

## 주문형 교육 시스템에서 효율적인 스케줄링 기법 설계

김 종 훈 · 한 금 희

jkim@ns.cheju-e.ac.kr · hkumhee@www.cuk.ac.kr

제주교육대학교 컴퓨터교육과 · 가톨릭대학교 컴퓨터공학부

### 요 약

주문형 교육 시스템(EOD)에서 스케줄링 기법은 시스템의 효율을 높이고, 한번의 디스크 검색으로 여러 학습자들의 요구를 만족시키도록 설계되어야 한다. 이에 본 논문에서는 학습자 시스템의 자원을 활용함으로써 서버의 디스크 입출력과 버퍼 소비를 줄이는 스케줄링 기법을 제안하였다. 본 논문에서 제안한 스케줄링 기법은 모든 서비스를 서버가 제공하는 것이 아니라 해당 자료를 저장하고 있는 학습자 시스템으로 하여금 서비스하도록 한다. 그럼으로써 학습자들의 초기지연시간 단축을 가져오고 서버의 처리량이 증가하게 된다. 본 논문에서 제안한 기법과 기존의 기법의 성능을 시뮬레이션으로 비교한 결과 제안한 기법이 효율적인 성능을 나타냈다.

## Design of an Efficient Scheduling Strategy in Education-On-Demand Systems

Jong-Hoon Kim · Kumhee Han

Cheju National University of Education · Catholic University of Korea

### ABSTRACT

Scheduling strategy in the Education-on-Demand system should be designed to service the isochronous requests from several learners with single disk retrieval to improve the performance of the system. In this paper, the scheduling strategy which reduces the disk I/O and buffer consumption by utilizing the resources of the learner's system has been proposed. All the requests from the learner are not serviced from the server but forwarded to the learner's system which preserves the requested information. Therefore, the server can increase the throughput and reduces the initial latency time of the user. The proposed strategy has been compared to the conventional method by simulations and proved to be more efficient.

### 1. 서 론

ATM[1]과 Myrinet[2]와 같은 초고속 통신망이 개발되고 멀티미디어 데이터를 효율적으로 처리할 수 있는 워크스테이션(또는 개인용 컴퓨터)이 출현됨에 따라 주문형 교육 시스템이 고안되고 있다. 주문형 교육 시스템은 교사의 강의를 미리 멀티미디어 서버에 데이터 베

이스화한 다음 학습자들이 원하는 시간에 원하는 강의를 통신망을 통해 서비스하는 시스템이다. 그러나 이와 같은 기술을 현실화하기 위해서는 대용량의 데이터를 효율적으로 저장하고 있다가 보다 많은 학습자들에게 원하는 서비스를 제공하는 서버 기술의 지원이 있어야 한다[3, 4].

주문형 교육 시스템에서는 문자 데이터와는 달리 동일한 교육 비디오 자료에 대한 요구가 많이 발생할 것 이므로 특정 비디오 자료에 대한 요구가 발생했을 때 해당 자료의 내용이 다른 학습자 시스템의 버퍼에 유지되고 있을 확률이 높을 것이다. 그러므로 이를 효율적으로 활용하는 연구가 필요하나 주문형 교육 시스템에서는 이를 고려한 연구는 진행된 바가 없다. 또한 최근 학습자 시스템의 성능이 놀라울 정도로 발전하여 일반적인 분산 시스템 환경에서는 학습자 시스템 자원을 효율적으로 활용하는 연구[5]가 진행되고 있는데 주문형 교육 시스템 환경에서는 이에 대한 연구 결과나 진행이 미진한 상태에 있다.

그러므로 본 논문에서는 요구가 발생했을 때 무조건 서버가 서비스를 제공하는 것이 아니라 해당 교육 비디오를 저장하고 있는 학습자 시스템이 서비스를 제공하는 정책을 제안하였다. 제안한 정책의 효과를 검증하기 위해서 시뮬레이션을 통해 기존 스케줄링 기법과의 성능을 비교하였다. 성능 비교를 통해 본 논문에서 제안한 정책이 효율적인 성능을 나타내는 것을 확인하였다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 관련 연구로 주문형 교육과 기존의 스케줄링 정책들에 대해 살펴본다. 3장에서 본 논문에서 제안한 스케줄링 정책에 대해 설명하고 4장에서는 성능 평가를 위한 실험 환경을 기술하며 실험 결과와 분석 내용을 설명한다. 그리고 5장에서 결론과 향후 연구를 논의한다.

## 2. 관련 연구

본 장에서는 본 논문과 관련된 연구로 주문형 교육과 기존의 요구 스케줄링 기법에 관한 연구에 대해 살펴본다.

### 2.1 주문형 교육

주문형 교육(EID: Education On Demand)이란 일반 사용자들이 마치 VCR 테이프를 보는 것처럼 중앙에 있는 비디오 서비스센터를 통하여 자신이 원하는 영화를 원하는 시간에 얼마든지 즐길 수 있는 시스템을 VOD(Video On Demand)라고 하는 것처럼, 일반 사용자들이 컴퓨터 또는 TV를 사용하여 원하는 교육 내용을 원하는 장소에 통신망으로 받을 수 있도록 하는 통신 서비스이다. 이것은 교육의 주체가 교수자에서 학습

자로 전이되는 과정에서 나타난 용어인데, 지금까지의 교육이 교육을 실시하는 자의 의도에 따라 이루어졌다 면 앞으로의 교육은 학습자 개인이 정확한 자기 평가결과에 따라 내용, 수준, 방법을 선택할 수 있어야 할 것이다. 이를 적절히 지원하기 위해서는 시간과 공간의 제약을 받지 않는 원격교육이나 스스로 학습할 수 있는 코스웨어, 학습자가 이를 이용하고자 할 때 불편함이 없이 학습할 수 있게 하는 학습관리시스템 등이 필요한데, 이를 통칭하여 주문형 교육이라 한다.

### 2.2 기존의 요구 스케줄링 기법

피기백킹[6]은 비디오의 상영속도가 5%정도 빨라지거나 느려지는 것은 사용자가 감지하기 힘들다는 점을 이용한 방법으로 서비스 중에 동일한 비디오에 대한 요구가 들어오면 앞서고 있는 비디오의 속도를 늦추고 새로 시작하는 비디오의 속도를 빠르게 조정하여 두 사용자에게 동일한 스트림으로 서비스하는 기법이다. 그러나 서버가 검색해야 하는 데이터의 양의 변화가 매우 심하게 되어 디스크 입출력 양이 큰 폭으로 변하고, 각기 다른 속도로 검색하는 데에 서버의 능력이 요구되는 문제점이 있다. 또한 널리 이용되는 압축 기법인 MPEG 형식의 파일에는 적용하기가 힘들다는 문제점도 있다.

브릿징[7, 8]은 동일한 비디오에 대한 요구가 있을 때 가장 먼저 온 요구에 대해서는 디스크를 검색하지만 이 검색한 데이터를 메모리에 버퍼링 함으로 뒤에 발생한 요구들에 대해서는 디스크가 아닌 메모리에서 서비스해주는 정책이다. 그러나 이 기법은 매우 많은 양의 메모리 공간을 필요로 하는 단점이 있다.

배칭[9, 10]은 배칭 간격 동안 동일한 사용자 요구들을 모아 하나의 입출력 스트림으로 서비스하는 방법이다. 이러한 배칭은 브릿징이나 피기백킹처럼 큰 메모리의 요구도 없고 서버가 데이터의 검색 속도를 빠르거나 느리게 하는데 능력을 소모하지 않아도 된다는 장점이 있다. 하지만 모든 요구에 동일하게 배칭 간격이 적용되므로 클라이언트가 불필요하게 기다리는 일이 생길 수 있다. 즉 인기 없는 비디오를 요구한 사용자 또는 평일 새벽시간대처럼 사용자가 거의 없는 시간에 요구한 사용자가 이에 해당된다.

체이닝[11]은 클라이언트들의 디스크를 활용함으로써

네트워크 I/O 요구량을 감소시킬 뿐만 아니라 더불어 서버의 자원을 절감하는 데에 목적이 있다. 체이닝에서는 첫 번째로 도착한 시청자의 요구는 서버로부터 서비스 받게 되며 동일한 비디오를 신청한 그 다음 번에 두 번째로 들어온 요구는 첫 번째 시청자인 클라이언트의 디스크로부터 제공받는 것이다. 세 번째 도착한 요구는 물론 두 번째 클라이언트로부터 제공받는 식으로 진행된다. 그러므로 일찍 도착한 시청자의 요구가 배정 간격 내에 늦게 도착하는 요구들을 기다릴 필요가 없게 되는데 이는 클라이언트의 워크스테이션으로부터 직접 멀티캐스트로 서비스 받게 되기 때문이다. 이런 식으로 시청자의 요구는 서버에 더 이상 부담을 준다기보다 공헌하는 역설적인 일면을 지니게 된다. 시청자의 요구가 드물게 들어오면 사실상 체이닝 기법을 사용하기가 어렵게 된다. 체인이 길어지면 더 많은 클라이언트가 같은 비디오 스트림을 공유하게 되므로 바람직하다. 그러나 클라이언트의 디스크 자원을 활용한다는 것은 비현실적인 정책으로 지적할 수 있다.

### 3. 제안한 스케줄링 기법

본 논문에서 가정하는 시스템은 각 학습자를 클라이언트로 하고 교육 서비스 제공자를 서버로 하는 클라이언트-서버 구조이다. 특히 서버는 저장 서버와 관리자로 구성되어 있다. 저장 서버는 교육 비디오 자료를 검색하여 요청한 학습자에게 자료를 보내주는 일을 담당하는 부분으로 요구들에 대한 서비스는 라운드-로빈 방식으로 이루어진다. 관리자는 학습자의 요구들에 대한 관리와 이를 처리하는 부분으로 요구에 대한 서비스가 가능한지를 검사하여 서비스를 지시하거나 큐잉하여 저장 관리한다.

주문형 교육 시스템에서는 일반적인 텍스트 데이터와는 달리 동일한 교육 비디오 자료에 대한 요구가 많이 발생하므로 특정 비디오에 대한 요구가 발생했을 때 해당 비디오 스트림이 다른 학습자 시스템의 버퍼에 유지되고 있을 확률이 높다. 이러한 점을 고려하여 본 논문에서 제안한 기법에서는 요구가 발생했을 때 무조건 서버가 서비스를 제공하는 것이 아니라 해당 비디오 스트림을 버퍼에 저장하고 있는 학습자 시스템이 있는지를 검색하여 있을 경우에 작업을 학습자 시스템에게 넘겨 동작하게 된다. 이렇게 하여 서버의 부하가 감소하

게 되고 더욱 많은 학습자들에 대한 서비스를 제공하게 된다. 최근 시스템의 성능은 놀라울 정도로 발전하여 자신의 버퍼에 저장되어 있는 비디오 스트림을 다른 학습자 시스템에게 보내는 단순한 작업을 처리하는 것은 전혀 문제가 되지 않는다.

서비스하는 교육 비디오의 시작 부분을 버퍼에 저장하고 있는 학습자 시스템이 있는지에 대한 정보를 관리자가 테이블에 유지하고 있다. 관리 테이블은 매우 간단한 구조를 지니고 있다. 만약 200가지 종류의 교육 비디오를 서비스하는 시스템이라면 크기가 200인 두 개의 정수형 배열로 이루어진 테이블 하나만 있으면 된다. 만약 1번 비디오를 10번 클라이언트가 보기 시작하면 TABLE[1][1]에 클라이언트 번호 10을 저장하고 TABLE[1][2]에는 클라이언트가 비디오 자료를 보기 시작한 시간을 저장하면 된다. 그리고 주기적으로 TABLE[i][2]의 내용을 검색하여 비디오 시작 부분이 TABLE[i][1]번 클라이언트의 버퍼에서 쫓겨날 시간이 되면 TABLE[i][1]에 NULL을 저장한다.

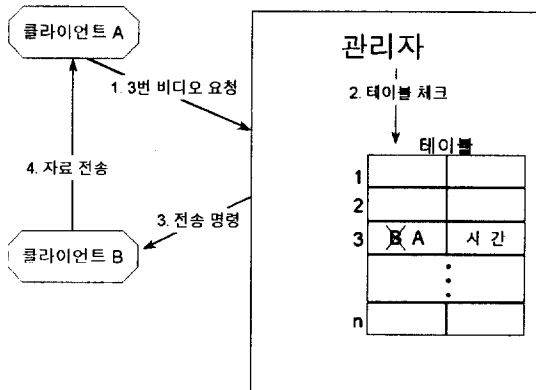
주문형 교육 시스템은 전통적인 시스템과는 달리 접근 형태를 예측할 수 있고 블록 단위로 동작하지 않고 완전한 멀티미디어 객체(예, 교육 비디오) 단위로 동작하므로 본 정책에서 학습자 시스템의 버퍼 교체 정책은 FIFO(First-In First-Out)를 따른다.

제안한 기법의 동작을 요구의 형태에 따라 분류하여 살펴보면 다음과 같다.

- 학습자로부터 요구가 발생했을 때는 요구된 비디오의 시작 부분이 다른 학습자 시스템의 버퍼에 있는지의 여부에 따라 동작이 달라진다.

요구된 비디오의 시작 부분이 다른 학습자 시스템의 버퍼에 있을 경우의 동작은 <그림 1>의 예를 통해 살펴본다. 학습자 A로부터 3번 비디오에 대한 요구가 들어오면 관리자는 관리 테이블을 검색하여 3번 비디오의 시작 부분을 버퍼에 저장하고 있는 학습자 시스템이 있는지를 점검한다. 만약 해당 비디오를 저장하고 있는 학습자 시스템(B)이 있을 경우에 관리자가 자료를 전송하라는 명령을 학습자 시스템 B에게 보낸다. 그러면 학습자 시스템 B에서 해당 비디오 스트림을 학습자 A에게 전송하게 된다. 그리고 관리자는 관리 테이블의 내

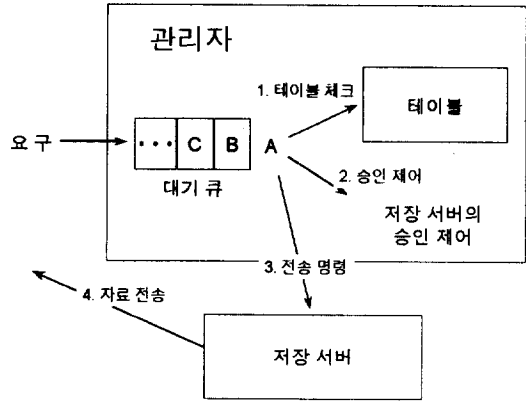
용을 B에서 A로 갱신하고 시간 정보도 갱신한다. 그리고 요구된 비디오의 시작 부분이 다른 학습자 시스템 버퍼에 없을 경우에는 해당 요구를 관리자의 대기 큐에 큐잉한다.



<그림 1> 클라이언트 A가 요구한 3번 교육 비디오가 클라이언트 B의 버퍼에 있을 경우의 동작

• 큐잉 되었던 요구는 주기적으로 대기 큐에서 디큐되어 처리되는데 다음 과정을 따른다.

요구된 교육 비디오의 시작 부분이 다른 학습자 시스템 버퍼에 있을 경우에는 관리 테이블을 검색하여 해당 비디오의 시작 부분을 저장하고 있는 학습자 시스템이 있으면 <그림 1>과 동일하게 처리한다. 그리고 저장 서버가 요구를 처리할 수 있을 때 만약 요구된 교육 비디오의 시작 부분을 저장하고 있는 학습자 시스템이 없을 경우에는 저장 서버가 요구를 승인할 수 있는지를 판단한다. 승인 제어는 묵시적으로 FCFS(First-Come First-Served) 방식으로 처리되지만, 저장 서버는 디스크 대역폭, 스트림 용량, 버퍼 및 네트워크 대역폭 등이 충분할 때 새로운 비디오 스트림의 연속적인 전송을 시작할 수 있는 것이다. 승인이 가능하면 요구를 저장 서버로 전송하여 저장 서버에서 비디오 스트림을 학습자에게 전송하도록 한다. 그리고 관리자는 관리 테이블의 내용을 갱신한다. 그리고 처리가 불가능한 경우에 만약 저장 서버가 요구를 처리할 수 없을 때 즉 자원이 충분하지 않을 경우에는 큐에서 대기한다. <그림 2>은 이 동작을 그림으로 나타낸 것이다.



<그림 2> 디큐된 요구를 처리하는 동작

<그림 3>은 이들 동작을 알고리즘으로 나타낸 것이다.

```

학습자부터 요구 발생
테이블 체크;
if (요구된 비디오의 시작 부분이 다른 학습자 시스템에 존재) {
    학습자 시스템에게 자료전송 요구;
    테이블 갱신;
} else {
    인큐;
}

큐에 요구가 존재
디큐;
테이블 체크;
if (요구된 비디오의 시작 부분이 다른 학습자 시스템에 존재) {
    학습자 시스템에게 자료전송 요구;
    테이블 갱신;
} else if (저장 서버가 서비스 가능) {
    저장 서버가 자료 전송;
    테이블 갱신;
} else {
    큐에서 대기;
}
    
```

<그림 3> 알고리즘

4. 실험

본 장에서는 성능 평가를 위한 실험 환경을 살펴보

고 실험 결과와 그에 따른 분석 내용을 설명한다.

4.1 실험 환경

제안한 기법으로 얻어지는 성능 향상을 정확하게 파악하기 위하여 FCFS 방식을 적용한 배칭 기법의 성능과 비교하였는데, 배칭 기법은 배칭 간격을 다양하게 하여 실험한 결과 중에서 가장 효율적인 성능을 나타낸 것으로 선택하였다.

실험은 시뮬레이션을 통해 평가하였다. 사용된 요구 작업부하는 실험의 편리함을 위하여 가상적으로 만들어 사용하였다. 가상 요구 작업부하는 주어진 수만큼의 요구를 생성할 수 있도록 되어있으며 작업부하는 요구하는 학습자 시스템의 번호와 원하는 교육 비디오 자료의 번호로 이루어져 있다. 요구하는 교육 비디오 번호는 인기 교육 비디오와 비인기 교육 비디오를 형성하기 위해 지수분포를 따르도록 하였다. 즉 인기있는 교육 비디오의 요구는 자주 발생하고 비인기 교육 비디오에 대한 요구는 드물게 발생한다. 본 실험에서는 20,000개의 요구로 이루어진 작업부하를 만들어 이용하였으며 요구 발생 시간은 다양하게 하며 실험하였다.

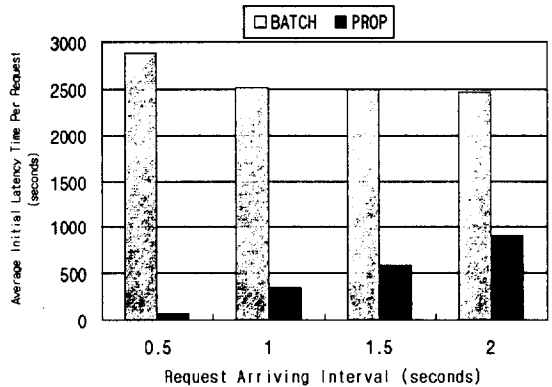
시뮬레이터는 C++를 사용하여 개발하였으며 작업부하를 입력하는 부분과 관리자과 저장 서버로 구성된다. 작업부하에서 요구들은 주기적으로 서버 시뮬레이터로 입력되며 시뮬레이터에서 관리자는 대기 큐를 두어 들어오는 요구들을 저장하며 큐에 들어온 순간부터 시간을 측정한다. 시뮬레이터의 동작은 알고리즘과 동일하게 작동하며 시뮬레이터에서는 각 요구에 대한 초기 지연시간과 서버의 처리량을 측정한다.

시뮬레이션 파라미터는 서버가 제공하는 교육 비디오 자료의 개수는 100개, 각 비디오의 길이는 100분, 서버 스트림 용량은 200개로 하였다. 그리고 한 라운드는 1초로 가정하였으며 한 라운드당 한 학습자에게 보내는 데이터의 양은 MPEG-1[12]을 기준으로 192KB로 하였다. 192KB 데이터가 네트워크를 통해 전송되는 시간은 ATM/155Mbps[1]를 고려하여 네트워크 오버헤드를 포함하여 9.8 milliseconds인데 10 milliseconds로 고정시키고 실험하였다. 각 정책에 대한 성능 비교 척도로는 요구당 초기 지연시간과 서버의 처리량을 사용하였으며 시뮬레이션에서 학습자 요구 발생 간격을 다양하게 하여 실험하였다. 단, 신청했던 교육 비디오를 보던 중에

이탈하거나 신청했다가 취소하는 경우는 없는 것으로 가정하였다. 또한 VCR 동작은 고려하지 않았으며 이 동작을 고려하려면 정책에 있어서 추가적인 보완이 있어야 한다.

4.2 실험 결과

제안한 기법과 배칭 기법의 성능을 비교하였다. 본 실험의 제안한 기법에서는 정확한 성능향상을 파악하기 위하여 전혀 배칭을 하지 않았다.

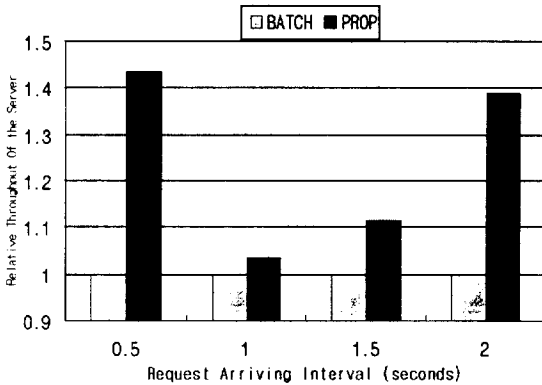


<그림 4> 정책에 따른 초기지연시간

<그림 4>는 스케줄링 정책에 따른 초기지연시간을 나타낸 것이다. 그림에서 BATCH는 배칭 기법을 나타낸 것이고 PROP는 제안한 정책을 나타낸 것으로 각 학습자 시스템의 버퍼 용량은 압축된 데이터 5분 동안의 분량을 저장할 수 있는 크기이다. 그리고 X축은 학습자로부터 발생하는 요구 발생 간격을 의미한다. 그림의 결과에서 몇 가지 사항을 관찰할 수 있다. 우선 모든 경우에 있어서 제안한 기법이 배칭 기법에 비해 매우 효율적인 초기 지연시간을 나타내는 것을 볼 수 있다. 이는 자주 요구되는 비디오에 대한 서비스를 다른 학습자 시스템으로 하여금 처리하게 함으로 전체 시스템 입장에서 더욱 많은 학습자들에 대한 서비스를 제공할 수 있기 때문이다. 다음으로 제안한 기법에서는 요구 발생 간격이 커짐에 따라 요구당 초기 지연 시간이 지연되는 것을 확인할 수 있는데 이는 요구 발생 간격이 커지면 커질수록 비디오의 시작 부분을 다른 학습자 시스템의 버퍼에 저장하고 있는 동안 발생하는 요구의

발생 횟수가 감소하기 때문이다.

<그림 5>는 정책에 따른 서버의 처리량을 상대적으로 나타낸 것이다. 정책간의 성능에 있어서는 제안한 기법이 배칭 기법에 비해 4-45% 가량 서버의 처리량이 증가하는 것을 볼 수 있다. 그러므로 본 논문에서 제안한 기법은 학습자 입장에서 초기지연시간이 짧아짐은 물론이고 서버의 처리량 측면에 있어서도 좋은 성능을 나타내는 효율적인 정책임을 알 수 있다.



<그림 5> 정책에 따른 서버의 상대적인 처리량

5. 결론

교육 환경의 발전으로 주문형 교육 시스템 개념이 시선을 끌고 있다. 이러한 주문형 교육 시스템에서 보다 많은 학습자들에게 서비스를 제공하기 위해 본 논문에서는 한번의 디스크 검색으로 더욱 많은 학습자들의 요구를 서비스하는 스케줄링 기법을 제안하였다. 주문형 교육 시스템에서는 일반적인 텍스트 데이터와는 달리 동일한 교육 비디오에 대한 요구가 많이 발생하므로 특정 비디오에 대한 요구가 발생했을 때 해당 비디오의 내용이 다른 학습자 시스템의 버퍼에 유지되고 있을 확률이 높다. 본 논문에서 제안한 정책은 이러한 점을 활용하였는데 요구가 발생했을 때 무조건 서버가 서비스를 제공하는 것이 아니라 해당 교육 비디오를 저장하고 있는 학습자 시스템이 있는지를 검색해서 있을 경우에 작업을 학습자 시스템에게 넘겨 작동하게 하였다. 제안한 정책의 효과를 검증하기 위하여 시뮬레이션을 통해 제안한 기법과 배칭 기법의 성능 비교를 다각적으로 시행하였다. 성능 비교를 통해 본 논문에서 제안한 정책

이 요구당 초기 지연시간과 서버의 처리량에 있어서 매우 효율적인 성능을 나타내는 것을 확인하였다.

제안한 정책은 학습자 시스템의 버퍼만을 활용하는데 앞으로 서버의 버퍼도 함께 고려하여 동작하는 정책으로 확장할 것이다.

참고문헌

- [1] D. E. McDysan and D. L. Spohn, ATM: Theory and Application, McGraw-Hill, 1995.
- [2] N. J. Boden et al., Myrinet: A Gigabit-Per-Second Local Area Network, IEEE Micro, 15(1):29-36, February 1995.
- [3] D. Gemmell, H. vin, D. Kandlur, P. Rangan, Multimedia Storage Servers: A Tutorial and Survey, IEEE Computer, 1995.
- [4] L. Golubchik, J. Lui, and R. Muntz, Reducing I/O Demand in Video-On-Demand Storage Servers, ACM Sigmetrics Conference, pages 25-36, May 1995.
- [5] Jong-Hoon Kim, Se-Woong Eom, Sam H. Noh, and Yoohun Won, Striping and Buffer Caching for Software RAID File Systems in Workstation Clusters, Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), pages 544-551, Austin, Texas, May 31-June 5, 1999.
- [6] C. Aggarwal, J. Wolf, and P. Yu, On Optimal Piggyback Merging Policies for Video-on-Demand Systems, Technical Report, IBM RC 20337, February 1996.
- [7] Mohan Kamath, Krithi Ramamritham, and Don 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.
- [8] Asit Dan and Dinkar Sitaram, A Generalized Interval Caching Policy for Mixed Interactive and Long Video Environments, In IS&T SPIE Multimedia Computing and Networking, San Jose, January 1996.
- [9] 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, pages 25-32, 1994.
- [10] H. Shachnai and P. Yu, The Role of Wait Tolerance in Effective Batching: A paradigm for Multimedia Scheduling Schemes, IBM Research Report, RC 20038, 1995.
- [11] S. Sheu, K. Hua, and W. Tavanapong, Chaining: A Generalized Batching Technique for Video-On-Demand Systems, In Proceedings of the IEEE International Conference on Multimedia Computing and Systems, pages 110-117, June 3-6, 1997.
- [12] O. Rose, Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems, University of Wurzburg Research Report Series No. 101, February 1995.

#### 김 종 훈

1990 목원대학교 수학교육과 (이학사)  
 1992 동국대학교 대학원 통계학과 (이학석사)  
 1998 홍익대학교 대학원 전자계산학과 (이학박사)  
 1998-1999 홍익대학교 부설 과학기술연구소 연구원  
 1998-1999 한국전자통신연구원 OS연구팀 Post-Doc.  
 1999-현재 제주교육대학교 컴퓨터교육과 전임강사  
 2000-현재 과학영재교육센터 초등정보반 지도교수  
 관심분야: 컴퓨터 교육, 리눅스, 웹 프로그래밍, 초등  
 정보영재교육

#### 한 금 희

1971 성심여자대학교 화학과 (이학사)  
 1981 미국 RPI 대학교 computer science (이학석사)  
 2000 홍익대학교 대학원 전자계산학과 (이학박사)  
 1982-현재 가톨릭대학교 컴퓨터공학부 교수  
 관심분야: 멀티미디어 시스템, 프로그래밍 언어론