



A Study on The Deffering Method of The Optional Tasks in The Imprecise Real-Time Scheduling

Gi-Hyeon Song^{*}, Keun-Hyeong Jeon

Department of Management Information System

ABSTRACT

The imprecise computation technique ensures that all time-critical tasks produce their results before their deadlines by trading off the quality of the results for the computation time requirements of the tasks. The NORA algorithm produces minimum total error but has low schedulability problem by prior executing the optional parts of the earliest deadline tasks than the mandatory parts of the latest deadline tasks. To improve this problem, the DOP algorithm which delays the optional parts of the earliest deadline tasks but this algorithm seems to be almost similier with the previous famous SMF(Schedule Mandatory First) algorithm. Therefore, this paper proposes a new method which delays the optional parts of the earliest deadline tasks and demonstrates several superior advantages by comparing and analyzing in the aspects of the schedulability and total error with previous the NORA, DOP(SMF) algorithms.

© 2014 KKITS All rights reserved

KEY WORDS : Imprecise computation technique, Minimum total error, Schedulability, Delay optional parts, On-line scheduling

ARTICLE INFO: Received 21 January 2014, Revised 10 February 2014, Accepted 14 February 2014.

1. 서론

온라인 실시간 시스템에서는 태스크들이 일정하게 발생하지 않는다. 또한, 외부환경의 영향을 받

^{*}Corresdponing author is with the Department of Management Information System, Daejeon Health Sciences College, 21 Chungjeong-ro, Dong-Gu, Daejeon KOREA

E-mail address: ghsong@hit.ac.kr

아 순간 과부하가 걸릴 수도 있다. 이러한 과부하 상황에서는 비정밀 계산이 필요하게 된다 [1~6, 9]. 본 논문에서는 단일 프로세서 상에서 비주기적으로 발생하는 비정밀 태스크들을 다룬다. 비정밀 계산 기법에서 각 태스크는 필수부분과 그 뒤에 이어지는 선택부분으로 구성되어진다. 필수부분은 반드시 그 태스크의 만기 이전에 실행이 완료되어야 하는 것 이고 [10] 선택부분은 선택부분이 포

함된 그 태스크나 다른 태스크들의 만기에 맞추기 위해 필요하면 그 선택부분이 실행 완료되기 전에 중단될 수도 있다 [2]. 즉, 비정밀 태스크는 필수 부분과 선택부분으로 나뉘는데, 필수부분은 만기 전에 반드시 완료 되어야 하는 것이고, 선택부분은 필수 실행부분이 완료된 이후에 결과의 질을 높이기 위해 수행된다 [8].

온라인 실시간 스케줄링 알고리즘인 NORA 알고리즘 [2]은 태스크들의 필수부분들을 보장하기 위해 예약리스트를 관리한다. 예약 리스트를 할당하는 순서는 EDF(Earliest Deadline First)와 반대로 만기가 늦은 태스크들을 먼저 뒤쪽에 할당하는 정책을 쓴다. 그러나 새로운 태스크가 도착하면 필수부분을 할당할 수 없어 스케줄 성공률(Schedulability)이 떨어지는 단점이 있다.

다시 말하자면, 최소의 총 오류를 산출하지만 만기가 빠른 태스크들의 선택실행부분들이 만기가 늦은 태스크들의 필수실행부분들보다 먼저 실행됨으로서 모든 태스크들의 필수실행부분들에 대한 스케줄가능성이 낮아지는 문제가 발생한다. 그 단점을 보완하기 위해 최근에는 선택적 태스크들의 지연방법을 사용한다. 이러한 방법을 사용하는 가장 최근의 연구인 DOP 알고리즘 [1]은 태스크들의 선택부분들의 실행순서를 조절하는 것인데 필수부분들을 먼저 스케줄 하고, 선택부분들은 나중에 스케줄 하는 것이다. 그래서 최대한 지연 가능한 선택부분들 즉, 만기가 늦은 선택적부분들이 스케줄 되어 진다. 그럼으로서 이후에 발생할 새로운 태스크들의 필수적부분들에 유연하게 대처 할 수 있게 된다. 그러나 DOP 알고리즘은 NORA 알고리즘보다 필수부분들의 스케줄 가능성을 높이지만 지연된 선택부분들의 실행요구시간의 합에 비례하여 총 오류가 증가하는 단점이 있으며 스케줄링 과정에서 만기 순서 위반 (Deadline Order Violation) 현상이 발생한다. 또한, DOP 알고리즘은 그 스케줄

링 방법 면에서 기존의 SMF (Schedule Mandatory First) 알고리즘과 매우 유사하다고 판단된다.

그래서, 본 논문에서는 비정밀 태스크들의 선택적부분들을 DOP 알고리즘 처럼 최대한 지연시키지 않고 특정 한계를 계산하여 그 한계까지만 지연시키어 필수적부분들에 대한 스케줄가능성을 높이며 DOP 알고리즘 이나 SMF 알고리즘의 총 오류와 비슷한 수준의 총 오류를 산출하며 만기 순서 위반이 전혀 발생하지 않는 새로운 알고리즘인 DMO (Deffering Method of the Optional Tasks) 알고리즘을 제안하고 스케줄가능성과 총 오류 면에서 그 성능을 비교 분석하여 본 알고리즘의 우수성을 입증하고자 한다. 본 논문의 구성은 다음과 같다. 제 2 장은 비정밀 실시간 스케줄링의 관련 연구들을 기술하고 제 3 장은 기존의 선택적 태스크 지연 방법들에 대하여 기술하고 제 4 장에서는 선택적 태스크를 지연 시키는 새로운 방법에 대하여 기술한다. 마지막으로 제 5 장에서는 결론을 기술하겠다.

2. 관련 연구

실시간 온라인 스케줄 환경에서 다루어지는 비정밀 태스크는 다음 속성을 갖는다. 선점형 비정밀 태스크 $T = \{T_1, T_2, T_3, \dots, T_n\}$ 가 주어졌을 때, 태스크 T_i 는 $\{R_i, D_i, M_i, O_i\}$ 로 표현한다. 여기서, R_i 는 준비시간, D_i 는 만기, M_i 는 태스크 T_i 의 필수부분의 수행요구시간, O_i 는 태스크 T_i 의 선택부분의 수행요구시간을 나타낸다 [6, 8].

2.1 SMF 알고리즘

SMF (Schedule Mandatory First) 알고리즘은 필수 실행부분들이 있으면 이 부분들을 먼저 실행하

고 실행할 필수부분이 더 이상 없다면 선택부분들을 실행하는 것이다. SMF 알고리즘은 필수 실행부분을 먼저 실행함으로써 뛰어난 필수 실행부분의 스케줄링 성공률을 보인다.

그러나 실행 가능했던, 만기가 빠른 태스크들의 선택부분들을 실행하지 못함으로써 총 오류가 현격히 증가하게 된다.

2.2 NORA 알고리즘

NORA 알고리즘은 비정밀 태스크들의 필수부분들의 실행을 보장하기 위해 예약리스트를 관리한다. 이 예약리스트는 만기가 늦은 태스크 순으로 필수부분들을 뒤쪽에 할당하여 채워 나가는 방식을 사용한다. 그리고 이 예약리스트는 갱신되는 시점이 있는데 새로운 태스크가 도착하거나 태스크가 완료되는 시점이 바로 그 시점이다. 스케줄링시, 현재 시간이 예약된 시간에 도달하게 되면 예약여부를 알아본다. 만일 현재 처리중인 태스크가 예약되어 있으면 예약리스트에 할당된 것을 취소하고 처리중인 태스크를 계속 실행한다. 만일 현재 처리중인 태스크가 예약되어 있지 않으면 태스크를 종료시키고 도달된 예약 태스크를 처리한다. 이렇게 스케줄함으로서 비정밀 태스크들의 모든 필수적부분들이 만기 이전에 완료되고 총 오류가 최소화 된다. NORA 알고리즘은 특정한 상황과 제약조건 하 (FMC : feasible Mandatory Constraint)에서만 최소 오류를 가지게 된다. 제약조건 이외의 상황에서는 다른 알고리즘들로는 스케줄 가능한 태스크 집합을 NORA 알고리즘으로 스케줄 할 수 없는 상황이 발생하기도 한다.

즉, 예약리스트에 할당할 수 없는 상황이 발생하여 필수적부분들의 스케줄 성공률이 낮아지는 결과를 초래하기도 한다.

표 1. 태스크 집합 $T = \{T_1, T_2, T_3\}$
Table 1. Task Sets $T = \{T_1, T_2, T_3\}$

	R_i	M_i	O_i	D_i
T_1	0	3	4	8
T_2	0	7	7	14
T_3	5	8	2	16

스케줄링 예제로 <표 1> 과 같은 태스크 집합을 가정해 보자. 이 태스크 집합에 NORA 알고리즘을 적용해 본다. NORA 알고리즘은 예약 리스트를 관리하는데 만기가 늦은 태스크들의 필수부분들을 뒤쪽에 할당하여 채워나간다. <그림 1>는 <표 1>에 제시된 태스크 집합 $T = \{T_1, T_2, T_3\}$ 에 대한 스케줄 과정 중에 발생된 예약 리스트의 모습이다.

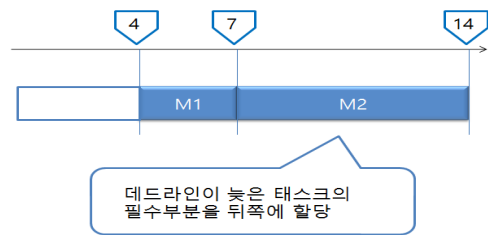


그림 1. 예약리스트
Figure 1. Reservation List

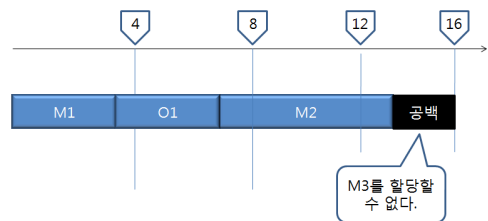


그림 2. 최종 스케줄링 결과
Figure 2. Final Scheduling Result

업데이트된 예약리스트를 이용한 스케줄 결과는 <그림 2>와 같다. NORA 알고리즘은 수용된 태스크 집합에 대하여 최소의 총 오류를 산출한다.

그러나 새로운 태스크 T_3 가 도착하면 위 <그림 2>처럼 필수부분을 할당 할 수 없어 스케줄 성공률이 낮아진다. 만일 태스크 T_3 가 도착되지 않았다면 NORA 알고리즘은 수용된 태스크 집합에서 최소의 총 오류를 산출하게 된다.

NORA 알고리즘의 의사코드는 다음과 같이 구성된다.

Step1 : 이벤트 없을 때, 태스크 큐의 맨 앞 부분에 있는 태스크를 CPU 처리에 할당

Event_1 : 현재 태스크가 정상 종료 되거나, 만기로 조기 종료 되었을 때

Event_2 : 예약리스트의 처음 위치에 도달될 때

Event_3 : 새로운 태스크가 도착할 때

Event_1에서 수행되는 처리 내용은,

처리_1 : 태스크 큐로부터 현재 태스크를 제거함

처리_2 : 현재 태스크의 예약을 취소함

처리_3 : Step1 으로 감

Event_2에서 수행되는 처리 내용은,

만일, 현재 태스크를 위해 할당된 시간이 있다면 그 시간을 현재 태스크에 할당하고 현재 태스크의 예약을 취소한다.

그렇지 않으면, 현재 태스크를 종료하고 태스크 큐로부터 지운다.

Event_3에서 수행되는 처리 내용은,

처리_1 : 현재 태스크의 예약 리스트를 갱신함

처리_2 : 새로운 태스크에 대한 예약을 만듦

처리_3 : 태스크 큐에 새로운 태스크를 삽입

처리_4 : Step1 으로 감

2.3 DOP 알고리즘

DOP 알고리즘은 EDF (Earliest Deadline First) 기반의 실시간 스케줄링 알고리즘으로 SMF (Schedule Mandatory First) 방법을 사용하여 NORA 알고리즘의 단점을 해소하는 방법을 제시한다.

DOP 알고리즘은 스케줄 가능한 선택부분들의 우선순위를 최대한 낮게 조정 한다. 그리하여 새로운 태스크가 발생한 경우 지연시킨 선택 부분들을 포기하면서 새로운 태스크의 필수 부분을 처리한다. 이 작업을 위하여 수용 태스크 [7]를 수행한다. 일반적으로 수용 태스크는 새로운 태스크가 도착될 때 이 새로운 태스크의 필수적부분이 스케줄 가능한지를 테스트 하는 것 인데, 그 테스트 결과 시스템에 수용 되어질 것 인지, 아니면 거절(reject) 될 것 인지를 판별하게 된다.

만약, 이 새로운 태스크가 테스트 결과 시스템에 수용된다면 기존의 아직 종료되지 못한 태스크들과 함께 만기가 짧은 순서로 각 태스크들의 필수적 부분들을 먼저 처리하고 남은 여유시간이 있다면 만기이내에 스케줄 가능한 선택적 부분들을 만기가 짧은 순서대로 각 태스크의 만기내에서 스케줄 시킨다. 결국, SMF (Schedule Mandatory First) 방법으로 스케줄링 하게 되는 것이며 SMF 방법과 차이가 없어 보인다.

3. 기존의 선택적 태스크 지연 방법

비정밀 태스크 계산 모델에서는 모든 태스크들을 만기 내에 스케줄 시키는 것이 불가능 하므로 최대한 많은 필수부분들과 선택부분들을 처리 하는 것이 관건인데 NORA 알고리즘은 동적으로 도착하는 온라인 태스크들의 필수 부분들을 처리하지 못 할 경우가 종종 발생하고, DOP 알고리즘이나 SMF (Schedule Mandatory First) 알고리즘은

만기가 빠른 태스크들의 선택부분들을 최대한 지연시키는 관계로 그러한 선택부분들을 처리하지 못하여 총 오류가 많아지는 단점이 있다. DOP 알고리즘은 가급적이면 선택부분들의 우선순위를 낮게 조정 한다. 그리하여 새로운 태스크가 발생한 경우 지연된 선택 실행부분들을 포기하면서 새로운 태스크의 필수 실행부분을 먼저 처리 하게 한다. DOP 알고리즘은 레드 큐를 사용하는데 이 곳에 삽입되어 있는 태스크를 실행하게 된다. 레드 큐는 만기를 기준으로 만기가 빠른 태스크들을 앞부분에 배치한다. 만일 새로운 태스크의 만기가 현재 실행 중인 태스크의 만기보다 빠르면 새로운 태스크가 현재 실행 중인 태스크를 선점하여 스케줄 된다. DOP 알고리즘 에서는 수용 테스트(acceptance test) 기능이 포함 되어 있어 새로운 태스크에 대하여 레드 큐에 들어 갈 수 있는지 아니면 포기 될 지를 판단하는 과정을 수행 하게 된다.

DOP 알고리즘 에서는 비정밀 태스크들의 선택부분들의 수행 순서를 최대한 낮게 조정하는데 차후에 도착 할 수 있는 새로운 태스크에 유연하게 대처하기 위해 필수부분은 앞에서, 선택부분은 뒤에서 스케줄 한다. 이 알고리즘은 최대로 지연 가능한 선택부분들을 우선적으로 스케줄 하는 방법을 사용한다. 결과적으로 만기가 너무 늦어서 포기 될 수 있었던 선택부분들의 수행시간들을 늘일 수 있고 새로운 필수 실행 부분들의 스케줄 가능성 또한 높일 수 있다. DOP 알고리즘은 태스크의 선택부분들의 수행순서들을 조절하는 것으로서 필수부분은 앞에, 선택부분은 뒤에 배치하는 전략을 사용한다. 이 점에 있어서 SMF 알고리즘과 매우 유사하다.

4. 선택적 태스크들을 지연시키는 새로운 방법

4.1 DMO 알고리즘

SMF (Schedule Mandatory First) 방법에 기반한 DOP 알고리즘은 비정밀 태스크들의 필수 부분들을 우선적으로 먼저 실행하기 때문에 NORA 알고리즘에 비하여 필수적부분들의 스케줄링 성공률(Schedulability)이 현저히 높다. 그러나 선택부분들을 최대한 지연시키는 관계로 실행 가능 할 수 있었던 만기가 짧은 선택부분들의 만기를 놓쳐 실행하지 못함으로써 총 오류가 많아지는 단점이 발생하였다. 또한 SMF (Schedule Mandatory First)알고리즘과 그 방법에 있어서 매우 유사하며 만기 순서 위반 현상이 발생된다.

한편, NORA 알고리즘은 임의의 시점에서 스케줄 대상이 되는 비정밀 태스크들의 필수적부분들이 반드시 스케줄 가능하다는 FMC (feasible Mandatory Constraint) 제약조건을 만족하는 한 최소의 총 오류를 산출하지만 차후 도착되는 온라인 태스크들의 필수적부분들을 융통성있게 스케줄시키지 못하여 필수적부분들의 스케줄 가능성이 낮아지게 된다.

이에 본 논문에서는 비정밀 태스크들의 선택적부분들을 만기가 빠른 순서로 어느 특정 태스크까지만 지연시키어 비정밀 태스크들의 스케줄링시 만기 순서를 철저히 준수하면서 필수적부분들의 스케줄가능성 면에서 NORA 알고리즘 보다 월등히 우수하고 만기 순서를 지키지 못하는 DOP 알고리즘과는 필수적부분들의 스케줄가능성과 총 오류 면에서 유사한 성능을 보이는 새로운 DMO 알고리즘을 제시하였다.

먼저 본 알고리즘의 중요한 특성인 만기 순서 준수를 설명 하기 위해 태스크 집합을 <표 2> 처럼 설정해 보았다.

표 2. 태스크 집합
Table 2. Task Sets

	R_i	M_i	O_i	D_i
T_1	0	4	3	7
T_2	0	2	1	12
T_3	0	3	1	13
T_4	0	2	3	16
T_5	0	3	4	18

<표 2>의 태스크 집합을 SMF 알고리즘이나 DOP 알고리즘으로 스케줄링하면 다음과 같은 순서로 될 것이다.

$$M_1M_2M_3M_4M_5O_4(2)O_5(2)$$

위의 스케줄링 결과를 살펴보면 태스크들의 스케줄 순서가 < 1-2-3-4-5-4-5 > 처럼 만기가 가장 늦은 T_5 가 스케줄 된 후 이 보다 만기가 짧은 T_4 가 스케줄 됨을 볼 수 있는데 이러한 현상을 만기 순서 위반 (Deadline Order Violation) 이라 한다. 한편 제안된 DMO 알고리즘을 적용한 결과는 아래처럼 태스크 스케줄 순서가 철저히 만기를 준수하고 있음을 볼 수 있다. 이 부분이 바로 제안된 알고리즘과 DOP 알고리즘이 차별화 되는 본 알고리즘의 중요한 특징이다.

$$M_1M_2M_3O_3M_4O_4M_5$$

본 알고리즘의 구성은 다음의 <그림 3>과 같다. DMO_Schedule()에서는 임의의 시점에서 스케줄 대상이 되는 태스크들을 대상으로 하여 만기가 짧은 태스크 순으로 각 태스크의 필수부분과 선택부

분 실행요구시간의 합이 그 태스크의 만기 보다 작거나 같은지 판단하는 과정을 반복하여 이 조건이 만족되지 않는 첫 번째 태스크의 인덱스 (OSI) 를 구한다. 그런 다음, 이 태스크 인덱스 보다 작은 인덱스를 갖는 태스크들의 선택적 부분들은 무조건 지연시키며 이 태스크 인덱스 보다 같거나 큰 인덱스를 갖는 태스크들의 선택적부분들을 그들 각각의 만기 이전 까지만 스케줄 시킨다.

```

void DMO_Schedule(ith, InterStart, InterEnd)
{
    OSI = Find_OSI (InterStart, InterEnd);
    Length = InterEnd - InterStart;
    i =1
    While(i <= NbtPrevStep)
    {
        Task = ListTask[i];
        Perform_Mandatory_Scheduling;
        if (i >= OSI && TaskSystem[Task].Oi > 0)
        {
            Task = ListTask[i];
            Perform_Optional_Scheduling;
        }
        i++;
    }

    int Find_OSI (InterStart, InterEnd)
    {
        ST = InterStart;
        for (i=1; i<=NbtPrevStep; i++)
        {
            Task = ListTask[i];
            SST = ST + TaskSystem[Task].Mi
                    + TaskSystem[Task].Oi;
            if (SST > TaskSystem[Task].Di) return (i);
            else ST = SST;
        }
    }
}
    
```

그림 3. 제안된 DMO 알고리즘
Figure 3. The Proposed DMO Algorithm

4.2 DMO 알고리즘의 예

DMO 알고리즘의 예를 들면, 태스크 만기의 오름차순으로 정렬되어 있는 표 2의 태스크 집합에서 T_1 의 필수부분(4)과 선택부분(3) 실행요구시간의 합(7)이 이 태스크의 만기(7) 보다 작거나 같으므로 다음으로 만기가 큰 T_2 의 필수부분(2)과 선

택부분(1) 실행요구시간의 합(3)을 이전 태스크들의 실행요구시간들의 합(7)과 더하여 그 결과 값(10)을 T_2 의 만기(12)와 다시 비교한다. 이번에도 이 태스크의 만기 보다 작거나 같으므로 다음으로 만기가 큰 T_3 의 필수부분(3)과 선택부분(1) 실행요구시간의 합(4)을 이전 태스크들의 실행요구시간들의 합(10)과 더하여 그 결과 값(14)을 T_3 의 만기(13)와 다시 비교한다.

이제, T_3 의 만기(13)가 T_3 보다 만기가 작거나 같은 태스크들의 실행요구시간들의 합(14) 보다 작으므로 이 조건이 만족되지 않는 첫 번째 태스크의 인덱스 (OSI) 3을 구할 수 있게 된다. 그런 다음, 이 태스크 인덱스 보다 작은 인덱스를 갖는 태스크들(T_1 과 T_2)의 선택적 부분들은 무조건 지연시키며 이 태스크 인덱스 3 보다 같거나 큰 인덱스를 갖는 태스크들(T_3, T_4, T_5)의 선택적부분들을 그들 각 각의 만기 이전 까지만 스케줄 시킨다.

4.3 실험결과

본 실험에서는 비정밀 태스크 $T_i = \{R_i, D_i, M_i, O_i\}$ 를 구성하는 각 각의 인자들인 R_i, D_i, M_i, O_i 값 들을 임의의 난수들로 랜덤하게 생성하여 태스크 T_i 를 정의하고 이러한 태스크 100개로 구성된 태스크 집합 100개를 생성하여 제안된 DMO 알고리즘을 기존의 NORA 알고리즘 및 DOP 알고리즘과 필수적부분들에 대한 스케줄 가능성과 총 오류 면에서 비교해 보았다.

다음의 <표 3>은 각 알고리즘별로 100개의 태스크 집합들 중 스케줄 가능한 태스크 집합의 개수(왼쪽)와 총 오류(오른쪽)를 비교한 것이다. 아래의 <그림 4>, <그림 5>와 같은 실험분석에 의하면 DMO 알고리즘은 스케줄링 성공률(Schedulability)과

총 오류 면에서 거의 DOP 알고리즘과 비슷함을 나타내었다.

표 3. 스케줄 가능성(왼쪽)과 총오류(오른쪽)
Table 3. Schedulability(left) and Total Error(right)

No.	SMF, DOP	EDF, NORA	DMO	No.	SMF, DOP	EDF, NORA	DMO
1	99	0	91	1	277	243	282
2	98	0	95	2	259	214	269
3	99	0	95	3	280	248	295
4	99	0	95	4	272	225	278
5	99	0	93	5	219	185	225
6	96	0	84	6	247	214	252
7	100	0	91	7	208	177	220
8	98	0	90	8	232	203	235
9	99	1	93	9	277	231	277
10	100	0	96	10	271	231	273

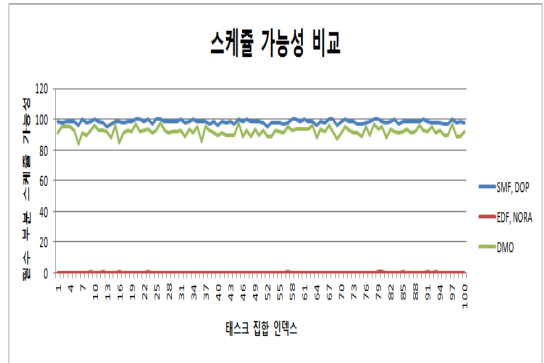


그림 4. 스케줄 가능성의 실험결과
Figure 4. Experimental Result of the Schedulability

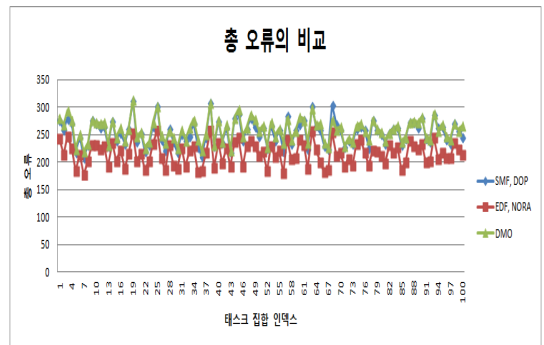


그림 5. 총 오류(Total Error)의 실험결과
Figure 5. Experimental Result of the Total Error

그러나 DMO 알고리즘의 특징은 만기 순서 위반(Deadline Order Violation)이 발생하지 않으면서도 DOP 알고리즘과 비슷한 스케줄링 성공률(Schedulability)과 총 오류(total error)를 나타내었다는데 큰 의미가 있다고 본다. 한편, EDF나 NORA 알고리즘에 비하여 제안된 DMO 알고리즘이 스케줄링 성공률(Schedulability)이 대단히 높다. 그 이유는 만기가 짧은 태스크들의 선택부분들을 만기를 기준으로 특정 태스크까지만 지연시키기 때문이다. 여기서, EDF나 NORA 알고리즘은 선택부분들을 지연시키지 않으며 필수부분이든 선택부분이든 만기를 기준으로 만기가 짧은 태스크 순서로 처리 하는 것을 원칙으로 한다.

5. 결 론

본 논문에서는 비정밀 계산모델에서 선택적부분들을 지연시키는 새로운 알고리즘 (DMO)을 제안하였다. 기존의 선택적부분 지연 알고리즘인 DOP 알고리즘은 비정밀 태스크들의 필수적부분들을 먼저 처리하고 시간적 여유가 있을 때 선택적 부분들을 그 들 각 각의 만기 이내에 처리하는 SMF(Schedule Mandatory First) 방법을 적용하다 보니 필수적부분들의 스케줄가능성은 EDF나 NORA 알고리즘 보다 현격히 높아지나 만기가 짧은 비정밀 태스크들의 선택적부분들을 최대한 지연시키는 관계로 이들이 그 들 각 각의 만기를 놓치어 스케줄 되지 못함으로써 총 오류가 증가되는 문제가 발생했으며 선택적부분들을 최대한 지연시켜 맨 나중에 스케줄함으로서 만기 순서 위반 현상이 초래되었다.

그래서 본 논문에서는 비정밀 태스크들의 선택적부분들을 최대한으로 지연시키지 않고 특정 한계를 계산하여 그 한계까지만 지연시키는 새로운 지연 전략을 적용한 DMO 알고리즘을 제안하였다. 실험 결과 DMO 알고리즘은 만기 순서를 철저히 준

수 하면서도 필수적부분들의 스케줄 가능성 면에서 EDF나 NORA 알고리즘에 비하여 월등히 우수하며 만기 순서 위반(Deadline Order Violation)이 발생하는 DOP 알고리즘 과도 비슷한 성능을 보였다. 총 오류 측면에서는 EDF나 NORA 알고리즘 보다는 지연되는 선택적부분들의 실행시간의 합계에 비례하여 총 오류가 증가하는 단점은 있으나 이는 필수적부분들의 스케줄가능성을 높이기 위하여 부득이한 것이며 DOP 알고리즘 과는 비슷한 총 오류 성능을 보였다. 그러나 DOP 알고리즘에서는 만기 순서 위반이 발생하는 반면 본 DMO 알고리즘은 철저하게 비정밀 태스크들의 만기들을 준수 한다는데 의미가 크다.

비정밀 계산 모델에서 선택적 부분들을 특정 한계까지만 지연시키어 필수적부분들의 스케줄가능성을 높이며 비정밀 태스크들의 만기 순서를 준수 하는 새로운 알고리즘인 DMO는 실시간 운영체제의 구현시 스케줄링 오버헤드로 인한 만기 위반을 줄일 경우나 고객 요구식 비디오(VOD)와 같은 비디오 분배 시스템을 취급하는 유선 TV 분야, 그리고 고객이 요구하는 마감일(due dates)에 맞추어 생산하는 생산시스템 등에 응용될 수 있으리라 사료된다.

향후, 비정밀 태스크들의 온라인 스케줄링시 필수적부분들의 스케줄링가능성과 총 오류 라는 두 가지 핵심적인 성능들을 모두 최적으로 충족할 수 있는 더욱 합리적인 알고리즘들에 대한 연구가 기대된다.

References

- [1] H. P. Choi, and Y. S. Kim, *An EDF based real-time scheduling algorithm for imprecise computation*, Korea Information Processing Society(A), Vol. 18-A, No. 4, pp. 143-148, 2011.

- [2] W. K. Shih, and J. W. S. Liu, *On-line scheduling of imprecise computations to minimize error*, SIAM Journal on Computing, Vol. 25, No. 5, pp. 1105-1121, 1996.
- [3] W. K. Shih, C. R. Lee, and C. H. Tang, *A fast algorithm for scheduling imprecise computations with timing constraints to minimize weighted error*, in Proceedings of the 21st IEEE Real-Time Systems Symposium, pp. 305-310, 2000.
- [4] W. K. Shih, J. W. S. Liu, and J. Y. Chung, *Fast algorithms for scheduling imprecise computations*, SIAM Journal on Computing, Vol. 20, pp. 537-552, 1991.
- [5] J. M. Chen, W. C. Lu, W. K. Shih, and M. C. Tang, *Imprecise computations with deferred optional tasks*, Journal of Information Science and Engineering, Vol. 25, pp. 185-200, 2009.
- [6] <http://tinyos.net/tinyos-2.x/doc/pdf/tep106.pdf>
- [7] Hard Real-Time Computing Systems <http://download.springer.com>
- [8] J. W. S. Liu, K. J. Lin, W. K. Shih, A. C. S. Yu, J. Y. Chung, and W. Zhao, *Algorithms for scheduling imprecise computations*, Comput. Magazine, 24, pp. 58-68, 1991.
- [9] J. H. Kim, K. H. Song, K. H. Choi, and G. H. Jung, *Performance evaluation of on-line scheduling algorithms for imprecise computation*, Proceedings Fifth International Conference on Real-Time Computing Systems and Applications, 1998.
- [10] J. W. S. Liu, W. K. Shih, K. J. Lin, R. Bettati, and J. Y. Chung, *Imprecise computations*. Proceedings of The IEEE, Vol. 82, No. 1, pp. 83-94, 1994.

비정밀 실시간 스케줄링에서 선택적 태스크들의 지연 방법에 관한 연구

송기현, 전근형

대전보건대학교 경영정보과

요약

비정밀 계산 기법은 모든 긴급한 태스크들이 그 태스크들의 계산시간 요구조건들과 그 결과들의 질을 서로 상호 보완함으로써 그들의 만기들 이전에 그들의 결과들을 산출하는 것을 보장한다. NORA 알고리즘은 비정밀 온라인 실시간 태스크 집합에서 최소의 총 오류를 산출하지만 만기가 빠른 태스크의 선택실행부분이 만기가 늦은 태스크의 필수실행부분보다 먼저 실행됨으로서 모든 태스크들의 필수실행부분들에 대한 스케줄가능성이 낮아지는 문제가 발생한다. 이 문제를 개선하기 위하여 만기가 빠른 태스크들의 선택실행부분들을 지연시키는 DOP 알고리즘이 제안되었지만 이는 기존의 SMF 알고리즘과 차이가 없어 보인다. 그래서 본 논문은 만기가 빠른 태스크들의 선택실행부분들을 지연시키는 새로운 방법을 제안하고 기존의 NORA 및 EDF, DOP(SMF) 방법과 스케줄가능성과 총 오류 면에서 비교 분석하여 본 알고리즘의 우수성을 나타내고자 한다.

감사의 글

본 논문은 대전보건대학교의 2013학년도 학술연구조성비를 지원 받음.



Gi-Hyeon Song received the bachelor's degree and M.S. degree in the Department of Computer Sciences and Statistics from the Chungnam National University in 1985 and 1987 respectively. He

received the Ph.D. degree in the Department of Computer Engineering from Ajou University in 1999. He has been a professor in the Department of Management Information System at Daejeon Health Sciences College since 1990. His current research interests include real-time scheduling, embedded system, radar tracking. He is a member of the KKITS.

E-mail address: ghsong@hit.ac.kr



Keun-Hyeong Jeon received the bachelor's degree in the Department of Electronic Engineering from the Chungnam National University in 1987. He received the M.S. degree in the Chungnam National University in 1989. His current research interests include computer science. He is a member of the KKITS

E-mail address: arepairman@naver.com