論文 95-9-9-25

VOD상에서 프리젠테이션 방향성을 이용한 동적 캐쉬 스케쥴링 방법

正會員 李 承 潤*, 朴 昊 均*, 柳 煌 彬*

Dynamic Cache Scheduling Policy using Presentation Direction on VOD

Seung Yun Lee*, Ho Kyun Park*, Hwang Bin Ryou* Regular Members

要 約

VOD(Video-On-Demand) 서비스는 디지털 통신망을 통한 비디오 대여 서비스로써 대표적인 대화형 멀티미디어 서비스이다. VOD 시스템에 있어, 다중 사용자의 시청 요구는 시간적, 공간적으로 다양하게 분산되는 사용자 세션의 특성을 갖는다. 또한, 여러 시청의 요구들 사이에는 동일하거나 인접한 미디어 블럭을 요구하는 세션간의 국부성(Inter-session locality)이 존재하며, 이러한 국부성은 잠재적인 캐싱의 가능성을 제공한다.

본 논문에서는 각 사용자 요구에 대한 국부성에 기초하여 VOD 서비스 중에 발생될 수 있는 다양한 요구에 효과적으로 대처할 수 있는 동적 캐쉬 스케쥴링 기법을 제안한다. 이 기법은 사용자 요구의 진행 방향성을 고려하여 캐싱 여부를 결정하기 때문에 다중 사용자에 의한 다양한 요구(play, pause, rewind, fastforward)에 대해서도 최적의 캐싱 효과를 가질 수 있다. 따라서, 이것은 사용자와 서버간의 I/O와 이에 따르는 통신 대역폭을 최소화시킴으로써, 더 많은 사용자를 수용할 수 있을 뿐 만 아니라, 최적의 비용으로 VOD 서비스할 수 있는 가능성을 제공한다.

ABSTRACT

VOD(Video-On-Demand) is a video rental service over a digital network and is typical interactive multimedia services. In VOD systems, requests of user can be characterized by spatially and temporally distributed user sessions. There also exists an inter-session locality which requires the same or neighbouring media blocks between the user requests. This locality, therefore, provides a potentiallity for efficiency caching.

This paper proposes a dynamic cache scheduling policy, based on the locality of each user request, which effectively handles various requests for VOD service. The policy considers a presentation direction of user requests so that an optimal caching effect for various requests(play, pause, rewind, fastforward) of multiple users can be achieved. It minimizes the communication band-

*광운대학교 전자계산학과
Dept. of Computer Science Kwangwoon University
論文番號 : 95261-0807
接受日字 : 1995年 8月 7日

width between the server and the users, and hence can service more users. Consequently the suggested policy provides a possibility to optimize service cost in VOD service.

Ⅰ. 서 론

VOD 서비스는 기본적으로 통신망을 통한 비디오 대여 서비스라 할 수 있으며, 이를 이용하는 사용자는 자신이 원하는 시간에 원하는 형태로 자유로운 조작을 통한 서비스를 제공받을 수 있다. 즉, 사용자는 마치 자신의 VCR을 조작하듯이 서비스를 제공받을 수 있는 특징을 지니고 있다. 또한, 이러한 비디오 서비스는 영화에 대한 서비스뿐만 아니라 뉴스, 홈 쇼핑, 게임, 교육, 광고 등 향후의 미래 지향적인 서비스를 가능하게 할 것이다[4,13].

VOD 서비스를 실현하기 위해서는 미디어 압축 기술, 대용량의 미디어를 처리하는 서버 구현 기술, 고속 전송을 위한 통신 기술 등을 필요로 한다. 특히, MPEG과 같은 동화상 압축 기술[6,23]과 B-ISDN 구현의 대표적인 기술인 ATM 기술의 성숙, 그리고 RAID와 같은 디스크 배열을 이용한 대용량 저장 서버의 구현 기술[6] 등은 VOD 서비스에 대한 보다 많은 실현 가능성들을 제시해준다고 할 수 있다. 효과적인 VOD 서비스를 제공하기 위해서는 다중 사용자들의 다양한 요구를 수용해야 하기 때문에 보다 많은 통신 대역폭이 제공되어야 하며, 동시 다발적인 사용자 요구 집중에 대한 적절한 스케쥴링이 가능한 서버와 신뢰성 있는 통신망의 구성 및 운영이 필수적이다.

지금까지 VOD에 관해 많은 연구가 진행되고 있다. 멀티미디어 데이터를 저장하고 전송하기 위한 서버의 설계시 고려해야할 문제점들에 대해 연구된바 있으며, 특히, 다중 사용자에 의해 요구되는 엄청난 서버 I/O는 VOD 구현에 가장 큰 장애 요인이기 때문에 그 부하를 최소화할 수 있도록 서버를 설계하는 연구가 많이 이루어지고 있다[1,2,7,8,10,12]. 또한 통신망 측면에 있어서, 사용되는 통신 대역폭을 최소화하기 위해 멀티캐스트를 이용한 서비스 방법[5,9]과 예약을 기초로한 선 스케쥴링[11]이나 분산 서버 환경에서 인기도에 근거한 확률 모델을 이용한 영화 재 할당 방법[3]이 연구된 바 있다.

본 논문에서는 서버의 I/O와 통신망 대역폭을 최소화하기 위해 가입자 장치(CPE)에 인접한 자국 교환기(local Switch) 부분에 캐쉬 버퍼를 두고 여기서 발생되는 사용자 요구의 국부성(locality)을 이용하여 적절한 버퍼 스케쥴링을 함으로써 서버 I/O와 이에 따르는 통신 대역폭을 줄이고자 하는데 그 목적이 있다.

Ⅱ. VOD 시스템

2.1 VOD 시스템 특성 및 요구사항

2.1.1 시스템 특성

VOD 서비스는 기존의 단방향 방송 형태(broadcast)와 달리 사용자의 요구에 의한 개별적인 서비스를 하는 특징을 지니고 있다. 이러한 대화형 서비스를 제공하기 위해서는 많은 기술적인 요구 사항들을 해결해야 한다. 특히, 사용자들의 동시적이고 개별적인 요구를 수용할 수 있는 VOD 서버의 설계는 필수적이라 할 수 있다. 또한, 이를 뒷받침해줄 수 있는 고속의 통신망과 서비스를 효율적으로 수행할 수 있는 서버에 대한 적절한 분산 시스템 등이 필요하다. VOD 시스템을 구축할 수 있는 기본 구조로는 VOD 서버, 통신망, 가입자 장치(CPE)로 크게 분류할 수 있다.

2.1.2 시스템 요구 사항

VOD 서비스를 제공하기 위해서는 대용량의 서버로부터 고속의 통신망에 이르기까지 다양한 요구 사항들이 충족되어야 한다.

VOD 서버

VOD 서버는 비디오 소스 프로그램을 제공하는 통신망 요소이며, 그 프로그램들은 일반적으로 디스크-배열 장치와 같은 대용량의 고속 액세스가 가능한 저장매체에 저장되어 있어야 한다. VOD 서버는 다중 사용자의 요구를 수용할 수 있도록 병렬 서버 등을 통해 고속의 스위칭 및 데이터베이스 I/O를 수행할 수 있도록 설계되어야 하며, 사용자 요구에 대해 실시간 처리를 할 수 있어야 한다. 또한, 제한된 서버 용량에서 보다 많은 정보

를 서비스하기 위해서는 각종 미디어 정보에 대한 압축 저장이 가능해야 한다[2,5].

통신망

일반적으로 VOD 데이터는 지속적으로 많은 양의 정보가 고속으로 전송되는 폭발적, 동시성의 특징을 갖기 때문에 통신망의 특성도 이를 수용할 수 있어야 한다. 현재, VOD 실현 기술로 ADSL이 개발되었으며, 이것은 기존의 전화망에서 운영되고, 기존의 음성대역폭을 훨씬 넘는 약 1.5Mbps의 전송 속도를 지닌다. 따라서 제한적이나마 VOD 서비스를 가능하게 하며, ADSL-II의 경우 약 6Mbps이상의 고속 전송이 가능하게 됨에 따라 보다 다양한 형태의 VOD 서비스를 제공할 수 있다. 이것은 새로운 망의 구축을 요구하지 않고 서비스할 수 있다는 측면에서 그 의미를 갖는다고 할 수 있다. 하지만, 기존의 망은 B-ISDN으로 진화하게 될 것이고 VOD 서비스도 이러한 광대역 통신망에서 운영될 수 있어야 할 것이다. 따라서, VOD와 같이 고속의 비디오 전송을 요하는 서비스를 제공하기 위해서는 통신망에 대해서 높은 연결성을 요구하는데, ATM, FDDI, DQDB, 100Mbps Ethernet 등이 VOD 서비스를 실현할 수 있는 대표적인 망 전송 기술이다.

가입자 장치

가입자 장치(CPE)는 VOD 서버로부터 전송되어온 MPEG 정보를 다시 복호화하여 사용자 디스플레이 장치에 전송하는 기능과 사용자의 요구를 서버에게 알려주는 기능을 수행하며, 이는 각각 다른 채널을 통해 처리된다. 또한, 가입자 장치는 적절한 버퍼를 둠으로써 통신망 지터 등으로 인한 지연으로부터 디스플레이의 연속성을 보장시키는 역할을 한다.

Ⅲ. 제안하는 VOD 시스템 구조

3.1 기본 구조

제안하는 VOD 시스템은 그림 1과 같이 ATM을 전송 방식으로 하는 B-ISDN과 같은 고속의 통신망 상에서 운영되며, 구성·요소로써 대용량의 비디오 저장 장치를 갖는 고속의 서버와 다수의 지역 서버 및 가입자 장치(CPE)로 구성된다. 지역 서버에는 자신이 담당하는 가입자들에 대한 서비스를 위한 스위치와 비교적 적은 용량의 비디오 서버, 그리고 본 논문에서 제안하는 캐쉬 스케쥴러가 존재한다. 본 논문에서 제안하는 통신망의 구조는 현재의 서비스 통신망은 앞으로 광대역 통신망으로 변화될 것이며, 이때 제안하는 캐쉬 스케쥴링에 의한 이득을 보다 극대화시킬 수 있다.

3.2 캐쉬 스케쥴러

본 논문에서는 VOD 서비스시 통신망 비용을 절감시킴과 동시에 효율적인 서비스를 제공하기 위한 방법으로써, 자국 교환기에 캐쉬 버퍼를 두고 모든 가입자들에 대한 요구사항들을 파악하여 적절한 캐쉬을 수행하는 스케쥴링 방법을 제안한다.

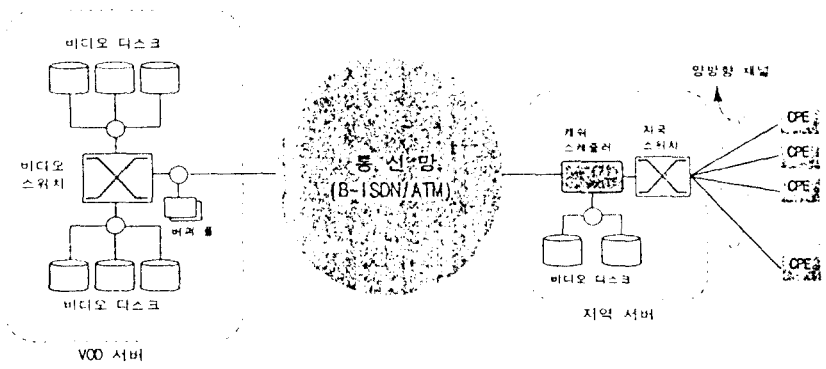

그림 1. 제안하는 VOD 시스템 구조
Fig. 1. Proposed VOD system architecture

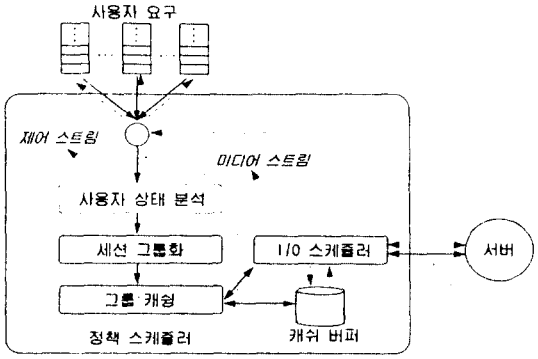

그림 2. 캐쉬 스케쥴러 모델
Fig. 2. Cache Scheduler Model

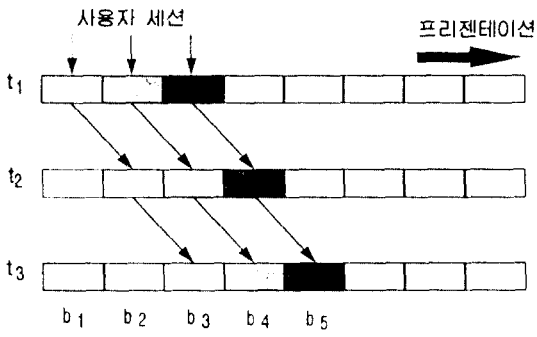

그림 3. 선형 VOD 세션의 예
Fig. 3. Example of VOD Session with Linear Presentation

3.2.1 기본 모델

사용자로부터의 비디오 정보 검색 요구는 그림 2와 같이 지역 서버내의 캐쉬 스케쥴러에 입력되고, 시스템은 모든 사용자 요구에 대해 자신의 스케쥴링 주기 내에서 접수된 요구를 분석하고, 여기서 얻어진 사용자들의 시청 방향에 따라 캐쉬 블럭을 그룹단위로 결정한다.

사용자의 프리젠테이션 상태를 분석하여 캐싱을 결정하는 캐싱 정책 스케쥴러와 여기서 결정된 스케쥴링에 의한 실제의 입출력을 담당하는 I/O 스케쥴러, 그리고 캐쉬 버퍼로 구성된다. 정책 스케쥴러는 사용자의 프리젠테이션 상태를 분석하고 각 사용자의 요구 블럭의 국부성에 따라 사용자 세션들을 그룹화시킨다. 여기서 캐싱을 그룹 단위로 함으로써 캐싱에서 얻어지는 이득을 더욱 높일 수 있으며, 이것은 얻어진 그룹들 사이에서 이득 순으로 캐싱함으로써 가능해진다.

사용자 요구 상태의 분석은 모든 사용자의 요구 블럭에 대해 그 국부성 여부를 기준으로 캐싱 그룹의 대상으로 만들고, 각 사용자들의 프리젠테이션 진행 상태(세션)의 동일성에 따라 실질적인 캐싱 그룹을 결정한다. 사용자의 세션 상태는 T-VOD를 기준으로 했을 때, 기존의 VCR에서 제공하는 기능인 play, pause, fast-forward, rewind의 모든 기능을 제공해야하기 때문에 그 변화의 정도는 매우 다양하다고 할 수 있으며, 이렇게 변화하는 세션의 상태에 따라 스케쥴러는 동적으로 캐싱할 수 있는 방법을 취한다. 즉, 세션의 상태에 따라 스케쥴러는 변화되는 캐싱 그룹을 생성하고 생성된 캐싱 그룹에 대해 그 이득의 우선 순위에 따라 캐싱 여부를 결정한다. 또한, 새롭게 변화된 캐싱 그룹에 대한 기존의 캐쉬 버퍼와의 페이지 교체는 그 그룹의 길이와 이득에 관계하여 새로운 페이지가 생성되도록 한다. I/O 스케쥴러는 실질적인 디스크 I/O 또는 캐쉬 버퍼로부터의 I/O를 담당하며, 정책 스케쥴러에 의해 결정된 그룹 단위의 블럭들에 대해 캐쉬 버퍼와의 페이지 교체 및 VOD 서버로부터의 I/O를 처리한다.

Ⅳ. 동적 캐쉬 스케쥴링 방법

4.1 기본 개념

일반적으로 사용자의 요구는 대부분 영화를 선형적으로 시청하게 되고, 그러한 사용자가 서로 인접하거나 중복되었을 경우 적절한 버퍼링을 함으로써 서버로부터의 I/O를 줄일 수 있다. 하지만, 사용자는 항상 선형적인 시청만 하지 않는다는 점을 감안하여, 이에 대한 적절한 고려를 필요로 한다. 따라서, 서비스 중에 발생될 수 있는 비 선형 시청에 대한 고려를 위해 본 논문에서는 각 사용자의 시청 상태를 파악하여 그 방향성을 가지고 동적인 캐싱 스케쥴링 방법을 제안한다.

4.1.1 선형 프리젠테이션에서의 캐싱

선형 프리젠테이션은 모든 서비스 사용자가 시청을 시작해서 끝날 때까지, 단순히 play 동작밖에 할 수 없는 경우를 말한다. 예를 들어 그림 3과 같이, 시간 t_1에서

사용자 u_1, u_2, u_3는 동일한 프리젠테이션에서 인접한 연속적인 블럭 b_1, b_2, b_3를 요구할 때, 그 사용자들이 모두 play 상태라면 시간 t_2에서는 블럭 b_2, b_3, b_4를 요구하게 될 것이다. 따라서, 모든 사용자의 재생율(γ)이 동일하다고 했을 때, 시간 t_1에서 블럭 b_2, b_3을 캐싱한다면 시간 t_2에서는 블럭 b_4만을 서버로부터 가져오면 되기 때문에 전체적으로 (3-1)=2γ만큼의 대역폭을 절감시킬 수 있다. 그러나, 실제의 서비스에 있어서는 사용자가 언제 다른 요구를 할지 모르기 때문에 그러한 비 선형적인 프리젠테이션에 대한 적절한 캐쉬 스케쥴링 기법이 요구된다.

4.1.2 비선형 프리젠테이션에서의 캐싱

일반적으로, 사용자의 요구에 변화가 발생되면 그것은 서비스 스케쥴러에게 전달되고, 스케쥴러는 다음 서비스를 수행하기 전에 각 사용자들에 대한 요구 변화를 미리 알 수 있게된다. 그러므로, 스케쥴러는 현재 캐쉬 메모리의 상태를 갱신하기 위한 사용자 요구를 예측 가능하게 된다. 스케쥴러는 그러한 정보에 의해 미리 예측된 정보를 이용하여 다음 시간에서의 캐쉬 메모리 이용 효율(대역폭 이득)을 극대화시킴으로써 비선형 프리젠테이션에서의 효과적인 캐싱을 지원할 수 있게 된다. 예로써, 그림 4와 같은 사용자의 프리젠테이션 요구가 있을 때, 시간 t_1에서 사용자 u_1과 u_2는 play 상태이고, u_3는 fastforward 상태이고 사용자 u_4와 u_5는 pause 상태임을 나타낸다. 그리고 각 사용자의 요구는 시간이 지남에 따라 각기 다르게 변화하는 것을 볼 수 있다. 이 때, 시간 t_1에서 t_7까지에 걸친 캐싱 내용에 따른 각 시간에서의 대역폭 이득을 표 2에서 보여주고 있다.

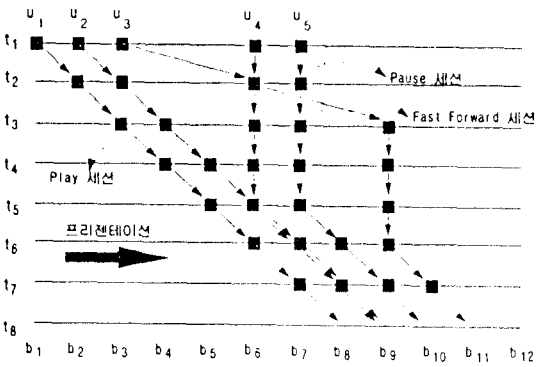

그림 4. 비선형 VOD 세션의 예
Fig. 4. Example of VOD Session with Non-linear Presentation

각 시간 t에서 모든 사용자의 재생율을 γ라 하고, 사용자 요구 수는 $N^{request}(t)$, 대역폭 이득은 $R^{gain}(t)$, 전체 스케쥴링 주기를 T라 하면, 전체 요구 수와 대역폭 이득은 다음과 같다.

$$N_{total}^{request} = \sum_{t=1}^{T} N^{request}(t) \tag{1}$$

$$R_{total}^{gain} = \left\{ \sum_{t=1}^{T} R^{gain}(t) \right\} * \gamma \tag{2}$$

이 때의 효율 E는 식(1)과 식(2)에 의해 다음과 같이 구할 수 있다.

$$E = \sum_{t=1}^{T} e(t) = \frac{R^{gain}}{N^{request}} \times 100(\%) \tag{3}$$

따라서, 표 1에서의 캐싱 효율을 시간 t_1에서 t_7까지 모두 계산하면 다음과 같은 결과를 얻는다.

$$avg(\frac{4}{5}, \frac{5}{5}, \frac{3}{5}, \frac{4}{5}, \frac{5}{5}, \frac{4}{5}, \frac{4}{5}) \approx 83\%$$

4.2 스케쥴링 방법

4.2.1 모델링

가정 1 : 사용자가 시청하는 프리젠테이션의 요구블럭을 b라 할 때, 이것이 곧 캐싱되어지는 메모리의 단

표 1. 캐싱에 의한 대역폭 이득 예
Table 1. Example of Bandwidth Gain by Caching

시간	사용자 요구 블럭	유효 캐쉬 내용	서버 I/O	대역폭 이득
t_1	b_1, b_2, b_3, b_6, b_7	b_1, b_2, b_6, b_7	b_3	4
t_2	$b_2, b_3, b_6 \times 2, b_7$	b_2, b_3, b_6, b_7		5
t_3	b_3, b_4, b_6, b_7, b_9	b_3, b_6, b_7	b_4, b_9	3
t_4	b_4, b_5, b_6, b_7, b_9	b_4, b_6, b_7, b_9	b_5	4
t_5	$b_5, b_6 \times 2, b_7, b_9$	b_5, b_6, b_7, b_9		5
t_6	$b_6, b_7 \times 2, b_8, b_9$	b_6, b_7, b_9	b_8	4
t_7	$b_7, b_8 \times 2, b_9, b_{10}$	b_7, b_8, b_9	b_{10}	4

위 블럭이 된다. 각 프리젠테이션은 γ의 속도로 재생되고, 각 사용자의 요구에 따라 각기 다른 세션을 가지며, 한 프리젠테이션은 b/γ의 시간 단위로 세그먼트되어진다고 할 때, 각 세션의 시작시간은 해당되는 시간 세그먼트(time segment)와 동기화(synchronize) 된다■

가정 1에 의해 각 사용자 세션은 블럭으로 정의할 수 있으며, 한 프리젠테이션에 대한 세션은 블럭 $b_1,\ldots,b_m$으로 분할할 수 있고, 각 세션 i에 대해 블럭 b_i는 공통적으로 같은 시작 시간을 갖는 세션의 집합이라 할 수 있다. 즉, 같은 블럭을 요구하는 사용자들은 모두 같은 시간에 재생된다. 따라서, 각 블럭에 대한 사용자 세션의 정보로써, 사용자 요구 블럭 u_m^{block}과 이에 대한 상태 u_m^{state}을 정의한다. 그리고 인접한 요구 블럭들에 대해 사용자 요구 상태에 따라 캐싱을 위한 그룹 G를 결정한다. 그리고 캐싱 그룹 G에 대한 이득을 구함으로써 실질적인 캐싱 여부를 결정할 수 있게 된다. 그룹이 결정되었을 때, 그 길이가 L^{group}이고, 이 그룹에 대한 요구 수 $n^{request}$라고 하면, 이 때의 이득 r^{gain}은 다음과 같다.

$$r^{gain}=\left\{\sum_{i=1}^{L^{group}} n_i^{request}-1\right\}*\gamma \qquad (4)$$

따라서, 주어진 캐쉬 메모리의 크기 M에 대해 그룹 단위의 대역폭 이득 순으로 캐싱 기법을 적용하면 주어진 캐쉬 메모리의 효율을 높일 수 있다.

가정 2 : 단위 시간에서의 스케쥴링 시간 T는 모든 사용자에 대한 요구 블럭 검색 및 전송 시간을 보장한다. 사용자의 요구 블럭에 대한 재생 시간 내에 스케쥴링 한다■

가정 2는 수락 제어를 위한 것으로, 모든 사용자에 대해 각 스케쥴링 시간에 있어, 사용자 요구 블럭에 대한 서비스 시간은 각 사용자들의 미디어 재생 시간보다 작거나 같아야 한다. 따라서, 한 서비스 라운드에서 서비스 시간은 사용자 i가 요구하는 블럭을 읽는 시간 S_i^{block}와 블럭간의 이동에서 비롯되는 스위칭 오버헤드 ρ를 더한 시간이 되며, 이 시간은 각 사용자들의 재생 시간 중 가장 빠른 시간보다 작아야 지연을 갖지 않는 서비스를 제공할 수 있다[12].

$$\sum_{i=1}^{m}(S^{block_i}+\rho^{\max})\le \min_{i\in[1,m]}\gamma^i \qquad (5)$$

이 때, 식(5)로부터 요구되는 버퍼 메모리 M의 크기는 다음 조건을 만족해야 한다.

$$\sum_{i=1}^{m}\{(S_i^{block}+\rho^{\max})*t^{segment}\}\le M \qquad (6)$$

여기서 $t^{segment}$는 각 시간 슬롯의 길이이며, 사용자 요구 블럭을 검색하는 시간과 단위 시간 슬롯을 곱함으로써 메모리의 단위로 변환할 수 있다.

4.2.2 알고리즘 및 수행 절차

본 논문에서 제시하는 알고리즘은 기본적으로 선형 프리젠테이션에 적용되는 스케쥴링 기법을 확장하여 사용자 요구가 변화하더라도 이에 효과적으로 대처할 수 있도록 하며, 이는 각 시간 단위로 스케쥴링을 할 때, 변화된 요구를 인식하여 그 중에서 캐싱 효율이 가장 높은 성분(사용자 요구)들을 추출한 뒤, 이들을 그룹화하여 잠재적인 캐싱 그룹들을 만들고 효율이 높은 캐싱 그룹부터 캐싱하기 때문에 캐쉬 메모리의 효율을 항상 높게 유지시킬 수 있다.

이를 위해 본 논문에서 제안하는 방법은 서비스가 진행되는 동안 스케쥴러가 모든 사용자들의 요구를 분석하여 서로 인접한 요구 블럭들을 찾아내고 이들의 방향성을 조사하여, 방향성에 따라 캐싱 여부를 결정한다. 예를 들어, 인접한 블럭을 요구할지라도 그 사용자들의 프리젠테이션 방향이 play나 pause가 아니라면, 이들은 캐싱 그룹에서 제외시킨다.

이렇게 해서 얻어진 캐싱 그룹들에 대해 각각의 길이와 대역폭 이득을 계산하고, 추출된 캐싱 대상 그룹들 중 가장 효율이 높은 순으로 캐싱함으로써 제한된 캐쉬 버퍼 내에서 최적화된 스케쥴링을 가능하게 할 수 있는 것이다. 이러한 캐쉬 스케쥴링의 세부적인 과정은 다음과 같다.

① 캐쉬 메모리를 초기화 한 후, 각 단위 시간에서 최적 캐싱을 위한 스케쥴링을 수행한다.

② 각 시간에서 각 사용자의 시청 요구 상태와 요구 블럭을 파악한다.

③ 각 시간 t에서 요구되는 블럭에 대한 그룹화를 실시한다. 따라서, 캐싱은 블럭들의 집합인 그룹 단위로 이루어지며, 각 블럭들 간에는 어떤 규칙을 따라야만 하나의 그룹으로 특성 지워질 수 있다.

□ 그룹화 조건

1. 요구되는 블럭은 서로 인접해야 한다.

2. 인접한 블럭은 서로 동일한 방향성을 가져야 하며, play나 pause의 경우에만 그룹의 대상으로 적용된다.

④ 구해진 각 블럭 그룹에 대해 속성을 계산한다. 즉, 이미 구해진 잠재적인 캐쉬 그룹들에 대해 실질적인 캐쉬 여부를 결정하기 위해 각 그룹의 길이, 블럭의 요구 상태, 사용자 요구 수, 대역폭 이득을 구한다.

(a) 그룹의 길이($L_k^{group}(t)$)는 그룹 내에서 요구하는 첫 블럭의 위치와 마지막 블럭의 위치에 의해 결정될 수 있다.

$$L_k^{group}(t) = u_{m+l}^{block}(t) + u_m^{block}(t) + 1 \tag{7}$$

(b) 그룹 내의 각 블럭에 대해 사용자 요구 상태를 알기 위해 블럭별 요구 상태 집합을 구한다.

$$G_{k,i}^{state}(t) = \sum_{i=1}^{l} u_m^{state}(t), \text{ if } u_i^{block}(t) = b_i \tag{8}$$

(c) 그룹내에서 사용자의 요구수($n_k^{request}(t)$)는 앞서 구한 길이 만큼에 대한 각 블럭의 요구수를 누적시킴으로써 얻어진다.

$$n_k^{request}(t) = \sum_{i=1}^{l} n_k^{request}(t), \quad l = L_k^{group}(t) \tag{9}$$

(d) 그룹내에서의 대역폭 이득은 그룹내에 속한 각 블럭들을 요구하는 사용자 수에 의해 얻어지며, 정확한 이득은 사용자 상태 집합을 고려했을 때 얻어질 수 있다.

$$r_k^{gain}(t) = \begin{cases} (\sum_{i=1}^{l} n_k^{request}(t) - 1) * \gamma, \\ \quad if \quad G^{state_{i+1}}(t) = \{play\} \\ (\sum_{i=1}^{l} n_k^{request}(t) - 1) * \gamma, \\ \quad if \quad G^{state_{i+1}}(t) = \{pause\} \end{cases} \tag{10}$$

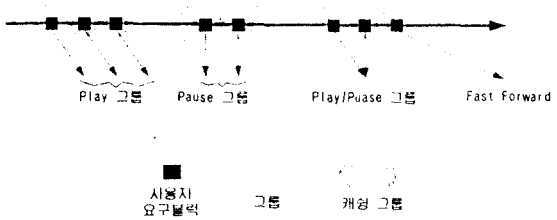

그림 5. 사용자 세션에 대한 그룹화 및 캐쉬 예
Fig. 5. Example of Grouping and Caching for User Sessions

결정된 그룹내에서 실제로 캐쉬의 대상이 되는 블럭은 그 그룹의 처음 위치에 있는 블럭을 요구하는 사용자의 상태에 따라 다르다. 즉, 그룹 시작에 있는 사용자가 Play 상태이면 그 블럭은 캐쉬에서 제외되지만, Pause상태이면 캐쉬 블럭으로 포함된다.

⑤ 계산된 각 그룹들 중, 대역폭 이득이 가장 큰 그룹부터 캐쉬한다.

여기서 얻어지는 캐쉬 그룹의 결정 조건은 다음과 같다.

$$f^{del}(t) = \sum_{i=1}^{m} L_{a_i}^{group}(t), \quad f^{del}(t) \le M^{cache} \tag{11}$$

즉, 결정된 캐쉬 그룹들은 현재 캐쉬 버퍼에 적재되어 있는 그룹들과 비교하여 그 대역폭 이득의 우선 순위에 따라 재구성될 수 있다.

$$M_{remain}^{cache} = M_{remain}^{cache} - f^{del}(t) \tag{12}$$

캐쉬 버퍼로의 저장은 모두 그룹 단위로 하며, 실제 데이터는 물론이고 각 그룹들의 길이, 대역폭 이득이 저장되게 된다. 또한 캐쉬 메모리의 사용은 동적인 할당과 제거를 위해 그룹 할당 테이블(Group Allocation Table)을 둔다.

한편, 캐쉬 버퍼가 이미 모두 사용되고 있을지라도 사용자 요구의 변화에 의해 새롭게 생성된 잠재적 캐쉬 그룹의 대역폭 이득이 이미 캐쉬 내에 있는 어떤 그룹의 대역폭 이득 보다 클 경우가 있기 때문에 캐쉬되어져야 할 필요가 있다.

□ 그룹 재 생성 조건 :

1. 그룹의 길이(L^{group})가 같거나 작을 경우
2. 대역폭 이득(r^{gain})이 더 큰 경우

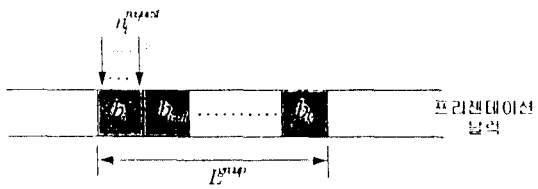

그림 6. 캐쉬 그룹의 이득
Fig. 6. The Gain of Caching Group

이와 같은 절차에 의해 각 단위 시간에서 사용자 요구의 특성을 기초로 이득 우선 순위의 캐쉉 그룹을 결정함으로써 보다 높은 서비스 효율을 얻을 수 있게 된다.

Ⅴ. 시뮬레이션 및 성능 평가

5.1 시뮬레이션 방법

본 논문에서는 제안하는 캐쉉 기법의 성능을 평가하기 위해 시뮬레이터를 구성하였으며, 캐쉉 효율을 평가하기 위한 모듈과 시뮬레이션을 위한 데이터 생성 모듈로 나뉜다. 시뮬레이션을 위해 사용되는 정보의 자료 구조는 사용자의 시청 상태를 구분하는 부분과 캐쉬 정보를 표현하는 부분 그리고 접수된 요구에 대한 캐쉉 그룹들에 관한 정보 및 이득에 관한 내용을 정의하고 있다. 또한, 각 스케쥴링 단위 시간에서의 사용자 세션 정보를 위한 자료를 정의함으로써 수락 제어를 위한 정보를 처리하도록 하였다. 기본적으로, 사용자의 시청 상태는 기존의 VCR에서 제공하는 기능을 기준으로 설정한다.

캐쉬 버퍼에 대한 정보로써 버퍼 내용(블럭 인덱스), 페이지 교체 대상 플래그 등을 정의하고, 실질적인 캐쉉의 단위인 그룹들에 대한 정보로써 그룹의 상태, 시작/끝 블럭, 길이, 이득 등에 대한 정보를 정의하며, 그 내용은 표2와 같다.

5.2 시뮬레이션 환경

본 논문에서는 제안하는 동적 캐쉬 스케쥴링 기법을 시뮬레이션하기 위해, 표3과 같은 시뮬레이션 환경을 설정하였다. 약 60분 정도의 양을 갖는 단일 프리젠테이션(약 5~600MB)에 대해, 300~500명의 사용자가 시간이 변함에 따라 임의로 세션요구를 하는 상황을 가정하였다. 또한, 사용자의 요구를 play, pause, fast-forward, rewind의 4가지 상태를 기준으로 전체 시간에 대한 요구 변화 빈도(p)를 변경시키며 시뮬레이션을 실시한다.

$$p = \frac{N^{request}_{u^{new} \in \{pause,\ rewind,\ forward\}}}{N^{request}} \tag{13}$$

사용자의 최초 요구 도착 간격은 포아송 분포(Poisson Distribution)를 갖는 것으로 하였으며, 프리젠테이션에 대한 단위 블럭(b_i)은 약 1MB로 나누었

표 2. 캐쉬 및 사용자 세션 정보 자료 구조
Table 2. Data structure of Cache and User Session Information

```
typedef enum { PLAY, PAUSE, RW, FF, STOP } UState;

typedef struct {
    struct {
      int Buff;              /* Cache Buffer                    */
      int useFlag;           /* Buffer to be used check byte    */
      unsigned int time;     /* Time                            */
    } cbuff[T_MAX_BUFF];
     unsigned int remain;
} CACHE;

typedef struct {
    struct {
      UState GS_state;       /* State of Grpup                  */
      unsigned int start;    /* Start block index of Group      */
      unsigned int end;      /* End block index of Group        */
      unsigned int length;   /* Length of Group                 */
      unsigned int gain;     /* Bandwidth Gain of Group         */
    } pos[MAX_USER];
      unsigned max_grp;      /* Number of Maximum Group         */
} GROUPS;
```

표 3. 시뮬레이션 환경 및 매개 변수
Table 3. Simulation Environment and Parameters

구 분	매개변수	내 용
사용자	수	300~500명
	도착간격	포아송 분포(0.1~1.0 user/second)
	요구변화	p(요구 변화 빈도)
프리젠테이션	수	단일 프리젠테이션
	길이	60분(약 5~600Mbyte)
스케쥴링	버퍼크기	100~200 Mbyte
	단위블럭	약 1Mbyte

다. 또한, 제안하는 스케쥴링 기법의 성능을 평가하기 위해 LRU 캐싱 기법과 비교하여 시뮬레이션을 실시한다.

5.3 성능 평가

제안하는 캐싱 기법에 대한 성능을 평가하기 위해 시뮬레이션 대상은 캐쉬 스케쥴링 적용시의 서버로부터의 I/O가 어떻게 감소하는가를 중심으로 알아본다.

먼저, 그림 7은 사용자가 단지 play만 할 수 있는 선형 프리젠테이션에 대한 성능을 평가한 결과이다. 여기서, 제안하는 캐쉬 스케쥴링에 의한 방법은 기존의 방법보다 캐쉬 버퍼의 크기가 커짐에 따라 더 높은 성능을 나타내고 있는데, 그 이유는 제안하는 스케쥴링 방법에서는 다음 시간에 다시 요구되어질 블럭만을 우선적으로 캐싱하기 때문이다. 그리고 캐쉬 메모리의 크기가 커짐에 따라 두 방법의 효율 차가 커지는 이유는, 제안하는 방법의 경우 요구 집중이 많아지는 블럭들이 집중하는 경우, 이에 대한 그룹화시 그 이득율이 상대적으로 높아지고 이러한 그룹들에 대한 캐싱의 우선 순위를 높게 결정하고 있기 때문이다.

그림 8은 사용자의 시청상태가 다양하게 변화할 수 있는 비 선형 프리젠테이션에 대한 성능을 평가한 결과이다. 이것은 캐쉬 메모리 크기가 증가함에 따라 동일한 프리젠테이션에 대해 사용자 요구 변화 빈도 p를 세가지 경우(p=0.2,0.3,0.5)로 달리하여 시뮬레이션 한 결과이다. 여기서 볼 수 있는 특징으로써, 기존의 캐싱 방법에서 볼 수 있는 결과는 사용자 요구 빈도가 높아질수록 캐싱 효율이 조금씩 나아지는 것을 볼 수 있는데, 이것은 그 만큼 사용자 요구의 변화가 많아지고 결국 여기서 비롯되는 요구의 중복 확률이 높아지기 때문이다. 하지만 제안하는 캐싱 스케쥴링 방법에 의한 결과는 사용자 요구 빈도에 그렇게 민감하지 않고 거의 유사한 결과를 보이고 있다. 즉, 이것은 캐싱 효율이 사용자 요구 빈도 변화에 상대적으로 의존적이지 않다는 결과를 보이고 있는데, 제안하는 방법에서는 스케쥴링시 항상 캐쉬 메모리를 최적화 되도록 하는 캐싱 전략을 사용하기 때문이다.

그림 9는 사용자 요구 변화 빈도 값 p를 변화시켜가면서 캐싱 효율을 시뮬레이션한 결과이다. 이 결과에서도 기존의 캐싱 방법은 요구 변화 빈도가 높아질 수록 그 효율이 점차 좋아지는 반면, 제안하는 캐쉬 스케쥴링 방법은 약간의 변화가 있긴 하지만 평균적으로 사용자 요구변화의 빈도에 독립적이라는 결과를 다시 한번 확인할 수 있다.

그림 10은 고정된 캐쉬 메모리 상에서 사용자 수를 변화시켰을 때의 효율을 시뮬레이션한 것이다. 여기서 볼 수 있는 결과는 제한된 캐쉬 버퍼에서 그 버퍼 메모리의 효율을 극대화시킬 수 있을 때, 가장 좋은 결과를 나타내고 있는데 이것은 사용자 수가 증가함에 따라 필요로 되는 버퍼의 크기도 증가하기 때문에 증가 인원에 대한 서버 I/O 효율도 그만큼 감소하게 되는 것이다. 결과적으로 서버가 요구하는 캐쉬 메모리의 크기는 수용할 수 있는 사용자와 관계하여 결정할 문제이며, 요구되는 캐쉬 메모리가 커질 경우 계층적인 캐쉬 메모리 구조를 두는 것도 바람직할 것이다.

그림 11은 사용자의 요구 시작에 대한 도착율을 변화시켰을 때의 요구 세션 스트림의 수를 측정한 시뮬레이션 결과이다. 여기서 세션 스트림은 캐쉬 버퍼에서의

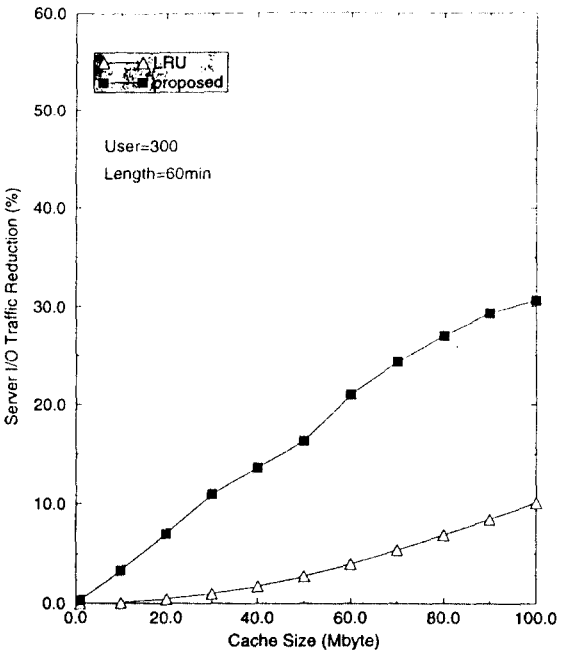

그림 7. 선형 프리젠테이션에 대한 캐쉬 효율
Fig. 7. Cache Efficiency of Linear Presentation

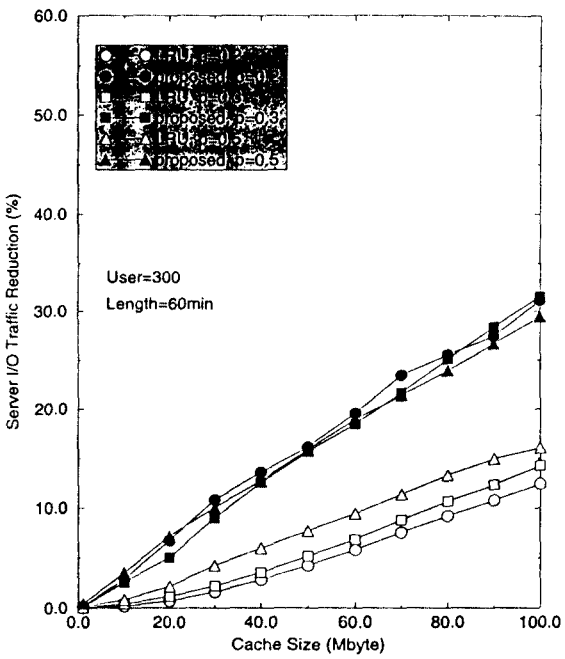

그림 8. 비선형 프리젠테이션에 대한 캐쉬 효율 (p=0.2~0.5)
Fig. 8. Cache Efficiency for Non-linear Presentation (p=0.2~0.5)

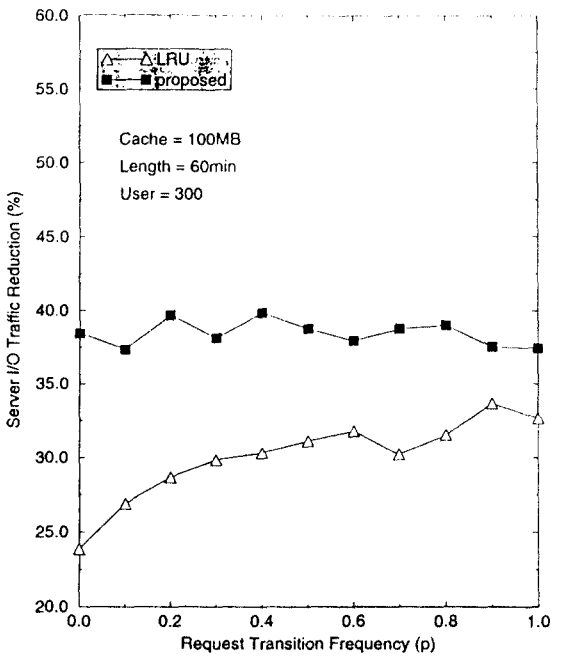

그림 9. 사용자 요구 변화 빈도에 따른 캐쉬 효율
Fig. 9. Cache Efficiency for various User Request Frequencies

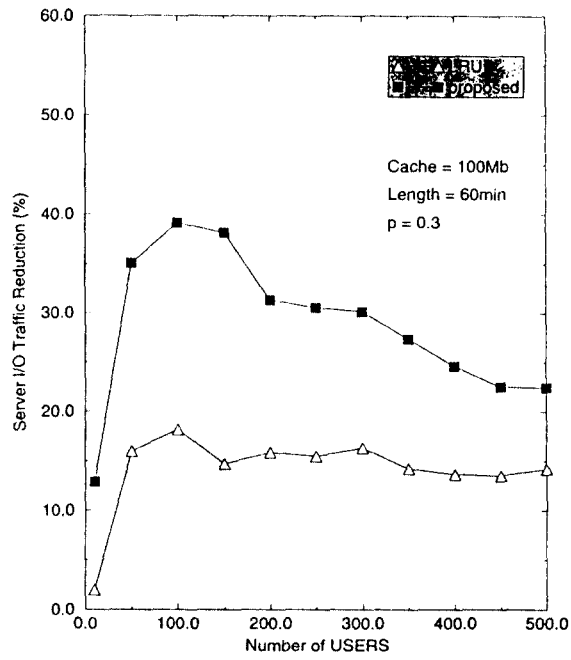

그림 10. 비선형 프리젠테이션에 대한 사용자 수 별 캐쉬 효율
Fig. 10. Cache Efficiency for various Numbers of Users with Non-linear Presentation

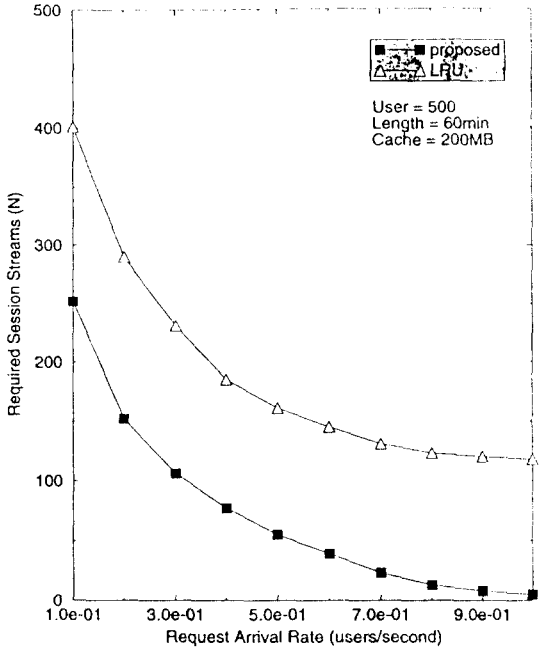

그림 11. 사용자 요구 도착율에 따른 요구 세션 스트림 수
Fig. 11. Required Number of Session Streams for various User Arrival Rates

I/O가 아닌 서버로부터의 I/O를 의미하는 것으로, 앞에서의 시뮬레이션에서 사용한 서버 I/O 감소율이 아닌 실질적인 세션의 수에 대한 측정을 실시함으로써 VOD 서버에 대해 사용자당 요구되는 세션의 수를 보다 구체적으로 말해주고 있는 결과이다. 특히, 사용자 도착 간격이 1에 가까워지면 제안하는 방법의 경우 실제 요구되는 세션 스트림이 거의 필요 없게 되는 것을 볼 수 있는데, 이것은 모든 요구의 작업 집합(working set)이 모두 캐쉬 버퍼내에 존재하기 때문이다. 사용자의 요구 도착의 간격이 작게되면 요구되는 블럭들의 인접할 확률이 높아지게 되고 이것은 곧 이득이 높은 캐쉬 그룹으로 묶이게 되어 그만큼 효율이 높아지게 되기 때문에 사용자 요구의 도착 간격이 짧아 질 수록 제안하는 스케쥴링 방법은 적은 세션의 수만 가지고도 서비스를 할 수 있도록 해주는 것이다.

Ⅵ. 결 론

일반적으로 VOD 서비스는 다중 사용자 요구에 대한 효과적인 서비스가 가장 큰 해결점으로 대두되고 있으며, 이를 위한 고속의 서버 및 통신망은 필수적인 서비스 구성 요소들이라고 할 수 있다. 특히, VOD 서버에 대한 사용자 요구의 폭발적인 집중은 서비스 병목 현상을 초래하게 되며, 이에 대한 효율적인 서비스 스케쥴링 방법은 필수 불가결한 요소이다.

본 논문에서는 VOD 시스템에서 동일한 프리젠테이션에 대해 다중 사용자가 갖는 지역적, 시간적 국부성을 바탕으로 한 동적 캐쉬 스케쥴링 기법을 제안하고, 스케쥴링 기법에 대한 알고리즘을 기술하였다. 이러한 스케쥴링 방법은 각 사용자의 시청 진행 방향성을 고려하여 캐쉬 여부를 결정하기 때문에 보다 높은 캐쉬 적중률을 지닐 수 있도록 하며, 제한된 캐쉬 버퍼에 대해 그 활용도를 극대화시킬 수 있다는 것을 보였다. 또한, 시뮬레이션을 통해 VOD 시스템에 대해 본 논문에서 제안하는 동적 캐쉬 스케쥴링 방법의 적용이 기존의 캐쉬 알고리즘에 비해 보다 향상된 성능을 갖는다는 것을 입증하였다. 따라서, VOD 시스템에 대해 제안하는 캐쉬 스케쥴링 기법은 VOD 서버 및 통신망의 I/O 및 통신 대역폭을 줄일 수 있음은 물론이고, 보다 효율적이고, 경제적인 서비스를 가능하게 할 수 있을 것이다. 특히, VOD 서비스 망의 규모가 커지고 서비스 요구가 다양해질 때의 환경에서 보다 효과적인 스케쥴링 결과를 얻을 수 있을 것이다.

참고문헌

1. Christo Faloutsos, Christodoulakis, "Design and Performance Consideration for an Optical Disk-Based Multimedia Objects Server", ACM Transaction on Information Systems, Vol. 10, No. 12, pp.87-95, 1994.
2. Deepak R. Kenchammana-Hosekote, Jaideep Srivastava, "Scheduling Continous Media in a Video-On-Demand Server", Proceedings of the International Conference on Multimedia Computing and Systems, pp.19-28, 1994.
3. D.Venkatesh, T.D.C. Little, "Probablilistic Assignment of Movies to Storage Devices in a Video-On-Demand System", Proceedings of 4th International Workshop on Network and Operating System Support for Digital Audio

and Video, pp.213-224, 1992.

4. Dinesh Venkatesh, Thomas D. C. Little, "Prospects for Interactive Video-On-Demand", IEEE Multimedia, pp.14-24, 1994.
5. Dinkar Sitaram, Asit Dan, Perwez Shahabuddin, "Scheduling Poilicies for an On-Demand Video Server with Batching", Proceedings of the ACM Multimedia '94, pp.15-23, 1994.
6. Garth Gibson, Randy H. Katz, David A. Patterson, Peter Chen, "Introduction to Redundant Arrays of Inexpensive Disks(RAID)", COMPCOM-89, the 34th IEEE Computer Society International Conference, 1989.
7. George Homsy, David P. Andersons, "A Continuous Media I/O Server and Its Synchronization Mechanism", EEE Computer, Vol. 24, No. 10, pp.57-61, 1991.
8. James F.Kurose, Don Towsley, Jayan K. Dey-Sircar, James D.Salehi, "Providing VCR Capabilities in Large-Scale Video Servers", Proceedings of the ACM Multimedia '94, pp.25-32, 1994.
9. Mostafa H. Ammar, Kevin C. Almeroth, "Proving A Scalable, Interactive Video-On-Demand Service Using Multicast Communication", Proeedings of the ICCCM '94, pp.292-296, 1994.
10. Philip Lougher, Doug Shepherd, "The Design of a Storage Server for Countinous Media", The Computer Journal(special issue on multimedia), Vol. 36, No. 1, pp.32-42, 1993.
11. P.Venkat Rangan, Christos H.Papadimitriou, Srinivas Ramanthan, "Information Caching For Delivery of Personalized Video Programs on Home Entertainment Channels", Proceedings of the International Conference on Multimedia Computing and Systems, pp.214-223, 1994.
12. Srinivas Ramanthan, P. Venkate Rangan, Harrick M. Vin., "Designing an On-Demand Multimedia Service", IEEE Communications Magazine, Vol. 30, No. 7, pp.56-65, 1992.
13. Willem Verbiest, Danieli Delodder, Henri Verhille, "Interactive Video On Demand", IEEE Communications Magazine, pp.82-88, 1994.

李 承 潤(Seung Yun Lee) 정회원

1991年 2月 : 光云大學校 電子通信工學科 (工學士)
1995年 8月 : 光云大學校 電子計算學科 (理學碩士)
1995年 9月~現在 : 光云大學校 電子計算學科 博士課程 在學中

朴 昊 均(Ho Kyun Park) 정회원

1987년 2월 : 광운대학교 전자계산학과(이학사)
1989년 8월 : 광운대학교 대학원 전자계산학과(이학석사)
1993년 8월 : 광운대학교 대학원 전자계산학과 박사과정 수료
1992년 3월~현재 : 신흥전문대학 전자계산과 조교수

柳 煥 彬(Hwang Bin Ryou)정회원

1975年 2月 : 仁荷大學校 電子工學科(工學士)
1977年 7月 : 延世大學校 産業大學院 電氣電子工學科(工學碩士)
1989年 2月 : 慶熙大學校 大學院 電子工學科(工學博士)
1981年~現在 : 光云大學校 電子計算學科 教授
1995年~現在 : 新技術 研究所 研究員