

분산구조해석을 이용한 구조설계최적화 Structural Design Optimization using Distributed Structural Analysis

박종희* 정진덕** 전한규** 황진하***
Park, Jong-Hoi Jung, Jin-Duck Jeon, Han-Gyu Hwang, Jin-Ha

ABSTRACT

Distributed processing approach for structural optimization is presented in this study. It is implemented on network of personal computers. The validity and efficiency of this approach are demonstrated and verified by test model of truss. Repeated structural analysis algorithm, which spend a lot of overall structural optimization processes, are based on substructuring scheme with domain-wise parallelism and converted to be adapted to hardware and software environments.

The design information data are modularized and assigned to each computer in order to minimize the communication cost. The communications between nodes are limited to static condensation and constraint-related data collection.

1. 서론

계획으로부터 해석, 설계 등으로 이루어지는 구조공학 프로젝트에서 설계된 구조에 대한 수학적 모델을 통해 거동을 산정하는 비교적 단선적인 해석과정에 비해 선택의 문제인 설계과정은 해석과 총합의 반복 결합으로 수행되는 총체적 작업이며, 오늘날 해석이나 설계 등, 구조공학의 여러 과정들은 전산환경에서 수행된다.

또한 전산과학 분야의 급속한 발전에 힘입어 공학계에서도 구조물에 대한 설계자동화의 노력이 꾸준히 이어져 왔으며 구조최적화도 이것을 이루기 위한 하나의 방법론이다. 구조최적화는 요구된 설계조건을 만족시키면서 최대의 경제성을 갖는 구조계를 도출하는 과정이며, 실행측면에서 해석과 최적화모듈의 반복결합으로 이루어지는 전산과정이다.

그러나 구조물이 대형화되고 복잡해짐에 따라 대형시스템을 최적화하는 문제는 대단히 많은 설계변수와 제한조건을 갖기 때문에 전산처리 시 상당한 메모리와 시간을 필요로하고 소형컴퓨터 상에서는 실행키 어려울 뿐만 아니라 대형컴퓨터 상에서도 많은 비용이 요구된다.

* 정회원 · 충북대학교 강사

** 충북대학교 구조시스템공학과 석사과정

*** 정회원 · 충북대학교 구조시스템공학과 교수

특히 구조최적화에서 대부분의 CPU 시간을 반복되는 거동해석과 민감도해석에 사용하므로 이들에 드는 소요시간 및 그에 따른 비용이 전체 과정에 미치는 영향은 대단히 크며, 공학전반환경 또한 여러 전문영역의 병렬처리와 통합화방향으로 흐르고 있어 보다 강력하고 경제적인 전산자원을 필요로 한다.

이러한 점들을 해결하기 위한 대안으로 80년대 중반이후 전산과학분야에서 활발히 이루어지고 있는 병렬처리 방법을 도입, 구조공학 분야에서 슈퍼컴퓨터나 병렬컴퓨터를 활용하는 방법이 국내·외에서 활발히 연구되고 있다. 그러나 이러한 방법들은 대부분 고가의 병렬컴퓨터 상에서 이루어졌기 때문에 특수한 목적이 아닌 경우 접근이 용이하지 않았다. 이러한 문제를 해결하기 위한 방안으로 최근에 경제적인 측면을 고려하여 네트워크로 연결된 컴퓨터들을 이용하는 분산처리 방법이 주목받고 있다.

분산처리란 독립된 컴퓨터들을 네트워크로 엮어 하나의 작업을 각 컴퓨터에서 나누어 계산하도록 하는 것을 말한다. 분산처리가 갖는 가장 큰 장점은 병렬처리처럼 고가의 장비없이도 기존의 장비를 활용하여 대형의 문제를 최소한의 비용으로 풀 수 있다는 점이다.

본 연구에서는 이러한 분산처리기법을 구조최적화문제에 적용, 수없이 반복되는 설계 및 해석과정의 효율성을 높이하고자 한다. 분산환경은 WINDOWS 98를 플랫폼으로 PC 5대를 네트워크로 묶어 구축하였으며, 처리방법으로는 영역단위 병렬성을 갖는 부구조화 알고리즘에 기초한 프로그램을 구성하고 효율성을 검증하였다.

2. 분산환경과 연산

분산처리란 넓은 의미에서는 병렬처리의 일부분이지만 특히 서로 관련된 여러 개의 작업이 각각 여러 개의 클라이언트노드로 나누어져서 각자 일을 수행하는 것을 말한다. 이 방법은 독립된 컴퓨터들을 병렬처리 컴퓨터의 한 연산요소로 보고 이들을 소프트웨어적으로 하나의 병렬처리 컴퓨터로 묶어 일종의 가상 운영체제하에 작업을 수행한다.

이러한 역할을 하는 PVM(parallel virtual machine)은 가능한 한 범용성과 유연성을 지니기 위해 개별메모리를 갖는 MIMD형 병렬컴퓨터를 기본모델로 하여 설계되었고 서로 다른 기종의 컴퓨터들을 엮어 하나의 가상적인 병렬형 컴퓨터처럼 동작하게 해주는 일종의 가상 운영체제(virtual operating system)이며 다양한 언어로 프로그램을 코딩할 수 있다.

PVM의 구성은 크게 데몬과 라이브러리로 나눌 수 있는데, 데몬은 각 컴퓨터에 설치되어 이들을 하나의 가상적인 컴퓨터로 구성한다. 라이브러리는 사용자의 프로그램 내에서 호출되어 각 컴퓨터간의 정보교환 및 작업할당 등을 가능하게 해준다. 이것은 데몬 프로세스인 pvmd3 라는 프로그램과 각 컴퓨터간의 정보교환 및 작업할당을 가능케 해주는 통신 함수들의 라이브러리인 libpvm3.a로 구성된다.

PVM은 두 개의 주된 프로그래밍 모델을 갖는데, 먼저 주종모델(master-slave model)은 하나의 통제프로그램이 마스터 프로세서로 다른 모든 프로세서들을 생성하며, 초기 입력데이터의 전달 및 계산결과의 수집과 표현 등을 맡아서 처리하며, 슬레이브 프로세서는 마스터프로세서의 통제를 받아 실제의 계산과정을 수행하는 모델이다. 나머지는 호스트가 없는 모델로서 하나의 프로세서가 모든 다른 프로세서들을 생성시키고 작업이 끝날 때까지 통신과정 없이 주어진 문제에 대한 각 프로세서 자신의 역할을 수행하며, 요구되는 입·출력은 개별 프로세서상에서 수행할 수 있다. 본 연구에서는 상기 두가지 모델 중 주종모델을 사용하며, 주 프로세서는 통제프로그램으로서의 역할뿐만 아니라 슬레이브 프로세서와 같이 실제의 계산까지 수행한다.

3. 문제형성 및 알고리즘

전체 구조시스템의 총중량이나 경비를 목표함수로, 구조해석을 통해 산정되는 거동량 등에 관한 등식 또는 부등식 제한조건으로 구성된 구조최적화 문제는 다음과 같이 정의된다.

$$\text{Min. } F(X) \quad (1)$$

$$H(X) = 0 \quad (2)$$

$$G_j(X) \leq 0, \quad j=1, 2, \dots, N_c \quad (3)$$

$$X_n^L \leq X_n \leq X_n^U, \quad n=1, 2, \dots, N_v \quad (4)$$

식 (2)는 등식 제한조건으로서 전체 구조시스템에 대한 평형방정식으로 주어진다. 식 (3)과 식 (4)는 부등식 제한조건으로서 거동제한조건과 설계변수크기 제한조건을 들 수 있다. 거동제한조건은 변위조건, 응력조건, 고유치조건 등이 있으며 설계변수크기 제한조건은 선형부재의 단면치수에 대한 제한조건, 판의 최소두께에 대한 제한조건 및 셸의 최대높이에 대한 제한조건 등이 이에 속한다.

일반적으로 해석과 총합의 반복결합으로 이루어지는 구조최적화과정은 대부분의 계산시간을 거동해석과 민감도해석에 보내게 되는 바, 본 연구에서는 거동해석과정을 분산처리함으로써 전체설계과정의 효율성을 제고토록 하였다¹⁾.

세부적으로 마스터에 해당하는 컴퓨터는 해석모듈 실행 시 각 슬레이브 컴퓨터들로부터 전달되는 데이터 취합, 합성 및 연산과 최적화모듈을 담당하고 모든 과정을 제어하게 된다. 나머지 컴퓨터들은 문제 형성 시 분할된 부구조들을 각각 할당받게 된다. 마스터노드의 실행단계는,

- Step 1 : 슬레이브 컴퓨터들을 초기화시키고 각 슬레이브컴퓨터에 태스크 ID를 부여한다. 이때 각 슬레이브 컴퓨터들은 데이터모듈을 통해 필요한 설계정보를 입력받게 된다.
- Step 2 : 초기 설계변수로 구조해석을 수행하며 각 슬레이브노드로부터 해당 부구조의 응축된 경계강성도매트릭스와 경계하중벡터를 전달받고, 유효경계강성도매트릭스와 유효경계하중벡터로 합성한다.
- Step 3 : 응축된 시스템방정식을 풀어 경계변위를 산정한 뒤 해당 슬레이브노드로 이들 데이터를 전송한다. 이때 각 슬레이브 컴퓨터는 이들로부터 해당 부구조의 내부변위와 응력값을 산출한다.
- Step 4 : 각 슬레이브노드에서 산출된 거동제한조건과 목표함수값을 전달받아 전체구조에 할당하도록 이들을 조합하고 재구성한 뒤 최적화모듈로 전달한다.
- Step 5 : ALM을 이용하여 최적화를 수행, 설계변수를 수정한다. 이때 주어진 수렴조건을 만족하면 각 슬레이브 컴퓨터를 종료시키게 되며 그렇지 않다면 새로운 설계변수값을 해당 슬레이브노드로 전달한 뒤 단계 1로 피드백하게된다.

4. 적용 및 분석

본 연구에서 제시한 알고리즘의 타당성 및 효율성을 검증하기 위해 상기한 분산환경에서 펜티엄급 IBM 호환 PC를 5대까지 연결하여 200 부재 트러스구조에 대해 수치실험을 수행하였다. 분산처리에 사용된 개인용 컴퓨터 5대가 각기 다른 성능을 지니고 있기 때문에 마스터와 슬레이브를 다양하게 선택하여 수치실험을 수행하였으며, 최종구성은 마스터서버로 펜티엄 II 350 Mhz(RAM 64M)를, 슬레이브 노드로 사용된 클라이언트로 2번 부구조를 펜

티엄 II 330 Mhz(RAM 64M), 3번 부구조에 펜티엄 II 266 Mhz(RAM 64M), 4번과 5번 부구조 AMD K62 300Mhz(RAM 32M)에 할당하였다. 마스터서버의 캐쉬메모리는 512 Kbyte 이고 슬래이브 프로세서들은 256 Kbyte이다.

수치실험결과는 단일 컴퓨터상에서 수행한 전체구조에 대한 순차최적화과정(method I)과 분산구조해석을 이용한 최적화과정을 비교·분석하였으며, 각 방법에서 최적화알고리즘은 Augmented Lagrange Multiplier방법을 이용하였다.

분산처리의 성능은 고속화배수와 효율성으로 평가할 수 있는데, 프로세서 증가시의 고속화배수 S_p 와 효율성 E_p 는 전체구조를 하나의 프로세서로 처리한 시간과 이를 여러 개의 프로세서로 처리했을 때의 시간과의 비를 나타내며 각각 다음과 같이 표현한다.

$$S_p = \frac{T_1}{T_p} \quad (5)$$

$$E_p = \frac{S_p}{P} \quad (6)$$

이 식은 Amdahl의 법칙을 기초로 병렬구조해석에 Fulton 등²⁾이 적용한 식이며 여기서, T 는

표 1. 설계자료

Modulus of elasticity	: 30,000.0 kips
Allowable stress limit	: ± 30.0 kips
Allowable displacement	: ± 0.5 in
Specific weight	: 0.283 lb/in^3
Lower limit on area	: 0.10 in^2
Upper limit on area	: None
No. of design variable	: 200

Loading condition	Node No.	Load(kips) in direction		
		x	y	z
1	1, 6, 15, 20, 29, 34, 43, 48, 57, 62, 71	1.0	0.0	0.0
	1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, ..., 71, 72, 73, 74, 75	0.0	-10.0	0.0

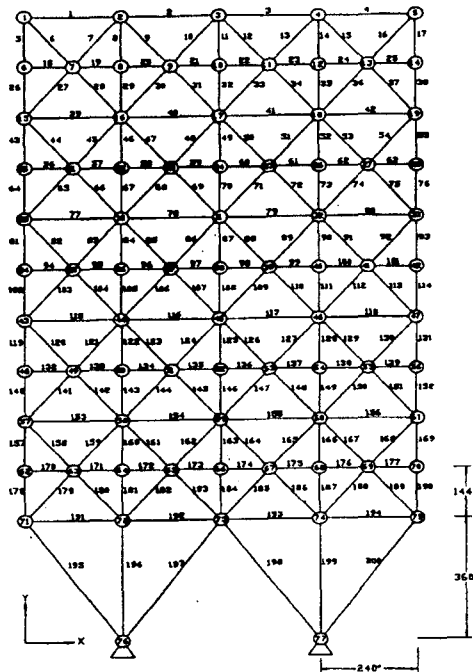


그림 1. 200 부재 평면트러스

표 2. 목표함수 값

Algorithm	Data	Objective function (lb)
Method I		25,708
Present		25,693
Reference		25,874

표 3. 분산처리 수행결과

No. of Substructures	CPU	speed up	efficiency
1	1 CPU (PC)	1	-
2	2 CPU (PC)	2.39	1.2
3	3 CPU (PC)	3.11	1.03
4	4 CPU (PC)	3.99	0.99
5	5 CPU (PC)	4.92	0.98

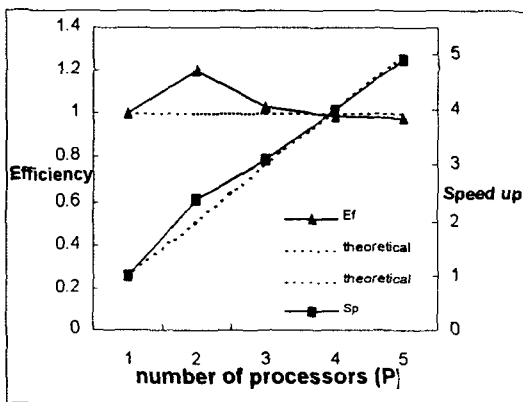


그림 2. 고속화배수 및 효율성

단일 노드 또는 분산처리 된 프로그램의 전체 실행시간을 나타내며, P는 분산처리에 사용된 프로세서 또는 개인용 컴퓨터의 갯수이다.

시험모델로 사용된 200개의 부재를 갖는 트러스구조는 그림 1과 같으며, 설계자료는 표 1에, 본 연구에서 제시한 방법의 목표함수 값은 표 2에서 비교하여 제시하였다.

그림 1의 트러스를 2개에서 5개의 부구조로 영역분할하고 1대부터 5대까지의 컴퓨터상에서 구조최적화를 수행하였으며, 단일 PC상의 결과를 기준으로 분산처리상의 연산효과를 식 (5)와 (6)을 이용, 고속화배수와 효율성을 표 3에 나타내었다.

2대와 3대의 컴퓨터를 활용하여 분산처리한 경우, 고속화배수 및 효율성이 이론치보다 좋은 결과를 나타내고 있다. 이것은 수 없이 반복되는 거동해석과정의 분산처리에서 오는 효과이며, 3개 ~ 5개의 부구조로 영역분할된 경우에 네트워크로 연결된 컴퓨터의 수가 늘어날수록 상대적으로 고속화배수와 효율성이 떨어지는 것은 병렬 컴퓨터상에서도 일반적으로 나타나는 현상으로 주원인은 자료송·수신의 성능과 관련된다. 그러나 이 결과는 전자의 경우에 비해 상대적 효율성이 떨어질 뿐이며 거의 이론치에 근접한 결과를 나타내고 있다.

그림 2는 200 부재를 갖는 트러스구조의 고속화배수 및 효율성을 그래프화로 나타내었다.

5. 결론

고성능 컴퓨터의 활용이나 최근 병렬 및 분산처리에 대한 활발한 연구는 단일프로세서가 갖는 연산처리의 한계를 극복하기 위한 자연스런 대응이며 그런 점에서 네트워크로 연결된 일반 PC들을 이용하는 분산처리방법의 활용 또한 바람직한 방향이라 할 수 있다.

본 연구는 기존 컴퓨터 자원을 활용하여 구조최적화과정에서 수없이 반복되는 해석과정을 분산처리하여 전체 최적화과정의 효율성을 높이도록 하였다. 세부적으로 거동해석의 분산처리 알고리즘은 영역단위의 병렬성을 갖는 부구조화에 근거하고 하드웨어 및 소프트웨어 설정에 맞춰 변환 재구성었으며, 표준모델에 적용한 수치실험 결과 충분한 가능성 및 효율성을 보여준다.

6. 참고문헌

1. 황진하, 고덕구, 박종희, "부구조화에 기초한 매트릭스구조 해석의 병렬/분산처리", *대한토목학회*, 제20권, 제1-A호, 2000년, 1월, pp. 115-124
2. S. Y. Synn and R. E. Fulton, "Practical Strategy for Concurrent Substructure Analysis", *Computer & Structures*, Vol. 54, No. 5, 1995, pp. 939-944.
3. G. Yagawa, N. Soneda and S. Yoshimura, "A Large Scale Finite Element Analysis Using Domain Decomposition Method on Parallel Computer", *Computers & Structures*, Vol. 38, No. 5/6, 1991, pp.615-625
4. J. Padovan and A. Kwang, "Hierarchically Parallelized Constrained Nonlinear Solvers with Automated Substructuring", *Computers & Structures*, Vol. 41, 1991, pp.7-33
5. H. Adeli and O. Kamal, "Concurrent Analysis of Large Structures- I. Algorithms", *Computers & Structures*, Vol. 42, 1992, pp.413-424
6. O. Storaasli and P. Bergan, "A Nonlinear Substructuring Method for Concurrent Processing Computers." *AIAA Journal*, Vol. 25, 871-876 (1987).
7. M. E. M. El-Sayed and C. K. Hsiung, "Optimum Structural Design with Parallel

Finite Element Analysis", Computers & Structures, Vol. 40, 1991, pp. 1469-1474.

8. 박효선, 성장원, "분할 방법에 따른 분산구조 해석법의 성능분석", 한국전산구조공학회 발표논문집, 제12권, 제1집, 1999년, pp. 50-57.
9. G. Al and B. Adam, *et. al*, "PVM: A Users's Guide and Tutorial for Networked Parallel Computing", The MIT Press, Cambridge, London, England, 1994.