

부구조기법을 이용한 PC level 분산구조해석법

Distributed Structural Analysis Method on Network of PCs using Substructuring Techniques

박 효 선* 박 성 무** 성 창 원*** 김 재 홍****
Park, Hyo Seon Park, Sung Moo Sung, Chang Won Kim, Jae hong

ABSTRACT

Efficiency of design process for large scale structures highly depends on the development of efficient structural analysis and structural response control algorithms because a successful design involves a number of structural analysis based on iterative structural response control process. In this paper, distributed structural analysis model on multiple personal computers connected by ethernet network is presented. To reduce communication cost required in the process of analysis, substructuring techniques are adopted to evenly distribute computational loads on each processor. With its applications on structural analysis of plane frame structures, performance of the proposed computational model are presented in detail.

1. 서 론

최근 국내에서의 건물의 초고층화 현상은 초고층 건물 구조분야의 연구활성화 및 실용화를 유도하고 있으며, 이로인해 초고층 구조분야에 대한 연구가 활발히 진행되고 있다. 이와같은 초고층 분야의 연구들과 기존의 중·고층 건물구조에 관련된 연구와의 구조해석적 측면에서 나타나는 차이점은 초고층 구조물이 막대한 수의 자유도와 부재를 가지고 있다는 것이다. 이러한 자유도수와 부재수 때문에 기존의 구조해석법은 해석에 있어서 장시간을 요하게 되며, 특히 내진해석이나 최적설계에 있어서 이 현상은 더욱 심각한 실정이다.^{1),2),3)} 그러므로 초고층 구조물의 해석에 있어서 자유도수와 부재수에 관련된 구조해석적 측면에서의 문제점들을 해결하기 위한 실용적이고도 신속한 해석법의 개발이 시급한 실정이다.

그러므로 본 논문에서는 여러 대의 개인용 PC를 이용한 가상의 고성능 구조해석용 전산 모델을 MPI^{4),5)}를 이용하여 개발하고 기존 부구조기법⁶⁾을 이용하여 대형 구조물의 부재 및 자유도를 네트워크에 연결된 각 개인용 PC에 할당하여 분산계산하는 분산구조해석법 (distributed structural analysis method)을 개발하여 이의 적용성을 분석한다.

* 영남대학교 건축공학과 조교수
** 영남대학교 건축공학과 부교수
*** 영남대학교 건축공학과 박사수료
**** 영남대학교 건축공학과 석사과정

2. 부구조기법을 이용한 분산구조해석법

현재까지 사용되고 있는 구조해석 프로그램들은 PC 한 대에서 모든 구조해석 작업이 수행된다. 따라서 구조물의 자유도수가 커지고 부재수가 증가하면, 구조해석에 소요되는 시간이 비선형으로 증가하게 된다. 이것을 해결하기 위해 구조해석 프로그램을 네트워크로 연결된 다수의 개인용 PC에서 나누어 수행시킴으로서 구조해석에 소요되는 시간을 줄일 수 있다. 이러한 분산구조 해석방법은 여러 가지가 있으나, 본 연구에서는 전체 구조물을 여러개의 작은 부구조물로 나누어 해석하는 부구조 기법을 사용한다.^{7),8),9)} 이 부구조기법은 개인용 PC의 고성능화로 인하여 그 자체로서는 경쟁력을 잃고 있는 실정이므로, 이 부구조기법을 네트워크로 연결된 여러대의 개인용 PC에서 운용하는 분산구조해석법에 적용함으로써 자유도수가 많은 대형 및 초고층 구조물을 해석하는데 소요되는 해석시간을 단축할 수 있다.

2-1 부구조기법

대형 및 초고층 구조물을 여러개의 작은 구조물로 분할하여 각각의 구조물을 독립적으로 해석하고 그 다음에 전체 구조물의 해석을 진행시킨다. 분할된 n번째 구조물에 대한 평형 방정식은 다음과 같다.

$$\begin{bmatrix} \mathbf{K}_{bb}^{(n)} & \mathbf{K}_{bi}^{(n)} \\ \mathbf{K}_{ib}^{(n)} & \mathbf{K}_{ii}^{(n)} \end{bmatrix} \begin{bmatrix} \mathbf{U}_b^{(n)} \\ \mathbf{U}_i^{(n)} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_b^{(n)} \\ \mathbf{P}_i^{(n)} \end{bmatrix} \quad (1)$$

여기서, 아래첨자 b와 관련된 항은 부구조들의 경계부(interface dof)를 의미하고, i는 부구조들의 내부(inner dof)를 의미한다.

식 (1)에 의해서 각각의 독립된 부구조물에 대한 해석을 행할 수 있으므로, 변위는 다음과 같이 정의된다.

$$[\mathbf{U}_i^{(n)}]_{\text{경계부 구속시}} = (\mathbf{K}_{ii}^{(n)})^{-1} \mathbf{P}_i^{(n)} \quad (2)$$

그리고 각각 독립된 부구조물 사이의 경계부에서의 반력은 다음과 같다.

$$\mathbf{R}_b^{(n)} = \mathbf{K}_{bi}^{(n)} (\mathbf{K}_{ii}^{(n)})^{-1} \mathbf{P}_i^{(n)} \quad (3)$$

식(1), (2)와 (3)에 의해서 부구조물에 대한 해석은 완료되었고, 그 다음은 각각의 부구조물들 사이의 경계 부분만으로 조합된 강성매트릭스를 이용하여 전체 구조물을 해석한다. 이를 위해서 경계부분에 대한 강성매트릭스를 조합해야 하며 이것은 각각의 부구조물들에 대해서 행해져야 한다. 각각의 부구조물에 대한 경계부의 강성 매트릭스는 다음과 같다.

$$\mathbf{K}_b^{(n)} = \mathbf{K}_{bb}^{(n)} - \mathbf{K}_{bi}^{(n)} (\mathbf{K}_{ii}^{(n)})^{-1} \mathbf{K}_{ib}^{(n)} \quad (4)$$

식(4)에 의한 각각의 부구조물에 대한 경계부의 강성매트릭스를 중첩함으로써 전체구조물에 대한 등가의 강성매트릭스를 만들 수 있다. 식(4)까지의 절차들이 완성되고 나면 전체구조물의 경계부에 대한 해석을 수행할 수 있으며, 이것을 식으로 나타내면 다음과 같다.

$$\mathbf{K}_b \mathbf{U}_b = \mathbf{Q}_b \quad (5)$$

식(5)의 오른쪽에 있는 외력항은 다음과 같이 정의된다.

$$\mathbf{Q}_b = - \sum_{i=1}^n \mathbf{R}_b^{(i)} + \mathbf{P}_b \quad (6)$$

여기서 $\mathbf{R}_b^{(i)}$ 는 각각의 부구조물에서 경계부에 발생하는 반력이다.

식(6)에 의해서 \mathbf{Q}_b 를 얻고 난후, 식(5)에 의해서 전체 구조물을 해석하면 변위를 구할 수 있으며, 이것은 아래와 같다.

$$U_b = K_b^{-1} Q_b \quad (7)$$

식(7)에 의해서 경계부의 변위가 계산되고 나면, 다음은 각각의 부구조물 내부의 변위를 수정할 수 있다. 이것은 식(2)에 의한 부구조물의 내부변위가 경계부를 고정시킨 경우에 발생하는 것이므로 경계부의 변위가 결정되면, 부구조물의 내부변위가 수정되어야 한다. 따라서, 다음에 의해서 부구조물의 내부변위를 수정한다.

$$U_i^{(n)} = -(K_{ii})^{-1} K_{ib} U_b + U_i^{(n)} \quad (8)$$

식(1)에서 식(8)까지의 절차를 수행함으로써 전체 구조물에 대한 해석을 완료하게 된다.

2-2. 분산구조해석법

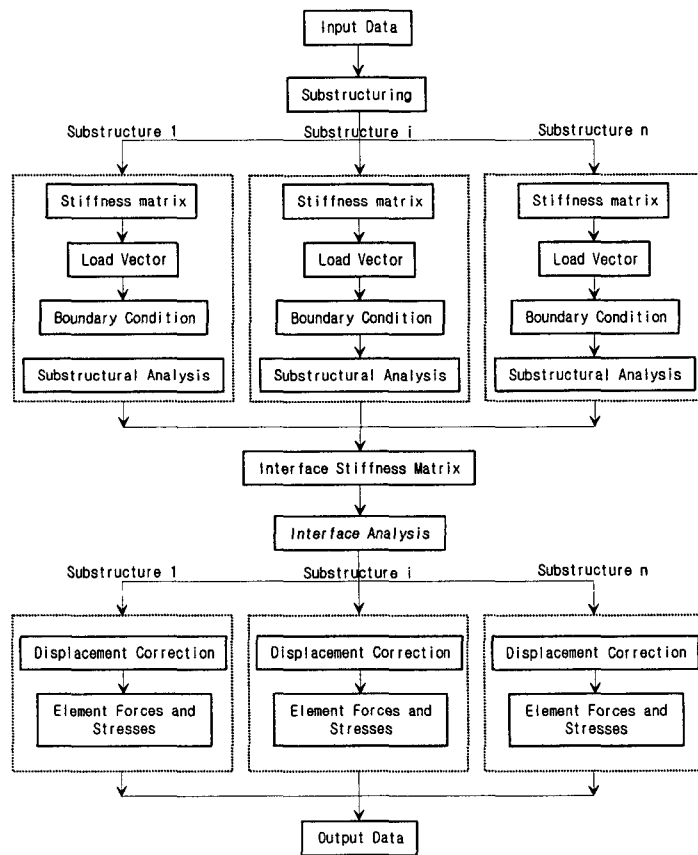


그림 1. 부구조기법을 이용한 분산구조해석법의 흐름도

앞절에서 설명한 바와 같이 식(1),(2),(3),(4),(6)은 전체구조물을 여러개의 작은 부구조물로 나누었을 때의 각 부구조물에 대한 구조해석을 수행하는 루틴이며, 이 루틴을 부구조물의 개수와 동일한 slave processor로 분산시켜 master로부터 부구조물의 해석에 필요한 모든 data를 전달받아 부구조물에 대한 해석을 수행하며, 식(5)와 (7)은 각 부구조물에 대한 해석을 수행하는 과정에서 계산된 interface stiffness matrix를 slave에서 master로 넘겨서 master에서 이를 조합하여 전체구조물의 경계부에 대한 강성매트릭스를 만들고 이를 해석하여 경계부의 변위와 반력을 구하는 루틴이다. 또한 식(8)은 경계부분의 해석이 종료된 다음, 부구조물의 경

계부분을 제외한 부구조물의 내부에 존재하는 모든 자유도들에 대한 변위들을 수정하는 루틴이며 이 루틴을 master에서 계산된 interface에 대한 변위를 넘겨받아 수행된다. 마찬가지로 이러한 수정루틴은 부구조물의 개수만큼 반복되어야 한다. 또한 분산구조 해석법의 효율성을 높이기 위해서는 네트워크에 연결된 개인용 PC들(slave)에서의 작업량을 균등히 하여 해석을 수행해야 하며, 작업량이 균등하지 않게 되면 각각의 slave에서 계산되는 소요시간에 차이가 발생하여 slave들에서 계산된 결과가 master로 전달되는 시간이 불필요하게 길어지게 되어 분산구조 해석법의 효율성이 저하된다. 이러한 load balancing^{10,11)}을 위해서 Multi-level 부구조 기법이 있지만, 본 논문에서는 Multi-level 부구조기법을 사용하지 않고 부구조사이의 경계부분이 하나의 층이나 최대 2개의 층에 걸쳐서 발생하도록 제한하였다. 이것은 해석적 측면만을 고려한다면 바람직한 기법이 아니라고 판단되지만 부구조 기법을 최적설계에 까지 적용한다고 고려한다면 본 논문에서 사용한 기법은 무리가 없는 것으로 판단된다.

부구조기법을 이용한 분산구조해석법을 흐름도로 나타내면 그림 1과 같다.

따라서 다음의 가정하에서 수행된다.

- 1) 각각의 부구조물 사이의 경계부분(Interface)은 하나의 층이나 최대 2개층에 걸쳐서 발생한다.
- 2) 부구조물의 개수와 slave process의 수를 동일하게 한다. 즉, 원하는 부구조물의 개수만큼의 slave process를 가져야 한다.
- 3) solver는 LU Decomposition을 사용한다.
- 4) 본 논문에서는 분산구조해석법의 data 분산을 위해서 MPI (Message-Passing Interface)를 사용한다.

3. 적용예제

적용예제로서 평면골조의 3경간 54층 구조물 (total dof : 660)과 7경간 90층 구조물 (total dof : 2184)을 사용하며, 하중은 풍하중을 횡방향으로 작용시켰으며, 경간은 10m, 층고는 4m로 하였다. 두 모델을 분산구조 해석법을 이용하여 해석한 결과를 고찰해기로 한다.

3-1. 부구조물 수의 변화에 따른 부구조물 내부의 dof수와 부구조물들 사이의 경계부 dof수의 변화

전체 구조물을 여러개의 작은 부구조물로 나눔에 따른 inner dof의 수는 다음의 함수 형태로 변화하며, 그래프로 나타내면 그림 2와 같다.

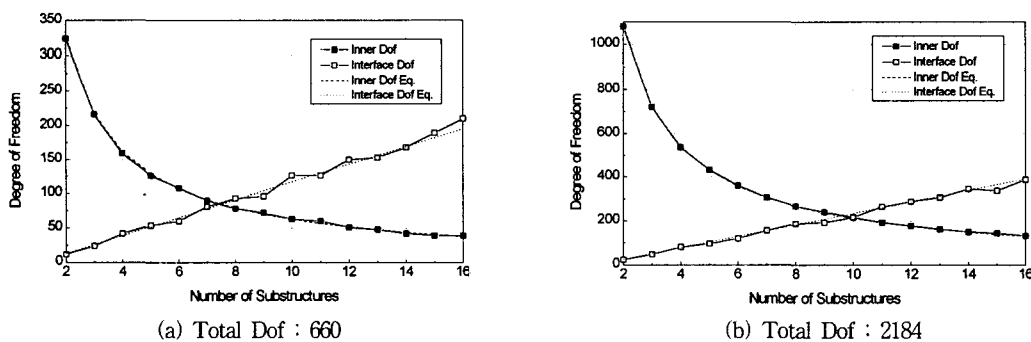


그림 2. 부구조물의 개수에 따른 Dof의 변화양상

부구조물의 해석은 각각의 slave에서 수행되며 부구조물의 수가 증가함에 따라 inner dof수가 분수함수 형

태로 감소하기 때문에 slave에서 소요되는 시간은 실제로 감소하는 반면 부구조물 경계부에서의 해석은

$$\text{Inner Dof} = \frac{\text{Total Dof} + 0.52981 * 3 * (\text{Number of bay} + 1) * \text{Number of Substructures}}{\text{Number of Substructures}} - 2.41057 * (\text{Number of bay} + 1) \quad (9)$$

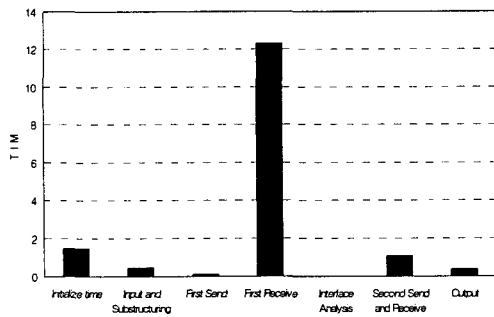
$$\text{Interface Dof} = 1.08333 * 3 * (\text{Number of Bay} + 1) * (\text{Number of Substructures} - 1) \quad (10)$$

master에서만 수행되며 부구조물의 수가 증가함에 따라 interface dof가 선형적으로 증가하기 때문에 master에서의 소요시간은 실제로 증가하게 되지만 전체 소요시간은 sequential 알고리즘의 해석시간보다 감소하게 된다.

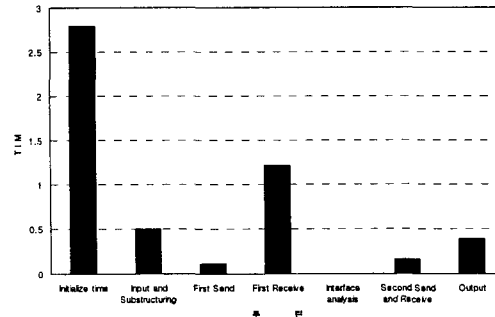
3-2. 분산구조해석 알고리즘의 각 루틴별 소요시간

분산구조해석 알고리즘의 각 루틴별 소요시간을 분석해 보면, slave process의 수가 증가함에 따라 분산구조해석 알고리즘의 초기화부분의 소요시간이 증가함을 알 수 있으며, 특히 dof의 수가 적은 구조물에서는 초기화에 소요되는 시간이 부구조물의 해석시간보다 훨씬 커지게 된다는 것을 알 수 있다. 각 루틴에 대한 소요시간을 그래프로 나타내면 그림 3과 같다.

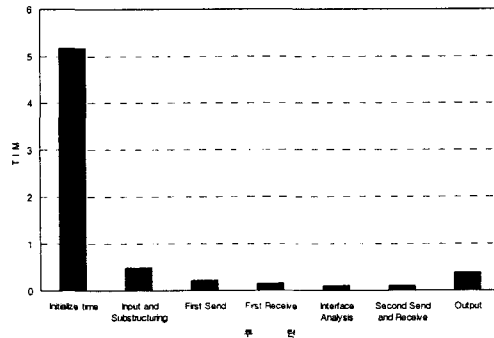
그림 3을 살펴보면, 전체 소요시간에 비해 각 루틴이 차지하는 비율은 slave processor의 수가 증가함에 따라 순수해석시간이 다른 루틴들에 비해 점점 감소하는 경향을 알 수 있으며, 특히 총 dof의 수가 660개이고 slave processor의 수가 8개인 경우에는 순수 해석시간의 비율이 다른 루틴들보다 감소하는 것을 알 수 있다. 따라서 전체 dof의 수가 적은 구조물에서는 slave processor를 너무 많이 증가시키면 초기화에 소요되는 시간 때문에 분산구조해석 알고리즘의 효율성이 저하된다는 사실을 알 수 있다.



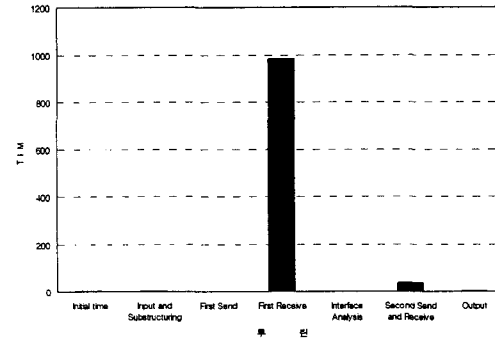
(a) Processor 2개 (Total Dof : 660)



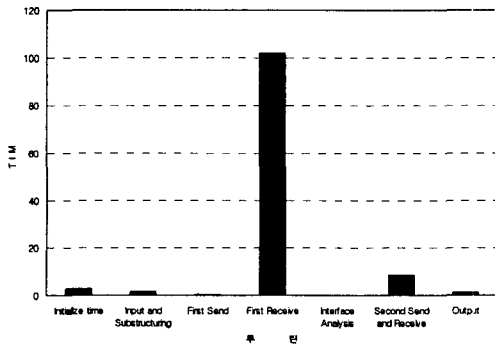
(b) Processor 4개 (Total Dof : 660)



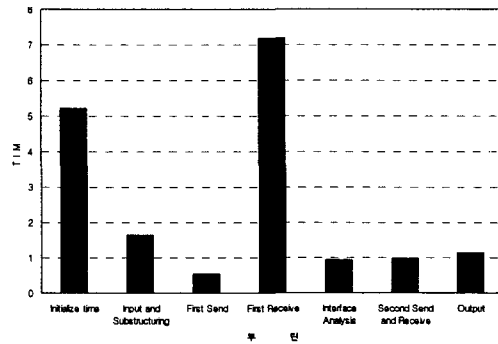
(c) Processor 8개 (Total Dof : 660)



(d) Processor 2개 (Total Dof : 2184)



(e) Processor 4개(Total Dof : 2184)



(f) Processor 8개(Total Dof : 2184)

그림 3. 각 루틴별 소요시간

3-3. LU Decomposition Solver의 특성분석

본 연구에서 채택한 LU Decomposition solver는 루틴의 반복횟수가 순자유도수 (이하 NNS)의 세제곱에 비례하므로, 이 solver에서 소요되는 시간을 regression analysis에 의해 예측할 수 있으며, 이 소요시간을 순자유도에 대한 함수로 나타내면 다음과 같다.

$$\begin{aligned} \text{Analysis Time} = & 1.3234E-06 * NNS^3 - 0.00028 * NNS^2 \\ & + 0.02941 * NNS - 0.9328 \end{aligned} \quad (11)$$

식(11)과 실제 해석에 소요되는 시간과를 비교해 보면 최대오차가 약 10%정도라는 것을 알 수 있으며, 따라서 식(11)은 실제 해석시간과 근사하다는 것을 알 수 있다. 또한 식(11)은 부구조물의 해석뿐만 아니라 Interface 부분의 해석에서도 동일하게 적용된다는 것을 알 수 있다.

실제 해석에 소요되는 시간과 식(11)에 시간을 그래프로 나타내면 그림 4와 같다.

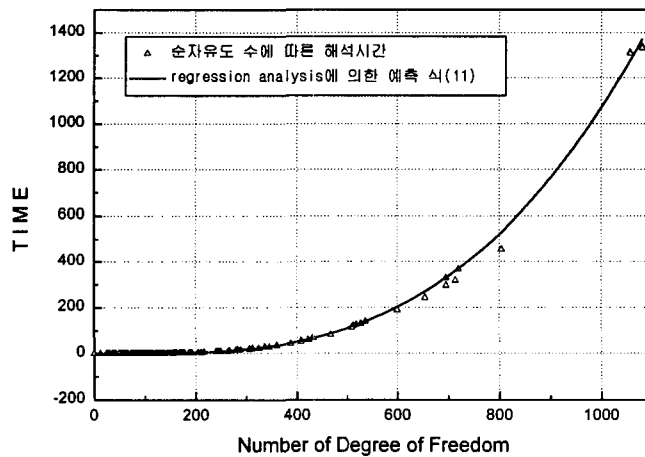
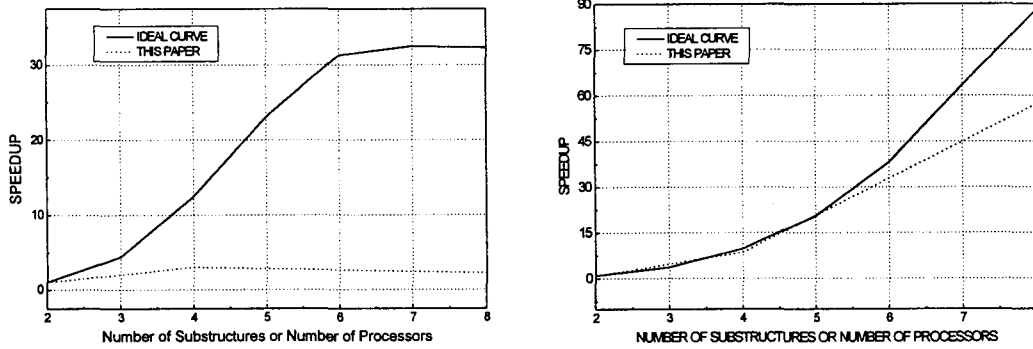


그림 4. 실제해석 시간과 regression analysis에 의한 예측과의 비교

3-4. Speedup의 비교

3-1절과 식(11)을 근거로하여 이론적인 speedup¹²⁾을 규정하고 이것을 본 연구에서 얻어낸 알고리즘에서의 speedup과를 비교하여 그래프로 나타내면 그림 5와 같이 나타난다.



(a) Total Dof : 660

(b) Total Dof : 2184

그림 5. 이론적인 Speedup과 본 알고리즘에 의한 Speedup의 비교

그림 5(a)에서 볼 때, 이론적인 speedup의 곡선에 비해 본 알고리즘의 speedup이 현저하게 저하되어 있는 것을 알 수 있는데 이것은 그림 3의 전체 Dof에 대한 루틴별 소요시간을 고려해 봄으로써 그 원인을 찾을 수 있다. 그림 3의 (a), (b)와 (c)에서 나타나는 양상은 순수해석에 관련된 소요시간의 비율보다 초기화에 소요되는 시간의 비율이 현저하게 높게 나타나고 있으며, 이 초기화 시간 때문에 분산구조해석법에서의 speedup이 저하된 것으로 판단된다. 또한 그림 5(b)는 그림 5(a)보다는 speedup이 훨씬 양호하게 나타나고 있는데, 이것은 그림3의 (d), (e)와 (f)에서 보는 것처럼, 순수해석시간의 비율이 알고리즘의 초기화에 소요되는 시간의 비율에 비해 훨씬 많으므로 분산구조해석법에서의 speedup이 향상된 것으로 판단된다. 한편 그림 5(b)를 보면 processor의 수가 5개이상인 경우, 이론적인 speedup의 곡선보다 본 알고리즘의 speedup 곡선이 다소 저하되는 양상을 나타내는데 이것은 processor의 수가 증가함에 따른 순수해석시간의 감소로 인해서, 분산구조해석 알고리즘의 초기화를 비롯한 순수해석시간을 제외한 다른 루틴에서의 소요시간의 비율이 증가하고 있기 때문인 것으로 판단된다.

4. 결 론

부구조기법을 이용한 분산구조해석법에 대한 결론으로서 다음을 얻을 수 있다.

- (1) 각각의 부구조물에 대한 inner dof의 개수는 부구조물의 개수에 따른 분수함수의 형태로 감소하며, interface dof의 개수는 부구조물의 개수에 따라 선형적으로 증가하는 양상을 나타내고 있다. 이러한 현상에 의해서, slave processor의 수가 증가하면, 각각의 slave에서 소요되는 해석시간은 감소하는 반면 master에서의 interface에 대한 해석시간은 증가하는 것으로 나타난다.
- (2) 분산구조 해석법의 알고리즘에서 사용한 solver는 LU Decomposition이며, 이 solver는 반복횟수가 순자유도수의 세제곱에 비례한다. 따라서 LU Decomposition의 다른 solver를 사용한다면, speedup의 양상이 바뀔 수 있다. 그러므로 최적의 solver를 사용하는 것이 필요하다고 판단된다.
- (3) 동일한 dof를 가지는 구조물에서, slave processor의 수가 증가하면 전체해석 시간에 비해 순수해석시간의 비율이 감소하는 반면, 다른 루틴들에서 소요되는 시간의 비율은 증가하게 된다. 따라서 이러한

현상을 고려하여 적절한 slave processor의 수를 결정하는 것이 필요하다고 판단된다.

- (4) 전체 구조물의 dof의 수가 660개인 경우에는 이론적인 speedup곡선에 비해 본 알고리즘의 speedup이 상당히 저하한다. 또한 전체 구조물의 dof의 수가 2184개인 경우에는 이론적인 speedup곡선에 비해 본 알고리즘의 speedup이 양호하게 나타난다. 따라서 부구조기법을 이용한 분산구조해석법은 전체구조물의 수가 적은 구조물의 해석에서는 speedup의 향상을 기대하지 못한다. 하지만 본 연구가 정역학적인 해석을 그 대상으로 하였으므로, 순수해석시간이 해석시간의 거의 대부분을 차지하는 동해석이나, 최적설계를 목적으로 한다면, 전체 구조물의 자유도 수가 적은 구조물에서도 양호한 speedup을 얻을 수 있으리라고 판단된다.

따라서 부구조기법을 이용한 분산구조해석법의 개발에 있어서, solver에 대한 수정과 더불어 전체 dof의 개수에 따른 최적의 slave processor수에 대한 평가가 요구되며, 또한 동역학이나, 최적설계에 대한 분산구조해석법의 연구가 필요할 것으로 사료된다.

참 고 문 헌

1. H. Adeli and O. Kamal, Parallel Processing in Structural Engineering, Elsevier Applied Science, 1993
2. M. Papadrakakis, Solving Large-scale Problems in Mechanics, John Wiley & Sons, 1993
3. Hyo Seon Park and Hojjat Adeli, "Distributed Neural Dynamics Algorithms for Optimization of Large Steel Structures", Journal of Structural Engineering, ASCE, Vol. 123, No. 7, pp. 880-888, July, 1997
4. Message Passing Interface Forum, MPI-2: Extensions to the Message-Passing Interface, University of Tennessee, July, 1997
5. Message Passing Interface Forum, MPI-2 Journal of Development, University of Tennessee, July, 1997
6. J. S. Przemieniecki, Matrix Structural Analysis of Substructures, Journal of the AIAA, Vol. 1, No. 1, pp. 138-147, January 1963
7. A. K. Noor and H. E. Lowder, Approximate Reanalysis Techniques with Substructuring, Journal of the Structural Division, Proceedings of the American Society of Civil Engineers, Vol. 101, No. ST8, August 1975
8. A. K. Noor, H. A. Kamel and R. E. Fulton, Substructuring Techniques - Status and Projections, Computers & Structures, Vol. 8, pp. 621-632, 1978
9. William Weaver, Jr. and Paul R. Johnston, Structural Dynamics by Finite Elements, Prentice-Hall INC., 1987
10. David E. Keyes, Ahmed Sameh and V. Venkatakrishnan, Parallel Numerical Algorithms, Kluwer Academic Publishers, 1997
11. Lou Baker, B. J. Smith, Parallel Programming, McGraw-Hill, 1997
12. 박효선, "대형 구조물을 위한 병렬 구조해석 및 설계", 전산구조공학회지, 제9권, 제3호, pp 47-53, 1996년 9월