

승인을 향상을 위한 동적 버퍼 공유 사이즈 결정 메카니즘의 설계 및 평가

박 규 석, 송 태 섭, 김 연 실*
경남대학교 컴퓨터 공학과

Design and Evaluation of a Mechanism for
the Decision of Dynamic Buffer Sharing Size
to improve on admission rate

Kyoo-Seok Park, Tae-Seob Song, Yeon-Sil Kim
Dept. of Computer Engineering, Kyungnam University

요 약

대용량 데이터, 높은 전송율, 실시간 제한의 특성을 가진 연속미디어 데이터 서비스를 지원하는 시스템은 집중적인 I/O 발생으로 인해 서비스 가능 사용자 수는 활용할 수 있는 여유자원에 의해 제한된다. 그러므로 본 논문은 인접한 요구간의 시간 간격(interval)을 버퍼에 캐싱하여 공유하는 기법을 기반으로, 동적으로 버퍼 공유 크기를 결정하고 요구들을 그룹핑함으로써 인접한 블록들의 버퍼 점유를 막고, 디스크 액세스를 감소시킨다. 또한 그룹 반환을 통해 버퍼 사용 효율을 높이고 여유 자원을 확보함으로써 승인제어에서 자원 활용 효율이 향상됨을 시뮬레이션을 통해 보인다.

I. 서론

VOD와 VOD-like 서비스가 인기를 얻으면서 수천 명의 사용자를 지원하면서 사용자 서비스 품질을 유지할 수 있는 서버 시스템에 대한 요구가 높아지고 있다. 사용되는 연속미디어 데이터는 대용량 데이터, 높은 전송율, 실시간 제한의 특성을 가지고 있고, 사용자 서비스를 지원하는 시스템은 집중적인 I/O 발생으로 인해 서비스 가능 사용자 수는 활용할 수 있는 여유자원에 의해 제한된다. 그러므로 인기 있는 비디오에 대한 동시 서비스

수를 증가시키기 위하여 I/O 대역폭과 버퍼와 같은 자원을 공유하는 여러 기법들을 통해 I/O 자원의 요구를 줄인다.

데이터 공유 효과를 최대로 하기 위한 기법으로는 지정된 시간 안에 도착하는 사용자들 하나의 I/O 스트림으로 서비스하는 batching, 디스크에서 패치된 데이터를 버퍼에 저장하고 사용자가 요구하는 데이터를 버퍼에서 서비스함으로써 디스크 I/O를 감소시키는 buffering, 그리고 후행 요구와 동일한 자원으로 사용할 수 있도록 사용자의 재생 속도를 변화시키는 piggybacking 등이 있다. 이는 시스템

이 데이터 액세스를 제어하여 read-only 데이터를 공유함으로써 다수의 사용자를 지원할 수 있는 방법이다. 부가적인 I/O를 요구하는 interactive VOD 기능을 지원하기 위하여 batching과 buffering을 함께 사용하거나 시스템의 성능향상을 목적으로 buffering과 caching을 통합한 기존 시스템[4]은 버퍼와 I/O 대역폭이 반비례 관계를 갖고 있으므로 항상 tradeoff를 경험하게 된다.

기존의 버퍼 공유 기법은 다수 요구간의 시간적 지역성을 고려하여 버퍼에 저장된 데이터를 다수의 요구가 공유하도록 함으로써, 동시에 서비스 가능한 사용자 수를 늘여서 디스크 대역폭 활용도를 높이고자 한다. 그러나 간단한 버퍼 공유기법에서의 버퍼와 디스크 대역폭 병목현상으로 인해 시스템 자원 활용도가 낮아지기 때문에 최적 크기의 버퍼 공유를 결정할 필요가 있다.[7]

본 논문은 I/O 오버헤드를 감소시키기 위해 한번 서비스된 데이터를 다음 사용자를 위해 버퍼에 캐싱하고, 사용자의 데이터 공유를 허용하는 최적의 버퍼 공유 크기를 결정하는 그룹핑 알고리즘을 제안한다. 사용자 요구 도착 시간에 따라 동적인 공유 크기를 결정하고, 버퍼를 공유하는 요구들을 그룹핑함으로써 인접한 블록들의 버퍼 점유를 막고, 디스크 액세스를 감소시킨다. 또한 그룹 반환을 통해 버퍼 사용 효율을 높이고 여유 자원을 확보함으로써 승인제어에서 자원 활용 효율이 향상됨을 보인다.

II. 관련연구

2.1 SHR Scheme

SHR은 스트림을 동일한 데이터율을 갖는 세그먼트 단위로 나누고, 후행 요구가 도착한 시점부터 선행 요구의 데이터를 캐싱하여 공유하여 실제 디스크 대역폭에 비해 많은 요구를 지원할 수 있다. 또 배칭을 통해 스케줄링 주기를 조정함으로써 사용자 응답시간을 감소시킨다.[2]

버퍼크기, 데이터율, 평균 도착시간에 의해 버퍼 공유 크기가 결정되기 때문에, 버퍼 공유의 크기가 너무 크거나 오랜 기간동안 버퍼에 남아 있게 될 경우, 이후에 도착하는 요구의 버퍼 공유 효율을 감소시키고, 디스크에서 데이터를 액세스 하도록 함으로써 디스크 I/O가 증가하는 단점이 있다.

2.2 연속 미디어 캐싱 알고리즘

같은 연속미디어 데이터를 접근하는 이웃한 두 요구사이의 시간적 지역성을 나타내는 시간 간격(interval)을 저장하는 interval caching [5]기법은

짧은 interval을 우선으로 버퍼를 할당하여 데이터를 저장한다. 이는 연속적인 액세스가 가능한 객체를 캐싱함으로써 높은 버퍼 히트율을 제공하며, 접근패턴의 변화를 동적으로 수용하지만, 제한된 크기의 캐쉬에 대해 사용 요구수를 최대로 하기 위해 interval의 크기가 작은 것부터 서비스를 해주기 때문에 긴 interval을 가진 요구의 응답시간이 상당히 길어지며, 요구가 들어올 때마다 크기를 비교하기 때문에 처리 오버헤드가 높다.

2.3 고정된 공유크기의 버퍼 공유

구체적인 공유 크기가 제시된 이 기법은 시스템의 효율을 고려하여 최적의 버퍼 공유 크기(distance threshold)를 결정하고, 인접한 두 요구간의 간격이 공유 크기를 초과하지 않으면 버퍼 공유를 허용한다[3]. 이 기법은 할당된 공유크기가 고정적이기 때문에 인기 비디오 액세스의 경우, 인접한 블록으로 인한 버퍼 점유현상이 발생하며 서비스되는 디스크 스트림 수가 제한될 가능성이 있다.

2.4 배칭과 버퍼링을 이용한 자원 할당

부가적인 I/O 요구하는 VCR-like interactive 요구를 지원하기 위해 batching과 buffering의 장점을 이용하여 효과적으로 자원을 관리하는 기법[6]으로, 인기없는 비디오에 대해서는 batching interval 이내의 요구들을 하나의 I/O 스트림으로 서비스하고, partition 모델을 이용하여 VCR 기능 이후 정상 재생을 시작할 때 요구되는 작업부하를 감소시키면서, 적절한 시스템 크기를 결정하여 비용 효율을 향상시킨다.

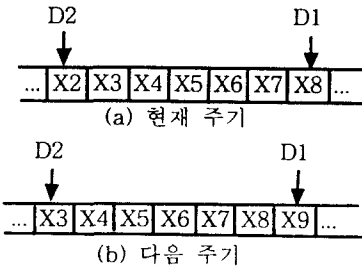
III. 그룹핑 알고리즘

요구 도착 시간에 따라 동적으로 공유 크기를 결정하고, 요구들을 그룹핑하여 사용자 요구 도착 패턴의 변화로 인한 캐싱의 효율적 측면을 고려하고, 그룹반환을 통해 자원 활용도를 높이는 기법을 제시한다.

3.1 버퍼 공유

다음은 본 논문에서 사용되는 버퍼 공유 기법을 설명하고 있다. (a)는 현재 주기에서 두 요구가 버퍼에서 읽고 있는 데이터 블록을 나타내고 있으며 그 사이의 블록은 공유된다. (b)는 다음 주기에서 요구 D1이 디스크에서 패치한 블록 X9가 후행 요구를 위해 저장되고, 요구 D2는 이전 주기에서 읽은 블록의 다음을 공유 버퍼에서 읽고 있음을 보

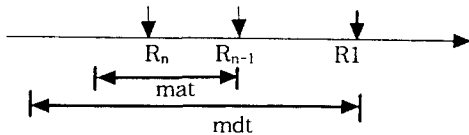
여준다.



<그림 1> 버퍼 공유 기법

3.2 버퍼공유 크기 결정

제안 알고리즘에서는 네 가지 요구 도착 패턴을 고려하여 동일한 연속미디어를 참조하는 사용자 요구 사이의 시간 간격을 고려해서 최대 공유시간 (max distance threshold, mdt)과 최대 도착 시간 (max arrival time, mat)을 기준으로 버퍼 재참조 허용 여부를 결정하고, 그 요구들은 그룹핑한다. 인접한 후행 요구에 대해서는 최대 도착 시간을 고려하여 그룹화 여부를 결정한다. 후행 요구가 동일 그룹에 포함되면 그 사이의 버퍼 크기만큼 증가된 버퍼 공유크기를 구한다.[7]



<그림 2> 버퍼 공유 크기 결정

상황 1) $R_n - R_1 \leq mdt, R_n - R_{n-1} \leq mat$

그림 2에서 짧은 시간간격을 가진 요구의 도착 패턴으로 R_n 이 mdt 와 mat 이내에 도착했기 때문에 R_1 과 동일한 그룹에 포함하고, 버퍼 공유를 허용한다. 이때 공유 버퍼 크기는 R_1 에서 R_n 까지이다.

상황 2) $R_n - R_1 \leq mdt, R_n - R_{n-1} > mat$

두 요구사이의 시간 간격이 mat 보다 클 경우에는 그 사이의 버퍼를 캐싱하여 공유하는 것보다 각각 다른 그룹으로 분리하여 그룹핑하면 보다 효율적으로 제한된 자원을 사용할 수 있다. 그러나 두 요구가 mdt 이내에 있을 때는 계속적으로 도착하는 후행 요구와의 시간 간격을 고려하여 다음 조건에 따라 그룹핑 여부를 결정한다.

<조건> $R_n - R_{n-1} - mat \geq R_{n+1} - R_n$

위의 조건이 만족되면 R_1 에서 R_{n+1} 까지 함께 그룹핑하고, 공유 버퍼의 크기는 R_1 에서 R_{n+1} 까지이다. 만족되지 않으면, R_1 에서 R_{n-1} 까지 그룹핑하며, 공유 버퍼 크기는 R_1 에서 R_{n-1} 까지이다.

상황 3) $R_n - R_1 > mdt, R_n - R_{n-1} \leq mat$

R_n 이 mdt 를 지나서 도착하였지만 그 선행요구와 시간 간격이 짧다. 이것은 짧은 시간 내에 동일 데이터에 대한 요구가 도착하는 경우이다. 그래서 R_n 과 R_1 은 그룹핑되고 버퍼 공유가 허용된다. 이때 공유 버퍼의 크기는 R_1 에서 R_n 까지이다.

상황 4) $R_n - R_1 > mdt, R_n - R_{n-1} > mat$

mat 를 지나서 도착하는 R_n 은 같은 그룹에 포함시키지 않고, 그룹의 크기는 R_{n-1} 이후에 증가된 mat 만큼 감소시킨다. 이때 공유버퍼의 크기는 R_1 에서 R_{n-1} 까지이다.

3.4 그룹 결정 알고리즘

그림 3은 버퍼 공유 그룹의 크기를 결정하는 알고리즘을 보여준다.

```

if( $R_n \leq mdt$ ) then
    if( $R_n - R_{n-1} \leq mat$ ) then 그룹화 // 상황1
    else if( $R_n < limit$ ) then // 상황2
        diskfetch ;
        if( $R_n - R_{n-1} - mat \geq R_{n+1} - R_n$ ) then 그룹화
        else 증가된 버퍼 크기 감소 ;
    else if( $R_n > mdt$ ) then // 상황 3,4
        if( $R_{n-1} < limit$ ) then
            diskfetch ;
            if( $R_{n-1} - R_n \leq mat$ ) then 그룹화 // 상황3
            else 증가된 버퍼크기 감소 ; // 상황4
        if( $R_n > limit$ ) then 그룹초기화 ;
    
```

<그림 3> 그룹 결정 알고리즘

3.3 파라미터

그룹 결정 파라미터인 mat 는 요구사이의 최대 도착 시간을 제한하며, mdt 는 그룹을 결정하는 값이다. 또 $limit$ 는 그룹의 크기가 mdt 를 넘어갈 때 그룹의 최대 크기를 제한하기 위해 사용된다.

사용자 요구에 대해 더이상 제공될 버퍼와 디스크 대역폭 자원이 없으면 사용자 요구는 거절된다. 제안 알고리즘에서 사용되는 파라미터는 다음과 같다.

표 1. 파라미터

파라미터	정의
BUF	전체 버퍼 크기
BW	최대 디스크 대역폭
mat	최대 도착 시간
mdt	최대 공유 결정 시간
limit	최대 그룹 크기 결정 시간
Bi_f	그룹 i의 첫 번째 요구의 버퍼크기
Bi_l	그룹 i의 마지막 요구의 버퍼크기
Rd	디스크 액세스율
DFb	디스크에서 가져오는 버퍼 크기
DFbw	각 요구의 디스크 대역폭 크기
Gi_Buf	그룹당 버퍼 크기
req_num	그룹내 요구수
Ret_n	반환그룹수
Ret_Bi	반환 그룹의 버퍼 크기
round_buf	사용 버퍼 크기
round_bw	사용 디스크 대역폭 크기

3.5 그룹 반환을 이용한 승인제어

각 그룹은 한 주기를 단위로 최초 요구에 의해 디스크에서 버퍼로 데이터가 전송되면서 그룹 크기는 증가된다. 그룹 형성이 완료되면 마지막 요구에 의해 버퍼에서 읽은 블록이 반환되면서 그룹은 축소된다. 이때, 주기 t마다 사용자 요구에 대해 디스크에서 패치되는 버퍼 크기(DFb)는 t * Rd이다.

활용 가능 버퍼의 양을 조정함으로써 전체 버퍼 사용 효율을 향상시키기 위한 그룹 반환은 다음과 같이 이뤄진다.

- ① 각 그룹의 요구수를 비교하여, 오름차순으로 정렬한다.
- ② 요구수에 따라 정렬된 그룹의 버퍼 크기를 비교하여 오름차순으로 정렬한다.

그룹 반환은 최소의 요구수와 최소의 그룹 크기를 가진 것부터 반환하게 되는데, 반환으로 인한 버퍼 공유의 복잡한 관계를 최소로 하기 위해 위와 같은 방법으로 그룹을 선택한다. 이로 인한 버퍼와 그룹수의 변화는 전체 버퍼 사용량과 디스크 대역폭에 영향을 준다.

따라서 동일 그룹내의 요구들은 한번의 디스크 액세스로 서비스되고, 주기당 그룹수의 변화는 사용 디스크 대역폭 크기를 결정한다. 각 그룹의 Bi_f와 Bi_l, 그리고 그룹 생성으로 인해 디스크에서 패치되는 버퍼 크기(DFb), 반환 그룹의 버퍼 크기는 주기마다 변화되는 버퍼량을 결정하는 요인이 된다.

▶ 매주기 버퍼 변화량(BUF/T) :

$$\sum_{i=1}^G B_{i_f} - \sum_{i=1}^G B_{i_l} + DFb - \sum_{i=0}^{Ret_n} Ret_B_i$$

▶ 매주기 디스크 대역폭 변화량(BW/T) :

$$\sum_{i=1}^G G_i - \sum_{i=1}^G \text{소멸그룹} + \sum_{i=1}^G \text{생성그룹} - Ret_n$$

(여기서, n=소멸그룹 수, m=생성그룹 수)

서버는 매 주기마다 기존 서비스를 유지하면서 새로운 요구에 대한 자원 할당과 수용 여부를 결정한다.

$$\text{round_buf} + \text{BUF}/T \leq \text{BUF}$$

$$\text{round_bw} + \text{BW}/T \leq \text{BW}$$

각 사용자 요구의 연속적인 서비스는 요구 도착시 그림 4의 알고리즘에 의해 처리된다.

```

if(요구 도착) then
  if(버퍼 사용이 제한 가용량에 도달) then
    반환 그룹 선택 ;
    그룹 반환 ;
  if(그룹 축소 단계) then
    그룹크기 감소 ;
    전체 버퍼와 대역폭 감소 ;
  if(여유 버퍼와 대역폭으로 요구서비스가능) then
    if(그룹의 최초요구)
      그룹초기화 ;
      disk에서 fetch ;
    else 그룹화 알고리즘 적용 ;
  else no_ admision ;
end
    
```

<그림 4> 요구 도착 처리 알고리즘

IV. 시뮬레이션 결과

4.1 시뮬레이션 환경

기존의 그룹핑 알고리즘과 그룹 반환을 수행한 그룹핑 알고리즘을 비교하기 위해 각 여유 버퍼 크기에 따라 그룹 반환을 적용시키면서 시뮬레이션을 수행하였다. 전송속도 3MB/s인 다수의 디스크로 구성된 저장시스템과 500MB의 버퍼를 이용하고, 0.5MB/s의 데이터 재생율을 처리하는 시스템을 가정하였다.

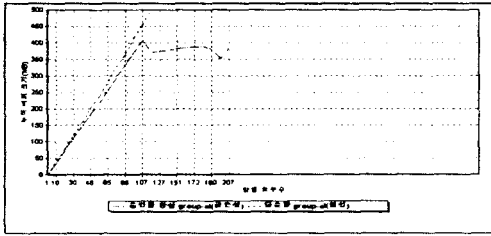
서버의 승인제어기는 가용 버퍼크기의 한계에 이르렀을 때 그룹반환을 수행하고 후행요구를 서비스할 수 있도록 사용자 요청을 받아들인다.

4.2 결과 및 분석

4.2.1 가용 버퍼 사용량의 변화

가용버퍼의 임계치에 이르면 최소 요구수를 가진 그룹을 우선으로 반환한다. 다음으로 최소 공유 크기를 갖는 그룹을 반환한다. 그림 5에서 전체 버퍼 크기의 0.8을 임계치로 두었을 때 버퍼 가용

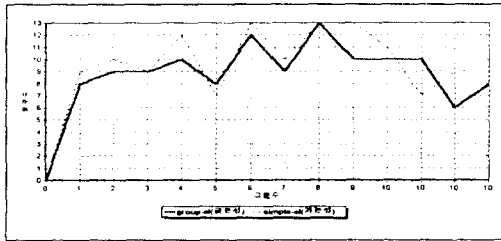
량의 변화를 볼 수 있다.



<그림 5>

4.4.2 그룹당 요구수

그룹 반환을 통해 그룹핑 기법은 이후에 도착하는 요구들을 수용할 수 있으며, 그룹당 요구수는 거의 비슷한 수치를 보인다. 여유자원이 확보되면서 이후의 생성 그룹은 이전과 비슷한 요구수를 갖는 그룹이 생성됨을 알 수 있다.



<그림 6>

V. 결론

연속미디어 처리에서 요구되는 특징을 고려하여 서비스하고 인기 있는 비디오에 대한 동시 서비스 수를 증가시키기 위하여, I/O 대역폭과 버퍼와 같은 자원을 공유하는 기법들은 I/O 자원의 요구를 줄일 수 있다. 그러나 전체적인 자원의 효율성을 감소시키는 고정 크기의 버퍼 공유의 단점을 보완하여 시스템 자원 효율을 향상시키고, 사용자 요구 도착 패턴의 변화를 수용할 수 있는 시스템이 요구된다.

본 논문에서는 요구간의 시간적 지역성을 고려하여 버퍼 공유 크기를 동적으로 결정하고, 요구 그룹핑을 통해 서버가 단일 스트림으로 다중 사용자를 지원한다. 또한 가용 버퍼의 상태에 따라 그룹 버퍼를 반환하고, 여분의 사용자원을 확보함으로써 시스템 안정성을 유도하였다.

기존의 버퍼공유 기법에 비해 첫째, 디스크 대역폭 활용을 향상시킬 수 있다. 이것은 그룹당 요구수의 증가로 인해 전체 그룹수는 감소되고 따라서 디스크 액세스를 줄일 수 있기 때문이다. 둘째, 동

적으로 버퍼 공유 크기를 결정함으로써 요구도착 변화에 잘 적응할 수 있다. 셋째, 가용자원인 버퍼에 사용 임계치를 두어 고장 허용 자원을 확보하고, 각 그룹들 중에서 최소 단위(요구수와 크기)의 그룹을 선택하여 반환함으로써 시스템의 복잡한 처리로 인한 오버헤드를 감소시킬 수 있었다. 앞으로 집중적인 I/O가 발생하는 시스템에서 가변 데이터율을 갖는 사용자 요구를 수용하면서, 각 그룹의 크기 확장과 축소를 고려한 승인 효율성에 대하여 연구한다.

참고 문헌

- [1] A. Dan, D. Dias, R. Mukherjee, D. Sitaram, and R. Tewari. Buffering and Caching in Large Scale Video Servers. In Proc. of COMPCON, 1995.
- [2] M. Kamath, K. Ramamritham, and D. Towsley. Continuous Media Sharing in Multimedia Database Systems. In Proceedings of the 4th International Conference on Database Systems for Advanced Applications, pages 79-86, 1995.
- [3] Weifeng Shi and Shahram Ghandeharizadeh. Trading memory for disk bandwidth in video-on-demand servers, Computer science Technical Report 97-653, USC, June, 1997.
- [4] Vivek S. Pai, Peter Druschel, Willy Zwaenepoel, I/O-Lite: A unified I/O buffering and caching system, Rice University CS Technical Report TR97-294, October 27, 1997
- [5] Renu Tewari, Asit Dan, Dinkar Sitaram and Harrick M. Vin. Multi-level Caching in Bandwidth and Space Constrained Distributed Multimedia Environments, ACM, April 1996.
- [6] Mary Y.Y. Leung, John C.S. Lui, and Leana Golubchik, Buffer and I/O Resource Pre-allocation for Implementing Batching and Buffering Techniques for Video-on-Demand Systems, the Intl. Conference on Data Engineering(ICDE '97), Birmingham UK, April 1997.
- [7] K.S. Park, T.S. Song, and Y.S Kim, Design and Evaluation of the Grouping Scheme for the Decision of Dynamic Buffer Sharing Size. Korean institute of maritime Information and Communication Science, May 1998.