論文
3-5-2

VOD 서버의 초기 대기시간 최소화와 성능 향상을 위한 동적 스트림 합병 기법
(Dynamic Stream Merging Scheme for Reducing the Initial Latency Time and Enhancing the Performance of VOD Servers)

김 근 혜* 최 황 규**
(Geun-Hye Kim) (Hwang-Kyu Choi)

요 약

VOD 시스템의 구성에 있어 핵심 요소라 할 수 있는 VOD 서버는 대용량의 멀티미디어 정보를 저장·관리하며 여러 가입자가 동시에 요구하는 멀티미디어 서비스를 연속적으로 처리하기 위한 대용량 데이터의 실시간 처리 능력을 갖는 컴퓨터 구조를 필요로 한다. 이때 VOD 서버는 가능한 많은 사용자들에게 동시에 실시간 서비스를 지원하기 위해서 정교한 디스크 스케쥴링과 데이터 배치기법이 필요하며, 본 논문에서는 이러한 기법들에서 큰 문제점인 초기 대기시간을 감소시키기 위한 방법으로 동적 스트림 합병 기법을 제안한다. 제안된 기법은 VOD 서버에서 비디오 서비스의 경우 약간의 QoS 변화가 서비스의 질에 큰 영향을 미치지 않는 점을 이용하여 디스크 상의 데이터 배치 기법과 스케쥴링 방법은 기존 방법과 동일하게 하면서, 서비스 요청의 도착시간을 기준으로 일정 시간 내에 들어오는 새로운 서비스에 대하여 재생 속도를 약간 증가시켜 이전의 서비스와 합병시킴으로써 적은 양의 버퍼만으로 초기 대기시간 줄이고 주어진 디스크 능력 하에서 최대한의 사용자를 수용할 수 있도록 한다. 본 논문은 제안된 기법의 성능을 수식과 시뮬레이션을 통하여 분석하며, 그 결과 제안된 기법이 기존의 방법에 비하여 초기 대기시간이 감소됨을 나타내며, 또한 동일한 조건하에서 수용 가능한 사용자 수가 증가되어 성능이 향상됨을 보인다.

ABSTRACT

A VOD server, which is the central component for constructing VOD systems, requires to provide high bandwidth and continuous real-time delivery. It is also necessary to the sophisticated disk scheduling and data placement schemes in VOD servers. One of the most common problem facing in such a system is the high initial latency time to service multiple users concurrently. In this paper, we propose a dynamic stream merging scheme for reducing the initial latency time in VOD servers. The proposed scheme allows clients to merge streams on a request as long as their requests fall within the reasonable time interval. The basic idea behind the dynamic stream merging is to merge multiple streams into one by increasing the frame rate of each stream. In the performance study, the proposed scheme can reduce the initial latency time under the minimum buffer use and also can enhance the performance of the VOD server with respect to the capacity of user admission.

---

* 정회원 : (주) 비씨카드
논문접수 : 2002. 2. 26
* 정회원 : 강원대학교 전기전자정보통신공학부 교수
심사완료 : 2002. 4. 3
※ 본 연구는 강원대학교 BK21 사업단 지원에 의한 연구결과에 일부임.

1. 서 론

최근 저장 장치, 통신, 데이터 압축 기술의 발달로 가능해진 비디오, 오디오, 데이터, 그래픽 등 다양한 정보 표현 형태의 조합인 멀티미디어 응용 분야가 인터넷의 활용 증대에 따라 급속히 확산되고 있다. 특히 여러 종류의 멀티미디어 응용 분야 중 VOD(Video On Demand) 서비스는 기존의 TV나 유선 방송을 대체할 수 있는 획기적인 서비스 분야로 이에 대한 많은 연구가 이루어지고 있으며 여러 상용 제품들이 출현하여 널리 활용되고 있다[3][8]. VOD 서비스 사용자는 자신이 원하는 시간에 원하는 멀티미디어 정보를 인터넷이나 전용 통신망을 통하여 받아 보게 된다.

이를 위한 VOD 시스템의 구성에 있어 핵심 요소라 할 수 있는 VOD 서버는 대용량의 멀티미디어 정보를 저장·관리하며 여러 가입자가 동시에 요구하는 멀티미디어 서비스를 연속적으로 처리하기 위한 대용량 데이터의 실시간 처리 능력을 갖는 컴퓨터 구조를 필요로 한다[1][3][9]. 즉, VOD 서버는 동영상을 포함한 대용량의 멀티미디어 데이터를 실시간으로 검색하여 각각의 세그먼트들이 끊김 없이 사용자의 디스플레이 장치에 도착하도록 해야 하는 실시간 특성을 보장해야 한다. 이러한 VOD 서버에서 가능한 많은 사용자들에게 동시에 실시간 서비스를 지원하기 위해서는 보다 정교한 디스크 스케줄링과 데이터 배치기법이 필요하다[1-3][9-13]. 이들 기법 중에서 제약 분할 할당(constrained and partitioned allocation)과 라운드로빈 스케줄링의 조합된 기법은 간단하면서도 이러한 조건을 비교적 잘 만족한다[7].

그런데 이 기법에서 가장 큰 문제점은 초기 대기시간(initial latency time)이 매우 길다는 것이다. 여기서 초기 대기시간이란 서버에 새로운 요구가 도착해서 첫번째 데이터 세그먼트가 서버의 메모리로 이용 가능하게 적재되는데 걸리는 시간을 의미한다. 초기 대기시간의 증가는 비교적 짧은 상영시간을 갖는 비디오 클립이나 비디오 게임 같은 실시간 대화형 서비스에서 큰 문제가 될 수 있다. 이러한 초기 대기시간 문제를 해결하기 위한 주된 연구로는 데이터 배치 기법이나 디스크 스케줄링 기법을 변화시키는 방법[7][9]과 시작 세그먼트들을 별도의 디스크나 메모리에 중복 저장하는 방법 등이 있다[4][5]. 그러나 이 방법들도 초기 대기시간을 줄이기 위해 많은 양의 버퍼를 요구하는 등의 문제점을 가지고 있다.

본 논문은 VOD 서버에서 비디오 서비스의 경우 약간의 QoS 변화가 서비스의 질에 큰 영향을 미치지 않는 점을 이용하여 짧은 초기 대기시간을 유지하면서도 적은 양의 버퍼를 필요로 하는 동적 스트림 합병 기법을 제안한다. 제안된 기법은 디스크 상의 데이터 배치 기법과 스케줄링 방법은 기존 방법과 동일하게 하면서, 서비스 요청의 도착시간을 기준으로 일정 시간 내에 들어오는 새로운 서비스에 대하여 재생 속도를 약간 증가시켜 이전의 서비스와 합병시킴으로써 적은 양의 버퍼만으로 초기 대기시간 줄이고 주어진 디스크 능력 하에서 최대한의 사용자를 수용할 수 있도록 한다. 본 논문은 제안된 기법의 성능을 수식과 시뮬레이션을 통하여 분석하며, 그 결과 제안된 기법이 기존의 방법에 비하여 초기 대기시간이 감소됨을 나타내며, 또한 동일한 조건하에서 수용 가능한 사용자 수가 증가되어 성능이 향상됨을 보인다.

본 논문의 구성은 서론에 이어 2장에서 VOD 서버와 관련된 최근 연구들에 대하여 기술하며, 3장에서는 본 논문에서 제안하는 기법에 대하여 설명한다. 4장에서는 제안된 기법의 성능 분석에 대해 기술하고, 마지막으로 5장의 결론으로 맺는다.

2. 관련 연구

VOD 서버에서 가능한 많은 사용자들에게 동시에 실시간 서비스를 지원하기 위한 디스크 스케줄링과 데이터 배치 기법들 중에서 비교적 우수한 성능을 보이는 제약된 분할 할당(constrained and partitioned allocation) 기법[4][5][7]은 초기 대기시간이 길다는 문제점을 가지고 있다. 이 초기 대기시간 문제를 해결하기 위한 지금까지의 연구는 다음과 같으며, 이들 기법들을 설명하는데 사용되는 파라미터들은 <표 1>과 같다.

<표 1> 파라미터
<Table 1> Parameters

파라미터	설 명
N	처리량
Mem	총 메모리 크기, bytes
TR	전송률, bytes/s
S	세그먼트 크기, bytes
R	나눠진 영역 수
Cyl	실린더 수
r(d)	d 트랙에 대한 탐색 오버헤드 계산 함수
T	요구의 한 주기 서비스 시간
Xi	i 번째 스트림
Xi,j	i 번째 스트림의 j 번째 세그먼트

2.1 제약 분할 할당 기법 (Constrained and Partitioned Allocation scheme)

디스크 저장장치의 검색에서 가장 시간을 많이 차지하는 부분은 디스크의 탐색시간(seek time)으로 세그먼트 배치와 디스크 스케쥴링 등의 문제를 고려할 때 가장 중요한 요소이다. 제약된 분할 할당 기법은 이 요소를 줄이기 위한 기법으로 디스크에서 세그먼트 간의 거리를 일정 트랙 수 이내로 제한하여 저장함으로써 검색 시간을 줄이는 방법이다[7]. 즉, 스트림을 n개의 세그먼트들로 나눈 후 연속적인 세그먼트들은 d 트랙보다 적은 거리로 배치하여 저장함으로써 탐색거리를 최대 d 트랙으로 감소하는 기법이다. 예를 들어, 스트림 X_i을 n개의 $X_{i,1}$, $X_{i,2}$, $X_{i,3}$, $\cdots$, $X_{i,n}$ 세그먼트로 나눈 후 연속적인 세그먼트들은 d 트랙 보다 적은 거리로 배치하여 저장한다.

$$|Track(X_{i,j+1}) - Track(X_{i,j})| \le d$$

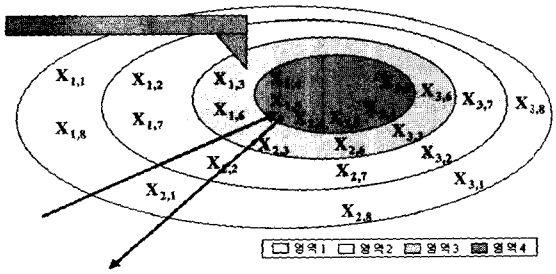
제약된 분할 할당 기법은 탐색 거리를 디스크의 반지름에서 최대 d 트랙으로 감소하여 디스크의 탐색 시간을 크게 줄였다. 그러나 동시에 여러 스트림을 탐색하는 시간은 제한하지 않았기 때문에 동시에 여러 스트림을 탐색할 경우 디스크의 탐색 시간을 알 수 없다. 예를 들어, 두 개의 스트림 X_i, X_{i+1}이 있다고 가정해 보자. 같은 스트림 X_i의 연속된 두 세그먼트 $X_{i,1}$, $X_{i,2}$의 탐색 거리는 d 트랙 보다 작다. 그러나, 다른 스트림에 속해 있는 두 세그먼트 $X_{i,1}$와 $X_{i+1,1}$의 탐색 거리는 제한하지 않았으므로 최악의 경우 디스크의 끝과 끝에 저장되어 있을 수 있다. 이것은 멀티미디어 데이터의 연속 재생을 불가능하게 할뿐만 아니라 만약 이를 가능하게 하기 위해서는 경우에 따라 매우 큰 초기 대기시간을 요구한다. 따라서, 제약 분할 할당 기법은 동시에 여러 사용자들을 서비스 해야하는 VOD 서버에서는 적합하지 않은 데이터 배치 기법이다.

2.2 양방향 배치 기법 (Bidirectional Scheme)

양방향 배치 기법은 하나의 스트림을 n개의 세그먼트들로 나누어 R개의 동등한 트랙들의 영역으로 분할된 디스크에 저장하는 기법으로 연속적인 세그먼트들은 인접 영역에 배치된다. 이때 연속된 세그먼트의 최대 탐색 거리는 2/R이 된다. 동시에 여러 사용자의 요청이 들어오면, 디스크 헤드가 각 영역에서 요구하는 모든 세그먼트들을 추출하여 서비스하므로 서로 다른 스트림의 세그먼트간 탐색거리는 Cyl/R이다.

[그림 1]은 3개의 스트림을 저장하고 있는 디스크를 4개의 영역으로 분할하였을 때의 양방향 배치 기법을 보인다. [그림 1]에서 $X_{1,1}$ 세그먼트는 영역 1에 저장되고, 다음 세그먼트들은 영역 2에서 영역 4까지 저장된 후, 반대 방향으로 영역 4에서 영역 1까지 저장된다. 모든 스트림들의 세그먼트들도 이와 같이 배치한다. 예를 들어, 동시에 스트림 X_1과 X_2 요구가 들어왔다면 영역 1에서 $X_{1,1}$을 추출한 후 $X_{2,1}$을 추출하여 서비스하고, 그 다음 영역 2에서 $X_{1,2}$와 $X_{2,2}$을 추출하여 서비스한다. 그 이후의 세그먼트들도 디스크의 헤드가 영역 4에 도

달할 때까지 같은 방법으로 서비스된다. 디스크 헤드가 영역 4에 도달하면, 디스크 헤드가 움직이기 전에 한 번 더 서비스한다. 따라서 이 기법은 먼저 들어온 요구가 영역 2를 서비스하고 있을 때 새로운 요구가 들어오면, 디스크 헤드가 영역 4에 도달해서 다시 영역 4에서 영역 1로 되돌아 올 때까지의 초기 지연 시간을 갖는다.

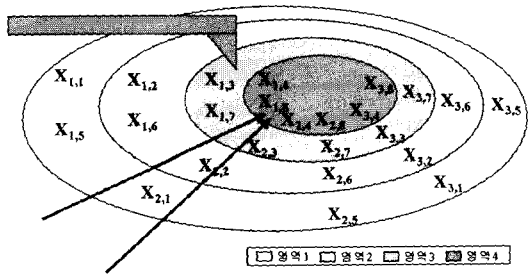


[그림 1] 양방향 배치 기법
[Fig. 1] Bidirectional Scheme

2.3 단방향 배치 기법(Unidirectional Scheme)

단방향 배치 기법은 양방향 배치 기법과 유사한 방법이지만 세그먼트들을 가장 바깥쪽의 영역으로부터 가장 안쪽의 영역으로 위치시켜 양방향 배치 기법의 초기 대기시간을 반으로 감소시킬 수 있는 방법이다. [그림 2]는 3개의 스트림을 저장하고 있는 디스크를 4개의 영역으로 분할하였을 때의 단방향 배치 기법을 보인다. [그림 2]의 예에서 세그먼트 $X_{1,1}$부터 $X_{1,4}$까지는 양방향 배치 기법과 동일하게 저장하고, 다시 영역 1부터 세그먼트 $X_{1,4}$ 이후의 세그먼트들을 저장한다. 즉, 디스크 헤드는 영역 1에서부터 영역 4까지의 한 방향으로만 움직이면서 세그먼트들을 읽어온다.

이 기법의 초기 지연 시간은 먼저 들어온 요구가 영역 2를 서비스하고 있을 때 새로운 요구가 들어올 때 가장 크며, 디스크의 헤드가 영역 4에 도달한 후 바로 영역 1로 돌아오므로 양방향 배치 기법의 초기 지연 시간의 반에 해당한다. 그러나, 여전히 큰 초기 지연 시간을 갖고 있다.



[그림 2] 단방향 배치 기법
[Fig. 2] Unidirectional Scheme

2.4 순차적 기법(Sequential Scheme)

순차적 기법은 단방향 배치 기법을 사용하면서 동시에 여러 스트림의 세그먼트 검색 시에 순차적으로 요구된 세그먼트들을 추출하는 기법이다. 예를 들어, 동시에 스트림 X_1, X_2, X_3를 순차적으로 서비스한다고 가정할 때, 같은 영역에 있는 세 개의 세그먼트 $X_{1,3}, X_{2,3}, X_{3,3}$가 $X_{3,3}, X_{2,3}, X_{1,3}$ 순서로 저장되어 있다면 순차적 기법은 세그먼트의 저장 순서대로 읽어 버퍼에 저장한 후 $X_{1,3}, X_{2,3}, X_{3,3}$ 순서로 서비스한다. 따라서, 디스크 헤드가 순차적으로 세그먼트들을 추출하여 탐색 거리는 Cyl/(R * N)으로 감소하나 요구하는 버퍼의 양은 2 * N * S로 증가한다.

2.5 복사 기법(Replication Schemes)

복사 기법은 스트림의 초기 일부 세그먼트를 별도의 저장장소에 복사본을 둠으로써 초기 대기시간 없이 즉시 서비스 할 수 있도록 하는 기법이다. 복제 기법들은 기본적으로 단방향 배치 기법과 순차적 기법을 따르며 메모리에 복사본을 두는 기법과 디스크에 복사본을 두는 기법이 있다[4][5].

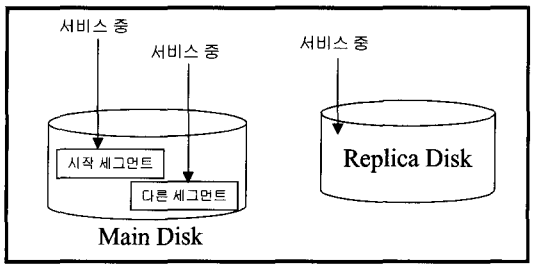
1) 메모리에 복사본을 두는 기법

디스크에 저장되어 있는 스트림들의 초기 일부 세그먼트들을 메모리에 복사하여 저장해 두는 기법으로 초기 대기시간은 초기 세그먼트가 메모리에 적재되는데 걸리는 시간을 의미하므로 초기 지연시간은 0으로 줄어든다.

그러나 복사본을 메모리에 저장해 두기 때문에 메모리 비용이 늘어난다. 더구나 복제해 두는 세그먼트의 양은 데이터 배치 기법에 따라 크게 달라지는데 단방향 배치 기법의 경우 현재 디스크 헤드가 접근하고 있는 영역에 있는 세그먼트 까지 복제해둔 세그먼트를 읽어오기 때문에 초기 세그먼트부터 나는 영역 수만큼의 세그먼트까지 메모리에 저장되어 있어야 한다. 따라서 메모리가 부가적으로 요구된다.

2) 디스크에 복사본을 두는 기법

스트림의 초기 일부분을 메모리가 아닌 별도의 디스크에 복사해 두는 기법으로 [그림 3]과 같이 시작 세그먼트를 서비스하고 있는 중 새로운 요구가 들어왔을 때 기다리지 않고 다른 디스크에 복사해둔 시작 세그먼트를 서비스함으로써 초기 대기 시간을 줄인다. 그러나 새로운 사용자 요구들이 시간의 차이를 두고 연속적으로 들어오면 초기 대기시간을 줄이기 위해 많은 양의 세그먼트를 버퍼로 읽어 들인 후 서비스가 끝날 때까지 이 세그먼트들을 버퍼에 유지시켜야하며 이를 위한 많은 양의 버퍼를 필요로 한다.



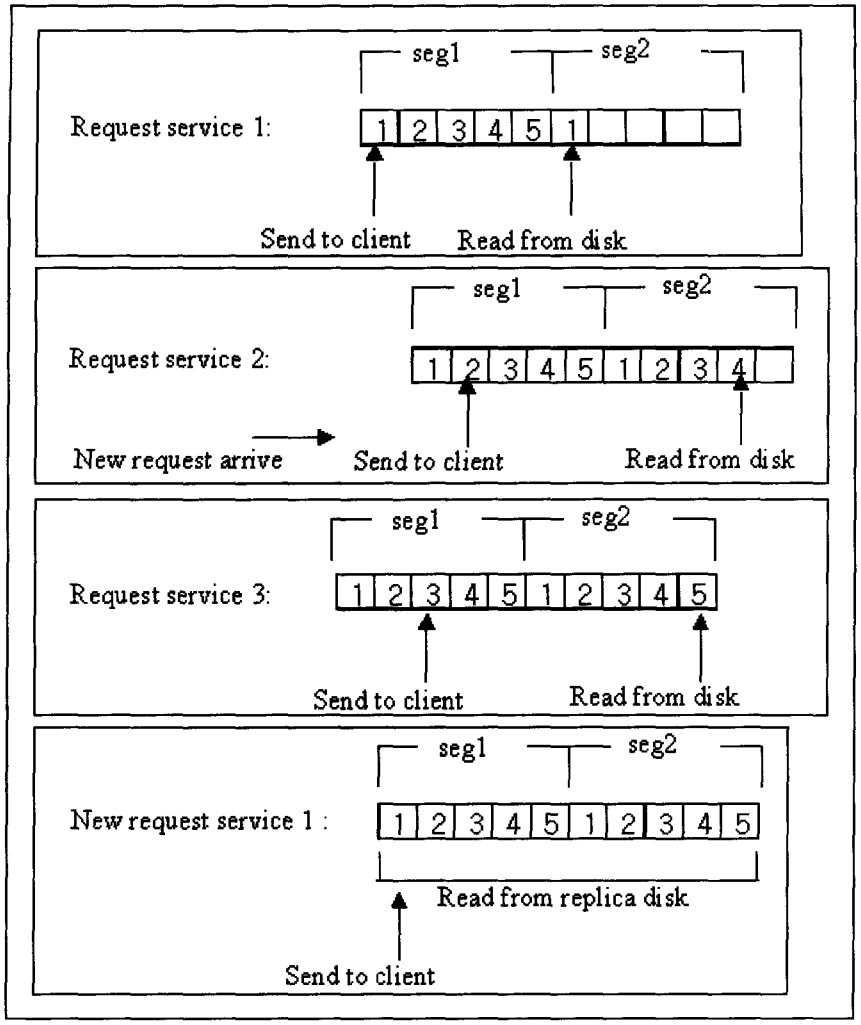
[그림 3] 디스크 복사 기법

[Fig. 3] Disk Replication Scheme

3. 동적 스트림 합병 기법

앞에서 살펴본 바와 같이 VOD 서버에서 보다 많은 사용자들에게 실시간 서비스를 보장하기 위한 기존의 데이터 배치 기법들은 큰 초기 대기시간을 갖거나 많은 양의 버퍼를 사용한다는 단점을 가지고 있다. 본 논문에서는 이러한 단점을 개선하여 적은 양의 버퍼를 사용하여 초기 대기시간을 최소화함과 동시에 사용자의 수용 능력을 최대화 할 수 있는 새로운 동적 스트림 합병 기법(Dynamic Stream Merging Scheme)을 제안한다. 제안된 동적 스트림 합병 기법은 멀티미디어 서비스, 특히 비디오 서비스의 경우 약간의 QoS 변화가 서비스의 질에 큰 영향을 미치지 않는 점을 이용하여 시간적으로 서로 인접한 일부 스트림을 점차적으로 합병하여 서비스함으로써 적은 초기 지연시간의 유지에 필요한 버퍼의 양을 줄이고 주어진 디스크 능력하에서 최대한의 사용자를 수용할 수 있도록 한다.

동적 스트림 합병 기법은 기본적인 디스크 스케줄링과 데이터 배치기법으로 단방향 배치 기법과 순차적 기법을 사용하며 초기 대기시간의 감소를 위하여 디스크 복사 기법을 채택한다. 그러나 제안된 기법은 디스크 스케줄링 과정에서 기존의 서비스에 대하여 일정 시간 내에 들어온 새로운 서비스 요구에 대하여 재생 속도를 약간 증가시켜 기존의 서비스와 합병시켜 나간다. 이때 새로운 서비스 요구에 대한 재생 속도의 증가를 위해서는 먼저 디스크 복사 기법에 따라 초기 세그먼트들이 복사된 디스크로부터 기존 사용자 서비스에 대해 현재 진행중인 디스크 영역까지에 해당하는 새로운 서비스의 초기 세그먼트를 먼저 버퍼에 읽어들인 후 기존의 정상적인 재생속도에 비하여 증가되는 비율에 따라 버퍼내의 일부 프레임을 건너뛰어 서비스함으로써 주어진 시간 내에 같은 디스크 영역의 기존 서비스와 합병시켜 나간다. 이 합병 처리 과정은 [그림 4]에서 예를 들어 차례대로 설명한다.



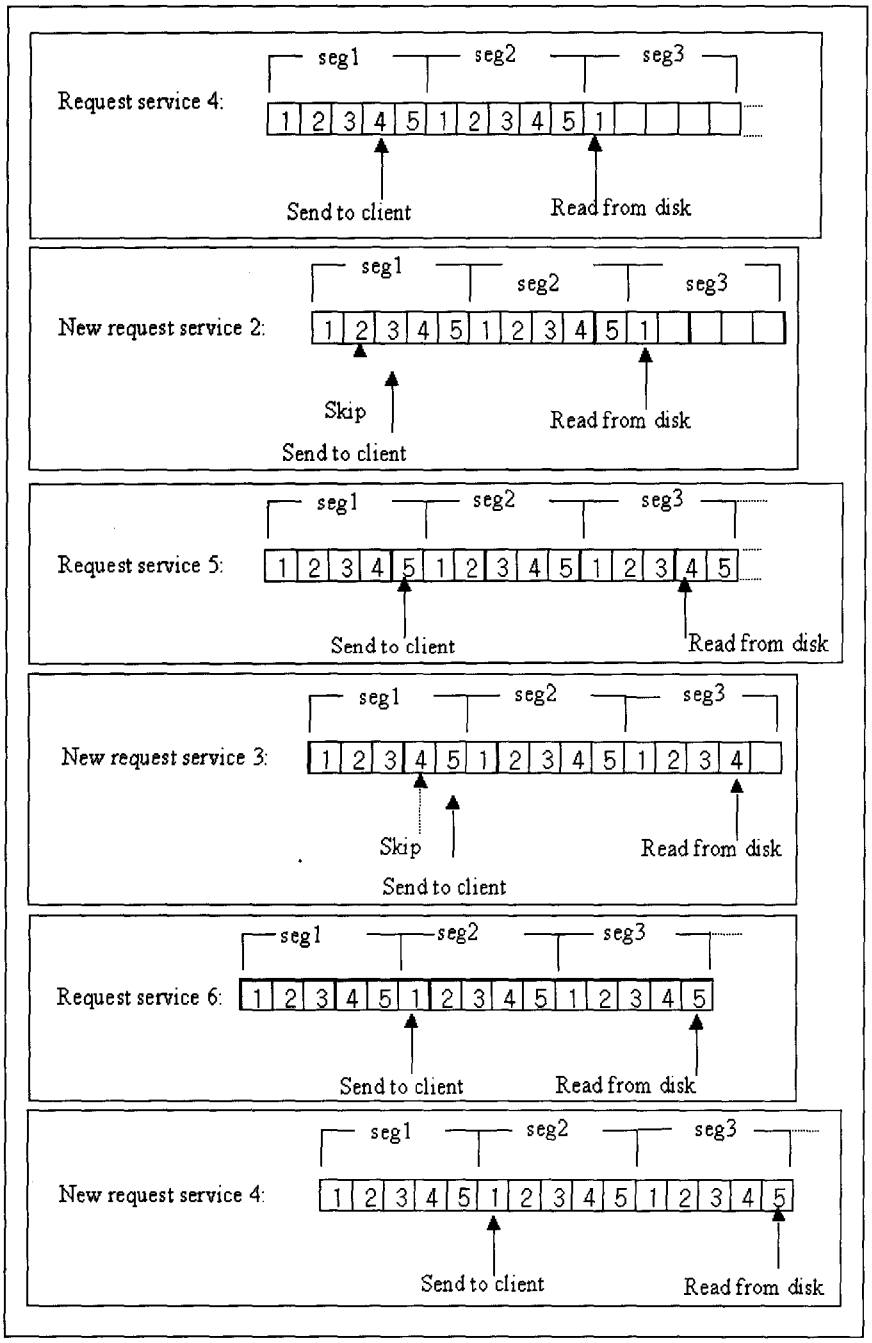
[그림 4] 동적 스트림 합병 기법의 처리 과정 예

[Fig. 4] An example of the dynamic stream merging scheme

[그림 4]의 예는 한 세그먼트가 5개의 프레임으로 구성되어 있고 기존의 스트림과 합병을 위하여 재생 속도를 2배로 증가하는 경우이다.

[그림 4]에서 요구 서비스 1(Request service 1)은 기존의 사용자 서비스를 위해 세그먼트 1을 읽어 버퍼에 저장해 놓은 후 디스크 헤드가 세그먼트 2가 저장되어 있는 영역 2를 읽기 시작하는 것을 나타낸다.

디스크 헤드가 영역 2를 읽고 있는 중 새로운 사용자 요구가 들어오는 것을 요구 서비스 2(Request service 2)에서 보인다. 기존 사용자 서비스는 요구 서비스 3(Request service 3)에 나타낸 바와 같이 새로운 요구가 들어오는 것과 관계없이 세그먼트의 내용을 버퍼로 읽어 들인다.



[그림 4] 동적 스트림 합병 기법의 처리 과정 예(계속)

[Fig. 4] An example of the stream merging scheme (continued)

새로운 요구 서비스 1(New request service 1)은 디스크 헤드가 영역 1이 아닌 곳에 있기 때문에 시작 세그먼트를 복사해 놓은 다른 디스크로부터 디스크 헤드와의 영역 차이만큼을 미리 읽어 들인다. 이 때부터 디스크 헤드는 기존의 사용자 서비스에 대한 세그먼트를 읽을 때 같은 영역에 존재하는 새로운 사용자 요구에 대한 세그먼트를 순차적 기법으로 읽어 들여야 하며, 요구 서비스 4(Request service 4)와 새로운 요구 서비스 2(New request service 2)는 이 과정을 나타낸 것이다.

새로 들어온 사용자 서비스를 위해 버퍼로 읽어 들인 세그먼트의 내용은 증가된 재생 속도를 적용하여 서비스된다. 증가된 재생 속도의 적용은 새로 들어온 사용자 요구를 위해 사용되는 버퍼량이 최소 버퍼 요구양인 2*(세그먼트 크기)가 되면 더 이상 적용하지 않는다. 즉, 이때부터는 정상적인 디스크 영역에서 읽혀진 세그먼트에 의하여 정상적인 속도로 재생된다. 이 과정은 [그림 4]의 새로운 요구 서비스 4(New request service 4)에 해당한다.

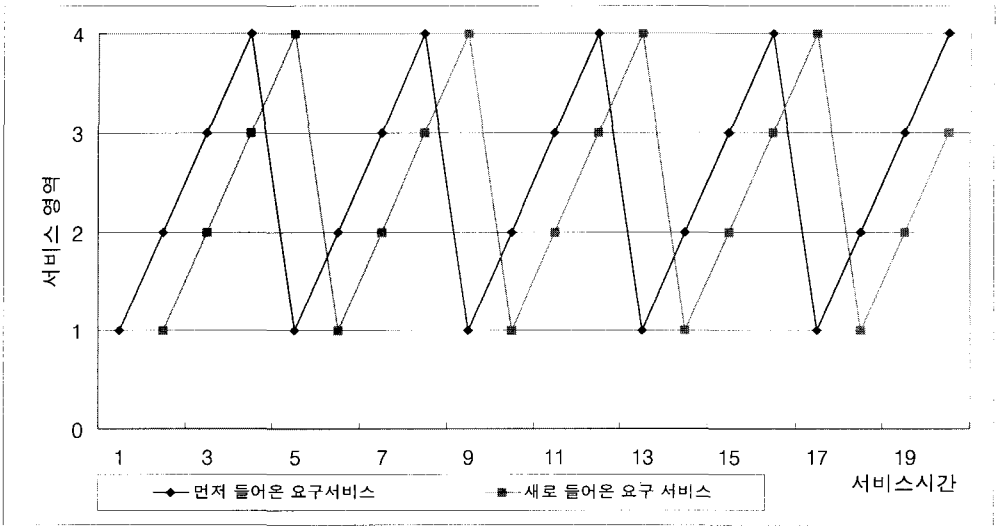
```
ArrivalRequest()  // 새로운 사용자 요구 도착 처리 과정
{
   if (디스크의 헤더 접근 영역 != 영역1) // 복제된 초기 세그먼트 읽기
   {
      현재 디스크 헤더 접근 영역과 영역 1의 영역 차이 계산;
      영역 차이 수만큼 복사해 둔 세그먼트들을 읽어 버퍼에 저장;
      재생 속도 설정;  // 서비스의 질에 영향을 미침
   }
   else     // 시작 세그먼트를 가져옴
   {
      현재 디스크 헤더 접근 영역에서 요구한 스트림의
      세그먼트를 읽어 버퍼에 저장;
   }
}

Service()      // 서비스 처리 과정
{
   if (재생 속도 설정 == true)
   {
      설정되어 있는 재생 속도에 따라 일부 프레임은
         서비스하지 않고 건너뜀;
      if (사용하고 있는 버퍼의 양 <= 2*세그먼트 크기)
            // 최소 버퍼 요구량은 2*세그먼트 크기
         재생 속도 = 정속;
   }
   else
      정속 서비스;
}
```

[그림 5] 동적 스트림 합병 알고리즘

[Fig. 5] The dynamic stream merging algorithm

즉, 제안하는 동적 스트림 합병 기법은 기존에 들어온 사용자 요구를 서비스하고 있는 도중 새로운 요구가 도착했을 때, 디스크 헤드가 새로운 요구의 시작 세그먼트를 읽어 들이지 못하는 영역에 있다면 시작 세그먼트를 복사해 둔 디스크에서 일정량의 시작 세그먼트 복사본을 읽어온다. 여기서 필요로 하는 버퍼의 양은 기존 요구와의 영역차이 만큼에 해당하는 시작 세그먼트 크기와 같으며, 이때부터 원래의 데이터가 있는 디스크의 헤드는 기존의 요구를 위한 세그먼트를 읽어 들인 후 같은 영역에 있는 새로운 요구를 위한 세그먼트를 읽어 들인다. 새로운 요구는 버퍼의 사용량이 최소가 될 때까지 일정 비율로 프레임을 삭제하여 서비스함으로써 버퍼의 사용량을 줄일 수 있다. [그림 5]는 지금까지의 과정을 새로운 서비스 요구의 도착을 처리하는 과정과 정상적인 재생 처리 과정으로 나누어 알고리즘으로 정리한 것이다.

기존의 기법과 동적 스트림 합병 기법의 처리 과정의 비교를 위하여 [그림 6]에서는 기존의 기법을 적용한 서비스 과정을 나타낸 것이다. 이 예에서는 4개의 영역으로 나누어져 있는 디스크에 두 개의 서비스 요구가 1개의 영역 차이를 두고 들어온 경우를 보이는 것으로 서비스가 끝날 때까지 서비스 영역 차이를 유지한다. 이는 곧 초기에 사용하는 버퍼의 요구량을 서비스가 끝날 때까지 유지하는 것을 말한다. 여기서 x축은 서비스 단위 시간을 의미하고, y축은 요구하는 세그먼트가 저장되어 있는 영역을 의미한다. 즉, 새로 들어온 요구는 먼저 들어온 요구와 다른 영역을 요구하므로 복사한 세그먼트를 읽어들여 우선 서비스하고, 디스크 헤드가 먼저 들어온 요구에 대한 서비스가 다음 영역인 영역3의 세그먼트를 읽을 때 새로 들어온 요구를 위한 세그먼트를 같이 읽어 들이므로 3*(세그먼트 크기) 만큼의 버퍼를 사용하게 되고 이 버퍼를 서비스가 끝날 때까지 유지하고 있어야 한다.



[그림 6] 기존 기법에서의 재생 처리 과정

[Fig. 6] Processing of playing in the existing schemes

이와 비교하여 [그림 7]은 [그림 6]과 같은 조건에서 접근하는 영역이 같은 영역이 될 때까지 프레임을 삭제하여 서비스하는 것을 보인 것으로서 1개의 영역 차이에 대한 두 개의 요구는 2개의 영역을 서비스한 후 기존의 스트림과 합병되어 추가의 버퍼를 필요로 하지 않는다.

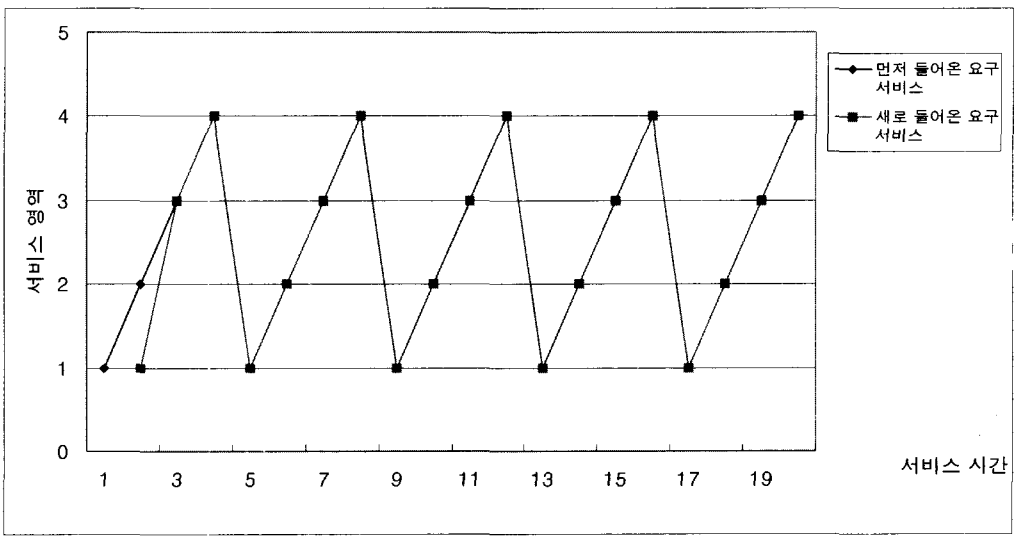
결과적으로 동적 스트림 합병 기법은 디스크에 복사본을 두는 기법을 적용하여 적은 초기 대기 시간을 갖도록 하고, 요구의 간격에 따라 처음 읽어 들이는 많은 양의 세그먼트들이 차지하는 버퍼를 서비스가 끝날 때까지 유지하는 기존의 기법의 단점을 세그먼트의 일부분을 서비스하지 않는 기법으로 버퍼 사용을 최소로 줄인다. 또한 최소 버퍼 요구량으로 줄어드는 데까지 걸리는 시간은 불과 몇 초에 지나지 않고, 줄어든 후에는 더 이상 프레임을 삭제하지 않기 때문에 서비스의 질을 많이 낮추지 않고도 버퍼 사용률을 감소시킬 수 있다.

4. 성능분석

4.1 수식적 성능분석

위에서 기술한 데이터 배치 기법들은 동시에 N개의 사용자 요구를 서비스하기 위해 디스크 헤드가 한 영역에 도착하면, 서비스 주기 시간(T) 안에 가장 먼저 서비스 되고있는 스트림의 세그먼트뿐만 아니라 다른 스트림의 세그먼트들도 읽어야하므로 N개의 세그먼트를 읽는 총 시간 T는 영역에서 세그먼트를 탐색하고 전송하는 시간과 N개의 세그먼트를 전송하는 시간의 합과 같으므로 $\frac{N \times S}{TR} + N \times r(d)$와 같다. 또한 세그먼트의 크기 S는 한 주기동안 재생 할 수 있어야 하므로 $S = DR \times T$ 이다.

이를 이용하여 기존의 제약 분할 할당 기법, 양방향 기법, 단방향 기법, 순차적 기법, 디스크에 복사본을 두는 기법과 본 논문에서 제안하는 동적 스트림 합병 기법의 초기 대기시간 $T_{Latency}$와 버퍼 요구량 B를 구하면 다음과 같다.



[그림 7] 동적 스트림 합병 기법에서의 프레임 삭제

[Fig. 7] Frame skip in the dynamic stream merging scheme

1) 제약 분할 할당 기법

이 기법은 하나의 스트림을 n개의 세그먼트로 나누어 각 세그먼트의 탐색 거리를 d로 제한하여 배치하였으나 다른 스트림간의 탐색 거리는 제한하지 않는다. 따라서 동시에 여러 스트림을 서비스 할 경우 초기 대기시간을 예측 할 수 없다.

2) 양방향 배치 기법

각 스트림의 시작 세그먼트는 첫 번째 영역에 저장되어 있으므로 디스크의 헤드가 첫 번째 영역에서 마지막 영역에 도달한 후 다시 첫 번째 영역으로 되돌아오는데 까지 걸리는 시간이 곧 초기 대기시간을 의미한다. 따라서 초기 대기시간 $T_{Latency}$는 다음과 같이 나타낼 수 있다.

$$T_{Latency} = 2 \times R \times N \times \left(\frac{S}{TR} + r\left(\frac{Cyl}{R} \right) \right) + N \times r\left(\frac{Cyl}{R} \right).$$

따라서 양방향 배치 기법은 큰 초기 대기시간을 갖는 대신 적은 양의 버퍼를 사용한다. 이때 버퍼 요구량은 $B = 2 \times S \times N$ 이다.

3) 단방향 배치 기법

단방향 배치 기법의 초기 대기 시간은 디스크 헤드가 첫 번째 영역에서 마지막 영역에 도달한 후 시작 세그먼트가 저장되어 있는 첫 번째 영역으로 되돌아오는데 까지 걸리는 시간이지만 양방향 배치 기법과 비교하여 초기 대기시간이 반으로 감소한다. 그러므로, 초기 대기시간 $T_{Latency}$는 아래와 같고, 사용하는 버퍼 요구량은 양방향 배치 기법과 같은 $B = 2 \times S \times N$ 이다.

$$T_{Latency} = R \times N \times \left(\frac{S}{TR} + r\left(\frac{Cyl}{R} \right) \right) + N \times r\left(\frac{Cyl}{R} \right)$$

4) 순차적 기법

디스크에서 단방향 배치기법으로 놓여 있는 것을 순차적으로 읽어 들이므로 초기 대기 시간 $T_{Latency} = Cyl \times R \times N$ 이고 버퍼 요구량은 다음과 같다.

$$B = ((R+1) \times (N-1) + 2) \times S \leq Mem$$

5) 디스크에 복사본을 두는 기법

복사 디스크에 초기 몇 개의 세그먼트들을 복사해 두므로 초기 대기 시간은 $T_{Latency} = r(Cyl)$ 이 된다. 또한 순차적 기법을 따르므로 버퍼 요구량은 순차 기법과 동일하다.

6) 동적 스트림 합병 기법

디스크 복사 기법을 적용하므로 초기 대기시간은 디스크 복사 기법에서와 같이 $T_{Latency} = r(Cyl)$ 이 된다. 버퍼 요구량은 서비스 시작에서는 디스크 복사 기법과 동일하나 점차 감소하므로 최종 버퍼 요구량은 다음과 같다.

$$B = R + 1 + N \times 2 \times S \leq Mem$$

4.2 시뮬레이션 성능평가

본 논문에서 제안한 동적 스트림 합병 기법의 성능은 시뮬레이션을 통하여 실험적으로 분석되었다. 시뮬레이션은 Sun Sparc 2.6 환경에서 객체지향 언어인 JAVA를 이용하였다. 실험에서 각 사용자의 요구는 서로 독립적이라 가정하였고 요청 시간 간격은 지수 확률 분포로 하였다. 또한 모든 비디오의 재생 속도는 1.5Mbps로 하였다. 디스크 파라미터는 IBM Ultrastar 9ES 디스크의 최악(worst case)의 경우의 값[6]을 적용하여 전송 속도는 108.6Mbps, 실린더 수는 8420개로 하였다. 초기 대기시간의 증가는 비교적 짧은 상영시간을 갖는 비디오 클립이나 비디오 게임 같은 실시간 대화형 서비스에서 큰 문제를 가지므로 전체 시뮬레이션 시간은 비교적 짧게 20분으로 하였고, 하나의 스트림 재생시간은 10분으로 하였다.

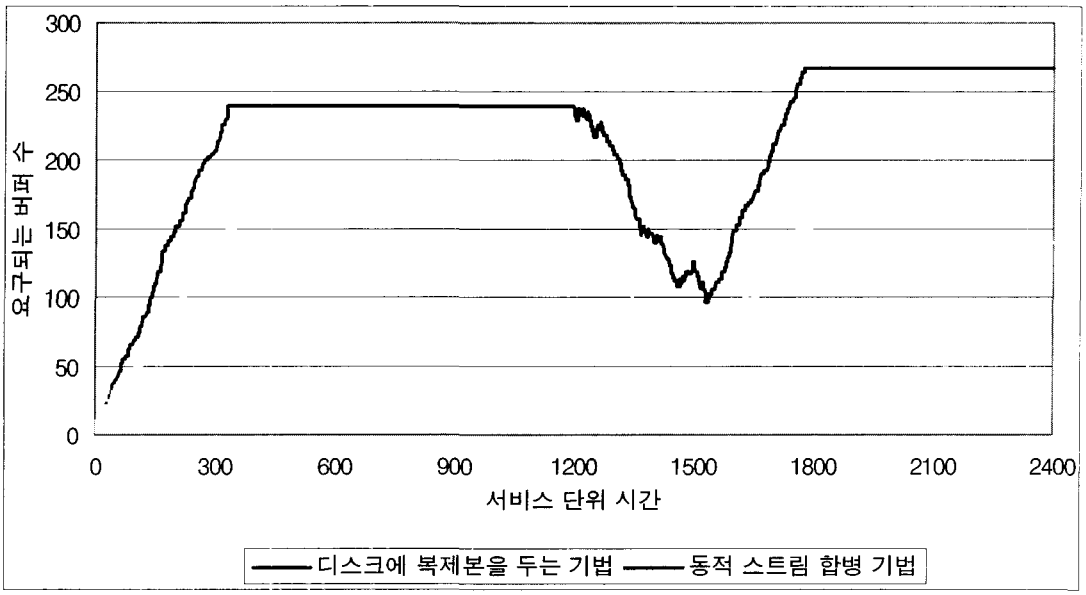
한 세그먼트는 15개의 프레임을 갖는 것으로 하였고, 하나의 서비스 단위시간은 한 세그먼트를 서비스하는데 걸리는 시간을 의미한다. 일반적인 비디오 서비스는 1초에 30프레임을 서비스하므로 한 세그먼트를 상영하는데 걸리는 시간은 0.5초이다. IBM Ultrastar 9ES 디스크 값을 적용하였을 때 동시에 사용할 수 있는 최대 사용자 수는 72명이다.

그러나, 이 값은 영역 수의 변화와 초기 대기시간 변화에 의해 줄어들 수 있다. 따라서, 실험에서 동시에 사용할 수 있는 최대 사용자 수는 70명으로 조절하였다.

성능 분석은 영역 수 변화와 프레임 합병 율에 따른 버퍼 요구량 비교와 시스템 메모리에 따른 처리량, 처음 서비스를 요구한 사용자의 초기 대기시간 변화를 조사하였다. 먼저, 실험 결과로 [그림 8], [그림 9]는 디스크 영역의 수를 4와 10으로 했을 때 요구되는 버퍼 수를 비교한 것으로 디스크 복사 기법과 비교하여 제안된 기법이 적은 양의 메모리를 요구함을 볼 수 있다. 또한 영역의 수가 커질수록 디스크에 복제본을 두는 기법과 비교하여 제안된 동적 스트림 합병 기법이 적은 양의 메모리를 요구함을 볼 수 있다. 이는 영역의 수가 커질수록 디스크 헤드와의 영역 차이가 커지므로 보다 많은 세그먼트를 버퍼로 읽어 들여야하기 때문이다.

[그림 8]에서 디스크 복사 기법의 경우 서비스 단위시간 0부터 사용자의 요구가 지수 분포로 들어옴으로써 요구하는 버퍼량은 계속 증가하다가 서비스 단위 시간 약 330에서 디스크가 동시에 서비스 해 줄 수 있는 한계인 70명에 도달하였기 때문에 더 이상 요구를 받아들이지 않고 일정 상태의 버퍼를 유지하고 있음을 보인다.

현재 서비스하고 있는 요구들 중 어떤 요구가 서비스를 마치고 나갈 때까지 기존의 요구들만 서비스하기 때문에 요구하는 버퍼량은 계속 일정하게 유지하다가 처음에 들어온 요구들이 서비스를 끝내고 나가기 시작하는 시점인 서비스 단위시간 약 1200부터 요구되는 버퍼량이 변하게 된다. 마찬가지로 동시에 서비스하는 사용자의 수가 70명에 이르면 다시 일정 상태를 보인다. [그림 8]에서 동적 스트림 합병 기법이 디스크 복사 기법에 비해 늦게 일정 상태를 보이는 것은 요구하는 버퍼량이 감소하는데 약간의 시간이 소비되기 때문이다. 또한, 동적 스트림 합병 기법은 디스크 복사 기법 보다 약 100개 정도의 버퍼를 적게 요구하는 것을 볼 수 있다.



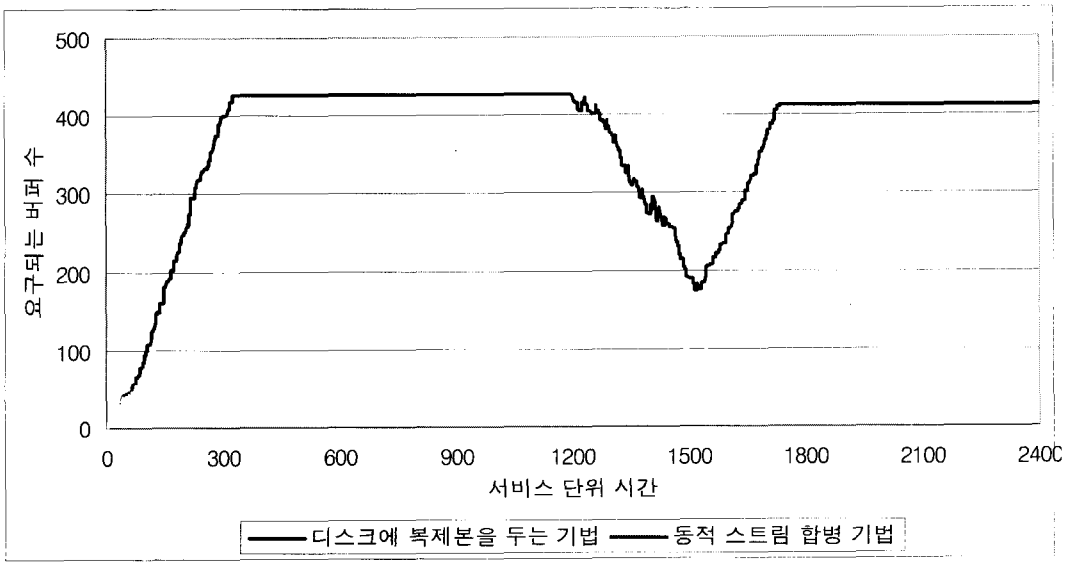
[그림 8] 영역 수가 4이고 합병 율이 5%일 때 요구 버퍼량

[Fig. 8] Size of the required buffer where the number of regions is 4 and the merging ration is 5%

[그림 9]는 영역 수가 10이고 합병 율이 5%일 때 요구하는 버퍼량을 보인다. [그림 8]에서 디스크복사 기법과 동적 스트림 합병 기법이 요구하는 버퍼 수는 약 100개 정도의 차이를 갖는 반면, [그림 9]는 약 250개 이상의 차이를 갖는다. [그림 9]가 [그림 8]에 비해 더 큰 차이를 보이는 것은 영역의 수를 더 크게 나눌수록 디스크 복사 기법에서는 디스크 헤드가 접근하고 있는 영역과의 차가 커져 더 많은 양의 세그먼트를 버퍼에 읽어 놓아야 하기 때문이다. 반면, 동적 스트림 합병 기법은 처음에는 많은 버퍼를 사용하다가 점차 최소 버퍼 요구량으로 줄여 나가기 때문에 나눠지는 영역 수와 관계없이 한 요구당 2*(세그먼트 크기)의 버퍼만을 사용하게 된다. 따라서, 디스크 복사 기법과 동적 스트림 합병 기법의 버퍼 요구량의 차이는 나눠지는 영역의 수를 크게 할수록 더 커진다.

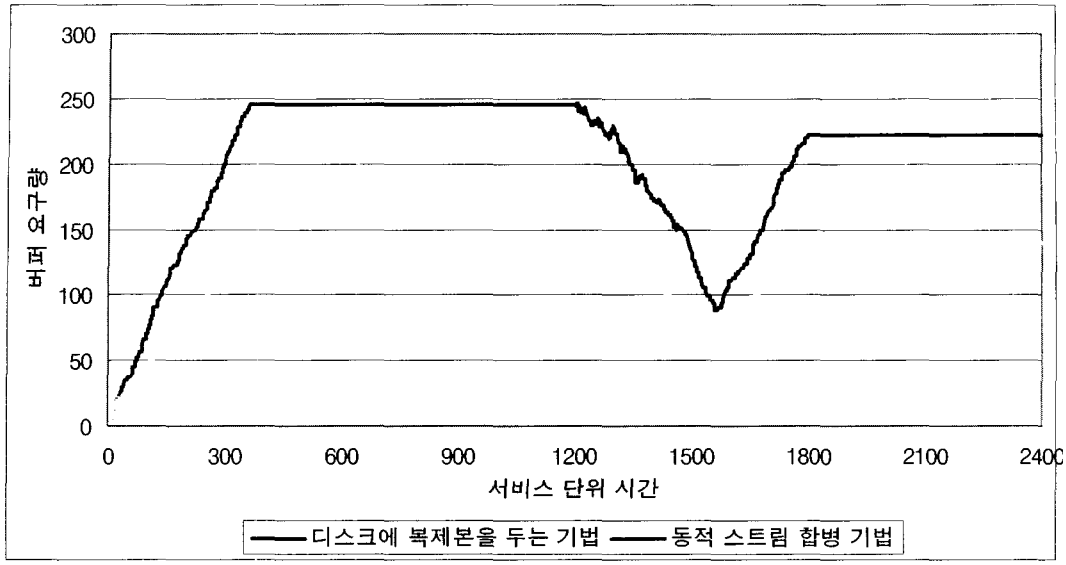
[그림 10]은 영역 수가 4이고 합병 율이 10%일 때 요구하는 버퍼량을 보인 것으로 디스크 복사 기법의 경우 앞의 [그림 8]과 유사하게 나타난다. 그러나, 동적 스트림 합병 기법의 경우 요구하는 버퍼 수가 일정하게 되는 시간이 좀 더 빨라졌음을 알 수 있다.

[그림 11]은 나눠지는 영역 수에 따라 가장 처음 서비스를 요구하는 사용자의 시작 세그먼트를 읽어 오는 데 걸리는 초기 대기시간을 나타낸 것으로 나눠지는 영역 수가 커질수록 초기 대기시간이 적어지는 것을 보인다. 이는 디스크를 많은 영역 수를 갖도록 나누면 한 영역에 해당하는 트랙 수가 적어지기 때문에 더 적은 트랙에서 시작 세그먼트를 찾기 때문이다. 여기서 초기 대기시간은 위에서 정리한 수식들을 적용하였다.



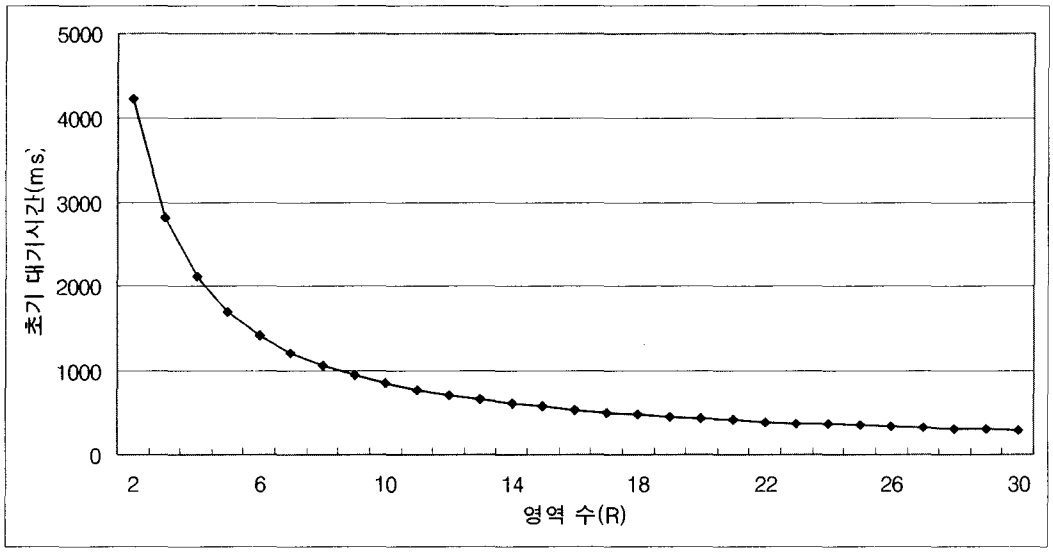
[그림 9] 영역 수가 10이고 합병 율이 5%일 때 버퍼 요구량

[Fig. 9] Size of the required buffer where the number of regions is 10 and the merging ratio is 5%



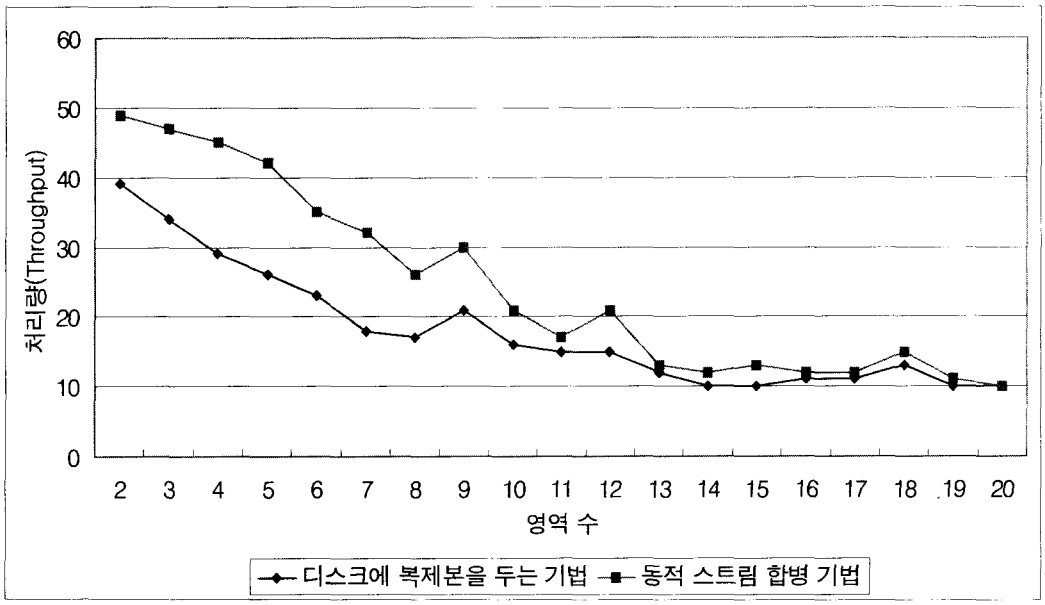
[그림 10] 영역 수가 40이고 합병 율이 10%일 때 버퍼 요구량

[Fig. 10] Size of the required buffer where the number of regions is 4 and the merging ratio is 10%



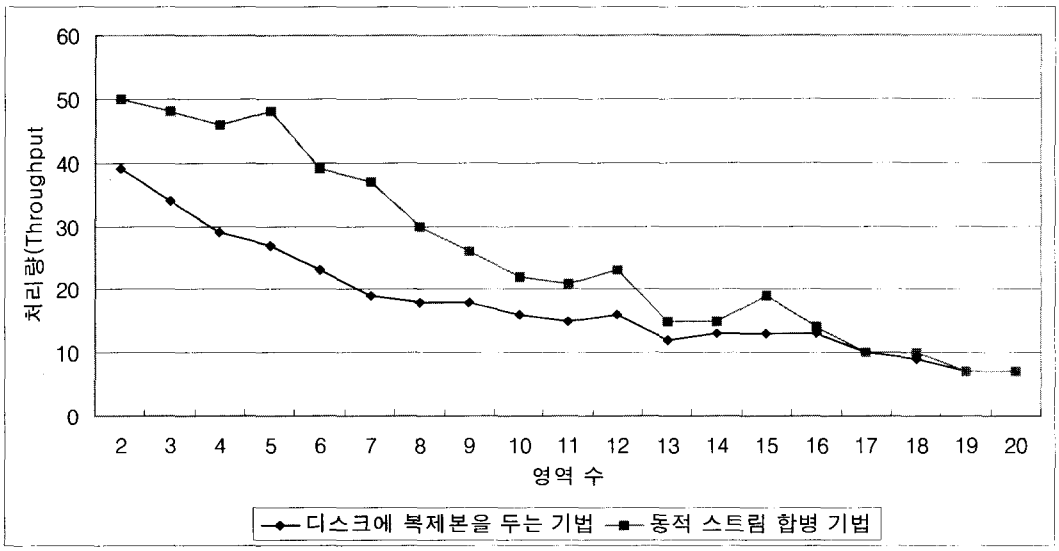
[그림 11] 영역 수에 따른 처음 사용자 요구의 시작 세그먼트를 읽어오는데 걸리는 초기 대기시간

[Fig. 11] The initial latency time for reading the initial segments of the first user



[그림 12] 합병 율이 5%이고 시스템 메모리가 100MB일 때 처리량

[Fig. 12] The throughput where the merging ratio is 5% and the size of the system memory is 100MB



[그림 13] 합병 율이 10%이고 시스템 메모리가 100MB일 때 처리량

[Fig. 13] Throughput where the merging ratio is 10% and the size of the system memory is 100MB

[그림 12], [그림 13]은 시스템의 메모리가 100MB이고 합병율을 5%, 10%로 적용할 때 처리량(throughput)을 보인 것이다. 앞의 성능분석에서 나눠지는 영역의 수가 커질수록 디스크 복사 기법이 동적 스트림 합병 기법에 비해 더 많은 버퍼를 요구하는 것을 보았다. 따라서, 시스템 메모리를 빨리 소모하는 디스크 복사 기법이 더 적은 처리량을 나타내며, 이것은 나눠지는 영역 수가 커질수록 더 적어지며 합병율이 커질수록 더 큰 처리량을 나타냄을 보인다.

5. 결 론

VOD 시스템의 구성에 있어 핵심 요소라 할 수 있는 VOD 서버는 대용량의 멀티미디어 정보를 저장·관리하며 여러 가입자가 동시에 요구하는 멀티미디어 서비스를 연속적으로 처리하기 위한 대용량 데이터의 실시간 처리 능력을 갖는 컴퓨터 구조를 필요로 한다. 이때 VOD 서버는 가능한 많은 사용자들에게 동시에 실시간 서비스를 지원하기 위해서 정교한 디스크 스케쥴링과 데이터 배치기법이 필요하며, 이들 기법 중에서 제약 분할 할당(constrained and partitioned allocation)과 라운드로빈 스케쥴링의 조합된 기법은 간단하면서도 이러한 조건을 비교적 잘 만족한다.

본 논문에서는 이러한 기법들에서 큰 문제점인 초기 대기시간을 감소시키기 위한 방법으로 동적 스트림 합병 기법을 제안하였다. 제안된 기법은 VOD 서버에서 비디오 서비스의 경우 약간의 QoS 변화가 서비스의 질에 큰 영향을 미치지 않는 점을 이용하여 디스크 상의 데이터 배치 기법과 스케쥴링 방법은 기존 방법과 동일하게 하면서, 서비스 요청의 도착시간을 기준으로 일정 시간 내에 들어오는 새로운 서비스에 대하여 재생 속도를 약간 증가시켜 이전의 서비스와 합병시킴으로써 적은 양의 버퍼만으로 초기 대기시간 줄이고 주어진 디스크 능력 하에서 최대한의 사용자를 수용할 수 있도록 하였다. 제안된 기법의 성능은 수식과 시뮬레이션을 통하여 분석되었으며, 그 결과 제안된 기법이 기존의 방법에 비하여 적은 버퍼 요구량으로 초기 대기시간이 감소됨을 나타내며, 또한 동일한 조건하에서 수용 가능한 사용자 수가 증가되어 성능이 향상됨을 보였다.

본 논문에서 제안한 스트림 합병 기법은 최근 인터넷과 멀티미디어 활용의 확산에 따라 일반화되어 가고 있는 인터넷 방송, 디지털 방송, 각종 동영상 멀티미디어 서비스 등에서 대용량 서버의 구축에 효과적으로 활용될 수 있으며, 앞으로 이러한 실제 시스템에 적용한 실험을 통하여 그 활용 가능성과 효율성을 검증할 예정이다.

※ 참고문헌

[1] A. L. N. Reddy and J. C. Wyllie, "I/O issues in a multimedia system," IEEE Computer, Vol. 27, No. 3, March 1994.

[2] D. A. Adjeroh and K. C. Nwosu, "Multimedia Database Management Requirements and Issues," IEEE Multimedia, July, 1997.

[3] D. Wu, Y. T. Hou, W. Zhu, Y. Q. Zhang, and J. M. Peha, "Streaming Video Over the Internet: Approaches and Directions," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 11. No. 1 Feb. 2001.

[4] E. Chang and H.Garcia-Molina, "Effective Memory Use in a Media Server," Proc. of 23rd Very Large Database Conference, Aug. 1997.

[5] E. Chang and H. Garcia-Molina, "Reducing Initial Latency in Media Server," IEEE Multimedia, Vol 4, No 3, July 1997.

[6] IBM, Hard Disk Drives, http://ssdweb01.storage.ibm.com/hardsoft/diskdrdl/ultra/9esdata. htm#3.

[7] S. Ghandeharizadeh, S. Kim, and C. Shahabi, "On Configuring a Single Disk Continuos Media Server," Proc. of Sigmetrics Performance Evaluation, Vol.23, No.1, 1995.

[8] T. D. C. Little and D. Venkatesh, "Prospects for Interactive Video-on-Demand," IEEE Multimedia, Vol. 1, Fall 1994.

[9] T. Kunii, "Issues in Storage and Retrieval of Multimedia Data," ACM Multimedia Systems, Vol. 3, No. 5-6, 1995.

[10] 김근혜, 최황규, "실시간 멀티미디어 저장 서버에서 초기 대기시간 최소화를 위한 동적 스트림 합병 기법," 99 추계 종합학술대회 논문집, 한국정보과학회, 1999년 10월.

[11] 마평수, 조창식, 진윤숙, 신규상, "초기 대기시간 감소를 위한 디스크 재스케쥴링 방법," 99 추계 종합학술대회 논문집, 한국정보처리학회, 1999년 10월.

[12] 이종민, 이귀영, 이홍규, "멀티미디어 서버에서의 실시간 처리 디스크 스케쥴링 기법," 정보과학회지, 제14권, 제9호, 1996년 9월.

[13] 장은정, 최황규, "동적 버퍼 분할에 의한 연속 매체 데이터 버퍼 관리 방법," 98 하계 종합학술대회 논문집, 한국통신학회, 1998년 7월.

김 근 혜

1998년 2월 강원대학교 컴퓨터공학과(학사)
2000년 2월 강원대학교 컴퓨터공학과(석사)
2000년 8월 ~ 2001년 8월 윤텔 근무
2001년 9월 ~ 현재 (주) 비씨카드 근무
관심분야 : 멀티미디어 시스템, 데이터베이스 시스템, 멀티미디어 정보검색, 인터넷 응용 등

최 황 규

1984 2월 경북대학교 전자공학과(학사)
1986년 2월 한국과학기술원 전기 및 전자공학과(석사)
1989년 8월 한국과학기술원 전기 및 전자공학과(박사)
1994년 7월 ~ 1995년 7월 Univ. of Florida Database R&D Center 방문교수
199년 3월 ~ 2001년 2월 강원대학교 전기전자정보통신공학부 교수
관심분야 : 멀티미디어 시스템, 데이터베이스 시스템, 멀티미디어 정보검색, 병렬 I/O 시스템 등