메타휴리스틱스 결합을 이용한 태스크-프로세서 매핑

박경모*, 홍철의**
*가톨릭대학교 컴퓨터정보공학부
**상명대학교 소프트웨어대학 소프트웨어학부
e-mail: *kpark@catholic.ac.kr **hongch@smu.ac.kr

Mapping Tasks to Processors in Combination with Metaheuristics

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

요 약

본 논문에서는 분산메모리 멀티프로세서 시스템에서 태스크와 프로세서 노드간의 매핑에 관한 최적화 문제를 메타휴리스틱스(metatheuristics)의 장점을 효과적으로 결합한 새로운 방안을 소개한다. 태스크-프로세서 할당에 있어 부하균형을 고려한 MFA-GA 하이브리드 알고리즘을 제안하고 기존의 할당 방안들과 성능실험을 통해 비교 분석한다. 우리의 합성 휴리스틱을 이용하면 각 방법을 단독으로 사용하는 것 보다 매핑 품질과 수행시간 면에서 개선된 성능결과를 얻을 수 있음을 보여주었다.

1. 서론

분산메모리 멀티프로세서 시스템에서 태스크들이 프로세서들에 적절하지 않게 배치되면 낮은 프로세서 이용률과 불균형 작업부하로 인하여 손실을 입는다. 이 문제는 매핑(mapping) 문제의 한 형태인 태스크-프로세서 할당 문제로 연결된다[1,2]. 태스크 프로세서 할당이란 병렬 시스템의 여러 프로세싱 노드들에 병렬 프로그램의 통신 태스크 세트를 매핑하는 것이다. 태스크-프로세서 노드의 매핑의 목적은 솔루션의 품질에 손상 없이 수행시간을 최소화하는 것이다. 이러한 매핑 목적 달성을 위해 프로세서들 간의 메시지 전달의 통신비용을 최소화시키면서 태스크들을 노드들에 균등하게 배분해야한다. 즉, 부하균형(load balancing)을 통한 병렬 시스템 성능을 최대화시키는 것은 중요하다.

본 연구에서는 분산메모리 멀티프로세서 노드들을 갖는 병렬처리 시스템에서의 부하균형을 고려한 새로운 매핑 솔루션을 소개한다. 제안 알고리즘은 메타휴리스틱스(metatheuristics)[1] 최적화 기법으로 유전자 알고리즘(genetic algorithms), 시뮬레이티드 어닐링(simulated annealing), 평균장 어닐링(mean field annealing) 등을 포함한다. 이런 기법들 간의 비교 연구가 부족한 실정에서 우리는 휴리스틱 알고리즘의 효과적인 결합에 의한 좋은 품질의 매핑 솔루션을 위한 시뮬레이션 연구 실험결과를 발표한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에서 다룰 멀티프로세서 태스크 할당문제를 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. 태스크-프로세서 할당 문제

본 연구에서 다룰 태스크-프로세서 할당 문제에 대하여 공식적으로 서술한다. 태스크 그래프 $G(V, E)$는 $|V| = N$, $(1, 2, \cdots, i, j, \cdots N)$로 표시되고 공집합이 아닌 정점(vertex)의 집합이다. G의 정점들은 병렬프로그램의 태스크를 나타낸다. 정점 가중치 w_i는 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 $G(P, D)$ 는 $|P| = K$ 노드와 $|D| = \binom{K}{2}$ 모서리를 갖는 모든 정점간 모서리가 존재하는 완전그래프(complete graph)이다. G의 노드들은 $(1, 2, \cdots, p, q, \cdots K)$로 목표로 하는 멀티프로세서를 표시한다. 모서리 가중치 d_{pq}, $(1 \le p, q \le K)$, $p \ne q$ 프로세서 p와 q 사이의 통신비용을 나타낸다. 여기서 태스크 할당문제란 N-to-1 매핑 문제로 $M: V \rightarrow P$를 구하는 것이다. 즉 태스크 그래프 G의 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.

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

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

$M(i)$ 는 태스크 i가 매핑되는 프로세서 p를 표시한다. (1) 식에서 G_p의 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.

스핀 매트릭스(spin matrix)는 N 태스크 행과 K 프로세서 열로 구성되며 표현 방안으로 사용된다. 즉, $N\times K$ 스핀은 솔루션을 인코드 하는데 사용된다. 스핀 (i, p)의 출력 s_{ip} 는 태스크 i를 프로세서 p에 매핑할 확률을 표시한다. 여기에서 s_{ip}는 $0 \le s_{ip} \le 1$ 범위에서 연속 변수이다. MFA 알고리즘이 하나의 솔루션에 도달하면 스핀 값은 그 결과를 0 또는 1로 수렴한다. 만일 s_{ip} 가 1로 수렴되면 태스크 i는 프로세서 p에 매핑되는 것을 의미한다. 태스크-프로세서 매핑에 사용되는 비용함수는 다음과 같다.

$$C(s) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}\sum_{p=1}^{K}\sum_{q\neq p} e_{ij}s_{ip}s_{jq}d_{pq} + \frac{r}{2}\sum_{j\neq 1}^{N}\sum_{j\neq i}\sum_{p=1}^{K} s_{ip}s_{jp}w_i w_j \quad (3)$$

e_{ij} : 태스크 쌍 i 와 j 사이의 모서리 가중치

w_i : 태스크 그래프에서 태스크 i의 가중치

d_{pq}: 프로세서 그래프에서 프로세서 p와 q 사이의 모서리 가중치(edge weight)

$s_{ip} \times s_{jq}$: 태스크 i와 j 가 서로 다른 프로세서 p와 q에 각각 매핑되는 확률

$e_{ij}s_{ip}s_{jq}d_{pq}$: 태스크-프로세서 매핑에 따른 가중치를 갖는 프로세서 간의 통신(inter-processor communication: IPC) 비용

식 (3)에서 4개의 시그마 합계 항은 태스크 그래프 (G_t)의 각 모서리 쌍에 대한 프로세서 그래프(G_p)의 모든 프로세서 쌍을 포함한다. 따라서 이것은 스핀 매트릭스의 값에 의해 표현되는 매핑을 위한 총 IPC 비용을 표시하며 최소화되어야 한다.

식 (3)에서 3개의 시그마 합계 항은 태스크들이 개개의 프로세서들에 매핑된 가중치들의 내부 곱에 대해 합산한다. 동등한 양의 태스크 가중치들이 모든 프로세서에 매핑될 때 전역적인 최소치가 나온다. 만일 불균형 매핑이면 3개의 합계 항을 불균형 매핑으로 인한 페널티를 주면서 불균형 양의 제곱으로 증가시킨다. 파라미터 r 은 상기한 매핑 문제에 있어 두 가지 최적화 목적 간의 균형을 유지하기 위해 고안되었다.

3. 메타휴리스틱스

3.1 평균장 어닐링 알고리즘

본래 평균장 어닐링(mean field annealing, MFA)은 열평형(thermal equilibrium) 상태에 있는 '스핀(spin)'이라 부르는 입자(particle)들 시스템의 상태를 추정하기 위해 물리학의 평균장 근사법(mean-field approximation)에서 나온 것이다. MFA 알고리즘은 시뮬레이티드 어닐링(simulated annealing, SA)와 홉필드 신경망(Hopfield neural network, HNN) 모델을 결합한 것이다. SA 기법은 최적화 문제에서 좋은 솔루션을 구하는데 많은 계산 시간이 소요되는 단점을 가지고 있다. 이 원인은 평형 상태에 도달하기 전에 개개의 온도에서 수많은 무작위 탐색을 수행하기 때문이다. 이 문제를 MFA에서는 SA에 대한 결정적 근사법(deterministic approximation)을 이용하여, 즉 어닐링 프로세스를 통해 얻은 통계자료를 평균함으로써 문제를 해결한다. 다시 말해 MFA는 SA 기법에서 무작위로 상태를 변화시키는 것과 달리, 평균장 근사법을 사용하여 산출되는 평균값으로 대치시키는 방법으로 열(온도) 평형 상태에 빠르게 도달할 수 있다는 장점이 있다.

MFA는 HNN과 긴밀하게 관련되어 있다. HNN란 비동기적 갱신(asynchronous updating)에 기반을 둔 완전그래프로 무작위로 선택된 뉴론(neuron)중 오직 하나의 단위만이 각 시간적 단계에 관여된다. 상호작용하는 뉴론들이 역동적 공동집단 계산(dynamic collective computation) 특성으로 인하여 하드웨어 구현에 잠재적인 강점이 있다. HNN에서는 gradient descent 방법으로 엔코드된 비용함수를 최소화한다. HNN은 보통 작은 크기의 최적화 문제에는 효율적으로 적용이 가능하나, 문제 크기가 커질수록 적용이 곤란하여 확장성(scalability)이 좋지 않은 약점을 기지고 있다. MFA에서는 HNN과 같은 방법으로 상태를 표현한 다음 시스템을 열평형 상태에 도달시키기 위하여 SA에서와 같이 반복기법을 적용한다.

MFA구현 관련해 냉각스케줄(cooling scheduling)은 주어진 문제의 특성과 비용함수에 따라 민감하게 작용하므로 다음과 같이 적절한 스케줄을 선택한다.

- 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 $\varepsilon/1000$ 내에 존재할 때의 온도로 설정한다.

- 임의의 온도 k에서의 Markov 체인 길이 : L_k

각 온도에서의 평형상태에 도달하기 위한 상태변환

의 회수로 연속해서 태스크 수만큼 비용의 변화가 허용치 ε내에 존재할 때의 상태변화 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 태스크 할당, 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 프로세서 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(individual) 수로 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.
- 생식(reproduction) 연산자: 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 남아 있도록 보전한다.
- 연산자 수행 회수: HA에서 각 온도에서 수행되는 유전연산자 수행 회수는 20번으로 설정하였다.
- GA 개체집단으로부터 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 0 의 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 비용 변화를 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 하이브리드 알고리즘

분산메모리 멀티프로세서 병렬시스템에서 부하균형을 고려한 효율적인 매핑을 위한 평균장 어닐링(MFA)과 유전자 알고리즘(GA)을 결합한 하이브리드 알고리즘(hybrid algorithm, HA)을 기술한다.

HA에서는 개선된 평균장 어닐링 알고리즘을 적용하였다. 즉, MFA에서는 각 온도에서 열평형 상태에 도달한 후 각 부하의 각 프로세서에 대한 할당이 확률로 표현되므로, 그 확률에 따라 유전자 알고리즘을 적용하기 위한 개체집단을 만들 수 있다. 따라서 각 온도에서 열평형 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, 노드 구조 초기화;
스핀 매트릭스 생성;
부하대 통신비 계수 r 설정;
초기온도 T0를 설정하여 T= T0;
while T ≥ 최종 온도 Tf do
        MFA 수행;
        스핀 매트릭스로부터 GA 집단 생성;
        SGA 수행; /* GA 연산자 수행시 발생하는
        비용함수 변화를 SA의 Metropolis 임계식을
        이용하여 상태변화를 수행함 */
        GA개체집단으로부터 MFA스핀 매트릭스 생성;
        부하대 통신계수 r 조정;
        T= αT
```

(그림 1) 매핑을 위한 하이브리드 알고리즘

4. 시뮬레이션 실험 결과

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

실험 환경은 태스크 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.

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 다음 식을 이용하여 구해진다.

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

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>은 초기의 계수 r을 유지 시켰을 때와 r값을 각 온도에서 변화 시켰을 때의 각 노드에 할당된 부하의 종료 시간 중 최대종료시간을 표로 나타내었다. <표 1>에서 N은 부하의 수, |E|는 총 통신 개수, K는 노드의 수를 각각 나타낸다. <표 1>에서 알 수 있듯이 r값을 각 온도에서 변화시키는 것이 훨씬 좋은 결과를 나타낸다.

<표 2>는 각 알고리즘에서의 최대종료시간을 나타낸다. 기존의 MFA 및 1차 비용식을 이용한 GA-1 알고리즘은 최소 제곱 비용함수를 이용한 GA-2알고리즘보다 최대 종료시간이 더 길게 나타난 반면 새로이 제안한 HA는 평균 9%의 성능 향상을 가져왔다. 그러나 HA의 성능향상이 통신의 개수가 적을 때 크게 나타난 반면 통신의 개수가 증가함에 따라 성능향상비가 감소한다. 이는 <표 1>의 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 HA를 적용할 수 있는 가능성을 제시한다. 여기서 보여주는 결과는 학술발표회 논문의 분량 제한으로 부분적인 것이다.

<표 1. 부하-통신계수 r에 따른 최대종료시간>

문제 크기			초기 r 유지		계수 r 변화		성능 향상	
N	|E|	K	MFA	HA	MFA	MGA	MFA	HA
200	200	16	336.65	134.65	138.15	120.2	60%	13%
	400	16	707.45	324.15	525.85	320.1	54%	39%
	600	16	845	526.75	767.15	544.25	38%	29%
	200	36	193.35	90.3	84.65	71.95	53%	15%
	400	36	430.5	235.8	307.25	218.45	45%	29%
	600	36	651.55	420	559.15	412.15	36%	26%
400	400	16	627.1	256.4	279.05	222.75	59%	20%
	800	16	1189.95	617.15	888.1	587	48%	34%
	1200	16	1862.8	971.9	1557.4	987.5	48%	37%
	400	36	410.5	143.85	152.85	128.65	65%	16%
	800	36	834.45	410.9	617	385.15	51%	38%
	1200	36	1376.8	714.65	1065.5	692.95	48%	35%
						평균	50%	28%

<표 2> 각 알고리즘의 최대종료시간과 GA-2를 기준으로 한 성능향상 비교표

5. 결 론

병렬처리 시스템에서 부하균형을 고려한 태스크-프로세서 매핑을 위한 하이브리드 알고리즘을 제안했다. 우리의 방안은 메타휴리스틱스(SA,MFA,GA)의 장점을 결합한 알고리즘으로 각 기법을 단독으로 사용하는 것 보다 매핑 솔루션 품질과 수행시간 면에서 더 좋은 성능결과를 가져옴을 확인하였다.

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] Y. K. Kwok and I. Ahmad, "Static Scheduling Algorithms for Allocating Directed Task Graphs," *ACM Computing Survey*, 31(4), pp. 406-417, December 1999.