

3차원 대형구조물의 동적해석을 위한 병렬 알고리즘 개발

Development of Parallel Algorithm for Dynamic Analysis of Three-Dimensional Large-Scale Structures

김 국 규* 성 창 원** 박 효 선***
Kim, Kuk Kyu Sung, Chang Won Park, Hyo Seon

ABSTRACT

A parallel condensation algorithm for efficient dynamic analysis of three-dimensional large-scale structures is presented. The algorithm is developed for a user-friendly and cost effective high-performance computing system on a collection of Pentium processors connected via a 100 Mb/s Ethernet LAN. To harness the parallelism in the computing system effectively, a large-scale structure is partitioned into a number of substructures equal to the number of computers in the computing system. Then, for reduction in the size of an eigenvalue problem, the computations required for static condensation of each substructure is processed concurrently on each slave computer. The performance of the proposed parallel algorithm is demonstrated by applying to dynamic analysis of a three dimensional structure. The results show that how the parallel algorithm facilitates the efficient use of a small number of low-cost personal computers for dynamic analysis of large-scale structures.

1. 서 론

구조물이 더욱더 고층화, 대형화되는 경향으로 구조공학에서 다루어야하는 문제들의 규모가 대형화되고 자유도와 부재수가 증가하고 있다. 반복적 구조 해석에 기초한 대형구조물의 효율적 구조설계는 부재와 자유도 수를 고려한 효율적 정적·동적 구조해석 기법의 개발을 필요로한다. 특히, 동적해석은 해석시간의 대부분이 고유치해석에 소요되기 때문에 하나의 프로세스를 이용하는 직렬 알고리즘을 이용하는 경우에는 구조설계 업무의 효율성을 기대하기는 어렵다. 컴퓨터 성능의 발달로 인해서 프로세스 자체의 속도증가가 계속 이루어지고 있지만, 고성능 대형 연산기는 유지관리 및 사용성에 여러 가지 문제점들을 가지고 있으므로 저렴한 프로세스를 여러 개 연결하여 사용하는 방법이 오히려 경쟁력이 있을 수도 있고, 이러한 여러 대의 프로세스로 계산량을 나누어 처리하는 병렬해석기법에 관한 연구가 활발하게 진행되고 있다.¹⁾²⁾

효율적인 동적해석기법의 개발은 효율적인 고유치해석 기법의 개발을 의미한다. 이러한 고유치해석시 유효 자유도의 수를 줄여서 수치해석의 효율성을 증가시키기 위한 노력이 많이 시도되어 왔으며, 모우드 합성법(component mode method)이나 축약법(condensation method)³⁾⁴⁾ 등의 형식으로 개발되어 고유치문제의 차수를 감소시켜 전체적인 동적해석에 소요시간은 단축하였으나, 차수를 감소시키기 위한 강성행렬의 축약등 고유치 문제 전 단계에서의 연산량이 증가하여 비효율적이다. 따라서, 3차원 대형구조물의 동적해석에서의 강성축약 및 고유치해석에서도 기존의 연산환경(PC)에서의 소요시간을 단축시키기 위해서 새로운 구조해석 알고리즘의 개발이 필요하다.

* 영남대학교 건축공학과 석사과정
** 영남대학교 건축공학과 박사수료
*** 연세대학교 사회환경·건축공학부 조교수

그러므로 본 논문에서는 ethernet network card로 연결된 다수의 개인용 컴퓨터들을 WMPI(Windows Message Passing Interface)를 이용해서 하나의 시스템으로 구성하여 PC수준의 고성능 연산기를 구축하고, 기존의 부구조기법과 축약법^{3,4)}을 이용하여 3차원 대형 구조물의 동적해석을 위한 병렬알고리즘을 개발하고자 한다. 그리고 개발된 알고리즘을 3차원 가새골조 구조의 동적해석문제에 적용하여 병렬 알고리즘의 효율성을 비교/분석하고자 한다.

2. 축약의 기본개념 및 방법

축약법은 많은 자유도중에서 수치해석상의 효율성을 증대시키기 위해서 자유도의 수를 줄이는 기법으로 전체 자유도 중에서 축약시 최종적으로 남게되는 주자유도와 축약되어지는 보조자유도로 분리하여 해석상에 필요로 하는 주자유도만으로 구성된 축약모델을 구성하는 방법이다. 이러한 축약법은 축약 후 구조물이 보조자유도가 모두 제거되고 최종적으로 해석에서 요구되어지는 주자유도만으로 구성되어 해석의 효율성을 증가시키게된다.

축약법에는 정적축약과 동적축약이 있으며, 정적축약은 1965년 Guyan에 의해 처음 제시되었으며⁵⁾ 집중질량으로 근사화된 구조물에 대해서 정적평형식을 이용하여 질량이 없는 자유도를 소거시켜 축약한다. 동적축약은 질량행렬에 대해서도 축약을 시행하며 Irons에 의해 제시되었고⁶⁾ 보조자유도를 소거할 때 고유치를 포함하는 고차항들이 제거되므로 해석결과에 있어서 고차모우드에 대해서 오차를 포함하게 된다. 따라서 본 연구에서는 집중질량(lumped mass)을 이용한 정적축약을 이용하여 보조자유도에 대한 축약을 시행한다.

정적축약방법의 전체 흐름은 다음과 같다.

- 1) 고유치문제의 크기를 줄이기 위해 전체자유도 x 를 주자유도 x_m 과 보조자유도 x_s 로 분리한다.

$$x = \begin{Bmatrix} x_s \\ x_m \end{Bmatrix} \quad (1)$$

- 2) 강성행렬 K 의 자유도를 주자유도와 보조자유도로 분리하고 재배열하면 다음과 같다.

$$\begin{bmatrix} K_{ss} & K_{sm} \\ K_{ms} & K_{mm} \end{bmatrix} \begin{Bmatrix} x_s \\ x_m \end{Bmatrix} = \begin{Bmatrix} F_s \\ F_m \end{Bmatrix} \quad (2)$$

식 (2)에서 외력항 $F=0$ 이고, 식 (2)의 강성행렬에서 1행을 전개하면 다음과 같다.

$$K_{ss} x_s + K_{sm} x_m = 0 \quad (3)$$

$$x_s = -K_{ss}^{-1} K_{sm} x_m \quad (4)$$

- 3) 식 (4)의 보조자유도 x_s 의 항을 식 (1)에 대입하여 전개하면 다음과 같다.

$$x = \begin{bmatrix} -K_{ss}^{-1} K_{sm} \\ I \end{bmatrix} \cdot x_m = T \cdot x_m \quad (5)$$

- 4) 식 (3)에 식 (4)를 대입하면 전체 자유도에서 보조자유도 x_s 를 주자유도 x_m 으로 완전히 축약된다.

$$K_{ss} [-K_{ss}^{-1} K_{sm} x_m] + K_{sm} x_m = [K_{mm} - K_{ms} K_{ss}^{-1} K_{sm}] x_m \quad (6)$$

- 5) 복잡한 식 (6)을 변환행렬 T 를 이용한 간단한 식으로 전환하면 최종적으로 축약된 강성행렬 K_o 가 구성된다.

$$[K_{mm} - K_{ms} K_{ss}^{-1} K_{sm}] x_m = [T^T K T] x_m = K_o x_m \quad (7)$$

본 논문에서의 축약법은 기존에 제시된 정적축약법⁷⁾으로 각 slave 프로세스에서 층별 축약과정과 부구조물의 축약과정을 거친 후 master 프로세스에서 각 층의 집중질량이 작용되는 3개의 주자유도로 최종적으로 축약하는 과정에 이용된다.

3. 부구조기법을 이용한 해석 알고리즘

전체 구조물에 대한 모든 해석작업을 한 대의 PC에서 수행하는 것이 기존 상용 구조해석 프로그램들의 직렬해석법인데 반해서, 본 논문에서 구현하고자하는 분산연산모델은 PC수준에서 고성능 연산기의 성능을 발휘할 수 있게 연결하여 병렬 연산시스템의 네트워크를 구성하고, 구조물을 여러 개의 부구조물로 나누는 부구조기법(substructuring techniques)⁷⁾과 사용 자유도수를 줄이는 축약법(condensation method)⁸⁾을 이용하여 분산구조해석법에 적용가능한 병렬해석 알고리즘을 개발하여 3차원 대형구조물의 동적해석시에 대부분의 시간이 소요되는 고유치해석에 소요되는 시간을 단축하고자 한다. 이를 위하여 먼저 직렬알고리즘을 개발하고 직렬알고리즘과 병렬알고리즘의 해석시간 비교를 통해서 직렬에 대한 병렬의 효율성을 확인하고자한다.

3.1. 부구조기법을 이용한 직렬해석 알고리즘

본 논문에서는 부구조기법과 축약법을 이용하여 병렬 고유치해석법을 개발하기 위해서 먼저 직렬 고유치 해석법을 제시한다. 본 알고리즘에서 대형 구조물을 해석할 때는 수치해석 계산상 효율성을 높이기 위해서 많이 이용되는 축약법을 사용하고, 전체 구조물을 하나로 해석하여 축약시 남게되는 주자유도의 선택이 불분명하여 주자유도의 선택에 따라 해의 정확도에 영향을 미치게되는 연속질량(consistent mass)대신 집중질량(lumped mass)을 이용하고 정적축약법을 적용하였다.

본 논문에서 기존의 부구조 기법을 이용한 직렬 알고리즘의 단계를 간략하게 정리하면 다음과 같다.

Step 1. 전체 구조물을 작은 구조물인 부구조물로 나누고 이를 다시 부구조물내의 층수로 나눈다.

Step 2. 층별 부구조물 강성에 대한 정적축약 단계

Step 2-1. 각 층으로 나누어진 작은 구조물의 강성행렬을 구성

Step 2-2. 층에 대하여 구성된 강성행렬 중에서 층의 주절점을 제외한 모든 절점에서 2개의 횡방향 자유도와 수직축에 대한 1개의 회전 자유도를 보조자유도, 나머지 자유도를 주자유도로 선택해서 LU decomposition을 이용하여 정적축약을 적용

Step 2-3. 각 층에 대해서 구조물 축약 강성행렬을 구성

Step 3. 각 부구조물별 부구조물 강성에 대한 정적축약 단계

Step 3-1. 축약된 층 강성행렬을 조합하여 부구조물의 강성행렬을 구성

Step 3-2. 부구조물에 대하여 구성된 강성행렬 중에서 부구조물의 경계부 절점의 2개의 회전 자유도와 1개의 수직 자유도와 내부 절점 중에서 주절점의 2개의 횡방향 자유도와 1개의 수직축에 대한 회전 자유도를 주자유도, 그 외의 모든 자유도를 보조 자유도로 선택해서 LU decomposition을 이용하여 정적축약을 적용

Step 3-3. 각 부구조물에 대해서 구조물의 축약 강성행렬을 구성

Step 4. 전체 구조물 강성에 대한 정적축약 단계

Step 4-1. 축약된 부구조물 강성행렬을 조합하여 전체 구조물의 강성행렬을 구성

Step 4-2. 전체 구조물에 대하여 구성된 강성행렬 중에서 주절점의 2개의 횡방향 자유도와 1개의 수직축에 대한 회전 자유도를 주자유도, 그 외의 모든 자유도를 보조 자유도로 선택하여 LU decomposition을 이용하여 정적축약을 적용

Step 4-3. 전체 구조물에 대해서 축약 강성행렬을 구성

Step 5. 주절점의 주자유도에 대한 질량행렬을 구성

Step 6. 축약 강성행렬과 집중 질량행렬을 이용하여 고유치해석

이와 같이 단계로 직렬 알고리즘이 구성된다.

직렬 알고리즘 축약단계에서 각 slave 프로세스에 대해서 순차적으로 slave 프로세스수만큼 반복적으로 강성행렬을 축약하는 부분이 병렬화가 가능하게 되는 부분이 된다.

3.2. 병렬 해석법 개발을 위한 병렬 연산 시스템 구현

본 논문에서 개발한 병렬 해석법은 전체 자유도의 크기가 큰 경우에 그 자유도를 여러 개의 작은 부분으

로 자유도를 나누고, 나누어진 작은 문제들을 ethernet network에 연결된 다수의 PC에 분산 할당하여 계산 하도록 구성되어 있다.

병렬 해석법 적용 위한 병렬 연산시스템 모델의 구현은 Windows 98을 operating system으로 사용하는 펜 티엄III 500MHz의 CPU와 128Mbyte 메인 메모리의 PC와 100Mbps의 ethernet network card, 100base-T switching hubs로 구성되어 있다. 또한 연산에 필요한 정보 및 데이터는 Message Passing방식을 적용하고 이때 통신은 WMPI(Message Passing Interface for Win32 Platforms)패키지⁹⁾를 이용한다.

시스템의 구성은 한 대의 master 프로세스를 중심으로 다수의 slave 프로세스들이 master 프로세스에 연결되는 구성을 가지며 그림 1와 같다. 주 연산은 slave 프로세스에서 수행되고 master 프로세스에서 할당된 양만큼 수행하게 되며, 이론적으로 slave들의 연산량은 균등하며 따라서 연산시간도 동일하다.

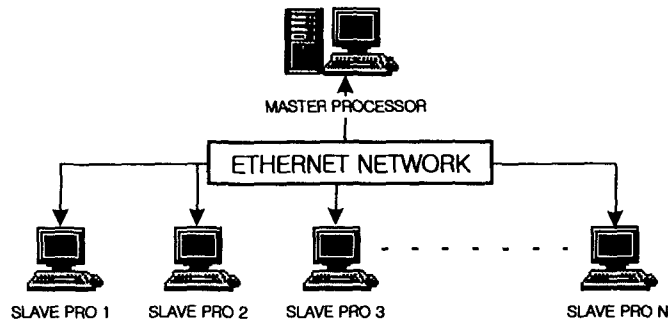


그림 1 병렬 연산 시스템 구성도

3.3 부구조기법을 이용한 병렬해석 알고리즘

기존의 부구조기법은 PC나 워크스테이션의 메모리 제약 등과 같은 저성능을 개선할 목적으로 개발되었으며, 근래에는 PC와 워크스테이션의 고성능화로 인해 그 사용성을 잃고 있다. 그러나, 부구조기법과 분산구조 해석법^{10,11)}을 위한 병렬 연산시스템을 조합하면 기존의 부구조기법에서 부구조물의 수만큼 순차적으로 반복 해석하는 루틴을 분산구조해석법의 각 slave에서 계산되도록 분산 할당하여, 동시에 수행함으로써 직렬해석에 비하여 이론적으로는 전체 해석시간을 1/number of slave의 수로 줄일 수 있다.

3.1절에서 언급한 직렬해석 알고리즘에서 Step 2부터 Step 3까지는 병렬화가 가능한 부분이고, 본 논문에서 대형구조물을 N개의 부구조물로 분산하는 경우 동적해석을 위한 병렬 고유치해석 알고리즘의 흐름도는 그림 2와 같다.

병렬해석¹²⁾ 알고리즘에서는 전체 구조물을 여러 개의 부구조물로 나누고, 나누어진 각 부구조물들을 각 slave에 1:1로 균등하게 분산, 할당하여 동시에 각 부구조물의 해석을 수행한다. Master에서 전체 구조물의 해석에 필요한 데이터를 입력받아서 각 부구조물들이 균등한 자유도를 가지도록 부구조물을 분할하고, 분할된 부구조물의 해석에 필요한 데이터를 WMPI를 이용하여 병렬 연산 시스템을 통하여 slave로 동시에 분산 할당한다. 연결된 각 slave에서는 master로부터 전송받은 데이터를 이용하여 각층, 부구조물순으로 강성행렬에 대한 축약을 동시에 수행하게 된다. 그리고 slave에서의 행렬축약이 완료되면 각 slave에서 master로 축약강성행렬을 전송하면 master에서 주절점에 대해 3.1절에서 언급한 각층 당 3개의 주자유도에 대해서 전체 구조물에 대한 마지막 축약을 하게되고 주자유도에 대한 집중질량(lumped mass)를 산정하여 최종적으로 master에서 고유치해석을 수행하게 된다.

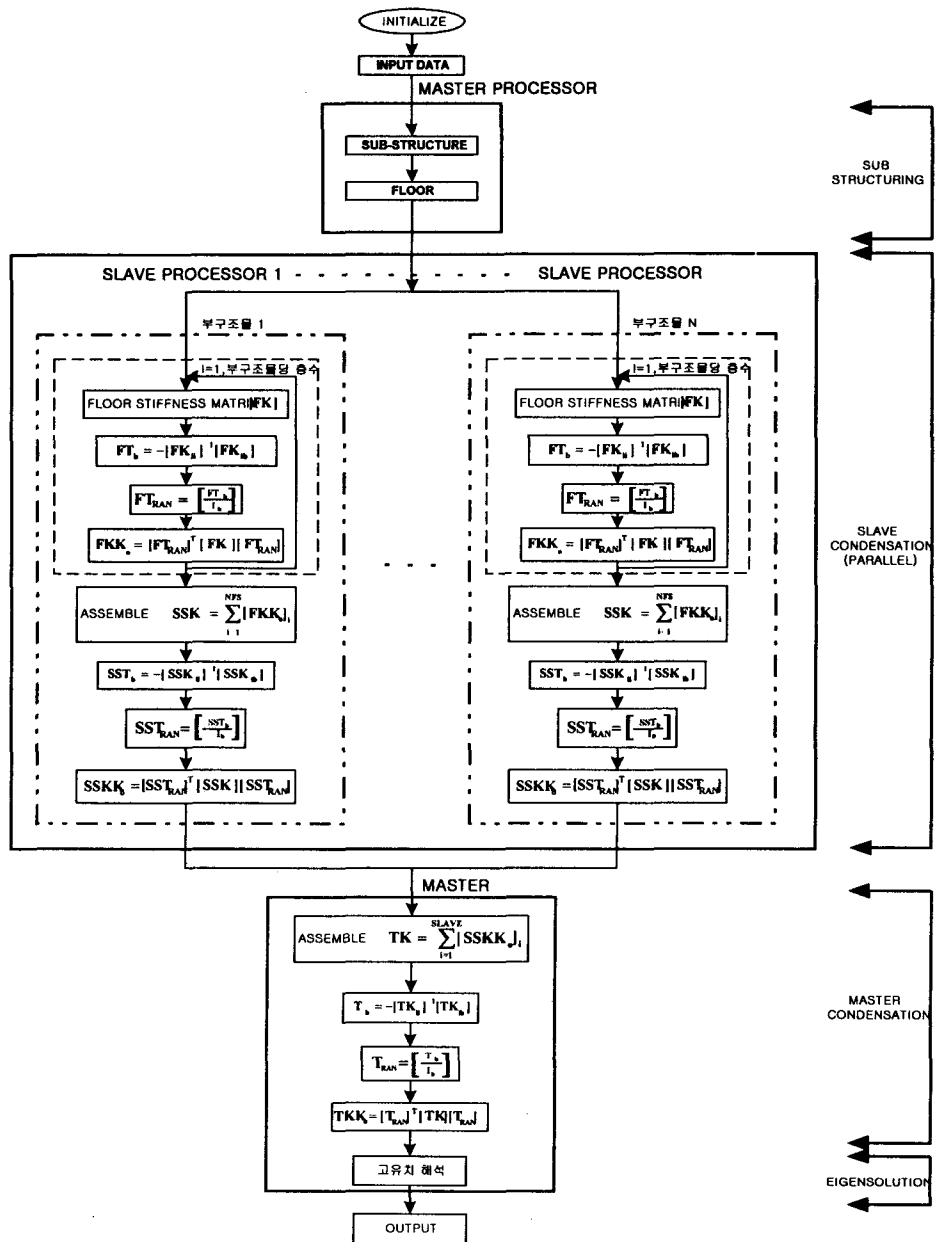


그림 2. 부구조기법을 이용한 병렬해석 흐름도

4. 예제 적용

병렬동적해석알고리즘의 성능분석을 위한 적용예제는 3차원 가새골조의 3×3경간 32층 구조물(전체 자유도수:3168, 전체 부재수:2048, 전체 절점수:528)을 사용하며, 구조물의 질량은 바닥판에 집중된 집중질량을 이용하고, 경간은 8m, 층고는 4m로 하였다.

4.1 slave 프로세스 수의 변화에 따른 전체해석시간

병렬 알고리즘의 slave 프로세스에서는 부구조물내부의 각 층별 보조자유도에 대한 축약과정 후 부구조물에서 다시 한번 부구조물의 보조자유도에 대해서 축약을 실시한다. 전체 구조물의 자유도수를 부구조물수로 나눈 자유도수가 각 부구조물의 전체 자유도가 된다. 따라서, 부구조물 수에 따른 slave 프로세스에서의 해석량은 부구조물이 많아지면 각 부구조물의 자유도수가 작아지게 되고 각 층별 축약을 수행한 후 각 부구조물에 대해서 축약을 하게 되므로 slave 프로세스에서의 해석시간이 감소되는 반면에, master 프로세스에서는 각 부구조물에 대한 축약한 강성행렬들을 조합하고 두 개의 횡방향 자유도와 수직축에 대한 회전 자유도를 주자유도로 선택을 하여 축약을 하는 부분에서는 부구조물의 수가 증가함에 따라서 경계부 자유도가 많아지게 되므로 보조자유도수는 증가하게 되고 주자유도는 각층 당 3개의 자유도로 일정하게 된다.

부구조물의 수가 많아질수록 부구조물의 경계부 자유도수가 증가하게 되고 따라서, master 프로세스에서 조합해야하는 보조자유도수가 증가하게 되므로 강성행렬의 차수가 많아지게 되어 보조자유도의 축약시 많은 해석시간을 요구하게 된다. 최종적으로 축약된 강성행렬과 질량행렬로 고유치해석을 하게된다.

병렬연산시스템에 연결된 slave 프로세스의 수를 2, 4, 8로 변화시키면서 각 프로세스에 1개의 부구조물을 할당하여 병렬해석을 수행한 경우 전체해석시간 및 해석 단계별 소요시간은 표 1과 같다. 그리고 프로세스 수의 증가에 따른 해석시간은 그림 3과 같이 감소함을 알 수 있다.

표 1. 프로세스의 수에 따른 병렬알고리즘의 해석단계별 소요시간

프로세스수 \ 해석단계	2		4		8	
	시간(sec)	비율(%)	시간(sec)	비율(%)	시간(sec)	비율(%)
Input & Substructuring	0.05	0.03	0.06	0.08	0.06	0.12
Send Data	0.48	0.30	0.71	0.92	1.54	3.14
Slave Condensation	150.80	94.32	55.6	71.08	13.15	26.85
Master Condensation	8.42	5.22	20.13	25.94	32.57	66.51
Mass Computation	0.05	0.03	0.05	0.06	0.06	0.12
Eigen Solution	1.60	0.10	1.49	1.92	1.59	3.25
전체 해석시간	161.40	100	77.60	100	48.97	100

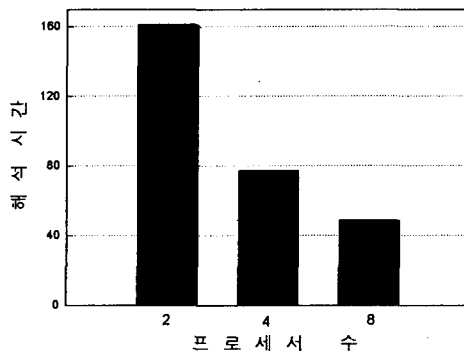


그림 3. 프로세서 수에 따른 해석시간

4.2 부구조물수에 따른 병렬 알고리즘의 각 해석단계별 소요시간

일반적으로 가상의 병렬 시스템을 구성하는데 소요되는 시간은 slave 프로세스의 수가 증가함에 따라서 증가하는 경향을 보이거나, 병렬 시스템을 구성하는 시간은 해석에 직접적으로 소요되는 시간이 아니므로 전체 해석시간에서 제외시켰다.

표 1에서 각 해석단계별 시간을 보면 slave condensation 단계에서는 부구조물의 수가 증가함에 따라서 반 비례하게 소요시간이 감소됨을 볼 수 있다. 이는 부구조물 수가 증가함에 따라서 각 부구조물의 자유도크기가 줄어들고 따라서 각 slave 프로세스에서 축약해야하는 자유도수가 감소하게 되므로 해서 slave 프로세스에서 소요되는 시간이 줄어들게 된다. 그러나, master condensation 단계의 소요시간은 부구조물의 수가 증가함에 따라서 비례적으로 증가하는 양상을 나타낸다. 이는 각 slave 프로세스에서 연산된 부구조물의 축약된 강성행렬을 master 프로세스에서 조합하여서 다시 master condensation 단계를 거치는 과정에서 각 부구조물에서 발생하는 경계부 자유도가 master 프로세스에서 보조자유도수의 증가가 되고 따라서 master condensation에서 소요되는 시간을 증가시키는 요인이 된다.

4.3 Speed-up(성능향상) 분석

본 연구에서 개발된 병렬 알고리즘의 성능은 프로세스 수의 증가에 따른 해석시간의 감소정도를 평가하기 위하여 프로세스가 2개인 경우를 기준으로 식(8)과 같이 비교하면 그림 4와 같다.

$$SP = \frac{T_{pn}}{T_{p2}} \quad (8)$$

여기서 T_{p2} 와 T_{pn} 은 각각 slave 프로세스의 수가 2 개 및 n개를 사용한 경우의 해석시간을 나타낸다.

그림 4에서 볼 수 있듯이, 이론적으로 SP의 곡선에 비해서 부구조물의 수가 증가할수록 SP가 점차 감소함을 알 수 있다. slave 프로세스의 수가 각각 4, 8개일 경우에 2.08과 3.3 정도의 SP를 나타내고 있다. 이것은 4.3절에서 언급했듯이 각 slave 프로세스의 수가 증가함에 따라서 부구조물의 경계부 자유도수가 증가하게 되고 이로 인해서 master에서 축약해야하는 보조 자유도수가 증가하게 된다. 따라서 master의 축약시간이 길어지게 됨으로서 master 프로세스의 해석시간이 상대적으로 길어지게 됨에 따라 SP가 감소하는 현상을 나타낸다. 따라서 전체적인 해석시간은 master 프로세스에서의 축약시간과 slave 프로세스에서의 축약시간이 대부분을 차지하게 된다. 이로 인해서 병렬 알고리즘에서의 성능평가에서는 부구조물수가 증가함에 따라서 부구조물에 대한 축약과정은 slave 프로세스가 계산하므로 병렬성능을 증가시키는 반면에 master 프로세스에서는 축약과정에서 축약해야하는 보조자유도수가 증가할수록 병렬성능을 감소시키는 원인이 된다.

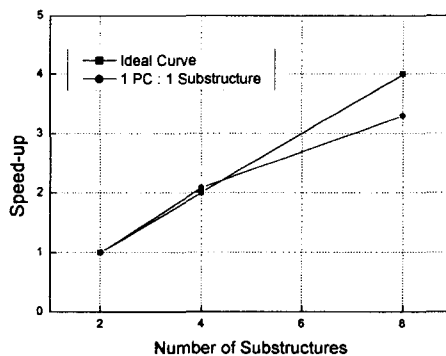


그림 4 Speed-up

5. 결 론

3차원 대형 구조물의 동적해석을 효율적으로 수행하기 위한 병렬 해석 알고리즘에 대한 결론을 다음과 같이 얻을 수 있다.

- (1) 전체 구조물을 부구조기법을 이용하여 각각의 부구조로 나누고 병렬연산시스템에 연결된 각각의 컴퓨터에 할당하여 부구조물의 축약을 동시에 수행하도록 병렬알고리즘을 개발하였다.
- (2) 개발된 알고리즘에 의하여 대형구조물의 동적해석에 요구되는 고유치문제의 크기를 다수의 컴퓨터를 이용하여 효과적으로 줄여 동적해석에 소요되는 slave 프로세스의 축약시간을 감소시켰다.
- (3) 적용된 예제의 경우 사용하는 컴퓨터의 수가 증가하면 1대의 컴퓨터에서 해석하는 부구조물의 자유도수에 비해서 master 프로세스에서 축약하는 경계부 자유도수가 비례적으로 증가하여 병렬알고리즘의 성능이 감소하는 경향이 나타난다. 따라서, 본 예제보다 자유도수가 더 많은 예제를 적용할 경우에는 slave 프로세스에서 해석하는 자유도수가 증가하게 되고 master에서 축약하는 경계부 자유도수가 상대적으로 감소할 경우에는 병렬성능이 더 증가할 것으로 판단된다.
- (4) 본 연구에서 개발한 동적해석을 위한 병렬 알고리즘에서는 부구조물수와 slave 프로세스수에 따라서 해석시간의 차이를 보이기 때문에 전체 자유도수와 경계부 자유도수에 따른 가장 최적의 부구조물수와 slave 프로세스수의 파악하는 연구가 병행되어야 할 것으로 보인다.

감사의 글

본 논문은 1999년도 한국학술진흥재단의 연구비에 의하여 연구되었음. (KRF-99-041-E00598)

참고 문헌

1. L. Baker and B. J. Smith, "Parallel Programming", McGraw-Hill, 1997
2. H. Adeli and H. S. Park, Neurocomputing for Design Automation, CRC Press, 1998.
3. J. S. Przemieniecki, "Matrix Structural Analysis of Substructures", AIAA Journal, Vol. 1, No. 1, pp. 138-147, January 1963
4. A. K. Noor, H. A. Kamal and R. E. Fulton, "Substructuring Techniques - Status and Projection", computer & Structures, Vol. 8, pp. 621-632, 1978
5. R. J. Guyan, "Reduction of Stiffness and Mass Matrices", AIAA, Vol. 3, No. 2, pp. 380. 1965
6. W. Weaver, Jr. and P. R. Johnston, "Structural Dynamics by Finite Elements", Prentice-Hall INC., 1987
7. 김재홍, 성창원, 박효선, "부구조 기법을 이용한 병렬 고유치해석 알고리즘 개발", 한국전산구조공학회 발표논문집, 제19권, 제2호, pp. 411-418, 1999년, 10월
8. 성창원, 박효선, "분할 방법에 따른 분산구조 해석법의 성능분석", 한국전산구조공학회 발표논문집, 제12권, 제1집, pp. 50-57, 1999년 4월
9. Message Passing Interface Forum, "MPI-2 Journal of Development", University of Tennessee, July, 1997
10. 성창원, 박효선, "PC 네트워크 시스템을 이용한 대형구조물의 분산구조해석법", 대한건축학회 논문집, 제15권, 7호, pp. 21-28, 1999년 7월
11. 박효선, 박성무, 성창원, 김재홍, "부구조기법을 이용한 PC level 분산구조해석법", 한국전산구조공학회 발표논문집, 제11권, 제2집, pp. 53-60, 1998년 10월
12. 박효선, "대형 구조물을 위한 병렬 구조해석 및 설계, 전산구조공학회지", 제9권, 제3호, pp. 47-53, 1996년 9월