스트리밍 캐쉬 서버를 위한 가중치 윈도우 기반의 캐쉬 교체 정책

(A Weighted-window based Cache Replacement Policy for Streaming Cache Server)

오 재 학 † 차 호 정 †† 박 병 준 †††
(Jaehak Oh) (Hojung Cha) (Byungjoon Park)

요 약 본 논문에서는 스트리밍 미디어 캐싱 서버의 효율적인 캐싱을 위하여 가중치 윈도우에 기반한 캐쉬 교체 정책을 제시하고 성능을 분석한다. 제안된 캐쉬 교체 정책은 참조 횟수, 참조량, 참조 시간 등의 정량적인 인자들과 사용자 요구 주기를 적용하여 기존 캐쉬 교체 정책과 차등화된 개념을 도입하였으며, 최근 참조 경향에 높은 가중치를 부여함으로써 변화하는 콘텐츠 선호 경향에 빠르게 적응하는 구조를 제시하였다. 교체 정책 성능 분석은 시뮬레이션 환경 구축을 통해 실험하였으며 기존의 캐쉬 교체 정책인 LRU, LFU와 SEG보다 참조 적중률, 참조량 적중률, 시작 지연률과 반입량에서 향상된 결과를 보였다.

키워드 : 스트리밍 캐쉬 서버, 캐쉬 교체 정책

Abstract This paper presents and analyzes the performance of a weighted-window based cache replacement policy for the efficient media caching in streaming media cache servers. The proposed policy makes, for each cached object, use of the reference count, reference time, amount of media delivered to clients and, in particular, the periodic patterns of user requests. Also, by giving weights to the recently referenced media contents, the replacement policy adequately and swiftly reflects the ever-changing characteristics of users preferences. The simulation studies show that the performance of the proposed policy has improved over the conventional policies such as LRU, LFU and SEG - in terms of hit ratio, byte hit ratio, delayed start and cache input.

Key words : Streaming cache server, Cache replacement policy

1. 서 론

최근 동영상 스트리밍 서비스는 다양한 콘텐츠 개발과 사용자 증가로 보편화 추세에 있지만, 상대적인 네트워크 트래픽 증가로 사용자 대기 시간과 스트리밍 서버 부하를 증가시키고 있다[1]. 따라서, 네트워크 트래픽 증가에 따른 사용자 QoS 보장을 위해 선호하는 콘텐츠를 대상으로 콘텐츠 일부를 저장하고 관리하는 동영상 미디어 캐싱의 필요성이 대두되고 있다. 동영상 미디어 캐싱 시스템은 사용자와 가까운 지역 네트워크에 위치하여 사용자가 선호하는 콘텐츠를 빠르게 전송하고 트래픽을 지역 네트워크로 한정하는 효과를 갖는다. 이러한 캐싱 시스템 성능은 제한된 저장 공간에서 콘텐츠 혹은 콘텐츠 일부에 대한 가감을 관리하는 효율적인 캐쉬 교체 정책에 따라 많은 영향을 받는다[2].

동영상 미디어 캐싱은 기존 웹 캐싱 환경의 캐쉬 오브젝트와 구별되는 대용량, 재생 연속성, 실시간의 특징이 있으며, 캐쉬 교체 정책은 사용자 참조 경향에 따른 오브젝트 크기의 가변성이 고려되어야 한다. 일반적으로 운영체제와 웹 환경과 같은 기존의 캐싱 분야는 대상 콘텐츠의 전체를 캐쉬 오브젝트로 관리하는 정적인 캐쉬 관리 기법에 근거한 교체 정책을 적용한다. 운영체제 캐쉬는 메모리 상에 균등 분할된 페이지에 기반하여 메모리 지역 참조 특성에 효율적인 LRU, LFU 등의 교체 정책을 적용하고 참조 적중률에 근거한 효율성으로 평가된다. 웹 캐쉬는 다양한 콘텐츠에 따른 이종 크기 콘텐츠를 대상으로 참조 시간과 오브젝트 크기 등을 고려한 교체 정책이 적용되고 네트워크 서비스로써 참조 적

· 이 논문은 2002학년도 연세대학교 학술연구비의 지원에 의하여 이루어진 것임
† 비 회 원 : (주)코어세스
ojh@corecess.com
†† 종신회원 : 연세대학교 컴퓨터과학과 교수
hjcha@yonsei.ac.kr
††† 비 회 원 : 광운대학교 컴퓨터과학과 교수
bjpark@cs.kwangwoon.ac.kr
논문접수 : 2002년 8월 23일
심사완료 : 2003년 7월 18일

중률, 참조량 적중률과 웹 서버로부터의 콘텐츠 반입량에 따른 효율성으로 평가된다. 반면, 동영상 미디어 캐쉬에서는 대용량 콘텐츠의 완전(full) 캐쉥으로 인한 저장 공간 낭비를 막기 위해 사용자 참조 경향에 따라 캐쉬 오브젝트의 크기를 조절할 수 있는 교체 정책을 필요로 한다. 즉, 캐쉬 교체 정책은 참조 빈도와 평균 참조량에 따라 오브젝트 크기를 결정해야하며, 그 성능 측정 기준은 웹 캐쉬에서와 유사하지만 재생 연속성과 실시간성을 고려한 사용자 시작 지연등의 사용자 QoS가 고려되어야 한다.

일반적으로 캐쉬 교체 정책은 과거 참조 경향을 일반화하여 효율적인 캐쉬 오브젝트를 구성한다. 이와 같은 측면에서 동영상 미디어의 캐쉬 교체 정책은 기존 캐쉬 적용 환경과 구별되며 오브젝트 특성과 사용자 경향이 반영된 일반화된 정책이 요구된다. 정량적인 인자에 기반한 운영체제 캐쉬와 웹 캐쉬는 요구와 응답에 대한 시간 제한성이 적고 일반적으로 수 바이트에서 수 메가 바이트 사이로 작아서 응답 지연에 대한 체감적인 지터(jitter)가 낮다. 또한, 시간대에 상관없는 접근성으로 주기적인 요구 경향이 약하다. 반면, 동영상 콘텐츠의 사용자 참조는 시간 제한적 서비스(time constrained service)이며 수 분에서 수 시간 동안 참조되는 경향이 있고, 일반적으로 일정한 생활 주기를 갖는 사용자들은 동영상 콘텐츠에 대한 참조 경향과 시간대 요구 편중성이 높을 것이다. 이러한 문제점은 현재와 과거 시점에 따라 각 주기의 가중치를 차등 적용함으로써 해결 가능하며 미래 참조에 대한 캐쉬 교체 정책의 효율성을 높일 수 있다.

본 논문과 관련된 연구에는 웹 캐쉥과 동영상 미디어 캐쉥 분야가 있다. 웹 캐쉥의 캐쉬 교체 정책에서는 참조 시간, 참조 빈도, 전송 시간, 오브젝트 크기에 따른 LRU, LFU, LAT, SIZE를 기반 정책으로 웹 오브젝트의 특성을 반영한 Hyper-G[3], Pitkow-Recker[4], LRU-Threshold[5], Log(Size)+LRU[5], LUV[6] 등이 제시되고 있다. Hyper-G 알고리즘은 최근 참조 시간과 오브젝트 크기에 기반한 LFU기법을 사용하고 Pitkow/Recker 알고리즘은 24시간 동안 참조한 오브젝트를 제외한 나머지 오브젝트에 대해서 LRU을 적용한다. LRU-Threshold는 LRU에 기반하면서 정해진 크기를 넘어서는 오브젝트를 캐쉥 대상에서 제외한다. Log(Size)+LRU는 Log(Size)값이 가장 크고 최근에 참조되지 않은 오브젝트를 희생 대상으로 선택한다. LUV는 대상 오브젝트의 캐쉥 시점부터 현재 시점까지의 참조 경향을 재참조 잠재성으로 일반화하고 최근 참조 경향에 높은 가중치를 부여한다. 그 외에 평균 지연시간을 고려한 Lowest-Latency-First[7], 오브젝트의 인기도와 연관성을 고려한 GreedyDual*[8], 오브젝트의 크기와 마지막 참조 시간을 함수로 표현한 LRV(Lowest Relative Value)[9] 등의 캐쉬 교체 정책이 있다. 동영상 미디어를 대상으로 하는 캐쉬 교체 정책으로는 세그먼트를 이용한 캐쉬 교체 정책(SEG)이 있다[10]. 이는 캐쉥 블럭을 세그먼트 단위로 관리하여 최근에 참조되지 않은 세그먼트 중에서 가장 높은 순위의 세그먼트를 교체 대상으로 선택하여 캐쉥 공간을 만든다. 세그먼트에 기반한 정책은 한번의 연산으로 많은 양의 블럭을 확보할 수 있는 반면에 스트리밍 서버로부터 많은 데이터를 받아들여야 하는 점에서 입출력 시스템과 스트리밍 서버에 부담을 갖게 한다.

본 논문에서는 참조 횟수, 참조량, 참조시간과 오브젝트 크기 등의 정량적인 캐쉬 인자들과 사용자 요구 주기를 적용하여 기존 캐쉬 교체 정책과 차등화된 개념의 가중치 윈도우에 기반한 캐쉬 교체 정책을 제시한다. 가중치 윈도우 기반의 캐쉬 교체 정책은 설정된 사용자 요구 주기에 따라 정량적인 인자들의 가중치를 차등 적용하고 최근 참조경향에 높은 가중치를 부여함으로써 변화하는 콘텐츠 선호경향에 신속히 적응하는 구조이다. 또한, 콘텐츠 단위의 캐쉬 교체 정책으로써 대상 오브젝트의 평균 참조량을 기존 캐쉬에서 정적 오브젝트와 동일하게 취급하고 평균 참조량 대 캐쉥량의 비를 캐쉥 서비스에 대한 오브젝트 위급도로써 반영하였다.

논문의 구성은 다음과 같다. 2장에서 가중치 윈도우 기반의 캐쉬 교체 정책 개념과 실행 모델을 기술하고, 3장에서 시뮬레이션을 통해 교체 정책의 성능 평가와 경향을 분석한다. 4장에서 결론을 맺는다.

2. 가중치 윈도우 기반의 캐쉬 교체 정책

다음은 본 논문에서 제시하는 가중치 윈도우 기반의 스트리밍 미디어 교체 정책 및 알고리즘에 대하여 기술한다.

2.1 캐쉬 오브젝트와 가중치 윈도우

캐쉬 오브젝트는 스트리밍 서버에서 전송받아 캐쉬내에 저장되고 대용량, 연속성의 특징을 갖는 동영상 미디어이며 사용자 참조 방식과 캐쉬 교체 정책에 따라 원본 미디어의 일부분이 캐쉬에 저장된다. 이는 캐쉬 오브젝트를 스트리밍 서버의 원본 미디어와 같이 완전한 콘텐츠로 관리하면 제한된 저장 공간과 대용량 미디어의 특징으로 인해 효율적인 미디어 캐쉥 시스템을 구성하는데 어려움이 있기 때문이다. 따라서 동영상 미디어 캐쉥 시스템은 원본 미디어의 일부분을 관리할 수 있는 구조를 지원해야 하고, 사용자의 콘텐츠 선호 경향과 활용 패턴에 따른 캐쉬 교체 정책을 지원해야 한다. 특히, 사용자의 콘텐츠 활용 패턴은 콘텐츠에 따라 처음부

표 1 캐싱 정책 인자

인자	설명	인자	설명
W	주 윈도우	$M_{transfer}(i)$	w(i)의 미디어 전송량
w(i)	i 번째 부 윈도우	M_{mean}	$M_{transfer}/A$
V	W의 가중치	$M_{mean}(i)$	$M_{transfer}(i)/a(i)$
v(i)	w(i)의 가중치	M_{prefix}	미디어 캐싱량
A	누적된 참조수	$M_{prefix}(i)$	w(i)의 미디어 캐싱량
a(i)	w(i)의 참조수	M_{suffix}	M_{size} - M_{prefix}
M_{size}	미디어 크기	$M_{replace}$	M_{prefix} - M_{mean}
$M_{transfer}$	누적된 전송량		

터 끝까지 활용하는 경우와 일부분까지만 활용하는 경우로 구분할 수 있으며, 이를 '평균 참조량'으로 계산하여 캐싱된 오브젝트 크기와 비교함으로써 캐쉬 교체 정책을 결정하는 중요한 인자로써 활용할 수 있다. 즉, 평균 참조량 이하를 캐싱했다면 그 차이량은 스트리밍 서버로부터 반입되어야 하는 중요성이 있고, 이상이면 차이만큼 제거할 수 있는 여유량으로 판단할 수 있다.

표 1은 본 논문에서 사용한 캐싱 정책 인자들이다. M_{size}는 원본 미디어 크기를 의미하고, M_{mean}은 참조 횟수로 전송량을 나눈 평균 전송량이다. 그리고 M_{prefix}는 캐쉬 내에 저장하고 있는 캐쉬 오브젝트의 크기이며, $M_{replace}$는 M_{prefix}와 M_{mean}의 차이량이다. M_{suffix}는 전체 미디어를 재생하는 사용자들을 위해서 스트리밍 서버로부터 전송받을 양을 의미한다. 캐쉬 오브젝트에 대한 이러한 분류를 한 이유는 제한된 저장 공간을 효율적으로 사용하기 위해서 교체 가능한 블록량과 캐쉬 오브젝트의 위급도를 나타내기 위해서이다. 우선적으로 교체 가능한 블럭인 $M_{replace}$는 그 값이 클수록 위급도가 낮고 0에 가까울수록 위급도가 높아진다. 또한 0 이하인 경우에는 콘텐츠 반입률과 더불어 사용자 QoS에 미치는 영향이 크며 수용 제한이 될 확률이 높다. 이러한 맥락에서 캐쉬 오브젝트의 M_{mean}은 캐쉬 교체 정책을 반영하는 관리 단위로서 의미를 가지며 웹 오브젝트와 같은 독립적인 캐싱 단위로 의미를 부여할 수 있다. 각 인자들은 런타임(run-time)시에 캐쉬 오브젝트의 한 캐쉬 블록이 입출력 완료된 후에 갱신된다.

캐쉬 오브젝트는 사용자 참조에 따라 서비스 중인 활성(active) 오브젝트와 사용자 참조가 없는 비활성화(inactive) 오브젝트로 구분된다. 먼저, 사용자 콘텐츠 참조의 전제 조건은 콘텐츠 시작부터 끝까지 순차적으로 발생하고 임의 위치로 옮겨갈 수 없으며 임의 위치에서 중단할 수 있음을 가정한다. 활성 오브젝트는 다수 사용자가 동시에 스트리밍 서비스를 받을 수 있으며, 선행 참조된 스트림에 의해서 콘텐츠 반입과 캐싱이 수행된다. 활성 오브젝트는 순차 참조 경향에 따라 오브젝트의 임의 블록을 삭제할 수 없기 때문에 캐쉬 블럭 요구의 교체 대상으로 적용되지 않는다. 따라서, 캐쉬는 활성 오브젝트의 원본 미디어를 캐싱할 공간이 필요하며, 이들의 총합이 캐쉬 저장 공간을 초과할 수 없다. 비활성화 오브젝트는 현재 사용자 참조가 없는 콘텐츠로 캐쉬 저장 공간 부족으로 발생하는 블럭 교체 요구의 대상이 되며, 이들은 캐쉬 교체 정책에 따라 가중치를 계산하여 우선 순위를 정해 교체 대상으로 선정된다.

'가중치 윈도우'는 현재 시점에서 일정 거리의 과거 시간까지를 시간 윈도우로 설정하여 그 시간 동안에 발생한 사용자 요구들 중에 현재에 가까운 요구 일수록 높은 가중치를 부여하여 최근의 사용자 참조 경향을 잘 반영할 수 있는 개념이다. 캐쉬 교체 정책에 있어 최근의 사용자 경향을 잘 반영해야하는 이유에 대해서 윈도우 크기를 결정하는 사용자 생활 패턴과 사용자 요구 패턴을 이해하는 것이 중요하다. 예를 들어, 직장 생활을 하는 사용자들의 생활 패턴은 직장 업무 시간과 가정에서 보내는 시간으로 분류하여 가정에서 휴식을 취할 때 동영상 콘텐츠를 시청한다면 그 시간대에 요구량이 많을 것이고 수면 시간이나 일하는 시간에는 상대적으로 적은 요구량을 보일 것이다. 즉, 일상생활의 하루는 사용자 요구 패턴이라 할 수 있으며 크게는 일주일, 한 달을 설정할 수 있다. 따라서 가중치 윈도우는 패턴 주기 내에서 요구 편중 경향을 잘 반영하기 위해 최근 참조량이 많은 콘텐츠를 빠르게 높은 우선순위에 도달시키며, 참조 빈도가 낮아지거나 없는 경우에 대해서는 시간이 지남에 따라 이전 참조 경향에 대한 가중치를 낮추고 점차적으로 우선순위를 하향시켜 재참조에 대한 위험성을 고려하였다. 또한 가중치 윈도우는 윈도우 범위를 일정 시간 간격으로 나누어 가중치 적용 단위를 설정하고 그 시간이 지남에 따라 윈도우를 이동시키는 슬라이딩 개념을 적용하였다.

그림 1은 가중치 윈도우의 구조를 보여주고 있다. W는 사용자 생활 패턴으로 전체 가중치 윈도우를 의미하는 주(major) 윈도우이다. $w(i)$는 W를 일정 시간 간격으로 설정된 가중치의 적용 단위가 되는 부(minor) 윈도우이다. 각 $w(i)$에 대한 가중치는 $f(i) = 1 - \frac{i}{n}$, $i \in \{0, 1, 2, ..., n-1\}$에 따라 결정된다. 이때, n은 주 윈도우 W 내의 부 윈도우 수이다. W에 대한 $w(i)$의 시간 간격 설정에 있어 다음을 고려해야 한다. $w(i)$가 짧을수록 사용 경향을 정확하게 기록하고 경향을 잘 반영할

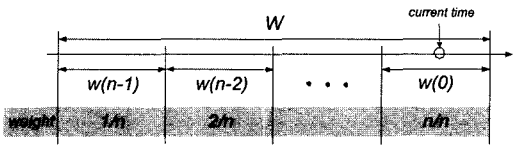


그림 1 가중치 윈도우의 구조

수 있지만 미디어 재생 시간 이하로 짧아지더라도 여러 부 윈도우에 경유하게 돼 효과적이라 할 수 없으며, 오히려 시스템 처리 부하를 가중시킬 수 있다. 반면, $w(i)$가 길수록 콘텐츠 참조 경향은 정확성이 낮아지게 된다. 따라서, $w(i)$의 크기는 대상 미디어의 재생 시간을 고려하여 설정해야 한다.

2.2 캐쉬 교체 정책

캐쉬 교체 정책은 스트리밍 미디어의 전송 패킷을 일정 크기의 균일 블럭 단위로 저장하여 캐쉬 오브젝트로 관리하고 동영상 미디어의 연속 재생을 고려해 캐쉬 오브젝트의 마지막 블럭을 교체 대상으로 한다. 또한, 제한된 시간 범위를 갖는 가중치 윈도우 내에서 콘텐츠 참조의 주기성과 편중성을 주요 인자로 하는 최신 참조 경향에 높은 가중치를 반영하는 효율적인 캐쉬 저장 공간의 활용 정책이다. 이를 가중치 윈도우 기반의 캐쉬 교체 정책(이하 'WIN'이라고 함)이라 한다.

$$v(i) = a(i) \cdot \frac{M_{mean}(i)}{M_{prefix}(i)}, \quad where i = 0,1,2,\ldots,n-1 \quad (1)$$

$$V = \tau \cdot \sum_{i=0}^{n-1} (1 - \frac{i}{n}) \cdot v(i) \quad (2)$$

$$\tau = \frac{M_{mean}}{M_{prefix}} \quad (3)$$

식 1, 식 2, 식 3은 W와 $w(i)$에 대해 가중치를 적용한 참조값 V와 $v(i)$를 계산하는 수식이다. 식 1과 식 2는 부 윈도우에서 가중치 값 $(1-\frac{i}{n})$와 참조에 대한 정량적인 값을 곱하여 계산된다. 가중치 값은 부 윈도우에서 발생한 참조수 $a(i)$와 전송 평균값 분에 캐싱량을 곱해서 계산된다. 즉, $v(i)$는 $w(i)$ 내에서 발생한 참조수에 대해 사용자의 콘텐츠 이용 경향에 대한 위급도를 계산하여 $w(i)$의 콘텐츠 활용도를 얻는다. 식 2는 $v(i)$ 총합에 $w(i)$ 가중치를 적용하고 캐쉬 교체 정책 경향을 나타내는 τ를 곱하여 계산된 것으로 W의 중요도를 나타내고 높은 수치일수록 콘텐츠 간에 높은 우선순위를 나타낸다. 식 3에서 τ는 현재 캐쉬 오브젝트 크기에 대한 평균 미디어 전송량 비이며, 평균 미디어 전송량은 참조 적중수와 적중에 대한 전송량 합에서 계산된 평균값이다. 따라서 τ는 캐쉬 오브젝트 크기에 대해 요구된 미디어의 위급도를 반영할 수 있다. 즉, 평균 전송량보다 적게 캐싱된 미디어는 그 이상 캐싱되었을 때보다 위도급을 높이 평가할 수 있으며, 캐싱량이 캐쉬 아웃될 미디어를 선택할 기준이 된다. τ값의 범위에 따라 캐쉬 교체 정책에 미치는 영향은 다음과 같다.

(a) 제한하지 않을 경우 : τ는 $\frac{M_{mean}}{M_{prefix}}$에 따라 $0 \le \tau$의 범위를 갖는다. 캐쉬 오브젝트는 참조 빈도와 참조량의 증가로 M_{mean}이 커지거나 블럭 교체 대상이 되어 M_{prefix}가 M_{mean}보다 상대적으로 작아질수록 높은 우선순위를 갖는다. 이는 M_{prefix}가 M_{mean} 이하인 캐쉬 오브젝트에서 사용자 참조가 발생했을 때, 캐싱량 부족으로 인한 사용자 QoS 저하와 스트리밍 서버로부터의 반입량 증가를 막기 위해서이다. 따라서, 캐쉬 오브젝트가 낮은 참조 빈도를 갖더라도 캐쉬 저장 공간에서 제거되기 보다는 M_{mean} 이하로 남아 있을 확률이 높아진다. 캐쉬 오브젝트는 가중치 윈도우 W내에 참조가 없는 경우에만 캐쉬에서 제거된다.

(b) 최대 값을 제한하는 경우 : 위 (a)의 경우 콘텐츠 크기가 작아질수록 τ의 비중이 커지기 때문에 참조율이 낮더라도 주 윈도우내에 참조가 없는 경우를 제외하면 캐쉬에서 제거되지 않을 확률이 높아진다. 따라서 콘텐츠 수는 증가하지만, 참조량 적중률은 감소하게 된다. 이러한 단점을 보완하기 위해 τ 범위를 $0 \le \tau \le 1$로 제한한다. 즉, M_{prefix}가 M_{mean}보다 작을 경우에 τ를 1로 제한함으로써 참조 빈도가 낮은 오브젝트가 블럭 교체를 통해 작아진 M_{prefix}로 인한 우선순위 상승을 제한할 수 있다. 이럴 경우 캐쉬 내에 오브젝트는 (a) 경우에 비해 캐쉬 오브젝트 수는 감소하지만, 캐쉬 내에 오브젝트는 커지는 효과가 있다.

(c) 무시하는 경우 : τ를 무시할 때, V는 $v(i)$의 총합에 부 윈도우의 가중치를 곱한 미디어 가중치이다. 따라서, 최근에 발생한 참조 빈도와 전송량은 캐쉬 교체 정책 경향을 나타내는 중요한 인자이며 시간대와 콘텐츠별로 편중하거나 콘텐츠 선호도 변화에 따라 캐쉬 오브젝트 교체에 효율적이다.

가중치 윈도우 기반의 캐싱 알고리즘은 그림 2와 그림 3과 같으며 대상 오브젝트에 대한 캐쉬 블럭 요구와 가중치 윈도우 갱신으로 구분된다. 캐쉬 블럭 요구는 먼

```
block ALLOCATE_BLOCK() {
  object vO; // a victim cache object
  IF (there is no free block) {
     select the least precedence vO among inactive objects;
     free the last block of vO;
     IF (the cached size of vO is not zero) {
       subtract the size of block from the cached size of vO;
       // v(0) = a(0) · (M_mean(0)/M_prefix(0))
       // V = τ · (v(0) + Σ_1^n (1-i/n) · v(i))
       calculate the weight of vO;
     } ELSE {
        remove a related information of vO from cache;
       }
     update a precedence list;                    }
  RETURN free block;
}
```

그림 2 캐쉬 블록 할당

```
VOID ADVANCE_WINDOW () {
  WHILE (for all objects in cache) {
      record a(i), M_mean(i), M_prefix(i)
            for current minor windows;
      update M_prefix and M_mean;
      advance current minor windows;
```

그림 3 가중치 윈도우 갱신

저 빈 블럭을 찾고, 빈 블럭이 없으면 희생 오브젝트를 찾아 블럭 교체를 수행한다. 희생 오브젝트는 사용자 참조가 없는 오브젝트 중에서 가장 낮은 우선순위에 있는 오브젝트를 선택하며 블럭 교체 이후 희생 오브젝트의 가중치와 우선순위를 계산하여 갱신한다. 또한, 블럭이 추가되는 오브젝트도 가중치와 우선순위를 갱신한다. 이러한 과정은 런타임 시에 현재 부 윈도우 $w(0)$에서 수행되며, 캐쉬 블럭 가감에 따른 가중치 변경은 식 1, 2, 3에서 현재 캐싱량(M_{prefix})에 영향을 받기 때문이다. 최종적으로 재 계산된 희생 오브젝트의 가중치 값에 따라 우선순위 리스트를 갱신한다. 그림 3에서 가중치 윈도우 갱신은 $w(0)$가 종료되는 시점에서 가장 오래된 부 윈도우 $w(n-1)$의 참조 경향을 삭제하고 새로운 $w(0)$를 구성한다. 나머지 $w(i)$들은 각각 한 단계 과거 윈도우로 이동한다. 또한, 부 윈도우 순서를 갱신하여 가중치 적용 비중을 차등 적용하게 한다. 캐쉬 오브젝트 가중치가 동일한 경우에 우선순위는 사용자 QoS에 미치는 영향을 고려하여 사용자의 평균 전송량 M_{mean}과 캐싱량 M_{prefix}에서 계산되는 교체 블럭 여유량 $M_{replace}$이 작을수록 우위 우선순위를 배정한다. $M_{replace}$값이 동일한 경우에는 스트리밍 서버에서의 반입량을 고려하여 평균 전송량이 많은 것에 따르며, M_{mean}도 같으면 가장 최근 참조된 오브젝트에 우위 우선순위를 배정한다.

그림 4는 시간축 이동에 따른 가중치 윈도우 개념과 캐쉬 오브젝트의 참조 경향에 대한 예를 보여준다. 가중치 윈도우는 부 윈도우 단위로 이동하고 W(n-2)와 W(n-1)을 거쳐 현재의 가중치 윈도우 W(n)에 있으며 부 윈도우 $w(3)$, $w(2)$, $w(1)$, $w(0)$로 구성된다. 캐쉬 오브젝트는 $w(n)$을 기준으로 활성 오브젝트 m_4와 비활성 오브젝트 m_1, m_2, m_3으로 분류된다. 각 오브젝트의 참조 패턴을 살펴보면, 오브젝트 m_1은 윈도우가 W(n-2)에서 W(n)으로 이동하는 동안 4번의 참조가 발생하였고, m_2는 W(n-2)에서 1번의 참조가 발생하였다. m_3은 m_2의 참조가 발생된 다음에 참조가 있었고, m_4는 3개의 윈도우에 걸쳐서 5번의 참조가 발생하였다. m_2와 m_3은 W(n)에서 참조가 없기 때문에 가중치 값은 0이 되고, 가중치가 동일하기 때문에 캐싱된 양과 평균 전송량, 가장 최근 참조 시간을 비교하여 우선순위를 결정한다. 따라서 m_2와 m_3의 우선순위는 W(n)내에서 참조가 없으므로 캐싱된 양과 평균 전송량이 0이며 m_3이 최근 참조 시간이 앞서므로 m_2보다 높은 우선순위에 배정된다. 따라서 비활성 오브젝트의 우선순위는 높은 순위에 따라 m_1, m_3, m_2이다. 캐쉬 저장 공간이 부족할 때, W(n)의 $w(0)$에서 m_4의 블럭 요구는 가장 낮은 우선순위인 m_2의 마지막 블럭에서 블럭 교체가 수행된다.

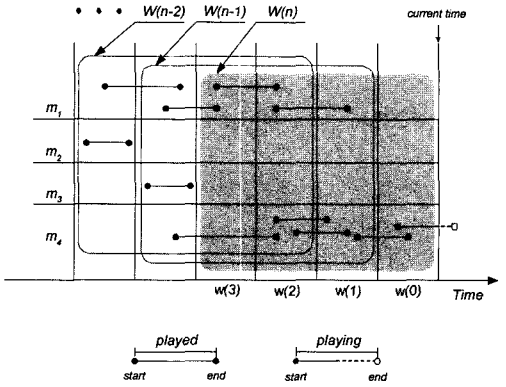


그림 4 슬라이딩 윈도우와 분석 모델

3. 성능분석

논문에서 제시한 가중치 윈도우 기반의 WIN 캐쉬 교체 정책을 참조시간에 대한 정책인 LRU, 참조량에 대한 정책인 LFU, 세그먼트 기반의 LRU 정책인 SEG와 비교분석하였다. 성능분석은 각 정책에 대해 동일한 시뮬레이션 환경 하에서 측정하였으며, 7일간의 시간 범위와 1,000개의 콘텐츠를 대상으로 Zipf [11] 요구 분포를 구성하여 실험하였고 콘텐츠 크기, τ, 콘텐츠 요구 분포 이동을 인자로써 적용하였다. 실험구성은 τ, 콘텐츠 크기 종류, Zipf 요구 분포와 콘텐츠 요구 분포 이동을 적용한 실험과 콘텐츠의 우선순위 변화 과정을 보였고, WIN의 τ 설정에 따른 성능 변화를 사용자 QoS를 고려한 SEG와 비교분석하였다. 세부적인 실험환경은 표 2와 같다. 최대 허용 스트림수는 캐쉬 시스템에서 허용 가능한 스트림 수이며, 평균 클라이언트 자료 요구율은 콘텐츠에 대한 미디어 소비율을 나타낸다. 캐쉬 용량은 최대 콘텐츠 크기와 최대 허용 스트림 수를 통해 최소 캐쉬 저장 공간(50 KBlocks)을 계산하고 10 KBlocks 단위로 증가시켜 구성하였다. 요구 분포의 프라임 시간은 하루를 기준으로 사용자 요구가 편중되는 시간이다. 각 실험에서는 참조 적중률, 참조량 적중률, 지연된 참조률과 캐쉬 반입량을 측정하였다. 참조 적중률은 전체 사용자 요구에 대한 캐쉬 콘텐츠에 적중률을 의미하고, 참조량 적중률은 사용자 요구 시점에서 캐쉬에 존재하

표 2 캐쉬 교체 정책의 실험 환경

최대 허용 스트림 수	50 (스트림)
평균 클라이언트 자료 요구율	128 (KByte/sec)
클라이언트 초기 요구량	20 (sec)
콘텐츠 크기	75, 150, 225, 300, 375, 450(MBytes)
콘텐츠 수	1,000 (편)
캐쉬 용량	50, 60, 70, 80, 90, 100 (KBlocks)
캐쉬 블록 크기	512 (KBytes)
W	1 (일), 24 (시간)
w	1 (시간)
실험시간	7 (일)
요구 분포의 프라임 시간	22 (시)
요구 분포	Zipf (0.0), Zipf (0.5), Zipf (1.0)

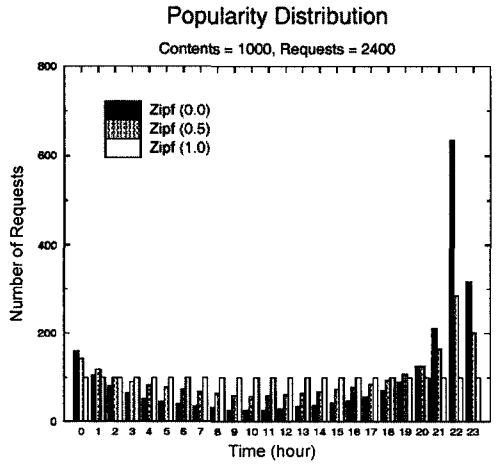


그림 5 시간대에 따른 사용자 요구 분포

는 콘텐츠량에 대한 원본 콘텐츠 크기에 대한 비율을 의미한다. 지연된 참조률은 사용자 요구가 적중되지 않거나, 캐쉬량이 클라이언트 초기 요구량에 미달될 때에 전체 요구에 대한 비율이고, 캐쉬 반입량은 서버로부터 선 반입된 양이다.

그림 5는 24시간동안 1,000편의 콘텐츠에 대해 Zipf (θ)의 θ 인자를 0.0, 0.5, 1.0으로 변화를 주고 22시를 사용자들이 영화를 시청하는 프라임 시간으로 설정하여 얻은 Zipf 요구 분포이다. θ 인자 값이 1.0이면 시간대별로 균일하게 배정되는 평균 분포를 이루고, 0.0이면 프라임 시간대를 기준으로 편중분포를 이룬다. 본 실험에서는 프라임 시간대를 기준으로 우측과 좌측 시간에 대해 교차로 배정함으로써 사용자 요구가 점진적으로 증가하고 감소하도록 요구 분포를 구성하였다. 그림 6(a)는 그림 5에서 나타난 시간대별 분포를 168시간(7일)에 걸쳐 1,000편의 콘텐츠에 대한 선호도 분포를 보여준다. 시뮬레이션 환경에서는 사용자들의 요구를 고르게 발생시키기 위해서 난수 발생을 통해 Zipf 값에 근사값을 갖도록 하였다. 그림 6(b)는 그림 6(a)의 분포에 대해 24

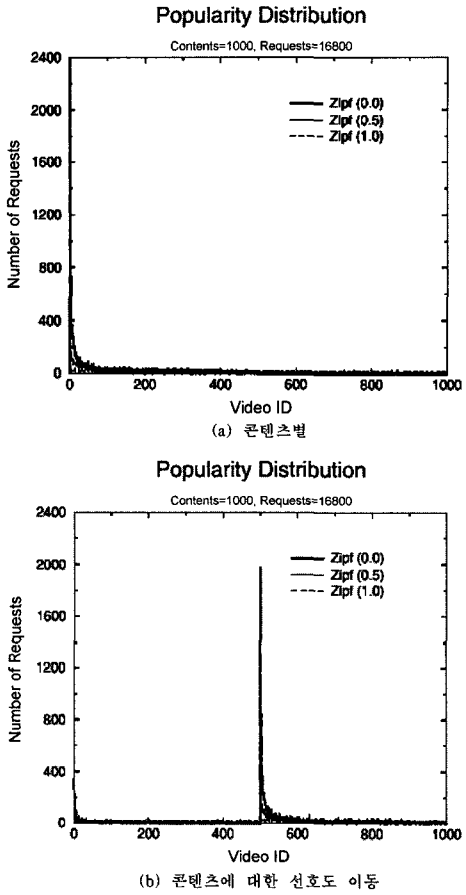


그림 6 사용자 요구 분포

시간이 경과한 후 콘텐츠에 대한 선호도를 이동시켜 168시까지 발생한 분포로 24시를 기준으로 선호도가 0에서 500번으로 변경되는 것을 보여준다. 이러한 사용자 선호도 변화는 사용자 요구를 실생활에서 발생하는 것과 유사한 경향을 주고 캐쉬 교체 정책에 어떻게 반영되는지를 측정하기 위한 것이다. WIN 캐쉬 교체 정책은 τ 인자 적용에 따라 τ 무시, $0 \leq \tau \leq 1$과 $0 \leq \tau$일 때를 구분하여 실험하였다. τ 무시일 때는 주 윈도우에서 부 윈도우로 분류된 가중치 적용에 따라 참조 빈도와 참조량 인자를 차등하게 적용함으로써 최근 콘텐츠 참조 경향에 비중을 둔 실험이다. τ 인자를 적용한 $0 \leq \tau \leq 1$과 $0 \leq \tau$의 실험은 τ 인자를 현재 캐쉬량과 평균 전송량의 비율로 적용하여 현재 캐쉬량에 비중을 두어 캐쉬 오브젝트의 위급도를 나타내고 재참조에 대한 사용자 QoS를 향상시키기 위한 것이다. $0 \leq \tau \leq 1$일 때는 $0 \leq \tau$에서 콘텐츠 사용 경향을 제한적으로 적용한 실험이다. 실험 결과는 콘텐츠 선호도에 따른 미디어 캐쉬경향과 τ에 따른 캐쉬효과로 제시한다.

3.1 τ = 1일 때, 콘텐츠 선호도 변화가 없는 경우

τ = 1일 때, 콘텐츠 선호도 변화가 없는 경우의 실험은 콘텐츠의 캐쉬량 대 전송 평균량 비율을 가중치 윈도우의 부 윈도우에만 적용하고, 사용자가 선호하는 콘텐츠 참조 분포의 이동을 배제한 실험이다. 캐쉬 교체 정책 실험은 Zipf (0.0)에 대하여 캐쉬 크기 별로 참조 적중률, 참조량 적중률, 지연된 참조률과 반입량을 측정하였다. 그림 7은 Zipf (0.0)에 대한 실험 결과로 사용자 요구를 시간대와 콘텐츠 별로 편중 분포를 발생시켜 얻은 결과이다. 그림 7(a)에서 참조 적중률은 캐쉬 용량이 증가함에 따라 캐쉬 저장 공간에 존재하는 콘텐츠가 증가하므로 각 정책들의 성능은 향상되고 있다. LRU는 최근에 발생한 사건의 시간에 기반하여 유사한 경향을 보인다. 반면에 WIN과 LFU는 전체 혹은 부분적으로 누적 방식에 따르고 있어 자주 발생하는 콘텐츠 위주로 캐쉬를 구성하여 상대적으로 좋은 성능이 나타나고 있다. LFU는 발생한 요구에 대한 전체 누적 방식으로 시간대 분포가 바뀌더라도 콘텐츠의 참조 패턴이 반복되므로 참조률이 높고, WIN은 적용되는 가중치가 슬라이딩 윈도우의 진행과 범위에 따라 우선순위를 결정하므로 주기적으로 반복되는 콘텐츠 요구 상황에 LFU에 상대적으로 낮은 참조 적중률의 정확성을 가질 수 있다. 즉, 주기적인 사용자 요구와 편중 상황에서 WIN 캐쉬 정책과 LFU의 성능 차이는 사용자 콘텐츠 우선순위와 각 정책의 우선순위가 근접할수록 콘텐츠 이동이 적게 발생하기 때문에 나타나게 된다. 그림 7(b)는 캐쉬 내에 존재하는 콘텐츠 크기와 원본 콘텐츠 크기의 비율로 전체 요구에 대한 참조량 적중률을 나타낸 것이다. 참조량

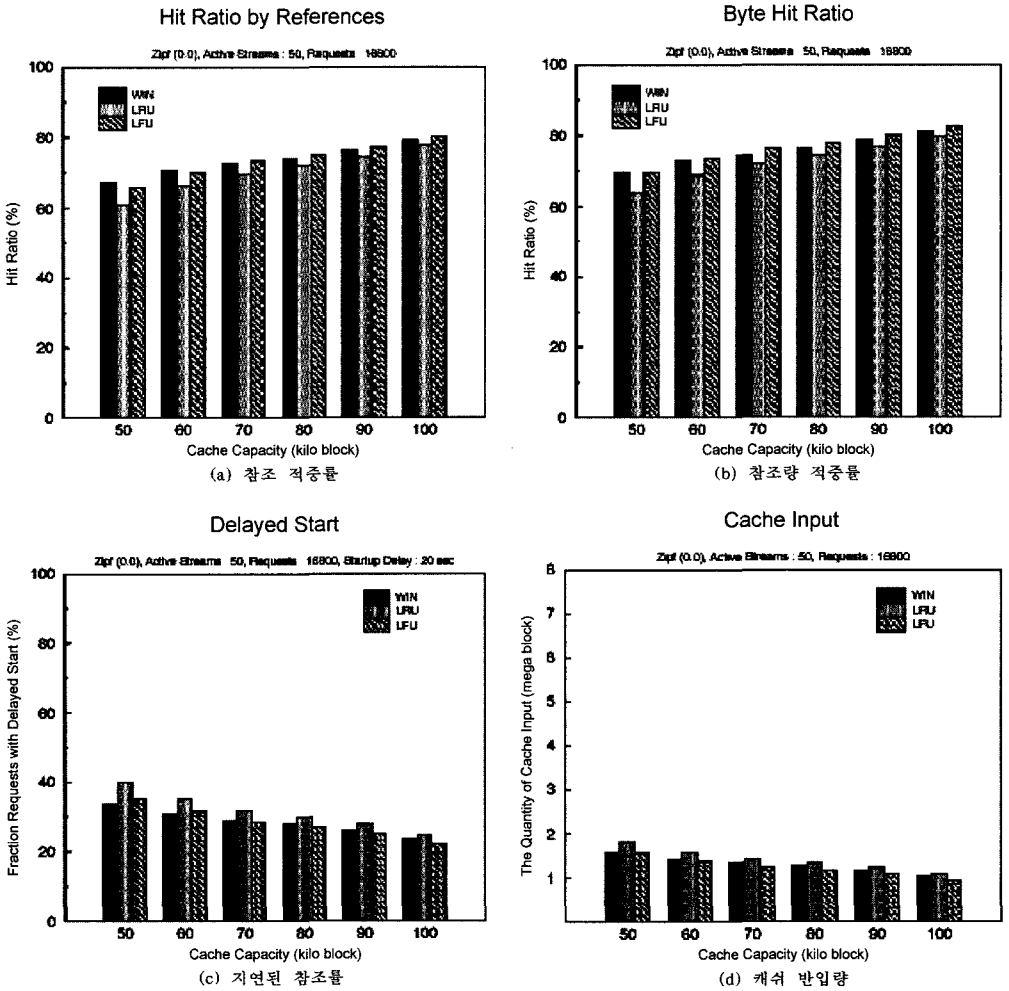


그림 7 성능비교 : τ = 1, Zipf (0.0), 콘텐츠 선호도 변화

적중률이 높게 나오기 위해서는 참조되는 콘텐츠량이 원본 콘텐츠에 비례적으로 많이 존재해야하고, 우선순위가 높은 콘텐츠일수록 많은 양을 가지고 있어야 한다. 각 정책들 모두는 그러한 경향을 반영하고 있지만, LRU와 같이 최근 참조 시간에 의존한 정책들은 가장 최근에 발생한 사건에 높은 우선순위를 부여함으로써 콘텐츠 간에 빈번한 우선순위 변화로 인해 사용자 참조가 높은 콘텐츠이더라도 시간대 편중성과 참조 간격이 길면 교체 대상으로 선정하기 때문에 상대적으로 낮은 성능을 보였다. 그림 7(c)에서 사용자 QoS량으로 설정한 20초 분량의 콘텐츠 크기의 존재율에 대한 실험에서 낮게 나타나는 이유가 이를 반증한다. 그림 7(c)는 각 정책들이 우선순위에 따라 차등하게 캐싱된 콘텐츠 크기를 유지하는지 보여준다. 그림 7(d)는 캐쉬 정책에 따라 스트리밍 서버에서 반입되는 량이다. 반입량은 캐쉬 서버가 의도하는 네트워크 대역폭의 균일화에 대한 성능을 반영하는 중요한 요소로 캐쉬에 반입되는 양이 작을수록 좋은 성능으로 평가되므로 참조 적중률과 참조량 적중률의 성능에 반비례한다.

3.2 τ = 1일 때, 콘텐츠 선호도 변화가 있는 경우

τ = 1일 때, 콘텐츠 선호도 변화가 있는 경우의 실험은 콘텐츠의 캐싱량 대 전송 평균량 비율을 가중치 윈도우의 부 윈도우에만 적용하고, 사용자가 선호하는 콘텐츠 참조 분포의 이동을 전제로 한 실험이다. 콘텐츠 선호도는 그림 5에서 시간대의 사용자 요구 분포에 그림 6(a)의 콘텐츠 참조 분포를 그림 6(b)와 같이 콘텐츠 선호도에 변화를 준 것이다. 일반적으로 콘텐츠의 사용자 선호는 여러 가지 요인에 따라 낮은 우선순위에서 높은 우선순위로의 이동과 높은 우선순위에서 낮은 우선순위로의 이동을 가정해야 한다. 예를 들어, 콘텐츠에 대한 첫 요구 이후 집중적으로 요구가 발생하면 콘텐츠 등록과 함께 빠른 시간 안에 높은 우선순위에 도달되어야 하고 최대한 많은 양의 콘텐츠를 캐쉬에 유지하는 것이 유리하다. 반면, 한동안 선호 경향을 높았다가 점차적으로 선호 경향이 낮아지면 콘텐츠의 우선순위도 점차적으로 낮아져야하고, 최종적으로 캐쉬에서 제거되어야만 캐쉬 저장 공간을 효율적으로 사용할 수 있다. 일정 시간이 지남에 따라 콘텐츠에 대한 사용자 선호 경향이 변하는 환경에서의 실험은 선호 콘텐츠 변화에 각각의 정책이 캐쉬 내에 콘텐츠 구성에 빠르게 적응하느냐에 따라 교체 정책의 성능이 결정된다. WIN 정책은 가중치 윈도우 범위 내에서 우선순위를 정하여 변화하는 환경에 빠르게 적응하도록 설계되어 가장 최근에 발생한 사용자 참조에 높은 가중치를 부여하고 신규 콘텐츠 원활한 캐싱과 참조가 낮은 콘텐츠의 삭제함으로써 LFU와 LRU보다 높은 성능 차이를 보였다. LFU는 기존의 누적 참조된 콘텐츠와 새롭게 사용자 우선순위 경향이 높아지는 콘텐츠가 경합하여 캐쉬 내에 서로 혼재됨으로써 캐쉬의 효율을 떨어뜨리는 원인이 된다. LRU는 최근 사건에 기반하기 때문에 사용자 선호 경향 변화에 빠르게 적응할 수 있으나, 반대 경우의 결과에서 가장 최근에 발생한 참조 시점보다 누적된 가중치 반영이 중요하다는 원칙에 따라 사용자 참조 경향이 바뀌는 시점을 기준으로 이전 결과는 LFU>WIN>LRU, 이후 결과는 WIN>LRU>LFU으로 분석된다. 즉, LRU는 우선순위가 변하는 시점에서는 좋은 성능을 나타낼 수 있으나 사용자 참조 경향이 누적될수록 효율성이 낮은 것을 알 수 있다.

Zipf 요구 분포에서 선호도 이동에 따른 실험은 24시간의 시간대 요구 경향을 168시간 동안 적용하고, 처음 24시가 되는 시점에서 콘텐츠에 대한 선호도 분포 경향을 그림 6(b)와 같이 반영하고 이후 144시간 동안의 경향을 분석하였다. 그림 8는 편중 분포에 따른 실험으로서 전체적인 성능은 그림 7의 Zipf 요구 분포에 따른 실험에 비해서 다소 성능 차이를 보인다. 성능 차이 요인은 24시에 발생하는 콘텐츠 요구 분포의 이동으로 캐싱된 콘텐츠가 교체되는 과정에서 참조 적중률과 참조량 적중률 등이 낮아지기 때문이다. 반면, 각 정책들 간의 비교에서 WIN 캐쉬 교체 정책이 우선순위 변화를 다른 정책에 비해 상대적으로 잘 반영하고 있음을 확인할 수 있다. LFU 정책은 콘텐츠 요구 분포 이동에서 24시 이전에 높은 우선순위에 있는 콘텐츠들이 낮은 우선순위로 이동하고 캐쉬에서 제거되는 기간이 길어 낮은 성능을 보이고 있다. LRU 정책은 현재 시점에서 가장 참조되지 않은 콘텐츠를 선택하므로 콘텐츠 요구 분포의 이동에 대해 영향 관계가 약하다. 그림 8(a), 8(b), 8(c), 8(d)들은 편중 분포에서 콘텐츠 요구 분포 이동에 대한 참조 적중률, 참조량 적중률, 지연된 참조률, 캐쉬 반입량에 대한 결과를 보여 주고 있다.

그림 9는 사용자 선호 경향 변화 환경에서 사용자 참조 경향의 변화에 따라 각각의 정책들이 우선순위 반영 상태를 보여주는 것으로 가장 높은 우선순위에 있던 콘텐츠가 더 이상의 사용자 참조가 없는 경우(high-low)와 신규 콘텐츠의 사용자 참조가 집중될 경우(low-high)를 정책별로 보여준다. WIN 정책에서는 high-low 상황에서 시간이 지남에 따라 콘텐츠 가중치가 서서히 감소됨에 따라 그 우선순위도 낮아지고 low-high에서는 최근 참조에 높은 비중을 두기 때문에 빠르게 높은 우선순위를 반영함을 알 수 있다. 반면 LRU 정책은 기존에도 우선순위 변화가 빈번하게 발생하고 high-low에서 시간이 지남에 따라 빠르게 NC 상태에 도달하고 low-high는 정책 특성에 따라 최초 참조 시점부터 최

상위 우선순위에 위치함을 알 수 있다. LFU는 low-high로 누적된 참조에 따라 우선순위가 증가하지만 high-low 상황에서 우선순위 변화가 빠르게 반영되지 않아 낮은 성능의 원인임을 알 수 있다.

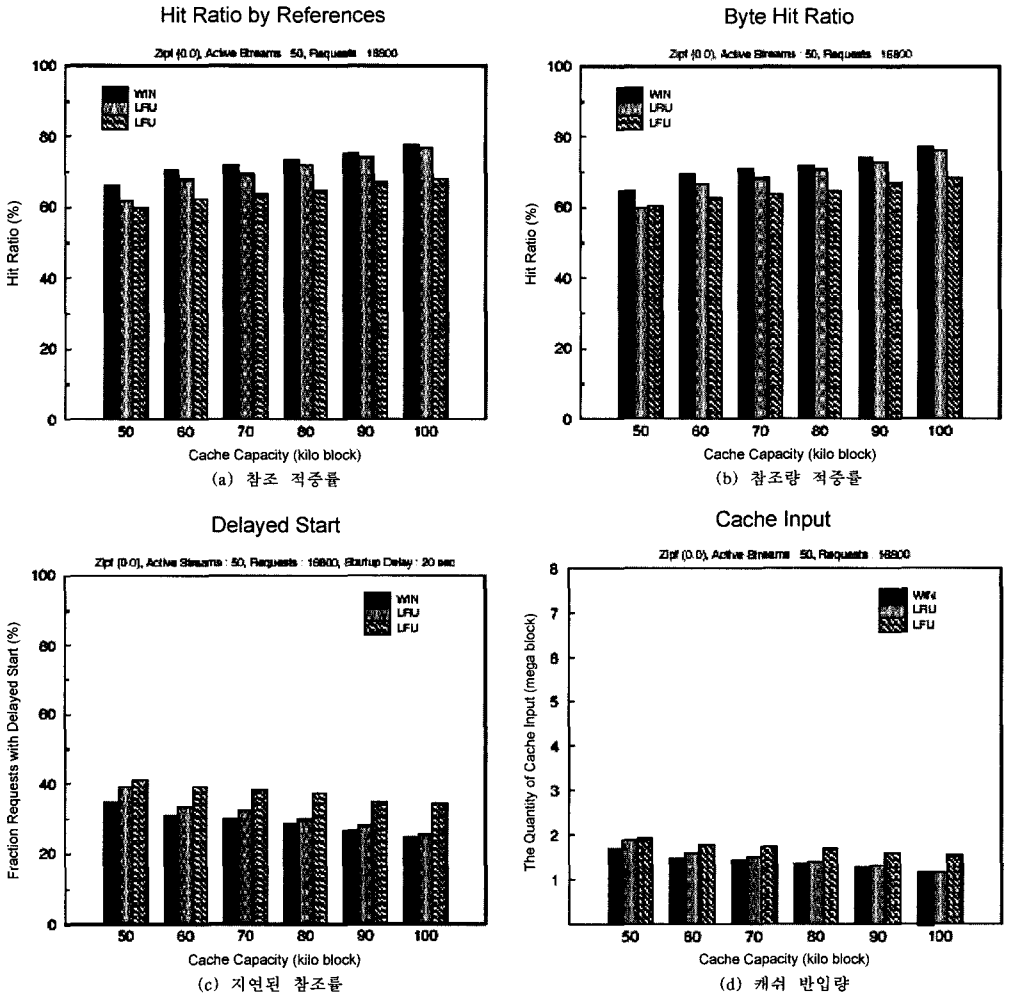


그림 8 성능 비교 : τ = 1, Zipf (0.0)

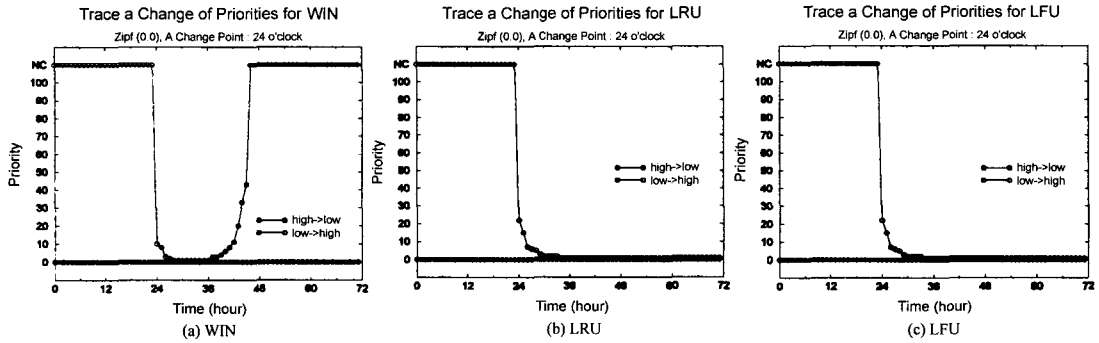


그림 9 우선순위 변화 : τ = 1, Zipf (0.0), 콘텐츠 선호도 변화

3.3 $0 \leq \tau \leq 1$과 $0 \leq \tau$일 때, 콘텐츠 선호도 변화가 있는 경우

$0 \leq \tau \leq 1$과 $0 \leq \tau$일 때, 콘텐츠 선호도 변화가 있는 경우의 실험은 가중치 윈도우의 주/부 윈도우에 콘텐츠의 캐쉬량 대 전송 평균량 비율을 적용하고 콘텐츠 선호도 변화를 전제로 한 실험이다. $0 \leq \tau \leq 1$일 때, WIN 캐쉬 교체 정책은 낮은 참조 빈도를 갖는 콘텐츠가 블럭 교체에 따라 캐쉬량이 작아지더라도 M_{prefix}와 M_{mean}의 비율에 따라 위급도를 높여 캐쉬 아웃을 방지하는 효과를 보인다. 또한, M_{prefix}와 M_{mean}의 비율을 최대 값을 1로 제한하여 거의 참조되지 않는 콘텐츠에 대해서 제한을 두었다. 실험은 Zipf (0.0)에서 콘텐츠 크기 종류에 따라 주기적인 요구 분포, 요구 분포 이동 상황을 적용하였다. 그림 10에서 참조 적중률은 $0 \leq \tau \leq 1$의 캐쉬 아웃 방지 효과에 따라 콘텐츠 수가 증가하여 $\tau = 1$인 WIN, LRU, LFU보다 높은 적중률이 나타났다. 지연된 참조율은 콘텐츠 증가와 주 윈도우의 범위에서 벗어나 참조되지 않는 콘텐츠의 캐쉬 아웃으로 참조 적중율에 비례적으로 지연율이 낮아졌다. 반면, 참조량 적중률은 $\tau = 1$인 WIN에 비해 콘텐츠들의 크기가 작아서 성능은 낮지만 그 외의 정책에 비해 향상된 결과를 보였다. 각 정책들의 캐쉬 반입량은 참조량 적중률 실험과 유사한 경향을 보이지만, 콘텐츠 캐쉬량, 콘텐츠 수와 참조 적중률에 영향을 받는다. 즉, $\tau = 1$인 WIN과 LRU, LFU는 캐쉬 미스 일 때 우선순위가 낮은 콘텐츠이거나 신규 반입 콘텐츠에서 반입량이 많고, 나머지 정책들은 캐쉬 히트일 때 서비스 거부가 적어 반입량이 많다. WIN 캐쉬 교체 정책의 참조 적중률과 참조량 적중률이 동일한 향상 경향을 보이지 않는 것은 $\tau = 1$에 비해 콘텐츠 크기가 작아지고 콘텐츠 수가 증

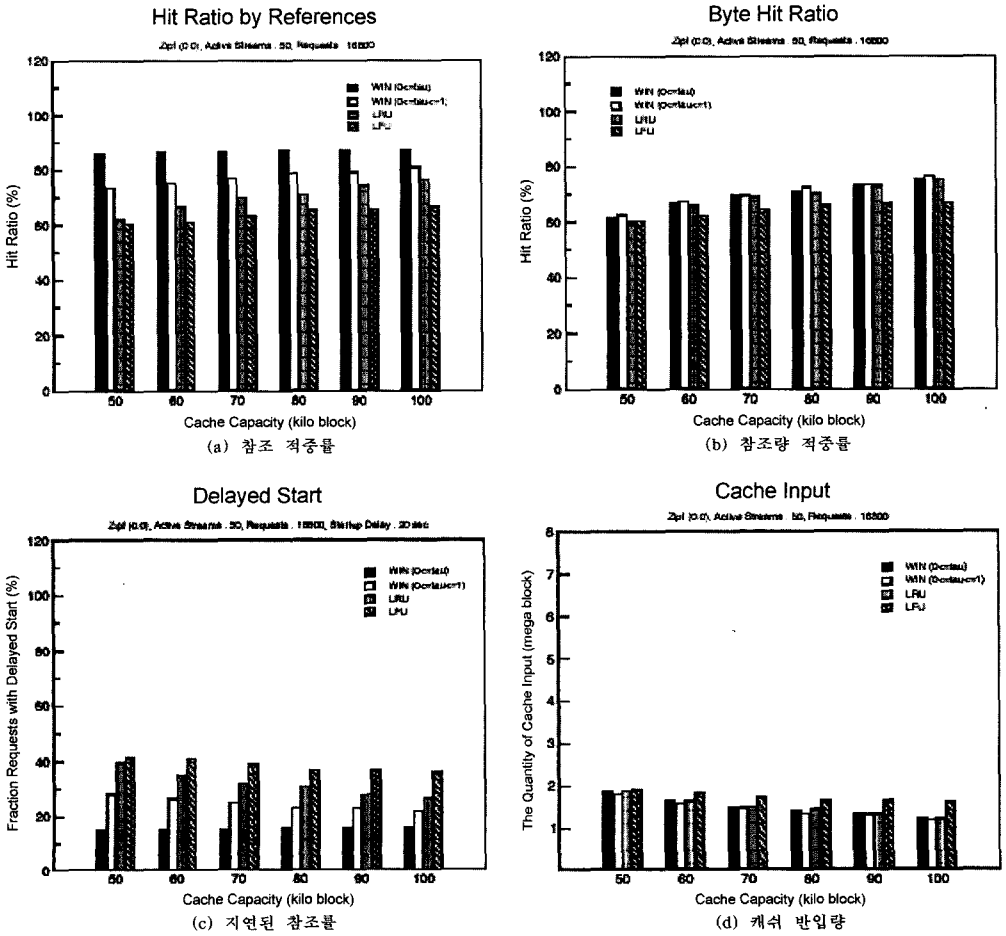


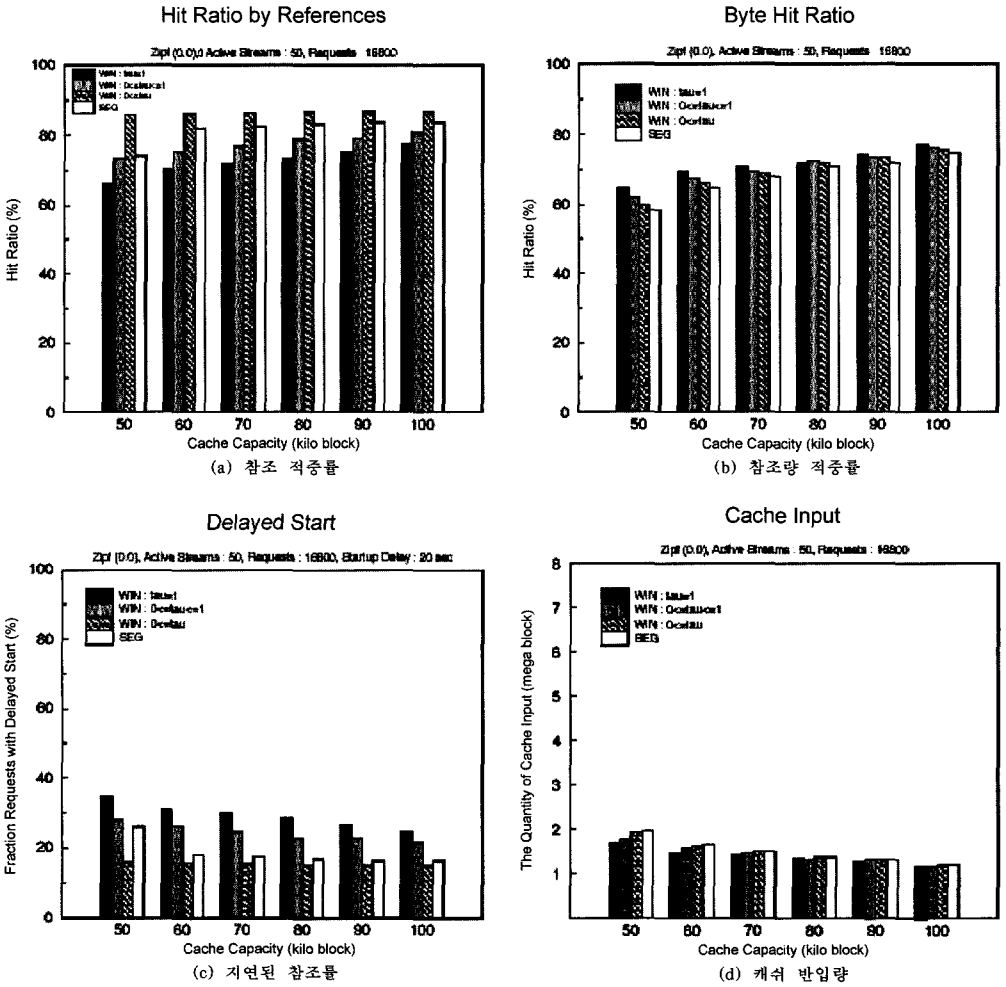
그림 10 성능비교 : Zipf (0.0), 콘텐츠 선호도 변화

가하기 때문이다.

$0 \leq \tau$ 일 때, $0 \leq \tau$인 WIN은 τ를 $\frac{M_{mean}}{M_{prefix}}$에 따라 결정하고 사용자 콘텐츠 활용 패턴을 강조한 것으로 캐쉬에서의 콘텐츠 제거가 완만하게 적용될 수 있다. 그림 10에서 $0 \leq \tau$인 WIN은 LRU와 LFU에 비해 약 10~20%이상의 참조 적중률과 지연된 참조률의 향상을 보이고 있으나, 참조량 적중률과 캐쉬 반입량은 유사한 성능 경향을 보이고 있다. WIN 참조 적중률이 캐쉬 저장 공간 크기가 커질수록 완만하게 상승하는 이유는 τ가 콘텐츠 우선순위에 높은 비중을 차지함을 의미하고 원본 미디어 크기에 비해 캐쉬량이 다양한 비율로 존재하기 때문이며 저장 공간 증가에 따라 콘텐츠 수를 증가시키지 않았기 때문이다.

3.4 WIN과 SEG의 비교 실험

지금까지의 실험에서 각 캐쉬 교체 정책들은 클라이언트의 초기 재생 지연 방지를 위한 사용자 QoS에 대해서 캐쉬 저장 공간에 콘텐츠의 캐쉬되어진 양에 의존하였다. 본 실험에서 추가된 SEG 캐쉬 정책[10]은 콘텐츠 관리를 위한 세그먼트 구조와 사용자 초기 시작 지연 방지를 위한 QoS 정책을 제안하였다. SEG의 콘텐츠는 캐쉬 블록 단위인 세그먼트로 구성되어 있으며 세그먼트 그룹 관리를 통해 i번째 그룹에 세스먼트를 배정하는 구조이다. 이는 콘텐츠 후미의 세그먼트 그룹을 교체 대상으로 하여 제한된 캐쉬 저장 공간을 효율적으로 확보하기 위해서이다. 사용자 초기 시작 지연 방지를 위한 QoS 정책은 콘텐츠의 세그먼트 그룹을 초기 시작 지연을 위한 그룹과 나머지 그룹으로 분류하고, 캐쉬 저장 공간도 초기 시작 지연을 위한 1차 캐쉬와 나머지 공간인 2차 캐쉬로 분류한다. SEG 교체정책은 현재시



(a) 참조 적중률 (b) 참조량 적중률

(c) 지연된 참조률 (d) 캐쉬 반입량

그림 11 WIN과 SEG 성능비교 : Zipf (0.0), 콘텐츠 선호도 변화

간(T)과 마지막 참조된 시간(T′)의 차이값(T-T′)에 세그먼트의 거리(i)를 곱한 식의 역수 ($\frac{1}{((T-T')\times i)}$)를 계산하여 작은 값을 갖는 희생 세그먼트 그룹을 선택한다. 그리고 초기 시작 지연을 위한 그룹은 나머지 세그먼트 그룹에 우선한다. SEG 교체정책은 가장 최근에 참조되지 않은 세그먼트 그룹을 교체 대상으로 선정하는 측면에서 LRU를 변형한 정책이라 할 수 있다.

SEG 캐쉬 정책은 2차 캐쉬 있는 세그먼트 그룹에 비해 1차 캐쉬의 세그먼트 그룹은 오랫동안 참조되지 않아야만 캐쉬 아웃될 수 있기 때문에 1차 캐쉬 내에 있는 콘텐츠 수에 많은 영향을 받는다. 본 실험에서는 SEG 교체 정책의 1차 캐쉬 비율을 디폴트값인 10%로 설정하였다. 참조 적중률과 지연된 참조률에서 $0 \leq \tau$ 일 때의 WIN 정책이 SEG에 비해 높은 성능의 원인은 WIN 정책에서 참조 빈도수가 낮은 콘텐츠에 대한 캐싱량이 작아질수록 τ값이 역비례로 증가하고 전송 지연 방지량 이상의 캐싱량을 유지함으로써 참조 상태에 있지 않은 콘텐츠들이 유사한 크기일 가능성이 높아 그 수가 증가하고, SEG 정책은 1차 캐쉬에 들어 있는 세그먼트의 제한량(전송 지연 방지량)에 근사하게 캐싱된 콘텐츠의 2차 캐쉬에 존재하는 세그먼트들에 우선하기 때문에 콘텐츠 캐쉬 아웃이 적절하게 이루어지지 않고, 1, 2차 캐쉬의 LRU 정책에 따른 효과에 영향 받기 때문이다. 그림 11에서 SEG 정책은 WIN 정책에서 τ에 따른 비교에서 $0 \leq \tau$와 $0 \leq \tau \leq 1$의 중간 성능을 보였으며, 참조량 적중률과 캐쉬 반입량은 유사한 성능을 보였다.

4. 결 론

본 논문에서는 스트리밍 미디어 캐싱 서버의 효율적인 캐싱 구조를 위하여 참조 횟수, 참조량, 참조 시간 등의 정량적인 인자들과 사용자 요구 주기를 적용하여 최근 참조 경향에 높은 가중치를 부여함으로써 변화하는 콘텐츠 선호 경향에 빠르게 적응하는 가중치 기반의 캐쉬 교체 정책을 제시하였다. 교체 정책의 시뮬레이션 실험은 기존의 LRU, LFU와 SEG 캐쉬 정책과 비교분석하여 향상된 결과를 나타냈다.

가중치 기반의 캐쉬 교체 정책은 콘텐츠 참조 편중성, 시간대 편중성, 콘텐츠 선호도 이동 등의 캐싱 환경에서 교체 정책의 τ 인자에 따라 네트워크 트래픽 감소나 사용자 QoS 향상을 목적으로 하는 효율적인 캐싱 시스템 구축에 적용할 수 있다. 적용 환경에 따른 τ 인자 설정은 다음과 같다. 첫째, 사용자 요구의 시간대 편중성과 콘텐츠 편중성이 높고 콘텐츠 선호도 변화가 잦은 경우, 캐싱 시스템은 τ 인자를 무시하여 편중성이 높은 콘텐츠의 캐싱 비율을 높이고 콘텐츠 참조 빈도와 참조량을 향상시키고 네트워크 트래픽을 감소시키는 효과가 있다. 또한, 잦은 콘텐츠 선호도 변화에 빠르게 적응하는 장점이 있다. 둘째, 첫째 경우와 반대로 콘텐츠에 대한 편중성이 낮은 경우, 캐싱 시스템은 τ 인자를 $0 \leq \tau$으로 설정하여 클라이언트의 시작 지연과 지터에 대한 사용자 QoS를 향상시키는 것이 효율적이다. 즉, 캐싱 시스템은 콘텐츠 편중성이 낮으면 캐쉬 자체의 효율성이 떨어지기 때문에 캐쉬 내에 되도록 많은 콘텐츠를 유지하는 것이 유리하다. 셋째, 선행한 두 경우의 중간 경향에 해당되며, 초기 일반적인 상황이나 캐싱 환경을 모르는 경우에 적용할 수 있다. 넷째, 셋째와 동일한 상황에서 ISP 데이타 센터에 1차 캐쉬 서버가 있고 POP 내에 2차 캐쉬 서버가 있는 경우이다. 보통 1차 캐쉬 서버는 기간 망에 연결되어 있어 네트워크 트래픽 감소를 목적으로 τ를 무시하고, 2차 캐쉬 서버는 사용자 QoS 향상을 위해 $0 \leq \tau$으로 설정하여 상호 보완적으로 구성할 수 있다. 또한, 그 역의 설정도 상호 보완관계를 구성할 수 있지만, 동일한 τ 인자 설정은 피해야 한다.

본 연구의 향후 과제는 사용자 패턴과 미디어 특성을 고려한 주 윈도우와 부 윈도우 크기를 결정하는 문제와 캐싱 시스템 구축에 적용하여 실제 사용자 요구 경향에 대한 실험을 통해 효율성을 검증하는 것이다.

참 고 문 헌

[1] S. Acharya, "Techniques for Improving Multimedia Communication over Wide Area Networks," *Ph.D. Thesis*, Cornell University, 1999.

[2] M. Hofmann, T. S. Eugene Ng, K. Guo, S. Paul, and H. Zhang, "Caching Techniques for Streaming Multimedia over the Internet," *Technical Report*, Bell Laboratories, April. 1999.

[3] K. Andrews, F. Kappe, H. Maurer, and K. Schmaranz, "On Second Generation Hypermedia Systems," *Proceedings of ED-MEDIA,World Conference on Educational Multimedia and Hypermedia*, June 1995, pp.127-136.

[4] J. Pitkow, and M. Recker, "A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns", 2nd Int. WWW Conf, Chicago, Oct 1994, pp.1039-1045.

[5] M. Abrams, C. Standridge, G. Abdulla, S. Williams, and E. Fox. "Caching Proxies: Limitations and Potentials," *Proceedings of 1995 World Wide Web Conference*, Boston, 1995, pp.119-133.

[6] H. Bahn, S. Noh, S. Min, and K. Koh, "Efficient Replacement of Nonuniform Objects in Web Caches," *IEEE Computer*, Vol.35, No.6, June 2002, pp.65-73.

[7] R. Wooster, and M. Abrams, "Proxy Caching that

Estimates Page Load Delays," Proceedings of the Sixth International WWW Conference, Santa Clara, CA, April 1997, pp.325-334.

[8] S. Jin, and A. Bestavros, "GreedyDual* Web Caching Algorithm: Expoiting the Two Sources of Temporal Locality in Web Request Streams," *Proceedings of the 5th International Web Caching and Contents Delivery Workshop*, Lisbon, May 2000.

[9] L. Rizzo, and L. Vicisano, "Replacement Policies for a Proxy Cache," *IEEE/ACM Transactions on Networking, 8(2)*, February 1998, pp.158-170.

[10] K. Wu, P. S. Yu, and J. L. Wolf, "Segment-based Proxy Caching of Multimedia Streams," *Proceedings of the 10th International WWW Conference*, May 2001, pp.36-44.

[11] G. K. Zipf, *Human Behavior and the Principle of Least Effort*, Addison Wesley, Reading, MA, 1948.

오 재 학

1997년 광운대학교 컴퓨터과학 학사 1999년 광운대학교 컴퓨터과학 석사 2002년 광운대학교 컴퓨터과학 박사. 현재 (주)코아세스 선임연구원. 관심분야는 멀티미디어 시스템, 실시간 시스템, 운영체제

차 호 정

1985년 서울대학교 컴퓨터공학 학사 1987년 서울대학교 컴퓨터공학 석사 1991년 University of Manchester 전산학 박사. 1993년~2001년 광운대학교 컴퓨터과학과 부교수. 2001년~현재 연세대학교 컴퓨터과학과 교수. 관심분야는 멀티미디어 시스템, 운영체제, 내장형시스템

박 병 준

1984년 서울대학교 컴퓨터공학 학사 1988년 미국 미네소타대학교 컴퓨터과학 석사. 1997년 미국 일리노이대학교 (UIUC) 컴퓨터과학 박사. 미국 Epic Systems, SPSS 등에서 연구원으로 활동하였으며, 현재 광운대학교 컴퓨터과학과 교수. 관심분야는 인공지능, 데이타마이닝, 기계학습 등