2001년 한국정보처리학회 추계 학술발표논문집 제8권 제2호

병렬 처리 시스템에서 확장된 유전자 알고리즘을 이용한 태스크 스케줄링 설계

박월선*, 윤성대*

*부경대학교 전자계산학과
e-mail:wspark@unicorn.pknu.ac.kr

A Design of the Task Scheduling using a Extended Genetic Algorithm in Parallel Processing Systems

Weol-Seon Park*, Sung-Dae Youn*

*Dept. of Computer Science, Pukyung National University

요 약

병렬프로그램을 멀티프로세서로 스케줄링하는 문제의 해를 구하기 위하여 본 논문에서는 확장된 유전자 알고리즘을 적용한다. 확장된 유전자알고리즘인 MSEGA는 각 노드의 선행관계에 관한 휴리스틱한 정보와 간단한 일차원 배열구조가 통합된 염색체 코딩방법과 염색체 구성인자 중 우성 유전인자의 형질을 다음세대로 존속시키는 교배연산자와 프로세서 효율성이 고려된 평가 함수등으로 순서제약이 있는 병렬프로그램 스케줄링 문제 및 FFT(Fast Fourier Transform)형태의 데이터 흐름도상에서 관련 연구 중 Hou의 유전자 알고리즘과 BEA(binary-exchange algorithm)에 의한 스케줄링 결과보다 전체실행시간에 있어 MSEGA에 의한 스케줄링이 더 우수함을 보였다.

1. 서론

컴퓨터 성능의 비약적인 향상과 소프트웨어 기술의 발전에 힘입어 해결하기 어려운 NP-complete문제(Non_deterministic Polynomial-complete)인 정보처리, 로봇 제어, 고속의 실시간 처리 등에 대해 멀티프로세서 시스템의 병렬처리로 전체 작업 처리시간을 줄일 수 있게 되었다. 그러나 이러한 병렬처리에 따른 프로세서간의 통신 오버헤드와 낮은 시스템 효율 등으로 인하여 N개의 프로세서들을 가진 멀티프로세서라 해도 반드시 속도향상(speedup)이 N배 된다고는 말할 수 없다[4].

성능저하를 개선하기 위해서 먼저, 병렬 처리할 병렬 프로그램의 크기(grain size)를 적절한 논리적 모듈로 분할한 다음 프로세서들에게 가능한 한 균등한 작업량을 할당하고 프로세서간의 통신량, 즉 교환해야 하는 데이터량이 최소화될 수 있도록 스케줄링 해야한다. 본 논문은 이러한 스케줄링 문제에 대한 해를 구하기 위해서 다양한 탐색과 조합 최적합 타입 문제(combinatorial-optimization type)에 최근 부각되고 있는 유전자알고리즘(Genetic Algorithm : GA)을 적용한다. 관련 연구들 중 Hou 유전자 알고리즘이 특정 height를 가지는 유전인자에 의해서만 교배나 변이가 행해짐으로 좁은 해 탐색 공간으로 인한 준최적해(sub-optimal solution)나 최적해(the best solution)를 유전자 연산자인 교배나 돌연변이에 의해 검색될 가능성이 없을 수 있는 단점을 보안하여 넓은 해 탐색공간과 비교적 빠른 검색을 제공하는 확장된 유전자 알고리즘 기반의 스케줄링 기법을 제안한다(Multiprocessor Scheduling_Extended Genetic Algorithm : MSEGA). 이 기법은 선행관계에 관한 정보와 간단한 일차원적 배열구조를 통합하여 디코딩이 없는 염색체 코딩방법과 염색체 구성인

자 중 우성 유전인자의 형질은 그대로 다음세대로 종속시키면서 교배하는 교배연산자함수 그리고, 세대를 재구성하기 위하여 엘리트(elite)방법과 룰렛휠(roulette wheel)방법을 혼합한 선택방법, 프로세서의 효율성을 고려한 평가함수(evaluation function)로 정법한 해만을 생성한다. 본 논문에서, MSEGA를 적용하는 방법을 설명하기 위해 나타낸 병렬 태스크 그래프의 예는 1988년 Kruatrachue와 Lewis에 의해 행렬곱셈 문제를 20㎒ 사이클의 M68000에서 분석한 것이다[1].

모의실험은 랜덤하게 생성된 병렬 태스크 그래프 상에서 노드 수와 노드를 연결하는 링크 수를 10%, 20% 증가시킨 후, 제안한 알고리즘과 Hou의 유전자 알고리즘을 적용하여 전체작업 수행시간을 비교하였다. 또한 Hou의 알고리즘에서는 통신오버헤드를 고려하지 않았으므로 기존 관련 연구들 중 2진 교환 알고리즘(Binary Exchange Algorithm : BEA)[2]를 대표적인 병렬프로그램들 중 하나인 FFT(Fast Fourier Transform)의 일반적 형태와 제어 및 하드웨어 구현을 쉽게 할 수 있도록 데이터 흐름도를 변형시킨 Constant Geometry FFT에서 비교 실험하였다.

본 논문의 구성은 2장에서 스케줄링 문제를 분석한 후 MSEGA를 적용한 결과를 보이고, 3장에서는 MSEGA에 관하여 기술하고, 4장에서는 랜덤하게 생성된 태스크 그래프와 FFT 형태의 태스크 그래프에서 제안한 알고리즘의 성능 분석을 위하여 기존 알고리즘들과 비교 실험하고, 끝으로 5장에서 결론 및 향후 연구과제를 보인다.

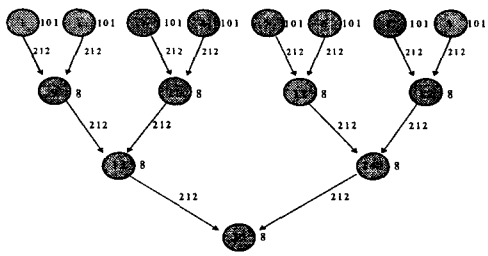
2. 멀티프로세서 스케줄링

본 논문에서는 스케줄링 할 문제를 방향성과 가중치가 있는 그래프로 나타냈으며, 병렬태스크 그래프를 기호로 표현한다면 G=(V, E, W, [d])가 된다. 여기서 V는 병렬프로그램을 적절한 크기(grain size)로 나눈 논리적 모듈단위인 태스크(task)들의 집합으로 원형의 노드로 표시하고, E는 처리될 태스크사이의 선행제약 조건들의 집합으로 방향을 나타내는 간선으로 표시한다. W는 각 태스크의 처리 시간으로 노드 옆에 수치로 나타냈다. [d]는 병렬처리로 인한 프로세서간 통신비용으로 간선 옆에 수치로 나타냈으며 모의 실험에서는 비교대상 알고리즘에 따라 생략 가능하다. 태스크의 번호는 그래프의 레벨(level)정보에 따라 재명명(rename)된다. 그리고 노드들을 실제 병렬처리 할 때는 선행제약조건에 위배되지 않도록 해야한다. 즉, 같은 레벨에 있는 태스크들은 우선순위가 동등함으로 노드의 순서에 상관없이 처리할 수 있지만 만약 각 노드에 연결된 선조 노드들(predecessors)이 수행되지 않았다면 연결된 모든 선조 노드들이 먼저 실행될 수 있도록 동기화를 가진 후 처리 해야만 한다. 또한, 그래프 상의 모든 노드들은 중복 없이 한번은 수행되어야 하며 프로세서들에 의해 병렬로 처리될 때 걸린 전체수행시간은 적을수록 좋은 결과이다.

태스크 그래프에 대한 정확한 이해를 돕기 위해 1988년 Kruatrachue와 Lewis에 의한 정적 멀티 프로세서 스케줄링을 위한 프로그램 분석을 예를 들어 설명한다. 분석될 문제는 두 개의 2×2 행렬A, B를 곱하여 행렬 C를 얻은 후 C의 4개 요소를 합하는 것으로, 최종 결과를 얻기 위해 8번의 곱셈과 7번의 덧셈을 해야 하는 병렬 처리가 가능한 프로그램이다. 그림 1은 위에서 분석한 문제를 태스크 그래프 형태로 재구성한 4-레벨 2진 트리(tree)로 8개의 곱셈은 처음 8개의 노드에서 수행되고, 7개의 덧셈은 나머지 7개의 노드에서 수행됨을 나타낸다.

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

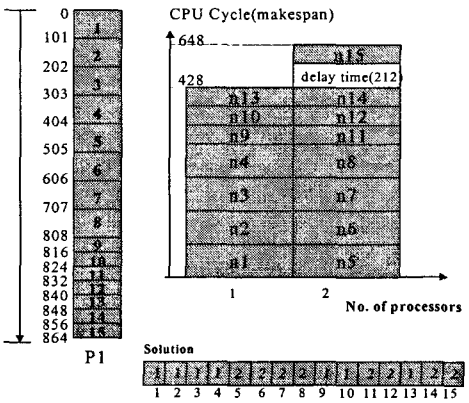
$$Sum = C_{11} + C_{12} + C_{21} + C_{22}$$



(그림1) 태스크 그래프

(그림2) 멀티프로세서 상호관계

그림 2는 태스크를 처리할 모든 프로세서는 동일한 성능을 가지며 서로 완전 연결되어 있음을 나타낸다. 그림 3의 (a)는 그림 1에 대하여 단일 프로세서에 의한 순차 실행으로 통신지연이 발생하지 않은 상태로 15개의 노드를 처리하는데 총 864 사이클이 걸림을 나타내고, 그림 3의 (b)는 MSEGA를 이용하



(a) sequential schedule (b) parallel schedule with MSEGA

(그림 3) 스케줄링 결과

여 두 개의 프로세서로 병렬 처리한 결과가 648사이클이 되는 최적해들 중 하나를 보인 것이다. 산출된 해가 나타내는 의미는 첫 번째 프로세서가 병렬 태스크들 중 1,2,3,4,9,10,13을 처리하고 두 번째 프로세서가 5,6,7,8,11,12,14,15태스크를 처리한다는 것이다.

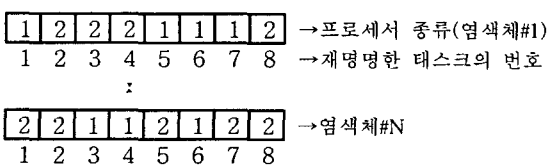
3. MSEGA

유전자 알고리즘을 어떤 문제 해결에 사용하기 위해서는 해를 염색체 형태로 코드화 하는 방법, 새로운 염색체를 생성시키는 유전 연산자인 교배와 돌연변이의 방법, 염색체에 의해 표현되는 해의 적합도를 측정하는 평가함수 그리고 모집단을 구성하는 방법이 고안되어야 하며, 위 요소들이 어떻게 구성되어지는 가에 따라 유전자 알고리즘의 특성과 성능이 결정된다[3].

3.1 초기 모집단 구성

본 논문에서의 코딩방법은 병렬 프로그램 그래프의 레벨에 따라 태스크의 번호를 재명명(rename)한 후 태스크를 처리할 프로세서의 종류로 표시한다.

초기모집단구성은 모집단의 크기만큼 프로세서의 종류를 그림4와 같이 랜덤하게 생성한다.



(그림 4) 초기 모집단 구성

3.2 평가

MSEGA로 탐색된 해의 문제에 대한 적합도를 평가하기 위해 프로세서의 효율성과 전체실행시간에 중점을 둔 식(1)를 사용하여 적합치(fitness value)를 계산한다. 적합치가 클수록 좋은 해가된다.

$$Fitness = \alpha \times (\beta - pmax_i) \qquad (1)$$

where,

$pmax_i = max(pms_i)$ $\quad i=1, \ldots, Pro\_num, \quad j=1, \ldots, Pop\_size$

$gmax = max(pmax_i)$

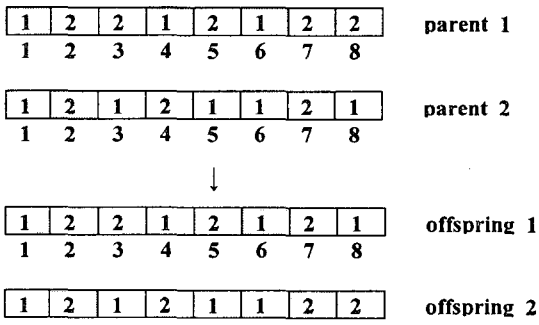
$$\begin{cases} \alpha = 500, \text{if} \left(\dfrac{\min(pms_i)}{\max(pms_i)} > 0.9\right) \\ \alpha = 10, otherwise \end{cases}$$

$\beta = gmax \times 0.001 + gmax$

pmsi : Each processors makespan yielded by a schedule $\quad i = 1,, Pro\_num$

3.3 교배

새로운 염색체를 생성하기 위해서 교배비율만큼 랜덤하게 부모1, 2를 선택한 후 부모1에서 실행시간이 가장 적은 프로세서의 유전자 위치(우성 유전인자)를 자식1에게 유전시킨 후 마스크 값에 따라 0이면 부모1에서, 1이면 부모2에서 유전인자를 선택한다.



(그림 5) 교배 과정

3.4 돌연변이

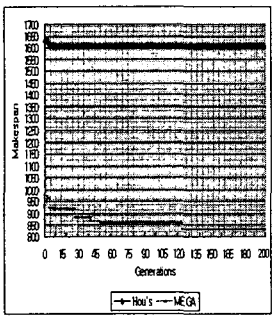
넓은 해 공간에 대해 최적 해를 탐색할 수 있도록 모든 유전인자에 대하여 랜덤하게 임의의 수를 발생했을 때 그 수가 돌연변이 비율보다 작은 값이면 그 위치의 유전인자에 프로세서의 종류를 랜덤하게 발생하여 대체한다. 돌연변이 된 유전인자를 가진 새 염색체도 빠른 수렴을 위해서 우수한 해가 될 수 있는 가능성이 있을 때만 유전되게 한다.

4. 모의 실험

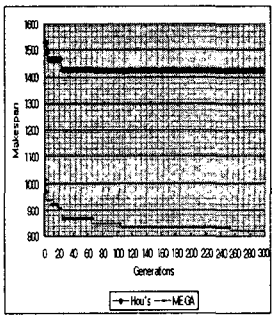
랜덤하게 생성된 병렬 태스크 그래프의 작업노드 수를 20, 50, 100개로 변화시키며, 각 노드의 실행비용은 1에서 100까지 하였다. 프로세서 수는 2에서 6으로 하고, 프로세서간의 통신비용은 40으로 정하였다. 각 세대의 모집단의 크기는 7에서 40사이로 하며, 각 세대에서의 최적해가 30세대 이상 계속 동일하게 나타날 때를 최종 해로의 수렴조건으로 하였다. 또한 교배와 돌연변이를 발생시키기 위한 비율은 경험에 의한 확률치로써 0.4(P_c), 0.2(P_m)로 설정한 MSEGA와 Hou의 유전자 알고리즘을 적용하여 스케줄링한 후 작업완료시간(makespan)을 비교하였다. 또한 각 그래프에 노드를 연결하는 링크 수를 총 노드 수의 10%, 20%를 추가하여 노드간의 종속성을 강화시킨 후 다시 스케줄링 하여 비교하였다. 그리고, 랜덤하게 생성된 그래프 대신 이미 잘 알려진 병렬 프로그램들 중 하나인 FFT(Fast Fourier Transform) data flow의 일반적 형태에 적용하여 실험하였다. 마지막으로, 통신비용 파라메터가 추가된 Constant Geometry FFT에 대해서도 제안한 알고리즘의 성능을 분석하기 위해, 통신비용을 전혀 고려하지 않은 Hou 알고리즘 대신 비교적 작은 n-point FFT 병렬 연산 수행 시에 효율적인 2진-교환 알고리즘(BEA)을 적용하였다. 그림 6과 그림 7은 실험들 중 랜덤하게 생성된 그래프의 작업노드 수가 100일 때와 통신비용이 고려되지 않은 FFT에서 Hou와 MSEGA를 각각 비교한 것이다.

5. 결론

제안한 유전자 알고리즘을 NP-complete 문제인 멀티 프로세서 스케줄링 문제에 적용하였을 때, 산출된 해의 타당성과 유용성을 증명하기 위하여 Hou의 유전자 알고리즘, 통신비용이 고려된 FFT를 스케줄링하는 기존의 BEA방법과 전체 작업완료시간에 대하여 비교 실험하였다.



(그림 6)작업노드 수100



(그림 7)16point FFT

모의실험 결과 랜덤하게 생성된 그래프에서는 제안한 알고리즘이 Hou의 알고리즘보다 평균 1.4배의 speedup을 보이며, 10%, 20% 추가된 링크수의 그래프와 통신비용이 고려되지 않은 FFT에서도 각각 평균 1.5배의 향상을 보였다. 통신비용이 고려된 FFT에서는 제안한 알고리즘이 BEA 알고리즘보다 평균 1.6배의 speedup을 보였다. 제안한 방법의 장점은 프로세서 수에 제한 없이 통신비용과 링크 수가 추가되고 또한, 노드수가 많아질수록 비교 알고리즘들보다 전체수행시간이 짧은 해를 검색할 수 있다는 것이다.

향후 연구 과제로는 NP-complete 문제인 멀티프로세서 스케줄링 문제에 더 많은 고려사항을 추가한 후, 제안한 유전자 알고리즘을 병렬 시스템 상에서 실제 적용하는 것이다.

참고문헌

[1] Kai Hwang, Advanced ComputerArchitecture, 3rd Ed, McGraw-Hill, 1993

[2] Vipin Kumar, Ananth Grama, Anshul Gupta and George Karypis, Introduction to Parallel Computing, the Benjamin/Cummings, 1997

[3] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3rd Ed., Springer-Verlag, 1995

[4] 김종현, 병렬컴퓨터구조론, 생능출판사, 1996