 CMES, vol.67, no.3, pp.239-264, 2010

Parallel Computing Performance of Thermal-Structural Coupled Analysis in Parallel Computing Resource

Jong Keun Moon[1] **and Seung Jo Kim**[2]

Abstract: Large structural problems with high precision and complexity require a high-performance computation using the efficient parallel algorithm. The purpose of this paper is to present the parallel performance of thermal-structural coupled analysis tested on a parallel cluster system. In the coupled analysis, the heat transfer analysis is carried out, and then the structural analysis is performed based on temperature distribution. For the automatic and efficient connection of two parallel analysis modules, the several communication patterns were studied. The parallel performance was demonstrated for the sample and the real application problems, such as a laminated composite material by the DNS(Direct Numerical Simulation) approach and an aerospace launch vehicle model.

Keywords: Parallel Computation, Thermal Load Structural Analysis, Steady-state and Time Transient Heat Transfer Analysis, Coupled Analysis, Cluster Supercomputer.

1 Introduction

There is an increasing demand for high-performance computing that allows high-fidelity analysis of systems with enough precision and complexity. Therefore, the development of the high-performance computing algorithm and hardware system is needed to achieve these computational requirements.

In the last few years, the development of CPUs has focused on the increase of the number of cores in a single CPU due to the excessive heat generation caused by increasing the clock speed. In addition to the development trends of CPUs, the today's most supercomputers are developed by combining large amounts of CPU and compute node. Especially, the Cray XT5 supercomputer known as "Jaguar" which

[1] School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, Republic of Korea

[2] Flight Vehicle Research Center, Seoul National University, Seoul 151-742, Republic of Korea
AIAA Fellow, Fellow of IoP

is the world's fastest supercomputer is composed of 224,162 cores. However, the development of multi-core CPU architectures require programming engineers an additional work to implement the efficient parallel mechanism in their software. For the simulation and analysis that handle the system with enough precision and complexity, the high-performance computing is required, and the parallel computing mechanisms, such as MPI and OpenMP, can be a sort of solution to maximize the performance of a supercomputer system as well as a single computing resource. In this study, the efficient parallel computing algorithms for thermal-structural coupled analysis are proposed.

The effects of heat were treated in several fields, such as aerospace, power generation facilities, large structures, electronics, and etc[Karageorghis and Lesnic (2008); Noorzaei, Bayagoob, Abdulrazeg, Jaafar and Mohammed (2009)]. Likewise, heat transfer analysis was treated in conjunction with structural analysis because the thermal problems are affect the structures directly. Especially, thermal-structural coupled analysis has extensively been employed in the aerospace structures that lies in the extreme condition[Ohtake (1998)], such as high temperatures and the electronics that are integrated and miniaturized[Cheng, Yu and Chen (2005)] because deformation and stress of structures by thermal load can bring out serious damage. This features demonstrate that heat transfer analysis and structural analysis are coupled and inseparable.

Several studies on high-performance parallel structural analysis were already done in the previous work [Kim, Kim and Lee (2005); Kim, Lee and Kim (2002)]. The steady-state and time transient heat transfer analysis modules are newly developed based on the heat transfer equations [Bathe (2002); Cook, Malkus, Plesha and Witt (2002); Sladek, Sladek, Tan and Atluri (2008)] and fully uses parallel computation with the efficient data communication patterns. Section 2 briefly describes the finite element discretization used to analyze the heat transfer problem with conduction and convection, in which the temperatures at the boundaries of a solid body are given. In section 3, a flowchart of parallel thermal-structural coupled analysis is described in detail and the management of temperature-dependent material in structural analysis is discussed.

For structural analysis and solid mechanics problems, the direct solvers were used generally because of the numerical robustness that guarantees a solution within an estimated time. Therefore, most commercial finite element software for structural analysis employs a direct solver as their main equation solver. In this study, a parallel multi-frontal solver[Kim, Lee and Kim (2002), Duff and Reid (1973)] among the variety of direct solvers was used, and is described briefly in section 4.1. In section 4.2, two types of communication patterns for parallel computation are introduced. First, we studied and compared two types of communication patterns to

correlate two analyses, the structural analysis and heat transfer analysis. Second, the right hand side(RHS) term at time in time transient heat transfer analysis must be updated. In this process, the RHS term is stored separately in each process for parallel computation. Therefore, an efficient communication pattern is needed to have a good parallel computing performance, and is discussed in section 4.2.

In section 5, we showed the parallel computing performance for the coupled analysis by using the IPSAP(Internet Parallel Structural Analysis Program) code that is the general finite element software and can be found in the website (http://ipsap.snu.ac.kr). As a computing resource, a small size but high-performance sever cluster system was used to confirm the performance of the structural analysis, time-transient heat transfer analysis and coupled analysis for sample models and real structure models; the characteristics of a laminated composite material by the DNS(Direct Numerical Simulation) approach[Kim, Lee, Yeo, Kim and Cho (2002)] and an aerospace launch vehicle with a high pressure and high temperature in launching stage.

2 Finite Element Formulation

There is a variety of commercial finite element softwares and IPSAP codes for the structural analysis with a thermal load and heat transfer analysis. This section presents a general finite element formulation for the structural analysis with a thermal load and heat transfer analysis for an arbitrary solid body.

2.1 Structural Analysis with Thermal Load

The general problem for the structural analysis with a thermal load is considered. The total strain energy can be expressed as

$$U = \int_V \frac{1}{2} \{\varepsilon\}^T \{\sigma\} dV, \tag{1}$$

and the stress vector is also related to the strain vector by a constitutive relationship that expresses the elastic properties of the body. The general form of this relationship for linear elastic materials can be written as

$$\{\sigma\} = [D] (\{\varepsilon\} - \{\varepsilon_T\}), \tag{2}$$

where $[D]$ is a matrix of elastic moduli, $\{\varepsilon_T\}$ is a vector of initial strains due to thermal expansion. The thermal strain vector for a solid body made from an orthotropic material, $\{\alpha\}$ and temperature change, ΔT was defined.

$$\{\varepsilon_T\} = \{\alpha\} \Delta T, \tag{3}$$

The thermal coefficients α_{ij} are the components of a second-order tensor. Therefore, they transform like strain components. In general, only nonzero components of the thermal expansion tensor are $\alpha_{11} = \alpha_1$, $\alpha_{22} = \alpha_2$, and $\alpha_{33} = \alpha_3$. All other components are zero. Therefore, the transformation relations can be written from the transformation of stress[Reddy (2004)]. For the plate element, the only nonzero transformed thermal coefficients of expansion were α_{xx}, α_{yy}, and α_{xy}.

$$\begin{aligned}
\alpha_{xx} &= \alpha_{11}\cos^2\theta + \alpha_{22}\sin^2\theta \\
\alpha_{yy} &= \alpha_{11}\sin^2\theta + \alpha_{22}\cos^2\theta \\
\alpha_{xy} &= (\alpha_{11} - \alpha_{22})\sin\theta\cos\theta \\
\alpha_{xz} &= 0,\ \alpha_{yz} = 0,\ \alpha_{zz} = \alpha_{33}
\end{aligned} \tag{4}$$

2.2 Heat Transfer Analysis

In general, heat transfer analysis treats solids, liquids, and gases, with applications including heat exchangers, engines, and chemical processes. Heat flows within a solid body through conduction. Heat is transferred to or from a solid body by the convection of an adjacent fluid and by radiation. In addition, heat may be generated internally from the resistance to an electric current and flow externally across a boundary. The thermal problem in this paper is to determine the temperature field in a solid considering conduction and convection heat transfer. The general finite element equation can be derived as follows:

$$\begin{aligned}
&[C]\left\{\dot{T}\right\} + [K_T][T] = \{R_T\} \quad \textit{where} \quad [K_T] = [K] + [H] \\
&\{R_T\} = \{R_B\} + \{R_h\} + \{R_Q\}
\end{aligned} \tag{5}$$

Here, $[H]$ and $\{R_h\}$ mean the convection heat transfer across a surface.

The time transient method described in this paper is the generalized trapezoid rule. This rule is similar to Newmark's method used for structural dynamics analysis. The rule begins with the assumption that two temperature states $\{T\}_i$ at time t_i and $\{T\}_{i+1}$ at time t_{i+1} are related by

$$\{T\}_{i+1} = \{T\}_i + \Delta t\left[(1-\beta)\dot{T}_i + \beta\dot{T}_{i+1}\right] \tag{6}$$

Eq. 6 contains a factor β that is chosen by the user. The time derivative term can be eliminated using Eqs. 5 and 6 to give Eq. 7 as follows:

$$\left(\frac{1}{\Delta t}[C] + \beta[K_T]\right)\underline{T}_{i+1} = \left[\frac{1}{\Delta t}[C] - (1-\beta)[K_T]\right]\underline{T}_i + (1-\beta)\{R_T\}_i + \beta\{R_T\}_{i+1} \tag{7}$$

If $\beta \geq \frac{1}{2}$, the numerical analysis is unconditionally stable. According to this rule, $\beta \geq \frac{1}{2}$ was used in this paper. And if Δt is constant, a matrix that multiplies $\{T\}_{i+1}$ is generated and processed for equation-solving only once. Therefore, the equation set is then solved repeatedly for a sequence of right-hand sides.

3 Coupling of Structural Analysis and Heat Transfer Analysis

The structural analysis and heat transfer analysis modules were implemented by the finite element formulation in section 2, and finally a coupled analysis module was developed based on these two individual analysis modules. Fig. 1 shows the procedure for parallel thermal-structural coupled analysis. In Fig. 1, boxes with dashed lines represent the data communication routine in time transient heat transfer analysis. Coupled analysis and various data communication strategies will be examined in detail in chapter 4.

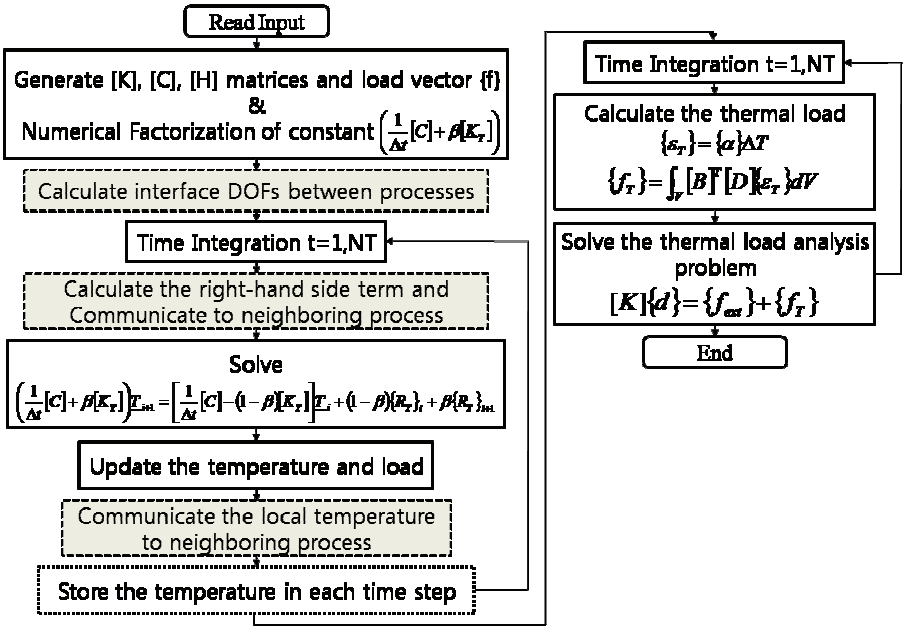


Figure 1: Flowchart of Parallel Thermal-Structural Coupling Analysis

As mentioned above, heat transfer analysis and structural analysis were developed independently of the original. This section represents the works to connect these two individual analysis modules efficiently.

The key work for coupling of these two analyses is a treatment of the temperature

result in each time step and is indicated in a box with dotted lines in Fig. 1. This is not needed in serial computation, but is required only in parallel computation. In parallel computation, the temperature obtained in each time step is distributed and stored to each process. However, the temperature stored in the other processes is needed to calculate the thermal load in the process of structural analysis. Therefore, an efficient data communication pattern should be constructed to minimize the computation time and will be described in detail in section 4.2.

The time transient heat transfer analysis is performed first, and the transient temperature outputs from this analysis are saved in the physical memory for the subsequent structural analysis[Zhu and Chao (2002)].

4 Parallel Implementation

4.1 Parallel Multi-Frontal Algorithm

The multi-frontal method is known as the best direct solution method in terms of computation, memory, and parallel efficiency requirements. It is extremely efficient in both a serial and parallel computing environment.

Duff and Reid(1973) first introduced the concept of the multi-frontal method as a generalization of the frontal method of Irons[Irons (1970)]. The key point of the frontal method is to eliminate the DOF(Degree of Freedom) during assembling every element. If one DOF is assembled completely, it can be eliminated directly because the coefficients and load vector related to an assembled DOF is not changed after assembly. A small dense matrix assembled in the current step is known as the 'frontal matrix'.

The multi-frontal method eliminates the multiple frontal matrices instead of a single front, as shown in Fig. 2. The entire problem is divided into several sub-domains. Each internal DOFs is carried out on a single front and merged with the neighbor sub-domain to eliminate the interface DOFs between the two sub-domains. Each frontal matrix is factorized as a dense matrix by Cholesky factorization.

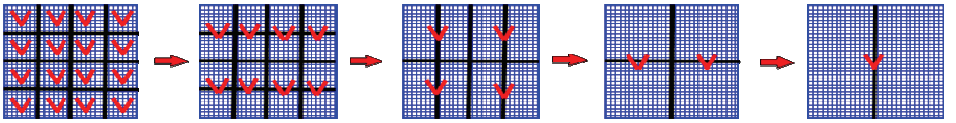


Figure 2: Illustration of multi-frontal factorization

A multi-frontal solver solves a problem in four steps. The first step is domain partitioning to solve efficiently. The second step is a symbolic factorization to construct

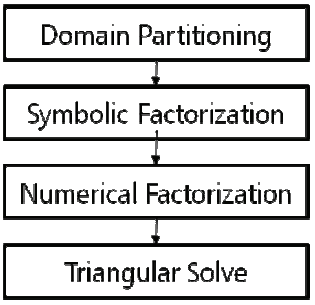


Figure 3: Procedure of a Multi-frontal Solver

an elimination tree. The third step is numerical factorization via an elimination tree i.e. a floating point operation. The last step is a triangular solve, which is a forward elimination and backward substitution. Fig. 3 represents the procedure in the multi-frontal solution method.

A multi-frontal solver is implemented to characterize finite element problems. Therefore, it does not require explicit assembly of a global stiffness matrix, whereas other sparse direct solvers require an assembling global stiffness matrix.

4.2 Parallel Process in Thermal-Structural Coupled Analysis

By Kim, Lee and Kim(2002) and Kim, Kim and Lee(2005), static structural analysis and vibration analysis modules based on a parallel multi-frontal solver were developed and the serial and parallel computing performance was already confirmed by several studies. The thermal load part was newly implemented on structural analysis, and the static and time transient heat transfer analysis modules were also developed based on a parallel multi-frontal solver. Finally, a thermal-structural coupled analysis module was constructed, and the efficient parallelization of this module is the core of this research, particularly in terms of data communication.

In a parallel computing process, the nodes and elements are separated, and stored in each physical process. Therefore, the data that is stored in a single process can be used by other processes during thermal-structural coupled analysis. A study was carried out to maximize the efficiency of data communication in coupled analysis and parallel time transient heat transfer analysis to improve the entire parallel computing performance.

4.2.1 Efficient Communication Patterns for Distributed Results

The temperature obtained by parallel heat transfer analysis is stored separately in each physical process. To calculate the loads due to temperature for structural analysis in coupled analysis, the separated temperature should be merged using the appropriate method. In this paper, two communication patterns were employed.

In step 1, the partial temperature data in each physical process was combined into the first physical process. In step 2, the total temperature data was transferred from the first process sequentially, as shown in Fig. 4. The advantage of this communication pattern is that the code can be implemented very easily and simply. However, this pattern is allowed only in limited computing environments. In step 2, the total temperature data was transferred from process 0 to process 1, and then transferred simultaneously from process 0 and 1 to process 2 and 3,. Therefore, it can be used only in the computing process in multiples of 2.

Fig. 5 shows the second communication pattern that the transfer traffic is minimized by dealing with information on the transfer in advance. In this pattern, only the data related to the non-interface DOFs in elements holding the interface DOFs are a target to transfer to the neighboring processes. For this reason, the number and content of data transmitted from Proc A to Proc B are different from that transmitted from Proc B to Proc A. In order to accomplish these tasks efficiently, the amount and location of data to communicate with the neighboring processes must be computed as a first step.

This study evaluated the performance of two communication patterns in coupled analysis for time transient heat transfer analysis with 30 time steps. As a test model, two 2-D topology models consisting of approximately 3 million and 5 million DOFs were used. As a computing environment, a Windows parallel cluster system with 8 computing nodes and 64 cores was used. A detailed description of this system will be given in the next section. Fig. 6 compares the performance of the two communication patterns.

The communication time for pattern 1 is proportional to the base-2 logarithmic scale, as depicted in Fig. 4. In particular, this pattern is limited to the number of processes in multiples of 2. As shown in Fig. 6, the communication time will increase proportionally depending on the increase in the number of DOFs. On the other hand, the communication time for pattern 2 is much lower than that for pattern 1 and does not increase with increasing number of processes and DOFs. This is because it was already performed to minimize the volume of communication data.

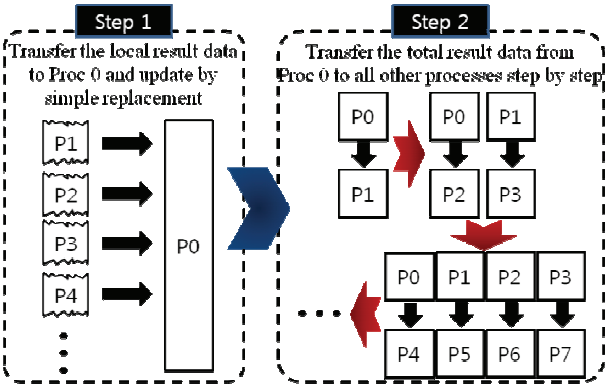


Figure 4: Comm. Pattern 1: Combining into One, then Transferring to All

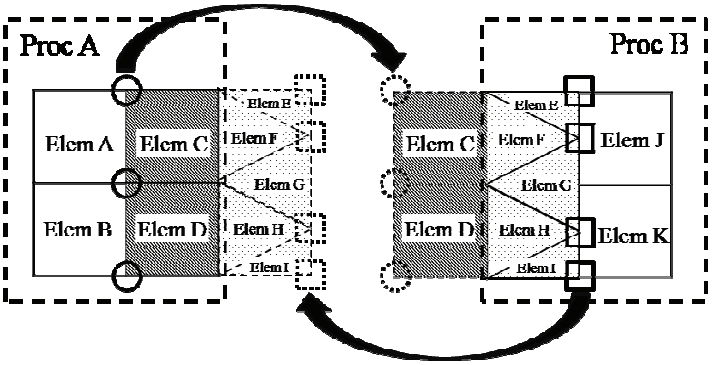
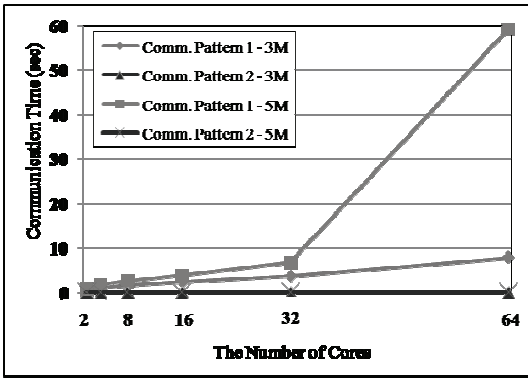


Figure 5: Comm. Pattern 2: Data Communication of Non-interface DOFs between the Neighboring Processes

4.2.2 Parallel Process of Right-hand Side Term in Time Transient Analysis

In parallel computing, the right-hand side term of Eq. 7, which is updated in each time step, is incomplete because each process does not provide information on the total nodes and elements, which must be supplemented by the updated term in other processes.

Considering that the temperature in the previous time step is known, the right-hand side term of Eq. 7 is calculated using capacity matrix [C], conductivity matrix [K], convection matrix [H] and temperature in the previous time step. The right-hand side term calculated in each physical process is communicated to the neighboring process and updated by simple addition. By removing the unnecessary data com-



# of Cores	Comm. Pattern 1 - 3M DOFs	Comm. Pattern 2 - 3M DOFs
2	4.51E-01	1.00E-02
8	1.55E+00	1.50E-02
32	9.70E+00	6.00E-03
64	7.92E+00	1.20E-02
# of Cores	Comm. Pattern 1 - 5M DOFs	Comm. Pattern 2 - 5M DOFs
2	7.72E-01	1.10E-02
8	2.67E+00	8.00E-03
32	6.80E+00	4.00E-03
64	5.93E+01	1.00E-03

Figure 6: Performance Comparison between Comm. Patterns 1 & 2

munication in this process, it may be possible to maximize the parallel computing performance. Therefore, this pattern is implemented by applying the efficient data communication routine shown in Fig. 5. Fig. 7 shows the data communication pattern transferring the computed right-hand side term for the interface DOFs. For example, considering that the circles denote the DOFs shared in both Proc 1 and Proc 2, the data for the circles only are communicated between Proc 1 and Proc 2, and other processes are performed in a similar manner.

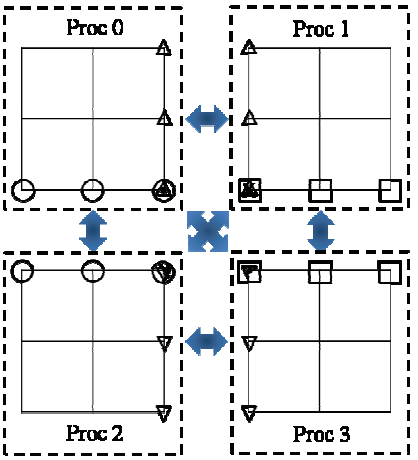


Figure 7: Communication Pattern of the Right-hand Side Term for the Interface DOFs

5 Computing Performance and Applications

5.1 Parallel Cluster System

As a test bed for computing performance testing of IPSAP, a parallel cluster system with 8 computing nodes and 16 Intel Core2Quad CPUs was used as shown in Fig. 8 and Table 1. Each computational node consists of two Intel Core2Quad Processors (2.93, 3.2 GHz) that share 64 GB of RAM. Each node is connected through an Infiniband 10Gbps switch. The network performance is one of the important commodities of cluster performance because every node in the cluster system communicates through the network. Infinitband can support data processing at 2.5GBytes per second.

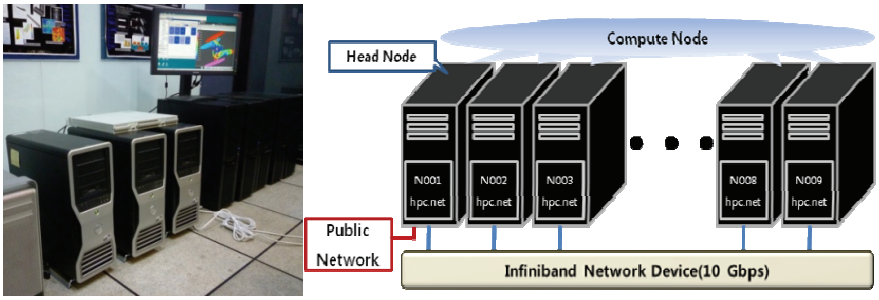


Figure 8: Parallel Server Cluster System

A Linux system has been usually used for high performance computing. However, because most PCs utilize Windows operating systems, it is not easy for a general user to control a Linux computing system. Windows HPC systems offer a user-friendly graphical interface that can overcome the limitations caused by the difficulty in using a Linux cluster system. Microsoft has developed a windows server version for high performance computing. The newest Windows version of high performance computing is Windows HPC server 2008. With HPC pack 2008, which includes a job scheduler and MPI program on Windows, it is possible to operate the cluster system easily because most functions and installation processes are controlled based on the graphical interface and instructions.

Each node in this cluster system is composed of 2 CPUs(8 cores). When performing parallel computation in only 1 compute node, it can be more powerful to employ parallel computation using OpenMP instead of MPI. The machine topology has a significant impact on the performance of all parallelization strategies[Rabenseifner,

Table 1: Hardware Summary of the Parallel Cluster System

	Unit Node	Total System
CPU	3 x Intel Core 2 Quad 3.2 GHz (4 Core X 2) 5 x Intel Core 2 Quad 2.93 GHz (4 Core X 2)	64 Cores
RAM	8 x DDR2 ECC 64GB (4GB X 16)	512 GB
HDD	SATA 500 GB	4 TB
Network	Infiniband (10Gbps) for MPI network	
OS / Compiler	Windows HPC Server 2008 / Visual Studio 2008	
MPI	MS-MPI in HPC Pack	

Hager and Jost (2009)]. Therefore, It should be confirmed which parallelization topology shows better computing performance between hybrid MPI/OpenMP and pure MPI on the cluster system. Simple structural analysis problems with 2-dimensional plate elements and 3-dimensional solid elements were solved in the system. Figs. 9 and 10 compare the resulting factorization time because the factorization time accounts for almost 80∼90 % of the total computing time.

The computing time according to the parallelization topologies for 2-D and 3-D mesh topologies are similar. However, hybrid MPI/OpenMP shows a slightly better result than the pure MPI, and 1-node/8-processes case in a pure MPI for a 3-D mesh could not be solved due to the lack of memory. In particular, in terms of memory usage, the 1-node/1-process case in hybrid MPI/OpenMP for 3-D mesh used approximately 37.8 GB of the total 64 GB of physical memory. On the other hand, as mentioned above, the 1-node/8-processes case in a pure MPI for a 3-D mesh used more than 64 GB of physical memory. Therefore, the computing time was meaningless and the analysis was stopped. It should be clear from the performance comparison that it is important to use the optimal number of OpenMP threads per MPI process for a given problem and system. In this study, the parallel computing performance was tested and verified using the hybrid MPI/OpenMP.

5.2 Verification of Parallel Computing Performance

5.2.1 Parallel Performance of Structural Analysis Module

The parallel performance of structural analysis was evaluated using both a scalability and speed-up test. All tests were performed using the abovementioned parallel cluster system. Simple problems with 2-D and 3-D elements were used in all tests. The factorization performances were used as a comparison target for the aforementioned reasons in Section 5.1.

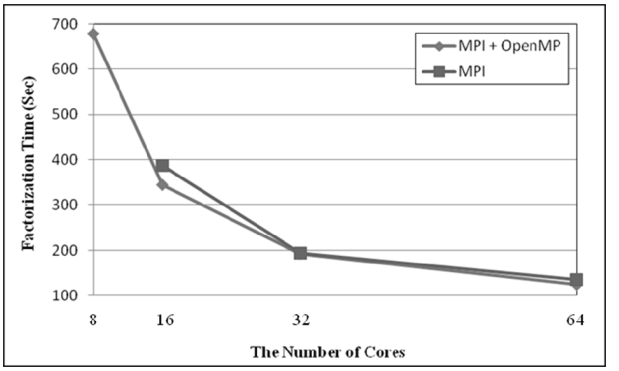


Figure 9: Factorization Time Comparison for 2-D Mesh Topology between Hybrid MPI/OpenMP and Pure MPI

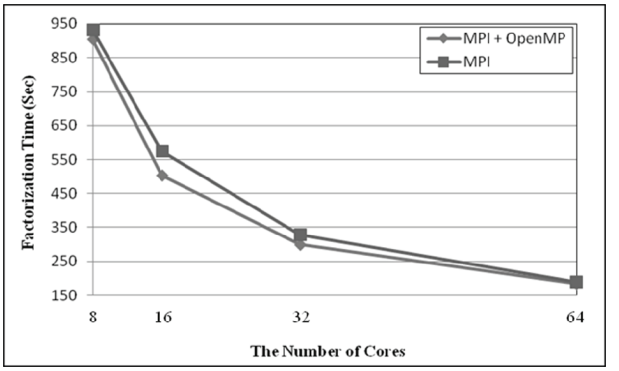


Figure 10: Factorization Time Comparison for a 3-D Mesh Topology between Hybrid MPI/OpenMP and Pure MPI

First, the scalability test was estimated. For the 2-D mesh topology, the size of the mesh assigned to each processor was kept constant. A similar number of finite element meshes was assigned to each processor. For the 3-D mesh topology, the total operation count in each processor was kept relatively constant [Kim, Kim and Lee (2005)]. The scaled speed-up is defined by the following formula:

$$S_c = \frac{P_{n-c}}{P_{1-1}} \tag{8}$$

where P_{1-1} is the performance of the 1-node/1-core computing environment in a pure MPI, i.e. without a multi-threading process, and P_{n-c} is the performance for

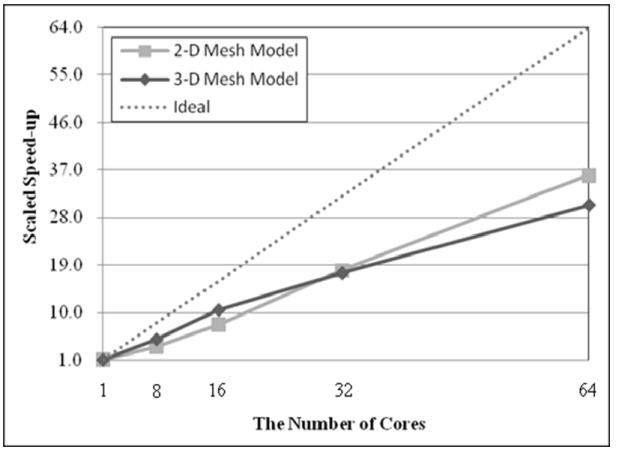


Figure 11: Scaled Speed-up for the 2-D and 3-D Mesh Topology

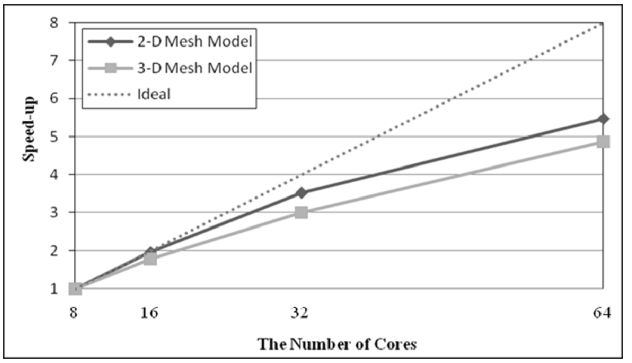


Figure 12: Speed-up Results for the 2-D and 3-D Mesh Topology

the n-node/c-cores environment in MPI/OpenMP, i.e. in each node, only 1 physical process job is assigned, but c-cores are used by the multi-thread process. Fig. 11 and Tables 2 and 3 present the scaled speed-up results for the 2-D and 3-D topology model. As shown in the results, the performance was scaled up continuously to 64 cores and was quite similar to the other previous studies for other computing environments, i.e., Linux cluster system and Unix system[Kim, Kim and Byun (2007)].

Second, the speed-up test was performed. The problem size increases in proportional to the number of cores used. The test problems were the structured quadrangle mesh model (1588x1588) for the 2-D mesh topology and the structured hexa-

Table 2: Scalability Test Results for the 2-D Mesh

# of Nodes & Cores	Mesh Topology	Number of Unknowns	Operation Count	Performance (Gflops)	Scaled Speed-up
1-1	375_375	848,256	2.26E+11	6.1	1.0
1-8	1060_1060	6,754,326	5.03E+12	20.7	3.4
2-16	1500_1500	13,518,006	1.44E+13	47.0	7.7
4-32	2120_2120	26,991,846	4.04E+13	110.4	18.0
8-64	3000_3000	54,036,006	1.15E+14	221.0	36.0

Table 3: Scalability Test Results for the 3-D Mesh

# of Nodes & Cores	Mesh Topology	Number of Unknowns	Operation Count	Performance (Gflops)	Scaled Speed-up
1-1	44_44_44	273,375	1.358E+12	9.4	1.0
1-8	62_62_62	750,141	1.067E+13	45.7	4.9
2-16	70_70_70	1,073,733	2.281E+13	97.8	10.5
4-32	80_80_80	1,594,323	4.910E+13	163.4	17.5
8-64	88_88_88	2,114,907	8.695E+13	283.4	30.3

hedral mesh model (80x80x80) for the 3-D mesh. The largest problem size that can be solved on single computing node was determined approximately. The speed-up is defined as follows:

$$S_n = \frac{P_n}{P_1} \tag{9}$$

where P_1 is the performance for a 1-node computing environment with OpenMP, i.e. using 8-cores just by executing 1 process, and P_n is the performance for the n-node environment in MPI/OpenMP. The speed-up results for both 2-D and 3-D models are listed and compared in Fig. 12 and Tables 4 and 5. As shown in Fig. 12, the relative performance was better in the 2-D mesh topology but the absolute performance was better in the 3-D mesh topology. In the 2-D mesh, the computing performance in the 8-nodes/64-cores case was approximately 5.5 times that in the 1-node/8-cores case. However, in the case of the 3-D mesh, the performance in the 8-nodes/64-cores case reached 265 Gflops.

Table 4: Speed-up Results for the 2-D Mesh Topology (Quad 1588_1588)

# of Nodes & Cores	Factorization Time (sec)	Performance (Gflops)	Speed-up
1-8	678.3	25.6	1.0
2-16	344.6	50.4	2.0
4-32	192.2	90.3	3.5
8-64	123.9	140.2	5.5

Table 5: Speed-up Results for the 3-D Mesh Topology (Hexa 80_80_80)

# of Nodes & Cores	Factorization Time (sec)	Performance (Gflops)	Speed-up
1-8	902.9	54.4	1.0
2-16	503.4	97.5	1.8
4-32	299.8	163.8	3.0
8-64	185.3	265.0	4.9

5.2.2 *Parallel Performance of Heat Transfer Analysis*

In the heat transfer analysis, the linear condition was considered. Hence, after constructing and factorizing the conductivity matrix and convection matrix for all elements, as shown in Fig. 1, substitution of the updated RHS term was performed repeatedly in each time step. The factorization algorithm in the heat transfer analysis is the same as the one in structural analysis. Therefore, the parallel computing efficiency is also the same. The point considered in this section is the parallel computing performance for substitutions in each time step. In static analysis, substitution does not take a great deal of time compared to factorization, but in time transient analysis, its parallel performance should be considered as an important factor because the substitution time is proportional to the number of time steps.

It was confirmed that the time transient module implemented in this study has a good parallel computing performance by two types of size models and number of time steps. As a test problem, simple structured quadrangle mesh models composed of 5 million and 10 million DOFs were used. The speed-up for time transient analysis is defined as follows:

$$S_n = \frac{T_1}{T_n} \tag{10}$$

where T_1 is the computing time in time transient analysis for multiple time steps in the 1-node computing environment with MPI/OpenMP, and T_n is the comput-

ing time in the n-node computing environment. Fig. 13 and Table 6 compare the speed-up results with respect to the size and number of time steps for time transient analysis. Fig. 13 shows that efficient performance in a parallel computing environment had been obtained. The same speed-up performance results in two models with different problem sizes suggest that the computing time needed for time transient analysis is proportional to the number of time steps.

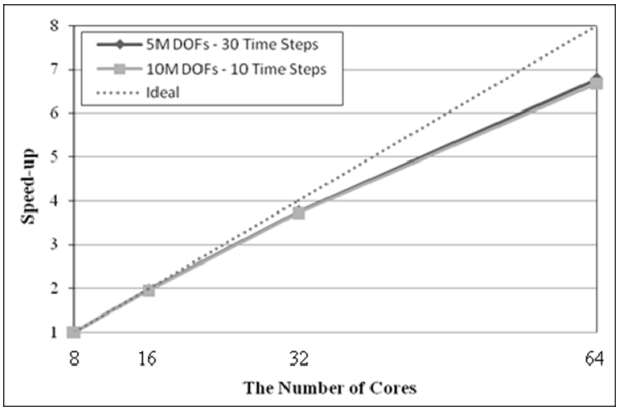


Figure 13: Speed-up Results w.r.t. Size and Number of Time Steps for Time Transient Analysis

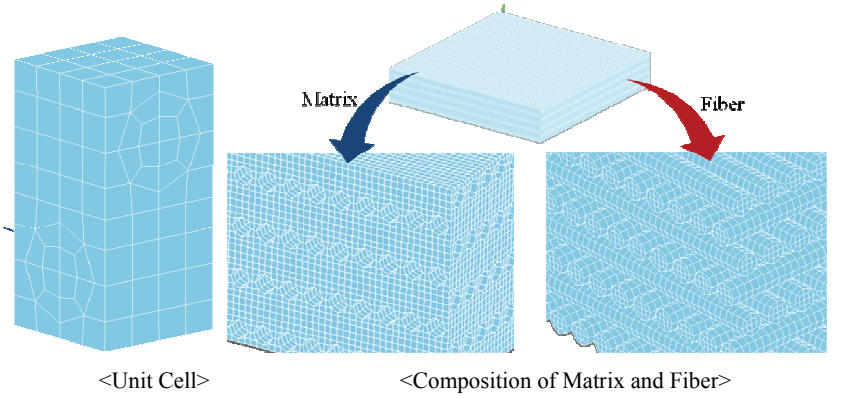


Figure 14: The DNS Model of the Composite Structure

Table 6: Speed-up Results w.r.t. Size and Number of Time Steps

# of Nodes & Cores	5M DOFs – 30 Time Steps		10M DOFs – 10 Time Steps	
	Time (sec)	Speed-up	Time (sec)	Speed-up
1-8	1007.2	1	659.4	1
2-16	512.9	1.96	337.9	1.95
4-32	268.7	3.75	177.4	3.72
8-64	148.6	6.78	98.8	6.68

5.2.3 Real Application Models for Coupled Analysis

The above section describes the high performance for structural analysis and time transient heat transfer analysis in a parallel computing environment. The same analysis module is applied to a coupled analysis module. The computing performance is also the same at those of the analysis modules. In this section, real structural models are applied to show the availability and computing performance of coupled analysis module. One model is a laminated composite model by the DNS approach and the other is an aerospace launch vehicle that experiences high pressures and temperatures during launch. The details of both models are as follows.

Due to the high stiffness and damage tolerance, Metal Matrix Composites(MMC) are commonly used in aircraft components and space systems. With these superior material characteristics, engineers can strengthen a structure with a minimum increase in mass. However, delamination between the matrix and fiber can occur due to a combination of two different constituents. Therefore, in order to utilize the laminate composite fully, it is important to predict the local behavior, such as failure of the fiber of composites. The homogenization method, which models all constituents together with their averaged material properties, is the traditional approach for predicting the macroscopic behavior of composite structures. However, there are limitations in finding damage or the initiation of delamination using the macroscopic approach.

Therefore, the microscopic approach, i.e. DNS(Direct Numerical Simulation) was used for these composites, as shown in Fig. 14. It is possible to predict the mechanical behavior on the microscopic level. A unit cell method containing the fiber and matrix was used to generate a DNS model, as shown in Fig. 14. A full model of the composite structure was made by copying and translating the unit cell.

In the microscopic approach of the DNS model, it is essential to increase the size of the finite element model of composite structures. Therefore, using parallel computing, the structural analysis problem of the laminate composites could be solved with

more than 3 million DOFs on the 8-nodes/64-cores cluster system. Using a unit cell shown in Fig. 14, the composite layers were stacked in the following sequence: 0°, 90°, 0°, 90°, 0°, 90°, 0° and 90° and 995,328 hexahedral solid elements were used for the model. The total number of DOFs was 3,083,715 in terms of the structural analysis model. Silicon carbide (SiC)[Bednarcyk and Arnold (2002); Oksanen, Scholz and Fabbri (1997)] and a Titanium alloy (Ti-15-3)[Bednarcyk and Arnold (2002); Ti-15-3] was used as the fiber material and matrix, respectively. This composite material is used widely because of the good mechanical behavior in high temperature aerospace applications. Table 7 lists the mechanical and thermal properties of the fiber and matrix.

Table 7: Mechanical and Thermal Properties of the Composite Constituents

Material	$E(GPa)$	ν	$\rho(kg/m^3)$	$\alpha(\times 10^{-6}/^{\circ}C)$	$k(W/mK)$
SiC	393.0	0.25	3150.0	3.56	17.80
Ti-15-3	89.5	0.3	4760.6	8.33	7.613

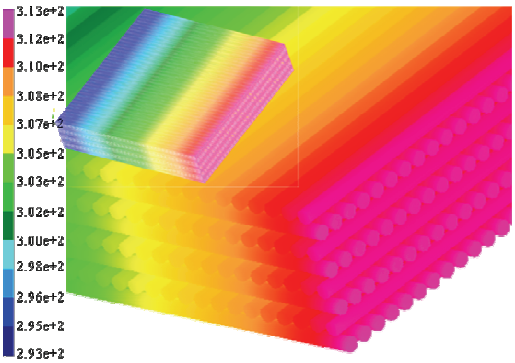


Figure 15: Temperature Distributions of the Laminated Composites

The problem was defined as a boundary condition, in which the temperature and displacement at one of the side faces is fixed, and as a load condition, where there is constant heat applied to the other side. The total elapsed time for thermal-structural coupled analysis was 814.8 seconds. Figs. 15-18 show the temperature, displacement and stress distributions. As a visualization toolkit, we used a pre/post software, DIAMOND[Moon, Kim, Kim, Park, Jang and Kim (2009)] that is developed as a general purpose GUI(Graphic User Interface) of IPSAP. As mentioned above, the mechanical behavior that could not be determined using the macroscopic ap-

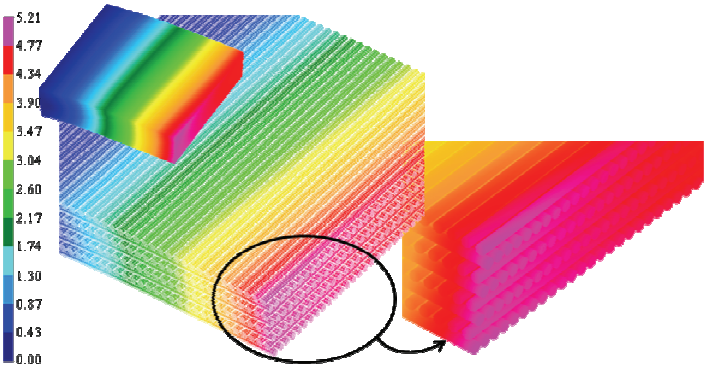


Figure 16: Displacement Distributions of the Laminated Composites

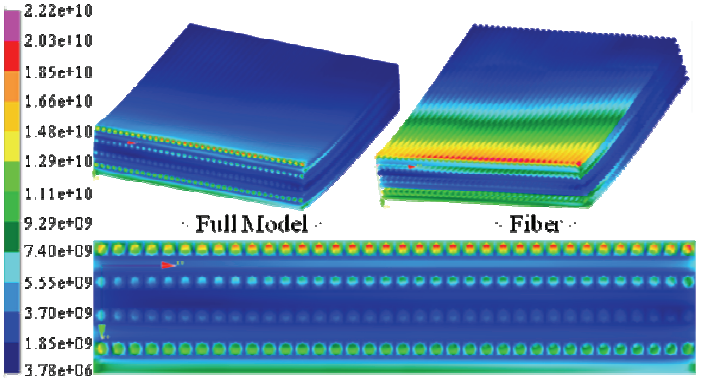


Figure 17: Von-Mises Stress Distributions of the Laminated Composites

proach could be predicte successfully using the DNS approach. As shown in Figs. 17 and 18, the stress is concentrated more on the fiber than on the matrix.

In addition, the analysis result was compared with the result solved by commercial software for verifying the accuracy of the result. MSC.Nastran was used as a commercial software that is said to be representative in the computational mechanical engineering. . As a test model, a DNS model composed of 98,304 elements and 315,315 DOFs was used modelling smaller than the above problem because it is not possible to solve the above problem in a serial compute node.

The temperature, displacement and stress results are compared in Table. 8 and Figs. 19-21. Table. 8 shows almost the same results for two softwares. Maximum temperatures in two softwares are completely same, the difference in maximum

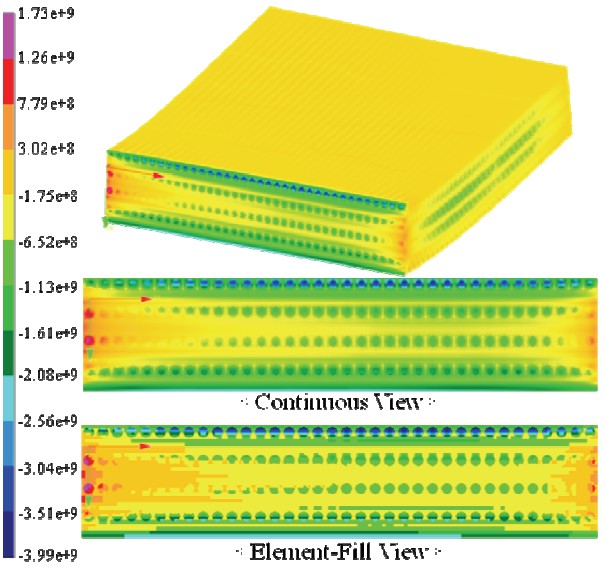


Figure 18: Distributions of the Laminated Composites

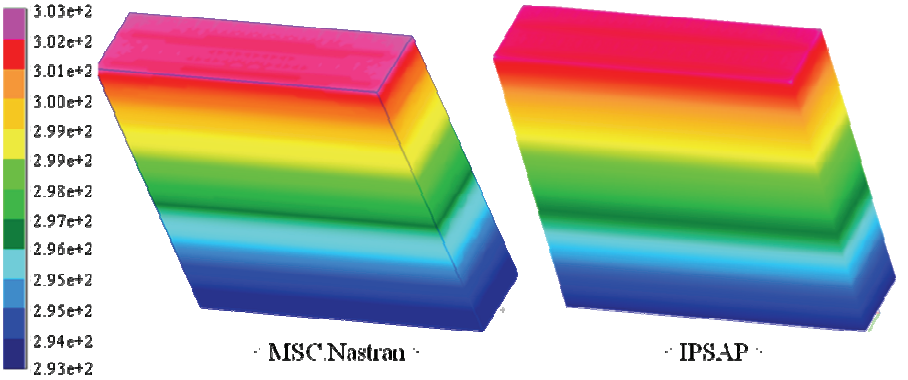


Figure 19: Comparison of Temperature Results for the Laminated Composite Model

displacement is 0.07% and the difference in maximum stress is 0.43%. Also, Figs. 19-21 show the similar distributions in two softwares. In Fig 21, the stress distribution looks like a little different, but it occurs because the post processing algorithm for the stress distribution of DIAMOND/IPSAP is significantly different from that of MSC.Nastran.

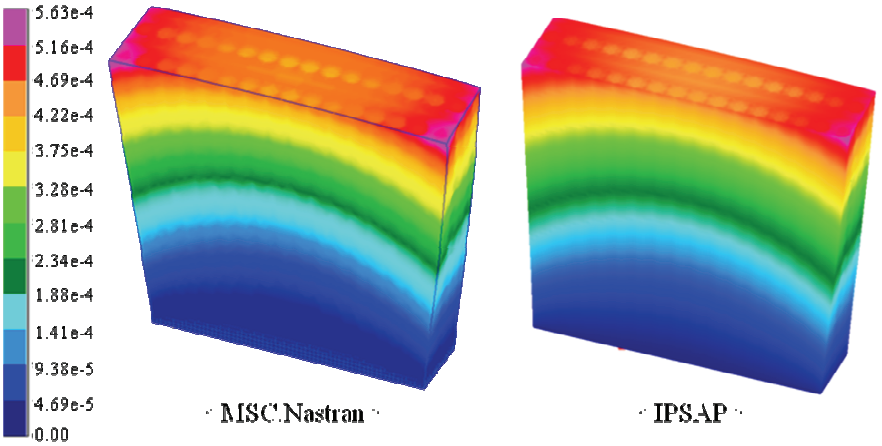


Figure 20: Comparison of Displacement Results for the Laminated Composite Model

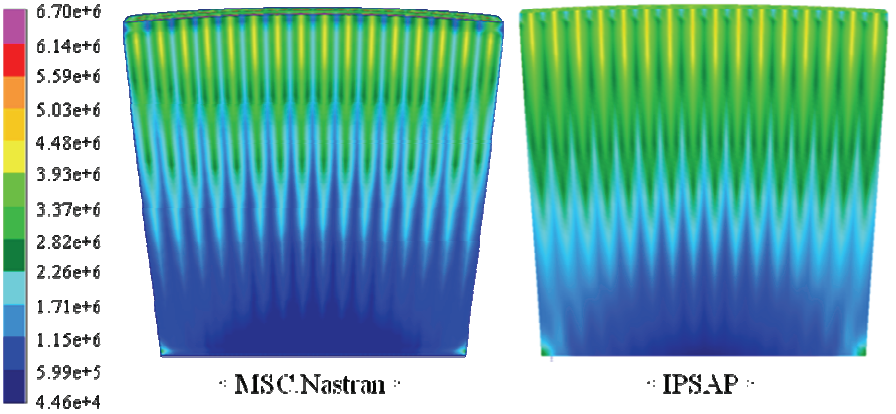


Figure 21: Comparison of Von-Mises Stress Results for the Laminated Composite Model

An aerospace launch vehicle was examined as a second real and practical application. Launch vehicles including satellites are exposed to harsh conditions in the launching stage, such as high pressures and temperatures. In addition, it is not possible to repair the structures after launch. Therefore, it is important to guarantee high structural reliability and safety, and account for as many scenarios in its launch stage as possible. Not even the slightest error should be allowed in the

Table 8: Result Comparisons between IPSAP and MSC.Nastran

	Max. Temperature	Max. Displacement	Von-Mises Stress	
			Max.	Min.
IPSAP	302.64	5.627e-4 (Diff. : 0.07%)	7.658e+6 (Diff. : 0.43%)	4.464e+4
MSC.Nastran	302.64	5.631e-4	7.691e+6	4.551e+4

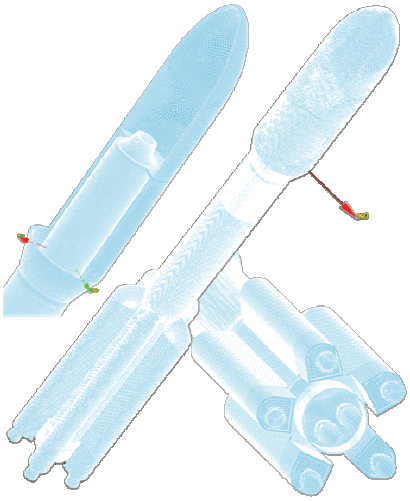


Figure 22: FE Model of Launch Vehicle, ATLAS V500

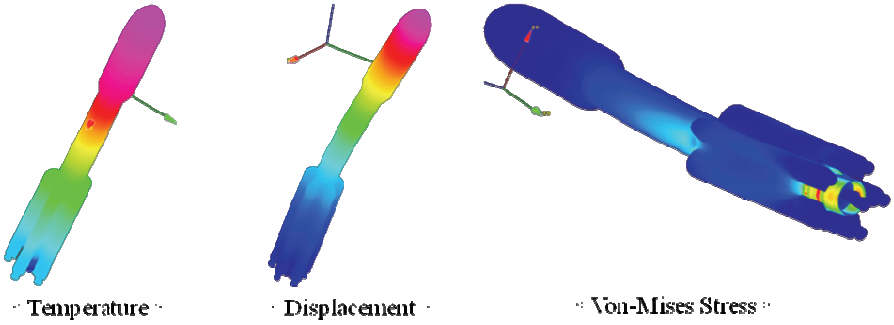


Figure 23: Temperature, Displacement and Von-Mises Stress Results of ATLAS V500 by Thermal-Structural Coupled Analysis

design due to their delicate nature and the astronomical cost of these vehicles themselves. As shown in Fig. 22, the ATLAS V500 launch vehicle modeled virtually is composed of 255,550 hexahedral solid elements, 400,517 nodes, and 1,201,551 DOFs in terms of the structural analysis model.

It was assumed that as a boundary condition, the temperature and displacement at the nodes around the nozzle would be fixed and excessive heat at the head of the payload fairing and rocket boosters occurs as a load condition. The total time elapsed for thermal-structural coupled analysis was 89.6. Fig. 23 shows the temperature, displacement and stress distributions.

6 Summary

This study examined the parallel performance of thermal-structural coupled analysis tested on a parallel cluster system. The steady-state and time transient heat transfer analysis codes were implemented on the structural analysis code, IPSAP based on a parallel multi-frontal solver. The results showed good scalability and speed-up performance for parallel computation. High computing performance was observed for several communication patterns considered in parallel coupled analysis and parallel time transient heat transfer analysis. Large-scale analyses could be performed successfully for real practical structural models, i.e. a laminated composite material modeled by DNS and an aerospace launch vehicle.

The electric components and finished products developed in industry are increasing in terms of their intensity and complexity. Due to the intensive characteristics, several problems induced by excessive heat can affect their structures. An analysis module is needed to solve the high-fidelity models effectively to handle a complex structure similar to the actual shape. Therefore, it is believed that the analysis reported in this paper will benefit a wide range of engineers and industries.

Acknowledgement: This research was supported by NSL(National Space Lab) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (2009-0092052) and the Flight Vehicle Research Center at Seoul National University.

References:

Bathe, Klaus-Jürgen (2002): *Finite Element Procedures*, Prentice Hall, New Jersey.

Bednarcyk, Brett A.; Arnold, Steve M. (2002): Fully Coupled Micro/Macro Deformation, Damage, and Failure Prediction for SiC/Ti-15-3 Laminates, *Journal of Aerospace Engineering*, Vol. 15, No. 3, pp.74-83.

Cheng, H. C.; Yu, C. Y.; Chen, W. H. (2005): An Effective Thermal-mechanical Modeling Methodology for Large-scale Area Array Typed Packages, *CMES: Computer Modeling in Engineering & Sciences*, Vol.7, No.1, pp.1-17.

Cook, Robert D.; Malkus, David S.; Plesha, Michael E.; Witt, Robert J. (2002), *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons.

Duff, S.; Reid, J. K. (1973): The multifrontal solution of indefinite sparse symmetric linear equations, *ACM Transactions on Mathematical Software*, Vol. 9, pp. 302-325.

Irons, B. M. (1970): A frontal solution program for finite element analysis, *International Journal for Numerical Methods in Engineering*, Vol. 2, pp. 5-32.

Karageorghis, A.; Lesnic, D. (2008): Steady-state nonlinear heat conduction in composite materials using the method of fundamental solutions, *Comput. Methods Appl. Mech. Engrg.*, Vol. 197, pp. 3122–3137.

Kim, Jeong Ho; Kim, Seung Jo; Lee, Chang Sung (2005): High-performance Domain-Wise Parallel Direct Solver for Large-scale Structural Analysis, *AIAA Journal*, Vol. 43, No. 3, pp. 662-670.

Kim, Seung Jo; Kim, Min Ki; Byun, Wanil (2007): Parallel Performance Analysis of Multi-Frontal Solver in Large Scale Finite Element Analysis, *International Conference on Computational & Experimental Engineering and Sciences(ICCES) 2007*, pp. 1758, Miami, Florida.

Kim, Seung Jo; Lee, Chang Sung; Kim, Jeong Ho Kim (2002): Large-Scale Structural Analysis by parallel multifrontal solver through Internet Based PCs, *AIAA Journal*, vol. 40, No.2, pp. 359-367.

Kim, Seung Jo; Lee, Chang Sung; Yeo, Hea Jin; Kim, Jeong Ho; Cho, Jin Yeon (2002): Direct Numerical Simulation of Composite Structures, *Journal of COMPOSITE MATERIALS*, Vol.36 No. 24, pp. 2765-2785.

Moon, Jong Keun; Kim, Jong Bum; Kim, Sang Min; Park, Kuk Jin; Jang, Chae Kyu; Kim, Seung Jo (2009): Performance Comparison of Efficient Finite Element System, DIAMOND/IPSAP in Parallel Environment, *2009 Asia-Pacific International Symposium on Aerospace Technology*, pp. 84-90, Gifu, Japan.

Noorzaei, J.; Bayagoob, K. H.; Abdulrazeg, A. A.; Jaafar, M. S.; Mohammed, T. A. (2009): Three Dimensional Nonlinear Temperature and Structural Analysis of Roller compacted Concrete Dam, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 47, No. 1, pp. 43-60.

Ohtake, Kunihiko (1998): Thermal analysis of the thermal protection system for the re-entry vehicle, *Comput. Methods Appl. Mech. Engrg.*, Vol. 151, pp. 301-310.

Oksanen, M.; Scholz, R.; Fabbri, L. (1997): On the longitudinal thermal diffu-

sivity of SiC-based fibres, *Journal of Materials Science Letters*, Vol. 16, No. 13, pp. 1092-1094.

Ti-15-3, URL:http://www.ulbrich.com/files/titaniumarticle.pdf.

Sladek, J.; Sladek, V.; Tan, C.L.; Atluri, S.N. (2008): Analysis of Transient Heat Conduction in 3D Anisotropic Functionally Graded Solids, by the MLPG Method, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 32, No. 3, pp. 161-174.

Rabenseifner, Rolf; Hager, Georg; Jost, Gabriele (2009): Hybrid MPI/OpenMP Parallel Programming on Clusters of Multi-Core SMP Nodes, *2009 Parallel, Distributed and Network-based Processing*, pp.427-436.

Reddy, J. N. (2004): *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis*, CRC Press.

Zhu, X. K.; Chao, Y. J.(2002): Effects of temperature-dependent material properties on welding simulation, *Computers & Structures*, Vol. 80, Issue 11, pp. 967-976.