대규모 반복 해석이 요구되는 작업의 병렬수행을 위한 가중치 기반 부하분산 알고리즘

A Weighted Load Balancing Algorithm for Parallelism of Massive Iterative Jobs

조재석[1], 정상진[2], 송용호[3], *#최동훈[4]

J. S. Cho[1], S. J. Jung[2], Y. H. Song[3], *#D. H. Choi (dhchoi@hanyang.ac.kr)[4]

[1,2]한양대학교 대학원 기계공학과, [3]한양대학교 정보통신공학부, [4]한양대학교 기계공학부

Key words : Heterogeneous Multiple Processor System, Load Balancing, Parallelism, Multidisciplinary Analysis

1. 서론

최근의 제품설계 경향은 설계하고자 하는 시스템이 점차 복잡해지고 있으며 이에 따라 해석할 때 드는 시간적, 수치적 비용이 크게 증가하고 있다. 수치적 비용이 많이 드는 해석 시스템에 대해 설계시간을 단축하면서 제품의 성능을 극대화하는 설계안을 얻기 위해 반응표면법이나 다구찌 기법 등과 같은 다양한 실제적인 설계기법들이 개발되고 있으나, 이러한 설계기법들을 적용한다할지라도 만족할만한 제품성능을 얻기 위해서는 여전히 많은 횟수의 설계안의 해석이 필요하다. 또한 정확한 해석을 위해 1회 해석에 오랜 시간이 소요되는 정밀한 CAE 해석모델을 사용할 경우 전체 설계 과정을 완료하는데 상당한 시간이 소요된다. 만약, 그림 1과 같이 설계문제가 연성관계를 가지는 여러 분야의 공학적 원리들을 동시에 고려해야 하는 다분야통합해석(Multidisciplinary Analysis; MDA)을 필요로 할 경우 소요시간은 더욱 커지게 된다.

이러한 문제를 해결하기 위해 복수의 프로세서로 구성된 병렬처리시스템 (Multiple Processor System; MPS)을 도입하여 설계과정에서 필요한 복수의 설계안의 해석을 병렬처리 하고자 하는 연구가 진행되고 있다 [1, 2, 3, 4]. 그러나 MDA에 기반한 설계문제의 경우 한 개의 설계안에 대한 해석을 위해 여러 분야의 CAE 모델의 해석이 순차적으로 반복수행 되어야 하는 MDA의 특성상, 선입선출 (First Come First Serve; FCFS) 알고리즘과 같은 기존의 일반적인 병렬처리기법으로 복수의 설계안에 대한 해석의 병렬처리를 효율적으로 수행하기 위해서는 MPS을 구성하는 모든 서버에 MDA에 필요한 각 분야의 CAE S/W가 모두 설치되어 있어야 한다.

그러나 실제 산업현장에서는 설계를 위해 사용하는 CAE S/W들이 상용 S/W임을 감안할 때 이는 매우 큰 비용을 필요로 하게 된다. 따라서 대규모 반복해석이 요구되는 설계문제를 MPS를 이용하여 소요시간을 단축한다는 것은 중소기업이나 소규모 연구소등에게 S/W 구매비용문제로 인해 실제로 구현이 매우 어려울 수밖에 없다.

따라서 각 서버의 성능이 각기 다르고, 해석을 수행하기 위해 필요한 CAE S/W들이 각 서버에 일부만 설치되어 있더라도 대규모 반복해석이 요구되는 설계문제를 효과적으로 병렬처리하기 위해 해석모델들이 각 서버에 부하가 균등하게 가해지도록 조절하는 새로운 부하분산 (Load Balancing) 알고리즘이 필요하다.

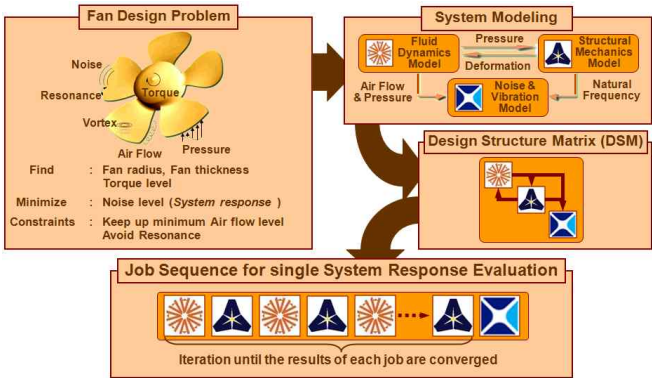


Fig. 1 Characteristic of Jobs Required for Problem

2. 가중치 기반 부하분산 알고리즘

대규모 반복 해석이 요구되는 설계문제를 다루기 위해 본 연구에서는 가중치 기반 부하분산 알고리즘을 개발하였다. 이 기법을 설명하기 위해, 설계기법에 정의된 N개의 실험점(Sampling Point)에 의한 시스템의 반응치를 구하기 위해 N번의 MDA를 수행해야 하는 대규모 통합해석문제를 고려하고자 한다. 통합해석시스템은 i개의 CAE 해석모델들로 구성되며, 각 모델의 해석순서는 그림 1과 같이 설계구조행렬 (Design Structure Matrix; DSM)을 이용하여 정의할 수 있다.

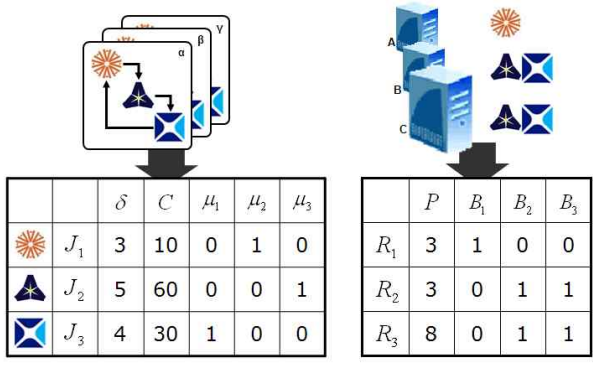
병렬처리를 수행할 MPS는 각 서버들의 성능이 각기 다르며, 설치된 CAE S/W들의 종류도 각기 다른 M개의 서버로 구성된 Heterogeneous MPS인 경우를 고려하였다. n번째 MDA의 DSM에 속한 i번째 모델의 k번째 해석은 작업 ${}_kJ_i^n$ 으로 모델링되며, ${}_kJ_i^n$ 의 수행완료에 필요한 연산의 예상치인 요구계산량은 ${}_kJ_i^n : C$ (unit : GHz·sec)로, DSM의 j번째 모델이 J_i^n 의 해석결과를 입력변수로 사용하는지 여부는 $J_i^n : \mu_j$ 로, MDA가 수렴하기까지 수행해야 하는 J_i^n 의 반복회수의 예상치를 $J_i^n : \delta$ 로 정의한다. 또한 MPS의 m번째 서버는 R_m으로 모델링되며, R_m의 성능은 $R_m : P$ (unit : GHz)로, R_m에 DSM의 i번째 모델의 해석을 수행할 수 있는 CAE S/W의 설치여부는 $R_m : B_i$로 정의한다. MDA와 MPS를 모델링한 예는 그림 2와 같다.

해석소요시간 ${}_kT_i^n$ 은 ${}_kJ_i^n$ 이 R_m 에서 수행되었을 때 완료에 소요되는 시간이며 식 (1)과 같이 정의할 수 있다.

$$ {}_kT_i^n = \frac{{}_kJ_i^n : C}{R_m : P} = c/p \quad (sec) \tag{1} $$

where ${}_kJ_i^n : C = \mathrm{c}\ GHz \cdot sec,\ R_m : P = \mathrm{p}\ GHz$

제안된 알고리즘에서는 MDA를 구성하는 각 CAE 해석모델에 가중치를 부여하며, 가중치를 이용하여 CAE 해석모델들 간의 서버할당의 우선순위를 정한다. MDA의 i번째 CAE 해석모델 J_i의 가중치는 세 종류의 항목으로 구성되며 첫 번째는 MDA가 수렴하기까지 J_i의 해석을 수행하기 위한 요구계산량의 총합, 두 번째는 MPS의 서버들 중 J_i의 해석을 수행할 수 있는 CAE S/W가 설치되어 있는 서버들의 가용계산성능의 합, 세 번째는 DSM에 정의된 CAE 모델의 해석순서에 따라 J_i의 해석이 완료되어야 해석을 수행할 수 있는 다른 모델들의 가중치의 합이다.



		δ	C	μ_1	μ_2	μ_3
	J_1	3	10	0	1	0
	J_2	5	60	0	0	1
	J_3	4	30	1	0	0

	P	B_1	B_2	B_3
R_1	3	1	0	0
R_2	3	0	1	1
R_3	8	0	1	1

Fig. 2 Example of Modeling CAE models and MPS

앞에서 기술한 세 가지 항목을 적용하여 각 CAE S/W모델의 가중치 W를 구하기 위해서는 각 작업들을 어느 서버에 할당할지 결정해야 하며 이를 위해 식 (2)의 분모부분에 있는 $R_m : B_i$ 들 중 어떤 값을 0으로 변경할 것인지 결정함으로서 각 CAE S/W모델의 요구계산량의 총합을 각 작업을 수행할 수 있는 서버들의 성능으로 나눈 값인 평균해석소요시간 T^{avg} 를 구해야 한다.

$$T_i^{avg} = \frac{J_i^{avg} : C}{\sum_{m=1}^{M} (R_m : B_i \times R_m : P)} \tag{2}$$

제안된 알고리즘은 식 (3)을 통해 가중치를 각 CAE 모델에 부여하며, 각 모델의 가중치를 구하기 위해서는 각 서버들이 어떤 작업들을 담당할지 결정해야 한다. 각 서버들이 담당할 작업들은 $Max(\sum T_i^n)$ 을 최소화할 수 있는 조합을 찾음으로서 결정된다. 가중치 W_i 는 J_i 의 해석결과를 입력변수로 사용하는 다른 모델들 중 J_i 보다 DSM에서 상단에 위치하는 모델이 존재하는지 여부에 따라 다른 식이 적용된다.

$$[\text{if } all \text{ of } J_i : \mu_D = 0 \ (1 \le D < i)]$$
$$W_i = T_i^{avg} + \sum_{d=i+1}^{I} (J_i : \mu_d \times W_d)$$

$$[\text{if } any \text{ of } J_i : \mu_D = 1 \ (1 \le D < i)] \tag{3}$$
$$W_i = T_i^{avg} + \sum_{d=i+1}^{I} (J_i : \mu_d \times W_d) + (J_i^{avg} : \delta - Itr) \times \sum_{d=D}^{i} T_d^{avg}$$

where Itr is current iteration No.

3. 전산실험 결과 및 고찰

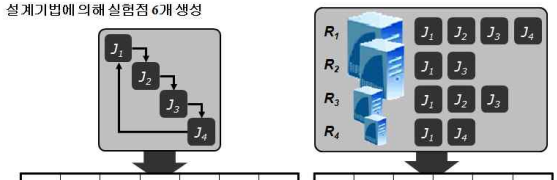
알고리즘의 성능을 평가하기 위해 C++ 로 구현한 시뮬레이터를 작성하였다. 작업과 자원, 큐를 구현하고, 각 CAE 모델의 해석이 서버에서 수행되는 과정을 모사하였다. 유휴상태에서 대기하는 서버가 발생하였을 때 서버에 할당할 작업을 결정하기 위해 제안된 알고리즘을 적용하였고 비교대상으로 MPS의 기본적인 구조인 싱글큐 선입선출 알고리즘을 선정하였다.

싱글큐 선입선출 알고리즘은 큐의 최상단에 대기 중인 작업을 수행할 수 있는 서버가 발생하면 작업을 서버에 할당하게 되며 작업을 수행할 수 있는 서버가 동시에 2개 이상 발생할 경우 서버의 성능에 대한 고려 없이 서버의 일련번호가 앞쪽에 있는 서버부터 할당하도록 구현된다.

제안된 알고리즘의 유효성을 검증하기 위해 그림 3과 같은 MDA 예제를 이용하였다. MDA의 DSM은 CAE 모델의 해석순서가 하나의 루프를 형성하는 간단한 형태이다. 각 CAE 해석모델에 의해 생성되는 작업들 중 가장 요구계산량이 큰 J_4 를 처리할 수 있는 서버는 R_1 과 R_4 이다. R_1 은 가용성능이 가장 큰 동시에 J_1 부터 J_4 까지 모든 작업을 수행할 수 있는 서버이며 따라서 R_1 이 어떤 작업들을 수행하느냐에 따라 전체 MPS의 효율성에 큰 영향을 미친다.

선입선출 알고리즘의 경우 병렬처리를 시작하면 싱글큐의 자료구조로 인해 작업 J_1^1 부터 작업 J_1^6 까지 6개의 작업이 모두 수행 완료되어야 큐의 뒤쪽에 대기하고 있는 작업 J_2^1 를 수행할 수 있게 되며 이후 과정에서도 동일한 현상이 나타나게 되므로 서버들이 유휴상태에서 대기하게 되는 시간으로 인해 병렬처리의 효율성이 떨어지는 결과를 나타낸다.

본 연구를 통해 개발된 가중치 기반 부하분산 알고리즘의 경우, 멀티큐에 의해 J_1^1 의 수행이 완료되는 즉시 J_2^1 이, J_2^1 의 수행이 완료되는 즉시 J_3^1 이 해당 작업을 관리하는 큐의 최상단에서 대기하며 서버를 할당받을 수 있게 되므로, 싱글큐 선입선출



	δ	C	μ_1	μ_2	μ_3	μ_4
J_1	4	10	0	1	0	0
J_2	4	20	0	0	1	0
J_3	4	40	0	0	0	1
J_4	4	50	1	0	0	0

	P	B_1	B_2	B_3	B_4
R_1	5	1	1	1	1
R_2	4	1	0	1	0
R_3	2	1	1	1	0
R_4	1	1	0	0	1

Fig. 3 Modeling of the example problem

알고리즘에 비해 유휴상태에서 서버가 대기하는 시간이 줄어들어 병렬처리의 효율성이 매우 높아졌다. 실험점 6개에 대한 MDA의 수행을 모두 완료하는데 싱글큐 선입선출알고리즘은 340 sec, 제안된 알고리즘은 280 sec의 시간이 소요되었으며 제안된 알고리즘의 성능이 약 18% 우수함을 확인할 수 있었다.

4. 결론

다분야통합해석과 같은 대규모 공학해석을 수행해야 하는 설계문제의 최적설계안을 얻기 위해 설계기법을 적용하는 과정에서 다분야통합해석이 반복적으로 요구되며, 이에 소요되는 시간을 단축하기 위해서는 병렬처리시스템을 도입할 필요가 있다. 본 연구에서는 서버들의 성능과 각 서버에 설치된 CAE S/W들의 종류가 각기 다른 Heterogeneous MPS를 이용한 가중치 기반 멀티큐 부하분산 알고리즘을 개발하였다. 멀티큐를 도입함으로써 큐에 대기 중인 작업들을 작업 종류에 따라 독립적으로 관리하도록 하였고, 각 CAE S/W의 해석 수행 시 서버를 할당받을 수 있는 우선순위를 나타내는 가중치를 도입하여 서버가 유휴상태로 대기하는 시간을 최소화하도록 하였다.

제안된 알고리즘의 성능검증을 위해 일반적인 병렬처리기법인 선입선출 알고리즘의 성능과 비교하는 전산실험을 수행하였다. 그 결과 대규모 반복해석이 요구되는 설계문제에 대한 병렬처리를 수행하였을 때, 제안된 알고리즘이 싱글큐 선입선출 알고리즘보다 효율적임을 확인하였다.

후기

본 논문은 2009년도 2단계 두뇌한국 21사업, 지식경제부 “c-MES 설계지원 플랫폼 기술 개발” 과제 (10033162-2009-11)의 지원으로 연구되었으며 지원해주신 각 당국에 감사의 뜻을 표합니다.

참고문헌

1. Ni, L.M. and Hwang, K., "Optimal Load Balancing in a Multiple Processor System with Many Job Classes," Software Engineering, IEEE Transactions, SE-11(5), 491-496, 1985.
2. Becker, W. and Waldmann, G., "Exploiting inter task dependencies for dynamic load balancing in High Performance Distributed Computing, the 3rd IEEE International Symposium, 1994.
3. Koch, P.N., Wujek, B., and Golovidov, O., "A Multi-Stage, Parallel Implementation of Probabilistic Design Optimization in an MDO Framework," the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2000.
4. Kato, T., "Realizing Grid Computing as Engineering System for Collaborative Parameter Study," GridWorld/GGF15, Boston, MA, USA, 2005