

부구조기법을 이용한 병렬 고유치해석 알고리즘 개발

Development of Parallel Eigenvalue Solution Algorithm with Substructuring Techniques

김재홍* 성창원** 박효선***
Kim, Jae Hong Sung, Chang Won Park, Hyo Seon

ABSTRACT

The computational model and a new eigenvalue solution algorithm for large-scale structures is presented in the form of parallel computation. The computational loads and data storages required during the solution process are drastically reduced by evenly distributing computational loads to each processor. As the parallel computational model, multiple personal computers are connected by 10Mbits per second Ethernet card. In this study substructuring techniques and static condensation method are adopted for modeling a large-scale structure. To reduce the size of an eigenvalue problem the interface degrees of freedom and one lateral degree of freedom are selected as the master degrees of freedom in each substructure. The performance of the proposed parallel algorithm is demonstrated by applying the algorithm to dynamic analysis of two-dimensional structures.

1. 서 론

초고층건물을 포함한 대형구조물의 구조해석은 많은 수의 자유도와 부재수를 포함한다. 또한 구조설계 업무는 반복적 구조해석에 기초하여 이루어지는 점을 감안하면 대형구조물의 구조해석 및 설계는 상당한 계산 시간과 비용이 소요된다. 그리고 해석 및 설계에 관련된 연산이 기존의 컴퓨터 1대에서 순차적으로 이루어지는 직렬알고리즘을 이용하는 경우, 구조설계 관련 업무의 효율성은 기대하기 힘들다. 특히 고유치해석을 수반하는 동적해석의 경우는 더욱 심각한 실정이다.

구조물의 고유진동수와 모우드의 결정이 선행되어야 하는 동적해석은 해석시간의 대부분이 고유치해석에 소요되기 때문에, 효율적 동적해석기법의 개발은 효율적인 고유치해석 기법의 개발을 의미한다. 이러한 맥락에서 고유치 해석시 유효자유도의 수를 줄여 수치해석의 효율성을 증가시키기 위한 노력이 많이 경주되어 왔다. 이는 모우드합성법(component mode method)^{1),2)}이나 축약법(condensation method)^{3),4),5),14)} 등의 형식으로 개발되어 고유치문제의 차수를 감소시켜 전체적인 동적해석에 소요되는 시간을 단축하였으나, 여전히 강성축약시 방정식해법 등의 고유치문제 전 단계에서 연산이 증가하여 비효율적이다.

대형구조물의 구조해석 및 설계의 효율성 증진은 고성능 연산기의 사용^{6),7),8)}을 보편화하거나, 대형구조물을 위한 효과적인 알고리즘의 개발로 이루어질 수 있다. 그러나 국내 실정상 고성능 연산기의 보편화는 많은 투

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

자비용으로 비현실적이므로, 보편화되어 있는 개인용 컴퓨터를 이용하여 고성능 연산기 수준의 성능을 발휘할 수 있는 연산기법과 이에 적합한 구조해석 알고리즘을 개발하는 것이 타당하다.

따라서 본 논문에서는 네트워크로 연결된 개인용 PC를 WMP^{9),10)}(Windows Message Passing Interface)를 이용하여 가상의 병렬시스템을 구축하고, 기존의 부구조기법과 축약법을 이용하여 대형 구조물의 동적해석에 다수의 개인용 PC를 이용하는 병렬 고유치해석 알고리즘을 개발하고자한다. 그리고 개발된 알고리즘을 2차원 평면골조의 고유치문제에 적용하여 이의 효율성과 적용성을 직렬해석법과 비교 분석한다.

2. 부구조기법을 이용한 병렬 고유치해석 알고리즘

기존의 상용화된 구조해석 프로그램들은 PC 한 대에서 모든 구조해석 작업을 수행하는 직렬해석법이다. 본 연구에서는 네트워크로 연결된 개인용 PC들을 이용하여 고성능 연산기의 성능을 발휘할 수 있는 병렬시스템을 구축하고, 대형구조물을 여러 개의 부구조물로 나누어서 각 부구조물의 연결부에서의 적합조건을 이용하여 해석하는 부구조기법(substructuring method)과 축약법을 이용하여, 분산구조해석법(distributed structural analysis method)에 적용 가능한 병렬 고유치해석 알고리즘을 개발하고 구조물의 고유치 해석에 소요되는 시간을 단축하여 동적해석의 효율성을 높이고자 한다.

현재 상용화된 건물의 동적해석 프로그램에서 일반적으로 이용되는 고유치해석 알고리즘의 가정은 지진 하중의 수직방향 성분은 무시하고, 수평방향에 대해서만 해석하며, 구조물의 응답은 바닥판의 수평방향의 응답으로만 표현된다고 가정해서 구조물의 질량은 lumped mass로 가정한다.

2-1 부구조기법을 이용한 고유치해석 알고리즘

정적해석에 관련한 분산구조해석법은 선행연구^{11),12),13)}에서 수행되었으며, 본 연구에서는 부구조기법과 축약법을 이용한 병렬 고유치해석법을 제시하고자 한다. 대형구조물을 해석하는 경우 계산의 효율성을 위하여 축약법이 많이 사용되고 있다. 그러나 consistent mass일 때의 축약법은 전체 구조물을 하나의 시스템으로 해석하며, 축약시 남게 되는 주자유도의 선택이 불분명하며 주자유도의 선택에 따라 해의 정확도에 영향을 미치게 되고 부구조기법의 적용성이 어렵게 되어 병렬화에 문제점들을 가지고 있다.

본 연구에서는 부구조기법에서 각 부구조물에 대해서 가상의 경계부와 경계부 사이의 내부층에 대해 1개의 횡방향 자유도를 남기는 lumped mass일 때 적용 가능한 정적축약법을 적용하였으며, 축약된 강성 매트릭스를 조합한 후 다시 횡방향 자유도에 대해서 정적축약법을 시행하여 최종적인 축약된 강성행렬을 구성하였다.

일반적인 고유치문제는 다음과 같이 표현된다.

$$\mathbf{K} \mathbf{x} = \lambda \mathbf{M} \mathbf{x} \quad (1)$$

여기서 \mathbf{K} 는 강성행렬, \mathbf{M} 은 질량행렬이고, \mathbf{x} 는 모우드벡터, λ 는 고유진동수의 제곱이다.

본 연구에서 개발하는 부구조기법을 이용한 병렬 고유치해석 알고리즘을 전체적으로 살펴보면 다음과 같다.

STEP 1. 전체 구조물을 N개의 부구조물로 분할한다.

STEP 2. 각각의 부구조물에 대해서 강성행렬을 구성한다.

$$[\mathbf{K}]_1, [\mathbf{K}]_2, \dots, [\mathbf{K}]_N \quad (2)$$

STEP 3. 각 부구조물의 강성행렬에 대해서 축약법을 적용한다.

부구조기법에 적용된 강성행렬의 축약법은 다음과 같다.

STEP 3-1. 각 부구조물에 대해 보조자유도 \mathbf{x}_s 와 경계부자유도 및 경계부 사이의 내부층에 대한 1개의 횡방향 자유도를 \mathbf{x}_m 으로 하는 주자유도를 분리한다.

$$\mathbf{x}_i = \begin{cases} \mathbf{x}_s \\ \mathbf{x}_m \end{cases} \quad \begin{matrix} \mathbf{x}_s \text{ (보조자유도)} \\ \mathbf{x}_m \text{ (주자유도)} \end{matrix} \quad (3)$$

STEP 3-2. 강성행렬을 주자유도와 보조자유도에 대해서 재배열한다.

$$\begin{bmatrix} \mathbf{K}_{ss} & \mathbf{K}_{sm} \\ \mathbf{K}_{ms} & \mathbf{K}_{mm} \end{bmatrix}_1 \begin{Bmatrix} \mathbf{x}_s \\ \mathbf{x}_m \end{Bmatrix}_1 = \begin{Bmatrix} \mathbf{F}_s \\ \mathbf{F}_m \end{Bmatrix}_1$$

$$\vdots \quad \quad \quad \vdots$$

$$\begin{bmatrix} \mathbf{K}_{ss} & \mathbf{K}_{sm} \\ \mathbf{K}_{ms} & \mathbf{K}_{mm} \end{bmatrix}_N \begin{Bmatrix} \mathbf{x}_s \\ \mathbf{x}_m \end{Bmatrix}_N = \begin{Bmatrix} \mathbf{F}_s \\ \mathbf{F}_m \end{Bmatrix}_N \quad (4)$$

고유치해석에서는 $\mathbf{F}_s = \mathbf{F}_m = 0$ 이며 식(4)의 첫 행을 전개하면 다음과 같다.

$$\begin{Bmatrix} \mathbf{x}_s \\ \vdots \\ \mathbf{x}_s \end{Bmatrix}_1 = -[\mathbf{K}_{ss}]_1^{-1} [\mathbf{K}_{sm}]_1 \{\mathbf{x}_m\}_1$$

$$\vdots$$

$$\begin{Bmatrix} \mathbf{x}_s \\ \vdots \\ \mathbf{x}_s \end{Bmatrix}_N = -[\mathbf{K}_{ss}]_N^{-1} [\mathbf{K}_{sm}]_N \{\mathbf{x}_m\}_N \quad (5)$$

STEP 3-3. 식(5)를 식(3)에 대입하여 각 부구조물의 변환행렬을 구성한다.

$$\begin{Bmatrix} \mathbf{x} \\ \mathbf{x}_m \end{Bmatrix}_1 = \begin{Bmatrix} \mathbf{x}_s \\ \mathbf{x}_m \end{Bmatrix}_1 = \begin{bmatrix} -\mathbf{K}_{ss}^{-1} \mathbf{K}_{sm} \\ \mathbf{I}_m \end{bmatrix} \{\mathbf{x}_m\}_1 = [\mathbf{T}]_1 \{\mathbf{x}_m\}_1$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$\begin{Bmatrix} \mathbf{x} \\ \mathbf{x}_m \end{Bmatrix}_N = \begin{Bmatrix} \mathbf{x}_s \\ \mathbf{x}_m \end{Bmatrix}_N = \begin{bmatrix} -\mathbf{K}_{ss}^{-1} \mathbf{K}_{sm} \\ \mathbf{I}_m \end{bmatrix}_N \{\mathbf{x}_m\}_N = [\mathbf{T}]_N \{\mathbf{x}_m\}_N$$

$$[\mathbf{T}]_1 = \begin{bmatrix} -\mathbf{K}_{ss}^{-1} \mathbf{K}_{sm} \\ \mathbf{I}_m \end{bmatrix}_1 \quad \dots \quad [\mathbf{T}]_N = \begin{bmatrix} -\mathbf{K}_{ss}^{-1} \mathbf{K}_{sm} \\ \mathbf{I}_m \end{bmatrix}_N \quad (6)$$

식(5)를 식(4)의 2행에 대입하여 각 부구조물의 주자유도만을 고려한 축약된 강성행렬을 구성하면 다음과 같다.

$$\begin{Bmatrix} [\mathbf{K}_{mm}]_1 - \{ [\mathbf{K}_{ms}]_1 [\mathbf{K}_{ss}]_1^{-1} [\mathbf{K}_{sm}]_1 \} \\ \vdots \\ [\mathbf{K}_{mm}]_N - \{ [\mathbf{K}_{ms}]_N [\mathbf{K}_{ss}]_N^{-1} [\mathbf{K}_{sm}]_N \} \end{Bmatrix} \{\mathbf{x}_m\}_1 = 0 \quad (7)$$

여기서 축약된 각 부구조물의 축약된 강성행렬은 다음과 같다.

$$[\mathbf{K}_0]_1 = [\mathbf{T}]_1^T [\mathbf{K}]_1 [\mathbf{T}]_1 = [\mathbf{K}_{mm}]_1 - \{ [\mathbf{K}_{ms}]_1 [\mathbf{K}_{ss}]_1^{-1} [\mathbf{K}_{sm}]_1 \}$$

$$\vdots \quad \quad \quad \vdots$$

$$[\mathbf{K}_0]_N = [\mathbf{T}]_N^T [\mathbf{K}]_N [\mathbf{T}]_N = [\mathbf{K}_{mm}]_N - \{ [\mathbf{K}_{ms}]_N [\mathbf{K}_{ss}]_N^{-1} [\mathbf{K}_{sm}]_N \} \quad (8)$$

STEP 4. 각 부구조물의 수만큼 각각의 축약된 강성행렬을 조합하면, 각 부구조물의 주자유도를 고려한 전체 구조물에 대한 강성행렬이 만들어지고, 만들어진 강성행렬을 다시 STEP 3에서처럼 전체 구조물에 대해 횡방향 자유도를 주자유도로 택하고 나머지는 보조자유도로 택하여 축약하면 최종적인 고유치해석을 위한

축약된 강성행렬(K_R)이 구성된다. (STEP 3과 동일과정)

STEP 5. 축약된 강성행렬(K_R)과 집중질량(M_R)으로 고유치해석을 한다. 여기에서 고유치 해석은 층수×층수의 차수를 갖는 정방행렬이다.

$$K_R \mathbf{x} = \lambda M_R \mathbf{x} \quad (9)$$

2-2 병렬 고유치해석 알고리즘

전체 구조물을 여러 개의 부구조물로 나누고, N개의 부구조물들은 Ethernet Network에 연결된 다수의 PC에 분산 할당하여 계산하도록 구성되어 있다. 본 연구에서는 개발한 병렬 고유치해석 알고리즘은 부구조물의 수와 slave processor의 수가 동일한 1 : 1 대응의 방식이고 구축된 병렬연산 시스템은 그림 1과 같다.

그림 1에서 master와 slave로 사용된 실험대상 PC들은 윈도우 98을 Operating System으로 사용하고 있으며, 펜티엄 450MHz의 CPU와 128Mbyte의 메인 메모리 그리고 10Mbps의 Ethernet Network카드로 구성되어 있다. 네트워크 상에 연결된 다수의 PC들 중에서 임의의 하나를 master로 설정하고 나머지 PC들 중에서 부구조물의 수만큼을 slave로 설정하며, master와 slave의 연산에 필요한 정보 및 데이터는 WMPI 패키지를 이용하여 전송한다

본 연구에서의 병렬 고유치해석 알고리즘의 분산구조 해석법을 살펴보면, 2-1절에서 STEP1과 STEP4, STEP5는 master processor에서 수행되며, 나머지 STEP은 slave processor에서 수행된다. 병렬 고유치해석 알고리즘의 흐름도는 그림 2와 같으며, master processor의 수는 1개이며, slave processor의 수는 부구조물의 수와 동일한 1 : 1 대응방식이다

그림 2의 흐름도를 살펴보면, master에서 전체 구조물의 해석에 필요한 절점, 부재, 경계조건을 모두 입력 받고, 입력받은 데이터를 부구조물의 수만큼 부구조기법을 이용하여 나눈다. 나누어진 부구조물에 대한 데이터를 master에서 slave로 연속적으로 전송하며, 각 slave에서는 전송된 데이터를 이용하여 강성행렬에 대해서 주자유도만을 남기는 축약을 수행한다. 각 부구조물에 대해서 축약된 강성행렬을 master로 전송하며 master에서는 전송된 축약된 강성행렬들을 조합한다. 조합된 강성행렬을 다시 횡방향 자유도에 대해서 남기는 축약을 시행하고, 최종적인 축약된 강성행렬과 질량행렬을 이용하여 고유치해석을 하게 된다.

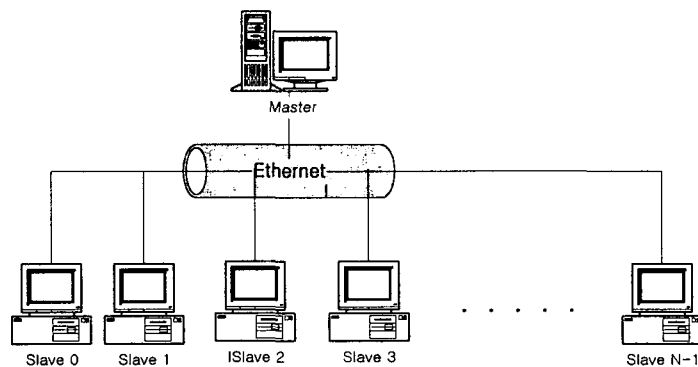


그림 1. 병렬 연산 시스템

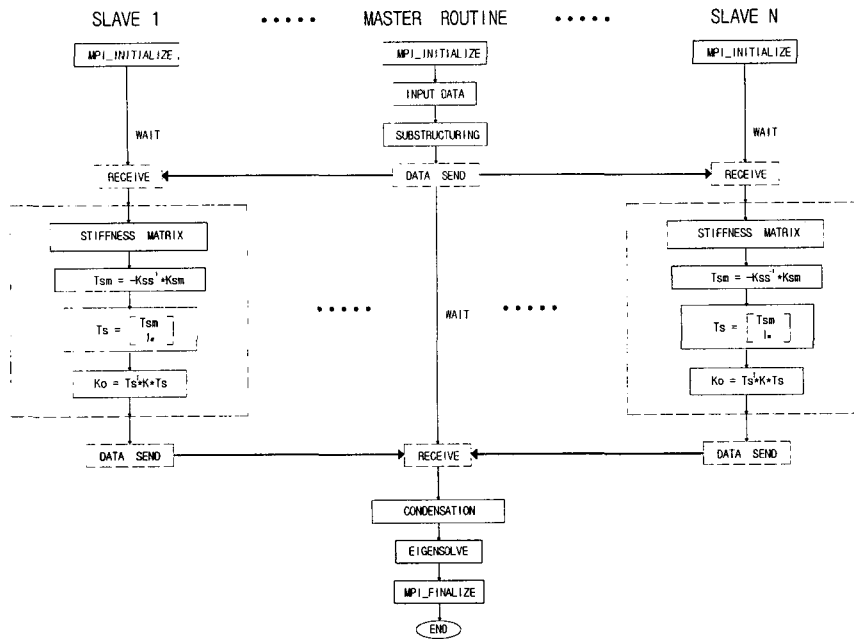


그림 2. 부구조기법을 이용한 병렬 고유치해석 알고리즘

여기에서 고유치해석에 이용되는 차수는 층수×층수의 차수이며, 고층의 구조물이라도 분산해석법으로 인해 직렬해석법에서의 전체적인 축약과정을 부구조물의 수만큼 slave processor에 할당하게 되므로, 직렬해석법에 비해서 병렬해석시 해석시간을 단축할 수 있다.

3. 적용 예 제

적용예제로서 평면골조의 전체 자유도수 2184개 (7경간 90층 구조물, 54×54의 lumped mass)의 구조물을 사용하며, 구조물의 질량은 바닥판에 집중된 집중질량으로 가정하고, 경간은 10m, 층고는 4m로 하였다. 구조물이 비감쇄 자유 진동일 때 직렬해석법과 병렬 고유치 해석법에서의 결과를 비교/평가하기로 한다.

3-1. 부구조물의 수에 따른 해석시간의 영향

병렬 고유치해석 알고리즘에서 slave processor에서 많은 해석시간을 요구하는 부분은 각 부구조물에서의 강성행렬의 보조자유도에 대한 축약과정이며, 여기에서 전체 주자유도와 보조자유도는 합은 전체해석 자유도수인 2184개이며, 부구조물의 수에 따른 slave processor에서의 해석량은 slave processor에서의 부구조물의 수에 따라 주자유도와 보조자유도의 수는 반비례하는 반면에, master processor에서는 각 부구조물에서 축약한 강성행렬들을 조합하고 횡방향 자유도를 주자유도로 선택하여 축약을 하는 부분에서는 보조자유도는 부구조물의 수가 증가함에 따라 증가하는 반면에, 주자유도의 차수는 질량행렬의 차수인 54×54로 항상 일정하며, 부구조물의 수가 많아질수록 master processor에서 조합해야 하는 강성행렬의 차수가 많아지게 되어 부구조물의 수가 증가할수록 보조자유도의 축약시 많은 해석시간을 요구하게 된다. 최종적으로 축약된 강성행렬과 질량행렬로 고유치해석을 하게된다. 고유치해석에 참여하는 질량행렬은 lumped mass로 가정하였기 때

표 1. 해석법에 따른 소요시간 (단위 : sec)

해석법	부구조물수		
	2	4	8
병렬해석법	316.19	42.40	8.18
직렬해석법 + 부구조기법	618.82	134.55	30.48
직렬해석법	2774.14		

문제 층당 1개의 차수를 가진다.

3-2. 각 루틴별 소요시간

해석법에 따른 소요시간은 표 1과 같으며, 병렬해석법에서의 각 루틴별 소요시간을 살펴보면 표 2와 같다. 표 2를 살펴보면 부구조물의 수가 증가할수록 각 master processor의 초기화 부분(가상의 병렬 시스템을 구축하는데 소요되는 루틴)의 소요시간이 slave processor수에 비례하여 증가함을 알 수 있다. 이는 저층구조물에 대해서는 초기화 시간이 부구조물의 수가 많아지게 되면 전체구조물의 해석시간의 대부분을 차지할 수 있음을 보여준다. 이는 병렬 고유치해석법은 고려되는 집중질량의 차수가 많은 고층구조물에 적합함을 보여준다. 그러나 분산구조 해석법을 적용하는데 있어서, 초기화 부분은 가상 병렬 시스템을 구축하는데 소요되는 시간이므로, 분산구조 해석법의 전체 해석 시간에 포함시키지 않는 것이 타당할 것으로 판단된다. 표 2에서 Receive Data루틴에 소요되는 시간이 부구조물의 수가 2개인 경우에는 전체 해석시간의 98.08%로 나타나고 있으며, 이는 병렬 고유치해석법이 전체의 해석시간을 단축시킬 수 있음을 보여준다. 그러나 부구조물의 수가 8개인 경우에는 전체의 해석시간의 18.22%의 소요시간을 차지해서 master processor에서의 해석시간을 증가시켜서 병렬 고유치해석 알고리즘의 성능을 저하시키는 요인이 된다.

표 2. 병렬해석법에서 각 루틴별 소요시간 단위:시간(sec), 비율(%)

부구조물수	2		4		8	
	시간	비율	시간	비율	시간	비율
초기화	0.53	0.17	1.01	2.38	2.25	27.54
Input + Substructuring	0.40	0.13	0.44	1.03	0.56	7.17
Send Data	0.15	0.05	0.06	0.15	0.08	0.99
Receive Data	313.29	98.08	38.74	91.38	1.50	18.22
Condensaton	0.33	0.10	0.61	1.43	2.31	28.19
Eigensolve	1.48	0.47	1.54	3.62	1.46	17.88
전체해석시간	316.19	100	42.40	100	8.18	100
해석시간 (초기화제외)	315.66 (sec)		41.39 (sec)		5.93 (sec)	

3-3. 부구조물의 수에 따른 병렬 고유치해석 알고리즘의 전체 해석시간

Slave의 소요시간은 축약시 제거되는 보조자유도의 수에 의해서 결정된다. 각 부구조물에서 축약은 각 slave에서 수행되며, 축약시 역행렬의 계산은 cholesky solver를 사용하였다. 직렬해석법의 해석시간과 병렬 고유치해석 알고리즘의 전체 해석시간을 살펴보면 그림 3과 같다.

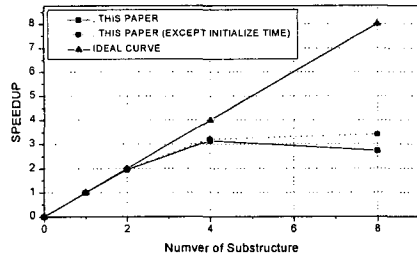
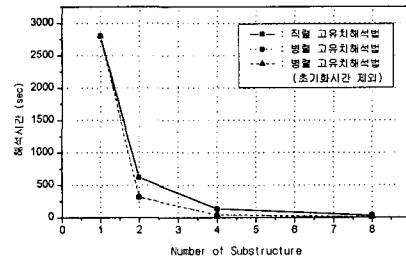


그림 3. 부구조물 수에 따른 해석시간 비교



4. Speed-up의 비교

그림 3에서처럼 부구조물의 수가 8개일 때 해석시간이 직렬해석법과 큰 차이가 없는 것은 3-1절과 3-2절에서 언급한 것처럼, master에서의 강성행렬의 축약시 보조자유도의 수가 증가하기 때문이다.

3-4. Speed-up(성능향상)의 비교

Speed-up(이하 SP)은 식(10)으로 정의하며, 본 논문에서의 직렬해석법과 병렬 고유치해석법의 SP⁸⁾를 비교 분석하여 해석결과를 그래프로 나타내면 그림 4와 같다.

$$SP = \frac{\text{직렬 고유치해석법의 해석시간}}{\text{병렬 고유치해석 알고리즘의 해석시간}} \quad (10)$$

그림 4에서 알 수 있듯이, 이론적인 SP의 곡선에 비해서 부구조물의 수가 증가할수록 SP가 현저히 감소함을 알 수 있다. 이것은 3-2절의 각 루틴별 소요시간에서 언급했듯이, 부구조물의 수가 많아질수록 master processor에서 축약하게 되는 보조자유도의 차수가 많아지게 되기 때문이다. 실질적으로 고유치해석에 참여하는 차수는 부구조물의 수에 관계없이 구조물 전체의 총수만큼의 차수를 갖는 정방행렬이 된다. 그래서 전체해석 해석시간은 각 부구조물에 대한 축약과 master processor에서의 축약된 강성행렬들의 조합 후 축약 과정에서 대부분을 차지해서 병렬 고유치해석에서의 성능평가에는 부구조물에 대한 축약과정은 slave processor가 계산하므로 병렬성능을 증가시키는 반면에, master processor에서의 축약과정에서는 차수가 증가할수록 병렬성능을 감소시키는 원인이 된다.

4. 결 론

대형 구조물의 병렬 고유치해석 알고리즘에 대한 결론으로서 다음을 얻을 수 있다.

- (1) 각 부구조물에 대한 주자유도와 보조자유도의 합은 전체자유도의 수로 일정하지만, 각 부구조물에서 보조자유도에 대한 강성행렬을 축약한 후 master processor에서 조합하는 강성행렬의 차수는 부구조물의 수가 증가할수록 증가하게 되어, master processor에서의 보조자유도의 축약과정에서 많은 해석시간을 요구하게 된다. 따라서 병렬성능을 증가시키기 위해서는 각 부구조물에서 보조자유도의 축약시 경계부 강성행렬에 대해서도 총당 1개의 주자유도를 남기고 나머지를 보조자유도로 택해서 축약한다면 master processor에서의 축약과정이 없어지기 때문에 병렬성능이 많이 증가할 것으로 보인다.
- (2) 병렬 고유치해석 알고리즘에서 사용한 solver는 cholesky solver이다. 각 부구조물의 축약시 가장 많은 시간을 요구하는 것이 역행렬의 계산이므로, 효율적 solver 또는 병렬 solver의 개발로 전체 동적해석의 효율성을 높일 수 있다.

- (3) 본 연구에서 개발한 병렬 고유치해석 알고리즘에서는 최적의 부구조물의 개수가 있으며, 최적의 부구조물의 수를 찾는 연구가 병행되어야 할 것으로 보인다. 본 알고리즘은 1대의 PC가 하나의 부구조물에 대해 해석하는 알고리즘이며, 부구조물의 수가 많아질수록 slave processor의 수가 증가하게 되어서 각 slave processor의 초기화 시간과 네트워크의 통신량의 증가로 병렬성능을 저하시킬 수 있으므로, 1대의 PC가 다수의 부구조물을 해석할 수 있는 병렬 고유치해석 알고리즘의 개발이 필요하다.
- (4) 적용결과 직렬해석시 46분이 소요되지만, 개발된 병렬고유치 알고리즘을 적용한 결과, 8대의 개인용 컴퓨터를 이용하면 46분이 소요되는 직렬해석 시간을 8초 정도로 단축할 수 있다.

참 고 문 헌

1. M. P. Singh and L. E. Suarez, "Dynamic Condensation with Synthesis of Substructure Eigenproperties", *Journal of Sound and Vibration*, Vol. 159, No. 1, pp. 139-155, Marth, 1992.
2. W. C. Hurty, "Dynamic Analysis of structural systems Using Component Modes", *AIAA*, Vol. 3, No. 4, pp.678-685, 1964.
3. N. Bouhaddi and R. Fillod, "A Method for Selecting Master Dof in Dynamic Substructuring Using the Guyan Condensation Method.", *Computers & Structures*, Vol. 45, No. 5/6, pp.941-946, 1992.
4. R. J. Guyan, "Reduction of Stiffness and Mass Matrices." *AIAA*. Vol. 3, No. 2, pp.380, 1965.
5. N. Bouhaddi and R. Fillod, "Substructuring Using a linearized Dynamic Condensation Method.", *Computers & Structures*, Vol. 45, pp.679-683, 1992.
6. H. S. Park and H. 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.
7. H. S. Park and H. Adeli, *Neuro-computing for Design Automation*, CRC Press, 1998.
8. K. Noor, "New Computing Systems and Future High-Performance Computing Environment and Their Impact on Structural Analysis and Design", *Computers & Structures*, Vol. 64, No. 1-4, pp.1-30, 1997.
9. Message Passing Interface Forum, *MPI-2 Extensions to the Message-Passing Interface*, University of Tennessee, July, 1997.
10. Message Passing Interface Forum, *MPI-2 Journal of Development*, University of Tennessee, July, 1997.
11. 박효선, 박성무, 성창원, 김재홍, "부구조기법을 이용한 PC level 분산구조해석법", 한국전산구조공학회 발표논문집, 제11권, 제2집, pp. 53-60, 1998년 10월.
12. 성창원, 박효선 "분할 방법에 따른 분산구조 해석법의 성능분석", 한국전산구조공학회 발표논문집, 제12권, 제1집, pp. 50-57, 1999년 4월.
13. 성창원, 박효선, "PC 네트워크 시스템을 이용한 대형구조물의 분산구조해석법", 대한건축학회 논문집, 제15권, 7호, pp21-28, 1999년 7월.
14. 박효선, 박성무, 권준혁, "동적 변위 제어법의 정식화", 한국전산구조공학회 발표논문집, 제11권, 제2집, pp. 53-60, 1998년 10월.