

# 시리즈 비디오 데이터의 접근 패턴에 기반한 프록시 캐싱 기법

홍현옥\*, 박성호\*\*, 정기동\*\*\*

## 요 약

급격하게 증가하고 있는 인터넷 사용자들은 인터넷을 통하여 좋은 품질의 연속미디어 데이터를 제공받기를 원하고 있다. 그러나 서버의 과부하, 네트워크의 혼잡, 사용자에 대한 응답 지연 등의 문제는 이러한 요구를 충족시키지 못하고 있다. 본 논문에서는 이러한 문제점을 해결하기 위하여 연속미디어 데이터의 특성과 사용자 접근 패턴을 고려한 네트워크 캐싱정책(PPC, PPCwP)을 제안한다. 본 논문에서 제안하는 PPC 기법은 주기적으로 각 연속미디어 객체의 구간 재생량을 측정하여 인기도 분포를 결정하고, 이 분포에 따라 결정되어진 캐싱 데이터의 양만큼 객체의 앞부분부터 네트워크 캐싱을 수행한다. 또한 새로 생성된 객체의 인기를 예측하기 위하여 시리즈 정보를 사용하여 캐싱할 데이터의 양을 결정하고 사용자의 요구가 발생하기 전에 인기도가 급상승할 비디오를 네트워크 캐시에 프리캐칭하는 PPCwP 기법을 제안한다. PPCwP 기법은 시리즈 그룹의 정보를 사용하여 선반입함으로써 사용자의 요구에 대한 초기지연시간과 서버의 부하를 감소시키며, 제한된 캐쉬 공간을 효율적으로 활용할 수 있도록 한다. 제안하는 기법 PPC와 PPCwP의 성능을 평가하기 위하여 실제로 운영되고 있는 인터넷 방송국 VoD(Video on Demand) 서버(iMBC 인터넷 방송국)의 로그 데이터를 이용한 트레이스 기반 모의실험을 수행하였다. 그 결과 LFU나 LRU보다 초기지연 시간과 BHR 측면에서 우수한 성능을 보였다.

## Proxy Caching Scheme Based on the User Access Pattern Analysis for Series Video Data

Hyeon-Ok Hong<sup>†</sup>, Seong-Ho Park<sup>\*\*</sup>, Ki-Dong Chung<sup>\*\*\*</sup>

## ABSTRACT

Dramatic increase in the number of Internet users want highly qualified service of continuous media contents on the web. To solve these problems, we present two network caching schemes(PPC, PPCwP) which consider the characteristics of continuous media objects and user access pattern in this paper. While there are plenty of reasons to create rich media contents, delivering this high bandwidth contents over the internet presents problems such as server overload, network congestion and client-perceived latency. PPC scheme periodically calculates the popularity of objects based on the playback quantity and determines the optimal size of the initial fraction of a continuous media object to be cached in proportion to the calculated popularity. PPCwP scheme calculates the expected popularity using the series information and prefetches the expected initial fraction of newly created continuous media objects. Under the PPCwP scheme, the initial client-perceived latency and the data transferred from a remote server can be reduced and limited cache storage space can be utilized efficiently. Trace-driven simulation have been performed to evaluate the presented caching schemes using the log-files of iMBC. Through these simulations, PPC and PPCwP outperforms LRU and LFU in terms of BHR and DSR.

**Key words:** Caching(캐싱), Proxy(프록시), Continuous Media(연속미디어), Replacement(재배치), Internet(인터넷)

\* 교신저자(Corresponding Author): 홍현옥, 주소: 부산광역시 부산진구 양정1동(614-715), 전화: 051)860-3281, E-mail: hohong@dit.ac.kr

접수일: 2003년 8월 25일, 완료일: 2003년 12월 29일

<sup>†</sup> 정회원, 동의공업대학 컴퓨터정보계열

\*\* 종신회원, 부산대학교 정보전산원  
(E-mail: shpark@pusan.ac.kr)

\*\*\* 종신회원, 부산대학교 컴퓨터공학과  
(E-mail: kdchung@pusan.ac.kr)

## 1. 서론

최근의 기가비트 이더넷(gigabit ethernet)이나 광채널(fiber channel) 등의 네트워크 기술의 발달과 고성능 컴퓨터 부품들의 발달로 초고속 인터넷의 보급이 본격화 되고 있다. 그러나 인터넷을 이용하는 사용자들과 인터넷 상에서 서비스되는 VOD와 같은 대용량 연속미디어 파일들이 급격하게 증가함에 따라서 시스템 효율이 떨어지고 있다. 인터넷에서 오디오와 비디오와 같은 연속미디어(continuous media) 데이터는 텍스트나 이미지와 같은 이산미디어(discrete media) 데이터에 비하여 높은 전송 대역폭을 필요로 하고 오랫동안 주기적으로 전송해야 하므로 서버와 네트워크의 부하를 가중시켜, 패킷 손실과 지터(jitter) 현상이 발생한다. 이러한 문제점을 저렴한 비용으로 해결하고 사용자들이 요구하는 서비스 품질을 만족시키기 위하여 네트워크 상에서의 연속미디어 프록시 캐싱(Proxy caching) 기법이 연구되고 있다.

일반적으로 프록시 캐쉬(proxy cache)는 그림 1에서처럼 서버와 사용자 사이에 존재하며 사용자와 인접한 곳에 위치한다. 즉 프록시 서버는 LAN과 WAN이 연결되는 라우터와 직접 연결되는 곳에 위치하여 데이터를 캐싱함으로써, 서버로의 접근을 줄여 중앙서버의 부하와 WAN의 교통량을 감소시키고 사용자가 서비스를 기다리는 초기지연시간을 감소시킨다 [1-5].

연속미디어 프록시 서버는 서비스가 승인된 요구에 대하여서는 지터나 쫓김(eviction) 현상이 발생하지 않도록 서비스하여야 하기 때문에 연속미디어 서비스에 적절하게 부합할 수 있도록 접근 패턴을 반영한 캐싱 기법과 재배치 기법을 사용하여 시스템 효율을 향상시킬 수 있다. 특히 인터넷 방송국이나 연속미디어 포털사이트 등에서 서비스되는 VOD 데이터의 인기도는 생성된 후 일정 시간 동안은 급속하게 증가한 후에 지수분포로 감소하는 특성이 있다. 그러므로 새로 생성되는 객체에 대하여서는 인기를 예측하여 초기의 일부 데이터를 프록시 서버에 선반입하는 기법에 관한 연구도 필수적이다.

본 논문에서는 연속물 비디오 데이터의 네트워크 캐싱을 효과적으로 지원할 수 있는 프록시 캐싱 기법을 제안하고 성능을 분석하기 위하여 실제 운영중인 인터넷 방송국 연속미디어 서버의 사용자 로그 데이터를 활용하였다. 사용자 접근 패턴에 대한 분석은 데이터의 배치, 캐싱, 선반입(prefetching) 등의 기본 자료로 이용된다. 본 논문에서는 iMBC 인터넷 방송국 VOD 서버의 로그파일을 분석하여 구간 재생량과 생명주기에서의 생성 직후부터 정해진 시간까지의 평균구간 재생량을 함께 고려한 캐싱 기법을 제안한다. 구간 재생량은 시간  $t$ 에서 각 객체들이 가지는 접근 편기성을 나타내는 단기간의 인기도 패턴(short-term popularity pattern)을 나타내며, 평균구간 재생량은 같은 시리즈 그룹의 생명주기를 반영한

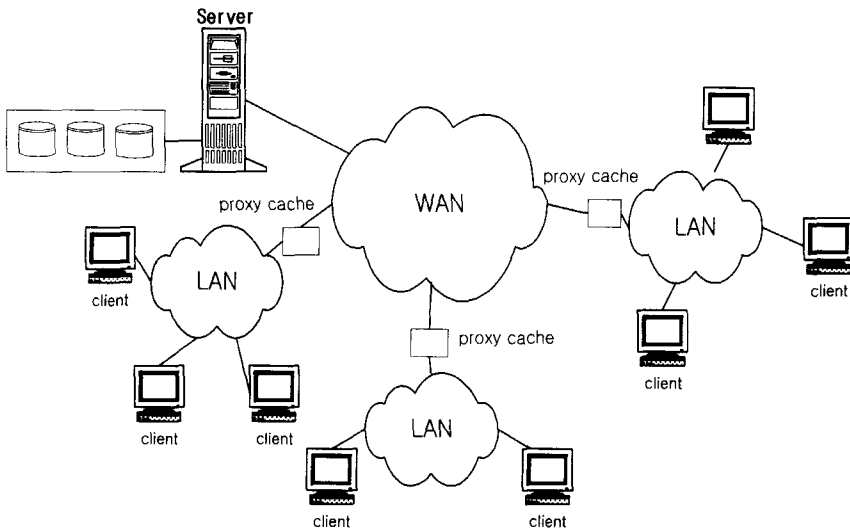


그림 1. 인터넷 상의 프록시 캐쉬

시리즈 정보에 따른 인기도 패턴을 나타낸다.

본 논문의 구성은 다음과 같다. 2장에서는 인터넷에서 프록시 서버의 캐싱 기법에 관한 연구를 살펴보고, 3장에서 연속미디어 서버의 트레이스 데이터(trace data)를 분석하여 시리즈 연속미디어 특성을 고려한 새로운 캐싱 기법을 제안한다. 시리즈 연속미디어 객체에 대한 사용자 접근 패턴을 반영한 인기도 모형을 제시하고 인기도에 따른 캐싱과 재배포 알고리즘을 구현한다. 4장에서는 실험을 통하여 기존의 다른 캐싱 기법들과 비교하여 제시한 기법의 성능을 평가하고, 마지막 5장에서 결론과 향후 연구방향을 제시한다.

## 2. 관련연구

### 2.1 연속미디어 데이터의 네트워크 캐싱

최근 들어 웹 문서를 위한 캐싱뿐만 아니라 연속미디어 데이터를 위한 네트워크 캐싱에 관한 연구도 진행되고 있다[6-11]. 이런 연구들은 전통적인 웹 프록시가 가진 결함을 보완하고 연속미디어를 효율적으로 서비스하기 위한 특별한 기술들을 제공한다.

[6]은 연속미디어 데이터에 대한 접근의 연속성을 고려하는 Interval caching의 개념을 프록시 캐쉬에 도입하였으며, 프록시의 캐쉬 공간과 대역폭을 프록시의 자원으로 규정하고 각 객체의 자원 요구량에 기반하는 캐싱 기법을 제안하였다.

[7]은 멀티미디어 스트림의 앞부분 일정 양만을 캐싱하여 사용자 요구의 초기지연을 감소시키면서 Workload smoothing을 수행하는 Prefix caching 기법을 제안하였다. [8]는 Layered-encoded stream을 이용하여 변화하는 네트워크 대역폭에 적응적으로 데이터를 전송하며 Layer 단위로 캐싱할 수 있는 기법을 제안하였고, [9]은 특정 양의 비디오 데이터를 미리 프록시에 선반입하여 서버로부터 WAN을 통해 전송되는 데이터의 양을 일정하게 만들고, 대역폭 smoothing을 수행하여 실질적인 네트워크 환경에서 고품질의 비디오 서비스를 제공할 수 있도록 하는 기법을 제안하였다.

그러나, 이러한 기존의 연구들은 대부분 프록시 서버를 이용하여 코어 네트워크와 서버의 부하를 감소시키는 기법을 제공하는데 초점을 맞추고 있으며 연속미디어 객체의 서비스 품질보장을 위한 캐싱 기법에 관한 연구는 아직도 미흡하다.

### 2.2 프록시 서버에서의 연속미디어 객체의 재배포 기법

재배포 알고리즘은 프록시 캐쉬의 효율성을 따지는 주요한 요인으로서 최근에 많이 연구되고 있다. HR(Hit Rate), BHR(Byte Hit Rate), 평균지연시간, 총 재배포 비용과 같은 다각적인 비용 측면에서의 매트릭스를 사용하여 다양한 재배포 정책이 제시되고 있다[10,11]. 이런 연구들은 전통적인 웹 프록시가 가진 결함을 보완하고 연속미디어를 효율적으로 서비스하기 위한 특별한 기술들을 제공한다. [10]은 연속미디어 데이터와 전통적인 미디어 데이터를 모두 효과적으로 처리하기 위한 프록시 서버의 고려 사항들을 나열하고, 여러 가지 다양한 기술들을 동적으로 결합하여 사용할 수 있어야 한다고 주장하였다. [9]와 [12]에서는 가장 낮은 해상도를 제공하는 기본 계층(base layer)에서 시작하여 점차 높은 해상도를 제공하도록 여러 계층으로 나누어 단계적으로 인코딩(layered encoding)한 연속미디어 객체를 대상으로 객체의 인기도에 따라 인기가 있을수록 보다 많은 계층이 캐싱되며, 이 때 캐싱되어 있지 않은 계층은 서버로부터 전송 받아 서비스된다. 그러나 이러한 기존의 연구들은 실시간으로 새로운 객체가 생성되는 연속미디어 서버의 환경과 사용자 요구의 특성을 고려하고 있지 않다. 프록시 서버에서의 재배포 정책의 성능은 웹 접근의 특성에 밀접하게 의존하기 때문에 웹 접근 패턴을 고려한 재배포 기법에 관한 연구가 필요하다.

### 2.3 프록시 서버에서의 선반입 기법

프록시 서버에서 현존하는 어떠한 캐싱 기법을 사용하더라도 40%~50% 이상의 성능향상을 기대하기 어렵다고 한다. 그러므로 캐쉬 참조율을 증가시킬 방법으로 선반입 기법을 함께 고려하기도 한다[1]. 선반입을 고려하기 전에 사용자 접근 패턴에 대한 분석을 하여야만 필요 없는 대역폭을 낭비하지 않게 된다. 사용자 접근 패턴을 정확하게 예측할 수 있는 가장 좋은 방법은 적합한 응용 분야의 자료를 수집하여 과거의 사용자 접근 패턴을 보고 분석하는 것이다.

웹 성능을 향상시키기 위하여 프록시가 사용되면서 관심은 대부분 프록시와 웹 서버 사이의 선반입 기술에 대한 연구로 집중되고 있다. 프록시 서버측에서의 선반입은 프록시 서버가 사용자들의 웹 접근 패턴을 분석하고 미래의 요구를 예측하여 예측되어

진 객체를 서버로부터 프록시 서버로 선반입하는 것을 의미한다[13-15]. [16]은 웹 서버와 프록시 사이에 선반입 기법을 적용하였을 때의 성능의 한계를 조사하고, 프록시 서버에서 이상적인 캐싱 기법과 이상적인 선반입 기법을 결합하여 사용하였을 경우 사용자의 요구에 대한 초기지연시간을 적어도 60% 감소시킬 수 있음을 보여주고 있다.

2.4 접근 패턴 분석에 관한 연구

사용자의 접근 패턴이나 인기도 모델을 사용할 경우 효율적인 캐싱 정책이나 배치 정책 등을 세울 수 있다. 현재, 텍스트나 이미지와 같은 웹 데이터가 인터넷을 통해 사용자에게 제공되는 웹 서버 및 웹 사용자의 부하에 관한 연구는 많이 이루어졌다[4]. 그러나 웹 서버 부하에 관한 연구는 연속미디어 데이터에 대한 사용자 접근의 특성을 반영하지 못하고 있다.

최근에 발표된 연속미디어 데이터를 대상으로 하는 서버의 부하 분석은 대부분 대학 캠퍼스나 특정 기업체와 같은 제한된 공간에서 특정 사용자들에 의해 접근되는 서버의 로그데이터를 중심으로 이루어졌다. [17]에서는 eTeach 시스템과 BIBS(Berkeley Internet Broadcasting System)의 로그데이터 분석을 통해 사용자 요구들 간의 도달 시간의 분포, 인기도에 따른 접속빈도의 시간적, 공간적 지역성(locality)에 대한 평가, 그리고 서버의 로드를 줄이기 위한 멀티캐스트 기술의 성능을 비교 분석하였다. [18]에서는 Hewlett-Packard 사의 각기 다른 특징을 가진 두개의 미디어 서버의 부하를 분석하였다. 두개의 미디어 서버에 대한 장기간에 걸친(2.5 years and 1 year 9 months) 사용자 접근 로그를 수집하여 동적이고 급격하게 사용자 접근 패턴을 변화시킬 수 있는 특징적인 요소로서 새로 생성된 파일의 영향(new files impact)과 사용자 접근 패턴의 변화 상태를 측정하기 위하여 생존기간(life span)을 제안하고 있다.

3. 시리즈 비디오 데이터의 접근 패턴에 기반한 프록시 캐싱 기법

본 장에서는 트레이스 데이터 분석을 통하여 시리즈 연속미디어의 특성을 살펴본다. 연속미디어 객체의 접근패턴에 따른 인기도인 구간재생량을 반영한

Popularity-based Proxy Caching 기법(PPC)과 높은 접근 편기성이 예상되는 새로 생성되는 객체에 대하여 시리즈 정보에 따른 인기도를 선반입에 반영한 Popularity-based Proxy Caching with Prefetching 기법(PPCwP)을 제안한다.

3.1 트레이스 데이터 분석

본 논문에서는 인터넷에서 서비스되고 있는 연속미디어의 특성과 연속미디어에 대한 사용자 접근 패턴을 분석하기 위하여 iMBC(http://www.imbc.com) 연속미디어 서버의 트레이스 데이터를 사용한다. iMBC는 정규 공중파 방송국인 MBC사의 방송 프로그램을 디지털화 하여 인터넷을 통하여 VOD 서비스를 제공하는 업체이다. 이 트레이스 데이터는 각 객체에 대한 사용자 요구의 재생 시작시간, 재생시간, 객체의 전체 재생시간 등의 정보를 가지고 있다. 본 절에서는 iMBC 로그 데이터를 분석하고, 분석한 데이터를 기초 자료로 하여 사용자 요구의 접근 특성을 제시한다.

3.1.1 트레이스 시리즈 비디오 데이터

표 1은 iMBC 연속미디어 서버가 보유한 객체의 특성을 보여준다. iMBC 연속미디어 서버는 매일 15개~20개의 새로운 연속미디어 객체를 생성한다.

표 1. 연속미디어 객체의 특성

구 분	특 성
측정기간 중 접근 객체의 수	2,900 개
측정기간 중 생성된 객체의 수	285 개
평균 전송률	100k/300k bps
연속미디어 객체의 재생 시간	5~120 min
측정기간 중 접근 객체의 전체 크기	약 290 기가바이트

3.1.2 시리즈 비디오 데이터의 특성

본 논문에서는 시리즈 연속미디어를 다음과 같이 정의한다.

(1) 주 5회씩 방영되는 일일드라마나 주 1~2회씩 방영되는 드라마와 같이 스토리가 연속성을 가지고 생성되어지는 연속미디어

(2) 같은 시간대에 같은 종류의 객체가 서비스되는 연속미디어

iMBC 인터넷 방송국 비디오 서버의 로그 데이터

에 나타난 시리즈 연속미디어 객체의 특성은 다음과 같다.

(1) 사용자는 새로 생성되는 연속미디어 객체를 선호한다. 표 2는 서버의 연속미디어 객체에 대한 사용자 요구의 특성을 보여준다. 최근에 생성된 객체에 대한 요구 비율이 상대적으로 매우 높다는 것은 사용자가 새로 생성되는 객체를 선호한다는 것을 나타내 준다.

(2) 시리즈 연속미디어 데이터의 경우 생성주기는 1일에서 1주일 단위이다. 생성주기가 1일인 객체들과 생성주기가 1주일인 객체들의 생명주기(life-cycle)는 크게 차이가 없다. 즉 새로운 객체가 생성되면 생성 직후부터 일정 시간 동안은 그 객체에 대한 인기도가 급격하게 상승하고 그 이후부터는 지수분포 형태로 감소한다. 그림 2와 그림 3은 측정기간 동안 각각 주 5회와 주 2회 생성되는 같은 시리즈에 속한 연속미디어 객체의 생명주기를 보여준다. 각 시리즈의 재생시간은 20분과 50분이며 측정기간 동안에 새로 생성되어진 연속미디어 객체들의 생명주기 패턴이 유사함을 보여준다.

(3) 인기 있는 비디오에 대한 접근 편기성이 높다. 그림 4는 iMBC 연속미디어 서버의 객체 중에서 상위 100위까지의 비디오의 재생량에 기반한 인기도를 나타낸 것이다. 인기 있는 비디오에 대한 접근 편

표 2. VOD 데이터에 대한 사용자 요구의 특성

구 분	요구 수	비 고
전체 객체에 대한 접근요구 수	1,887,064	100 %
측정기간 중 생성된 객체에 대한 접근요구 수	875,739	46.6 %

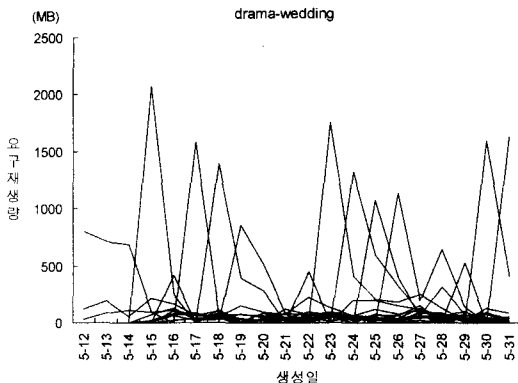


그림 2. 같은 시리즈에 속한 연속미디어 객체의 생명 주기(주 5회)

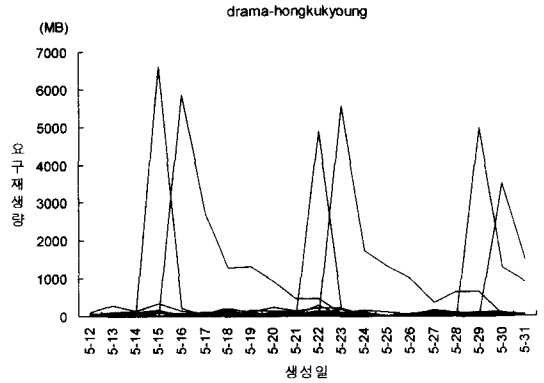


그림 3. 같은 시리즈에 속한 연속미디어 객체의 생명 주기(주 2회)

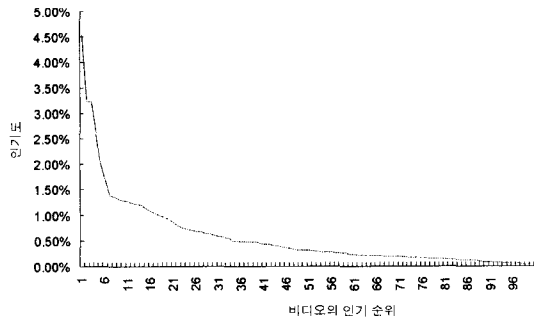


그림 4. 연속미디어 객체의 인기순위에 따른 재생량 분포 기성이 높다는 것을 알 수 있다.

### 3.2 Popularity-based Prefix Caching 기법

대용량의 연속미디어 데이터를 크기가 제한된 캐쉬에 캐싱할 때는 객체 단위로 캐싱하기보다 객체의 일부분을 세그먼트 단위로 캐싱할 수 있도록 하는 것이 바람직하다. Popularity-based Prefix Caching (PPC)는 객체의 인기도에 비례하도록 세그먼트 단위로 연속미디어 객체의 앞부분부터 캐싱을 수행한다. 많이 요구되는 객체일수록 프록시에 많은 양이 캐싱되어 있으므로 서버로부터 직접 전송되는 데이터양이 적다. 따라서 서버와 프록시 사이의 네트워크 교통량이 감소하며 서버의 부하도 감소한다. 또한, 다소 인기가 낮은 객체라 할지라도 앞부분의 약간의 데이터는 캐싱될 수 있으므로, 가능한 한 많은 사용자에 대한 서비스 초기지연시간이 감소된다.

PPC는 각 연속미디어 객체의 인기 분포와 캐싱되는 앞부분 데이터 양의 분포를 같게 하기 위하여 주

기적으로 프록시 내의 모든 객체의 인기도를 구간재생량(Short-term Popularity Pattern)에 기반 하여 측정하고, 측정된 인기도를 바탕으로 각 객체의 캐싱될 앞부분의 적정한 데이터양을 계산하여 유지한다. 그리고, 특정 객체에 대한 요구가 발생하여 전송을 할 때, 미리 계산되어 있는 그 객체의 캐싱할 적정 데이터양과 현재 캐싱되어 있는 앞부분 데이터의 양을 비교하여 뒷부분을 더 캐싱하고 재배치를 수행한다.

3.2.1 구간재생량에 기반한 인기도

본 절에서는 구간재생량에 기반하여 객체의 인기도를 계산하는 수식을 유도하며, 수식에 사용되는 기호에 대한 정의는 표 3과 같다.

그림 5는 t번째 주기에서 각 연속미디어 객체들의 구간재생량을 측정하는 것을 보여준다.

$DATA_{i,j}^t$  는 실선 구간동안에 사용자 j 가 객체 i 에 대해 재생한 데이터의 양이다. 주기 t 에서 객체 i 를 재생한 총 사용자의 수를  $C_i$  라 할 때, 주기 t 에서 객체 i 의 구간재생량  $DATA_i^t$  는 다음과 같다.

표 3. 수식에 사용되는 기호(1)

기 호	설 명
$S_{i,j}$	주기 t에서 사용자 j가 객체 i를 재생한 스트림
$DATA_{i,j}^t$	주기 t에서 사용자 j가 객체 i를 재생한 데이터의 양
$DATA_i^t$	주기 t에서 객체 i의 구간재생량
$D_i^t$	가중치를 부여한 주기 t에서 객체 i의 조정 재생량
$\alpha$	각 객체의 과거의 접근 정보를 사용하기 위한 가중치 값
$P_i^t$	주기 t에서 가 객체 i의 인기도

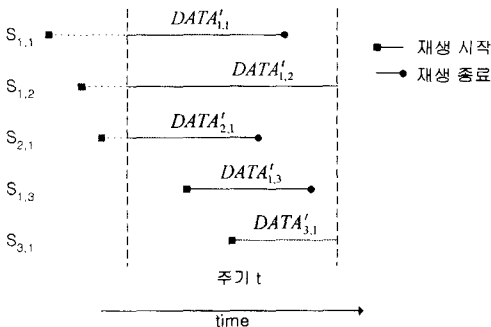


그림 5. 주기 t에서 각 연속미디어 객체의 구간재생량의 측정

$$DATA_i^t = \sum_{j=1}^{C_i} DATA_{i,j}^t \tag{1}$$

시리즈 연속미디어는 인기 있는 객체에 대한 접근 빈기성이 높으므로 프록시 캐쉬 용량을 일정 크기 이상1) 두면 그 구간에 접근되는 객체만으로는 캐쉬 저장 공간을 다 채울 수 없는 경우가 발생할 수 있다. 이러한 현상을 방지하기 위하여 프록시는 현재 캐싱되어 있는 각 객체들의 과거의 접근 정보를 사용한다. 과거의 접근 정보에 접근의 최근성을 반영하기 위하여 가중치를 사용하여 조정재생량인  $D_i^t$  값을 사용한다.

$$D_i^t = \alpha \cdot Data_i^t + \alpha \cdot (1-\alpha) \cdot Data_i^{t-1} + \dots + \alpha \cdot (1-\alpha)^{t-1} \cdot Data_i^1 + (1-\alpha)^t \cdot D_i^0 \tag{2}$$

$$\therefore D_i^t = \alpha \sum_{k=0}^{t-1} (1-\alpha)^k \cdot Data_i^{t-k} + (1-\alpha)^t \cdot D_i^0$$

여기에서 초기 추정 값  $D_i^0=0$  로 하면

$$D_i^t = \alpha \sum_{k=0}^{t-1} (1-\alpha)^k \cdot Data_i^{t-k}, \quad 0 < \alpha \leq 1 \tag{3}$$

계산된  $D_i^t$  값을 이용하여 주기 t에서 각 객체  $i, i \in [1..N]$ , 의 인기도  $P_i^t$  는 다음과 같이 계산할 수 있다.

$$P_i^t = \frac{D_i^t}{\sum_{k=1}^N D_k^t}, \quad 0 \leq P_i^t \leq 1 \tag{4}$$

3.2.2 캐싱과 재배치 수행

모든 객체들의 캐싱될 데이터양을 합한 것은 캐쉬의 크기와 같아야 하므로, 캐싱할 적정의 데이터양을 결정하는 것은 각 객체들에게 적정의 캐쉬 공간을 할당한다는 개념으로 볼 수 있다. 그리고 객체가 할당 받는 캐쉬 공간의 크기는 객체의 인기도에 비례하여야 한다. 그러나 3.2.1절에 제시되어진 인기도 수식

1) 본 논문에 트레이스 데이터를 이용하여 실험한 결과 10G Bytes 이상일 경우 재배치 척도 값 측정 구간에서의 정보만으로 전체 캐싱 공간을 활용하지 못하는 경우가 발생함.

을 그대로 사용하면 아주 인기가 높은 것은 그 객체의 길이보다 더 많은 공간을 할당 받아서 공간이 남게 될 수 있으며 인기가 낮은 것들은 거의 공간을 할당 받지 못하게 된다. 따라서 인기도를 다음과 같이 수정하고 인기도가 높은 객체부터 차례대로 캐쉬 할당량을 계산한다. 수식에 사용되는 기호에 대한 정의는 표 4와 같다.

$$P_i = \frac{D_i^t}{\sum_{k=1}^{i-1} D_k^t}, \quad \text{단, } D_i^t \geq D_{i+1}^t \quad (5)$$

그리고 각 객체의 캐싱할 최적의 데이터양은 다음과 같이 계산된다.

$$s_i^t = \min\{\hat{P}_i^t \times S, L_i\}, \quad i = 1 \quad (6)$$

$$s_i^t = \min\{\hat{p}_i^t \times (S - \sum_{k=1}^{i-1} s_k^t), L_i\}, \quad 1 < i \leq N \quad (7)$$

프록시는 계산 오버헤드를 감소시키기 위하여 각 객체의 인기도와 캐싱할 최적의 데이터 양을 주기적으로 계산해서 테이블로 유지한다. 그리고 특정 객체에 대한 사용자의 요구가 발생하여 전송을 할 때, 이미 계산되어 있는 값과 실제 캐싱되어 있는 데이터의

양을 비교하여 그 차이 값만큼 캐싱 또는 재배포를 수행한다.

PPC는 관리의 용이성을 위하여 세그먼트 단위로 캐싱과 재배포를 수행한다. 세그먼트 단위로 캐싱과 재배포를 수행하면 현재 프록시에 캐싱되어야 할 각 객체의 크기가 seg-size 보다 적으면 그 객체는 프록시에서 완전히 제거된다. 이렇게 하면 생명주기가 끝나가는 많은 객체들이 조금씩 프록시에 남아서 저장 효율을 감소시키는 현상을 방지할 수 있다.

캐싱과 재배포의 수행을 위한 자세한 알고리즘은 그림 6에 나타나 있다.

표 4. 수식에 사용되는 기호(2)

기호	설 명
$L_i$	객체 $i$ 의 전체 길이
$s_i^t$	객체 $i$ 의 캐싱할 최적의 데이터 양
$S$	캐쉬의 전체 크기
$Seg_i^t$	객체 $i$ 에 할당된 캐쉬의 세그먼트 수
seg-size	캐쉬 세그먼트의 크기
$\hat{P}_i^t$	주기 $t$ 에서의 객체 $i$ 의 수정된 인기도

```

cache_data (object  $i$ )
{
    if ( $s_i^{t-1} \geq s_i^t$ ) return;
     $seg_i^t = (s_i^t - s_i^{t-1}) / seg\_size$ ;

    if (free segments number in cache <  $Seg_i^t$ ) {
         $E = \{j : s_j^{t-1} > s_j^t, \quad 1 \leq j \leq N\}$ ;

        while(free segments number in cache <  $Seg_i^t$ ) {
             $j \in E$  인 임의의 객체  $j$ 를 선택;
             $j$ 의 캐싱된 부분 중 뒤쪽  $(s_j^{t-1} - s_j^t) / seg\_size$  세그먼트를 캐쉬에서 삭제;
             $E = E - \{j\}$ ;
        }
    }

    object  $i$ 의 뒷부분  $Seg_i^t$ 의 세그먼트를 추가로 더 캐싱;
    return;
}
    
```

그림 6. PPC 기법에서의 재배포 알고리즘

### 3.3 Popularity-based Prefix Caching with Prefetching 기법

Popularity-based Prefix Caching with Prefetching(PPCwP) 기법은 높은 접근 편기성이 예상되는 새로 생성되는 객체에 대하여 이전 시리즈물의 시리즈 정보에 의해 계산된 인기도에 따라 적절한 양 만큼 선반입한다. 사용자는 새로운 연속미디어 데이터를 선호하므로 대부분의 사용자의 요구는 최근에 생성된 비디오 객체에 집중되고 생성된 후 2~3일이 지난 비디오 객체는 거의 접근되지 않는다. 그러므로 사용자가 요구하기 전에 인기도가 급상승할 연속미디어 객체를 캐시에 선반입하게 되면 참조율이 향상되고 초기지연시간을 단축시킬 수 있다.

그림 7은 실험 기간동안 새로 생성되어진 시리즈 연속미디어 객체들의 생성 직후부터의 전반적인 생명주기를 보여준다. 21일간의 접근패턴을 관찰하기 위하여 24시간(1일) 단위로 재생된 데이터양의 변화를 그래프로 표현하였다. 비디오 데이터에 대한 사용자 접근 패턴의 일반적인 특징이 잘 나타나 있다. iMBC의 시리즈 연속미디어의 경우, 가장 재생량이 많은 시기는 생성된 후 1일 동안으로 그 동안은 급격하게 증가 하였다가 1일이 지난 후부터는 지수분포 형태로 감소하고 있다.

따라서 문제는 연속미디어 객체가 생성된 후 인기도가 급상승하는 시점이다. 이 시점부터 요구량이 급속하게 증가하므로 새로 생성된 비디오 객체에 대해 그 그룹의 이전 시리즈 객체들의 정보에 따라 일부를 선반입한다면 시스템 성능을 향상시킬 수 있다.

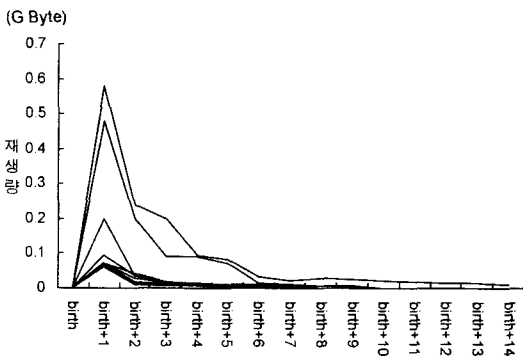


그림 7. 생성된 후 재생량을 기준으로한 인기도의 변화

#### 3.3.1 시리즈 정보를 이용한 인기도

선반입 기법을 효과적으로 적용하기 위해서는 사용자들의 연속미디어 객체에 대한 접근 패턴을 정확하게 반영해야 한다. 연속미디어 객체에 대한 접근 패턴을 반영하기 위하여 선반입 시스템은 가장 최근의 접근 로그에 기반하여 주기적으로 접근 패턴을 분석하고, 선반입되어야 할 웹 객체의 리스트를 생성해야 한다.

각 시리즈 연속미디어 그룹의 다음 객체가 생성되면 얼마만큼 선반입하는 것이 효율적인가를 고려하기 위하여, 본 논문에서는 새로 객체가 생성되면 그 그룹의 이전 시리즈 객체들의 생성 직후 첫 구간재생량의 평균값을 사용하여 예상되는 인기도 만큼 선반입한다.

그러므로 새로 생성된 객체  $Se_j^k$ 의 초기구간에 선반입할 예상재생량  $\overline{UP}_j^k$ 은 다음과 같이 나타낼 수 있다. 수식에 사용되는 기호에 대한 정의는 표 5와 같다.

$$\begin{aligned} \overline{UP}_j^k &= 0, \quad k = 1 \\ \overline{UP}_j^k &= \sum_{n=1}^{k-1} UP_j^n / (k-1), \quad k > 1 \end{aligned} \tag{8}$$

그림 8은 선반입에 사용되는 이전 시리즈들의 구간재생량을 보여준다.

표 5. 수식에 사용되는 기호(3)

기호	설명
$Se_j^k$	시리즈 $j$ 의 일련번호(횟수)가 $k$ 인 객체
$UP_j^k$	기존에 생성된 객체 $Se_j^k$ 가 생성된 주기에서의 재생량
$\overline{UP}_j^k$	새로 생성된 객체 $Se_j^k$ 가 생성된 주기에서의 예상 재생량

#### 3.3.2 선반입과 재배치 수행

같은 시리즈 연속미디어를 하나의 그룹으로 하여 PPCwP기법은 같은 시리즈 그룹의 이전에 생성된 객체들의 생명주기 곡선에 따른 시리즈 정보를 사용하여 예상되는 인기도를 계산한다. 계산된 적정의 양 만큼 선반입하고 현재 캐싱되어 있는 객체들의 조정



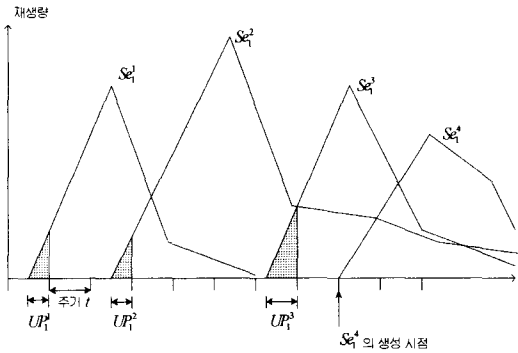


그림 8. 선반입에 사용되는 이전 시리즈들의 구간재생량  
된 인기도를 계산하여 재배포를 수행한다. 같은 시리즈 그룹의 새로운 객체가 생성되면 선반입할 인기도 값은 프록시 서버에 의하여 동적으로 조정된다. 선반입할 값이 정해지면 캐쉬 내의 객체들에 대한 가장 최근의 구간재생량에 기반하여 조정된 인기도를 계산한 후 PPC 기법에서와 같이 재배포를 수행한다. 같은 시리즈 그룹의 연속미디어 객체가 새로 생성되는 시점이 선반입과 재배포가 수행되는 시점이다.

선반입과 재배포 수행을 위한 알고리즘은 그림 9에 나타나 있다.

```

prefetch_data (object i)
{
    객체 i의 시리즈 번호 j와 횟수 k를 구함
    시리즈 테이블로부터  $\overline{UP}_j^k$  를 구함.
    If(  $\overline{UP}_j^k = 0$  ) return;
     $D_i^j = \overline{UP}_j^k$ 
    PPC 정책에 따라 재배포를 수행함.
    return;
}
    
```

그림 9. PPCwP 기법에서의 선반입 알고리즘

#### 4. 실험 및 성능 분석

본 장에서는 본 논문에서 제안한 네트워크 캐싱 정책인 PPC와 PPCwP의 성능을 실험을 통하여 기존 알고리즘과 비교하여 분석한다.

##### 4.1 실험 환경

연속미디어 프록시 서버 프로토타입은 Sun Enterprise-250에서 구현하였다.

본 논문에서는 연속미디어 프록시 서버의 실험 환경을 다음과 같이 가정하였다.

(1) 연속미디어 프록시 서버의 CPU자원, 메모리 자원, 저장장치(Hard Disk 등)의 대역폭 등은 충분하다고 가정하며, 저장공간의 크기만을 고려한다.

(2) 프록시와 사용자 사이의 전송지연은 100ms, 서버와 사용자 사이의 전송지연은 200ms로 가정한다.

(3) 원격지 서버와 프록시 서버간의 전송 오류 및 전송 오버헤드는 무시한다. 즉, 할당된 네트워크 대역폭은 데이터 전송에만 사용된다고 가정한다.

(4) 요구에 의하여 발생하는 모든 스트림의 대역폭은 300kbps로 가정한다.

(5) 본 논문에서 사용한 트레이스 데이터는 20대의 iMBC 연속미디어 서버 중 임의의 1대를 선택하여 21일간 수집한 것이다.

##### 4.2 PPC의 성능 분석

프록시 캐싱의 목적은 원격지 서버가 사용자의 요청에 의해 객체를 사용자 노드로 전송하기 위하여 필요한 자원의 효율적인 운영에 있다. 일반적으로 네트워크 캐싱의 성능은 ①사용자 노드 측의 초기지연, ②서버 자원의 효율성, ③코어 네트워크의 사용도 등의 세가지 측면에서 평가할 수 있다[13]. 서버 자원의 효율성과 코어 네트워크 사용도는 BHR(Byte Hit Rate)와 밀접한 관계에 있으며, 프록시 서버의 효율성은 BHR, 재배포데이터량과 밀접한 관계를 가지고 있다. 그러므로 본 논문에서는 연속미디어 네트워크 캐싱의 성능을 ①초기지연시간, ②BHR, ③재배포데이터량을 측정 기준으로 평가한다.

그림 10은 캐쉬 크기의 변화에 따른 LRU 알고리즘, LFU 알고리즘과 본 논문에서 제안하는 PPC 알고리즘의 성능을 초기지연시간, BHR, 재배포 데이터량으로 비교 분석한 결과를 보여준다. LRU 알고리즘과 PPC 알고리즘의 재배포 척도값 재설정 시간 간격은 20분으로 하였다.

PPC 알고리즘은 객체 단위로 캐싱을 수행하는 LFU 알고리즘보다 초기지연시간 면에서 12%~6%의 성능 향상을 보이고 있다. 캐쉬 크기가 적을수록 PPC 알고리즘과 LFU 알고리즘의 성능 차이가 크다는 것을 알 수 있다. PPC 알고리즘은 BHR에서도 LFU 알고리즘보다 약 7.5%~5% 정도의 높

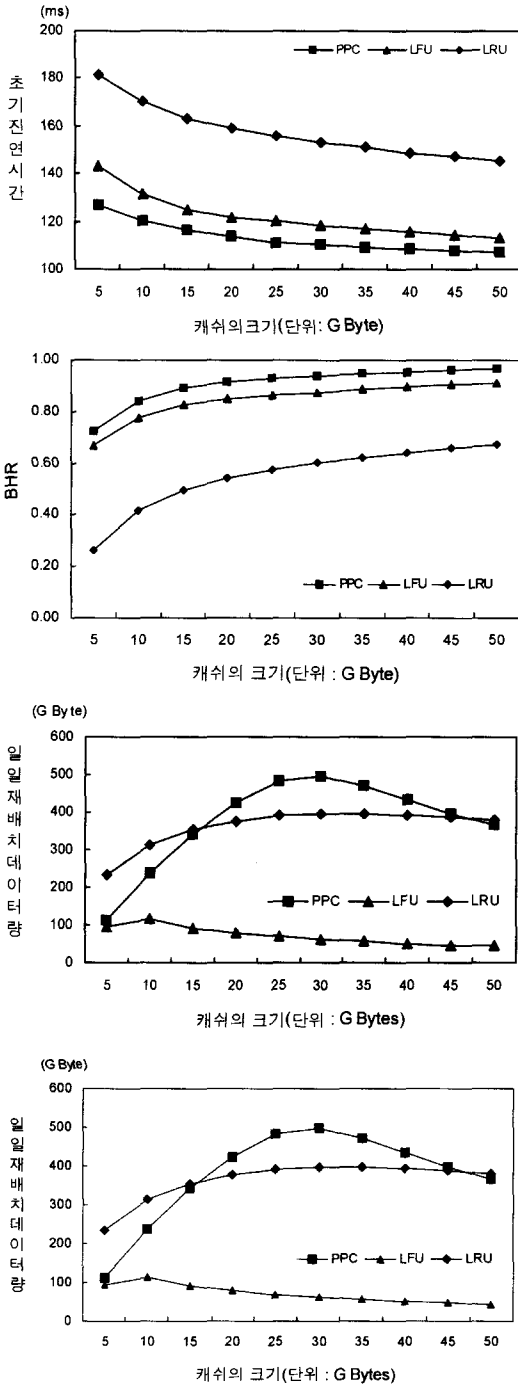


그림 10. 캐쉬 크기의 변화에 따른 PPC, LFU, LRU의 성능 비교

은 성능을 보여 주고 있다. PPC는 인기도에 비례하여 객체의 앞부분을 부분적으로 캐싱함으로써 인기도에 민감하게 반응함은 물론이고 객체 단위로 캐

싱하는 LFU나 LRU 보다 많은 수의 객체들을 프록시에 캐싱할 수 있으므로 초기 지연 시간과 BHR의 효율은 좋아지는 반면 재배치 데이터량은 늘어난다. 즉 접근 효율의 개선과 재배치로 인한 오버헤드는 반비례 관계에 있다는 것을 알 수 있다.

### 4.3 PPCwP의 성능 분석

그림 11은 PPCwP 알고리즘의 성능을 PPC 알고리즘과 비교 분석한 결과이다. 이 실험에서도 두 알고리즘의 재배치 척도값 재설정 시간 간격은 20분으로 하였다.

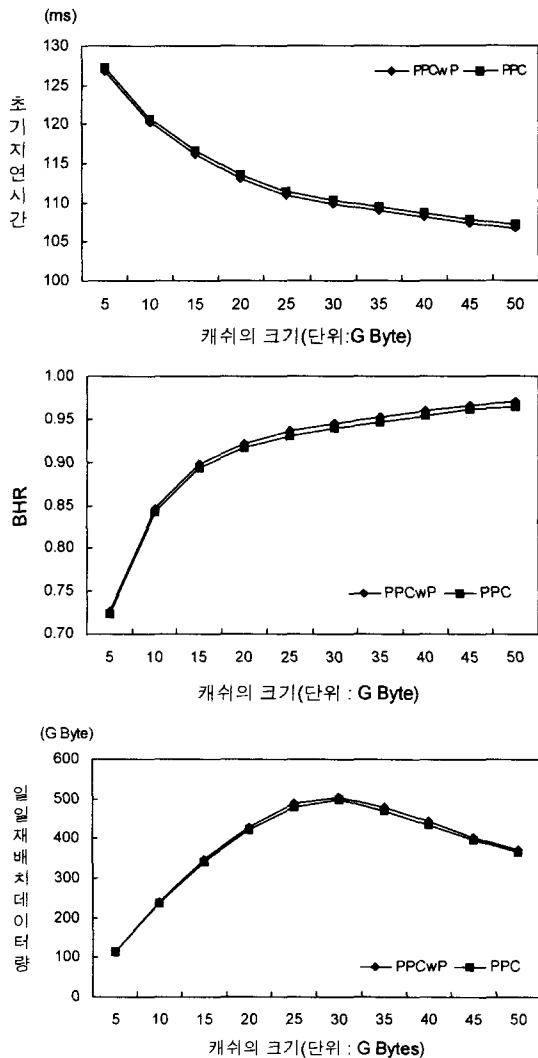


그림 11. 캐쉬 크기의 변화에 따른 PPCwP과 PPC의 성능 비교

새로운 객체가 생성되면 PPCwP는 그 객체의 예측된 인기도만큼 선반입한다. 이때 선반입 되는 데이터양은 PPC의 재배치 설정주기 동안의 같은 시리즈 그룹에 속한 모든 객체들의 초기구간의 평균 값이다. 다음 주기부터는 PPC의 인기도에 따라 재배치가 수행되므로 두 알고리즘의 성능 차이는 크게 나지 않으나 PPCwP가 구간간에 걸쳐서 하나의 요구 당 초기지연시간은 1ms 정도, BHR에서는 1% 정도의 성능이 향상되었음을 보여 준다. 시리즈 연속미디어 객체는 생성 직후부터 인기도가 상승하기 시작하면 12시간 정도 계속 증가하므로 선반입 구간 설정에 대한 연구가 보완된다면 재배치 데이터량을 감소시킬 수 있을 것이다.

## 5. 결론 및 향후 연구과제

본 논문은 실제 웹상에서 서비스 되고 있는 연속미디어 서버(iMBC 인터넷 방송국)의 접근로그를 분석하여 시리즈 연속미디어의 특성을 정의하고 요구 재생량에 기반한 객체의 인기도를 제시하였다. 그리고 시리즈 연속미디어의 특성에 따른 인기도를 구간 재생량과 시리즈 정보를 사용하여 모델링하였다.

제안하는 PPC와 PPCwP의 성능을 평가하기 위하여 실제로 운영되고 있는 인터넷 방송국 VOD 서버의 로그 데이터를 이용한 트레이스 기반 모의 실험을 수행하였다. 그 결과 PPC와 PPCwP는 인기도에 비례하여 객체의 앞부분을 부분적으로 캐싱함으로써 인기도에 민감하게 반응함은 물론이고 객체 단위로 캐싱하는 LFU나 LRU보다 많은 수의 객체들을 프록시에 캐싱할 수 있으므로 초기지연시간에서 12%~6%의 성능 향상과 BHR에서 약 7.5%~5% 정도의 높은 성능 향상을 보여 주었다.

본 연구와 관련하여 향후 연구과제는 다음과 같다.

(1) 네트워크 대역폭의 효율적인 사용을 위하여 시간대를 고려한 선반입 기법에 대한 연구가 필요하다.

(2) 재배치 데이터양을 감소시키기 위한 연구가 필요하다. 본 논문에서 제시하고 있는 PPCwP기법에서 사용하는 구간의 크기는 재배치 설정 시간과 동일하게 설정되어 있다. 인기도가 증가하기 시작하면 12시간 정도 계속 증가하므로 재배치 효율을 높일 수 있도록 선반입 구간 설정에 관한 연구가 필요하다.

(3) 각 객체의 생명주기 곡선을 사용하여 현재주

기에서 다음주기의 성장곡선을 예측할 수 있는 모형에 관한 연구가 필요하다.

(4) 이산미디어와 연속미디어가 혼합된 환경에서 프록시 서버의 캐싱 방법에 대한 연구와 성능 측정 방법 관한 연구가 필요하다.

## 참고 논문

- [1] M. Abrams, C. Standridge, G. Abdulla, S. Williams and E. Fox, "Caching Proxies: Limitations and Potentials," Proc. of the Fourth International World Wide Web Conference, Boston, 1995.
- [2] Braun H. and K. Claffy, "Web Traffic Characterization: An Assessment of the Impact of Caching Documents from NCSA's Web Server," Proc. of the Second International Conference on the World Wide Web, Chicago, 1994.
- [3] A. Luotonen, K. Altis, "World Wide Web Proxies", In Proc. of the First International Conference on the WWW, pp.147-154, 1994.
- [4] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications," Proc. IEEE INFOCOM '99, New York City, NY, March 1999.
- [5] S. Glassman, "A Caching Relay for the World Wide Web," Proc. of the First International World Wide Web Conference, Geneva, 1994.
- [6] R. Tewari, H. M. Vin, A. Dan, D. Sitaram, "Resource-based caching for Web servers", In Proc. SPIC/ACM Conference on Multimedia Computing and Networking, 1998.
- [7] S. Sen, J. Rexford and D. Towsley, "Proxy prefix caching for multimedia streams," In Proc. IEEE Infocom, 1999.
- [8] Reza Rejaie, Haobo Yu, Mark Handely, Deborah Estrin, "Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet", In Proc. of IEEE Infocom '2000, 2000.

[9] Y. Wang, Z.-L. Zhang, D. Du, and D. Su, "A Network-Conscious Approach to End-to-End video Delivery over Wide Area Networks Using Proxy Servers", In Proc. IEEE Infocom, 1998.

[10] S. Sahu, P. Shenoy and D. Towsley, "Design Considerations for Integrated Proxy Servers", In Proc. IEEE NOSSDAV '99, 1998.

[11] M. Reisslein, F. Hartanto, K. W. Ross, "Interactive Video Streaming with Proxy Servers", In Proc. IEEE Infocom, 2000.

[12] R. Rejaie, M. Handley, H. Yu, and D. Estrin, "Proxy Caching Mechanism for Multimedia Playback Streams in the Internet," Proc. of the 4th International Web Caching Workshop, Mar. 1999.

[13] Ken-ichi Chinen, Suguru Yanaguchi, "An Interactive Prefetching Proxy Server for Improvement of WWW Latency," INET97, 1997.

[14] Wcol Group, "WWW Collector the Prefetching Proxy Server for WWW ", 1997.

[15] L. Fan, P. Cao, W. Lin and Q. Jacobson, "Web Prefetching Between Low-Bandwidth Clients and Proxies: Potential and Performance," ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp.178-187, 1999

[16] T. M. Kroege, D. D. E. Long, and J. C. Mogul, "Exploring The Bounds of Web Latency Reduction from Caching and Prefetching," Proc. of the Usenix Symposium on Internet Technologies and Systems, Monterey, CA, Dec. 1997.

[17] Jussara M. Almeida, Jeffrey Krueger , Derek L. Eager and Mary K. Vernon, "Analysis of educational media server workloads," 11th International workshop on Network and Operating Systems support for digital audio

and video January 2001

[18] Ludmila Cherkasova, Minaxi Gupta, "Analysis of Enterprise Media Server Workload: Access Patterns, Locality, Dynamics, and Rate of Change", ACM NOSSDAV 2002.

[19] Jia Wang, "A Survey of Web Caching Schemes for the Internet", ACM Computer Communication Review, 29(5) pp.36-46, 1999.



**홍 현 옥**

1983년 2월 부산대학교 영문학과 학사  
 1986년 12월 미 텍사스 주립대학 전산학 석사  
 2003년 2월 부산대학교 전자계산학과 박사  
 1995년~현재 동의공업대학 컴

퓨터정보계열 조교수

관심분야 : 병렬처리 및 멀티미디어, 멀티미디어 콘텐츠



**박 성 호**

1996년 부산대학교 전자계산학과 졸업(학사)  
 1998년 부산대학교 대학원 전자계산학과 졸업(이학석사)  
 2002년 부산대학교 대학원 전자계산학과 졸업(이학박사)  
 2002년 9월~현재 부산대학교

정보전산원 조교수

관심분야 : VOD 시스템, 인터넷 캐싱, 멀티미디어 이동통신



**정 기 동**

1973년 서울대학교 졸업(학사)  
 1975년 서울대학교 대학원 졸업(석사)  
 1986년 서울대학교 대학원 계산통계학과 졸업(이학박사)  
 1990년~1991년 MIT, South Carolina 대학 교환 교수

1995년~1997년 부산대학교 전자계산소 소장  
 1978년~현재 부산대학교 전자계산학과 교수

관심분야 : 병렬처리, 멀티미디어