

동적으로 버퍼공유 사이즈를 결정하는 그룹핑 기법의 설계 및 평가

박 규 석, 송 태 섭, 김 연 실
경남대학교 컴퓨터 공학과
yskim@zeus.kyungnam.ac.kr

Design and Evaluation of the Grouping Scheme for the Decision of Dynamic Buffer Sharing Size

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

요 약

동일 데이터를 액세스하는 다수 요구간의 시간적 지역성을 고려하여, 인접한 요구간의 시간 간격(interval)을 버퍼에 캐싱하여 공유하면 서버의 응답시간이 향상되고, 디스크 액세스 감소로 인한 대역폭 활용이 높아지며, 따라서 보다 많은 사용자 서비스가 가능해진다.

본 논문에서는 VOD Server로 들어오는 요구 도착시간에 따라 버퍼 공유 크기를 결정하는 그룹핑(grouping)을 수행하게 하여 인접한 요구들의 공유 데이터 블록이 전체 버퍼를 점유하는 현상을 막고 디스크 액세스율을 감소시켜 대역폭의 활용도를 높일 수 있는 알고리즘을 구축한 후 시뮬레이션을 통하여 그 효율성을 분석한다.

I. 서론

정보 서비스를 위한 분산 멀티미디어 어플리케이션의 응용이 다양하게 확대되고 있으며, 특히 여러 사용자에게 연속미디어(예, 오디오, 비디오 등)를 동시에 전송하는 주문형 비디오(Video-On-Demand) 시스템에 대한 요구가 높아지고 있다. VOD 시스템은 대용량의 데이터, 높은 전송율, 그리고 실시간 제한이라는 특징을 가지고 있기 때문에 연속적인 비디오 전송을 위한 자원(예, 디스크 대역폭, 버퍼)의 할당이 요구된다. 예를 들어, 120분 길이의 MPEG 압축비디오의 경우, 4Mbits/s의

데이터 전송율과 3.6GB의 저장 공간이 필요하다. 개별적으로 사용자에게 자원을 할당하면 좋은 품질의 서비스를 기대할 수 있지만, 한정된 자원(디스크 대역폭과 버퍼)으로 인해 동시에 서비스할 수 있는 사용자 수는 제한된다. 따라서 다수의 요구간의 시간적 지역성을 고려하여 한번 읽혀진 데이터 블록을 다음의 요구를 위해 버퍼에 저장하고, 다수의 요구가 그 버퍼를 공유하도록 함으로써 서버의 적시 데이터 전송 능력을 향상시키고, 동시 서비스 사용자 수를 늘여서 디스크 대역폭 활용도를 높이는 버퍼 공유 기법이 사용되었다.

간단한 버퍼 공유 기법의 경우, 같은 비디오를 요구하는 인접 요구사이의 공유 버퍼가 전체 버퍼 영역을 차지하여 이후에 들어오는 요구들은 현재 서비스되는 요구중에서 하나가 끝날 때까지 대기 상태로 남아 있게 된다. 이때 디스크 대역폭은 휴면 상태에 있게 되므로 버퍼와 디스크 대역폭이 동시에 병목현상을 경험하게 된다.[3]

본 논문은 VOD Server에서 한번 서비스된 데이터를 다음 사용자를 위해 버퍼에 캐싱하고 다수의 사용자 요구를 지원하기 위한 버퍼 공유 크기를 결정하는 그룹핑 알고리즘을 제안한다. 요구의 도착 시간에 따라 동적으로 공유 크기를 결정하고, 버퍼를 공유하는 요구들을 그룹핑함으로써 인접한 블록들이 버퍼를 점유하는 현상을 막는다. 또 그룹핑을 통해 디스크 액세스를 감소시키고, 버퍼 사용 효율을 높인다. 본 논문의 구성은 다음과 같다. 2장에서는 기존의 버퍼 공유 기법에 대해 살펴보고, 3장에서는 본 논문이 제안하는 그룹핑 알고리즘을 설명한다. 4장에서는 시뮬레이션을 통해 성능을 평가하고, 5장에서 결론을 맺는다.

II. 관련연구

VOD 시스템의 연속미디어는 순차적으로 디스크에서 패치되는 특성을 가지므로 한번 액세스한 데이터를 최대한 버퍼에서 공유하면, 디스크 대역폭 부하를 감소시키게 되므로 서비스 받을 수 있는 사용자 수는 증가하게 된다.

2.1 간단한 버퍼공유 기법

한 주기내에서 동일한 소비율을 가진 n 개의 스트림이 버퍼를 공유하여 버퍼와 디스크의 최대 활용도를 높여려고 하였다. 어떤 읽기 시점에서 n 개 스트림이 요구하는 전체 버퍼의 크기는 버퍼 공유를 통해 50%의 사용 효율을 얻을 수 있다.[4] 이 기법은 가변적 데이터율을 갖는 데이터의 특징을 고려하지 않았고 시간성과 동기화 요구를 지원하지 못하기 때문에 연속미디어를 처리하는 시스템에서는 부적합하다.

2.2 Use-And-Toss(UAT)과 SHR Scheme

Use-And-Toss(UAT) 기법은 한번 버퍼에서 사용된 데이터는 반환시키고, 다른 데이터가 이 버퍼영역을 재사용하도록 한다. 데이터는 공유하지 않고, 버퍼 영역만을 공유한다. SHR은 연속미디어 캐싱과 배칭을 사용하여 공유의 효과를 높이는 기법으로 다음 사용자의 재생시간까지 버퍼를 보존

함으로써 버퍼 재사용을 가능하게 하고, 실제 디스크 대역폭이 지원 가능한 수보다 더 많은 요구를 지원한다.[2]

버퍼 공유를 결정하는 값인 최대 가능 interarrival time(max-iat)이 버퍼크기, 데이터율, 평균 도착시간으로 결정되기 때문에, 버퍼 공유의 크기가 너무 크거나 오랜 기간동안 버퍼에 남아 있게 될 경우, 새로운 요구들이 공유할 수 있는 버퍼의 양은 줄어들고 오히려 디스크 액세스율이 증가하는 단점이 있다.

2.3 연속 미디어 캐싱 알고리즘

같은 연속미디어 데이터를 접근하는 연속한 두 요구사이의 시간적 지역성을 나타내는 시간 간격(interval)을 저장하는 interval caching [5]과 distance caching[6] 기법이 있다. 이는 높은 버퍼 히트율을 제공하며, 접근패턴의 동적 변화를 수용하지만, 제한된 크기의 캐쉬를 사용하는 요구수를 최대로 하기 위해 interval 크기가 작은 것부터 서비스를 해주기 때문에 긴 interval을 가진 요구의 응답시간이 상당히 길어지는 경우가 발생한다.

2.4 고정된 공유크기의 버퍼 공유

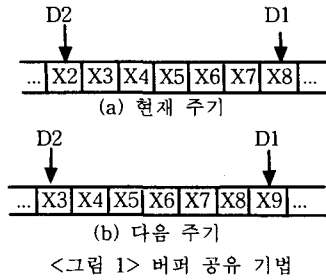
버퍼 공유 크기를 제한한 Weifeng S.과 Shahram G.는 인접한 두 요구간의 간격이 한번 결정된 최대 공유 크기(distance threshold)를 초과하지 않으면 버퍼 공유를 허용한다[3]. 그러나 인기있는 비디오의 경우 인접한 블록들이 버퍼의 대부분 차지하게 되어 제한된 수의 디스크 스트림만 서비스되는 경우가 발생한다.

III. 그룹화 알고리즘

기존의 버퍼 공유 알고리즘이 사용자 요구 도착 패턴의 변화로 인한 캐싱의 효율적 측면을 고려하지 않았다. 그래서 제안 알고리즘은 요구 도착 시간에 따라 동적으로 공유 크기를 결정하고, 요구들을 그룹핑하는 기법을 제시한다.

3.1 버퍼 공유

그림 1의 (a)는 현재 주기에서 두 요구가 읽고 있는 데이터 블록을 나타내고 있으며 그 사이의 블록은 공유된다. (b)는 블록 X9가 요구 D1에 의해 디스크에서 패치되어 공유를 위해 저장되고, 요구 D2는 이전 주기에서 읽은 블록의 다음 블록을 공유된 버퍼를 통해 읽고 있다.



3.2 버퍼공유 크기 결정

제안 알고리즘에서는 네가지 요구 도착 패턴을 고려하여 동일한 연속미디어를 참조하는 사용자 요구에 대해 최대 공유시간(max distance threshold, mdt)을 기준으로 버퍼 재참조 허용 여부를 결정하고, 그 요구들은 그룹핑한다. 최대 공유시간을 지나서 도착하는 인접한 후행 요구에 대해서는 최대 도착 시간(max arrival time, mat)을 고려하여 그룹확장 여부를 결정한다. 후행 요구가 동일 그룹에 포함되면 그 사이의 버퍼 크기만큼 증가된 버퍼 공유크기를 구한다.

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

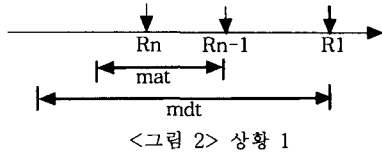


그림 2에서 짧은 시간간격을 가진 요구의 도착 패턴으로 R_n 이 mdt 와 mat 이내에 도착했기 때문에 R_1 과 동일한 그룹에 포함하고, 버퍼 공유를 허용한다. 이때 공유 버퍼 크기는 R_1 에서 R_n 까지이다.

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

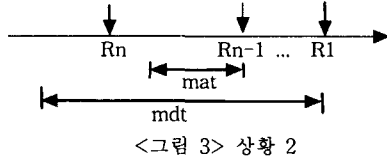


그림 3에서 두 요구사이의 시간 간격이 mat 보다 클 경우는 그 사이의 버퍼를 캐싱하여 공유하는 것보다 각각 다른 그룹으로 분리하여 그룹핑하면 보다 효율적으로 제한된 자원을 사용할 수 있다.

그러나 두 요구가 mdt 이내에 있을 때는 계속적으로 도착하는 후행 요구와의 시간 간격을 고려하여 다음 조건에 따라 그룹핑 여부를 결정한다.

$$\langle \text{조건} \rangle \quad R_n - R_{n-1} - mat \geq R_{n+1} - R_n$$

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

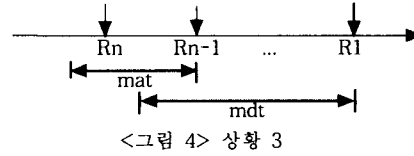


그림 4에서 R_n 이 mdt 를 지나서 도착하였지만 선행요구와 시간 간격이 짧다. 이것은 짧은 시간 내에 동일 데이터에 대한 요구가 도착하는 경우이다.(그림 4) 그래서 R_n 과 R_1 은 그룹핑되고 버퍼 공유가 허용된다. 이때 공유 버퍼의 크기는 R_1 에서 R_n 까지이다.

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

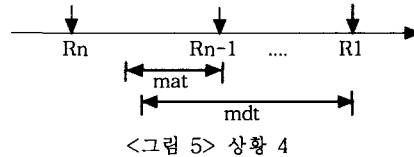


그림 5에서 mat 를 지나서 도착하는 R_n 은 같은 그룹에 포함시키지 않고, 그룹의 크기는 R_{n-1} 에서 mat 만큼 감소시킨다. 이때 공유버퍼의 크기는 R_1 에서 R_{n-1} 까지이다.

3.3 파라미터

그룹 결정 파라미터인 mat 는 요구사이의 최대 도착 시간을 제한하며, mdt 는 그룹을 구분짓는 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
round_buf	사용버퍼의 전체 크기
round_bw	사용 디스크 대역폭의 전체 크기
Rd	디스크 액세스율
DFb	디스크에서 가져오는 버퍼 크기
DFbw	각 요구의 디스크 대역폭 크기
G_Buf	그룹당 버퍼 크기
req_num	그룹내 요구수

3.4 그룹 결정 알고리즘

그림 6은 그룹결정 알고리즘을 보여준다.

```

if(Rn<mdt) then
  if(Rn-Rn-1<=mat) then 그룹화 // 상황1
  else if(Rn+1<limit) then // 상황2
    diskfetch ;
    if(Rn-Rn-1-mat>=Rn+1-Rn) then 그룹화
    else 증가된 버퍼 크기 감소 ;
  else if(Rn>mdt) then // 상황 3,4
    if(Rn+1<limit) then
      diskfetch ;
      if(Rn+1-Rn<=mat) then 그룹화 // 상황3
      else 증가된 버퍼크기 감소 ; // 상황4
    else // 상황 5
      그룹화 ;
    if(Rn>limit) then 그룹초기화 ;
  
```

<그림 6> 그룹 결정 알고리즘

3.5 승인제어

주기마다 각 그룹의 최초 요구에 의해 디스크에서 버퍼로 데이터가 전송되고, 그룹 형성이 완료된 후 마지막 요구에 의해 사용된 버퍼는 반환된다. 이때, 매 주기 t마다 사용자 요구에 대해 디스크에서 패치되는 버퍼 크기(DFb)는 $t * Rd$ 이다. 동일 그룹내의 요구들은 한번의 디스크 액세스를 필요로 하고, 매주기의 그룹수의 변화는 사용 디스크 대역폭 크기를 결정한다. 각 그룹의 최초 요구가 요구하는 버퍼 크기(B_{lf}), 마지막 요구의 버퍼 크기(B_{ll}), 새로운 그룹이 생성되어 디스크에서 패치되는 버퍼 크기(DFb)는 매주기마다 변화되는 버퍼량을 결정하는 요인이 된다.

매주기 버퍼 변화량 :

$$\sum B_i f - \sum B_i l + DFb$$

매주기 디스크 대역폭 변화량 :

$$\sum G_i - \sum \text{소멸그룹} + \sum \text{생성그룹}$$

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

$$\text{round_buf} + \text{버퍼 변화량} \leq \text{BUF}$$

$$\text{round_bw} + \text{디스크 대역폭 변화량} \leq \text{BW}$$

각 사용자 요구를 연속적인 서비스는 요구 도착시 그림 7의 알고리즘에 의해 처리된다.

```

if(요구 도착) then
  if(임의 그룹의 최초 요구가 마지막 블록을 읽음)
    그룹의 마지막 요구 버퍼 반환; // 그룹크기 감소
  if(버퍼와 대역폭 수용 가능) then
    if(그룹의 최초요구)
      그룹초기화 ;
      disk에서 fetch ;
    else 그룹화 알고리즘 적용 ;
  else no_ admision ;
end
  
```

<그림 7> 요구 도착 처리 알고리즘

IV. 시뮬레이션 결과

4.1 시뮬레이션 환경

본 논문이 제안한 그룹핑 알고리즘과 고정 공유 크기를 가진 알고리즘을 비교하기 위해 VOD 원형 시스템과 request-driven 시뮬레이션 모델을 사용하였다.

저장 시스템은 전송 속도 3MB/s를 가진 다수의 디스크로 구성되는데, 디스크 수는 최대 가용 디스크 대역폭을 결정하며, 동시처리 가능한 사용자 수를 제한한다. VOD 서버는 0.5MB/s의 데이터 재생율로 사용자 서비스를 제공하며, 요구 도착 시간에 따라 그룹핑된 요구들을 공유 버퍼를 이용해 서비스하면서 사용자와 상호 동작한다.

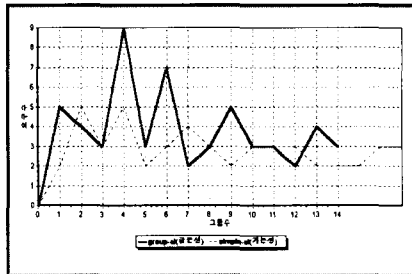
1초의 단위 시간 동안 디스크에서 블록단위의 액세스가 발생하고, 전송 블록의 크기는 500KB, 최대 디스크 대역폭 Rd는 10MB/s로 가정한다.

서버의 승인제어기는 가용 디스크 대역폭을 넘지 않도록 사용자 요청을 받아들이고, 디스크에서 다중 블록을 패치한다.

4.2 결과 및 분석

4.2.1 그룹당 수용 요구수

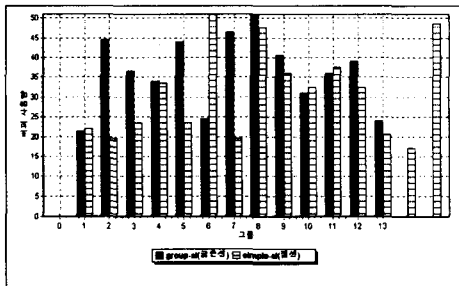
그룹당 포함되는 요구수는 그림 8에서와 같이 그룹의 약 60%가 높은 요구수를 가지는 것을 알 수 있다. 많은 요구가 한 그룹에 포함되면서 기존 알고리즘에 비해 전체 그룹의 수는 감소되어 전체 디스크 대역폭 사용을 줄이게 된다.



<그림 8> 그룹당 요구수

4.4.2 그룹당 사용버퍼 크기

그림 9에서 그룹당 사용 버퍼의 크기는 전체적으로 증가되고 사용자 요구도착 상황에 따라 동적으로 결정된다. 증가된 버퍼는 저장된 데이터를 더 많은 요구들이 오랜 시간동안 서비스 받을 수 있도록 함으로써 캐싱의 효율성을 더욱 향상시킨다.



<그림 9> 그룹당 버퍼 사용 크기

V. 결과

VOD 서버 시스템은 디스크에서 가져온 데이터를 버퍼에 저장하여 공유하도록 함으로써 보다 많은 사용자를 수용하고, 시스템 자원을 효율적으로 사용하려고 한다. 그러나 사용자 요구 도착 시간이 일정하지 않는 상황에서 고정된 버퍼 공유 크기로써 사용자 서비스를 하면 디스크 대역폭과 버퍼의 병목 현상을 피할 수 없게 된다.

본 논문에서는 사용자간의 시간 간격(interval)을 고려하여 앞으로 사용되어질 데이터를 저장하고

요구 도착 상황에 따라 요구들을 그룹핑하고 동적으로 공유버퍼 크기를 결정하는 그룹핑 알고리즘을 제안하였다. 기존의 버퍼공유 기법에 비해 그룹당 요구수는 높고, 전체 그룹수는 감소되므로 더 많은 요구가 디스크 대역폭을 활용할 수 있게 한다. 공유 버퍼의 크기는 전체적으로 약 60% 향상되었고, 동적으로 공유 크기를 결정함으로써 요구도착 변화에 잘 적응하며, 공유 사용자수가 증가되어 버퍼 사용효율을 높였다.

앞으로 MPEG 데이터의 가변 데이터를 처리와 웹을 통한 사용자 서비스에서 발생하는 지연으로 인한 디스크 대역폭의 변경을 고려한 버퍼 공유 기법에 대한 지속적인 연구가 요구된다.

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] R. Ng and J. Yang. Maximizing Buffer and Disk Utilization for News-On-Demand. In Proc. of the 20th VLDB Conference, Santiago, Chile, pp 451-462, 1994.
- [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] B. Ozden, R. Rastogi, and A. Silberschatz. Buffer replacement algorithms for multimedia storage systems. In Proceedings of the International Conference on Multimedia Computing and Systems, 172-180, June 1996.