인기 있는 비디오를 위한 적응적 예약기반 일괄처리 정책의 설계 및 평가

이 경 숙† · 배 인 한††

요 약

주문형 비디오 시스템에서 비디오 서버의 입출력 대역폭은 지연시간을 증가시키는 원인이 되는 중요한 자원이다. 공유를 통하여 비디오 서버의 입출력 요구를 감소시키는 다수의 방법들: 일괄처리, 브리징, 피기백킹이 사용되고 있다. 일괄처리는 일괄처리 윈도우 동안에 다른 비디오들에 대한 요청들을 지연시켜 현재 일괄처리 윈도우 동안에 도착하는 같은 비디오에 대한 많은 요청들을 같은 스트림을 사용하여 서비스한다. 본 논문에서는 비디오 서버 부하에 따라 인기 있는 비디오를 위한 비디오 서버 용량을 동적으로 예약하는 적응적 예약기반 일괄처리 정책을 제안한다. 제안된 정책의 성능을 시뮬레이션으로 평가하고, 단순 일괄처리 정책, 정적 예약기반 일괄처리 정책과 비교한다. 그 결과, 적응적 예약기반 일괄처리 정책이 단순 일괄처리 정책과 단순 예약기반 일괄처리 정책에 비해 서비스 율과 평균 대기시간을 향상시킴을 알 수 있었다.

Design and Evaluation of an Adaptive Reservation-based Batching Policy for Popular Videos

Kyung-Sook Lee† · Ihn-Han Bae††

ABSTRACT

In video-on-demand systems, the I/O bandwidth of video servers is the critical resource which contributes to increase in latency. Several approaches: batching, bridging, piggybacking are used to reduce the I/O demand on the video server through sharing. Batching delays the requests for the different videos for a batching window so that more requests for the same video arriving during the current batching window may be served using the same stream. In this paper, we propose an adaptive reservation-based batching policy which dynamically reserves video server capacity for popular videos according to video server loads. The performance of the proposed policy is evaluated through a simulation, and is compared with simple batching and static reservation-based batching policies. As the result, we know that the adaptive reservation-based batching policy more improves service ratio and average waiting time than simple batching and simple reservation-based batching polices.

1. 서 론

최근 정보통신 분야의 기술적인 발전으로 주문형 비디오, 홈쇼핑 등과 같은 여러 가지 주문형 멀티미디어 시스템들이 구현되고 있다. 오늘날의 정보 시스템은 단순히 커다란 멀티미디어 객체를 저장하고 검색하는 것뿐만 아니라 객체를 일정한 대역폭에서 계속적으로 제공하는 엄격한 실시간 요구사항을 만족시켜야 한다. 멀티미디어 시스템은 교육용 어플리케이션, 오락 기술, 도

† 준 회 원 : 대구효성가톨릭대학교 대학원 전산통계학과
†† 정 회 원 : 대구효성가톨릭대학교 전자정보공학부 교수
논문접수 : 1999년 3월 13일, 심사완료 : 1999년 9월 17일

서관 정보 시스템 등에서 중요한 역할을 하고 있으며, 이러한 시스템에서 가장 중요한 것은 다중 요청을 처리하는 방법이다. 즉, 사용자들은 비디오 등과 같은 객체를 요청하고, 적절한 지연시간 내에 요청한 객체를 관람하기를 희망한다. 여기서 지연시간은 요청이 도착한 시점부터 시스템이 디스크로부터 객체 읽기를 초기화하는 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 읽은 내용을 스케줄 하기 위한 불충분한 버퍼 공간, 불충분한 디스크 기억장치 등의 요인으로 발생한다. 그러한 지연 요소 중에서 입출력 대역폭(I/O bandwidth)은 매우 중요한 자원이므로 공유를 통하여 저장 서버의 입출력 요구를 감소시켜 동시에 서비스할 수 있는 요청 사용자들을 증가시키는 여러가지 접근 방법들이 다음과 같이 제안되고 있다[2].

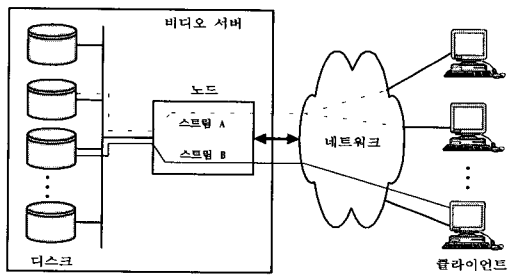
- 일괄처리(batching) : 일정한 일괄처리 윈도우동안 도착하는 동일한 객체에 대한 요청들을 묶어서 단일 입출력 스트림으로 전체 그룹을 서비스하는 방법이다.
- 브리징(bridging) : 중앙 처리기의 메모리를 버퍼로서 이용하는 방법이다. 특정 비디오에 대해 고정된 숫자의 프레임이 버퍼링 된다면, 대응하는 시간 간격 안에 발생하는 비디오에 대한 어떤 요청은 디스크가 아닌 버퍼에서 읽혀질 것이다. 그러나 브리징은 많은 양의 버퍼 공간을 요구하는 단점이 있다.
- 피기백(piggyback) : 동일한 객체에 대한 입출력 스트림이 하나로 병합될 때까지 진행중인 스트림의 디스플레이율을 조정하는 기법이다.

일괄처리 정책에서 매 일괄처리 윈도우마다 인기 있는 비디오에 대한 다수의 요청들이 대기 큐에 항상 존재할 것이다. 그러므로 인기 있는 비디오에 대한 요청들이 매 일괄처리 윈도우마다 스케줄 되어야 많은 비디오 요청들이 일괄처리 되어 하나의 입출력 스트림으로 서비스되므로 많은 비디오 프레임이 절약될 뿐만 아니라 많은 비디오 요청들이 스케줄 되어 관람자가 과도한 대기시간으로 서비스를 포기하는 서비스 이탈율도 감소할 것이다. 따라서 본 논문에서는 인기 있는 비디오들에 대한 요청들이 매 일괄처리 간격마다 스케줄 될 수 있도록 인기 있는 비디오들을 위한 서버 용량을 비디오 요청 도착율에 따라 동적으로 예약해두는 적응적 예약기반 일괄처리 정책을 제안하고, 그것의 성능을 시뮬레이션을 통하여 평가하고, 그리고 단순 일괄처리 정책, 정적 예약 기반 일괄처리 정책과 그 성능을 비교한다.

본 논문의 구성은 다음과 같다. 2장에서는 일괄처리 정책에 대한 관련 연구를 살펴보고, 3장에서는 본 논문에서 제안하는 인기 있는 비디오를 위한 적응적 예약기반 일괄처리 정책을 제안하고, 4장에서는 제안하는 적응적 예약 기반 일괄처리 정책의 성능을 시뮬레이션 통하여 평가하고, 그리고 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

최근 통신과 압축 기술의 발전으로 집중화된 서버 그룹으로부터 비디오를 재생하는 지리적으로 분산된 클라이언트들로 구성된 주문형 비디오 응용들이 구현되고 있다. 그러한 시스템의 일반적인 구조는 (그림 1)과 같다. 여기서 두 클라이언트의 요청이 일괄 처리되어 하나의 스트림 A로 스케줄 되고 네트워크에서 메시지가 각 클라이언트로 멀티캐스트 되고 있음을 보여준다.



(그림 1) 멀티미디어 비디오 서버 환경

서버와 네트워크에서 충분한 자원이 예약되어 있어야 스트림의 연속적인 배달을 유지할 수 있다. 서버에서의 병목현상은 디스크 대역폭이나 CPU 용량 때문에 발생한다. 각 디스크와 디스크 배열은 고정된 개수의 병행 비디오 스트림만을 지원할 수 있으므로 서버에 의해 동시에 지원될 수 있는 최대 비디오 스트림의 개수에 대한 제한이 있다. 이것을 서버 스트림 용량이라 한다[1]. 그리고 ATM과 같은 많은 통신 네트워크는 서버에서의 추가적인 오버헤드 없이 같은 메시지를 다수의 클라이언트에게 전송할 수 있는 멀티캐스트 기능을 갖추고 있

다. 이러한 특징은 주어진 클라이언트들을 지원하기 위하여 서버로 요청되는 스트림의 개수를 감소시키는데 이용될 수 있다. 예를 들어, 두 개의 클라이언트가 짧은 시간 내에 같은 비디오를 요청했다면, 첫 번째 클라이언트에 대한 재생을 연기하여 같은 서버 스트림으로 두 요청을 만족시킬 수 있다. 일반적으로 짧은 시간내의 같은 비디오에 대한 다중 클라이언트들의 요청은 함께 묶어서 하나의 스트림으로 서비스할 수 있는데 이러한 것을 일괄처리라 한다[3].

비디오에 대한 일괄처리 윈도우는 그 비디오에 대한 모든 요청들이 모아지고, 일괄처리 윈도우의 끝에서 하나의 스트림으로 서비스되는데 걸리는 시간으로 정의한다. 일괄처리 윈도우가 커지면 요청되는 서버 용량은 감소하지만 평균 클라이언트 대기시간은 증가한다. 따라서 일괄처리를 통한 서버 용량의 감소와 대기시간, 이탈율의 증가간에는 상반관계가 있다.

비디오 요청들을 묶어서 하나의 I/O 스트림으로 처리하는 여러 가지 일괄처리 정책들이 있다. 이러한 일괄처리 정책은 크기에 의한 일괄처리와 시간에 의한 일괄처리로 나눌 수 있다. 크기에 의한 일괄처리는, 각 비디오 j에 대해 미리 정해진 요청 개수 B_j를 일괄처리 윈도우라 하고, λ_j를 요청 도착율이라 하자. 비디오 j에 대해 B_j 만큼의 요청 개수가 누적되면 비디오 j를 위해 입출력 스트림을 초기화한다. 따라서 비디오 j에 대해 감소되는 입출력 스트림의 개수는 B_j-1이다. $E[N_j]$를 비디오 j의 요청된 I/O 스트림의 개수에 대해 일괄처리에 의해 감소되는 예상 스트림의 개수라 하고, L_j를 비디오 j에 대한 각 요청에 의해 발생되는 지연시간을 나타내는 랜덤 변수라 할 때, 예상 감소 스트림 개수와 예상 지연시간은 다음과 같다[2].

$$E[N_j] = B_j - 1$$

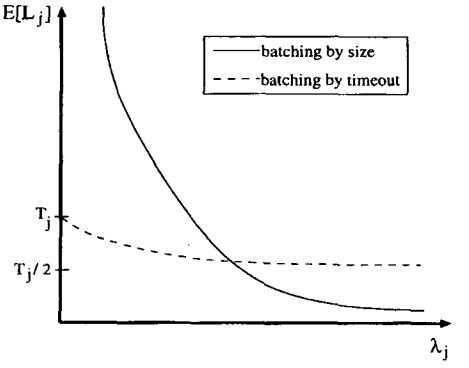
$$E[L_j] = \frac{1}{B_j}\sum_{i=1}^{B_j}\frac{B_j - i}{\lambda_j} = \frac{B_j - 1}{2\lambda_j}$$

비록 크기에 의한 일괄처리 정책이 저장 서버의 I/O 요청을 감소시키지만, 적절한 도착율 보다 낮은 요청에서 긴 지연시간이 발생할 수 있다. 시간에 의한 일괄처리는 시간 단위로 일괄처리 윈도우를 설정하는 것이다. 하나의 요청이 저장 서버에 도착하고 비디오 j에 대한 요청이 존재하지 않을 때 타이머가 설정된다. 시스템은 타이머를 초기화한 후 T_j 단위 시간 동안에 저장 서버에 I/O 요청을 제출한다. T_j 단위 시간 동안에 도착한 같은 비디오에 대한 요청들이 일괄 처리되고 타이머가 만기되었을 때 서비스된다. 비디오 j에 대한 요청 도착 프로세스가 도착율 λ_j를 갖는 포아송이라 가정하면, $E[N_j] = \lambda_j T_j$이다.

각 요청에 의해 발생되는 예상 대기시간을 평가하기 위하여, 그 시스템을 상수 준비시간(T_j)과 결정적 서비스 분포를 갖는 M/G/1 큐로 볼 수 있다. 이런 종류의 시스템에 대한 예상 지연시간은 다음과 같다[2].

$$E[L_j] = \frac{T_j(2 + \lambda_j T_j)}{2(1 + \lambda_j T_j)}$$

$E[N_j] = \lambda_j T_j$이므로 적당히 높은 요청 도착율에서 엄청난 저장 서버의 I/O 요청을 감소시킬 수 있다. 이것은 각 비디오 j에 대한 요청들의 일괄처리로 인한 지연이 T_j 단위 시간 보다 크지 않기 때문에 주문형 비디오 응용에 적당한 정책이다. (그림 2)는 두 가지 일괄처리 정책에 대한 예상 지연시간을 보여준다[2].



(그림 2) 배칭 정책들에 대한 예상 지연시간

좋은 비디오 스케줄링 정책은 일괄처리 윈도우뿐만 아니라 관람자의 이탈 확률과 대기시간을 고려해야 한다. 일괄처리를 위한 두 가지 일반적인 스케줄링 정책은 가장 긴 대기 요청을 갖는 비디오를 스케줄하는 FCFS(first come first served) 정책과 대기 요청들의 최대 개수를 갖는 비디오를 선택하는 MQL(maximum queue length) 정책이 있다. MQL은 일괄처리 되는 요청의 개수를 최대로 하기 위하여 큐 길이만 고려함으로써 인기 있는 비디오 스케줄링에 너무 적극적인 반면에, FCFS는 이탈을 줄이기 위하여 도착시간에 초점을 맞추고 큐 길이를 완전히 무시함으로써 MQL의 반대 효과를 가진다. C. Aggarwal의 연구[1, 2]는 factored queue

length의 개념을 도입하여 maximum factored queue length를 갖는 비디오를 스케줄하는 일괄처리 정책인 MFQ를 제안하였다. Factored queue length는 다른 비디오의 큐 길이에 구별하는 가중 요소를 적용함으로써 얻어진다. 여기서 factored queue length는 비디오의 큐 길이를 그것의 상대적 요청 회수의 제곱근으로 나눈 것으로 정의하였다. 그리고 MFG, FCFS, MQL을 시뮬레이션한 결과, MFQ가 평균 지연시간, 이탈율, 공평성에서 우수한 실험 결과를 보였다.

A. Dan의 연구[3]에서는 FCFS 정책을 확장한 FCFS-n 정책을 설명하였다. FCFS-n 정책에서는 서버 용량의 일부분이 n개의 인기 있는 비디오에 대한 요청들을 일괄처리하기 위하여 예약되고 선할당되어 진다. 인기 있는 비디오를 위해 서버 용량을 선할당함으로써 인기 있는 비디오에 대해 초과할 수 없는 최대 대기시간을 제공할 수 있고, 그리고 인기없는 비디오에 대한 요청이 인기 있는 비디오에 대한 서비스에 간섭하지 않으므로 상대적으로 작은 서버 용량으로 높은 수용 확률을 보장하였다.

3. 적응적 예약기반 배칭

본 논문에서 시간에 의한 단순 일괄처리 정책에 인기 있는 비디오를 위해 동적으로 서버 용량을 예약하는 적응적 예약기반 일괄처리 정책을 제안한다. 본 논문에서는 서버의 최대 디스크 대역폭을 논리적으로 서버 용량이라 한다. 단순 일괄처리에서는 비디오 요청들에게 도착순으로 서버 용량을 할당하므로 비디오 요청 도착율이 어느 정도 높아지면 가용 서버 용량이 없어 도착한 비디오 요청들은 블록되어진다. 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 스케줄 할 수 있을 것이다. 그러나 가용 서버 용량이 없으면 가용 서버 용량이 생길 때까지 기다려야 하고, 결국 비디오 서비스 대기시간이 길어지고 사용자의 서비스 이탈율이 증가하여 고품질의 주문형 비디오 서비스를 제공할 수가 없게 된다. 따라서 본 논문에서는 단순 일괄처리에서의 이러한 문제점을 해결하기 위하여 인기 있는 비디오에 대한 요청들이 매 일괄처리 윈도우마다 스케줄 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 작을 경우, 매 일괄처리 윈도우마다 비디오 요청이 발생하는 인기 있는 비디오가 스케줄되지 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. 따라서 본 논문에서는 비디오 서버의 부하에 따라 인기 있는 비디오의 개수와 그 비디오들을 스케줄하기 위해 예약되는 서버 용량이 동적으로 조절되는 적응적 예약기반 일괄처리 정책을 제안한다.

본 논문의 적응적 예약기반 일괄처리 정책에서는 먼저 서버의 비디오 요청 도착율에 따라 인기 있는 비디오를 위한 예약 서버 용량을 결정한다. 비디오 요청 도착율이 λ일 때 서버 용량이 예약되는 인기 있는 비디오의 개수(n)와 그 비디오들의 스케줄을 위하여 예약되는 서버 용량(SC)은 식 (1)과 같이 계산할 수 있다.

$$r_i = P_i \times \lambda \times T$$

$$n = r_i \geq 2\text{인 비디오의 개수}$$

$$SC = n \times \frac{L}{T} \tag{1}$$

여기서 r_i는 배칭 윈도우 동안의 i-번째 비디오의 요청 도착율, P_i는 i-번째 비디오의 인기도, T는 배칭 윈도우 그리고 L은 비디오의 길이를 나타낸다. 그리고 매 일괄처리 윈도우마다 같은 비디오에 대한 최소한 두개의 비디오 요청이 있어야 그 요청들을 묶어서 하나의 스트림으로 서비스하여 서버 용량이 절약할 수 있으므로 $r_i \geq 2$인 비디오를 인기 있는 비디오로 한다. 인기 있는 비디오를 위한 서버 용량이 예약되면, 인기 있는 비디오에 대한 요청은 예약 서버 용량에서 스케줄 되고, 비인기 비디오에 대한 요청은 비예약 서버 용량에서 스케줄된다. 본 논문에서 제안하는 적응적 예약기반 일괄처리 알고리즘의 구조는 (그림 3)과 같다.

적응적 예약기반 일괄처리 알고리즘에서는 먼저 식 (1)을 사용하여 비디오 요청 도착율에 따라 인기 있는 비디오를 위한 서버 용량을 예약한다. 그리고 비디오 요청이 도착하면 도착한 비디오 요청을 대기 큐에 넣고, 대기 큐의 선두에 있는 비디오 요청의 일괄처리 윈도우가 만기가 되면 그 비디오 요청의 스케줄링을 시도한다.

```
Algorithm Adaptive Reservation-based Batching
  Begin
    Reserve server capacity for popular videos according to
      arrival rate of video 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 n-hottest video
        If reserved server capacity is not empty
          Begin
            Batch the same video requests in waiting queue
              into a single video request;
            Schedule the video request with a single stream in
              reserved server capacity;
          End
        Else
          Block the video request;
      Else
        If unreserved server capacity is not empty
          Begin
            Batch the same video requests in waiting queue
              into a single video request;
            Schedule the video request with a single stream in
              unreserved server capacity;
          End
        Else
          Block the video request;
  End
```

(그림 3) 적응적 예약기반 일괄처리 알고리즘

그 요청된 비디오가 인기 있는 비디오에 속하고 예약된 서버 용량이 남아 있으면 대기 큐의 같은 비디오 요청들을 단일 비디오 요청으로 묶고, 그 비디오 요청을 예약된 서버 용량에서 단일 스트림으로 서비스한다. 예약된 서버 용량이 없으면 그 비디오 요청은 블록된다. 그리고 대기 큐의 선두에 있는 비디오 요청이 인기 없는 비디오이고 비예약 서버 용량이 남아 있으면 대기 큐의 같은 비디오 요청들을 단일 비디오 요청으로 묶고, 그 비디오 요청을 비예약 서버 용량에서 단일 스트림으로 서비스한다. 비예약 서버 용량이 없으면 그 비디오 요청은 블록된다.

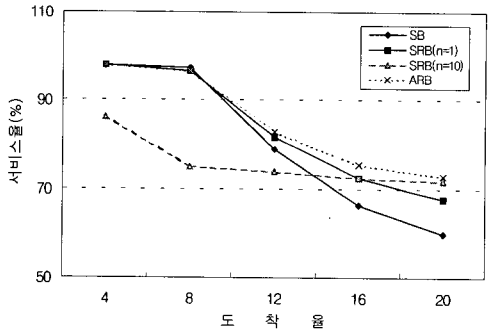
4. 시뮬레이션 및 성능 평가

본 논문에서는 시뮬레이션을 통하여 단순 일괄처리 정책(SB, Simple Batching), 정적 예약기반 일괄처리 정책(SRB, Static Reservation-based Batching)과 적응적 예약기반 일괄처리 정책(ARB, Adaptive Reservation-based Batching)의 성능을 비교하고 평가한다. 시뮬레이션 파라미터는 <표 1>과 같다. 여기서 비디오 i의 인기도는 Zipf의 법칙[7]을 따른다고 가정한다. 실험 결과는 시뮬레이션의 공정성을 기하기 위하여 서비스를 시작한 후 100분과 200분 사이의 100분간의 비디오 요청들에 대한 처리 결과로 각 정책의 성능을 평가하였다. 여기서 평가된 성능 평가 항목들은 비디오 요청 도착율에 따른 서비스율, 평균 대기시간, 프레임 절약율 그리고 버퍼 사용량이다.

〈표 1〉 시뮬레이션 파라미터

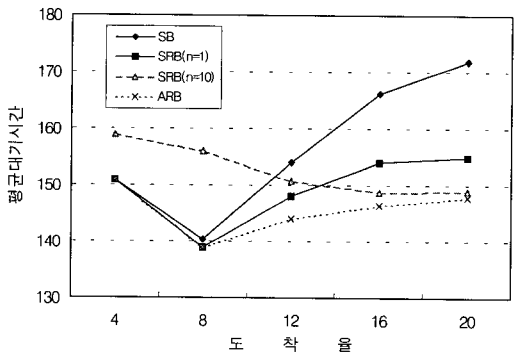
파 라 미 터	값
비디오의 개수	100개
서버 스트림 용량	400
비디오 디스플레이 율	30 프레임/초
배칭 윈도우	3분
도착율/분	4, 8, 12, 16, 20(포아송)
비디오 길이	100분
이탈시간	3분~5분(랜덤)

(그림 4)는 비디오 요청 도착율에 따른 일괄처리 정책별 서비스율을 보여준다. 여기서 서비스율은 전체 도착한 비디오 요청 개수에 대한 스케줄된 비디오 요청 개수의 백분율로 계산되었다. 여기서 비디오 서버가 저부하인 경우에 SB, SRB(n=1), ARB 모두 도착한 비디오 요청들을 거의 모두 스케줄 하였으나 세 정책 모두 비디오 서버의 부하가 높아질수록 서비스율이 떨어짐을 알 수 있다. 그러나 10개의 인기 있는 비디오에 대한 서버 용량을 예약한 정적 예약기반 일괄처리 정책(SRB(n=10))은 너무 많은 서버 용량의 예약으로 인해 예약된 서버 용량의 낭비가 발생하여 가장 낮은 서비스율을 보이고 있다. 그리고 비디오 서버 과부하 상황에서도 본 논문에서 제안하는 ARB가 SB, SRB 보다 높은 서비스율을 보이고 있다. 이것은 SB가 비디오 서버 과부하 상황에서 모든 비디오 요청들이 블록되는데 반해 ARB는 인기 있는 비디오를 위한 서버 용량이 예약되어 있어 서버 과부하 상황에서도 인기 있는 비디오 요청들은 일괄처리 되고 예약된 서버 용량에 의해 스케줄되는 결과이다. 그리고 1개의 인기 있는 비디오에 대해서만 서버 용량을 예약한 정적 예약기반 일괄처리 정책(SRB(n=1))은 서버의 비디오 요청 도착율이 높을 경우 예약 서버 용량이 부족하여 인기 있는 비디오들이 스케줄되지 못하므로 서비스율이 떨어진다. 따라서 비디오 서버의 부하에 따라 인기 있는 비디오의 개수와 그 비디오들을 스케줄하기 위한 예약 서버 용량을 적절히 조절하는 ARB가 가장 좋은 서비스율을 제공함을 알 수 있다.



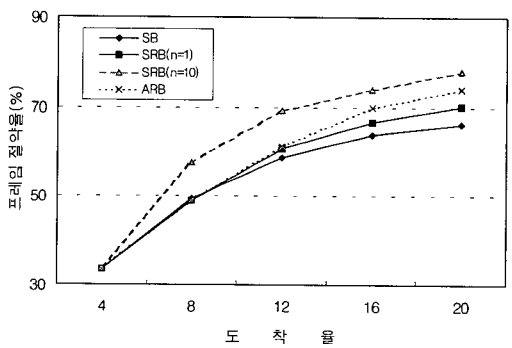
(그림 4) 비디오 요청 도착율에 따른 서비스율

(그림 5)는 비디오 요청 도착율에 따른 일괄처리 정책별 평균 대기시간을 보여준다. 평균 대기시간은 요청 비디오의 대기시간의 합을 총 요청된 비디오의 개수로 나누어 계산하였다. ARB는 SB와 SRB에 비해 더 낮은 평균 대기시간을 보여주고 있다. 비디오 서버 과부하 상황에서 SB에서는 모든 비디오 요청들이 블록되고, SRB(n=1)에서는 예약 서버 용량이 부족하여 인기 있는 비디오들이 블록되므로 평균 대기시간이 길어지고, SRB(n=10)에서는 비디오 서버 저부하에서 예약 서버 용량의 낭비로 인기 없는 비디오 요청들이 블록되어 평균 대기시간이 길어진다. 그러나 ARB는 비디오 요청 도착율에 따라 적절한 개수의 인기 있는 비디오들에 대한 서버 용량이 예약되므로 인기 있는 비디오와 인기 없는 비디오의 요청들이 적절히 스케줄되어 가장 좋은 평균 대기시간을 보여준다.



(그림 5) 비디오 요청 도착율에 따른 평균 대기시간

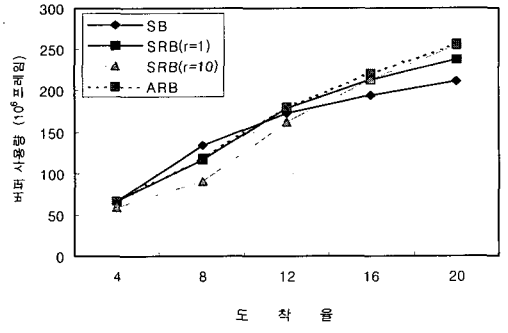
(그림 6)은 비디오 요청 도착율에 따른 일괄처리 정책별 프레임 절약율을 보여준다. 프레임 절약율은 절약된 프레임의 합을 전체 처리된 프레임의 합으로 나눈 백분율이다. 여기서 비디오 요청 도착율이 높아질수록 배칭 윈도우내에 같은 비디오에 대한 요청이 많아지므로 SB, SRB, ARB 모두 프레임 절약율이 높아진다. 그리고 비디오 서버 과부하 상황에서 ARB가 SB에 비해 높은 프레임 절약율을 보이고 있다. 이것은 비디오 서버 과부하 상황에서 단순 일괄처리는 일괄 처리할 수 있는 많은 인기 있는 비디오 요청들이 있으나 가용 서버 용량이 없어 그러한 요청들이 일괄 처리되고 스케줄되지 못하므로 비디오 프레임을 절약할 기회를 놓치게 된다. 그러나 본 논문의 ARB에서는 비디오 서버 과부하 상황에서도 인기 있는 비디오 요청들은 여전히 일괄처리되고 스케줄되므로 많은 비디오 프레임들이 절약된다. 그리고 (그림 4)에서 비디오 요청 도착율이 8일 때 SRB(n=10)의 서비스율이 가장 낮은데 반해 (그림 6)의 프레임 절약율은 가장 높다. 이것은 SRB(n=10)이 인기 있는 비디오 위주로 서버 스트림이 스케줄되었다는 것을 의미한다. 따라서 인기 없는 비디오들에 대한 비디오 요청들은 스케줄되지 못하여 대기시간이 길어지고, 결국 서비스의 이탈이 발생하여 고품질의 서비스를 제공할 수 없다. 그러나 본 논문에서 제안하는 ARB는 서비스율과 프레임 절약율 모두 높아 비디오 서버의 성능뿐만 아니라 사용자의 비디오 관람 대기시간이 감소하므로 QoS(Quality of Service)도 향상됨을 알 수 있다.



(그림 6) 비디오 요청 도착율에 따른 프레임 절약율

(그림 7)은 비디오 요청 도착율에 따른 일괄처리 정책별 버퍼 사용량을 보여준다. 모든 정책들이 비슷한 버퍼 사용량을 보여주나, 비디오 서버 저부하 상황에서 SRB(n=10)의 비디오 요청 서비스율이 다른 정책에 비해 낮으므로(그림 4 참조) 버퍼 사용량이 약간 적고, 비디오 서버 과부하 상황에서 SB의 비디오 요청 서비스율이 다른 정책에 비해 역시 낮으므로(그림 4 참조) 버퍼 사용량이 약간 적다. 그리고 본 논문의 ARB는 다른 정

책들과 비슷한 버퍼 사용량을 사용하여 우수한 비디오 요청 서비스율과 서비스 평균 대기시간을 제공함을 알 수 있다.



(그림 7) 비디오 요청 도착율에 따른 버퍼 사용량

5. 결 론

본 논문에서는 비디오 서버에 도착하는 비디오 요청 도착율에 따라 인기 있는 비디오를 스케줄하기 위하여 비디오 서버 용량을 가변적으로 예약해 두는 적응적 예약기반 일괄처리 정책을 제안하고, 그것의 성능을 시뮬레이션을 통하여 평가하였다. 그리고 제안하는 적응적 예약기반 일괄처리 정책의 성능을 단순 일괄처리 정책 그리고 정적 예약기반 일괄처리 정책과 비교하였다. 그 결과 모든 성능 평가 항목 : 서비스율, 평균 대기시간, 프레임 절약율, 그리고 버퍼 사용량에서 본 논문에서 제안하는 적응적 예약기반 일괄처리 정책이 단순 일괄처리 정책, 정적 예약기반 일괄처리 정책 보다 우수하였다. 따라서 적응적 예약기반 일괄처리 정책은 주문형 비디오 시스템의 성능뿐만 아니라 관람자의 서비스 이탈율과 평균 대기시간이 감소시켜 QoS도 향상시킴을 알 수 있었다. 앞으로의 연구 내용은 다양한 시뮬레이션 환경 하에서의 다양한 입출력 대역폭 감소 정책들간의 성능 평가, 그 정책들간의 QoS에 관한 평가, 그리고 일괄처리 정책과 피기백킹 정책을 혼합한 적응적 예약기반 하이브리드 정책에 관한 연구 등이다.

참 고 문 헌

[1] C. Aggarwal, J. Wolf and P. Yu "On Optimal Piggyback Merging Policies for Video-on-Demand Systems," Technical Report, IBM RC 20337, Feb. 1996.

[2] L. Golubchik, J. Lui, and R. Munts "Reducing I/O Demand in Video-On-Demand Storage Servers," ACM Sigmetrics Conference, pp.25-36, May. 1995.

[3] A. Dan, D, Sitaram, and P. Shahabuddin "Scheduling Policies for an On-Demand Video Server with Batching," Proceedings of the 2nd ACM Multimedia Conference, pp.15-23, 1994.

[4] H. Shachnai, P. Yu "An Analytical Study of Multimedia Batching Schemes," Technical Report, IBM RC 20662, Dec. 1996.

[5] C. Aggarwal, J. Wolf, and P. Yu "The Maximum Factor Queue Length Batching Scheme for Video-on-Demand Systems," Technical Report, IBM RC 20621, Nov. 1996.

[6] H. Schachnai, P.S. Yu, "Exploring wait tolerance ineffective batching for video-on-demand scheduling," Multimedia Systems, Vol.6, pp.382-393, Nov. 1998.

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

이 경 숙

e-mail : g6721001@cuth.cataegu.ac.kr

1990년 대구효성가톨릭대학교 수학교육학과(학사)

1993년 대구효성가톨릭대학교 대학원 전산통계학과(석사)

1998년 대구효성가톨릭대학교 대학원 전산통계학과 (박사과정 수료)

관심분야 : 분산 시스템, 멀티미디어 시스템

배 인 한

e-mail : ihbae@cuth.cataegu.ac.kr

1984년 경남대학교 전자계산학과 (학사)

1986년 중앙대학교 대학원 전자계산학과(석사)

1990년 중앙대학교 대학원 전자계산학과(박사)

1996년~1997년 Dept. of Computer and Information Science, The Ohio State University (Postdoc)

1989년~현재 대구효성가톨릭대학교 전자정보공학부 부교수

관심분야 : 운영체제, 분산 시스템, 멀티미디어 시스템, 이동 컴퓨팅, 이동 에이젼트