비디오 서버를 위한 예약기반 하이브리드 디스크 대역폭 절감 정책의 설계 및 평가

오 선 진[†] · 이 경 숙[††] · 배 인 한[†††]

요 약

주문형 비디오 시스템의 성능에서 핵심적인 문제는 클라이언트 요청들을 만족시키기 위해 요구되는 비디오 서버의 입·출력 대역폭이다. 이는 지연시간을 증가시키는 원인이 되는 중요한 자원이다. 따라서, 공유를 통하여 비디오 서버의 입·출력 요청을 감소시키는 일괄처리 및 피기백킹 등의 다양한 방법들이 사용되고 있다. 일괄처리 정책은 같은 객체에 대한 요청들을 묶어서 기억장치 서버에 대한 하나의 입·출력 요청을 만드는 것이고, 피기백킹 정책은 객체의 대응하는 입·출력 스트림들을 그룹으로 서비스할 수 있는 하나의 스트림으로 병합하기 위하여 진행중인 요청들의 디스플레이 율을 변경하는 정책이다. 본 논문에서는 인기 있는 비디오에 대한 요청들이 가능한 한 스케줄링 되도록 비디오 서버의 입·출력 스트림 용량을 비디오 요청 도착율에 따라 동적으로 예약해 둔다. 그리고 일괄처리의 단점인 스트림 요청 지연시간과 피기백킹의 문제점인 디스크 대역폭 낭비 등을 해결하는 예약기반 하이브리드 디스크 대역폭 절감 정책을 제안한다. 제안한 정책의 성능은 시뮬레이션을 통해 기존의 일괄처리와 피기백킹 정책들과 비교 평가한다. 시뮬레이션 결과에 따르면, 예약기반 하이브리드 디스크 대역폭 절감 정책이 기존의 일괄처리나 피기백킹 정책들에 비해 낮은 서비스 이탈확률과 높은 프레임 절약 백분율 그리고 일정한 서비스 평균대기시간을 제공함을 알 수 있다.

Design and Evaluation of a Reservation-Based Hybrid Disk Bandwidth Reduction Policy for Video Servers

Sun-Jin Oh[†] · Kyung-Sook Lee[††] · Ihn-Han Bae[†††]

ABSTRACT

A critical issue in the performance of a video-on-demand system is the required I/O bandwidth of the video server in order to satisfy clients' requests, and it is the crucial resource that may cause delay increasingly. Several approaches such as batching and piggybacking are used to reduce the I/O demand on the video server through sharing. Batching approach is to make single I/O request for storage server by grouping the requests for the same object. Piggybacking is the policy for altering display rates of requests in progress for the same object in order to merge their corresponding I/O streams into a single stream, and serve it as a group of merged requests. In this paper, we propose a reservation-based hybrid disk bandwidth reduction policy that dynamically reserves the I/O stream capacity of a video server for popular videos according to the loads of video server in order to schedule the requests for popular videos immediately. The performance of the proposed policy is evaluated through simulations, and is compared with that of batching and piggybacking. As a result, we know that the reservation-based hybrid disk bandwidth reduction policy provides better probability of service, average waiting time and percentage of saving in frames than batching and piggybacking policy.

키워드 : 주문형 비디오 시스템(VOD(video-on-demand) system), 비디오 서버(video server), 입출력 대역폭(I/O bandwidth), 디스크 대역폭 절감(disk reduction), 일괄처리(batching), 피기백킹(piggybacking)

1. 서 론

최근 정보통신 분야의 급속한 기술적인 발전과 더불어 문자나 숫자 데이터 중심의 통신 서비스에서 오디오, 비디오 및 고해상도 이미지 등의 멀티미디어 데이터를 전송하는 서비스의 제공이 가능하게 되었다. 특히 ATM(asynchronous transfer mode) 전송방식을 기반으로 하는 B-ISDN (broad-integrated services digital networks)의 구현이 가능해 지면서 다수의 클라이언트들이 서버에 접속하여 정보를 요청하고 이를 수신하는 주문형 비디오(video on demand), 전자상거래 시스템(electronic commerce), 전자 도서관 등과 같은 다양한 멀티미디어 시스템들이 실용화되고 있다. 오늘날의 정보통신 시스템은 단순히 커다란 멀티미디어 객체를

† 정 회 원 : 세명대학교 전산정보학부 정보통신학과 교수
†† 준 회 원 : 대구가톨릭대학교 대학원 전산통계학과
††† 정 회 원 : 대구가톨릭대학교 컴퓨터정보통신공학부 교수
논문접수 : 2001년 5월 28일, 심사완료 : 2001년 9월 12일

저장하고 검색하는 것뿐만 아니라 그 객체를 일정한 대역폭에서 지속적으로 제공하는 엄격한 실시간 요구사항을 만족시킬 수 있어야 한다. 이는 연속 매체의 전달 경로 상에서 필요한 자원(처리기, 디스크 대역폭, 메모리 대역폭, 통신망 대역폭 등)을 예약함으로써 연속 매체의 실시간 처리를 보장할 수 있다. 따라서, 대용량의 데이터가 요구하는 저장공간과 전송 대역폭을 줄이기 위해 사용되는 압축기법 그리고 데이터의 실시간 처리를 보장하기 위한 자원의 예약기법은 모두 연속 매체를 처리하는데 있어 필수적이다. 멀티미디어 시스템은 교육용 응용, 게임 기술, 도서관 정보 시스템 등에서 중요한 역할을 하고 있으며, 이러한 시스템에서 가장 중요한 것은 다수의 클라이언트들에게 주문형 서비스를 동시에 제공하는 것이다. 즉, 클라이언트들은 비디오 등과 같은 객체를 요청하고, 적절한 지연시간 내에 요청한 객체를 관람하기를 기대한다. 여기서 지연 시간은 요청이 시스템에 도착한 시점부터 디스크에서 객체 읽기를 초기화하기까지의 시간 간격으로 정의되고, 데이터가 실제적으로 디스플레이 장치에 전달될 때까지의 추가적인 지연은 디스크 지연에 비해 상대적으로 작기 때문에 무시할 수 있다. 이러한 지연은 서비스 요청을 위한 불충분한 대역폭, 디스크로부터 판독 내용을 스케줄링 하기 위한 불충분한 버퍼 공간, 그리고 불충분한 디스크 기억장치 등의 요인으로 발생한다. 이러한 지연 요소 중에서 특히 입출력 대역폭(I/O bandwidth)은 매우 중요한 자원으로 실시간 처리를 보장하면서 공유를 통하여 기억장치 서버의 입출력 요청을 감소시켜 동시에 서비스할 수 있는 요청 클라이언트의 수를 증가시킬 수 있어야 한다. 이를 위한 다수의 접근 방법들이 다음과 같이 제안되고 있다[1-4].

- 일괄처리(batching) : 일정한 일괄처리 윈도우 동안 도착하는 동일한 객체에 대한 요청들을 묶어서 단일 입출력 스트림으로 한꺼번에 서비스하는 방법이다.
- 브리징(bridging) : 중앙 처리기의 메모리를 버퍼로 이용하는 방법이다. 특정 비디오에 대해 고정된 개수의 프레임을 버퍼링 하여, 대응하는 시간 내에 발생하는 비디오에 대한 요청을 디스크가 아닌 버퍼로부터 읽는 방법이다. 그러나 브리징은 많은 양의 버퍼 공간을 요청하는 단점이 있다.
- 피기백킹(piggybacking) : 동일한 객체에 대한 입출력 스트림이 하나로 합병될 때까지 진행중인 스트림의 디스플레이 율을 조정하는 기법이다.

그러나 일괄처리 정책에서 스트림 요청의 지연시간과 기억장치 서버의 입출력 요청의 감소는 서로 상반관계에 있다. 그리고 피기백킹 정책에서 도착하는 비디오 요청은 즉시 스케줄링 되나 재생되고 있는 두 스트림이 하나로 합병될 때까지는 두 스트림이 동시에 사용되어서 디스크 대역폭의 낭비가 발생한다. 본 논문에서는 인기 있는 비디오 요청들이 가능한 한 스케줄링 되도록 하기 위해 비디오 서버의 입출력 스트림 용량 예약을 기반으로 한다. 그리고 일괄처리 정책에서의 긴 서비스 지연시간과 피기백킹 정책에서의 디스크 대역폭 낭비를 줄일 수 있는 예약기반 하이브리드 디스크 대역폭 절감정책을 제안하고, 성능을 시뮬레이션을 통하여 기존의 일괄처리와 피기백킹 정책들과 비교 평가한다. 본 논문의 구성은 다음과 같다. 2장에서는 공유를 통한 저장 서버의 입출력 대역폭 절감에 대한 관련 연구를 살펴보고, 3장에서는 본 논문에서 제안하는 예약기반 하이브리드 디스크 대역폭 절감정책을 설명하고, 4장에서는 시뮬레이션을 통하여 제안하는 알고리즘의 성능을 평가한다. 그리고 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

디스크 입출력 대역폭은 서비스의 지연 시간을 결정하는 중요한 자원이다. 이러한 지연 시간을 줄이는 경제적인 방법으로는 데이터 배치 기법(data layout techniques)과 스케줄링 기법을 향상시키거나 같은 객체에 대한 요청들간의 데이터 공유를 통하여 서비스 중인 각 요청들의 입출력 요청을 감소시키는 것이다. 여기서는 공유를 통하여 저장 서버의 입출력 요청을 감소시켜 동시에 서비스되는 클라이언트 요청의 수를 증가시키는 방법들을 살펴본다.

2.1 일괄처리

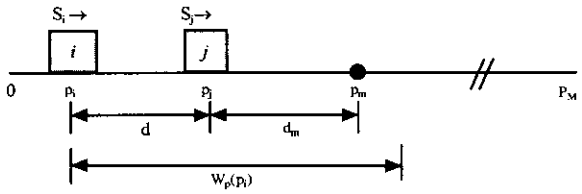
일괄처리는 같은 객체에 대한 요청들을 묶어서, 기억장치 서버에 대한 하나의 입출력 요청을 만드는 기법이다. 그러나 일괄처리 방법에서 스트림 요청의 지연 시간과 기억장치 서버의 입출력 요청의 감소는 서로 상반관계에 있다. 일괄처리 정책은 크기에 의한 일괄처리와 시간에 의한 일괄처리로 분류할 수 있다. 각 객체 i에 대해 미리 정해진 요청 개수 B_i를 일괄처리 윈도우라 할 때, 크기에 의한 일괄처리는 객체 i에 대해 B_i만큼의 요청 개수가 누적되면 객체 i를 위해 입출력 스트림을 초기화한다. 따라서 객체 i에 대해 감소되는 입출력 스트림의 크기는 B_i-1이다. 비록 크기에 의한 일괄처리 정책이 저장서버의 입출력 요청을 감소시키지만, 적정 도착율보다 낮은 요청에서 긴 지연시간이 발생할 수 있다. 시간에 의한 일괄처리는 시간 단위로 일괄처리 윈도우를 설정하는 것이다. 하나의 요청이 기억장치 서버에 도착하고, 같은 객체 i에 대한 요청이 존재하지 않을 때 타이머가 설정된다. 일괄처리 윈도우 B_i 동안 모아진 같은 객체 i에 대한 요청은 타이머가 만기되면 하나의 입출력 스트림으로 서비스된다[2].

주문형 비디오 시스템의 스케줄링 방법에 관한 연구는

클라이언트들의 요청이 도착하였을 때 이들을 공평하게 서비스하여 시스템의 처리율을 높일 수 있도록 하는데 그 목적이 있다. 효율적인 비디오 스케줄링 정책은 일괄처리 윈도우뿐만 아니라 클라이언트의 이탈 확률과 대기 시간을 고려해야 한다. A. Dan의 연구[2]에서는 멀티캐스트 전송을 이용하는 스케줄링 알고리즘으로 일괄처리를 위한 일반적인 스케줄링 정책인 가장 긴 대기 시간을 갖는 비디오 요청을 먼저 스케줄링 하는 FCFS(first come first served) 정책과 최대 개수의 대기 요청을 갖는 비디오를 먼저 선택하는 MQL(maximum queue length) 정책을 비교하였다. MQL은 일괄처리 되는 요청의 개수를 최대로 하기 위하여 큐 길이만 고려함으로써 인기 있는 비디오의 스케줄링에 너무 적극적인 반면, FCFS는 이탈을 줄이기 위하여 도착 시간에 초점을 맞추고 큐 길이를 완전히 무시함으로써 MQL의 반대 효과를 가진다[5]. A. Dan의 연구[7]에서는 FCFS 정책을 확장한 FCFS-n 정책을 제안하였다. FCFS -n 정책에서는 서버 용량의 일부분이 n개의 인기 있는 비디오에 대한 요청들을 일괄처리하기 위해 예약되고 선할당 되어진다. 이렇게 함으로써 인기 있는 비디오에 대해 초과될 수 없는 최대 대기시간을 제공할 수 있고, 인기 없는 비디오에 대한 요청이 인기 있는 비디오에 대한 서비스를 간섭하지 않으므로 상대적으로 작은 서버 용량으로 높은 수용 확률을 보장하였다. C. Aggarwal의 연구[6]는 factored queue length의 개념을 도입하여 maximum factored queue length를 갖는 비디오를 스케줄링 하는 일괄처리 정책인 MFQ를 제안하였다. Factored queue length는 차별적인 가중 요소를 서로 다른 비디오의 큐 길이에 적용함으로써 얻어진다. 여기서 factored queue length는 비디오의 큐 길이를 그것의 상대적 요청 회수의 제곱근으로 나눈 값으로 정의하였다. 그리고 MFQ, FCFS, MQL을 시뮬레이션 한 결과, MFQ가 평균 지연 시간, 이탈 확률, 공평성에서 우수한 실험 결과를 보였다.

2.2 피기백킹

피기백킹이란 동일한 객체에 대한 입출력 스트림들을 그룹으로 서비스 할 수 있는 하나의 입출력 스트림으로 병합하기 위하여 진행중인 요청들의 디스플레이 율을 조정하는 기법이며 일괄처리는 일종의 정적 피기백킹이다. 동적 피기백킹은 다음과 같이 수행된다. 요청에 따라 각 디스플레이 스트림에 대한 하나의 입출력 스트림이 초기화된다. 디스플레이 스트림이 같은 객체에 대한 다른 디스플레이 입출력 스트림에 동적으로 피기백킹 되는 것을 허용한다. 이는 두 개의 입출력 스트림을 하나로 합병하는 것으로 볼 수 있다. 합병 전에는 각각 한 개 이상의 디스플레이 스트림을 서비스하는 두 개의 입출력 스트림이 합병 후에는 두 개의 디스플레이 스트림을 동시에 서비스하는 한 개의 입출력 스트림이 된다. 합병은 요청의 디스플레이 속도를 조정함으로써 이루어진다. 즉, 각 요청에 대해 정상 속도로 디스플레이 하는 것이 아니라, 디스플레이들간의 간격을 좁히기 위해 더 느린 율로 혹은 더 빠른 율로 각 요청의 디스플레이 속도를 조절할 수 있다. 이렇게 하여 간격이 없어지면 두 디스플레이 스트림들은 하나의 입출력 스트림으로 합병되어 디스플레이 된다. 이러한 디스플레이 속도 조정이 어떻게 이루어지는지를 간단하게 살펴본다. 우선 기억장치 서버에 의해 작동되는 디스플레이 단위는 NTSC(national television system committee) 표준이고 초당 30 프레임의 율로 디스플레이 된다고 가정한다. 디스플레이 율의 가속과 감속은 비디오에 부가적인 프레임을 추가하고 삭제함으로써 가능하다. 그리고 정상율의 ±5% 정도의 실제 디스플레이 율 변경은 클라이언트에 의해 인지되지 않는다. 따라서 실제 디스플레이 율을 5% 낮추거나 높이는 것은 비디오 품질을 손상시키지 않고 가능하다는 가정에 기초한다[1].



(그림 1) 시스템의 상태

(그림 1)은 전체 디스플레이 기간이 P_M인 객체의 디스플레이 상태를 나타낸다. 각 디스플레이 스트림은 그 객체의 디스플레이내의 현재 위치로 식별된다. 즉, (그림 1)에서 스트림 i는 p_i로 식별되고, 디스플레이 속도 S_i로 이동한다. 입출력 스트림 i와 j를 합병하기 위하여, 먼저 $S_i > S_j$가 보장되어야 하고, 그 다음 합병 기회를 확인하기 위하여 피기백킹이 사용될 수 있는 거리 제한을 정의할 수 있다. 여기서 p_m은 입출력 스트림 i와 j가 합병되는 객체의 디스플레이 위치를, d는 입출력 스트림 i와 j간의 프레임 거리를, d_m은 합병 위치와 j의 현재 위치간의 프레임 거리를 나타낸다. 그리고 격차해소 윈도우(catch-up window), $W_p(p_i)$는 어떤 피기백킹 정책 p에서 스트림 i와 스트림 j를 하나의 스트림으로 합병할 수 있는 최대 거리로 정의된다. $W_p(p_i)$는 객체의 디스플레이에 있어서 위치 p_i에 대해 상대적으로 계산된다.

동적 피기백킹은 다음의 도착(arrival), 합병(merge), 종료(dropoff), 그리고 윈도우 횡단(window crossing) 등 네 가지 사건들 중 하나가 발생할 때 속도를 조정하는 정책들을 고려한다. 도착은 새로운 입출력 스트림의 초기화에 해당되고, 합병은 두 입출력 스트림의 합병을 나타내며, 종료는 객체의 디스플레이 끝에 해당된다. 그리고 윈도우 횡단은

(그림 1)에서의 격차해소 윈도우의 경계를 넘어가는 것을 의미한다. 합병이 빠를수록 더 많은 자원이 다른 요청들을 서비스하기 위해 사용될 수 있다. 그러므로 정상 속도에서 최대 가능 편차를 가정한 다음 세 가지 디스플레이 율인 가장 느린율 (S_{min}), 정상율 (S_n), 가장 빠른율 (S_{max})을 고려한다. 위와 같은 기본 개념을 바탕으로 제안된 동적 피기백 합병 정책들은 다음과 같다[1].

2.2.1 홀짝 감소 정책

홀짝 감소 정책은 최대 50% 정도의 입출력 요구를 감소시킨다. 이 정책은 연속적인 도착 스트림들을 합병하기 위해 짝을 지운다. 새로운 입출력 스트림이 시스템에 도착하고, 최대 격차해소 윈도우 $W_{oe}(0)$내에 S_{min}의 속도로 앞서 가는 동일한 객체를 위한 입출력 스트림이 있을 때 새로운 요청의 속도를 S_{max}로 설정한다. 그러면 이 두 개의 스트림은 일정 시간 후에 합병될 것이다.

2.2.2 단순 병합 정책

단순 병합 정책은 객체 디스플레이의 시작에서 상대적으로 측정된 격차해소 윈도우 $W_{sm}(0)$와 함께 어떤 객체의 디스플레이 시작에서 상대적으로 측정된 최대 합병 윈도우 $W_{sm}^{m}(0)$을 사용한다. $W_{sm}^{m}(0)$는 두 개의 스트림을 합병할 수 있는 가장 늦은 지점을 나타낸다. 즉, 스트림 i가 시스템이 도착했을 때, $W_{sm}(0)$내에 스트림 i에 앞서 있는 스트림 j를 찾았다면, 스트림 i와 j는 $W_{sm}^{m}(0)$의 오른쪽 경계에서 합병될 것이다. 일반화된 단순 병합 정책[8]에서 최대 격차해소 윈도우 (W_m)의 크기는 수식 (1)과 같이 계산된다. 여기서 L은 비디오의 프레임 길이를 나타낸다.

$$W_m = \frac{S_{\max} - S_{\min}}{S_{\max}} \cdot L \qquad (1)$$

단순 합병 정책의 기본 개념은 스트림들을 "합병 그룹"에 할당하는 것이다. 여기서 하나의 스트림 i가 그룹을 초기화하고, $W_{sm}(0)$에 있을 동안 시스템에 도착하는 모든 스트림들은 스트림 i에 합병된다. 단순 병합 정책의 알고리즘은 (그림 2)와 같다.

```
Algorithm Simple merging policy
begin
   Case arrival of stream i :
      If no stream within W_sm(0) is moving at S_min
         S_i = S_min ;
      else
         S_i = S_max ;
   Case merge of i and j
      drop stream i ;
         S_j = S_min ;
   Case window crossing, W_sm^m(0)
         S_j = S_n ;
end
```

(그림 2) 단순 병합 정책

2.2.3 탐욕 정책

탐욕 정책은 객체의 전체 디스플레이 기간 동안 가능한 많은 입출력 요청을 합병하는 정책이다. 탐욕정책은 초기 격차해소 윈도우 $W_g(0)$와 함께 객체 디스플레이내의 위치 p_i에 대해 상대적으로 측정되는 격차해소 윈도우 $W_g(p_i)$을 사용한다. 객체에 대한 요청이 도착했을 때 속도 조절은 홀짝 감소정책과 같은 방식으로 동작한다. 만약 격차해소 윈도우를 지나쳤고 스트림이 합병을 위해 아직 짝 지어지지 않았을 때, 앞선 스트림과의 합병 가능성을 고려하기 위해 $W_g(W_g(0))$을 검사한다. 만약 p_i 위치에서 합병이 일어나면, 새로운 격차해소 윈도우 $W_g(p_i)$가 계산된다. 만일 그 윈도우 내에 입출력 요청이 없으면, 그 요청의 속도는 S_n으로 설정된다. 그러나, $W_g(p_i)$내에 약간의 요청이 있고, 바로 앞선 요청이 S_n의 속도로 디스플레이 되고 있다면, 그 요청의 속도는 S_{min}으로 설정되고, 위치 p_i의 요청의 속도는 S_{max}로 설정된다.

2.3 적응적 예약기반 정책

적응적 예약기반 정책은 기존의 정책에 인기 있는 비디오를 위한 비디오 서버의 자원을 예약하는 개념을 도입한 것이다. 적응적 예약기반 일괄처리 정책(adaptive reservation-based batching, ARB)[18]은 시간에 의한 단순 일괄처리 정책에 인기 있는 비디오를 위해 동적으로 서버 용량을 예약하는 정책이다. 단순 일괄처리에서는 비디오 요청들에게 도착순으로 서버 용량을 할당하므로 비디오 요청 도착율이 어느 정도 높아지면 가용 서버 용량이 없어 도착한 비디오 요청들은 블록 되어진다. 이 때 인기 있는 비디오를 위한 가용 서버 용량이 있으면 인기 있는 비디오에 대한 같은 요청들은 일괄 처리되어 하나의 스트림으로 서비스되므로 많은 서버 용량이 절약되어 많은 비디오 요청들을 스케줄링 할 수 있을 것이다. 그러나 가용 서버 용량이 없으면 생길 때까지 기다려야 하고, 결국 비디오 서비스 대기시간이 길어지며 클라이언트의 서비스 이탈 확률이 증가하여 고품질의 주문형 비디오 서비스를 제공할 수 없게 된다. 이러한 문제점을 해결하기 위하여 인기 있는 비디오에 대한 요청들이 매 일괄처리 윈도우마다 스케줄링 될 수 있도록 서버 용량을 예약해 둔다. 이러한 예약은 정적 예약과 동적 예약으로 나눌 수 있다. 정적 예약은 비디오 요청 도착율에 관계없이 일정한 서버 용량을 예약하는 방식으로 서버 용량이 예약되는 인기 있는 비디오의 개수가 항상 일정하다. 이

방식은 비디오 요청 도착율이 높고 서버 용량이 예약되는 인기 있는 비디오 개수가 적을 경우, 매 일괄처리 윈도우마다 비디오 요청이 발생하는 인기 있는 비디오가 스케줄링 되지 못한다. 반면에 비디오 요청 도착율이 낮고 서버 용량이 예약되는 인기 있는 비디오의 개수가 많으면 서버 용량이 예약된 인기 비디오가 매 일괄처리 윈도우마다 요청이 발생하지 않아 예약 서버 용량이 낭비되는 단점이 있다. 적응적 예약기반 일괄처리 정책은 비디오 서버의 부하에 따라 인기 있는 비디오의 개수와 그 비디오들을 스케줄링 하기 위해 예약되는 서버 용량을 동적으로 조절함으로써 이러한 문제를 해결하였다.

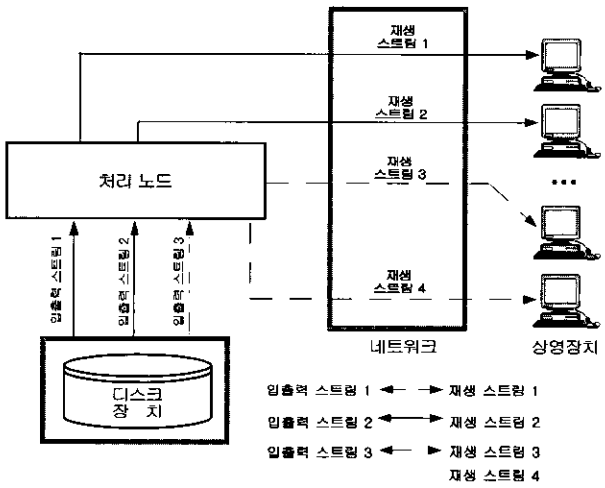
한편, 적응적 예약기반 피기백 합병 정책(adaptive reservation-based piggyback merge, ARPM)[19]은 기존의 동적 피기백 합병 정책에 인기 있는 비디오를 위한 비디오 서버의 입출력 스트림 예약이라는 개념을 도입한다. 기존의 피기백 합병 정책들은 도착하는 비디오 요청들에게 차례대로 비디오 서버의 입출력 스트림을 할당해 줌으로써 도착율이 어느 정도 이상 높아지면 모든 가용 입출력 스트림이 스케줄링 되어 도착한 모든 요청들이 무조건 기다려야 한다. 만일 이 때 비디오 서버에 가용 입출력 스트림이 있어 인기 있는 비디오 요청을 스케줄링 하면, 격차 해소 윈도우 내에는 인기 있는 비디오에 대한 디스플레이 스트림이 항상 존재하므로 많은 디스플레이 스트림이 합병되어 비디오 서버의 입출력 스트림이 많이 감소하게 된다. 그렇지 않으면, 비디오 서버의 가용 입출력 스트림이 생길 때까지 기다려야 하므로 스트림을 감소시킬 수 있는 기회를 놓치게 된다. 따라서 적응적 예약기반 피기백 합병 정책은 이러한 단점을 해소하여 인기 있는 비디오의 경우에 앞선 디스플레이 스트림과의 계속적인 합병이 가능하도록 비디오 서버의 입출력 스트림 용량을 예약해 둔다.

3. 예약기반 하이브리드 디스크 대역폭 절감 정책

주문형 비디오 시스템에서 서비스는 다수의 클라이언트들이 요청을 하면 보다 적은 자원으로 연속 매체들의 실시간 처리를 보장하면서 보다 많은 수의 클라이언트들에게 서비스를 제공할 수 있어야 한다. 이 절에서는 인기 있는 비디오 요청들을 위해 비디오 서버의 입출력 스트림 용량을 예약하고, 일괄처리 정책에서의 긴 서비스 지연시간 문제와 피기백킹 정책에서의 디스크 대역폭 낭비 문제를 줄일 수 있는 예약기반 하이브리드 디스크 대역폭 절감정책을 제안한다.

3.1 시스템 모델

본 논문에서는 (그림 3)과 같은 어떤 객체의 각 요청에 대해 하나의 재생 스트림과 이에 대응하는 입출력 스트림이 존재하는 비디오 서버 시스템을 고려한다. 처리 노드들은 디스크들로부터 필요한 데이터를 검색하고, 몇 가지 방법으로 그 데이터의 수정이 가능하며, 재생 스트림들을 사용하여 네트워크를 통해 적절한 상영 장치들로 전송하기 위한 입출력 스트림들을 사용한다.



(그림 3) 시스템의 개략적인 구조

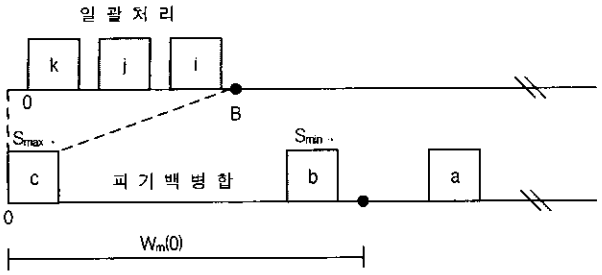
저장 서버의 입출력 요구는 같은 객체에 대한 요청들에 대응하는 다수의 재생 스트림들을 서비스하기 위하여 하나의 입출력 스트림을 사용함으로써 감소될 수 있다. (그림 3)에서 같은 객체에 대한 요청에 대응하는 재생 스트림 3과 4는 하나의 입출력 스트림을 사용하여 서비스된다. 그리고 저장 서버는 요청의 디스플레이 율을 동적으로 변경할 수 있는 기능, 즉, 어떤 객체의 디스플레이의 일부분을 동적인 시간 압축 또는 시간 확장의 기능을 갖는다고 가정한다.

3.2 예약기반 하이브리드 디스크 대역폭 절감정책의 시나리오

예약기반 하이브리드 디스크 대역폭 절감정책은 기존의 일괄처리 정책과 피기백킹 정책의 장점을 모두 수용한 정책이다. 일괄처리 정책은 일괄처리 윈도우가 만기가 된 다음 비디오 요청의 스케줄링을 시도하므로 많은 비디오 서버 입출력 요청을 감소시킬 수 있으나 스트림 요청 지연시간이 길어진다는 단점이 있고, 반면에 피기백킹은 도착하는 비디오 요청의 스케줄링을 즉시 시도하므로 스트림 요청 지연 시간은 짧아지나, 격차해소 윈도우 내에 같은 비디오에 대한 다수의 스트림이 존재하므로 감소되는 입출력 스트림 량이 일괄처리에 비해 작다는 단점이 있다. 따라서 제안하는 정책에서는 일반적인 일괄처리 정책에 비해 짧은 일괄처리 윈도우를 설정하고, 일괄처리 된 요청이 스케줄링 될 때 격차해소 윈도우 내에 같은 비디오에 대해 앞서 느린 속도로 일괄처리 되어 재생중인 스트림이 있는지 검사하여 두 스트림의 합병을 시도한다. 그렇게 함으로써 스트

림 요청 지연시간과 서버의 입출력 요청의 감소 모두에서 좋은 결과를 얻을 수 있다.

예약기반 하이브리드 디스크 대역폭 절감정책의 시나리오는 (그림 4)와 같다. 여기서 i, j, k는 요청 비디오를, a, b, c는 입출력 스트림을, B는 일괄처리 윈도우를, 그리고 $W_m(0)$는 격차해소 윈도우를 각각 나타낸다.



(그림 4) 예약기반 하이브리드 디스크 대역폭 절감정책의 시나리오

일괄처리 윈도우 안에 같은 비디오에 요청 i, j, k는 하나의 비디오 요청으로 일괄처리 되어 입출력 스트림 c로 스케줄링 될 때, 같은 비디오에 대한 입출력 스트림 b가 격차해소 윈도우 내에서 느린 속도(S_{min})로 재생되므로, 입출력 스트림 b와 c를 합병하기 위하여 입출력 스트림 c를 빠른 속도(S_{max})로 재생한다.

예약기반 하이브리드 디스크 대역폭 절감정책에서 인기있는 비디오를 위한 비디오 서버의 입출력 스트림 예약 용량은 수식 (2)에 의하여 구할 수 있다.

$$SC = \omega_h \times \left(k \times \frac{L}{T}\right) \qquad (2)$$

여기서 ω_h는 가중치로 $0<\omega_h<1$이며, T는 일괄처리 윈도우 그리고 L은 비디오의 프레임 길이를 나타낸다. 단순 하이브리드 디스크 대역폭 절감 알고리즘 구조는 (그림 5)와 같다.

```
Algorithm Hybrid Disk Bandwidth Reduction Algorithm
 Begin
  Case arrival of a video request ;
    Put the arrival video request in waiting queue ;
  Case the batching window of the front request in
       waiting queue is expired ;
    If the stream capacity is not empty
      Begin
          Batch the same video requests in waiting
            queue into a single video request(i) ;
        If no stream within W is moving at Smin
            Si = Smin ;
        Else
            Si = Smax ;
        End
    Else
        Block the video request ;
  Case merge of i and j :
        Drop stream i ;
        Sj = Smin ;
 End.
```

(그림 5) 단순 하이브리드 알고리즘

3.3 예약기반 하이브리드 디스크 대역폭 절감 알고리즘

예약기반 하이브리드 디스크 대역폭 절감 알고리즘의 구조는 (그림 6)과 같다. 예약기반 하이브리드 디스크 대역폭 절감정책에서는 먼저 인기 있는 비디오에 대한 요청을 비디오 서버의 부하에 관계없이 가능한 한 스케줄하기 위하여 인기 있는 비디오를 위한 서버 스트림 용량을 예약한다. 그리고 서버에 도착한 비디오 요청을 대기 큐에 넣는다. 대기 큐의 선두에 있는 비디오 요청의 일괄처리 윈도우가 만기가 되면 대기 큐 내의 같은 비디오에 대한 모든 요청을 하나의 요청으로 일괄처리 하여 하나의 입출력 스트림으로 스케줄링을 시도한다. 이 때 요청 비디오가 인기 있는 비디오이면 예약 입출력 스트림을 할당하고, 비인기 비디오이면 서버의 비예약 입출력 스트림을 할당한다. 그리고 스케줄링 할 비디오와 같은 비디오 스트림이 느린 속도로 격차해소 윈도우 내에 재생되고 있으면 빠른 속도로 재생을 시작한다.

```
Algorithm Reservation-based Hybrid
          Disk Bandwidth Reduction Algorithm(k)
 Begin
   Reserve the I/O stream capacity of video server for
     k-hottest videos requests ;
   Case arrival of a video request ;
        Put the arrival video request in waiting queue ;
   Case the batching window of the front request in
        waiting queue is expired ;
     If the requesting video is in the k-hottest video
       If the reserved stream capacity is not empty
         Begin
           Batch the same video requests in waiting queue
             into a single video request(i) ;
           If no stream within W is moving at Smin
             Si = Smin ;
           Else
             Si = Smax ;
         End
       Else
           Block the video request ;
 Else
    If the unreserved stream capacity is not empty
     Begin
        Batch the same video requests in waiting queue
          into a single video request(i) ;
        If no stream within W is moving at Smin
          Si = Smin ;
        Else
          Si = Smax ;
     End
    Else
        Block the video request ;
  Case merge of i and j :
        Drop stream i ;
        Sj = Smin ;
 End.
```

(그림 6) 예약기반 하이브리드 알고리즘

4. 시뮬레이션 및 평가

4.1 환경

본 논문에서는 모의실험을 통하여 단순 일괄처리(simple batching, SB), 일반화된 단순 피기백 합병(simple piggy-back merge, SPM)[1, 8], 단순 예약기반 일괄처리(simple reservation-based batching, SRB)[7], 적응적 예약기반 일괄처리(adaptive reservation-based batching, ARB)[18], 적응적 예약기반 피기백 합병(adaptive reservation-based piggy-back merge, ARPM)[19], 단순 하이브리드(simple hybrid, SH), 그리고 예약기반 하이브리드(reservation-based hybrids, RH) 정책들의 성능을 각각 비교 평가한다. 모의실험은 펜티엄 II 400MHz에서 Turbo C 2.0을 사용하여 실험하였으며, 사용된 파라미터는 <표 1>과 같다.

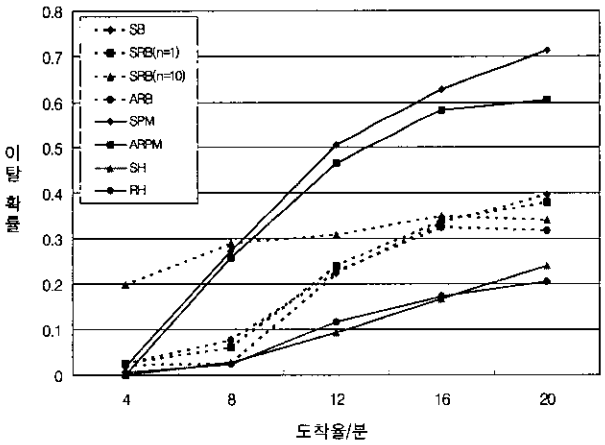
〈표 1〉 모의실험 파라미터

파라미터	값		비 고
비디오 개수(n)	100개		
서버 스트림 용량	400		
인기비디오 개수(k)	3		하이브리드, 피기백 합병
비디오 디스플레이 율	느린율(S_{min})	28.5 프레임/초	
	정상율(S_n)	30 프레임/초	
	빠른율(S_{max})	31.5 프레임/초	
일괄 처리 윈도우(T)	3분		일괄처리
	2분		하이브리드
도착율/분 (λ)	4, 8, 12, 16, 20 (포아송)		
비디오 길이(L)	100분		
격차해소 윈도우(W)	9.5분		하이브리드, 피기백 합병
이탈시간	3분~5분(랜덤)		
예약가중치 (ω_h)	0.6		하이브리드
예약가중치 (ω_p)	2		예약기반 피기백킹

여기서 비디오 i의 인기도는 Zipf의 법칙[9]을 따른다고 가정한다. 비디오 관람 패턴에 대한 통계에 의하면 특정 비디오에 대한 인기도가 편중되는 것으로 나타난다[15]. 이런 지역성(locality)은 Zipf의 법칙을 사용하여 표현할 수 있다[16, 17]. Zipf의 법칙에 의하면 M 비디오 중에서 n번째로 인기 있는 비디오를 선택할 확률은 C/n이고, 이때 C = 1/(1+1/2 +1/3 +……+1/M)이다. MPEG-1으로 압축된 비디오의 압축 비트율은 약 2 MBits/초이고, SCSI-2 버스 인터페이스를 사용하는 디스크들은 최대 약 10 MBits/초의 대역폭을 제공하므로 하나의 디스크는 동시에 5 스트림을 지원할 수 있다. 따라서 80개의 디스크가 8개의 열과 10개의 행으로 구성된 RAID-5 구조의 병렬 디스크인 디스크 배열로 구성하면, 그 병렬 디스크는 최대 400 스트림을 동시에 지원할 수 있다. 그리고 Zipf의 법칙에 의하면 100개의 비디오들 중에서 세 번째 인기 있는 비디오의 요청 확률이 0.071 정도이므로 인기 있는 비디오의 개수를 세 개로 하였다. 일괄처리 윈도우 크기와 서비스 이탈 시간은 클라이언트들의 일반적인 행동을 반영하여 2분~3분, 3분~5분 정도로 각각 설정하였다. 여기서 서비스 이탈 시간은 클라이언트가 대기시간이 너무 길어서 서비스를 포기하는 시간을 의미한다. 비디오의 길이는 비디오 대여점의 평균 비디오 길이인 100분으로 하였으며, 최대 격차해소 윈도우는 수식 (1)에 의해 (31.5 -28.5)/31.5×100 ≒ 9.5분이다. x 단위 시간 떨어진 스트림들이 합병되는 시점은 $t_m = xS_{min}/(S_{max} - S_{min})$으로 계산된다. 그리고 실험 결과는 모의실험의 공정성을 기하기 위하여 서비스를 시작한 후 100분과 200분 사이의 100분간 비디오 요청들에 대한 처리 결과로 각 정책의 성능을 평가하였다. 모의실험에서 평가된 성능 평가 항목들은 비디오 요청 도착율에 따른 이탈 확률, 프레임 절약 백분율 그리고 평균 대기시간이다.

4.2 결과

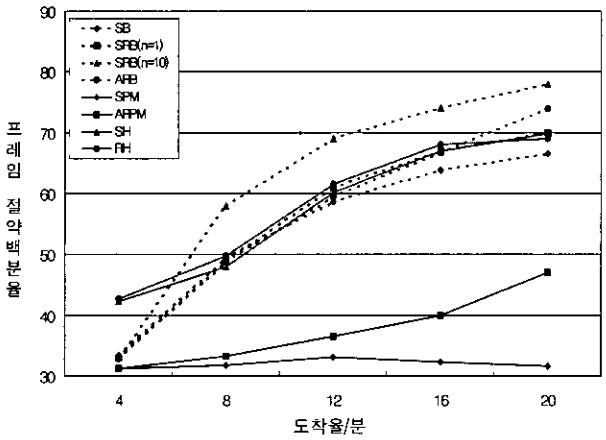
본 논문에서 제안하는 예약기반 하이브리드 디스크 대역폭 절감정책의 성능을 평가하고, 그것을 다른 디스크 대역폭 절감 알고리즘들의 성능과 비교한다.



(그림 7) 비디오 요청 도착율에 따른 각 정책별 이탈확률

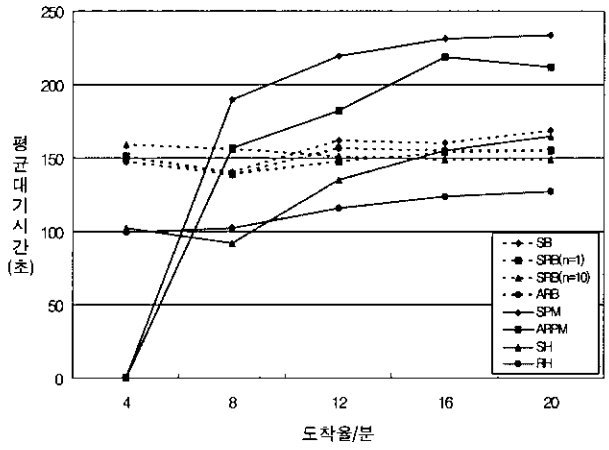
(그림 7)은 비디오 요청 도착율에 따른 각 정책별 이탈확률을 보여준다. 여기서 SRB정책의 n은 인기 있는 비디오의 개수를 의미한다. 그림에서 보는 바와 같이 일괄처리 정책(SB, SRB, ARB), 피기백킹 정책(SPM, ARPM) 그리고 하이브리드 디스크 대역폭 절감정책(SH, RH) 모두 도착율이 낮은 경우는 이탈 확률이 거의 비슷하지만 도착율이 높아질수록 이탈 확률은 증가하고, 각 정책간의 차이도 커짐을 알 수 있다. 특히 피기백킹 정책들(SPM, ARPM)의 경우 다른 정책들에 비해 비디오 요청율에 따른 이탈 확률이 높아짐을 알 수 있다. 예약기반 하이브리드 디스크 대역폭 절감정책에서는 단순 하이브리드 정책과 마찬가지로

격차해소 윈도우 내의 일괄처리 되어 상영중인 입출력 스트림들이 하나로 합병될 뿐만 아니라 서버 과부하 상황에서도 인기 있는 비디오 요청은 즉시 스케줄링 되므로 다른 예약기반 디스크 대역폭 절감 정책들(ARB, ARPM)에 비해 더 많은 입출력 스트림들을 합병시킬 수 있다. 따라서 비디오 요청 도착율에 관계없이 예약기반 하이브리드 디스크 대역폭 절감정책(RH)이 낮은 이탈 확률을 제공함을 알 수 있다.



(그림 8) 비디오 요청 도착율에 따른 각 정책별 프레임 절약 백분율

(그림 8)은 비디오 요청 도착율에 따른 각 정책별 프레임 절약 백분율을 보여준다. 절약된 프레임 백분율은 절약된 프레임의 합을 전체 처리된 프레임의 합으로 나눈 백분율이다. 이때 전체 처리된 프레임의 합은 (*서비스된 비디오의 개수* × *한 비디오의 프레임 개수*)로서 한 비디오의 프레임 개수는 (100 × 60 × 30) = 180,000 프레임이다. 그림에서 일괄처리 정책들이 비디오 요청 도착율이 높아질수록 일괄처리 윈도우 내에 같은 비디오에 대한 요청이 많아지므로 프레임 절약 백분율이 높아짐을 알 수 있다. 한편, 피기백킹 정책들은 다른 정책들에 비해 비디오 요청 도착율에 따른 프레임 절약 백분율이 낮음을 알 수 있다. 특히, SPM 정책의 경우 도착율이 증가할 경우에도 프레임 절약 백분율이 별 차이가 없는데 이는 SPM 정책이 도착율이 낮을 경우 합병 가능 스트림의 도착이 자주 발생하지 않으므로 합병 가능성이 낮고, 도착율이 높아져 많은 스트림 요구들이 도착하여도 전체 비디오 요청이 하나의 큐를 통해 FCFS 정책으로 관리되므로 인기 있는 비디오들에 대한 가용 스트림이 거의 없는 상태이므로 이탈 대기 시간 내에 스케줄링을 받기가 어려워지기 때문이다. 그러므로 도착율에 상관없이 프레임 절약백분율이 31~33% 정도의 일정한 수준을 유지한다. 그러나 ARPM 정책의 경우 낮은 도착율의 경우는 SPM 정책과 동일하지만 높은 도착율의 경우 인기 있는 스트림을 위한 가용 입출력 스트림이 많이 확보되어 있는 상태이므로 합병의 가능성이 SPM 정책에 비해 훨씬 높아진다. 그러므로 ARPM 정책은 도착율이 높을수록 프레임 절약 백분율이 높아짐을 알 수 있다. 또한 단순 하이브리드 정책의 프레임 절약 백분율은 단순 일괄처리 정책과 비슷하지만 비디오 서버 부하가 높아질수록 프레임 절약 백분율이 좋아짐을 알 수 있다. 일반적으로 예약기반 하이브리드 디스크 대역폭 절감정책이 프레임 절약 백분율이 높고, 비디오 서버가 과부하인 경우 ARB 정책이 하이브리드 정책에 비해 프레임 절약 백분율이 조금 더 높게 나타남을 알 수 있다. 이러한 결과는 일괄 처리 정책에서는 일괄 처리 간격을 3분을 적용하였고 하이브리드 정책에서는 대기시간을 줄이기 위하여 일괄 처리 간격을 2분으로 제한하였기 때문이다.



(그림 9) 비디오 요청 도착율에 따른 각 정책별 평균 대기시간

(그림 9)는 비디오 요청 도착율에 따른 각 정책별 평균 대기시간의 변화를 나타낸다. 평균 대기시간은 요청된 비디오의 대기시간의 합을 총 요청된 비디오의 개수로 나누어 계산하였다. 비디오 서버 부하가 낮을 경우 피기백킹 정책은 서버 스트림 용량이 있으면 즉시 스케줄링 되므로 대기시간이 짧다. 그러나 비디오 요청 도착율이 높아질수록 서비스 대기시간이 점점 길어진다. 예약기반 하이브리드 디스크 대역폭 절감정책은 비디오 부하가 낮은 경우만 피기백킹 정책보다 대기시간이 높고, 일반적으로 낮은 일정한 서비스 대기시간을 제공한다. 이는 일괄처리 정책이 낮은 비디오 서버 부하에서 큰 일괄처리 윈도우로 인하여 비디오 요청 서비스 대기시간이 길어지는 단점이 있고 피기백킹 정책이 높은 비디오 서버 부하에서 비디오 요청에 대한 긴 대기시간으로 인하여 서비스 이탈 확률이 높아진다는 단점이 있기 때문이다. 하지만 예약기반 하이브리드 디스크 대역폭 절감정책에서는 인기 있는 비디오를 위한 서버 스트림 용량을 예약하고, 위의 두 정책들의 단점들을 극복하기 위하여 일괄처리 윈도우를 줄이는 대신에 최대 격차해소 윈도우 내에서 일괄처리 상영되는 프레임들을 다시 합병함으로써 좋은 성능을 얻을 수 있었다고 본다.

5. 결 론

주문형 비디오 시스템에서 서비스는 다수의 클라이언트들이 요청을 하면 제한된 자원으로 연속 매체들의 실시간 처리를 보장하면서 보다 많은 수의 클라이언트들에게 서비스를 제공할 수 있어야 한다. 이를 위해 멀티미디어 서비스를 원하는 클라이언트의 요청을 일정시간 수집하였다가 한꺼번에 클라이언트들에게 서비스를 제공하는 일괄처리 정책과 동일한 객체에 대한 입출력 스트림이 하나로 합병될 때까지 진행중인 스트림의 디스플레이율을 조정하는 피기백킹 정책 등이 제안되었다. 그러나 일괄처리 정책에서는 긴 스트림 요청 지연시간이 초래되고, 피기백킹 정책에서는 디스크 대역폭의 낭비를 가져온다. 본 논문에서는 일괄처리 정책과 피기백킹 정책의 장점들만 수용하고, 인기 있는 비디오 요청들이 가능한 한 스케쥴되도록 비디오 서버의 입출력 스트림 용량을 별도로 예약해 두는 예약기반 하이브리드 디스크 대역폭 절감정책을 제안하고, 그것의 성능을 시뮬레이션을 통하여 분석하여 기존의 일괄처리 정책과 피기백킹 정책의 성능과 비교 평가하였다. 그 결과 성능 평가 항목인 비디오 요청 도착율에 따른 클라이언트의 이탈 확률, 클라이언트의 프레임 절약 백분율 그리고 클라이언트의 평균 대기시간에서 본 논문에서 제안하는 예약기반 하이브리드 디스크 대역폭 절감정책이 일괄처리 정책이나, 피기백킹 정책에 비해 우수한 성능을 보였고, 주문형 비디오 시스템의 성능뿐만 아니라 서비스 품질(QoS)도 향상시키는 우수한 디스크 대역폭 절감정책이라는 것을 확인할 수 있었다. 특히 주문형 비디오 시스템에서 입출력 스트림 스케줄링 정책으로 본 논문에서 제안하는 예약기반 하이브리드 알고리즘을 사용하여 낮은 서비스 이탈 확률과 일정한 서비스 평균 대기시간을 제공할 수 있음을 알 수 있었다. 따라서 예약기반 하이브리드 알고리즘을 주문형 비디오 서버의 입출력 스트림 스케줄링 정책으로 사용하면 주문형 비디오 서버의 성능뿐만 아니라 클라이언트의 QoS도 향상시킬 수 있을 것이다. 향후 연구과제는 비디오 서버의 부하에 상관없이 적절한 수준의 이탈율을 유지하는 동적 일괄처리에 관한 것이다.

참 고 문 헌

[1] L. Golubchik, J. Lui, and R. Munts "Reducing I/O Demand in Video-On-Demand Storage Servers," *In Proceedings of ACM Sigmetrics '95*, pp.25-36, May, 1995.

[2] A. Dan, D. Sitaram, and P. Shahabuddin "Scheduling Policies for an On-Demand Video Server with Batching," *In Proceedings of the 2nd ACM Multimedia Conference*, pp.25-32, 1994.

[3] M. Kamath, K. Ramamritham, and D. Towsley, "Continuous Media Sharing in Multimedia Database Systems," Department of Computer Science, University of Massachusetts, Technical Report 94-11, Feb. 1994.

[4] A. Dan, D. M. Dias, R. Mukherjee, and D. Sitaram, "Buffering and Caching in Large-Scale Video Servers," IBM Research Division, Technical Report RC 19903, Jan. 1995.

[5] H. Shachnai and P. S. Yu, "An Analytical Study of Multimedia Batching Schemes," IBM Research Division, Technical Report RC 20662, Dec. 1996.

[6] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, "The maximum Factor Queue Length Batching Scheme for Video-on-Demand Systems," IBM Research Division, Technical Report RC 20621, Nov. 1996.

[7] A. Dan, D. Sitram, and P. Shahabuddin, "Dynamic Batching Polices for an On-Demand Video Server," *Multimedia Systems*, Vol.4, No.2, pp.112-121, June, 1996.

[8] C. Arrarwal, J. Wolf, and P. S. Yu, "On Optimal Piggyback Merging Polices for Video-On-Demand Systems," IBM Research Division, Technical Report RC 20337, Feb. 1996.

[9] G. K. Zipf, *Human Behavior and the Principles of Least Effort*, Addison-Wesley, 1949.

[10] H. Shachnai, P. S. Yu, "Exploring wait tolerance in effective batching for video-on-demand scheduling," *Multimedia Systems*, Vol.6, No.6, Nov. 1998.

[11] K .A. Hua, Y. Cai, S. Sheu, "Patching : A Multicast Techniques for true Video-On-Demand Services," *ACM Multimedia '98*, 1998.

[12] J. Gafsi and E. W. Biersack, "Impact of buffer Sharing in Multiple Disk Video Server architecture," *In Proceedings in the 6th Open Workshop on High Speed Networks.* October, 1997.

[13] Kevin C. Almeroth, Mostafa H. Ammar, "The Use of Multicast Delivery to Provide a Scalable and Interactive Video-on-Demand Service," *IEEE Journal on Selected Areas in Communications*, Vol.14, No.6, pp.1110-1122, August, 1996.

[14] N. L. S. fanseca, R. A. Facanha, "The Look-Ahead-Maximize-Batch Batching Policy," *In Proceeding of GLOBECOM '99*, pp.354-358, Dec. 1999.

[15] T. Chiueh, M. Vernick, C.Venkatramani, "Performance Evaluation of Stony Brook Video Server," ECSL-TR-24, February, 1997.

[16] A.Chervenak, D. Patterson, and R. Katz, "Choosing the Best Storage System for Video Service," *In Proc. of ACM Multimedia 95*, pp.109-119, Aug. 1995.

[17] A. Dan and D. Sitaram, "A Generalized Caching Policy for Mixed Interactive and Long Video Workloads," *IBM*

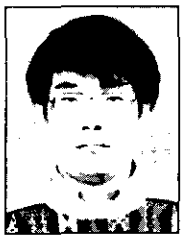
Research Report, RC 20206, Yorktown Heights, NY. 1995.
[18] 이경숙, 배인한, "인기 있는 비디오를 위한 적응적 예약기반 일괄처리 정책의 설계 및 평가", 한국정보처리학회논문지 제6권 제10호, pp.2790-2796, 1999.
[19] 배인한, 이경숙, "비디오 서버를 위한 적응적 예약기반 피기백킹 알고리즘의 설계 및 평가", 한국정보처리학회논문지 특집호 제7권 제2S호, pp.656-665, 2000.

오 선 진

e-mail : sjoh@telcom.semyung.ac.kr
1981년 한양대학교 공과대학(공학사)
1987년 미국 Wayne 주립대학교 컴퓨터과학과(Post Bachelor)
1989년 미국 Detroit 대학교 대학원 컴퓨터과학과(이학석사)
1993년 미국 Oklahoma 주립대학교 대학원 컴퓨터과학과 박사과정
1999년 대구가톨릭대학교 전산통계학과 전자계산전공(이학박사)
1994년~2000년 선린대학 전자계산과 조교수/학과장
2000년~현재 세명대학교 전산정보학부 정보통신학과 조교수
관심분야 : 분산 멀티미디어시스템, True VOD 시스템, 이동 컴퓨팅, 인터넷 컴퓨팅

이 경 숙

e-mail : g6721001@cuth.cataegu.ac.kr
1990년 대구효성여자대학교 수학교육학과 (학사)
1993년 대구가톨릭대학교 대학원 전산통계학과 전산전공 (석사)
2000년 대구가톨릭대학교 대학원 전산통계학과 전산전공 (박사)
관심분야 : 멀티미디어시스템, 고성능 주문형 비디오시스템



배 인 한

e-mail : ihbae@cuth.cataegu.ac.kr
1984년 경남대학교 전자계산학과 (학사)
1986년 중앙대학교 전자계산학과 (석사)
1990년 중앙대학교 전자계산학과 (박사)
1996년~1997년 Dept. of Computer and Information Science, The Ohio State University, USA (Post Doctor)
1989년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 부교수
관심분야 : 이동 무선망, 무선 인터넷, Ad Hoc 망, 분산시스템, 멀티미디어시스템