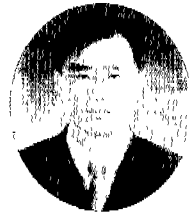


## 리눅스 클러스터와 대형병렬연산시스템에 의한 병렬 구조해석 기법



권 윤 한\*



박 호 선\*\*

### 1. 서 론

최근, 공학에서 다루고 있는 문제들의 복잡화, 정밀화로 인해 컴퓨터 계산속도에 대한 요구가 날로 증가하고 있고 이를 해결하기 위한 방법 중의 하나가 슈퍼컴퓨터를 사용하는 것이다. 슈퍼컴퓨터를 사용하는 가장 큰 이유는 일정 시간 안에 대용량 계산을 정확하게 계산하기 위함이다. 물론 하드웨어의 발전에 힘입어 개인용 컴퓨터들의 성능이 고성능화 되고 있으나 단순히 컴퓨터를 개인적 목적이 아닌 발전적인 연구 목적에 활용하고자 할 경우, 아직도 그 성능은 만족스럽지 못한 실정이다. 이에 고성능 슈퍼컴퓨터를 필요로 하게되었으나 그 비용이 만만치 않아 대기업이 아니고는 도입하는데 어려움이 많다. 최근 이에 대한 해결책으로 저비용으로 슈퍼컴퓨터 이상의 성능을 낼 수 있는 클러스터링 기술이 인기를 끌고 있다. 특히 리눅스 클러스터링에 대한 관심이 크게 늘고 있는데 본 기술기사에서 이러한 리눅스 클러스터 시스템의 개발과 구조해석 알고리즘의 개발을 대형병렬연산시스템(CRAY-T3E900)의 구조해석 성능과

비교하여 분석하고자 한다.

### 2. 병렬연산시스템 및 프로그래밍 모델

컴퓨터 계산 속도에 대한 요구는 날로 늘고 있으나, 단순히 컴퓨터를 개인적 목적이 아닌 발전적인 연구 목적에 활용할 경우, 아직도 그 성능은 만족스럽지 못한 실정이다. 이에 고성능 슈퍼컴퓨터를 필요로 하게되었으나, 그 비용이 고가여서 사용성이 저하되고 있는 실정이다. 최근 이에 대한 해결책으로 저비용으로 슈퍼컴퓨터 이상의 성능을 구현할 수 있는 클러스터링 기술에 대한 관심이 증대되고 있다.

따라서 대형구조물의 효율적인 구조해석을 위해 본 연구에서는 이러한 고성능의 연산성능이 구현 가능한 리눅스 클러스터링 시스템을 제작하였으며 다음과 같다.

#### 2.1 리눅스 클러스터 시스템

리눅스 클러스터링에 사용되는 리눅스 시스템과

\* 연세대 건축학과, 박사과정

\*\* 연세대 사회환경·건축공학부 조교수, 공학박사

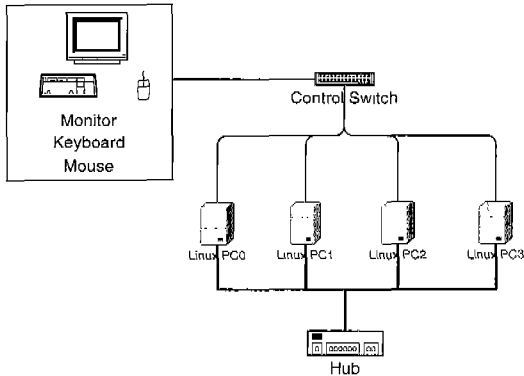


그림 1 리눅스 클러스터링 시스템

각종 소프트웨어들이 무료로 사용가능하고 하드웨어 또한 개인용 컴퓨터들로 구성될 수 있기 때문에 저렴하게 구축이 가능하다. 본 연구에서 구축된 리눅스 클러스터 시스템 환경은 그림 1과 같다.

시스템의 구성은 각 한 개의 모니터, 키보드, 마우스 그리고 제어스위치로 이루어진 제어부분, 펜티엄3-600Mhz의 CPU, 256Mbyte의 메인메모리와 하드디스크로 이루어진 4대의 본체인 주연산시스템 부분, 그리고 100Mbps 네트워크 카드와 100Mbps 스위칭 허브로 구성된 네트워크부분으로 크게 세 부분으로 구성된다.

### 2.2 고성능 병렬연산기

고성능 병렬 연산기 시스템은 Cray Research에서 개발한 고성능 병렬컴퓨터인 CRAY T3E이며 총 128개의 User PE와 8개의 Support PE로 구성되어 있다. Cray T3E 노드(Node)는 1개의 PE(Processing Element)와 Network router로 구성되어 있으며 각 PE의 성능은 450MHz, PE당 메모리는 128MBytes이며, 통신성능은 최대 500Mbytes/sec 정도이다. Cray T3E 시스템의 8개 노드마다 Globally-accessible I/O 채널로 연결 가능한 네트워크 Connection이 1개씩 장착되어 있고, 이 2개의 I/O 채널은 또한 1개의 GigaRing Channel을 형성함으로써 16PE당 하나의 GigaRing이 존재하며, GigaRing들이 인접한 PE들에만 속한 것이 아니고, 시스템 전체에 Global하기 때문에 I/O부하가 네트워크상의 특정 위치로 집중되는 것을 방지한다.

### 2.3 병렬프로그래밍 모델

대형 병렬연산기와 리눅스 클러스터 시스템의 병렬프로그램에 사용된 Message-Passing Toolkit으로는 PVM(Parallel Virtual Machine)-3을 사용하였으며, 프로그램 모델은 Master-Slave 모델이며 사용하고자하는 노드(PE)의 수에 따라 연산량이 서로 균형이 되도록 데이터량을 자동분할되도록 프로그래밍 하였다.

Master 노드는 초기 데이터의 입력, 작업량 분할, 결과 출력 등과 더불어 Slave 노드가 수행하는 주연산 부분도 담당하게 되며 Slave 노드는 Master 노드로부터 각종 구조물 데이터들을 전송받아 독자적으로 부재 강성매트릭스를 구성한 후 인접 Slave 노드와 Message Passing을 통해 필요한 정보를 교환하면서 각종 계산을 담당하게 된다.

### 3. PCG 알고리즘

본 기사에서 분산구조해석법에 적용될 PCG 알고리즘은 제어매트릭스(preconditioning matrix)와 기본적인 공액경사도법(conjugate gradient method)의 조합으로 이루어졌으며, 부재단위로 해석을 수행하므로 적은 저장량만을 필요로 하기 때문에 대형 구조물의 구조해석 문제에 있어서 계산상 효율적이며, 알고리즘 내의 대부분의 계산들이 매트릭스와 벡터의 곱셈들로 이루어졌기 때문에 직접법보다 병렬화가 용이하다. 이러한 이유에 의해 대형 구조물의 구조 방정식을 풀기에 매우 적합하다는 것이 많은 연구에 의해 증명되었다.

$Kx = f$ 와 같은 구조방정식해법을 위한 PCG 알고리즘은 변위벡터  $x_i$ 를 반복적으로 계산하여 잔차벡터  $r_i = f - Kx_i$ 가 허용오차 이내로 수렴하게 될 때 종료된다. 변위벡터의 수렴에 소요되는 반복회수를 줄이기 위한 방안으로 제어매트릭스를 이용하며 일반적인 부재단위의 PCG 알고리즘의 흐름도는 다음과 같다.

step 1) 제어매트릭스 구성  $C = M^{-1}$

- step 2) 초기값  $\mathbf{x}_0$  선택  
 $\mathbf{r}_0 = \mathbf{f} - \mathbf{K}\mathbf{x}_0$  계산  
 $\mathbf{h}_0 = \mathbf{C}\mathbf{r}_0$  계산  
 $\mathbf{p}_0 = \mathbf{h}_0$   
**for**  $i=0, 1, 2, \dots$ (수렴할 때까지 반복)
- step 3)  $\alpha_i = \frac{\mathbf{r}_i \cdot \mathbf{h}_i}{\mathbf{K}\mathbf{p}_i \cdot \mathbf{p}_i}$
- step 4)  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{K}\mathbf{p}_i$   
 $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$   
 $\mathbf{h}_{i+1} = \mathbf{C}\mathbf{r}_{i+1}$
- step 5)  $\frac{\mathbf{r}_{i+1} \cdot \mathbf{h}_{i+1}}{\mathbf{r}_0 \cdot \mathbf{h}_0} \leq \eta$  then stop,  
 else continue
- step 6)  $\beta_i = \frac{\mathbf{r}_{i+1} \cdot \mathbf{h}_{i+1}}{\mathbf{r}_i \cdot \mathbf{h}_i}$
- step 7)  $\mathbf{p}_{i+1} = \mathbf{h}_{i+1} + \beta_i \mathbf{p}_i$  (반복계속)

여기서  $\mathbf{K}$ 는 부제단위 강성매트릭스이며,  $\mathbf{p}_i$ 는 공액 탐색 방향,  $\mathbf{r}_i$ 는 잔차벡터,  $\mathbf{x}_i$ 는 변위벡터,  $\mathbf{f}$ 는 하중 벡터,  $\mathbf{h}_i$ 는 임시벡터이고,  $\alpha_i$ 와  $\beta_i$ 는  $\mathbf{p}_i$  방향으로 이동 거리를 제어하는 스칼라 값이며,  $\eta$ 는 수렴오차이다.

$\mathbf{K}\mathbf{x} = \mathbf{f}$ 와 같은 구조방정식은 제어 매트릭스  $\mathbf{M}$ 을 양쪽 항에 미리 곱함으로써 식(1)과 같은 PCG 알고리즘의 방정식이 구성된다.

$$\mathbf{M}^{-1} \mathbf{K}\mathbf{x} = \mathbf{M}^{-1} \mathbf{f} \tag{1}$$

제어매트릭스  $\mathbf{M}$ 은 대칭·양정(positive symmetric) 매트릭스이고 강성매트릭스  $\mathbf{K}$ 로부터 구해지며, 본 기사에서는 여러 가지 제어매트릭스 중 수렴성과 계산량의 관점에서 적절하며 구성이 쉬운 대각 제어매트릭스를 사용하였다. PCG 알고리즘 내에서 제어매트릭스  $\mathbf{M}$ 과 잔차 벡터  $\mathbf{r}_0$ 의 부가적인 계산이 필요하지만 제어매트릭스를 도입함으로써 전체 알고리즘의 수렴성을 향상시키는데 도움이 된다.

병렬 PCG 알고리즘과 구별하기 위해 일반적인 PCG 알고리즘을 순차 PCG(Sequential PCG; SPCG)

알고리즘이라 칭한다.

#### 4. 병렬 PCG 알고리즘

부제 단위의 SPCG 알고리즘에서 1회 반복시의 계산량은 크게 하나의 매트릭스-벡터 계산( $\mathbf{K}\mathbf{p}_k$ )과 2개의 내적 계산( $\mathbf{p}_k^T \cdot (\mathbf{K}\mathbf{p}_k)$ 과  $\mathbf{r}_{k+1}^T \cdot \mathbf{z}_{k+1}$ ), 그리고 세 개의 벡터계산( $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ ,  $\mathbf{r}_{k+1} = \mathbf{r}_k + \alpha_k [\mathbf{K}]\mathbf{p}_k$ ,  $\mathbf{p}_{k+1} = -\mathbf{z}_{k+1} + \beta_k \mathbf{p}_k$ )으로 구성된다. PCG 알고리즘의 계산량 중에서 매트릭스-벡터 계산이 대부분을 차지하므로 알고리즘의 판단 기준이 될 수 있으며, 병렬연산에 있어서 이 부분을 어떻게 분할하여 계산하느냐에 따라 병렬구조해석법의 성능이 좌우된다.

개발된 병렬 구조해석 알고리즘이 기존의 수치해석(SPCG)에 대해 고유 알고리즘의 특성을 수정하지 않으면서 주 계산을 담당하는 노드의 개수와 상관없이 항상 동일한 해석결과가 계산되도록 개발하였다.

병렬 PCG 알고리즘의 개발에 있어서 중요한 것은 각 노드 간의 통신량을 최소화하도록 계산량을 분할하는 방법이다. 따라서 본 연구에서 개발된 병렬 PCG 알고리즘의 분할 개념은 구조물 전체 절점 수를 Slave 수로 균등하게 나눈 뒤, 분할된 절점들에 연결되어 있는 부제들에 대한 데이터를 각 Slave들에게 할당하는 방법을 사용하였다. 이로 인해 각 Slave 노드들이 제어매트릭스 C와 임시벡터  $\mathbf{q}$ 를 자기자신 노드에 할당된 정보들만 가지고 이들을 구성함으로써 계산상 불필요한 통신을 최소화 시켰다.

알고리즘의 초기부분은 Master 노드로부터 Slave 노드들의 연산에 필요한 데이터(부제 및 절점 정보)들을 전송하게되며 각 Slave 노드들은 할당된 데이터를 이용하여 PCG 알고리즘의 반복구간에 필요한 변수들을 각각 계산하게 된다. 반복구간에서는 1회반복당 Master 노드와 Slave 노드 모두 구조물의 경계부분만큼의 벡터  $\mathbf{p}$ 를 인접노드로부터 송·수신하게 되고 또한 수렴과 각종 벡터들의 업데이트를 위해 스칼라 값인 각 Slave의  $\alpha$ 와  $\beta$ 의 값을 Master 노드로 송신하고 최종계산된 스칼라값

## 기술기사

인  $\alpha$ 와  $\beta$ 를 전송받음으로써 PCG 알고리즘을 수행하게 된다.

알고리즘의 후반부에는 Master 노드가 알고리즘의 수렴성을 검토하게 되고 수렴하게 되면 각 Slave 노드로 수렴정보를 송신함으로써 모든 Slave 노드가 알고리즘을 종료하게 된다.

### 5. 적용 및 분석

개발된 병렬알고리즘과 대형 병렬시스템의 효율성 및 성능을 분석하기 위해 90층 입체 트러스 구조물과 147층 트러스 구조물에 적용하였다. 90층 트러스 구조물은 본 연구에서 개발된 병렬 PCG 알고리즘의 성능을 분석하기 위해 제작된 3차원 트러스 구조물로서 6,552개의 절점, 19,656개의 층자유도를 가지며 147층 규모의 3차원 트러스는 1801개의 절점, 5403개의 자유도를 가진다.

한 개의 노드에서 순차적으로 실행하도록 프로 그래밍된 SPCG 알고리즘과 개발된 병렬 PCG 알고리즘을 예제에 적용하여 노드가 2개, 4개를 사용하여 구조해석한 결과, 절점번호 및 용력이 모두 동일하게 나타났으며, 이는 개발된 병렬 구조해석 알고리즘의 정확성과 안정성을 말해준다.

본 기사에서는 리눅스 클러스터와 병렬 전산기를 이용한 병렬 구조해석 알고리즘의 성능을 성능향상도(Spccd-up)와 구조해석에 소요되는 시간으로 분석하였다.

표 1은 두 예제의 SPCG 알고리즘과 병렬 PCG 알고리즘에 대한 반복구간에서 소요되는 시간을 나타내며, 그림 2는 각 예제에 대한 리눅스클러스터 시스템과 병렬연산기에서의 알고리즘 성능향상도를 보여준다.

표 1 노드수에 따른 병렬PCG알고리즘 소요시간(초)

예제 노드수 (PC대수)	90층(19656 DOF)		147층(5403 DOF)	
	CrayT3E	Linux	CrayT3E	Linux
1	224.82	208.86	74.19	66.01
2	98.67	112.40	32.97	37.35
4	51.21	58.61	17.12	21.45

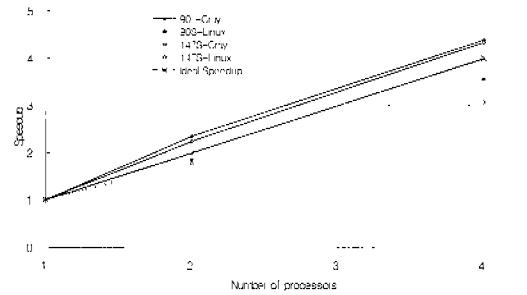


그림 2 성능향상도

### 6. 맺음말

리눅스 기반의 병렬컴퓨터 개발이 국내에서 활발히 진행되고 있는 실정이며 최근 개발된 포스테이터의 리눅스기반의 병렬컴퓨터 같은 경우, 그 성능이 세계 123위의 슈퍼컴퓨터로 인증 받고 있는 실정이다. 따라서 이러한 리눅스 클러스터 시스템은 구조해석 및 설계 분야에도 적용되어 곧 실용화가 될 전망이다.

본 기사에서 다룬 리눅스 클러스터 시스템과 대형 병렬 연산기인 Cray T3E를 이용한 대형 3차원 트러스 구조물을 해석할 수 있는 병렬 구조해석 알고리즘을 개발하여 적용한 결과, 다음과 같은 결론을 얻을 수 있었다.

- (1) 구조해석 및 설계 업무의 효율성에 대한 측면에서 볼 때, 단 하나의 CPU를 사용하는 개인용 컴퓨터나 워크스테이션을 이용할 경우보다 해석에 소요되는 시간을 많이 절감할 수 있기 때문에 훨씬 효율적이라는 것을 알 수 있다. 또한, 예제 적용결과 병렬알고리즘에 의한 해와 순차알고리즘에 의한 해가 동일하게 나타났으므로 병렬해석에 있어서의 안정성을 확인할 수 있었다.
- (2) 저가의 개인용 컴퓨터들로 구성된 리눅스 클러스터 시스템을 사용하여 대형 구조물의 구조해석에 적용시 이상적인 성능향상도에 근접한 결과를 보였으며 이는 대형 병렬 연산기의 성능과 비교하여 크게 뒤지지 않은 성능임을 알 수 있다. 따라서 비용적인 측면에서 보면 대형구조물의 해석 및 설계에 있어서 리눅스 클러스터

터 시스템을 구축하여 적용하면 고가의 대형 병렬 연산기를 이용하는 것보다 경제적이고 효율적임을 알 수 있다.

- (3) 대형 병렬 연산기에서 구조해석의 성능향상도가 모두 이상적인 성능향상도 이상이 되었는데 이는 소요메모리에 비해 실제 메모리가 부족하여 발생하는 현상으로 해석된다.

### 참 고 문 헌

1. 소프트뱅크 미디어, 마이크로 소프트웨어-클러스터링 파워, 2000년, 7월호, pp.208~237
2. 신영미디어, Linux@Work, 2001년12월호, pp.81~88
3. Al Geist, PVM : *Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing*, The MIT Press, 1994
4. 박효선, 박성무, 심창원, 김재홍, “부구조기법을 이용한 PC level 분산구조해석법”, 한국전산구조공학회 가을 학술발표회 논문집, 제11권, 제2집, 1998, pp.53~60
5. 박효선, 박성무, 권윤한, “PC level 병렬 구조해석법 개발을 위한 PCG 알고리즘”, 한국전산구조공학회 가을 학술발표회 논문집, 제11권, 제2집, 1998, pp.362~369
6. 심창원, 박효선, “분할방법에 따른 분산구조 해석법의 성능분석”, 한국전산구조공학회 발표논문집, 제12권, 제1집, 1999, pp.50~57
7. 심창원, 박효선, “PC level 분산구조 해석법의 효율성 예측”, 대한건축학회 학술발표논문집, 제19권, 제1호, 1999, pp.81~86
8. Hojjat Adeli and Hyo Seon Park, *Neurocomputing for Design Automation*, CRC Press, pp.129~131, 1998
9. 권윤한, 박효선, “대형병렬연산기를 이용한 병렬구조방정식 해법 개발”, 대한건축학회 발표논문집, 제20권, 제2호, 2000년, 10월
10. 권윤한, 박효선, “대형구조물의 분산구조해석을 위한 PCG 알고리즘”, 한국전산구조학회 논문집, 제12권, 제3호, 1999년, 9월 