

멀티미디어 서버를 위한 슬롯 어드레싱 및 재참조 시간 기반 버퍼링 시스템 설계

김진호* · 양종운* · 나인호*

군산대학교

A Design of Buffering System Based on Slot Addressing and Rereference Time for Multimedia Server

Jong-Un Yang* · Jin-ho Kim* · In-Ho Ra*

Kun San National University

E-mail : bedroses@kunsan.ac.kr

요 약

본 논문은 멀티미디어 서버를 위한 버퍼링 시스템 설계를 위한 연구로서 버퍼공유기법과 슬롯 어드레싱(Slot Addressing)기법을 이용하여 제한된 버퍼의 이용률을 극대화함으로써 멀티미디어 서버의 성능을 향상시킬 수 있는 기법을 제안하였다. 또한, 한번 사용된 데이터블록에 대하여 재참조 시간을 계산하고 후속 스트림이 재 사용할 수 있게 하여 멀티미디어 서버의 입출력 횟수를 줄임으로써 결과적으로 여유 대역폭을 이용하여 더 많은 사용자의 요구를 수용할 수 있는 버퍼링 시스템을 제안하였다.

ABSTRACT

This paper is study of buffering system for multimedia server. We propose the scheme that increasing the performance of multimedia sever by using the buffer sharing scheme and slot addressing algorithm to maximize fixed buffer utilization. Also, we calculate rereference time about a used block and next stream reuse that stram. As a result, we propose the buffering system satisfy the users request using the extra bandwidth that reducing multimedia server I/O.

1. 서 론

멀티미디어 데이터와 같은 대용량 데이터의 저장 및 검색이 가능해지고, 네트워크의 발달과 각종 멀티미디어 데이터에 대한 압축기술의 발전은 분산환경에서 멀티미디어 데이터의 처리를 가능하게 하고 있다[1]. 이러한 네트워크와 저장장치의 발전으로 멀티미디어에 대한 응용이 확산됨에 따라 주문형 비디오(VoD : Video on Demand), AoD(Audio on Demand), NoD(News on Demand), 홈쇼핑(Home Shopping), 원격화상교육, 원격화상회의, 원격진료 등이 점차 확산되고 있다[2].

멀티미디어 데이터는 문자나 숫자와 같은 데이터와 오디오, 비디오 같은 데이터가 통합된 데이터이다. 특히, 오디오나 비디오와 같은 멀티미디어 데이터를 지원하기 위해서는 몇 가지 조건을 만족해야 한다. 첫째, 오디오와 비디오 데이터는 지연에 민감한 특성을 가지고 있다. 오디오와 비디오 데이터를

저장하고 재생하는데 있어서 이들 데이터는 연속적인 연산을 수행해야 한다. 즉, 재생을 한번 시작하면 충분한 시스템 자원을 확보해야 하고 실시간 요구사항에 위반되지 않는 연속성을 보장해야한다. 둘째, 저장공간이나 대역폭에 있어서 오디오나 비디오 데이터는 많은 양의 시스템 자원을 필요로 한다. 셋째, 멀티미디어 객체는 오디오나 비디오, 텍스트 등과 같은 다양한 미디어들로 구성이 된다. 이러한 다양한 멀티미디어 데이터들이 검색되는 동안 동기화가 이루어져야 한다[3].

멀티미디어 서버가 사용자들에게 연속매체 데이터를 원활하게 제공하기 위해서는 다음의 요구사항을 만족해야 한다. 첫째, 오디오나 비디오 데이터는 어느 정도의 서비스 품질(QoS : Quality of Service)을 지속적으로 유지해 전달할 때 의미 있는 데이터로서의 가치가 있다. 따라서, 스트림의 재생이 시작되고 난 후 이 스트림은 연속적인 재생을 위하여 충분한 시스템의 자원을 확보, 유지해야된다. 둘째, 멀티미디어 서버는 동시에 다수의 사용자들에게 실시간 스트림 서비스를 제공하기 때문에 데이터의

재생과 전송에 필요한 시스템 자원들에 대한 효율적인 관리기법이 필수적이다[4].

본 논문의 구성은 다음과 같다. 먼저 II장 및 III장에서는 각각 본 연구의 목표 및 버퍼공유 기법의 필요성에 대하여 기술한다. IV장에서는 본 논문에서 제안한 버퍼공유기법과 버퍼 교체기법에 대하여 설명하고 V장에서는 제안된 시스템의 성능 평가결과에 대하여 기술한다. 마지막으로 VI장에서는 결론 및 향후 연구방향에 대하여 기술한다.

II. 설 계 목 표

슬롯 어드레싱 기법[5]의 경우 버퍼를 슬롯(Slot)으로 분할하고 데이터 블록을 조각(Portion)으로 나누어 이 나뉜 조각을 슬롯에 적재하여 효과적으로 버퍼를 공유하는 기법[6]을 제안하여 이상적인 버퍼공유에 근접한 효과를 거두었다. 하지만, 슬롯 어드레싱 기법의 경우 같은 데이터에 대한 접근을 고려하지 않아 같은 데이터 블록의 검색에 필요로 하는 추가적인 대역폭을 고려하지 않았다. 이에 본 논문에서는 슬롯 어드레싱 기법을 기반으로 동일한 멀티미디어 데이터에 대한 접근을 고려한 기법을 제시하고자 한다.

또한, 주문형 비디오와 같이 사용자가 비디오 데이터에 대한 재생 후 고속탐색이나 역탐색과 같은 사용자의 개입이 없는 경우 비디오 데이터는 순차적으로 접근이 이루어지기 때문에 특정 데이터 블록에 대한 참조시점을 예측할 수 있다. 그리고, 하나의 비디오 데이터가 여러 사용자들에 의해 재 참조될 경우 특정 데이터 블록의 재사용 시점도 예측할 수 있다. 따라서, 연속 매체의 재생 연산 중 요구되는 데이터의 전부 또는 일정 부분이 버퍼에 존재할 경우 디스크의 접근 횟수가 감소하여 시스템의 입출력 성능이 향상된다. 제한된 버퍼크기를 고려할 때 비디오 데이터를 모두 버퍼에 유지하기보다는 데이터 블록의 재 참조 가능성을 판단하여 재 참조 가능성이 높은 데이터 블록을 버퍼에 유지할 수 있는 버퍼교체 기법이 필요하다. 버퍼교체 기법은 버퍼의 성능과 밀접한 관련이 있기 때문에 효율적인 버퍼교체 기법을 개발하려는 연구가 선행되었다. 일반적인 기법들은 과거의 참조 유형을 기반으로 버퍼교체의 기준으로 이용하였지만 너무 적은 정보를 이용하여 데이터 블록의 가치를 판단하므로 다양한 참조 유형을 보이는 여러 응용에 효율적으로 대처하지 못하고 있다.

따라서, 본 논문에서는 버퍼공간에 대한 공유 기법과 데이터 블록의 공유 기법을 함께 적용하여 제한된 버퍼의 활용률을 높이고 멀티미디어 서버의 입출력 성능을 개선한 버퍼관리 기법을 제안하고자 한다.

III. 버퍼공유기법

n개의 스트림에 대한 총 버퍼 요구량은 각 스트림 S_i 가 각 사이클 동안 점유하고 있는 B_i 의 버퍼량에 기초한다. 그러나, 스트림 S_i 가 모든 시간에 걸쳐 버퍼 B_i 를 요구하지는 않는다. 거의 모든 경우에 있어서 항상 스트림 S_i 의 버퍼 요구량은 B_i 보다 작다. 이에 따라, 버퍼요구량을 최소화하여 버퍼 이용률을 최대한으로 하는 최적의 방법은 n개의 스트림에 대하여 버퍼를 공유하는 것이라고 말할 수 있다.

1. 버퍼 비공유 시 문제점

소비율 p_i 를 갖는 각 스트림 S_i 에서 압축되지 않은 스트림의 소비율은 재생율과 같다. 모든 스트림의 연속성 요구에 위반되지 않기 위해 n개의 스트림을 위해 S_1, \dots, S_n 은 모든 스트림에 대하여 $\sum_{i=1}^n p_i$ 인 총 소비율 보다 데이터 전송률이 커야한다[7]. 한 주기 안에서 S_i 를 읽기 위해 디스크는 t_i 시간을 소비한다. 스트림 S_i 에서 스트림 S_j 의 탐색/스위칭 시간을 s_{ij} 라 하면

$$t_1 + \dots + t_n + s_{1,2} + s_{n,1} \leq T \quad (1)$$

여기서, T는 주기의 총 길이이다. $s = s_{1,2} + \dots + s_{n,1}$ 이라 하면 동시 발생하는 스트림의 디스크 이용률 δ 는 다음과 같다.

$$\delta = \frac{t_1 + \dots + t_n + s}{T} \quad (2)$$

디스크가 완전히 사용될 때 디스크 이용률은 1이다. 각 스트림이 연속성 요구조건을 만족하려면 시간 T 동안 스트림 S_i 의 소비를 위해 시간 t_i 동안 스트림 S_i 의 충분한 데이터를 읽어와야 한다. 즉,

$$t_i \times R \geq T \times p_i \quad (3)$$

각 스트림의 각 버퍼가 사용된 양을 줄이기 위해 서는 다음 식 (4)를 만족해야 한다.

$$t_i \times R = T \times p_i \quad (4)$$

식 (4)로부터 $t_i/t_j = p_j/p_i$ 임을 알 수 있다. 바꾸어 말하면, 버퍼 소비를 최소화하기 위해서는 각 스트림을 위한 읽기 시간은 각 스트림의 소비율과 비례해야 한다. P를 총 소비율, 즉 $P = p_1 + \dots + p_n$ 이라 하면 t_i 는

$$t_i = (T \times \delta - s) \times \frac{p_i}{P} \quad (5)$$

위의 (5)식과 (3)식으로부터 T의 하한 경계값을 정할 수 있다.

$$T \geq \frac{(s \times R)}{(R \times \delta - P)} \quad (6)$$

데이터 전송률 R이 각 스트림의 소비율 p_i 보다 크면 각 스트림에 버퍼를 필요로 한다. 특히, 버퍼의 최대 크기는 스트림 S_i 의 읽기를 종료하자마자 필요로 한다. 그래서, 스트림 S_i 에 의해 필요로 하는 버퍼의 크기 $B_i = t_i \times R - t_i \times p_i$ 이다.

$$B_i = \frac{p_i \times (R - p_i \times (T \times \delta - s))}{P} \quad (7)$$

따라서, n개의 스트림을 위한 총 버퍼 요구량은

식 (8)과 같다.

$$B = \sum_{i=1}^n B_i \quad (8)$$

만약 B_{max} 가 시스템에서 사용 가능한 최대 버퍼의 수라면 $B < B_{max}$ 이어야 한다. 따라서, 사이클 길이 T 의 상한 경계값을 얻을 수 있다.

$$T \leq \frac{B_{max} * P}{\delta * \sum_{i=1}^n p_i * (R - p_i)} + \frac{s}{\delta} \quad (9)$$

이 수식은 (6)식과 함께 새로운 스트림 S_{n+1} 을 사용할 것인지를 결정할 수 있다. 특히, 수락제어는 S_{n+1} 의 특징을 포함해 (6)식과 (9)식을 이용해 결정한다[6].

2. 버퍼공유의 특징

n 개의 스트림에 의해 필요로 하는 총 버퍼의 양은 $\sum_{i=1}^n B_i$ 이다. 그러나 그림 1에서 알 수 있듯이 스트림 S_i 가 모든 시간에 걸쳐 버퍼 B_i 를 필요로 하는 것은 아니다[8].

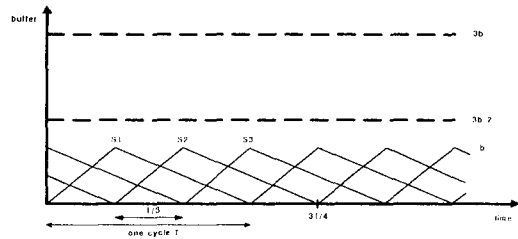


그림 21. 이상적인 소비율을 갖는 스트림의 버퍼 공유

그림 1은 한 사이클 안에 3개의 스트림 S_1, S_2, S_3 이 있고 각 스트림은 같은 소비율을 갖는다고 가정했을 때 버퍼소비를 보여주고 있다. 각 스트림은 같은 읽기 시간 t_i 를 가지고 사이클 길이 T 는 일반적으로 총 스위칭 시간 s 보다 훨씬 크다. $t_i = T/3$ 인 경우에 S_i 이 막 읽기를 끝내고 b 버퍼를 필요로 하는 시점인 $4T/3$ 시간에 필요한 총 버퍼의 요구량을 고려해보면 최대 버퍼량을 항상 필요로 한다. 읽기를 시작하려는 S_2 는 데이터를 모두 소비했다. 그래서 S_2 의 버퍼 요구량은 0이다. S_3 는 시간 T 에 b 버퍼를 요구한다. 그러나 $5T/3$ 시간에서 버퍼에 있는 모든 데이터가 소비되기 때문에 $4T/3$ 시점에서 S_3 는 $b/2$ 버퍼량을 요구한다. 즉, 3개의 모든 스트림에 의해 필요로 하는 버퍼의 총 량은 $b + 0 + b/2 = 3b/2$ 이다. 만약 모든 스트림들이 이상적인 소비율을 갖는다면 시간 t 동안 필요로 하는 총 버퍼 요구량은 변함없을 것이고 3개의 스트림을 위해 모두 $3b/2$ 버퍼가 필요하다. 따라서 버퍼를 공유하면 최대 총 버퍼 소비율의 50%의 절감효과가 있다[6].

3. 슬롯 어드레싱 기법

디스크로부터 읽어온 데이터는 사이클 내에서 읽어온 스트림의 읽기가 끝나기 전에 소비될 것이다. 예를 들면, 네 개의 스트림이 있을 때 각 사이클은 스트림 당 네 개의 읽기 주기로 나뉜다. 따라서, 한 읽기 주기에 소비되는 데이터의 양은 $pT/4$ 가 된다. 여기서 p 는 스트림의 소비율, T 는 사이클의 길이를 의미한다. 또한, 디스크 이용률은 1로 가정하고 스트림 간의 총 스위칭 시간은 읽기 시간과 비교할 때 작다고 가정한다.

일반적으로, n 개의 동시에 발생하는 스트림에 대하여 각 스트림의 pT/n 양은 매 읽기 주기동안 소비된다. 따라서, 한 사이클 동안 주어진 스트림에서 소비될 데이터는 pT/n 크기의 조각으로 나눌 수 있다. 이러한 경우 스트림의 첫 번째 조각은 그 스트림의 읽기 주기동안 소비되며 두 번째 이후의 조각들은 다른 스트림들의 읽기 주기동안에 소비되는 현상을 알 수 있다. 슬롯은 한 개의 스트림 조각을 수용할 수 있는 버퍼공간의 한 블록을 의미한다. 따라서 슬롯의 크기는 pT/n 가 된다.

n 개의 스트림에 대하여 각 읽기 주기 동안에 n 개의 빈 슬롯이 필요하며, 각 읽기 주기가 끝나는 시점에서는 읽은 스트림을 포함하여 n 개의 슬롯이 해제된다. 만약 n 개의 슬롯이 버퍼 공간의 최소량에 추가된다면 현재 읽은 데이터의 스트림은 n 개의 완전히 사용되지 않는 슬롯을 가질 수 있고 현재 읽기 주기가 끝나면 n 개의 슬롯은 해제되므로 다음 읽을 스트림 또한 n 개의 완전히 빈 슬롯을 갖게 된다.

표 1. 3개의 스트림을 위한 슬롯 할당

슬롯	읽기 주기										
	1	2	3	4	5	6	7	8	9	10	11
0	A1	B1	C1	A1	B1	C1	A1	B1	C1	A1	B1
1	A2	C2	B2	B2	A2	A2	C2	C2	B2	B2	A2
2		B2	A2	C2	B2	B2	A2	A2	C2	C2	B2
3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3
4		B3	B3	B3	B3	B3	B3	B3	B3	B3	B3
5			C3	C3	C3	C3	C3	C3	C3	C3	C3

표 1은 동시에 발생하는 세 개의 스트림에 대한 예이다. 표의 열은 읽기 주기를 나타낸다. 예를 들면, 첫 번째 읽기 주기는 스트림 A를 읽는 주기이다. 세 개의 스트림이 있을 때 세 개의 주기로 한 사이클을 구성한다. 표의 각 행은 읽기 주기의 끝에서 슬롯에 저장되는 각 스트림 조각을 나타낸다. 첫 번째 읽기 주기의 끝에서 A의 두 번째와 세 번째 조각(즉, A2와 A3)은 슬롯1과 슬롯3에 있다. 슬롯0은 각 읽기 주기의 끝에서 실제로는 비워지지만 그 주기에서 스트림에 대한 읽기 동작이 멈춘다는 것을 표시하고 있다.

7번째 읽기 주기를 살펴보면 스트림 A는 막 읽기를 끝나치고 슬롯 1, 3, 5를 점유하게 된다. 다음 읽기 주기에서 스트림 B를 할당하기 위해 세 개의 슬롯을 필요로 한다. 따라서 슬롯 0, 2, 4의 사용 가능한 세 개의 슬롯을 이용한다. 8번째 읽기 주기의 끝에서 A2와 C3조각이 소비되는 동안 B의 두 번째와 세 번째 조각은 슬롯2와 슬롯4에 위치한다. 따라서

다음 읽기 주기에 알맞게 슬롯1, 5, 0이 비게 된다. 13번째 주기도 7번째 읽기 주기와 동일한 현상이 발생한다. 따라서 시간상의 어떠한 시점에서 스트림의 특정 조각을 어디에 넣을 것인가를 예측할 수 있다[5].

슬롯 어드레싱 기법의 알고리즘은 그림 2와 같다.

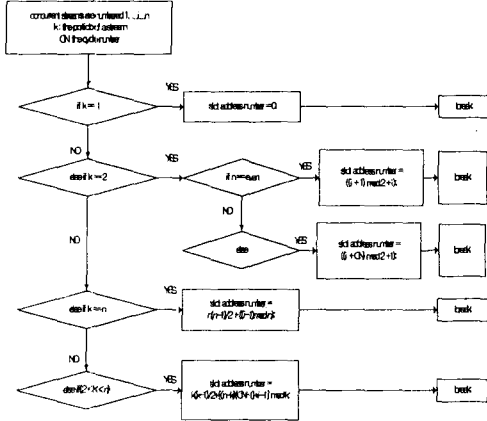


그림 22. 슬롯 어드레싱 알고리즘

IV. 재 참조 시간을 기반으로 한 버퍼 교체 기법

대용량의 실시간 연속 멀티미디어 데이터의 특성과 연속 미디어 데이터의 참조시점 예측이 가능한 특성을 이용하여 디스크로부터 읽어온 연속 미디어 데이터를 스트림 간에 공유함으로써 입출력 성능을 향상시킬 수 있다. 제안된 기법에서는 슬롯에 적재된 데이터의 재사용 가능성을 이용하기 때문에 재사용될 슬롯은 재 사용되지 않을 슬롯보다 큰 가중치를 갖는다. 또한, 동일한 비디오 스트림을 참조하는 스트림들의 참조 시간 간격 정보를 이용하여 재사용 가능성의 근거로 사용한다. 재 참조될 가능성이 높다고 판단된 스트림에 대해서는 최대한 오랫동안 버퍼에 유지해 디스크로부터 직접 데이터를 직접 읽어오는 것에 비해 바로 버퍼에 유지된 데이터를 사용함으로써 디스크의 접근횟수를 줄일 수 있을 뿐만 아니라 디스크 내역폭을 줄일 수 있고 또한 사용자의 평균 대기시간을 크게 줄일 수 있다. 스트림의 재 참조 가능성이 기준값과 비교했을 때 크면 재 참조될 가능성이 없다고 판단하여 버퍼를 재 할당하도록 한다.

스트림의 소비속도를 감안하면 스트림들에 대한 예상참조계획을 산출할 수 있으며 계산된 참조 시간 간격 정보는 다음의 표 2와 같은 스트림 관리 테이블에 기록된다. 스트림 관리 테이블에서 File_id는 스트림들이 참조하는 멀티미디어 데이터에 대한 구분을 하기 위한 파일 ID 정보를 갖는다. Reference_time은 멀티미디어 데이터에 대한 스트림

의 참조시점을 나타내며 Interval 에는 동일한 멀티 미디어 데이터를 참조하는 스트림들 간의 참조 시간 간격 정보를 저장한다.

제안된 기법에서는 스트림 관리 테이블을 이용하여 버퍼를 관리하므로 스트림 관리 테이블에는 서비스 중에 있는 스트림들에 대한 최신 정보가 유지되어야 한다. 참조간격을 변화시킬 수 있는 사건(즉, 새로운 요구의 도착이나 스트리밍 서비스의 종료)이 발생할 때만 스트림들의 참조 시간 간격을 재계산하여 스트림 관리 테이블에 반영해 스트림 관리 테이블 유지에 수반되는 부하를 줄일 수 있다.

이와 같이 재 참조 시간을 기반으로 한 버퍼 교체 기법의 알고리즘은 그림 3과 같다.

표 2. 스트림 관리 테이블

```
static struct Stream
{
    double File_id; //멀티미디어 데이터 설명
    double Reference_time; //참조한 시간
    double Interval; //참조간격
};
|Stream_Manager|USER_REQUEST|;
```

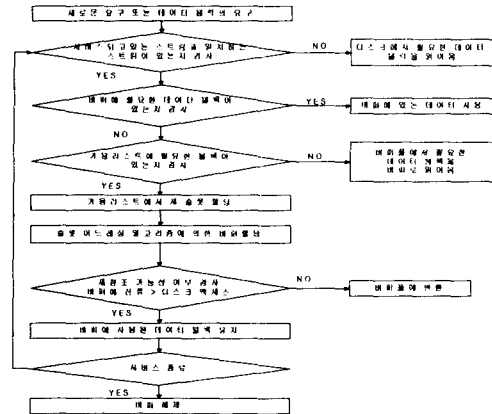


그림 23. 제안된 기법의 흐름도

V. 성능평가

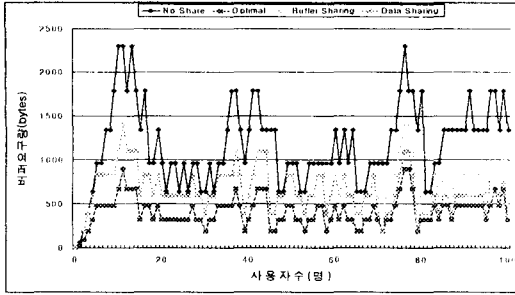


그림 4. 버퍼요구량 비교

그림 4는 100명의 사용자가 임의의 시간간격으로 50개의 비디오 스트림 중 하나의 비디오 스트림을 선택하여 멀티미디어 서버에 대한 서비스 요청이 있을 경우에 대한 시뮬레이션 결과를 나타낸 것이다. 4가지 기법 모두 필요로 하는 버퍼량의 변화는 사용자의 요구에 따라 비슷한 양상으로 변화하고 있음을 알 수 있지만, 버퍼공유기법을 적용했을 경우와 데이터공유를 적용했을 경우 버퍼 공유기법을 적용하지 않았을 경우보다 약 50%정도의 버퍼 절약을 보일 수 있을 뿐 아니라 사용자가 같은 비디오 데이터를 선택했을 경우 디스크 접근 횟수를 감소시켜 디스크 대역폭을 절약할 수 있음을 보이고 있다.

그림 5는 버퍼 공간의 크기를 변화시켰을 때 기존 기법들의 성능변화를 보여준다. 성능평가를 위해 클라이언트들의 스트림 서비스 요청이 멀티미디어 서버 시스템에 도착하는 평균 시간 간격은 비디오 데이터 길이의 절반인 900초로 정하고, 가용 버퍼 수의 범위는 200개 ~ 2000개로 설정하였다.

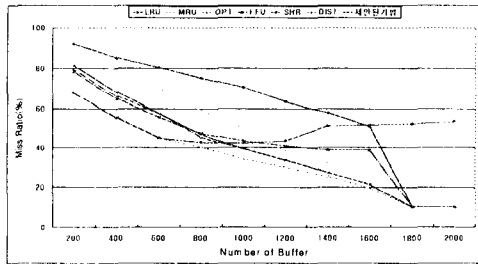


그림 5. 버퍼 공간의 크기 변화에 따른 성능 변화(1)

모든 클라이언트들이 하나의 비디오 데이터를 참조하는 경우에는 선행 스트림이 사용한 데이터 블록이 후속 스트림에 의해 재 사용될 가능성이 높다. 한편, 비디오 데이터의 길이에 비해 버퍼 공간의 크기가 작을 때는 제한된 버퍼 수로써 클라이언트들의 요구를 처리해야하기 때문에 버퍼 공간내의 데이터 교체가 빈번히 발생하고 버퍼 공간이 증가하면 디스크에서 읽어온 데이터를 버퍼공간 내에 더

많이 유지할 수 있기 때문에 캐쉬 부재율은 감소한다. 버퍼 공간의 크기가 비디오 데이터의 길이보다 커지면 비디오 데이터 전체를 버퍼에 유지할 수 있기 때문에 최초 스트림을 위해 디스크로부터 읽어온 데이터를 모두 버퍼 내에 유지할 수 있게 된다.

그림 6은 클라이언트들이 서버에 저장된 10개의 비디오 데이터를 선택해 스트림 서비스를 요청할 때 버퍼 공간의 크기변화가 버퍼 교체 기법들의 성능에 미치는 영향을 나타내고 있다. 선택되는 데이터의 수가 증가하면 버퍼 캐쉬에 유지되어 재 사용되는 가능성이 감소하여 버퍼 수가 작을 때는 상당한 디스크 입출력이 요구된다. 주문형 비디오 시스템과 같은 멀티미디어 응용에서는 클라이언트의 접근 유형이 인기도에 따라 일부 데이터에 편중되기 때문에 데이터 공유를 지원하는 버퍼 교체 기법들이 기존 기법들에 비해 우수한 성능을 보이고 있다.

그림 7은 클라이언트들이 선택할 수 있는 비디오 데이터의 수를 50개로 늘렸을 때 성능변화를 보여주고 있다. 그림 5와 비교해 상대적으로 캐쉬 부재율은 증가하지만 교체 기법들의 캐쉬 부재율 양상은 유사함을 알 수 있다.

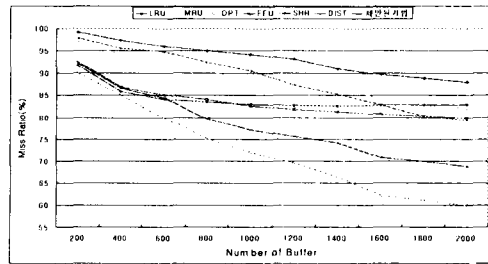


그림 6. 버퍼 공간의 크기 변화에 따른 성능 변화(2)

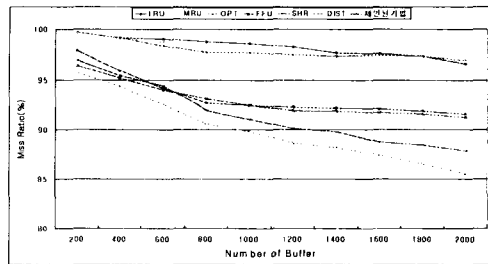


그림 7. 버퍼 공간의 크기 변화에 따른 성능 변화(3)

표 3. 성능평가

비디오 기법	1개	10개	50개
SHR	38.3%	22%	8.1%
DISTANCE	8.9%	19%	12.5%
FFU	25.4%	19.5%	7.9%
제안된 기법	15.3%	10.5%	3.4%

최적기법을 기준으로 각 버퍼 교체 기법들의 성능 수치를 비교해 보면 표 3과 같다.

VI. 결론 및 향후 연구 방향

본 논문에서는 버퍼공유를 위한 슬롯 어드레싱 기법과 선행 스트림이 사용한 데이터 블록을 버퍼에 유지하는 데이터 공유기법을 함께 이용하는 기법을 제안하여 최소의 버퍼를 이용하여 버퍼 활용율을 최대로 하고 또한 디스크의 접근을 최소로 하여 멀티미디어 서버 시스템의 성능을 향상시키는 기법을 제안하였다.

향후, 멀티미디어 서버에서 다양한 환경을 설정해 제안된 기법의 성능이 어떻게 변화하는지를 측정해 버퍼관리의 효율성을 검증해야하고, LAN환경에서 멀티미디어 데이터의 전송에 영향을 주는 변수를 설정하고 성능을 측정해 기존의 연구기법들과의 비교를 통한 분석작업이 필요하다.

이러한 결과를 토대로 주문형 비디오 같은 멀티미디어 응용뿐만 아니라 화상회의, 원격교육 등 다양한 분야에 응용할 수 있음을 확인해야할 것이다.

또한, 본 논문은 재생연산 위주의 주문형 비디오와 같은 멀티미디어 응용을 대상으로 하고 있다. 그러나, 진정한 의미의 주문형 비디오 서비스를 제공하기 위해서는 재생연산 이외에 고속 전진 탐색, 고속 역 탐색 등과 같은 사용자 개입을 허용하는 연산을 지원해야 한다. 따라서, 본 논문에서 제안한 버퍼 관리기법이 주문형 비디오 응용에 제대로 활용되기 위해서는 연속 매체 데이터에 대한 저장 및 재생 기법과의 연동을 위한 연구로 발전되어야 할 것이다.

참 고 문 헌

- [1] R. Arvinal, G. L. Cash, D. L. Duttweiler, H. M. Hang, B. G. Haskell, and A. Puri, "Image and Video Coding Standards," AT&T Technical Journal, pp. 67-89, 1993.
- [2] H. J. Chen and T. D. C. Little, "Physical Storage Organizations for Time-Dependent Multimedia Data," Proceeding of International Conference on Foundations of Data Organization and Algorithms, pp. 19-34, 1993.
- [3] Kun-Lung Wu, Philip S. Yu, "Increasing Multimedia System Throughput with Consumption-Based Buffer Management," Multimedia Systems, pp. 421-428, 1998.
- [4] J. Grimes, and M. Potel, "What is Multimedia?," IEEE Computer Graphics & Applications, pp. 49-52, 1991.
- [5] D. J. Makaroff and Raymond T. Ng, "Buffer Sharing Schemes for Continuous-Media Systems," pp. 1-28, 1996.
- [6] Raymond T. Ng and Jinhai Yang, "An Analysis of Buffer Sharing and Prefetching Techniques for Multimedia Systems," Multimedia Systems, 4: 55-69, 1996.
- [7] J. Gemmell and S. Christodoulaski, "Principles of Delay-Sensitive Multimedia Data Storage and Retrieval," ACM Trans. on Information Systems, Vol. 10, No. 1, pp. 51-90, 1992.
- [8] F. A. Tobagi, J. Pang, R. Baird, and M. Gang, "Streaming RAID™-A Disk Array Management System for Video Files," Proceedings of ACM Multimedia, pp. 393-400, August 1993.