

유전알고리즘 기반 콘크리트 구조물의 최적화 설계를 위한 멀티코어 퍼스널 컴퓨터 클러스터의 확장 가능성 연구

박 근 형¹ · 최 세 운¹ · 김 유 석¹ · 박 효 선^{1*}

¹연세대학교 건축공학과

A Study on the Scalability of Multi-core-PC Cluster for Seismic Design of Reinforced-Concrete Structures based on Genetic Algorithm

Keunhyoung Park¹, Se Woon Choi¹, Yousok Kim¹ and Hyo Seon Park^{1*}

¹Department of Architectural Engineering, Yonsei University, Seoul, 120-749, Korea

Abstract

In this paper, determination of the scalability of the cluster composed common personal computer was performed when optimization of reinforced concrete structure using genetic algorithm. The goal of this research is watching the potential of multi-core-PC cluster for optimization of seismic design of reinforced-concrete structures. By increasing the number of core-processor of cluster, decreasing of computation time per each generation of genetic algorithm was observed. After classifying the components in singular personal computer, the estimation of the expected bottle-neck phenomenon and comparison with wall-clock time and Amdahl's law equation was performed. So we could observed the scalability of the cluster appear complex tendency. For separating the bottle-neck phenomenon of physical and algorithm, the different size of population was selected for genetic algorithm cases. When using 64 core-processor, the efficiency of cluster is low as 31.2% compared with Amdahl's law efficiency.

Keywords : PC cluster, genetic algorithm, multi-core, reinforced concrete optimization

1. 서 론

1980년대부터 21세기 초반까지 최적화 기법에 사용하기 위해 많은 경험적 기법들(heuristic methods)이 연구되었다(Silver, 2004). 이들 기법이 건축분야에 적용되는 과정에서 나타나는 공통적인 문제점으로, 최적설계 탐색과정에 필요한 반복적인 구조해석을 위한 계산 시간의 증가를 들 수 있다. 특히 지진하중에 대한 저항성능을 평가하기 위해 비선형 구조해석을 수행하는 경우 선형 구조해석에 비해 계산시간이 비약적으로 증가한다. 이를 해결하기 위한 해결책으로는 해석 알고리즘을 간소화 하여 계산 자체를 줄이는 것과 고성능의 계산기기를 사용하여 계산시간을 줄이는 것으로 분

류할 수 있다.

현재 반도체 직접도의 단위가 더욱 정밀해짐에 따라 CPU의 성능이 발달하고, 여러 개의 CPU 혹은 GPU를 연결하는 등의 기술이 적용되어 슈퍼컴퓨터로 불리는 고성능 계산도구(HPC, High Performance Computer)의 성능이 크게 향상되었다. 2012년 11월 기준으로 세계에서 가장 빠른 HPC는 오크리지 국립연구소(Oak Ridge National Laboratory)에 설치되어 있는 'Titan-Cray XK7'로 총 560,640개의 코어의 Dodeca-core(2.2 GHz × 12-코어)로 구성되어 평균 17590.0 TFlops, 순간최대 27112.5 TFlops의 계산속도를 구현하였다(www.top500.org, 2013). TFlops는 HPC의 계산속도를 비교하기 위해 주로 쓰이는 단위로 1초 당 1조번의 수학 연

* Corresponding author:

Tel: +82-2-2123-7786; E-mail: hspark@yonsei.ac.kr

Received July 4 2013; Revised August 1 2013;

Accepted August 6 2013

©2013 by Computational Structural Engineering Institute of Korea

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

산처리를 수행할 수 있음을 뜻한다.

우리나라에도 기상청 등 국가가 운영하는 기관에서 슈퍼컴퓨터를 사용하고 있으나 2004년에 기상청에서 200억원에 구입했던 슈퍼컴퓨터 2호기(15.7 TFlops)를 2012년 11월에 무상으로 양도하려 했음에도 불구하고 인수할 대학연구소를 찾지 못하는 등 아직 슈퍼컴퓨터를 사용할 수 있는 기반이 마련되어 있지 않은 상황이다. 이와 같이 슈퍼컴퓨터는 구입가격도 높지만 운용을 위한 기본적인 전기 공급설비와 온-습도 유지설비 및 인력을 위한 높은 유지비용, 빠른 개발 속도로 인한 큰 감가상각 비율 등 경제적인 이유로 일반 중-소규모 연구소는 사용하기가 어렵다.

슈퍼컴퓨터를 구입하지 못하는 경우 대안방법으로 개인용 컴퓨터로 클러스터를 구성하여 슈퍼컴퓨터의 효과를 얻는 방법이 있다. PC 클러스터와 슈퍼컴퓨터는 CPU들 사이의 통신 성능이나 개별 CPU들의 성능의 차이는 있으나 여러 개의 CPU를 연결하여 하나의 고성능 계산기처럼 사용한다는 근본적인 원리는 같다. 슈퍼컴퓨터에 비해 구입 및 유지비용에 대한 부담이 적어 소규모 연구소에서 고성능 컴퓨터가 필요한 경우에 선택 가능한 대안이 된다(Song *et al.*, 2011).

이와 같은 클러스터 컴퓨팅을 위한 개인용 컴퓨터 또한 슈퍼컴퓨터의 발전과 함께 성능이 향상되어 왔으며, 현재 일반적으로 사용되는 PC는 대부분 단일 코어가 아닌 멀티코어로 이루어진 CPU를 내장하고 있다. 멀티코어 PC는 직렬기반의 계산 업무보다 병렬 기반의 계산 업무를 더 효율적으로 처리하는데 목적이 있다. 때문에 일반 PC에서도 성능을 효율적으로 사용하기 위해 MPCH2 등의 MPI 언어를 이용하여 CPU 내부에서 코어상의 병렬 처리를 하는 경우가 있다. 그러나 CPU 내부에서의 통신 속도는 PC간 통신 속도보다 월등히 빠르기 때문에 클러스터 PC를 구성하여 운용하는 것은 단일 PC의 병렬처리보다 복잡한 양상을 보인다.

구조물에 대한 비선형 지진응답해석에 있어서는 지진하중이 갖는 연속성으로 인한 직렬성 때문에 계산과정을 병렬화시키는 것이 어렵지만 최적 설계와 같은 여러 번의 독립적인 해석이 반복되는 경우 각 단위 계산을 동시에 수행하도록 병렬화함으로써 계산시간을 줄일 수 있다. 그러나 일반적인 구조물의 규모를 고려하면 슈퍼컴퓨터에서 계산을 할 만큼 많은 계산 분량을 가진 경우는 별로 없다. 이와 같이 알고리즘 내부에 독립적인 부분이 있어 병렬화가 가능하면서도 계산의 규모와 비용 때문에 슈퍼컴퓨터를 사용하기 어려운 경우 PC 클러스터를 구성하여 계산을 수행하는 것이 경제성을 얻을 수 있는 방법이 된다. 실제로 90년대에서 2000년대까지 구조해석을 위해 PC로 클러스터를 구성하여 사용하는 것에 대한 연구가 활발하게 진행되었다. 그러나 과거 PC의 CPU가

single-코어로 구성되어 있던 것에 비해 최근 PC는 멀티코어로 이루어진 CPU를 사용하면서 복잡한 데이터 통신 구조를 지니기 때문에 자원 증가량과 성능 향상의 관계를 선형, 혹은 낮은 차수의 비선형으로 추세할 수 없다.

본 연구에서는 구조물에 대한 비선형 지진응답해석의 효율성 향상을 위해 멀티코어를 내장한 PC로 구성된 클러스터의 특성을 관찰하고 관찰되는 scalability의 경향에 관한 기초적인 연구를 수행하였다.

2. 멀티코어 PC 클러스터

본 연구에 사용된 클러스터를 구성하는 PC는 Fig. 1과 같이 각 프로세서가 3.4GHz의 클럭 수를 가지는 쿼드 코어 프로세서 CPU와 4Gbytes의 램 2개, 일반 HDD로 구성되어 있다. 현재 일반 상용 CPU의 프로세서는 다중 파이프라인 방식을 채택하고 있어 프로세서들이 instruction을 병렬화하여 실행하고 있다(Kim, 2004; Darryl, 2011). 그러나 이 수준의 병렬화는 CPU에서 지원하는 기능으로 일반 연구자가 수정하는 것은 거의 불가능하기 때문에 본 연구에서 병렬화 대상에서 제외되었다.

CPU는 여러 개의 프로세서를 포함하고 있으며 각 프로세서는 개별적으로 프로그램을 수행할 수 있다. 각 프로세서들은 동시에 RAM이나 HDD에 접속하여 읽고 쓰는(Read & Write)것이 가능하지만 정보의 동기화 전략에 따라 memory의 같은 영역을 동시에 수정할 수는 없다(Reinders *et al.*, 2010). 이는 다수의 프로세서가 memory자원을 공유하기 때문에 발생하는 현상으로 경제성과 사용성 등을 이유로 PC내부의 자원이 한정되었기 때문이다. 같은 이유로 PC 당 외부로

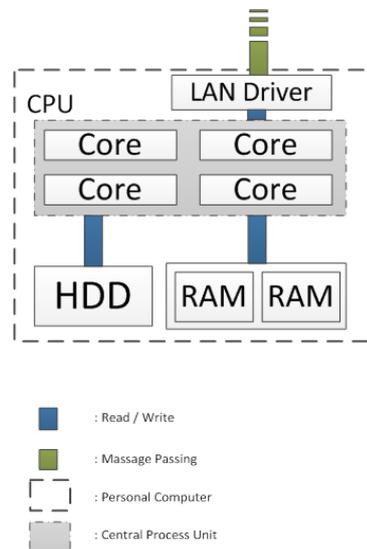


Fig. 1 Components of singular personal computer

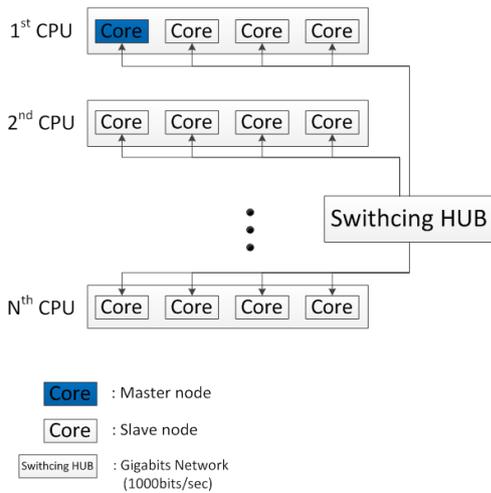


Fig. 2 Network structure of PC Cluster

연결되는 통신 port는 하나이며 본 클러스터에 사용된 컴퓨터도 같다. 때문에 프로세서들은 동시에 LAN에 접속하여 통신을 할 수 없으며 통신 구조는 Fig. 2와 같이 4개의 코어가 같은 통신선을 따라 switching hub에 연결되어 있다.

이와 같은 다층적인 성격을 보이는 데이터 통신방식 때문에 알고리즘이 적절한 병렬화 방식을 선택하지 못하는 경우 효율이 떨어질 수 있다. 때문에 기존 클러스터에서 사용했던 전통적인 병렬 방식인 master-slave가 존재하는 tree 모델을 멀티코어 클러스터에서 구현하는 경우 기존 클러스터에서 얻을 수 있는 효율과 다른 결과가 나온다.

3. 최적설계 및 병렬화 기법

3.1 유전알고리즘을 이용한 구조물 최적설계

구조물 최적설계에서 구조물의 규모가 크고 시스템이 복잡한 경우 단일 설계변수의 변화와 그에 따른 설계결과의 관계를 파악하기가 어렵다. 이와 같이 설계변수와 설계결과의 관계가 명확하지 않거나 전혀 알 수가 없어 최적해(Optimal solution)를 찾기가 어려운 경우에 사용할 수 있는 다양한 최적화 방법으로 결정론적 방법(Deterministic method), 확률론적 방법(Stochastic method), 발견적 방법(Heuristic method), 응답곡면법(Response-surface methodology) 등이 있으며 각 방법은 장단점을 지니고 있다(RAO *et al.*, 2009).

그 중 발견적 기법은 확률론적 방법과 같이 일정한 확률을 지니는 변수의 변화를 기반으로 탐색을 수행하지만 탐색 과정을 학습 및 평가하는 과정을 포함하고 있기 때문에 확률론적 방법의 개선된 방식으로도 볼 수 있으며 대표적인 예로 유전알고리즘(Genetic Algorithm)을 들 수 있다. 발견적

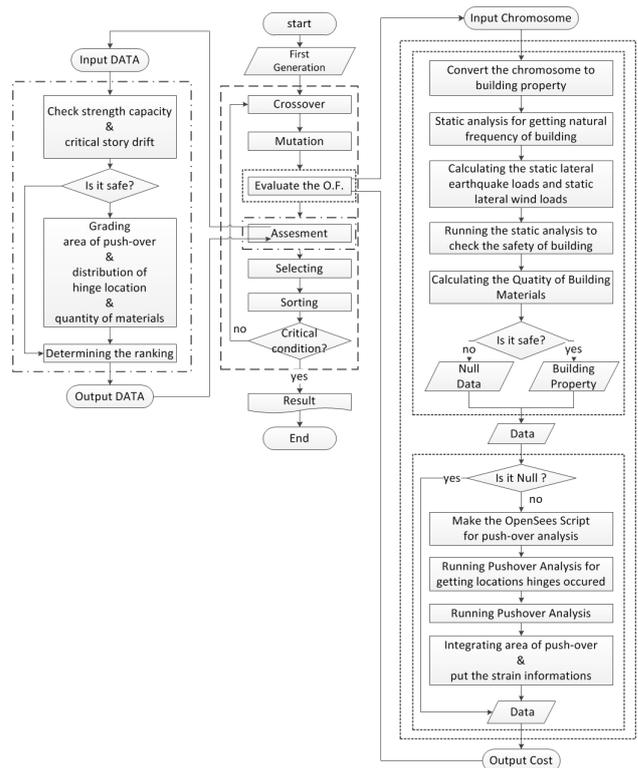


Fig. 3 Flow diagram of the NSGA-II for optimization of reinforced concrete with push-over analysis

기법의 진화 알고리즘에 속하는 기법 중 하나로 다양한 문제에 쉽게 적용할 수 있고 전역 최적해(Global optimal solution)를 탐색할 수 있는 장점이 있으나 근본적으로 반복되는 성질이 있어 개선되지 않은 단순기법의 경우 전산계산의 부담이 생기는 단점이 있다.

Simple GA는 탐색 능력이 효율적이지 않아서 그대로 사용하는 것에는 한계가 있기 때문에 다른 기법의 장점을 차용하는 등의 추가적인 개선이 필요하다. 때문에 본 연구에서는 다목적 최적설계에 적용할 수 있는 NSGA(Nondominant Sorting Genetic Algorithm)-II를 사용하였다.

본 논문에서 철근콘크리트 구조물의 최적설계를 위해 사용된 예제는 4층 4경간 철근콘크리트 2-D frame이며 push-over analysis를 통해 지진저항성능을 평가하는 방식으로 유전알고리즘의 개념도는 Fig. 3과 같다.

3.2 병렬 유전 알고리즘

본 연구에서 선택한 병렬방식은 control 코어 프로세서가 각 코어-프로세서로 population 단위의 계산 내용을 분리하여 task로 분배하는 방식으로 유전 알고리즘에서 가장 간단하게 구현할 수 있는 병렬화 방식이다. Slave-node 역할을 하는 코어-프로세서들은 배분받은 population의 계산을 각자 수행

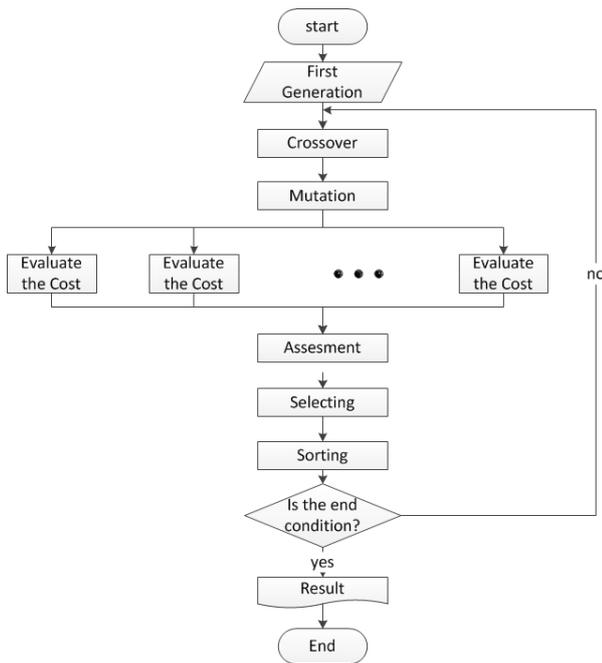


Fig. 4 Concept diagram of simple tree-model

한 후 master-node 역할의 코어-프로세서로 결과를 보낸다. 이 parallel model은 slave-node들은 오직 master-node 하고만 연결되어 통신하는 tree model(master-slave model)에 속한다.

2D-Frame의 비선형 정적 push-over 해석이 동반되는 최적화 설계는 전체 최적화 시간에서 각 population의 구조해석 시간이 차지하는 비중이 크기 때문에(Choi *et al.*, 2012) Fig. 4와 같이 population 단위의 병렬화를 함으로써 해석 시간을 효과적으로 줄일 수 있다.

4. 해석 시간을 기준으로 한 scalability 평가

Scalability는 시스템 자원(프로세서)을 증가시켰을 때 성능이 증가되는 정도를 말하며 본 연구에서는 속도 증가치는 병렬처리없이 프로그램이 실행되는 시간과 병렬로 실행되는 시간 사이의 비율로 정의하였다. 프로그램의 각 부분 혹은 전체의 계산에 소요되는 시간을 wall-clock time이라 하며 프로그램의 계산에 소요되는 계산시간의 효율계산을 위해 사용된다. 스크립트 상의 길이 비율과 실제 소요되는 시간의 비율은 다르기 때문에 실제 실행을 통해 확인해야 하는 부분이다. 얻어진 wall-clock time을 통해 클러스터의 코어-프로세서수를 확장할 때 증가한 프로세서의 개수와 감소한 전체 계산시간의 관계로 클러스터의 scalability를 확인한다.

최적 설계 알고리즘에 이용되는 유전알고리즘은 확률적으로 탐색하는 성질때문에 해가 수렴하기 위해 소요되는 세대

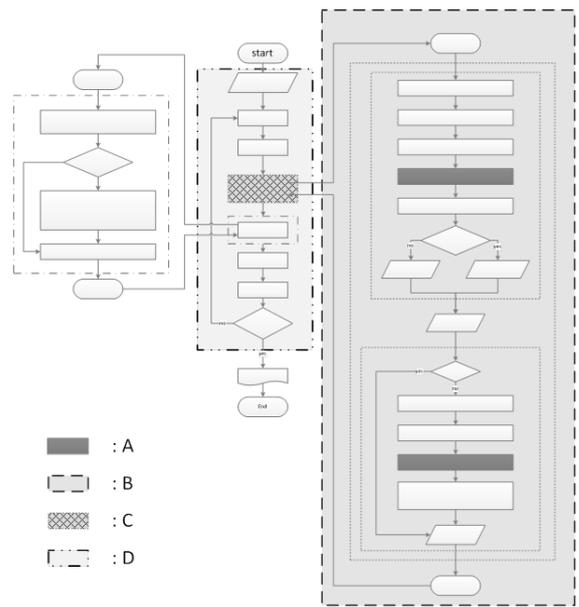


Fig. 5 Territory of paralleled part divided by wall-clock check point in analysis

수가 일정하지가 않다. 따라서 전체 최적화 완료하는 시간 외에 세대별 소요되는 시간을 별도로 기록하며 동일 최적화 과정을 20회 반복하여 scalability 성향을 관찰한다.

대상 알고리즘의 부분 별로 wall-clock-time을 측정하여 클러스터의 효율성을 점검하였다. Fig. 5와 같이 최적 설계 알고리즘에서 가장 작은 순서대로 구조해석을 수행하는 부분 A와 병렬화된 task의 전체 부분 B, 각 task가 코어-프로세서로 분배되기 전에서부터 결과를 받는 부분 C, 각 세대별 부분 D로 구분하여 시간을 기록하였다. 병렬화의 효율을 확인하는 방법에는 여러 가지가 있는데 가장 대표적으로 사용되는 방법은 Amdahl's law가 있으며 serial 해석 시 parallelizable part와 non-parallelizable part의 비율에 따라 구현할 수 있는 speed-up의 한계를 계산할 수 있으며 식은 다음 식 (1)과 같다.

$$S = \frac{1}{(1-P) + \frac{P}{N}} \quad (1)$$

S는 알고리즘이 최대도 도달할 수 있는 speed-up이며 P는 전체 알고리즘 계산 시간에서 병렬화가 가능한 부분에 소요되는 시간의 비율, N은 사용되는 프로세서의 개수이다. 이는 병목현상을 제외한 이상적인 식이기 때문에 통신에서 발생하는 지연현상을 고려하지는 못한다. 때문에 이 값은 상한선으로 고려해야 하며 통신에 의한 추가적인 지연시간 발생은 추가적으로 확인 및 고려해야 한다. 이를 확인하기 위해

C 부분과 D 부분을 구분하여 시간을 체크하였다.

단일 PC의 CPU는 4개의 코어-프로세서로 이루어져 있으며 하나의 cache memory, RAM, HDD를 단일 address로 완전 공유하기 때문에 SMP로 볼 수 있으며(Flynn *et al.*, 1972) distributed system인 클러스터와는 다른 양상을 보인다. 그러나 LAN은 공유하기 때문에 병목현상이 예상된다. 이를 확인하기 위해 단일 PC에서 1-2-4개의 순으로 코어-프로세서를 사용하는 scalability를 확인한다. 이때는 PC단위로 확장되는 경우의 scalability와는 다른 양상을 보일 것으로 예상하였다.

Distributed system인 PC 클러스터는 PC간 통신이 존재하며 각 PC마다 독립된 cache memory, RAM, HDD를 가지기 때문에 더욱 복잡한 양상의 scalability를 보일 것으로 예상하였다. 사용된 프로세서의 수는 1, 2, 4, 8, 16, 32, 64개의 경우로 나누어서 수행하였다.

GA의 최적 population 개수는 적용 예제 혹은 GA의 종류에 따라 달라질 수 있으며 심지어 크게 의미가 없을 수 있기 때문에 명확한 근거가 마련되지 않고 있다(Lim, *et al.*, 2007). 때문에 GA의 population 개수는 사용자의 경험을 바탕으로 조정되는 것이 대부분이다. 본 예제에서는 다른 최적화 예제(Choi, *et al.*, 2012)에서 GA를 이용해 수행되었던 최적화 설계 경우를 참고하여 population 개수가 40개인 경우에 대한 분석을 수행하였고, GA의 확률성을 고려하여 각 경우에 소요된 시간은 각 20회 수행된 값을 평균내어 기준하였으며 비탄성해석이 수행되는 10 generation에 대해 시간을 비교하였다.

Fig. 6에서 보이듯이 8개의 코어 프로세서를 사용한 경우까지는 시간의 감소효과가 확연하게 드러나지만 16개 이상을 사용한 경우부터 효율이 급격하게 줄어든다. master node 역할을 하는 컴퓨터에 통신을 시도하는 컴퓨터의 개수가 16 코어프로세서부터 4대가 되며 master node 컴퓨터의 코어 개수에 해당하는 숫자다. 이때 master node 컴퓨터의 CPU는 물리적인 병렬의 한계에 도달하며 이때부터 효율이 급감한다고 볼 수 있다. Fig. 7은 해석결과에 따른 speed-up 보여주며 실제 speed-up은 복잡한 비선형성을 보인다.

40population를 사용하는 경우는 2의 n 승으로 나뉘지 않는 경우도 있기 때문에 task 배분과정에서 프로세서 중 일부가 유휴상태인 경우가 발생하기도 한다. 이는 통신 속도에 따른 속도저하가 아닌 알고리즘 상 발생하는 병목현상이다. 때문에 GA의 population 개수를 결정짓는 중요 요인 중 하나인 hardware configuration에 기준하여 64개의 population을 운용해 보았다. 이와 같은 경우의 population 개수는 각 프로세서에 배분되는 task의 개수에 따라 병렬 효율성이 달라

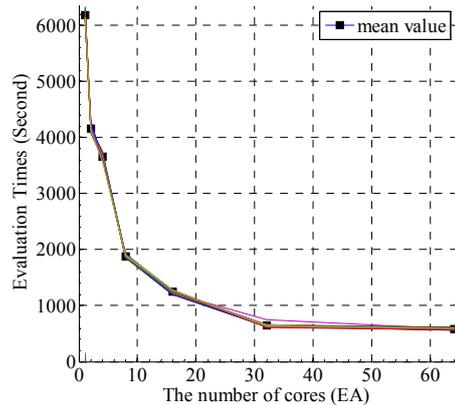


Fig. 6 Wall-clock times of 40 populations case

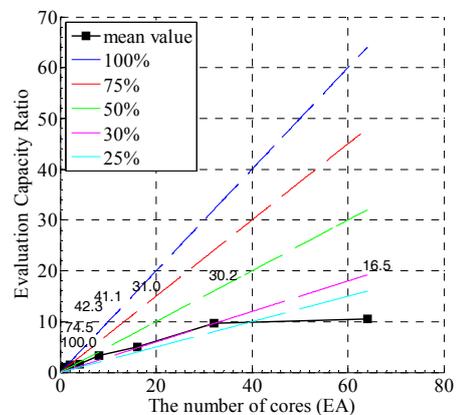


Fig. 7 Scalability of 40 populations case

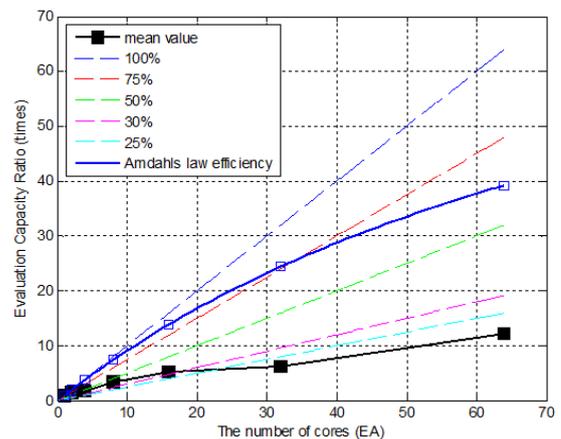


Fig. 8 Scalability of 64 populations case

짐을 확인하기 위하여 선택되었다.

64의 소수인 1, 2, 4, 8, 16, 32로 프로세서 수를 설정할 경우 64개의 task를 동등하게 배분할 수 있으리라 기대할 수 있다. 동시에 4 이상의 프로세서 사용 시 모두 4의 배수가 되어 quad-코어-프로세서 CPU의 특성을 가장 적절하게 보여줄 수 있는 숫자로 생각되었다.

Fig. 8은 Amdahl's law를 통해 예상한 이상적인 효율과 실제 speed-up의 경향을 보여준다. 40 population을 사용했을 때보다 조금 더 복잡한 경향을 보이며 대부분의 경우 효율이 더 떨어지나 64개의 코어를 사용한 경우는 40population의 경우보다 조금 더 나아진 경향을 보인다. 이는 알고리즘 상 발생했던 유희현상의 완화 때문으로 추정하였다.

5. 결 론

상용 PC 컴퓨터와 네트워크 장비로 클러스터를 구성하여 단순한 tree 병렬구조를 구현하였을 때 scalability의 관찰을 통해 효율의 한계 및 가능성을 수치적으로 확인할 수 있었으며, 상대적으로 저렴한 가격과 복잡하지 않은 장비들로 HPC를 구현할 수 있는 가능성을 확인하였다.

Hardware 측면에서 결과를 분석하는 경우 Tree-model 형태의 master-slave 병렬모델을 사용하는 경우 master node역할을 수행하는 코어 프로세서에 4개 미만인 동시 통신 접속 부하가 발생하는 경우 70%의 효율을 보이는데 반하여 4개 이상의 경우 효율이 40%대까지 급격하게 떨어짐을 확인하였다. 이는 master node 코어 프로세서가 위치한 PC의 quad-코어가 동시에 처리할 수 있는 한계와 관계가 있다고 볼 수 있다.

알고리즘 측면에서 결과를 볼 때 유전 알고리즘의 population의 개수가 클러스터를 구성하고 있는 코어 프로세서의 개수의 배수일 때 3~5% 효율이 향상됨을 확인하였다. 이는 코어 프로세서들의 유희상태가 감소함으로써 나타나는 현상으로 Amdahl's law에서 가정된 이상적인 상황으로 볼 수 있다. 실제로 Amdahl's law의 예측 값과 차이가 나는 이유는 유전알고리즘의 특성상 각 task의 해석시간이 일정하지 않아 완전하게 분배되지 못하여 유희상태가 발생하는 경우와 master node인 코어프로세서의 동시 통신 접속 부하에 의한 병목현상으로 구분할 수 있으며 본 연구에 사용된 예제는 전체 계산시간에 비해 통신시간이 매우 짧기 때문에 알고리즘에 의한 유희상태가 주요 원인으로 추정된다.

클러스터의 모든 자원을 사용하는 경우 일반 PC에서 해석을 수행하는 경우보다 최대 12배까지 빨라지는 것을 확인하였으나 기대했던 효율에 크게 못 미치는 19%의 효율을 보여 추가적인 개선의 필요성을 확인하였다. 이는 Amdahl's law로 예측한 성능의 31.2%에 해당한다.

본 연구를 발전시키는 방법으로 Hardware의 계층적인 통신 방식 특성에 맞춰 MPI와 OpenMP가 혼합된 스케줄러를 사용하는 병렬모델을 선택할 수 있다. 이러한 병렬모델을 고

려할 때 island genetic algorithm 등의 특화된 유전알고리즘모델이 적용 가능하다. 일반적으로 구조해석의 특성상 매트릭스 계산수행이 많기 때문에 상용 PC에 내장되어 있는 GPU를 이용하는 등의 잉여 hardware를 활용하는 방법도 필요하다.

감사의 글

이 논문은 2011년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구입니다(No. 2011-0018360).

참 고 문 헌

- Choi, S.W., Park, H.S.** (2012) Multi-objective Seismic Design Method for Ensuring Beam-hinging Mechanism in Steel Frames, *Journal of Constructional Steel Research*, 74, pp.17~25.
- Darryl, G.** (2011) Multicore Application Programming: For Windows, Linux, and Oracle SOLARIS, *Pearson Education*, pp.512.
- Flynn, Michael J.** (1972) Some computer organizations and their effectiveness, *Computers, IEEE Transactions on*, 100(9), pp.948~960.
- Kim, J.H.** (2004) Parallel Computer Architecture, Saengneung Publisher.
- Lim, D., Ong, Y.-S., Jin, Y., Sendhoff, B., Lee, B.-S.** (2007) Efficient Hierarchical Parallel Genetic Algorithms Using Grid Computing, *Future Generation Computers Systems*, 23, pp.658~670.
- RAO, Singiresu S.** (2009) Engineering optimization: theory and practice., *Wiley*, pp.813
- Reinders, James.** (2010) Intel Threading Building Blocks: Outfitting C++ for Multi-core Processor Parallelism, *O'Reilly Media*, pp.392.
- Silver E.A.** (2004) An overview of heuristic solution methods, *Journal of the Operational Research Society*, 55, pp.936~956.
- Song, Y., Shin, S.S., Jung, D., Park, T.** (2011) Numerical Analysis of Nuclear-Power Plant Subjected to an Aircraft Impact using Parallel Processor, *Journal of the Computational Structure Engineering*, 24(6), pp.715~722.
- www.top500.org** (2013)

요 지

본 논문에서는 유전알고리즘을 사용하여 철근콘크리트 구조물의 최적 지진설계를 효율적으로 수행하기 위해 클러스터를 사용하는 경우 확장성을 확인하였다. 클러스터를 구성하는 코어프로세서의 개수를 증가시키면서 유전알고리즘의 각 세대에 소요되는 시간의 감소를 관찰하였다. 단일 퍼스널 컴퓨터의 구성을 분류한 후, wall-clock time과 암달의 법칙으로 예상된 값을 비교하여 예상되었던 병목현상을 확인하였다. 이에 클러스터의 확장성에서 복합적인 요인에 의한 경향을 확인할 수 있었다. 병목현상의 물리적인 요인과 알고리즘 측면에서의 요인을 구분하기 위해 유전알고리즘의 개체수를 나누어 실험을 수행하여 결과를 확인하였다.

핵심용어 : PC 클러스터, 유전 알고리즘, 멀티 코어, 철근콘크리트 구조물 최적화.