

구역분할 디스크를 사용하는 멀티미디어 서버에서 새로운 세션 시작에 따른 스케줄링 지연 현상의 최소화

(Mitigating the Side-effect of Starting New Session in Multimedia Streaming using Multi-zoned Disk)

조 경 선[†] 원 유 집^{**} 신 일 훈^{***} 고 건^{****}
(Kyeongsun Cho) (Youjip Won) (Ilhoon Shin) (Kern Koh)

요 약 디스크의 구역분할기술(zoning technology)은 디스크의 저장용량과 평균 전송 대역폭을 증가 시킴으로써, 디스크 서브시스템의 성능을 향상시켰다. 멀티미디어 시스템에서 구역분할 디스크의 성능을 충분히 이용하기 위하여 이중 버퍼링을 하는 SCAN 스케줄링을 사용한다. 하지만, 이 방식은 새로운 스트림의 요청 시에 지터(jitter)를 발생시키는 문제점이 있다. 본 논문에서는 이 문제를 해결하기 위한 **선행 버퍼링(pre-buffering)** 기법을 제안한다. 선행 버퍼링은 디스크 서브시스템의 수학적 모델을 통하여 스트림의 개수에 따른 주기시간과 각 주기시간에 필요한 데이터 양, 그리고 새로운 스트림 요청 시에 발생하는 데이터의 부족분을 예측하고, 예상되는 데이터의 부족분을 각 스트림의 서비스 전에 미리 버퍼링함으로써, 지터를 방지한다. 선행 버퍼링 기법은 멀티미디어 서버에 적용되어 사용자에게 지터 없는 고품질의 서비스를 제공하는 데 기여할 수 있다.

키워드 : 멀티미디어 서버, 디스크 스케줄링, 선행 버퍼링, 지터

Abstract Zoning technology of disk improved the performance of disk subsystem with increase of storage capacity and average transfer bandwidth. SCAN disk scheduling with double buffering is used to utilize the performance of zoned disk in multimedia system. However, this method has a problem that generates jitter when the number of streams increases. In this article, we propose the novel approach, *pre-buffering* policy, to overcome this problem. Pre-buffering avoids jitter by buffering the lack of data before starting service, which is estimated from the current cycle length and the maximum cycle length. We can calculate cycle length, data size needed in each cycle and the possible lack of data caused by the increase of the number of streams using the numerical model of disk subsystem. Pre-buffering can be applied for multimedia systems and contribute to provide clients with high quality service without jitter.

Key words : Multimedia Server, Disk Scheduling, Pre-buffering, Jitter

1. 서 론

컴퓨터 처리능력의 빠른 성장과 네트워크의 발전은

VOD 서비스와 같은 멀티미디어 서비스를 네트워크 상에서 지원하는 것을 가능하게 하였다. 멀티미디어 데이터는 주로 비디오와 오디오로 구성된다. 이들은 원래 연속적인 아날로그 신호이며 그 정보량이 상당히 크므로 디지털화를 수행할 때에 인간이 인지하는 범위내의 데이터만 디지털로 변환된다. 컴퓨터 내의 다른 데이터와 다르게 멀티미디어 데이터는 데이터의 순서와 데이터의 서비스 한계시간(service deadline)이 중요하다. 이런 이유로 멀티미디어는 연속적인 미디어(continuous media)로 불리기도 한다. 디지털 미디어 정보는 데이터 블록에 설정된 서비스 한계시간에 맞추어 디스플레이나 스피커로

· 본 연구는 2003~2006년 한국과학기술재단의 젊은과학자연구동지원연구(R08-2003-000-11104-0)지원을 받아 수행되었습니다.

† 비 회 원 : ㈜넷엔티비 연구소 선임연구원

goodsun@netntv.co.kr

** 종신회원 : 한양대학교 공과대학 전자전기컴퓨터 공학부 교수

yjwon@ece.hanyang.ac.kr

*** 비 회 원 : 서울대학교 전기컴퓨터 공학부

jeje@oslab.snu.ac.kr

**** 종신회원 : 서울대학교 전기컴퓨터 공학부 교수

kernkoh@oslab.snu.ac.kr

논문접수 : 2001년 10월 8일

심사완료 : 2004년 5월 13일

재생된다. 이것이 멀티미디어 시스템에서의 연속성 보장(continuity guarantee)이다. 만약 데이터 블록이 한계시간 내에 도착하지 않으면 자연스러운 상영에 문제가 생긴다. 데이터가 한계시간 내에 도착하지 않아, 화면이나 소리의 끊김이 발생하는 현상을 지터(jitter)라 한다.

한편, 멀티미디어 시스템(이는 일반적으로 네트워크를 이용한 멀티미디어 서비스를 위한 시스템을 말한다.)은 크게 서버와 클라이언트, 네트워크로 구성된다. 클라이언트에서 최종적으로 디스플레이 될 때 지터를 발생시키지 않기 위해서는 시스템을 구성하는 모든 구성요소에서 데이터의 서비스 한계시간을 지켜야 한다. 이를 위하여 시스템의 모든 구성요소의 데이터 처리능력이나 전송 대역폭이 미디어의 재생 대역폭(playing bandwidth)보다 크거나 같아야 한다. 따라서, 시스템이 제공할 수 있는 미디어의 최대 재생 대역폭은 모든 구성요소들이 가지는 전송 대역폭 중에서 가장 낮은 전송 대역폭에 의해 결정된다.

멀티미디어 시스템의 구성 요소 중 특히 서버의 디스크 서브시스템은 물리적인 구동장치를 포함하고 있어서 다른 구성요소보다 처리능력이나 전송 대역폭이 낮다. 즉, 디스크는 멀티미디어 시스템의 주 병목 구간이 되며, 동시에 멀티미디어 데이터의 서비스 한계시간을 보장하는 데 장애가 되는 주 요인이다. 본 논문은 디스크의 처리 능력을 최대로 활용하는 동시에 멀티미디어 데이터의 서비스 한계시간을 보장하는 기법 제안을 목표로 한다.

한편, 디스크는 그 기학적 특징으로 인하여 바깥쪽 실린더로 갈수록 물리적 저장면적이 증가한다. 구역분할기술 이전의 디스크는 모든 실린더가 같은 수의 섹터를 가지고 있었다. 즉 바깥쪽 실린더로 갈수록 낮은 저장밀도로 데이터를 저장하는 것이다. 구역분할기술(zoning technology)은 디스크를 여러 개의 구역으로 나누어 구역 내의 실린더의 물리적 저장면적에 따라 섹터의 수를 다르게 설정하여 바깥쪽 구역의 저장용량을 증가시키는 기술이다. 디스크의 회전 각속도(rotational angular velocity)는 일정하므로 바깥쪽 구역의 섹터 수의 증가는 전송 대역폭의 증가를 가져온다. 따라서 구역분할 디스크(zoned disk)는 각 구역의 전송 대역폭이 다르다. 디스크 서브시스템에서 연속성 보장을 위하여는 일정한 데이터 전송 대역폭이 보장되어야 하는데, 구역분할 디스크는 각 구역의 전송 대역폭이 다르므로 연속성 보장을 유지하는데 어려움이 있다. 구역분할 디스크를 멀티미디어 시스템에서 사용하기 위해 SCAN 스케줄링과 이를 위한 순환적 데이터 블록 배치가 제안되었다[1]. 하지만, 이 기법은 다수의 사용자를 지원할 때 새로운 스트림의 추가 시에 생기는 주기시간의 증가로 인하여

서비스 스트림에 지터를 발생시키는 문제를 가지고 있다.¹⁾ 본 논문에서는 VOD 서버와 같은 멀티미디어 서버에서 구역분할 디스크를 사용했을 때 지터 없는 서비스를 보장하기 위한 기법을 제안한다.

2장에서는 구역분할기술과 SCAN 스케줄링을 소개하며 3장에서는 구역분할 디스크를 멀티미디어 시스템에서 사용했을 때 지터를 발생시키는 문제점에 대하여 설명한다. 4장은 이 문제점을 해결하기 위해 본 논문이 제안한 선행 버퍼링(pre-buffering) 기법을 소개하며, 선행 버퍼링으로 인해 추가적으로 필요한 버퍼 크기와 시작지연시간(startup latency time)의 증가에 대하여 기술한다. 5장에서는 4장에서 제안된 선행 버퍼링의 수학적 모델을 기반으로 시뮬레이션을 수행하며 6장에서는 관련 연구를 기술한다. 마지막으로 7장에서는 결론을 도출한다.

2. 디스크 서브시스템

2.1 디스크의 구역분할기술

디스크는 디스크 중심에 대하여 동심원인 여러 개의 실린더들로 이루어져 있다. 바깥쪽 실린더로 갈수록 디스크 중심에 대한 거리가 멀어지므로 실린더의 물리적인 저장면적이 커지고 디스크 헤드의 선속도도 증가하게 된다. 구역분할기술이 제안되기 전에는 모든 실린더는 동일한 수의 섹터를 가지고 있었다. 이는 모든 실린더가 제일 안쪽 실린더의 전송 대역폭의 제한에 따라 동일한 전송 대역폭을 가짐을 뜻하며 바깥쪽 실린더의 넓은 물리적 저장면적과 빠른 디스크 헤드의 선속도 특성을 이용하지 못함을 의미한다. 디스크의 구역분할기술은 바깥쪽 실린더의 저장면적과 디스크 헤드의 선속도 특성을 충분히 이용하기 위해 나왔는데 이는 디스크를 여러 개의 구역으로 나누어 각 구역의 실린더당 섹터 수를 다르게 설정함으로써 바깥쪽 구역의 저장용량을 늘리고 전송 대역폭을 증가시키는 기술이다. 이 기술이 적용되면 두 가지 측면에서 이점을 가지게 된다. 첫 번째는 바깥쪽 구역의 저장용량의 증가인데 바깥쪽 구역의 실린더는 물리적 저장면적이 충분히 크므로 더 많은 수의 섹터로 나눈다. 각 구역마다 실린더당 섹터 수를 다르게 설정하므로 바깥쪽 실린더의 저장용량은 증가한다. 두 번째로는 전송 대역폭의 증가를 들 수 있다. 이는 실린더의 섹터가 증가하였기 때문인데 같은 회전 각속도를 가지는 디스크 헤드가 실린더에 따라 더 많은 섹터를 읽게 되므로 바깥쪽 구역의 전송 대역폭은 늘어나게 된다. 결과적으로 구역분할 디스크는 저장용량과 평균 전송 대역폭의 증가를 가져왔다(그림 1).

1) 자세한 내용은 본 논문의 2장과 3장 참조.

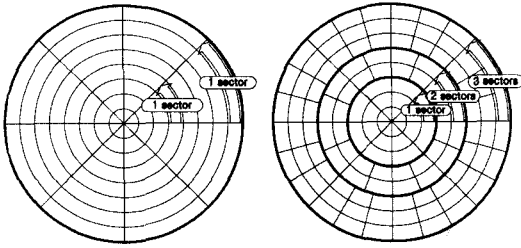


그림 1 기존의 디스크와 구역분할 디스크

2.2 연속성 보장을 위한 디스크 스케줄링

일반적인 디스크 스케줄링과는 다르게 멀티미디어 디스크 스케줄링은 데이터의 연속성을 보장하는 것이 중요하다. 멀티미디어는 정해진 한계시간 내에 데이터가 처리되어 연속적으로 서비스되어야 하므로 디스크 스케줄링도 한계시간 내에 데이터가 메모리에 전송될 수 있도록 수행되어야 한다. 기존의 디스크 스케줄링에는 FIFO (first in first out) 스케줄링, SSTF(shortest seek time first) 스케줄링, SCAN 스케줄링 등이 있다. SSTF 스케줄링은 데이터 블록의 배치를 특별히 고려하지 않는다면 최악의 경우 특정한 데이터 블록에 대하여는 응답하지 않을 수도 있으며 들어오는 순서에 대하여 한계시간 내에 데이터 블록을 보내주는 보장을 해 줄 수 없기 때문에 멀티미디어 스케줄링으로 사용하는 데는 어려움이 있다. FIFO 스케줄링은 들어오는 순서대로 읽어서 전송하므로 멀티미디어 데이터의 순서에는 문제가 없으나 탐색시간이 커지므로 디스크의 성능을 떨어뜨리는 단점이 있다. SCAN 스케줄링은 일정 주기마다 데이터 요청을 모아 한번에 바깥쪽 실린더에서 안쪽 실린더로 데이터를 읽으므로 탐색시간이 줄어든다. 그러나 주기마다 필요한 데이터의 요청을 받아 두었다가 일괄적으로 읽어 들이므로 데이터 블록의 순서가 다를 것이다. 따라서 읽어 들이면서 사용자에게 전송을 한다면 필요한 데이터 블록을 미처 읽지 못하여 생기는 데이터의 지연으로 멀티미디어 서비스에 문제를 일으킬 수 있다. 따라서 읽어 들인 데이터를 버퍼에 저장하였다가 다음 주기에 서비스하는 이중 버퍼링(double buffering) 기법을 사용한다. 이는 버퍼의 크기가 늘어나는 단점이 있지만 데이터 지연을 피할 수 있다.

$$T_s \times r_{pb} \leq n_i \times b \tag{1}$$

$$T_s \geq \sum_{i=1}^s \frac{n_i \times b}{r_{ds}} + \sum T_{sk} + \sum T_l + T_{fs} \tag{2}$$

이중 버퍼링 기법을 이용한 SCAN 스케줄링을 사용했을 때, 디스크에서의 연속성 보장 문제는 수식 (1)과 (2)로 표현될 수 있다. 수식 (1)은 임의의 주기에서 한

스트림이 버퍼에서 소모하는 데이터의 양이 디스크에 의해 버퍼로 읽혀 지는 데이터 양보다 작거나 같아야 함을 말한다. 만약 좌변 항이 우변 항보다 크다면 데이터의 서비스 한계시간이 지켜지지 않아 지터가 발생할 것이다. 수식 (2)는 다음 주기의 사용을 위해 디스크가 데이터를 버퍼로 읽어 오는 시간이 정해진 주기 시간보다 작아야 함을 말한다. 수식 (2)가 만족되지 않으면 정해진 시간 내에 디스크가 필요한 데이터를 버퍼로 공급할 수 없게 되어, 지터가 발생할 것이다. 이 때, 위 두 식을 만족시키는 가장 작은 주기 시간을 선택함으로써, 필요한 버퍼의 크기와 스트림의 시작 지연 시간을 최소화할 수 있다. 따라서 연속성을 보장하는 범위에서 최소의 주기시간으로 스케줄링을 수행하게 된다. 수식 (1)과 (2)를 비롯하여, 본 논문에서 사용되는 모든 변수들에 대한 설명은 부록의 표 1에 나와 있다.

$$T_s \geq \sum_{i=1}^s \sum_{j=1}^z \frac{n_i \times b}{Z \times r_j} + \sum T_{sk} + \sum T_l + T_{fs} \tag{3}$$

한편, 구역분할 디스크는 각 구역의 전송 대역폭이 다르므로 수식 (2)는 (3)과 같이 수정되어야 한다. (3)은 구역분할 디스크의 각 구역의 전송 대역폭이 서로 다를 것을 고려하여 수정한 식이다.

SCAN 스케줄링에서 주기시간은 디스크 서브시스템이 현재 서비스하고 있는 스트림의 개수에 의해 달라진다(수식 (3) 참조). 서비스 스트림의 개수가 늘어나면 주기시간도 늘어나야 한다. 이는 서비스하고 있는 스트림이 늘어나면 동일한 시간 내에 버퍼가 더 많은 데이터를 필요로 하기 때문이다. 이 때, 위에서 설명한 [1]이 제안한 기법을 수정 없이 사용하면 새로운 스트림의 요청 시에 발생하는 주기시간의 증가가 지터를 일으킨다. 이중 버퍼링에서는 데이터를 소모하는 주기가 데이터를 읽는 주기보다 한 주기씩 늦으므로 주기시간의 증가 시에 주기의 불일치가 발생한다(그림 2).

3. 새로운 스트림 요청에 의한 지터(jitter) 발생

시스템의 메모리와 시작지연시간을 최소화 하기 위하여 주기시간은 매 순간 스트림 수에 따라 연속성을 보장하는 범위에서 최소값으로 설정된다. 새로운 스트림 요청이 발생하면 주기시간 내에 더 많은 데이터를 필요로 하므로 주기시간을 늘리게 된다. 이 때 주기시간의 증가는 지터를 발생시킨다.

그림 2는 이중 버퍼링을 사용한 VOD 서버에서 주기시간의 증가로 인해 Stream 3에서 지터가 발생함을 보여 준다. T_1 이전에 디스크는 3개의 스트림을 지원하고 있고 주기시간도 이에 맞추어 설정되어 있다. T_1 에 S_4 의 요청이 들어오면, T_2 이후의 주기시간은 4개의 스트림에

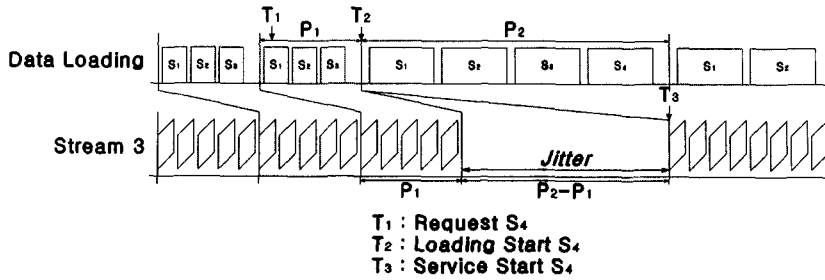


그림 2 주기시간의 증가로 인한 지터의 발생

맞도록 P_1 에서 P_2 로 증가한다. 그러나 서비스를 하고 있던 S_3 의 T_2 시점의 재생은 이전 주기인 P_1 길이만큼의 주기에서 읽은 데이터를 이용한다. 즉 T_2 부터 시작하는 S_3 의 재생 주기는 P_1 이 된다. 따라서 $P_2 - P_1$ 만큼의 지터가 발생한다. 이런 현상은 스트림이 추가될 때마다 일어난다.

4. 선행 버퍼링(pre-buffering)

본 논문은 3장에서 설명한 지터의 발생을 피하기 위해 선행 버퍼링(pre-buffering) 기법을 제안한다. 선행 버퍼링은 서비스의 시작 전에, 필요한 데이터의 일부를 미리 버퍼로 읽어들이어서, 서비스 중에 발생할 수 있는 데이터의 부족 현상을 피하는 기법이다. 선행 버퍼링의 양을 증가시키면 서비스 중에 발생하는 지터를 피할 가능성이 그만큼 커지지만, 메모리와 서비스의 시작지연시간이 증가하는 단점이 있다. 따라서 지터를 피하는 동시에 메모리와 서비스의 시작지연시간을 최소화하는 정확한 선행 버퍼의 크기를 구하는 것이 중요하다.

4.1 버퍼의 크기

2장 2절에서 구역분할 디스크에서의 연속성 보장 문제는 수식 (1)과 (3)으로 표현되었다. 이 때 주기는 수식을 만족하는 하므로 수식 (1)과 (3)을 등식으로 대체하여 연립 방정식으로 풀면 연속성을 보장하는 최소 최소값으로 설정되어야 의 주기 T_s 는 수식 (4)로, 주기에서 각 스트림이 읽어야 할 데이터 블록의 개수 n_i 는 수식 (5)로 표현된다.

$$T_s = \frac{\alpha(S)}{1 - \left(\frac{r_{pb} \times S}{Z} \times \sum_{j=1}^Z \frac{1}{r_j} \right)} \tag{4}$$

$$n_i = \left\lfloor \frac{\alpha(S)}{1 - \left(\frac{r_{pb} \times S}{Z} \times \sum_{j=1}^Z \frac{1}{r_j} \right)} \times \frac{r_{pb}}{b} \right\rfloor \tag{5}$$

수식 (5)에서 계산된 데이터 블록의 개수에 데이터 블록의 크기를 곱하면 주기에 읽어야 할 데이터의 바이트 수 $D_{cs:S}$ 를 수식 (6)과 같이 구할 수 있다. 한편, 이중 버퍼링을 사용하는 SCAN 알고리즘을 사용하므로 스트림이 주기시간에 미디어 재생을 하기 위하여 필요한 주기 서비스 버퍼(cycle service buffer), $B_{cs:S}$ 는 수식 (6)의 2배가 되어 수식 (7)로 표현된다.

$$D_{cs:S} = \left\{ \frac{\alpha(S)}{1 - \left(\frac{r_{pb} \times S}{Z} \times \sum_{j=1}^Z \frac{1}{r_j} \right)} \times r_{pb} \right\} \tag{6}$$

$$B_{cs:S} = 2 \times \left\{ \frac{\alpha(S)}{1 - \left(\frac{r_{pb} \times S}{Z} \times \sum_{j=1}^Z \frac{1}{r_j} \right)} \times r_{pb} \right\} \tag{7}$$

스트림의 서비스 중에 발생할 수 있는 지터의 최대발생시간은 디스크가 최대 사용자를 지원할 때의 주기시간과 현재의 주기시간의 차이만큼이다(식 (8)). 따라서 지터를 방지하는 선행 버퍼의 크기 $B_{p:S}$ 는 식 (9)와 같이 계산된다.

$$\Delta T_s = T_{Max} - T_s \tag{8}$$

$$B_{p:S} = \Delta T_s \times r_{pb} \tag{9}$$

최종적으로 각각의 스트림이 메모리에서 차지하는 스트림 버퍼(stream buffer)의 크기 $B_{st:S}$ 는 식 (7)에서 계산된 주기 서비스 버퍼와 식 (9)에서 계산된 선행 버퍼를 더한 값이므로 식 (10)과 같이 계산된다. 식 (10)은 식 (7)과 식 (9)에 식 (4)를 대입하여 식 (11)과 같이 다시 쓸 수 있다.

$$B_{st:S} = B_{cs:S} + B_{p:S} \tag{10}$$

$$B_{st:S} = (T_{Max} + T_s) \times r_{pb} \tag{11}$$

4.2 선행 버퍼링의 수행

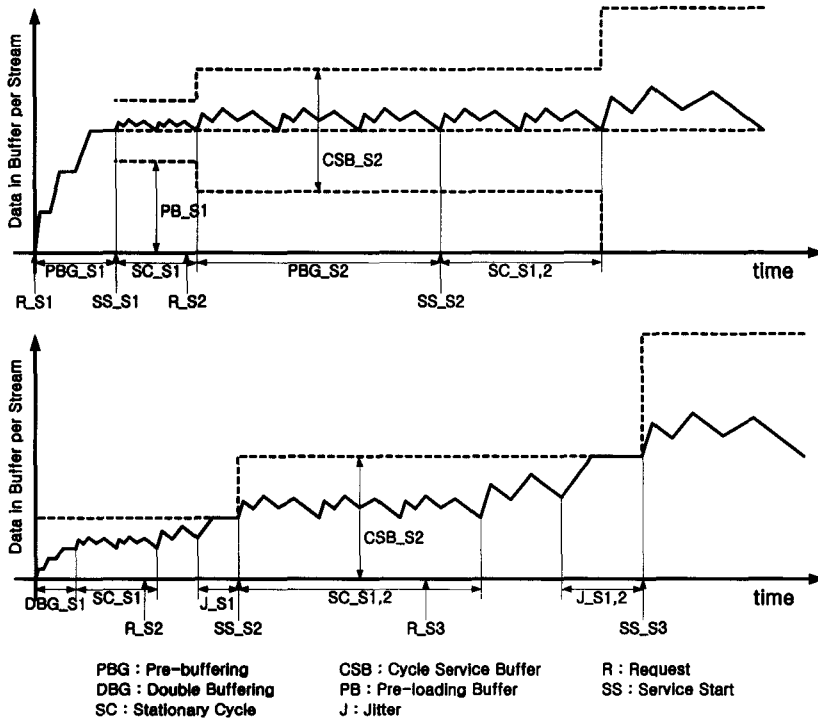


그림 3 선행 버퍼링의 수행

그림 3의 상단 그림은 선행 버퍼링을 수행하는 시스템에서의 시간에 따른 스트림 버퍼 내의 데이터 변화를 나타내고 있다. 첫번째 스트림인 S_1 의 요청이 들어오면 디스크는 선행 버퍼링과 주기 서비스를 위한 이중 버퍼링을 차례로 수행한다. 이 때 버퍼에 저장되는 데이터의 양은 식 (11)에서 계산된 값이 된다. 데이터를 버퍼에 채우는 작업이 끝나면, S_1 은 서비스를 시작한다. S_1 의 서비스 중에 S_2 의 요청이 들어오면 그 다음 주기부터 주기시간은 늘어나게 되고 S_2 의 서비스 시작 전에 선행 버퍼링을 수행한다. 선행 버퍼링을 마친 후 주기 서비스를 위한 이중 버퍼링을 수행하고 S_2 의 서비스가 시작된다. 이 때에도 서비스가 시작되기 전 버퍼에 저장되는 데이터의 양은 식 (11)에서 구해지는 값이다. 그림 3의 하단 그림은 선행 버퍼링을 수행하지 않는 시스템에서의 시간에 따른 스트림 버퍼 내의 데이터 변화를 나타낸다. S_1 의 요청이 들어오면 주기 서비스를 위한 이중 버퍼링을 수행하고 S_1 의 서비스를 시작한다. S_2 의 요청이 들어오면 그 다음 주기부터 주기시간을 늘리게 되는데 이때 S_2 에는 지터가 발생하게 된다.

선행 버퍼링을 수행하면 서비스 시작 전에 선행 버퍼링을 추가로 수행하므로 선행 버퍼링을 하지 않는 시스템보다 시작지연시간이 길어진다. 가령, 그림 3의 상단

그림의 S_2 는 $SS_{S_2} - R_{S_2}$ 만큼의 시작지연시간을 갖는데, 이는 하단 그림 S_2 의 시작지연시간인 $SS_{S_2} - R_{S_2}$ 보다 길다. 결국, 선행 버퍼링은 지터의 발생을 방지하는 장점을 갖지만 버퍼와 시작지연시간의 오버헤드가 있다.

5. 시뮬레이션

4장에 나온 선행 버퍼링의 수학적 모델을 기반으로 하여 시뮬레이션을 수행하였다. 본 논문에서 사용한 디스크 모델은 IBM Deskstar 34GXP이다. 디스크 제조사에서 제공하지 않은 데이터는 일반적인 디스크 모델의 값을 참고하였다. IBM Deskstar 34GXP의 주요 성능 인자는 부록의 표 2에 나와 있다.

그림 4와 그림 5는 비디오의 재생 대역폭이 각각 19.2 Mbps, 1.5 Mbps 일 때, 하나의 스트림이 차지하는 스트림 버퍼의 크기를 나타낸다. M#7, M#6, M#5는 각각 시스템이 서비스 가능한 최대 스트림의 개수를 7개, 6개, 5개로 제한했음을 의미한다. M#90, M#80, M#70, M#60도 마찬가지이다. Y축은 하나의 스트림이 차지하는 버퍼의 크기를, X축은 현재 시스템에 존재하는 스트림의 개수를 의미한다. 그림 4를 보면, 시스템에 존재하는 스트림의 개수가 증가함에 따라 하나의 스트

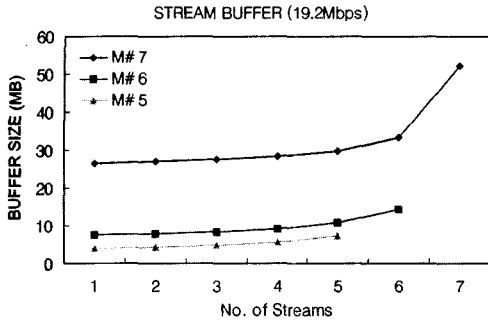


그림 4 선행 버퍼링을 수행할 때, 스트림 버퍼 크기 ($r_{pb} = 19.2Mbps$)

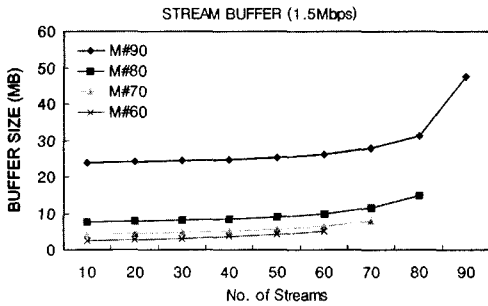


그림 5 선행 버퍼링을 수행할 때, 스트림 버퍼 크기 ($r_{pb} = 1.5Mbps$)

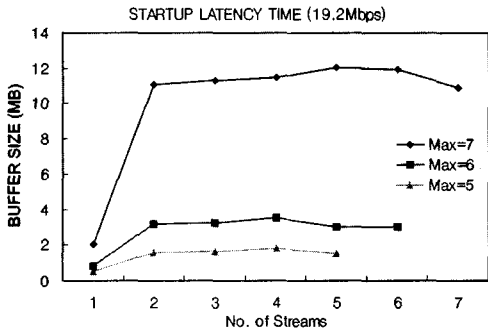


그림 6 선행 버퍼링을 수행할 때, 시작지연시간 ($r_{pb} = 19.2Mbps$)

림이 차지하는 버퍼의 크기가 급격하게 증가함을 보여 준다. 가령 스트림의 개수가 6개에서 7개로 증가할 때, 버퍼의 크기는 약 1.5배 증가한다. 이러한 경향은 그림 5에서도 보여진다. 스트림의 개수가 증가할수록 버퍼의 크기는 급격하게 증가한다. 이는 주기 시간이 스트림의 개수의 제공에 근사하게 비례하기 때문이다.²⁾ (수식 (4)

2) $\alpha(S)$. 즉 데이터 접근 시간은 스트림의 개수에 근사하게 비례하기 때문이다.

참조). 결국, 시스템의 디스크 대역폭 한도를 어느 정도 제한함으로써(가령 80%), 필요한 버퍼의 크기를 크게 줄일 수 있음을 알 수 있다.

그림 6은 재생 대역폭이 19.2Mbps일 때 하나의 스트림이 경험하는 평균 시작지연시간을 나타낸다. 시스템이 서비스 가능한 스트림의 개수를 각각 7, 6, 5로 제한하고 스트림의 개수에 따른 시작지연시간의 변화를 도시하였다. 앞의 버퍼 크기 변화와 비슷하게 시작지연시간도 스트림의 개수가 증가할수록 급격하게 증가한다(수식 (4) 참조). 최대 스트림의 개수를 7개까지 허용할 때 시작지연시간은 약 11초 정도까지 기록되지만 스트림의 개수를 최대 6개로 제한하면, 시작지연시간은 약 4초 정도를 기록한다. 즉, 서비스 가능한 최대 스트림의 개수를 제한함으로써 시작지연시간을 크게 줄일 수 있다.

6. 관련연구

지난 10여년간 멀티미디어 서버에 대한 많은 연구가 있었다[1-6]. [2]에서는 사용자와 서비스 제공자 사이에서 멀티미디어 서비스의 모델과 멀티미디어 시스템의 수학적 모델을 제시하였다. 멀티미디어에서의 중요한 문제는 연속성 보장인데 [3]에서는 연속성 보장을 위하여 FIFO(first in first out) 알고리즘을 제시하였고 [4]에서는 SCAN 알고리즘을 사용한 방법에 대하여 수식적 해석을 하였다. [5]에서는 FIFO 알고리즘과 SCAN 알고리즘의 혼합 형태인 GSS(Group Sweep Scheduling)을 제시하였다. [6]는 실시간 시스템의 입장에서 연속성 보장 문제를 다루었으며 하나의 디스크에서의 디스크 스케줄링에 대한 연구와 더불어 멀티미디어 파일 서버에서의 저장장치 시스템 설계에 대한 연구도 많이 이루어지고 있다[7-11]. 멀티미디어 스트림은 연속성 보장을 위하여 디스크에서의 보장된 전송 대역폭이 필요하다. 멀티미디어 디스크 스케줄링은 연속성을 보장하면서 다수의 사용자를 지원하는데 초점을 맞추고 있다. [12-13]은 디스크 스케줄링에 대한 연구를 디스크 배열로 확장하였다.

7. 결론

우리는 본 논문에서 멀티미디어 서버에서 구역분할 디스크의 특성을 충분히 이용하면서도 지터를 피할 수 있는 선행 버퍼링(pre-buffering) 기법을 제안하였다. 선행 버퍼링은 디스크와 서버의 정보를 통해 만들어진 디스크의 수학적 모델을 통하여 데이터의 부족분을 예측하고 스트림 서비스 전에 예측된 데이터 양 만큼을 버퍼링하여 지터 발생 시에 이 데이터를 소모함으로써 지터를 피하는 기법이다. 선행 버퍼링을 수행하면 스트림 서비스

전에 선행 버퍼 데이터를 읽어야 하므로 다소의 버퍼와 시작지연시간 오버헤드가 있다. 하지만 시뮬레이션을 통해 시스템이 서비스 가능한 스트림의 개수를 어느 정도 제한함으로써(가령 80%), 필요한 버퍼의 크기와 시작지연시간을 크게 줄일 수 있음을 알 수 있었다. 결국, 선행 버퍼링 기법은 다소의 버퍼와 시작지연시간 오버헤드가 있지만 지터를 방지할 수 있기 때문에 실제 멀티미디어 서버에 적용되어 사용자에게 좀더 좋은 품질의 서비스를 제공하는 데 기여할 수 있으리라 판단된다.

참 고 문 헌

[1] S. Ghandeharizadeh, S. H. Kim, and C. Shahabi, "Continuous Display of Video Objects Using Multi-Zone Disks," Second International Baltic Workshop on DB and IS, June 1996.

[2] D. P. Anderson, Yoshitomo Osawa, and Ramesh Govindan, "A File System for Continuous Media," ACM Trans. Comput. Syst., 10(4): 311-337, Nov 1992.

[3] P. Rangan, H. Vin, and S. Ramanathan, "Designing an on-demand multimedia service," IEEE Communication Magazine, 30(7):56-65, July 1992.

[4] D. R. Kenchammana-Hosekote and J. Srivastava, "Scheduling Continuous Media on a Video-On-Demand Server," In Proc. of International Conference on Multi-media Computing and Systems, Boston, MA, May 1994. IEEE.

[5] M.-S. Chen, D.D. Kandlur, and P.S. Yu, "Optimization of the Grouped Sweeping Scheduling (GSS) with Heterogeneous Multimedia Streams," In Proc. of ACM Multimedia '93, pp. 235--242, Anaheim, California, August 1993.

[6] A. L. N. Reddy and J. Wyllie, "Disk Scheduling in a Multimedia I/O system," In Proc. ACM Multimedia Conf., pages 225-233, ACM Press, New York, 1992.

[7] J. Gemmell and S. Christodoulakis, "Principles of Delay Sensitive Multi-media Data Storage and Retrieval," ACM trans. on Information System, 10(1):51-90, January 1992.

[8] J. Gemmell, "Multimedia Network File Servers: Multi-Channel Delay Sensitive Data Retrieval," In Proc. of 1st ACM Multimedia Conf. ACM, Oct. 1993.

[9] Harrick M. Vin, Pawan Goyal, Alok Goyal, and Anshuman Goyal, "A Statistical Admission Control Algorithm for Multimedia Servers," In Proc. of ACM Multimedia Conf., pages 33-40, San Francisco, CA, Oct. 1994.

[10] Harrick M. Vin, Alok Goyal, Anshuman Goyal, Pawan Goyal, "An Observation-Based Admission Control Algorithm for Multimedia Servers," In Proc. of 1st IEEE International Conf. on

Multimedia Computing and systems, pages 234-243, Boston, MA, May 1994.

[11] B. Ozden, A. Biliris, R. Rastogi, and Avi Silberschatz, "A Low-Cost Storage Server for Movie on Demand Databases," In Proc. of VLDB '94, 1994.

[12] Lougher P. and Shepherd D., "The design of a storage server for continuous media," The Computer Journal, 36(1):32-42, 1993.

[13] Antine Mourad, "Issues in design of a storage server for video-on-demand," Multimedia Systems, 1996(4):70-86, 1996.

부 록

1. 본 논문에서의 변수 정의

표 1 본 논문에서의 변수 정의

$a(S)$	탐색시간과 회전지연시간 등 데이터 접근시간
$B_{cs:S}$	각 스트림의 주기 서비스 버퍼 크기
B_{ps}	각 스트림의 선행 버퍼 크기
$B_{st:S}$	각 스트림의 스트림 버퍼 크기
b	디스크의 데이터 블록 크기
$D_{cs:S}$	주기시간에 읽는 데이터 크기
n_i	주기시간에 읽는 데이터 블록 개수
r_{ds}	디스크의 전송 대역폭
r_j	각 구역의 전송 대역폭
r_{pb}	미디어의 재생 대역폭
S	현재 서비스 스트림 개수
T_{fs}	디스크 헤드의 외각 실린더 복귀시간
T_l	디스크 헤드의 실린더 탐색시간
T_{Max}	최대 사용자 주기시간
T_S	스트림 개수에 따른 주기시간
T_{sk}	디스크 헤드의 섹터 회전지연시간
Z	구역의 개수

2. 시뮬레이션에 사용된 디스크 디바이스의 데이터

표 2 시뮬레이션에 사용된 디스크 디바이스에 대한 데이터

디스크 모델	IBM Deskstar 34GXP
저장용량	27.3 GB
전송 대역폭	13.8~22.9 MB/s
실린더 개수	17494
탐색시간 파라미터 a	2.0987e-3
탐색시간 파라미터 b	1.0132e-4
회전 지연시간	4.17 msec
데이터 블록 크기	4 KB



조 경 선

1999년 한양대학교 공과대학 전자·전자통신·전파공학과 학사. 2001년 한양대학교 대학원 전자통신전파공학과 석사. 2001년 9월~현재 ㈜넷앤티비 연구소 선임연구원. 관심분야는 방송 시스템, DTV, DMB, DAB, 멀티미디어 시스템, 멀티미

디어 네트워킹

원 유 집

정보과학회논문지 : 시스템 및 이론
제 31 권 제 5 호 참조



신 일 훈

서울대학교 계산통계학과 학사. 서울대학교 전산학과 석사. 현재 서울대학교 전기컴퓨터 공학부 박사과정. 관심분야는 멀티미디어 시스템, 웹캐쉬, 운영체제

고 건

정보과학회논문지 : 시스템 및 이론
제 31 권 제 2 호 참조