가변 비트율 데이터 스케쥴링을 위한 디스크 대역폭과 버퍼 할당의 Tradeoff 기법*

김 인 환 김 정 원 정 기 동
부산대학교 전자계산학과

A Tradeoff Scheme Between Disk Bandwidth and Buffer Allocation for VBR Data Scheduling

In-Hwan Kim Jeong-Won Kim Ki-Dong Chung
Department of Computer Science, Pusan National University

요 약

비디오 서버에서 가변 비트율 데이터를 위한 다양한 스케쥴링 기법이 제안되어 왔다. 가변 비트율로 저장장치에 저장된 비디오 데이터를 재생할 때 최악의 경우를 가정하여 디스크 대역폭을 할당하면 디스크 대역폭 이용율이 매우 낮아진다. 디스크 대역폭의 이용율을 높이기 위한 방법으로 선반입 기법을 사용하는데 이를 위해서는 많은 버퍼가 필요하다. 본 논문에서는 가변 비트율 데이터를 위한 윈도우별 선반입에 기반한 데이터 스케쥴링에서 시스템의 자원 상황을 고려하여 적절한 디스크 대역폭과 버퍼 할당을 Tradeoff 하는 기법을 제안한다. 제안한 기법은 비디오 데이터에 대한 메타 데이터를 미리 구축하고 이에 따라 자원을 윈도우 단위로 동적으로 할당한다. 선반입은 윈도우 단위로 실시하며 선반입에 필요한 버퍼가 과다하게 증가하는 것을 조절한다. 모의 실험에서는 기존의 가변 비트율 데이터 스케쥴링 기법과 제안한 기법의 자원 할당량과 동시 지원 가능 사용자 수를 실험하고 비교하였다.

1. 서론

VOD 시스템에서는 비디오 데이터를 저장, 전송하기 위해서 MPEG과 같은 고압축률의 압축 기법이 사용된다. MPEG은 프레임간 움직임 차를 이용하여 압축하므로 프레임들은 시간에 따라 크기가 가변적인 가변 비트율의 특성을 가지며 비트 변화율 역시 큰 편차를 보인다[1]. 그림 1은 영화 *마스크(The Mask)*의 MPEG1 데이터를 1초를 주기로 하여 초당 30 프레임으로 전송할 때, 주기별 요구 데이터량을 8 KByte의 블록의 개수로 나타낸 것이다. 주기당 요구 블록 수는 4 블록에서 29 블록까지 큰 편차를 보인다. 그런데 기존의 자원 관리 기법들은 대부분 고정 비트율을 가정하여 자원 이용률이 낮을 뿐 아니라 가변 비트율을 가지는 비디오 데이터를 전송하는데 있어 overflow나 underflow와 같은 상황에 대한 관리가 복잡해 진다. 따라서 비디오 서버는 비디오 데이터의 이러한 특성을 고려한 효율적인 자원 관리가 필요하다.

본 논문에서는 윈도우 단위로 최대 요구 대역폭을 설정, 선반입을 실시하여 디스크 대역폭과 버퍼를 동적으로 할당하는 기법인 WVS(Window-based VBR data Scheduling)[2]에서 시스템의 자원 상황을 고려하여 적절한 디스크 대역폭과 버퍼 할당을 Tradeoff 하는 XWVS(Extended Window-based VBR data Scheduling) 기법을 제안한다.

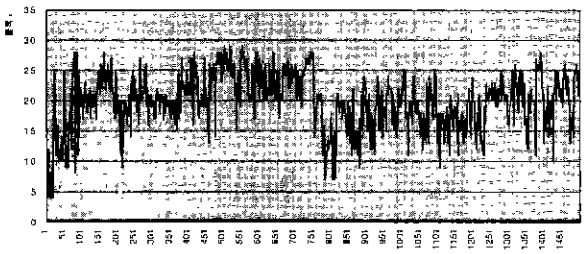


그림 1 주기당 요구 데이터

2. 관련 연구

가변 비트율로 압축된 MPEG 데이터를 디스크에 저장 및 검색하는 방법에는 일정 시간 동안의 데이터, 즉 가변 크기의 데이터를 읽는 CTL(Constant Time Length) 방식과 매 주기 마다 일정 크기의 데이터를 읽는 CDL(Constant Data Length) 방식 및 고정 크기의 데이터를 가변 개수로 읽는 Hybrid 방식이 있다. CDL의 경우 버퍼를 이용해 서버와 클라이언트간 재생률을 조정해야하므로 overflow 및 underflow에 대한 관리 비용이 높아진다[3].

비디오 데이터를 위한 버퍼 관리는 크게 정적 버퍼 할당 방식과 동적 버퍼 할당 방식이 있다[4]. 정적 버퍼 할당 방식은 전체 주기 동안 고정적으로 버퍼를 할당하는 방식으로 VBR 데이터의 경우 자원의 낭비가 크다. 동적 버퍼 할당 방식은 시간에 따라 다른 크기의 버퍼를 할당하는 방식으로 전체적으로 버퍼의 활용도는 높아지나 버퍼 관리 비용의 증가를 초래한다.

[5]에서는 디스크 대역폭 및 버퍼의 크기를 블록 단위로 환산하여 자원을 관리하고 데이터 접근 시 선반입을 통해 비트 변화율을 줄이려는 시도가 있었다. 또한 이 방법의 문제점인 선반입 버퍼가 증가하는 문제를 해결하기 위해 전체 주기를 제한된 최대 요구 대역폭으로 선반입하여 데이터를 스케쥴링하는 혼합 비트율(Hybrid Bit Rate) 스케쥴링과 이를 이용해 승인제어를 하려는 시도가 있었다. 그러나 이 방법 역시 전체 구간을 동일한 대역폭을 기준으로 선반입 함으로써 디스크의 대역폭을 효율적으로 사용하지 못하며 선반입을 위한 버퍼의 크기가 매우 커지는 단점이 있다. 또한 정적 버퍼 할당 방법을 사용하므로 버퍼가 성능 향상의 제약 조건이 된다.

본 논문에서는 Hybrid 방식을 확장하여 전체 스트림을 여러 윈도우로 나누고 나누어진 윈도우 단위로 최대 요구 대역폭을 설정하여 선반입 함으로써 선반입을 위한 버퍼의 크기를 크게 줄이고 또한 구간 단위로 동적으로 버퍼를 할당함으로써 버퍼를 효율적으로 사용하는 WVS 기법에서 시스템의 자원 상황에 맞게 적절히 자원 할당을 실행하는 데이터 스케쥴링 및 버퍼 관리 기법을 제안하고 실험하였다.

* 본 연구는 정보통신연구관리단의 '98 대학기초연구지원사업에 의해 일부 지원 받았음

3. XWVS (Extended Window-based VBR data Scheduling)

3.1 정의

XWVS에서는 전체 스트림에서 요구 대역폭이 비슷한 주기들을 하나의 단위로 묶어 같은 디스크 대역폭과 버퍼를 할당하는데 이를 윈도우라 한다 스트림을 윈도우로 나누는 방법은 [1,2]에서 설명하고 있다 표 1은 본 논문에서 사용되는 기호들의 정의이다

표 1 사용되는 기호들의 정의

W_i	윈도우 i, $W_i = (W_i.start, W_i.end)$
D	총 디스크 대역폭
B	총 버퍼 크기
$Length(W_i)$	윈도우 i의 길이
$MaxBW(W_i)$	윈도우 i 안의 최대 요구 대역폭의 크기
$Average(W_i)$	윈도우 i의 평균 요구 대역폭의 크기
θ	디스크 대역폭과 버퍼량 할당의 threshold
$BW(W_i)$	윈도우 i에 할당되는 대역폭의 크기
$Buff(W_i)$	윈도우 i에 할당되는 버퍼의 크기
$Accum(W_i)$	윈도우 i에서 선반입을 위해 할당되는 최대 누적 버퍼의 크기

비디오 서버는 스트림의 메타 데이터를 미리 구축하여 고객의 서비스 요구 시 이를 바탕으로 자원을 할당한다 표 2는 임의의 스트림에 대한 메타 데이터 테이블에 대한 예이다

표 2 스트림에 대한 메타 데이터 테이블

W_i	W_1	W_2	W_3	W_4
$(W_i.start, W_i.end)$	(1, 6)	(7, 22)	(23, 33)	(34, 38)
$Length(W_i)$	6	16	11	5
$MaxBW(W_i)$	7	8	8	5
$Average(W_i)$	4	4	5	4
$BW(W_i)$	6	5	7	6
$Accum(W_i)$	1	7	1	0
$Buff(W_i)$	7	12	8	6

3.2 대역폭과 버퍼량 사이의 Tradeoff

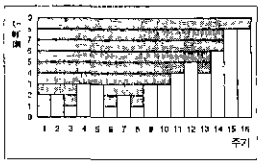
윈도우 단위로 선반입을 실시할 때 디스크 대역폭을 낮게 할당하면 선반입에 필요한 버퍼량이 증가하여 버퍼가 성능 제약 요인이 된다 반대로 디스크 대역폭을 너무 크게 할당하면 최악의 경우로 할당한 경우와 마찬가지로 디스크 대역폭의 이용률이 낮아져 디스크 대역폭이 성능 제약 요인이 된다 따라서 시스템의 자원 상황에 맞게 디스크 대역폭과 버퍼량을 할당하여야 한다 그림 2는 위의 메타 데이터 테이블에서 W_2 에 대한 예로써 디스크 대역폭 할당을 조절함에 따라 디스크 대역폭의 이용률과 선반입을 위해 필요한 버퍼가 변화함을 보여 준다 윈도우 i에 할당되는 디스크 대역폭은 식 (1)로 표현된다

$$BW(W_i) = Average(W_i) + \theta$$
$$(0 \le \theta \le MaxBW(W_i) - Average(W_i)) \quad (1)$$

윈도우 i 에 할당되는 버퍼량은 식 (2)로 표현된다

$$Buff(W_i) = Average(W_i) + \theta + Accum(W_i)$$
$$(0 \le \theta \le MaxBW(W_i) - Average(W_i)) \quad (2)$$

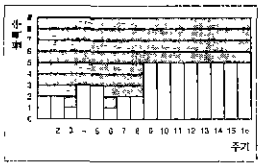
그림 2에서와 같이 $BW(W_i)$를 크게 할당하면 즉, θ를 증가시키면 $Accum(W_i)$가 낮아지고, $BW(W_i)$를 낮게 할당하면 즉, θ를 낮추면 $Accum(W_i)$가 높아진다 따라서 θ를 조절함으로써 시스템의 자원 상황에 맞게 디스크 대역폭과 버퍼 할당량을 조절하는 것이 가능하다



원래 대역폭

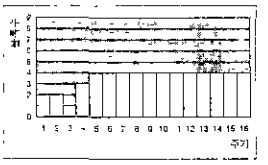
2 2 1 3 3 1 2 1 3 3 4 5 4 6 8 8

$Average(W_i) = 4$



$\theta = 1$ 로 선반입한 경우

대역폭	2 2 1 3 3 1 2 1 3 3 4 5 4 6 8 8
선반입후	2 2 1 3 3 1 2 2 5 5 5 5 5 5 5 5
선반입시 필요버퍼	0 0 0 0 0 0 0 1 3 5 6 6 7 6 3 0



$\theta = 0$ 로 선반입한 경우

2	2	1	3	3	1	2	1	3	3	4	5	4	6	8	8
2	2	1	3	4	4	4	4	4	4	4	4	4	4	4	4
0	0	0	0	1	4	6	9	10	11	11	10	10	8	4	0

그림 2 윈도우 선반입의 예

승인 제어 시 더 이상 사용자를 승인하지 못하는 경우는 디스크 대역폭에 여유가 없는 경우와 버퍼에 여유가 없는 경우의 두 가지로 나눌 수 있다 따라서 자원을 할당할 때 어느 한 자원은 여유가 있는데 다른 한 자원만이 모두 소비되는 경우를 막아 두 자원을 모두 최대로 사용하는 경우를 만들어 내는 것이 필요하다. 이는 모든 윈도우에서 디스크 대역폭과 버퍼가 같은 비율로 소비되도록 하면 된다

즉 모든 윈도우에서 디스크 대역폭 $BW(W_i)$과 버퍼량 $Buff(W_i)$ 이

$$\frac{Buff(W_i)MB}{BW(W_i)MB/\sec} = \frac{(Average(W_i)+\theta+Accum(W_i))MB}{(Average(W_i)+\theta)MB/\sec}$$
$$= 1\frac{MB}{MB/\sec} + \frac{Accum(W_i)MB}{(Average(W_i)+\theta)MB/\sec} = \frac{B(MB)}{D(MB/\sec)} \quad (3)$$

의 비율로 할당되면 된다.

즉, $\forall W_i \ \frac{Buff(W_i)}{BW(W_i)} = \frac{B}{D}$ 이면 고객의 요구 도착의 분포에 관계없이 항상 두 자원의 할당량은 같은 비율로 증가한다 따라서, 시스템의 자원 상황에 적절한 디스크 대역폭과 버퍼 할당량 사이의 tradeoff 가 가능하다

다음은 위에서 설명한 기법의 적용 예이다

디스크 대역폭 $D = 8\ MB/sec$ 이고, 버퍼량 $B = 32\ MB$ 일 때,

식 (3)에 의해 $\frac{Buff(W_i)}{BW(W_i)} = \frac{B}{D} = \frac{32MB}{8MB/\sec} = 4\frac{MB}{MB/\sec}$ 이므로

$$1\frac{MB}{MB/\sec} + \frac{Accum(W_i)MB}{(Average(W_i)+\theta)MB/\sec} = 4\frac{MB}{MB/\sec}$$

$$Accum(W_i)MB = 3\frac{MB}{MB/\sec}(Average(W_i)+\theta)MB/\sec = 3BW(W_i)MB$$

$$\therefore Accum(W_i) = 3BW(W_i)$$

즉, 선반입에 필요한 버퍼량이 할당되는 대역폭에 3 배가 되게 하는 θ를 선택한다 $Accum(W_i)$가 $BW(W_i)$의 정확히 3 배가 되는 θ가 존재하지 않을 경우에는 가장 근사한 비율을 만드는 θ를 선택한다 그림 2 를 예로 들면 $\theta = 0$ 일 때 $Accum(W_i) \approx 11$, $BW(W_i) = 4$ 로 $Accum(W_i)$ $BW(W_i) = 3$ 1 에 가장 근사하므로 최적의 θ 값은 0 이다

만약 디스크를 하나 더 설치하여 $D = 16$ MB/sec 라면

$$\frac{Buff(W_i)}{BW(W_i)} = \frac{B}{D} = \frac{32}{16} = 2\frac{MB}{MB/\sec}$$

$$1\frac{MB}{MB/\sec} + \frac{Accum(W_i)MB}{(Average(W_i)+\theta)MB/\sec} = 2\frac{MB}{MB/\sec}$$

$$Accum(W_i)MB = 1\frac{MB}{MB/\sec} \times (Average(W_i)+\theta)MB/\sec = BW(W_i)MB$$

즉, 선반입에 필요한 버퍼량이 할당되는 대역폭과 같이 되게 하는 θ를 선택한다. 그림 2를 예로 들면 θ = 1 일 때 $Accum(W_i) = 7$, $BW(W_i) = 5$ 로 $Accum(W_i)$ $BW(W_i) = 1$ 1 에 가장 근사하므로 최적의 θ 값은 1 이다

4. 모의 실험

4.1 모의 실험 모델

MPEG1 스트림 분석기를 이용한 결과 실험에 사용한 영화 *마스크 (The Mask)* MPEG1 스트림은 블록의 크기가 8KB 이고 1 초에 30 프레임을 재생할 경우 평균 20 block/sec, 최대 29 block, 최소 4 block 의 주기당 요구 블록수를 보였다. 실험에 이용한 스트림의 길이는 1500 초 분량이며, 사용자 요구의 도착은 포아송 분포로 발생시켰다

모의 실험은 다음과 같은 기존의 기법들과의 비교로 이루어졌다

① CBR : 선반입을 하지 않고 최악의 주기를 기준으로 디스크 대역폭을 할당 전체 스트림에 대하여 정적 버퍼 할당

② HBR : 스트림을 윈도우로 나누지 않고 전체 스트림 단위로 선반입 전체 스트림에 대하여 정적 버퍼 할당

③ WVS 윈도우 단위로 선반입하지만 모든 윈도우에서 동일한 Threshold 사용 즉, 시스템의 자원상황을 고려하지 않음 동적 버퍼 할당

4.2 XWVS 와 기존 기법들의 자원 할당량 비교

그림 3 은 대상 기법들에 대한 자원 할당량 비교이다 모든 기법에 대하여 디스크 대역폭 2500 block/sec, 버퍼 크기 5000 block 하에서 실험하였다. XWVS 는 시스템의 자원 상황을 고려하여 디스크 대역폭과 버퍼를 할당하므로 WVS 보다 두 자원의 할당량이 최대 및 평균의 경우에서 모두 낮다

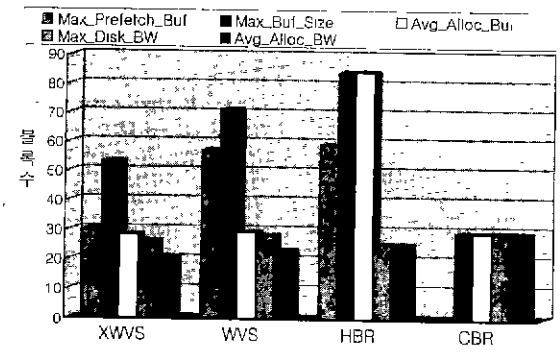
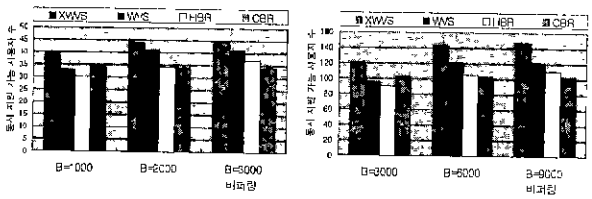


그림 3 기법들의 자원 할당량 비교

4.3 기법별 동시 지원 가능 사용자 수 비교

그림 4 는 기법별로 동시에 지원 가능한 사용자 수를 나타낸다 XWVS 는 기존의 다른 기법들에 비하여 충분한 버퍼가 있는 경우 뿐만 아니라 버퍼가 적은 경우에도 월등히 많은 사용자를 지원한다 이는 버퍼가 적은 경우에 선반입을 적게 하고 버퍼가 충분한 경우에는 디스크 대역폭을 낮추어 선반입을 크게 허여 디스크 대역폭의 이용율을 높이는 방법을 사용하기 때문이다 버퍼 크기가 디스크 대역폭의 3 배 이상이 될 경우에는 동시 지원 가능 사용자 수의 증가가 둔해지는데 이는 선반입에 필요한 버퍼 크기가 디스크 대역폭의 2 배 이상 되도록 디스크 대역폭 할당을 더 이상 낮출 수 없기 때문이다 즉, θ 가 0 이 되어도 선반입에 필요한 버퍼량이 디스크 대역폭 할당량의 2 배에 크게 못 미치거나 θ 를 더 이상 낮출 경우 윈도우 내에서 선반입을 끝낼 수 없는 경우가 많이 발생하기 때문이다 CBR 과 HBR 의 경우 버퍼 크기량이 늘어나도 동시 지원 가능 사용자수의 변화가 거의 없는 것은 시스템 자원 상황을 고려하지 않아 두 자원간의 Tradeoff 가 거의 일어나지 않기 때문이다 따라서 이 두 기법에서는 디스크 대역폭이 심한 bottleneck 자원으로 작용한다



(a) 디스크 대역폭 $D = 1000$ (b) 디스크 대역폭 $D = 3000$

그림 4 기법별 동시 지원 가능 사용자 수

5. 결론

기존의 가변 비트율 데이터 스케쥴링 기법들은 디스크 대역폭의 이용율이 낮으며, 이용율을 높이기 위한 방법인 선반입 기법 또한 과다한 버퍼 사용의 증가를 보인다 본 논문에서는 시스템의 자원 상황에 맞게 디스크 대역폭과 버퍼의 할당을 적절히 tradeoff 하는 XWVS 를 제안하였다 XWVS는 동시 지원 가능 사용자 수에 있어서 기존의 데이터 스케쥴링 기법에 비해 월등한 성능 향상이 있음을 보였다. 향후 연구과제로는 첫째, 자원의 가격을 고려하여 디스크 대역폭과 버퍼의 비용 효율적인 tradeoff에 관한 연구가 진행되어야 할 것이고 둘째, 윈도우 크기에 따른 시스템 성능 분석이 필요하다

참고 문헌

[1] In-Hwan Kim Jeong-Won Kim Ki-Dong Chung, "VBR Data Scheduling and Admission Control in Video Server", ISAS'98 Proceeding, July, 1998

[2] 양민영, "구간별 선반입에 기반한 VBR 데이터 스케쥴링 및 수용제어", 부산대학교 석사 논문, 1998

[3] Jaber A Al-Marri, Shahram Ghandeharizadeh, "An Evaluation of Alternative Disk Scheduling Techniques in Support of Variable Bit Rate Continuous Media", Proc of the International Conference on Extending Database Technology (EDBT) 1998

[4] 류연승, 고 건, "가변 데이터 율을 갖는 연속미디어 데이터를 위한 동적 버퍼 관리 기법", 정보과학회논문지(A) 제25권 제3호, 1998

[5] D Makaroff, G Neufeld, N Hutchinson, "An Evaluation of VBR Disk Admissions for Continuous Media File Servers", ACM Multimedia'97 Proceedings, 1997

[6] Weifeng Shi, Shahram Ghandeharizadeh "Trading Memory for Disk Bandwidth in Video-On-Demand Servers", Proceedings of 13th ACM Symposium on Applied Computing, 1998