations of the subdomains received from parent PE and sends analyzed data back to the parent PE. According to the design concept of HDDM, most computation is assigned to the children PEs while most communication occurs between parent and children PEs. As a means to reduce the data communication, it is useful to assign all balance becomes static. This analysis mode is called as parallel processor mode (p-mode) while the original analysis mode is named hierarchical processor mode (h-mode).

In the substructural iterative methods, the interface DOFs are statically condensed by using a direct solver such as the skyline method. On the other hand, the interface DOFs are solved by using the CG method. At first, the stiffness matrix for each subdomain is partitioned as follows :

$$K^i = \begin{bmatrix} K^k_{II} & K^k_{IB} \\ K^k_{BI} & K^k_{BB} \end{bmatrix} \tag{1}$$

where, k denotes the subdomain number, I is the internal DOFs, B is the interface DOFs. Equilibrium equations for the internal DOFs and interface DOFs are written as Eqs. (2) and (3), respectively.

$$K^k_{II}\, u^k_I + K^k_{IB}\, u^k_B = p^k_I \tag{2}$$

$$\sum_{k=1}^{NDOM} K^k_{BI}\, u^k_I + \sum_{k=1}^{NDOM} K^k_{BB}\, u^k_B = \sum_{k=1}^{NDOM} p^k_B \tag{3}$$

where u denotes the nodal displacement vector, p is the equivalent nodal external force vector and NDOM is the number of subdomains. Eq. (2) can be solved independently in each subdomain for u^k_I as Eq. (4). Then, eliminated by substituting into Eq. (3) and becomes as Eq. (5).

$$u^k_I = (K^k_{II})^{-1}(p^k_I - K^k_{IB}\, u^k_B) \tag{4}$$

$$\begin{aligned} &\sum_{k=1}^{NDOM} \{K^k_{BB} - K^k_{BI}\,(K^k_{II})^{-1} K^k_{IB}\} u^k_B \\ &= \sum_{k=1}^{NDOM} \{p^k_B - K^k_{BI}\,(K^k_{II})^{-1} p^k_I\} \end{aligned} \tag{5}$$

Eq. (5) is a system of linear algebraic equations whose unknown vector is u^k_B and can be solved by CG method because the coefficient matrix of it is usually symmetric. The iteration is continued until the following residual vector, g, becomes sufficiently small.

$$g=\sum_{k=1}^{NDOM}\{K_{BB}^{k}-K_{BI}^{k}(K_{II}^{k})^{-1}K_{IB}^{k}\}u_{B}^{k} \\ -\sum_{k=1}^{NDOM}\{p_{B}^{k}-K_{BI}^{k}(K_{II}^{k})^{-1}p_{I}^{k}\} \tag{6}$$

For improvement of the convergence, due to expensive computational cost to get the coefficient matrix, the diagonal components of K_{BB}^{k} in Eq. (1) are used for the preconditioning by the diagonal scaling method.

3.3 Balancing domain decomposition method

The BDD algorithm is based on the DDM combined with a preconditioned iterative method. If the interior DOFs of local subdomain matrices are eliminated, the problem becomes interface problem and is to be solved by a preconditioned iterative method. There are various DDMs which contain a process of solving a reduced matrix using iterative methods. In general, at each step, those require to solve the independent auxiliary problem on the local subdomains represented as $Mz=r$. Here, M is a symmetric positive definite matrix called as preconditioner, z is preconditioned residual vector and r is a residual vector. The BDD preconditioned operator is given by (Mandel, 1993):

$$M^{-1}S=P+(I-P)\left(\sum_{i=1}^{N}T_i\right)(I-P)^{T} \tag{7}$$

where, S is Schur complement matrix, T_i is the local subdomain correction and $I-P$ is the coarse grid correction. If the residual vector has no components of the coarse space, Eq. (7) can be simplified as:

$$M^{-1}S=(I-P)\left(\sum_{i=1}^{N}T_i\right) \tag{8}$$

The original BDDM employs the N-N type algorithm as local subdomain correction with a two- level weighted sum of the inverses of S_i matrices (Mandel, 1993). To calculate the inverse of them, the Moore-Penrose pseudo-inverse or some regularization can be utilized since S_i matrices are typically singular. However, it takes high computational cost and the regularization is less accurate. So, the diagonal scaling preconditioner was chosen for Si as local subdomain correction. Also, for parallelizing, the BDD algorithm was decomposed into the following two phases. At first, the coarse grid operator S_0 is applied in subdomain-based blocks and then, its process is completely parallelized subdomain-wise. Secondly, the coarse grid correction is applied to solve a linear system equation whose coefficient matrix is derived from the coarse grid operator. In the original BDD preconditioner of Eq. (8), the coarse grid correction is implemented after local subdomain correction. However, in the current BDD preconditioner of Eq. (7), the coarse grid correction is applied to the CG residual vector before local subdomain correction.

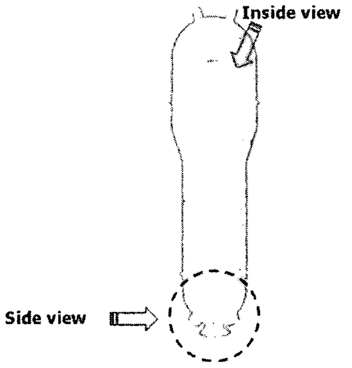
4. Numerical Experiments

4.1 Characteristics of steam generator

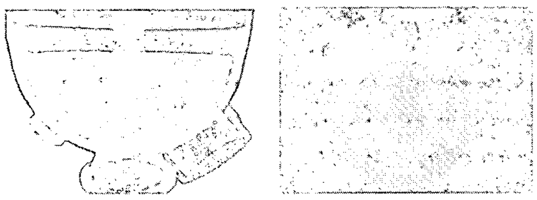
The steam generator is one of major component in pressurized water reactor (PWR). It is a representative heat exchanger that transfers heat from the primary reactor coolant system to the secondary-side feedwater in order to produce steam. In this research, a typical steam generator was chosen for numerical experiments. The overall height of it from bottom (primary inlet or outlet nozzle) to top (steam outlet nozzles) is about 20.8 m. The diameter of upper shell is about 5.1 m and that of lower shell is about 3.8 m. The thicknesses of shell are varies from 115 mm to 165 mm. For numerical experiments, the PWR steam generator was simplified by omitting internal subcomponents and modelled using INVENTOR (2000) and I-DEAS (2000). The shell of steam generator is made of SA508 Cl.3 carbon steel of which Young's modulus and Poisson's ratio are 210 GPa and 0.3, respectively.

4.2 Finite element analyses using ADVENTURE_Solid

Linear elastic stress analyses adopting both HDDM and BDDM algorithms were carried out for PWR steam generator. Two types of models were generated and analyzed using ADVENTURE_Solid mounted on PC cluster: the one is 6.6 million DOF model to get precise results and the other is 1 million DOF model to compare parallel performances. Fig. 3 illustrates the model with 6.6 million DOF mesh in accordance with different view points. It consists of 1,349,839 second-order tetrahedral elements and 2,206,664 nodes in which the each base mesh size was 63 mm and minimum mesh size was 5.77 mm. The model with 1 million DOF mesh consists of 1,349,839 first-order tetrahedral elements and 322,192 nodes. All of the models were generated by ADVENTURE_TriPatch and ADVENTURE_TetMesh.



(a) Front view



(b) Side view of lower shell and associated nozzles

(c) Inside view

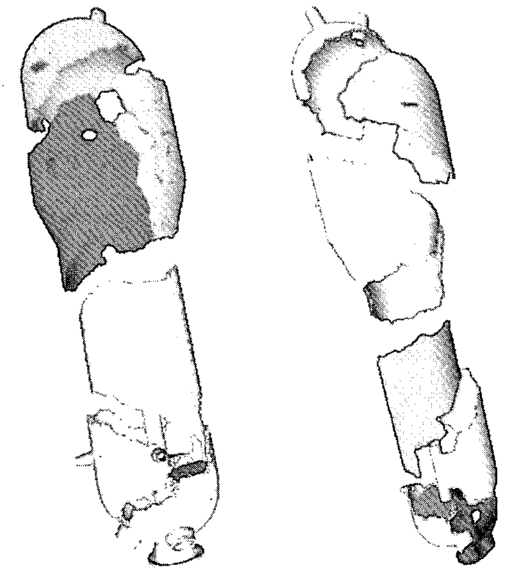
Fig. 3 Finite element mesh of steam generator for ADVENTURE_Solid analysis

Fig. 4 illustrates the model with 6.6 million DOF mesh decomposed by ADVENTURE_Metis as six and eight domains. To determine the number of elements in a domain (n), in this research, following simple equation was used.

$$n = N_{element} / N_{domain} \tag{9}$$

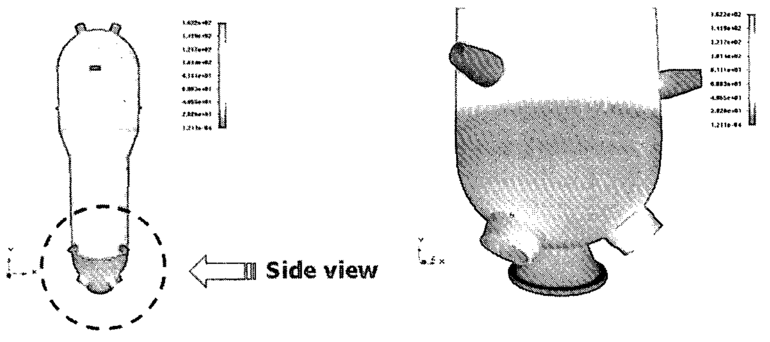
where, $N_{element}$ is the total number of elements and N_{domain} is the number of domain.

Fig. 5 depicts the corresponding von Mises stress distributions on the surface of steam gener-



(a) Six domains (b) Eight domains

Fig. 4 Domain decomposition of steam generator model



(a) Front view

(b) Side view of lower shell and associated nozzles

Fig. 5 Von Mises stress distributions on the surface of steam generator subjected to internal pressure

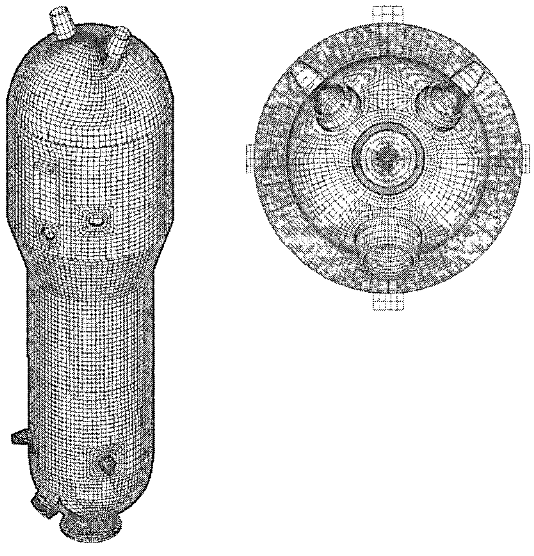
Table 2 Comparison of evaluation results between HDDM and BDDM

Parameter	HDDM (8 PEs)	BDDM (8 PEs)
Time	1447.73 sec	229.32 sec
Iteration	8750	49
Time/Iteration	0.165 sec	4.68 sec
Used memory per PE	143.76 MByte	268.03 MByte

ator under uniform internal pressure of 3 MPa. From the parallel performance point of view, in case of HDDM analysis, it took 1447.73 seconds (eight PEs condition, 1 million DOF model) through 8,750 CG iterations to attain a solution with a residual of 10^{-6} while, in case of BDDM analysis, 229.32 seconds (eight PEs condition, 1 million DOF model) was needed. Table 2 indicates the comparison results in accordance with the two analysis algorithms.

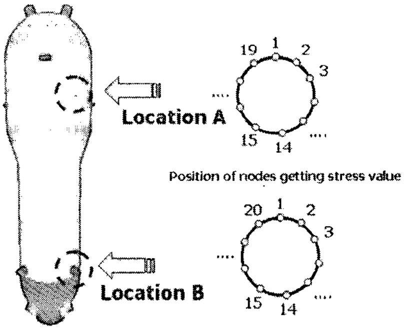
4.3 Comparison with reference solutions

The steam generator structural analysis results using ADVENTURE_Solid were compared to the corresponding reference solutions to verify the applicability of parallelized finite element analyses. At first, with regard to geometrically stabilized shell region, the simple equation of cylinder was used and obtained consistent results. Secondly, the comparable analysis was performed by using ANSYS (2004). Fig. 6 shows a finite element mesh of the steam generator which corresponds to that depicted in Fig. 3. It was composed of 58,479 eight node hexahedral elements (SOLID45 in ANSYS) and 77,590 nodes. Fig. 7 (a) illustrates the critical locations for comparison and Fig. 7(b) shows comparison of von Mises stress distributions obtained by ADVENTURE_Solid and ANSYS analyses. As shown in

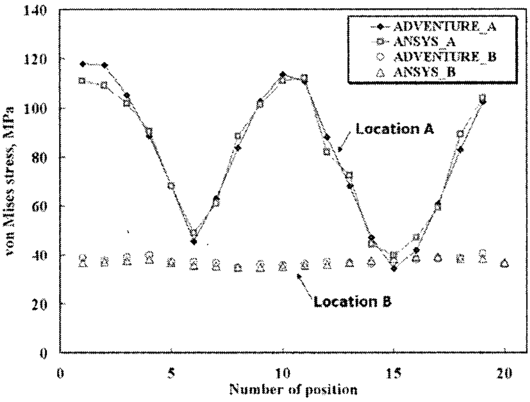


(a) Front view (b) Top view

Fig. 6 Finite element mesh of steam generator for ANSYS analysis



(a) Critical locations for comparison



(b) Von Mises stress distribution at critical locations

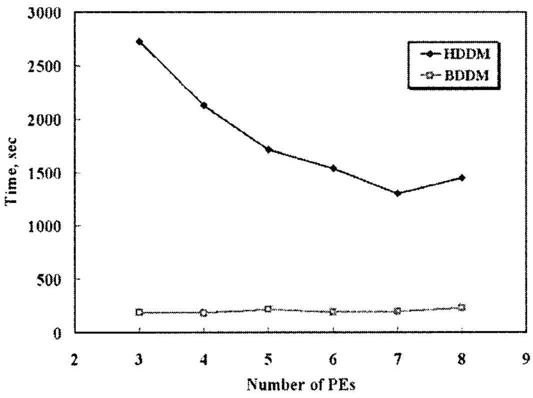
Fig. 7 Comparison of typical stress analysis results from ADVENTURE_Solid and ANSYS the figure, the two types of analyses results were almost same.

4.4 Quantification of parallel performance

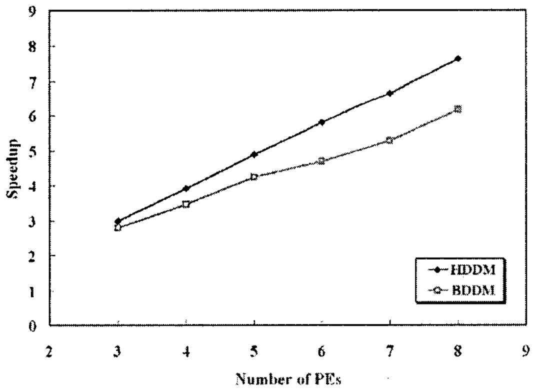
Any given piece of parallelized work will contain parts of the work that must be done serially by a single processor. Let $T(N)$ be the time required completing the task on N PEs, then, Amdahl's law (Joseph, 2004) can be expressed in terms of speedup as follows :

$$S(N)=\frac{T_s+T_p}{T_s+T_p/N} \tag{10}$$

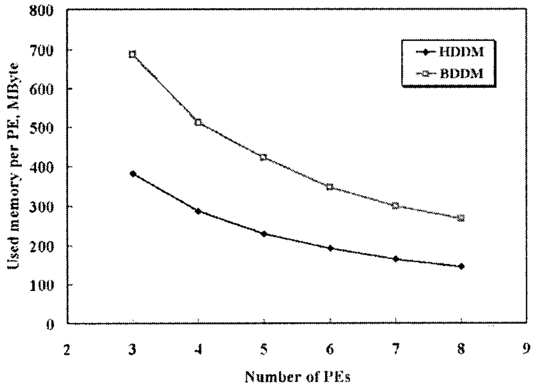
where, $S(N)$ represents the speedup, T_s is a serial portion and T_p is a parallelizable portion. Since the actual value of $S(N)$ is less than or equal to quantity of PEs, using Eq. (10), the parallel performance in accordance with increasing number of PEs can be predicted.



(a) Analysis time



(b) Speedup



(c) Used memory per PE

Fig. 8 Comparison of parallel performance

Fig. 8(a) shows comparison of consumed time according to number of PEs. When using three PEs, the analysis time adopting HDDM was about fourteen times longer than that of BDDM while six times longer for eight PEs. Fig. 8(b) represents comparison of speedup according to number of PEs. In cases of analyses adopting HDDM, the parallel efficiencies were almost 91%. However, in cases of analyses adopting BDDM, the values of speedup were less than those of HDDM and the differences were increased as the increase of PE numbers. Fig. 8(c) depicts comparison of used memory according to number of PEs. On the whole, the used memories by BDDM were about two times larger than those of HDDM.

5. Conclusions

This research was performed to construct a parallel process system for large-scale analysis of complex structure. Thereby, the following key findings have been derived.

(1) A promising parallel process system was developed. At this time, it consists of general purpose ADVENTURE solver operating under Linux environment as well as parallelized PC cluster composed of eight processing elements.

(2) Prototypal three dimensional full structural analyses of steam generator under internal pressure were carried out using the parallel process system. The evaluation results adopting 6.6 million DOF model agreed well to the corresponding reference solutions.

(3) Parallel performances were examined using

1 million DOF model. Thereby, it was proven that the efficiency of HDDM was higher than that of BDDM. Especially, the average speedup of HDDM was 91% while that of BDDM was 63%.

Acknowledgments

The authors are grateful for the support provided by a grant from IML at the Tokyo University and Safety and Structural Integrity Research Centre at the Sungkyunkwan University.

References

ANSYS Inc., ANSYS Version 8.0, 2004.

Autodesk Inc., INVENTOR Version 9.0, 2000.

Chang, Y. S., Jin, T. E., Song, T. H. and Jeong, I. S., 2001, "Lifetime Evaluation and Management for Class 1 Piping of Korean Lead Plant," 16th *International Conference on SMiRT*, CD-ROM.

DeRoeck, Y. H. and LeTallec, P., 1991, "Analysis and Test of a Local Domain Decomposition Preconditioner," 4th *International Symposium on Domain Decomposition Methods*, pp. 112~128.

Ha, M. Y, Kim, S. H, Kim, K. C and Son, Y. C., 2004, "A Numerical Study for the Three-Dimensional Fluid Flow Past Tube Banks and Comparison with PIV Experimental Data," *KSME International Journal*. Vol. 18, No. 12, pp. 2236~2249.

Joseph, D. S., 2004, *High Performance Linux Clusters with OSCAR, Rocks, openMosix, and MPI*, O'REILLY, Sebastopol, pp. 14~16.

Kim, J. S., Chang, Y. S., Jin, T. E. and Jeong, I. S., 2000, "Development Process of Aging Management Programs for Major SSCs of a PWR Plant," 12th *International Conference on Pacific Basin Nuclear Conference*, pp. 1301~1308.

Le Tallec, P. and Vidrascu, M., 1996, "Generalized Neumann-Neumann Preconditioners for Iterative Substructuring," 9th *International Symposium on Domain Decomposition Methods*, pp. 413~425.

Mandel, J., 1993, "Balancing Domain Decomposition," *Communications on Numerical Methods* in Engineering, Vol. 9, pp. 233~241.

Miyamura, T., Noguchi, H., Shioya, R., Yoshimura, S. and Yagawa, G., 2002, "Elastic-plastic Analysis of Nuclear Structures with Millions of DOFs using the Hierarchical Domain Decomposition Method," *Nuclear Engineering and Design*, Vol. 212, pp. 335~355.

SDRC Korea, I-DEAS Version 10.0, 2000.

Shoui, S., Yoshimura, S., Akiba, H., Ohyama, T. and Yagawa, G., 2000, "Parallel Visualization of Finite Element Solutions with Ten Million DOFs using PC Cluster," *Proceedings of European Congress on Computational Methods in Science and Engineering* (*ECCOMAS2000*), CD-ROM.

William, G., Ewing, L. and Anthony, S., 1999, *Using MPI*, MIT Press, Cambridge, pp. 289~328.

Yagawa, G. and Shioya, R., 1994, "Parallel Finite Elements on a Massively Parallel Computer with Domain Decomposition," *Computing Systems in Engineering*, Vol. 4, pp. 495~503.

Yagawa, G., Yoshimura, S. and Nakao, K., 1995, "Automatic Mesh Generation of Complex Geometries Based on Fuzzy Knowledge Processing and Computational Geometry," *Integrated Computer-Aided Engineering*, Vol.2, pp. 265~280.

Yoshimura, S., Hitta, H., Yagawa, G. and Akiba, H., 1999, "Parallel Automatic Mesh Generation of Nuclear Structures with Ten-million Nodes," 15th *International Conference on SMiRT*, Vol. II, pp. 21~28.

Yoshimura, S., Shioya, R., Noguchi, H. and Miyamura, T., 2002, "Advanced General-purpose Computational Mechanics System for Large-scale Analysis and Design," *Journal of Computational and Applied Mathematics*, Vol. 49, pp. 279~296.