

병렬 컴퓨터 시스템에서의 FFT 데이터 흐름도에 관한 유전 스케줄링 알고리즘

박월선*, 김금호*, 서루비**, 윤성대*

부경대학교 전자계산학과*

부경대학교 전산교육학과**

전화 : (051) 620-6398 / 팩스 : (051) 620-6398

Genetic Scheduling Algorithm for FFT Data Flows in Parallel Computers

Weol Seon Park*, Gum Ho Kim*, Ru By Suh*, Sung-Dae Youn*

Dept. of Computer Science, Pukyong National University*

Dept. of Computer Science Education, Pukyong National University*

E-mail : wspark@dol.puku.ac.kr

Abstract

We propose the genetic algorithm to apply three kinds of FFT data flows to be considered the overhead for the data exchange between processors that have the multi-scheduling problem on parallel computer. In the design of genetic algorithm, we propose the chromosome representation which can simply encode and decode a solution without any heuristic information, the evaluation function to be considered an efficiency of processor, and the genetic operator to inherit a superior gene from their parents. And we saw that the simulation result can verify better performance than the existing algorithm(BEA : binary exchange algorithm)in the face of execution time.

I. 서론

이산 푸리에 변환(DFT : discrete Fourier transform)은 시계열과 웨이브 분석, 회선, 디지털 신호 처리와 이미지 필터링을 포함하는 많은 과학적, 기술적인 응용에서 중요한 역할을 한다. DFT는 사인 곡선과 같은

주기적인 신호 사이클로부터 일정하게 샘플된 n 개의 점을 신호 주파수 스펙트럼으로 사상하는 선형 변환이다. 1965년에, Cooley와 Tukey는 $\Theta(n^2)$ 연산을 요구하는 DFT를 보다 빨리 수행하기 위해서 연산량을 $\Theta(n \log n)$ 으로 줄인 새로운 알고리즘 고속 푸리에 변환(FFT : fast Fourier transform)을 제안했다. 과학과 기술 분야에서의 푸리에 변환의 중요한 역할로 인해, 벡터와 병렬 컴퓨터 상에서 FFT 구현과 그것의 성능 평가에 지대한 관심을 가지고 연구되고 있다[1]. 그런데, N 개의 프로세서들을 가진 병렬컴퓨터의 속도 향상이 N 배가 되지 못하는 주요 원인은 병렬처리에 따른 오버헤드와 낮은 시스템 효율 때문이다. 이 원인에 의한 성능 저하를 개선하기 위해서 프로세서들에게 가능한 한 균등한 작업량을 할당하고 프로세서간의 통신량, 즉 교환해야 하는 데이터량이 최소화 되도록 스케줄링 해야 한다. 따라서, 제출된 작업의 응답시간을 줄이고 사용된 프로세서의 효율적 이용으로 경제적 비용을 감소시키는 스케줄링 문제에 관한 알고리즘이 많이 연구되고 있다. 이러한 기존 연구 중 2진-교환 알고리즘(BEA : binary-exchange algorithm)은 병렬 컴퓨터 시스템상에서 특히, 하이퍼큐브 구조에서 노드수가 적은 FFT에 효율적인 스케줄링 방법이다[1].

본 논문에서는 FFT에 대한 새로운 스케줄링 방법인 유전자 알고리즘 기반의 스케줄링 알고리즘을 제안한

다[2]. 유전자 알고리즘은 적자생존 또는 자연선택에 의해 자연계가 진화해왔다는 가설에 기반 한 문제해결 기법이다. 이는 랜덤하게 생성된 초기 해(생명체)들로부터 해의 적합치(주변환경의 적응능력)를 나타내며 복잡한 수학적 지식을 요구하지 않는 단순한 목적함수를 기준으로 해를 선택한다. 그 다음 교배와 돌연변이라는 간단한 유전자 연산자를 통해 계속 진화하다가 해를 요구하는 시점에서 그때까지 진화된 가장 우수한 해를 산출하는 방법이다. 이러한 유전자 알고리즘을 기반으로 한 제안된 스케줄링 알고리즘을 프로세서간 통신 오버헤드를 고려한 세 가지 형태의 FFT 데이터 흐름도에 적용하여 기존 스케줄링 방법중 하나인 BEA 과의 전체 실행시간을 비교한다.

제안된 유전자 알고리즘에서는 초기 해 구성 시 정법한 해를 생성하는 새로운 염색체 코딩방법 및 스케줄링 방법, 세대를 재구성하기 위하여 우수한 해를 판단 및 선택하는 방법, 해의 유전인자 중에서 우수한 유전인자는 그대로 종속시키면서 교배한 후 새로 생성된 교배해가 기존 해보다 더 적합할 가능성이 있을 때만 다음 세대로 유전시키는 교배 연산자 등을 보인다.

본 논문의 구성은 먼저 2장에서 일반적인 유전자 알고리즘 이론에 대해서 간략히 소개하고, 3장에서 FFT 스케줄링 문제에 제안된 유전자 알고리즘 적용방법을 기술한다. 그리고 4장에서는 제안된 유전자 알고리즘의 타당성을 입증하기 위해서 실험결과를 보인 다음, 마지막 5장에서 결론을 맺는다.

II. 유전자 알고리즘 구현

선행 제약과 통신부하가 있는 다양한 형태(topology)의 FFT를 멀티프로세서 시스템상의 병렬 프로세서들로 스케줄링 제약조건에 위배되지 않게 효율적으로 분배해서 각 프로세서들의 응답시간을 감소시키고 시스템의 처리능력을 향상시키는 스케줄링 결과 값을 유전자 알고리즘을 이용하여 검색한다. 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 한다. 이러한 요소들의 구성에 의해 유전자 알고리즘의 특성과 성능이 결정된다.

제안된 유전자 알고리즘은 다음과 같다.

<Procedure advanced genetic algorithm>

Step 1 :

Read the DAG and build a database which

includes the adjacency list(node $n \times n$), the cost list, the rate of crossover(P_c), the rate of mutation(P_m), the size of populations(Pop_size), the number of generations(T) and so on

Step 2 :

Randomize the chromosome based on the sort of processors to generate an initial population of size(Pop_size).

Step 3 :

For $i=2$ to T

Apply the scheduling function and compute fitness value with evaluation function.

Select chromosomes based on the modified roulette wheel.

Execute GA operators, Crossover and mutation, for each ratio.

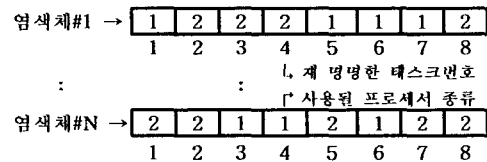
end for

Step 4 :

Select the best solution at the current generation.

2.1 초기 모집단 구성 (Initialization)

기존 연구에서는 스케줄링 제약 조건에 위배되지 않게 해를 구성하기 위해서 선조(predecessor)의 수나 계승자(successor)의 수를 계산한 정보를 기반으로 하여 해를 구성하지만[3]-[5], 본 논문에서 제안된 염색체 코딩방법은 쉽고 간단하게 병렬 프로그램을 나타내는 그래프의 level에 따라 태스크의 번호를 재 명명(rename)한 후, 태스크 속성에는 태스크를 처리할 프로세서의 종류로 표시하는 방법을 도입함으로써 같은 프로세서 내에서 선조를 처리하기 전에 자신을 처리하는 불법의 해(illegal solution) 또는 비 가능해(infeasible)를 생성하지 않는다. 따라서 초기 인구를 구성하기 위하여 모집단의 크기만큼 사용될 프로세서의 종류를 그림1과 같이 랜덤하게 생성한다.



(그림 1) 초기 모집단 구성

2.2 평가 (Evaluation)

각 염색체의 적합치를 평가하기 이전에 다른 프로세서 상에서의 위법한 해 또는 불가능한 해를 생성하지 않도록 스케줄링 한 다음 각 프로세서의 실행시간

(makespan)에 대해 식(1)과 같이 제안된 평가 함수에 의해 적합치(fitness value)를 구한다.

프로세서의 효율성을 고려하기 위하여 검색체를 스케줄링 했을 때 사용된 프로세서 중 가장 긴 실행시간을 가진 프로세서(max(pmsi))에 대해서 가장 짧은 실행시간을 가진 프로세서(min(pmsi))의 시간비율이 0.5보다 작으면 파라미터 α 값을 적절하게 변화시켜 다음 세대로 유전될 확률을 감소시키며, 모집단에서 가장 긴 실행시간을 가진 검색체라도 여러 번의 세대교체가 일어난 후에는 좋은 해가 될 가능성이 있음을 고려해서 적합치 함수는 식(1)과 같이 구현된다.

$$Fitness = \alpha \times (\beta - \max(pms_i))$$

$$\text{Where, } \textcircled{1} \begin{cases} \text{if} \left(\frac{\min(pms_i)}{\max(pms_i)} < \frac{1}{2} \right), & \alpha = 500 \\ \text{otherwise,} & \alpha = 10 \end{cases}$$

$$\textcircled{2} \quad pmax = \sum_{i=1}^n \max(\max(pms_i))$$

$$\beta = pmax \times 0.001 + pmax \quad (1)$$

여기에서 구한 적합치에 따라 현재의 해가 스케줄링의 목적에 어느 정도 적합한가를 판단하는 기준이 되며 적합치가 클수록 좋은 해다.

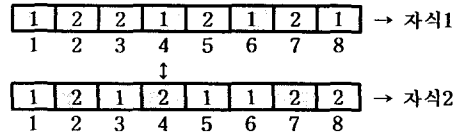
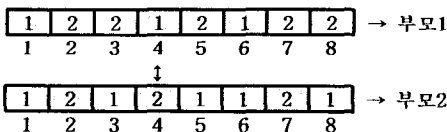
2.3 선택 (Selection)

적자생존 및 자연선택의 방법에 따라 자연환경에 잘 맞는 생명체만 계속 다음 세대로 유전되어 진화하듯이 우수한 해를 선택하여 다음 세대로 이전시키기 위해서 제안된 선택기법은 적합치를 내림차순으로 정렬한 다음 제일 좋은 해(실행시간이 가장 짧은 해)를 일부 선택하고 나머지는 룰렛 휠(roulette wheel)방법[4]으로 새로운 모집단을 구성한다.

2.4 유전자 연산자(Genetic operation)

2.4.1 교배 (Crossover)

제안된 교배방법은 새로운 검색체를 생성하기 위해서 교배비율만큼 랜덤하게 부모1, 2를 선택한 후 부모1에서 실행시간이 가장 적은 프로세서의 유전자의 위치(우성 유전인자)를 자식1에게 유전시킨 후 부모2에서 사용된 프로세서의 종류를 랜덤하게 발생하여 나머지 유전인자를 결정한다. 자식2도 부모2에서 위와 같은 과정으로 생성한다. 즉, 그림 1의 검색체 #1을 부모1로 가정하면 사용된 프로세서 실행시간을 알 수 있으므로 자식1이 그림2와 같이 생성된다.



(그림 2) 자손검색체 생성과정

교배를 통해 새로 생성된 검색체는 무조건 다음 세대로 전이되는 것이 아니라 빠른 수렴을 위해서 우수 한해가 될 수 있는 가능성이 있을 때만 유전된다.

2.4.2 돌연변이 (Mutation)

돌연변이방법은 더 넓은 해 공간을 가지고 최적 해를 탐색할 수 있도록 모든 유전인자를 대상으로 한다. 임의의 수를 발생시켜 그 수가 돌연변이 비율보다 작 은 값이면 그 위치의 유전인자에 프로세서의 종류를 랜덤하게 발생하여 대체한다. 돌연변이된 유전인자를 가진 새 검색체도 빠른 수렴을 위해서 우수한 해가 될 수 있는 가능성이 있을 때만 유전되게 한다.

III. 실험 (Simulation) 및 분석

본 논문에서는 제안된 스케줄링 알고리즘의 유용성과 실용성을 보이기 위해 그림4, 5, 6과 같은 다양한 형태의 FFT 데이터 흐름도에 대해서 BEA 스케줄링 방법과 비교 실험하였다. BEA는 한 비트 위치가 다르게 표시된 프로세서 쌍들 사이에서만 데이터 교환이 발생 하는 알고리즘이다.

시뮬레이션에 사용된 태스크들의 처리시간은 동일하게 설정하였으며, 통신비용은 서로 다른 프로세서사이 에서 데이터 교환이 이루어질 때만 계산하였다. 실험 에 사용된 모든 파라미터는 실험을 통해 얻어진 경험치로써 각 세대의 개체수는 60~90이고, 최대 세대수는 500으로 하며, 프로세서 수는 2, 교배확률은 0.4이고 돌연변이 확률은 0.2이다. 그리고 각 n-point FFT 문제에 제안된 알고리즘(PGA : parallel genetic algorithm)을 적용하여 각각 5회 반복 실험한 후 빈도가 높은 작업 완료시간으로 BEA와 비교하였다.

그림 7은 일반적인 FFT 형태인 그림4에 관해서 각 노드의 수행시간을 임의로 5로 가정한 후, 프로세서간 데이터 교환 시 7, 10, 15, 17, 20의 통신 오버헤드를 각각 부과했을 때 전체 수행시간을 보인다. 7, 10, 15에서는 기존 BEA와 동일한 수행시간을 보이며, 17이상에서는 PGA가 더 높은 적응력을 보여 전체 수행시간이 줄어들었다. 그림 8은 FFT 데이터 흐름의 보다 빠른 계산을 위하여 높은 radix를 사용한 그림 5에서 각 노드의 수행시간을 10으로 가정한 후, 수행한 결과로 통신부하 10이상에서 전체 수행시간이 줄어들었다.

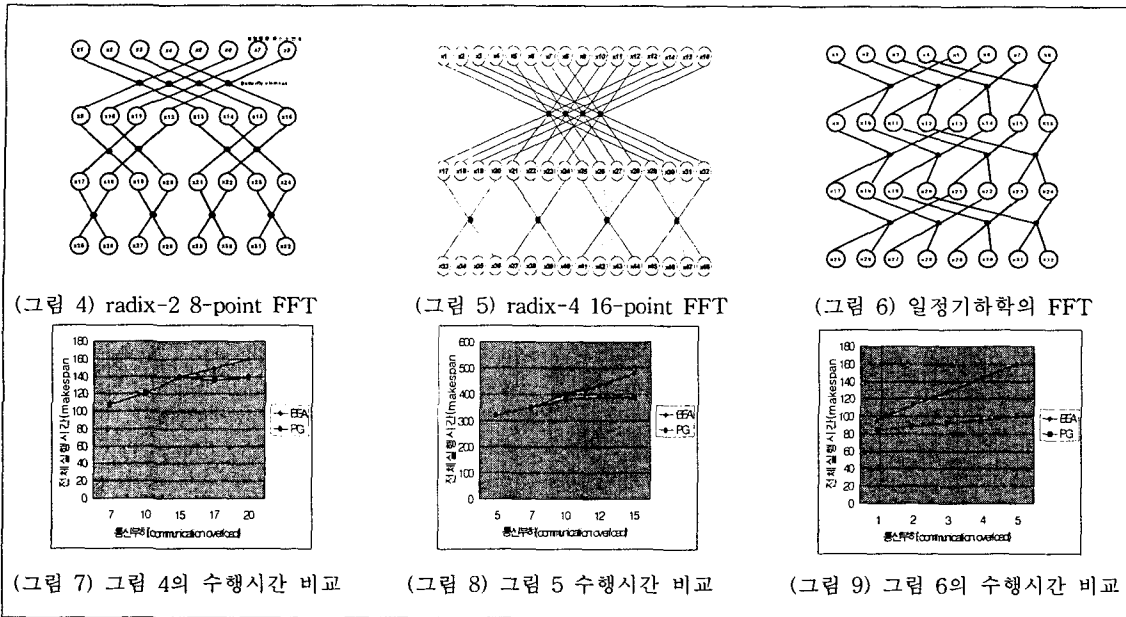


그림 9는 FFT의 제어를 보다 용이하게 하기 위하여 데이터 흐름을 변형한 그림 6의 FFT에 관하여 각 노드의 수행시간을 5로 가정한 후, 스케줄링한 결과로 각 통신부하에 대하여 전체적으로 수행시간이 BEA보다 더 짧게 나타나는 것을 알 수 있었다.

여러 종류의 FFT 데이터 흐름에 대해서 PGA를 적용한 스케줄링 결과가 기존 BEA보다 비교적 짧은 전체실행시간을 보이며 초기세대(약 30세대 이내)에서 수렴하였다. 특히, 그림 5, 6형태의 FFT에서 좀 더 좋은 스케줄링결과를 보였다.

IV. 결론

순서제약과 서로 다른 프로세서간의 통신부하를 고려한 다양한 형태의 FFT 데이터 흐름도를 병렬 컴퓨터상의 멀티프로세서로 스케줄링하는 문제를 풀기 위하여 새로운 방법인 유전자 알고리즘 적용하는 방법을 제안했다. 제안된 스케줄링 알고리즘(PGA)은 정법한 해만을 간단하게 생성하는 염색체 코딩방법과 세대별 가장 우수한해 즉, 실행시간이 가장 짧은 해들을 일부 선택하고 룰렛 휠 방법으로 나머지를 구성하는 세대 재구성 방법, 프로세서의 효율성을 고려한 목적함수, 우성 유전인자를 유전시키는 교배연산 등을 제안하여 실험하였다.

제안된 스케줄링 알고리즘과 기존 BEA 스케줄링 알고리즘을 비교 실험한 결과 BEA에 의해 생성된 전체 실행시간에 근접하거나 더 짧은 실행시간을 가질 수 있는 스케줄링방법을 검색함으로써 제안된 알고리즘의

타당성을 알 수 있었다.

향후 연구 과제는 FFT의 문제크기를 대폭 증가시킨 경우에 제안된 알고리즘을 적용하는 것과 FFT를 응용한 분야에 제안된 유전자 알고리즘을 적용하여 병렬 시스템 상에서 실제 구현하는 것이다.

참고문헌

- [1] V. Kumar, A. Grama, A. Gupta, and G. Karypis, Introduction to Parallel Computing Design and Analysis of Algorithm. 1994
- [2] 박월선, 박상일, 남은미, 윤성대, "효율적인 멀티프로세서 스케줄링을 위한 유전자 알고리즘 설계", 한국정보처리학회, 13회 춘계학술대회
- [3] Ricardo C. Corrêa, "Scheduling Multiprocessor Tasks with Genetic Algorithms," IEEE Trans. Parallel and Distributed Systems, Vol. 10, No. 8, Aug. 1999
- [4] Albert Y. Zomaya, "Genetic Scheduling for Parallel Processor Systems: Comparative Studies and Performance Issues," IEEE Trans. Parallel and Distributed Systems, Vol. 10, No. 8, Aug. 1999
- [5] E. Hou, N. Ansari and H. Ren, "A Genetic Algorithm for Multiprocessor Scheduling." IEEE Trans. Parallel and Distributed Systems, Vol. 5, No. 2, Feb. 1994 pp. 113-120