삽입 실시간 디스크 스케줄링기법과 양방향 SCAN기법

이덕용° 박창현 조행래
영남 대학교 대학원 컴퓨터공학과
kkiddy°@aiiis.yu.ac.kr, { pack, hrcho}@yu.ac.kr

Insertion Real-Time Disk Scheduling Scheme and A Both Direction SCAN Algorithms

Duckyong Lee° ChangHyeon Park Haengrae Cho
Dept. of Computer Engineering, Yeungnam University

요 약

실시간 스케줄링에서 시간당 처리량을 높이기 위해서 EDF에 SCAN기법을 추가하는 많은 방법이 연구되었다. 하지만 기존 기법들은 SCAN그룹을 생성할 때, 연속된 태스크들만 SCAN그룹의 포함 대상으로 고려하기 때문에 많은 제한이 따른다. 또한 SCAN기법은 처리방향이 고정되었기 때문에 시간적 손실이 많은 단점을 가진다. 본 연구에서는 연속되지 않은 태스크들을 SCAN그룹의 포함 대상으로 고려 할수 있는 태스크 삽입기법과, 기존의 SCAN그룹에서 합병하지 못하는 SCAN그룹들을 합병할 수 있는 SCAN합병기법, 마지막으로 SCAN그룹을 처리하는데 시간적 이점을 얻을 수 있는 양 방향 SCAN기법을 제시한다.

1. 서 론

디스크 스케줄링이란 디스크 입출력 대기 중인 요구의 처리순서를 결정하는 기법으로써 목적은 디스크 시스템의 성능향상에 있다. 디스크 스케줄링기법의 성능은 크게 3가지로 평가 할 수 있다. 첫째, 단위 시간당 얼마나 많은 디스크 입출력 요구를 서비스하는가를 평가하는 단위 시간당 처리량(Throughput)과, 둘째, 각 디스크 입출력 요구에 대해 얼마나 빠른 시간 내에 서비스하는가를 측정하는 평균응답시간(Response Time), 셋째, 예측성 판단을 위한 요소로서 응답시간의 분산(Variance)에 사용되는 응답시간의 예측성(Predictability)등을 사용해서 디스크 스케줄링을 평가 할 수 있다. 하지만 위와 같은 일반적인 디스크 스케줄링 평가 기준만 가지고는 실시간 태스크를 처리하는 실시간 디스크 스케줄링을 제대로 평가 할 수 없다.

실시간 태스크란 시간 제약성에 따라 마감시간 미스가 전체 시스템 성능에 치명적인 영향을 미치는 하드 실시간(Hard Real-Time) 태스크와 시스템 성능 저하가 완만한 소프트 실시간(Soft Real-Time) 태스크로 나눌 수 있다. 이와 같은 태스크를 서비스하기 위해서 실시간 디스크 스케줄링기법이 제공된다.

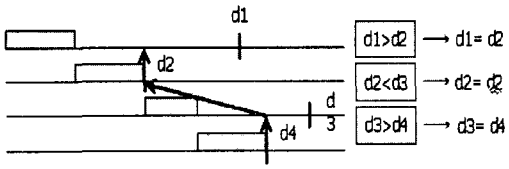
실시간 디스크 스케줄링이란 일반적인 디스크 스케줄링의 평가 기준에 마감시간이라는 중요한 인자를 포함한다[1-2]. 기존의 실시간 디스크 스케줄링에는 EDF와 EDF에 SCAN기법을 추가한 SCAN-EDF, SCAN그룹을 확장시킨 MSG를 사용한 DMS기법이 제안되었다.

2. 관련 연구

실시간 디스크 스케줄링은 광범위하게 연구되고 있고, 그것에 관한 많은 연구 결과가 발표 되었다. 가장 일반적인 실시간 디스크 스케줄링 기법으로는 EDF(Earliest Deadline First) 스케줄링기법이 있다[1]. EDF란 마감시간에 가장 가까운 태스크를 먼저 처리해 주는 기법을 말한다. EDF는 태스크를 처리하고, 다음 태스크로 헤드가 이동하는 시간을 최악으로 고려하기 때문에 시간당 처리량이 매우 떨어지는 단점을 가진다[2].

SCAN-EDF는 EDF 스케줄링 기법에서 비슷한 마감시간을 가진 태스크들을 SCAN방식으로 처리하여 시간당 처리량을 높였다[2]. 하지만 실제로 비슷한 마감시간을 가진 태스크들이 연속해서 들어오는 경우는 극히 드물고, 다른 마감시간을 가진 태스크들이 디스크를 요청하는 경우가 대부분이기 때문에 EDF와 비슷한 성능을 보인다[3].

DMS(deadline Modification SCAN)는 확장 SCAN그룹의 결정기법으로 MSG(Maximum Scannable Group)을 제안한다[3]. MSG의 개념은 태스크들의 마감시간을 어기지 않고 SCAN기법으로 다시 스케줄링 될 수 있는 연속된 태스크들의 범위를 말한다[3]. MSG 기법으로 다시 스케줄링 한 결과는 EDF의 순서를 무시하기 때문에 DMS는 마감시간을 수정해서 EDF순서를 유지 하면, 또한 EDF에서 수락할 수 없었던 태스크를 수락할 수 있는 가능성을 가질 수 있는 장점이 있다[3-4](그림 1). 하지만 MSG는 연속된 태스크만 SCAN그룹에 포함 할 수 있기 때문에 많은 제약을 가진다. 뿐만 아니라 앞에서 제안한 모든 SCAN기법은 한 방향만 지원하기 때문에 많은 시간적 손실이 생길 수 있다.

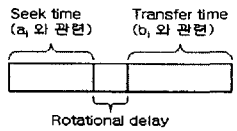


(그림 1) DMS의 Deadline Modification 기법

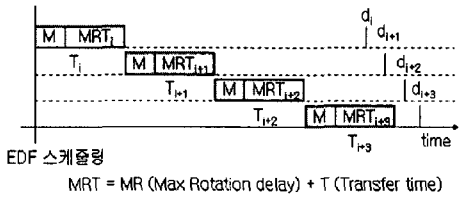
3. 본론

3.1. 문제 정의

디스크 요구 태스크의 디스크 접근시간은 탐색시간(Seek time), 회전 지연(Rotational delay), 전송시간(Transfer time)의 합으로 구성된다[5]. 실시간 태스크 $T_i\{r_i, d_i, a_i, b_i\}$에서 a_i는 탐색시간과, b_i는 전송시간과 관련이 있다[5-7]. 이를 그림으로 나타내면 (그림 2)과 같이 표현할 수 있다. 최악의 탐색시간을 M, 최악의 회전지연 시간을 MR, 전송시간을 T, MR과 T의합을 MRT로 정의 한다. 이를 바탕으로 EDF기법로 스케줄링한 결과는 (그림 3)과 같다($T_i \cdots T_{i+3}$).



(그림 2) 디스크 접근시간



(그림 3) EDF로 스케줄링한 결과

여기서 MRT의 T(Transfer time)부분은 줄일 수 없고, 만약 MR(Max Rotation delay)부분을 줄인다 하더라도 디스크 스케줄링의 성능 개선을 크게 기대할 수 없다. 결과적으로 M의 크기 및 개수를 줄임으로써 시간당 처리량을 높이고, 궁극적으로 실행 가능한 태스크의 수를 늘일 수 있다.

3.2. 가정

본 연구에서 사용할 SCAN그룹은 DMS에서 제안한 MSG를 사용한다[3]. 각각의 실시간 태스크가 독립적이고, MSG의 i번째 SCAN그룹 G_i가 태스크($T_i \cdots T_n$)를 포함하고, $i \leq n$ 일 때, 다음과 같은 가정들이 성립된다.

가정 1) SCAN그룹 G_i의 스케줄링 가능한 탐색 시간은 M + [$\alpha \times (n-i)$] 보다 작거나, 최대 같음. (α는 헤드를 멈추는 시간 + 헤드를 가속하는 시간)

가정 2) SCAN의 방향을 조절해도 M의 크기는 변화 없음.

가정 3) 연속된 SCAN그룹 G_i와 G_{i+1}사이의 스케줄링 가능한 탐색시간은 M보다 작거나 최대 같음.

증명 1 : MSG의 i번째 SCAN그룹 $G_i(T_i \cdots T_n)$에서 헤드의 이동경로 시간은 i번째 트랙에서 n번째 트랙으로 이동하는 시간에 각각의 태스크를 처리하기 위해 헤드가 트랙에 멈추는 시간과 다시 가속하는 시간의 합이다. M은 최악의 탐색 시간이기 때문에, G_j안의 태스크들을 SCAN방식으로 정렬하면 ($T_k \cdots T_m$)의 순서가 된다. 이때 $m-k=n-i$를 만족한다(수식 1). 즉 하나의 SCAN그룹 스케줄링 가능한 최대 탐색시간은 M가 같다. 하지만 실제로는 α(헤드를 멈추는 시간과 헤드를 가속하는 시간)이 각각의 태스크를 처리할 때 마다 붙는다(수식 2).

$$M \geq \sum_{k}^{m-1}(c_{i,i+1}) \qquad \text{(수식 1)}$$

$$M + [\alpha \times (m-k)] \geq \sum_{k}^{m-1}(c_{i,i+1}) \qquad \text{(수식 2)}$$

증명 2 : SCAN방식으로 정렬한 $G_j(T_k \cdots T_m)$를 k부터 m으로 처리하거나, m부터 k로 처리하던지 M의 크기는 변화 없다.

증명 3 : 연속된 G_i, G_{i+1}를 스케줄링 하기 위해서는 G_i의 마지막 태스크에서 G_{i+1}의 처음 태스크 까지 디스크의 헤드가 이동해야 한다. 최악의 탐색시간이 M이기 때문에 이 시간은 M보다 작거나 최대 같다.

3.3. 제안 알고리즘

제안 알고리즘은 M의 개수를 줄이는 기법을 제안한다. 제안 알고리즘은 처리 순서는 다음과 같다.

1) EDF 수락제어
2) 요구된 태스크를 M과 b, a를 이용하여 인덱스를 만듦.
3) EDF로 스케줄링함.
4) SCAN그룹(G)를 결정.
5) 인덱스를 이용해서 태스크를 G에 대기 태스크 삽입.
6) G를 SCAN순서를 스케줄링함.
7) G에 속한 모든 태스크들의 마감시간 d를 G에 마감시간으로 수정.
8) SCAN그룹 합병.
9) SCAN 방향 결정.

(그림 4) 처리 순서

1) EDF수락 제어 : 실시간 태스크가 요청이 되면 현재까지 수락되었지만 스케줄링되지 않은 태스크들과 함께 EDF수락제어를 실시하여, 수용할지 거절할지 결정한다[1].

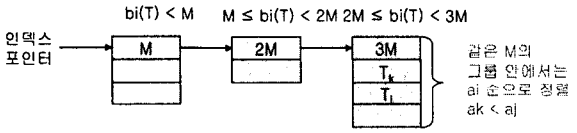
2) 인덱스 : EDF수락제어를 만족하는 태스크의 집합

$\{T_0 \cdots T_n\}$가 있으면, 각각의 태스크(T_i)의 b_i를 M의 배수를 기준으로 인덱스를 생성한다. 같은 M의 배수 인덱스 안에서는 a_i를 기준으로 정렬한다(그림 5).

3) EDF스케줄링 : 1)에서 EDF수락제어를 만족한 태스크들을 EDF로 스케줄링 한다.

4) SCAN그룹 결정 : MSG기법을 이용하여 SCAN그룹을 결정한다(3.2 SCAN그룹).

5) 태스크 삽입 : MSG는 연속된 태스크들만으로 SCAN그룹을 결정하기 때문에 많은 제약이 따른다. 불연속 된 태스크들이라도 SCAN그룹안의 태스크들의 마감시간을 어기지 않는 범위 안에서 삽입이 가능하다. SCAN그룹안의 탐색시간은 최대 M + [$\alpha\times$ (SCAN그룹에 참여하는 태스크의 수)]가 된다(가정 1). SCAN그룹에 참여하는 태스크의 개수를 n이라고 하면, 실제 현재 SCAN그룹의 탐색시간은 $n \times M$이다. 즉 $(n\times M)-(M+\alpha\times n)$의 시간적여유가 생긴다. 2)에서 만든 인덱스를 이용해서 삽입할 태스크를 선택한다. 선택방법은 삽입할 수 있는 가장 큰 M의 배수 태스크를 먼저 삽입한다. 만약 같은 M의 배수 태스크가 여러 개 존재 한다면, SCAN그룹 안에 태스크의 분포를 고르게 하기 위해서, SCAN그룹의 인덱스를 보고 홀수 번째는 a_i가른 쪽 짝수 번째는 a_i 가작은 쪽 태스크를 먼저 삽입한다. 이와 같은 방법으로 SCAN그룹을 스케줄링하기 전에 삽입할 태스크를 결정할 수 있다.



(그림 5) 인덱스

6) SCAN 스케줄링 : 4)에서 정한 SCAN그룹에 5)에서 삽입한 태스크를 합한 SCAN그룹을 SCAN기법으로 스케줄링 한다.

7) 마감시간 수정 : SCAN그룹안의 태스크들의 d(마감시간)은 모두 SCAN그룹의 마감시간보다 크기 때문에, 모든 태스크의 d를 SCAN그룹의 마감시간으로 수정한다.

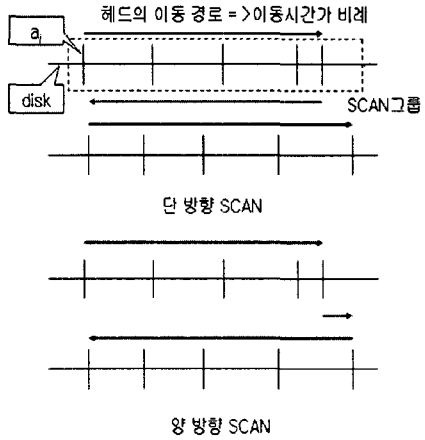
8) SCAN그룹 합병 : MSG와 삽입기법을 사용해서 생성된 여러 개의 SCAN그룹들이 다음조건을 만족할 때 합병이 된다. 즉 G_i의 마지막 태스크(T_j)와 G_{j+1}의 처음 태스크(T_k)를 비교해서 $a_j < a_k$ 성립하면 합병될 수 있다(그림 6). SCAN그룹 합병 기법은 각각의 SCAN그룹들이 합병되기 전에 이미 SCAN 순서를 유지하기 때문에 합병하고 재 스케줄링 할 필요가 없다.

9) SCAN방향 결정 : 앞의 가정 2)에서 SCAN의 방향을 조절해도 M의 크기는 변화가 없다는 것을 증명하였다. 일반적으로 잘 분포된 SCAN그룹들의 사이에서 양방향으로 SCAN을 실시하면, 단 방향 SCAN보다 헤드의 이동시간을 줄일 수 있다 (그림 7). SCAN그룹은 처음부터 시작해서 순방향(a_i 오름차순) SCAN과, 역방향(a_i의 내림차순)으로 번갈아 가면서 SCAN을 실시한다. 6)에서 이미 순방향으로 SCAN스케줄링을 했기 때문에 역방향은 뒤에서부터 처리를 한다. 만약 중간에 8)에서 제안한 합병된 SCAN그룹이 있으면 합병된 SCAN그룹은 순방향으로 처리한고, 다음에 이어지는 SCAN그룹을 역방향으로 처리한다.

```
i = 1;
while( MSG_{i+1} 존재 할 때까지)
{
        T_j = MSG_i 마지막 태스크;
        T_k = MSG_{i+1} 처음 태스크;
        if( a_j < a_k)
                SCAN merge;
        i++;

}
```

(그림 6) SCAN그룹 합병



(그림 7) SCAN그룹 사이의 디스크 이동시간

4. 참고 문헌

[1] B. Kao and R. Cheng, " Disk Scheduling,"In REAL-TIME DATABASE SYSTEMS, pp. 97-107, Kluwer, 2002.

[2] A. L. N. Reddy and J. Wylie, "Disk Scheduling in a Multimedia I/O System," In Proc. of the first ACM International Conference on Multimedia, Pages 225-233, Anheim, CA, 1993.

[3] R. I. Chang, W. K. Shin and R. C. Chang, " Deadline-Modification-SCAN with Maximum-Scannable-Groups for Multimedia Real-Time Disk Scheduling," Proc. IEEE RTSS, pp. 40-49, 1998.

[4] R.I. Chang, W.K. Shin and R.C. Chang, "Real-time disk scheduling for multimedia applications with deadline-modification-scan scheme," accepted by Real-Time Systems: The International Journal of Time-Critical Computing Systems, 1999. (An early version is published in IEEE Real-Time System Symposium, 1998).

[5] J. A. Stankovic and G. C. Buttazzo, "Implication of classical scheduling results for real-time systems," IEEE Computer, pp. 16-25, 1995.

[6] C. Ruemmler and J. Wilkes, "An introduction to disk drive modeling," IEEE Computer, pp. 16-28, 1994.

[7] R. P. King, "Disk arm movement in anticipation of future requests," ACM Trans. Computer Systems, vol. 8, no. 3, pp. 214-229, 1990.