병렬처리에서 평균장 어닐링과 유전자 알고리즘을 이용한 부하균형

*홍철의° **박경모
*상명대학교 소프트웨어학부, **가톨릭대학교 컴퓨터정보공학부
hongch@sangmyung.ac.kr°, kpark@catholic.ac.kr

Load Balancing Using Mean-Field Annealing and Genetic Algorithms in Parallel Processing

* Chul-Eui Hong°
Software School, Sangmyung University
** Kyeongmo Park
School of Computer Science and Information Engineering
The Catholic University of Korea

요 약

본 논문에서는 병렬처리에서 중요한 부하균형 문제에 대한 새로운 솔루션을 소개한다. 제안하는 매핑 알고리즘은 평균장 어닐링과 유전자 알고리즘을 합성한 휴리스틱 부하균형 기법이다. 합성된 알고리즘을 세 개의 다른 알고리즘들과의 성능향상비를 측정하는 성능평가 시뮬레이션을 개발하였고 솔루션 품질과 수행시간 면에서 우리의 방법은 기존의 것들 보다 개선된 실험결과를 얻었다.

1. 서 론

병렬처리에서 부하균형(load balancing)을 통해 멀티프로세서 노드들의 처리 성능을 최대화시키는 것은 중요한 연구주제이다. 부하균형의 목적은 병렬시스템의 모든 프로세서 노드들의 작업부하를 동등하게 배분하고 노드간의 통신 트래픽을 적게 하는 것이다. 본 연구에서는 분산메모리 멀티프로세서 노드를 갖는 병렬처리 시스템에서의 부하균형 문제에 새로운 솔루션을 소개한다. 제안 알고리즘은 평균장 어닐링(mean field annealing)[1]과 유전자 알고리즘(genetic algorithms)[2]에 기반을 둔다.

본 연구의 구성은 다음과 같다. 2장에서는 본 연구에서 다룰 부하균형 문제를 서술한다. 3장에서는 주어진 문제를 해결하기 위한 알고리즘들을 제시하고 구현 관련 내용에 대해 기술한다. 4장에서는 시뮬레이션 연구를 통해 제시한 알고리즘들의 성능을 측정 평가한 실험결과를 설명한다. 마지막으로 5장에서 결론을 맺었다.

2. 멀티프로세서 매핑 문제

본 연구에서 다루는 멀티프로세서 매핑(mapping) 문제에 대하여 공식적으로 기술한다. 태스크 그래프 $G_T(V, E)$는 $|V| = N$, $(1, 2, \cdots, i, j, \cdots N)$로 표시되고 공집합이 아닌 정점(vertex)의 집합이다. G_T의 정점들은 병렬 프로그램의 태스크를 나타낸다. 정점 가중치 w_i 는 태스크 i, $(1 \le i \le N)$ 에 따른 계산비용을 나타낸다. E는 V에 속하는 두 점 사이를 잇는 모서리(edge)의 집합이다. 모서리 가중치 e_{ij} 는 모서리 $(i, j) \in E$ 로 연결된 태스크 i 와 j 간의 상호작용의 양을 표시한다. 프로세서 그래프 $G_P(P, D)$ 는 $|P| = K$ 노드와 $|D| = \binom{K}{2}$ 모서리를 갖는 모든 정점 사이에 모서리가 존재하는 완전그래프(complete graph)이다. G_P의 노드들은 $(1, 2, \cdots, p, q, \cdots K)$로 목표로 하는 멀티프로세서를 표시한다. 모서리 가중치 d_{pq} , $(1 \le p, q \le K)$, $p \neq q$ 프로세서 p와 q 사이의 통신비용을 나타낸다.

주어진 N-to-1 매핑 문제는 $M: V \rightarrow P$를 구하는 것이다. 즉 태스크 그래프 G_T의 각 정점을 프로세서 그래프 G_P의 유일한 노드에 할당하는 것이다. 각 프로세서에 균형을 맞춘 부하(*Load*)를 유지하면서 노드간 전체 통신비용(*Comm*)을 최소화하는 것이 문제다. 수식으로 표현하면 다음과 같다.

$$Comm = \sum_{(i,j) \in E, M(i) \neq M(j)} e_{ij}, d_{M(i)M(j)} \qquad (1)$$

$$Load_p = \sum_{i \in V, M(i) = p} w_i, \ 1 \le p \le K \qquad (2)$$

$M(i)$ 는 태스크 i가 매핑되는 프로세서 p를 표시한다. (1) 식에서 G_T의 각 모서리(i, j) 는 정점 I와 j가 G_P의 다른 2개 노드에 매핑이 되면, 즉 $M(i) \neq M(j)$이라면 통신비용에 부담을 준다. 그 분량은 2개 태스크 간의 상호작용 양과 프로세서 p와 q 간의 단위 통신비용을 곱한 것이다. 여기서 $p = M(i)$, $q = M(j)$. 프로세서 부하는 해당 프로세서 노드에 할당된 여러 개의 태스크 가중치를 합계한 것이다.

3. 부하균형 기법

3.1 평균장 어닐링 알고리즘

물리학에서는 최적화 분야에서와 같이 대단히 큰 자유도를 갖는 시스템을 다루는 경우가 있다. 이 때 큰 자유도를 가지는 문제는 열 평형 상태에 있는 입자(particle)나 스핀(spin) 시스템들의 상태를 추정하는 평균장 근사법(mean-field approximation)으로써 간략하게 해결된다. 본래 평균장 어닐링(mean field annealing, MFA)은 물리

학 평균장이론 근사법에 바탕을 둔 시뮬레이티드 어닐링(simulated annealing, SA)으로부터 나온 것이다. SA 기법을 이용하여 실제 문제에서 최적화를 효율적으로 수행할 수 있음에도 불구하고 문제에 대한 솔루션을 구하는데 많은 계산 시간이 소요되는 단점을 가지고 있다. MFA는 SA 기법에서 임의로(randomly) 상태를 변화시키는 것과 달리, 평균장 근사법을 사용하여 산출되는 평균값으로 대치시키는 방법으로 열(온도) 평형 상태에 빠르게 도달할 수 있다는 장점이 있다.

MFA는 홉필드 신경망 (Hopfield neural network, HNN)과 SA를 조합한 방법으로 조합 최적화 문제에 적용된다. HNN은 작은 크기의 최적화 문제에는 효율적으로 적용 가능하나, 문제 크기가 커질수록 적용이 곤란하여 실제적인 문제에의 적용이 어렵다. MFA는 HNN에서와 같은 방법으로 상태를 표현한 다음 시스템을 열평형 상태에 도달시키기 위하여 SA에서와 같이 반복기법을 적용한다.

MFA구현 관련해 냉각 스케줄(cooling scheduling)은 문제의 특성에 따라 민감하게 작용하므로 부하균형 문제 및 비용함수에 적절한 스케줄의 선택이 필요하다.

- 초기 온도 : T_0

초기 온도는 태스크 수(N) 만큼 평균장 어닐링을 수행하였을 때 비용의 변화가 허용치 ε(= 0.5) 이하일 확률이 95%이상일 때의 온도로 설정한다.

- 최종 온도 : T_f

최종온도는 20회의 온도 변화동안 연속해서 비용의 변화가 ε/1000 내에 존재할 때의 온도로 설정한다.

- 임의의 온도 k에서의 Markov 체인 길이 : L_k

각 온도에서의 평형상태에 도달하기 위한 상태변환의 회수로 연속해서 태스크 수만큼 비용의 변화가 허용치 ε내에 존재할 때의 상태 변화 수로 설정한다.

- 온도 감소율 : $T_{k+1} = \alpha T_k$

온도 감소율 α는 실험적으로 구해진 최적값인 0.9로 정하였다.

3.2 유전자 알고리즘

유전자 알고리즘(genetic algorithm, GA)은 자연계의 적자생존의 원리에 기초하여 생물학적 진화 시스템을 모의 실험하여 태스크할당, 함수 최적화 등 중요한 많은 문제들에 대한 근사 최적해를 구하는데 성공적으로 널리 사용해왔다. 부하균형을 고려한 멀티프로세서 매핑 관련 GA 구현에 사용된 파라미터 세팅은 다음과 같다.

- 개체 표현:

태스크의 순서대로 할당되는 노드번호의 순서를 문자열로 표현한다. 즉, 태스크 0부터 4까지 5개의 태스크를 노드에 할당되어 있는 상태를 표현한 문자열 "1,5,4,1,5"은 태스크 0과 3는 노드 1, 태스크 1과 4는 노드 5, 노드 2는 노드 4에 할당되어 있는 상태를 표현한다.

- MFA에서 GA로의 개체 생성:

MFA에서 사용되는 N×K 스핀 매트릭스와 같은 확률로 랜덤하게 개체를 생성한다. 예로서, 임의의 태스크가 노드 0부터 4까지 할당되는 스핀 값이 0.2, 0.4, 0.1, 0.1, 0.2라면 그 태스크가 노드 0에 할당될 확률은 0.2, 노드 1에는 0.4, 노드 2와 3에는 0.1, 노드 4에는 0.2의 할당 확률을 가지고 개체를 생성하게 된다.

- 개체집단(population)의 크기:

개체 수를 말하며 본 실험에서는 100으로 설정하였다.

- 비용함수:

MFA와 같은 1차식 비용함수를 사용하였다.

- 선택(selection) 연산자:

집단의 전체 비용에 대한 개체의 비용 비율만큼 랜덤 함수를 사용하여 개체를 선택한다.

- 재생산(reproduction):

선택 연산자에 의하여 선택된 개체를 가지고 집단의 크기만큼의 새로운 집단을 생성한다.

- 교차(crossover) 연산자:

개체의 순서대로 2개의 개체를 선택하여 유전자 즉, 태스크의 노드 할당 상태를 나타내는 문자열의 부분을 서로 교환한다. 교환될 태스크의 선택은 전체 태스크수의 1/4이하로 제한 하였으며, 선택은 랜덤하게 이루어진다. 교차 연산자를 수행할 확률은 0.8로 설정하였다.

- 돌연변이(mutation) 연산자:

임의의 개체를 확률 0.05 만큼 선정하여 교환 및 이동연산을 수행한다. 교환 확률은 0.1, 이동확률은 0.9로 설정하였다. 교환 연산자는 한 개체에서 두 태스크를 임의로 설정하여 두 태스크의 할당된 노드를 서로 교환한다. 이동연산자는 2개 이하의 태스크를 임의로 선정하여 태스크에 할당된 노드를 무작위하게 변화시킨다.

- 최적 유전자 보전:

집단에서 최적의 비용을 가진 개체는 항상 집단에 남아있도록 보전한다.

- 연산자 수행 회수:

MGA에서 각 온도에서 수행되는 유전연산자 수행 회수는 20번으로 설정하였다.

- GA 집단으로부터 MFA 스핀 매트릭스 생성:

GA 집단에 대해 재생산 연산자를 수행하여 새로이 집단을 만든 후 집단에서의 태스크의 노드 할당 비율로 스핀 매트릭스를 생성한다. 예로서 개체 10개에서 태스크 0의 노드 할당이 0,1,0,1,0,1,0,1,0,0이면 태스크 0의 스핀은 0.6, 0.4, 0.0, 0.0,…로 설정된다.

- SGA(Genetic Algorithm with Simulated annealing):

주어진 온도에서 MFA에서의 평형 상태를 GA 수행 시에도 유지하기 위해서 GA 연산자 적용시 발생하는 비용변화를 SA에서 사용하는 Metropolis Criterion에 따라서 상태 변화를 수행한다. 주어진 Metropolis Criterion은 다음과 같다.

$$\Pr[\Delta C \text{ is accepted}] = \min\left(1, \exp\left(\frac{\Delta C}{T}\right)\right)$$

여기서 ΔC는 연산자 수행시 발생하는 비용변화를 나타내며, T는 현재 온도를 가리킨다.

3.3 MGA 합성 알고리즘

분산메모리 멀티프로세서 병렬시스템에서 효율적인 매핑을 통한 부하균형 유지를 위해 평균장 어닐링(MFA)과 유전자 알고리즘(GA)을 결합한 MGA 합성 알고리즘을 기술한다. 우리의 혼합형 휴리스틱 기법은 두 알고리즘의 장점을 합하여 성능 향상시킬 수 있음을 보여준다.

MGA에서는 개선된 평균장 어닐링 알고리즘을 적용하였다. 즉, MFA에서는 각 온도에서 열 평형 상태에 도달

한 후 각 부하의 각 프로세서에 대한 할당이 확률로 표현되므로, 그 확률에 따라 유전자 알고리즘을 적용하기 위한 개체집단(population)을 만들 수 있다. 따라서 각 온도에서 열 평형 상태 도달 후 만들어진 집단에 대하여 유전자 연산을 수행한 후 집단 내의 유전자 비율에 따라 다음 온도의 어닐링을 위한 상태를 결정한다. 이와 같이 평균장 어닐링과 유전자 알고리즘을 충분히 낮은 온도에서 시스템이 동결(freeze)될 때까지 위의 연산을 반복 수행한다.

변수 및 문제, 노드구조 초기화;
스핀 매트릭스 생성;
부하대 통신비 계수 r 설정;
초기온도 T_0를 설정하여 T= T_0;
while $T \geq$ 최종 온도 T_f **do**
    MFA 수행;
    스핀 매트릭스로부터 GA 집단 생성;
    SGA 수행; /* GA 연산자 수행시 발생하는
    비용함수 변화를 SA의 Metropolis 임계식을
    이용하여 상태변화를 수행함 */
    GA 집단으로부터 MFA 스핀 매트릭스 생성;
    부하대 통신계수 r 조정;
    T= αT

<그림 1. MGA 알고리즘 구조>

4. 시뮬레이션 성능 결과

본 장에서는 3장에서 설명한 MGA 합성 알고리즘을 같은 비용함수를 사용하는 MFA 및 GA-1과 비교한다. 또한 GA-2는 각 노드의 부하의 제곱을 최소화하는 최소 제곱 부하균형 기법을 비용함수로 사용하는 유전자 알고리즘을 나타낸다. 제안된 알고리즘(MGA)의 실험 결과를 상대적으로 우수한 성능을 나타내는 GA-2의 결과와 비교함으로서 객관적인 평가를 수행한다.

실험 환경은 태스크의 크기는 200과 400에 대해서, 멀티프로세서 노드는 격자(mesh)구조로 연결되어있다고 가정하였으며, 각 태스크의 크기는 [1..10]의 값 중 균등분포로 설정하며, 각 태스크 사이의 통신 크기는 [1..5]의 값 중에서 역시 균등분포로 선택한다. 통신의 개수는 일정하게 정하였으며 본 실험에서는 태스크 수의 1, 2, 3배를 통신의 크기로 각각 정하였다. 각 실험에서 같은 문제에 대해서 10회씩 2번 수행하여 총 20회를 수행한 결과의 평균값이다.

4.1 통신대 노드 부하의 비 : r

사용되는 1차식 비용함수에서 계수 r은 노드 부하와 노드사이의 통신 부하간의 균형을 이루기 위하여 사용된다. 초기에 난수 발생기를 이용하여 각 태스크의 노드 할당이 일항 분포를 이루게 하며 이 분포를 스핀(spin)값을 통해서 나타낸다. 초기 분포가 이루어진 후 계수 r은 다음과 같은 식을 이용하여 구해진다.

$$r = \frac{\text{통신부하}}{\text{노드수} \times \text{노드부하}} = \frac{\sum_{i=1}^{N}\sum_{j\neq i}\sum_{p=1}^{K}\sum_{q\neq p} e_{ij}s_{ip}s_{jq}d_{pq}}{K\times\sum_{i=1}^{N}\sum_{j\neq i}\sum_{p=1}^{K} s_{ip}s_{jp}w_iw_j}$$

노드부하와 통신부하의 비를 조정하는 계수 r은 MFA 과정의 각 온도에서 변화된 통신부하를 반영하기 위하여 각 온도에서 다음 식에 따라 변화된다. r_{old}는 전 온도에서 사용된 비율을 말하며 r_{new}는 현재 온도에서 새로이 계산된 계수 r을 가리킨다.

$$r_{new} = \begin{cases} 0.9\times r_{old} & \text{if } r_{new} < r_{old} \\ r_{old} & Otherwise \end{cases}$$

<표 1>는 각 알고리즘에서의 최대종료시간과 GA-2를 기준으로 한 성능 향상 비교를 나타낸다. 기존의 MFA 및 1차 비용식을 이용한 GA-1알고리즘은 최소 제곱 비용함수를 이용한 GA-2알고리즘보다 최대 종료시간이 더 길게 나타난 반면 새로이 제안한 MGA는 평균 9%의 성능 향상을 가져 왔다. 그러나 MGA의 성능향상이 통신의 개수가 적을 때 크게 나타난 반면 통신의 개수가 증가함에 따라 성능향상비가 감소한다. 이는 MFA의 성능향상비에서 알 수 있듯이 노드 부하대 통신비 계수 r의 선택이 적절하게 이루어지지 않은 것에 기인하는 것으로 추정된다. 따라서 계수 r의 변화를 적절하게 수행할 수 있는 알고리즘의 개발이 요구되어진다. 또한 문제의 크기가 커질수록 성능향상이 증가하는 것은 더 큰 문제에 MGA를 적용할 수 있는 가능성을 제시한다. 보여준 결과는 학술발표회 논문의 분량제한으로 일부분 이다.

<표 1. 각 알고리즘의 최대종료시간과 성능 향상 비교>

문제 크기			최대 종료 시간				성능 향상			
N	|E|	K	MFA	GA-1	GA-2	MGA	MFA	GA-1	GA2	MGA
200	200	16	138.2	184.6	147.7	120.2	6%	-25%	0%	19%
	400	16	525.9	389.4	326.7	320.1	-61%	-19%	0%	2%
	600	16	767.2	579.4	544.7	544.3	-41%	-6%	0%	0%
	200	36	84.7	136.6	90.2	72.0	6%	-52%	0%	20%
	400	36	307.3	281.9	222.2	218.5	-38%	-27%	0%	2%
	600	36	559.2	454.9	391.0	388.2	-43%	-16%	0%	1%
400	400	16	279.1	364.7	284.4	222.8	2%	-28%	0%	22%
	800	16	888.1	709.1	618.8	587.0	-44%	-15%	0%	5%
	1200	16	1557.4	1075.2	1041.1	987.5	-50%	-3%	0%	5%
	400	36	152.9	244.9	169.6	128.7	10%	-44%	0%	24%
	800	36	617.0	530.7	415.0	385.2	-49%	-28%	0%	7%
	1200	36	1065.5	850.5	732.8	693.0	-45%	-16%	0%	5%
						평균	-29%	-23%	0%	9%

5. 결 론

분산메모리 멀티프로세서 시스템에서 부하균형을 고려한 새로운 매핑 알고리즘(MGA)을 제안했다. MGA는 MFA와 GA을 결합한 합성 알고리즘으로 다른 알고리즘(MFA, GA-1, GA-2)들과의 성능을 비교한 결과 솔루션 품질과 수행시간 면에서 우리의 혼합형 휴리스틱 방법은 기존의 것들 보다 개선된 실험결과를 얻었다.

참고 문헌

[1] T. Bultan and C. Aykanat, "A New Mapping Heuristic Based on Mean Field Annealing," *Journal of Parallel and Distributed Computing*,16, pp. 292-305, 1992.
[2] K. Park and C.-E. Hong,"Performance of Heuristic Task Allocation Algorithms," *Journal of Natural Science, CUK*, Vol. 18, pp. 145-155, December 1998.