

# WWW를 위한 가중화 디스크 캐시 기법에 대한 연구

박 해 우, 강 병 옥  
영남대학교 대학원 컴퓨터공학과  
전화 : 053-810-1510 / 핸드폰 : 018-689-0075

## A Study of Weighted Disk Cache Method for World Wide Web

Hae-Woo Park  
Dept. of Computer Engineering, Graduate School, Yuengnam University  
E-mail : laceed@empal.com

### Abstract

As the use of world wide web is increasing, the number of connections to servers is increasing also. These interactions increase the load of networks and servers. therefore efficient caching strategies for web documents are needed to reduce server load and network traffics by migrating copies of server files closer to the clients that use those files. As One idea of caching policy, we propose a Weighted Disk Cache Replacement Policy(WDCRP) which analyses user's interaction to WWW and adds weight value to each web document. Especially the WDCRP takes account of the history data of cache log, the characteristics of Web requests and the importance of user interactive-actions.

일반적으로 이미 존재하는 대역폭을 효율적으로 사용할 수 있게 하는 소프트웨어적인 대안으로 "캐싱(caching)" 기법을 들 수 있다. 웹 사용자들에게 있어서 캐싱은 자주 요청되는 자료를 서버에 비해 상대적으로 사용자에게 가까운 거리에 존재하는 로컬 영역에 저장하고, 이 자료가 의미를 가지는 동안 반복해서 이용하는 개념이다. 따라서 서버와의 응답시간이 단축되고 전송속도가 증가하는 등 전체적인 네트워크 성능이 극대화되는 효과를 볼 수 있다.

본 연구에서는 기존의 프락시 캐시에서 사용되고 있는 기법을 응용하여 사용자가 요청한 웹 문서에 대한 사용자의 반응을 웹 캐시 정책에 반영하게 하여 클라이언트 측에 적합한 웹 캐시 정책을 수립한다.

## II. 캐시 교체 정책의 설계

### I. 서론

인터넷 웹(WWW) 서버에 대한 사용자 요구의 양은 계속 증가하고 있으며, 꾸준히 증가하는 인터넷 트래픽으로 인해 결국 서버 컴퓨터와 네트워크 링크의 과부하 문제가 지속적으로 대두되고 있는 상황에 직면해 있다.

### 2.1 기본 설계

기존 연구의 경우 웹 요구에 대한 특성화(characterization) 작업을 통해 추출된 패턴을 캐시 교체 정책으로 제시한다. 이러한 캐시 교체 정책들은 특정 웹 캐시 환경에서는 효율성을 가질 수 있으나, 클라이언트 측면에서의 웹 캐시 환경에서의 다양한 특성

을 모두 반영할 수 없는 문제점을 가진다. 따라서 좀 더 클라이언트 측면에서 웹 캐시 환경의 특성을 도출하여 이를 반영하여야 한다.

본 연구는 주어진 클라이언트 환경에서의 사용자 개인의 문서 중요도에 따른 웹 캐시 환경의 특성을 유도한다. 이를 위해 기존의 연구 방법 외에 “문서의 중요도”를 측정하는 “휴리스틱 요소”를 인자로 평가치에 포함하여 캐시 교체 정책을 조정할 수 있게 하는 캐시 교체 정책을 수립한다.

## 2.2 전체 구성

본 연구에서는 Value(i)라는 함수를 두어, 캐시 내에 저장된 문서들의 가치를 평가하고 캐시 내의 저장 공간이 필요한 경우 Value값이 가장 작은 문서부터 차례로 캐시에서 제거한다. 다음 [식 2.1]은 Value(i)의 계산식이다.

$$Value(i) = p_i \cdot \left(\frac{d_i}{s_i}\right) \quad [식 2.1]$$

- $p_i$  : 문서 i의 미래 참조 예측값
- $d_i$  : 문서 i를 근원지 서버에서 인출해 오는 데 걸린 시간
- $s_i$  : 문서 i의 크기(byte)

객체의 크기 및 인출 비용이 동일한 환경에서는 미래에 가장 참조 가능성이 적은 문서,  $p_i$ 가 가장 작은 문서를 캐시에서 삭제하면 되지만, 웹 환경에서는 웹 문서마다 사이즈와 인출 비용이 다르기 때문에 미래 참조율 외에 이러한 요소들을 함께 고려해서 웹 문서의 가치를 평가하는 것이 바람직하다.

정해진 캐시용량 하에서 높은 DSR을 얻기 위해서는 각각의 웹 문서를 단위 크기 당 가치로 평가해야 하며, 이는 인출 지연 시간과 참조 가능성을 동시에 고려한 것이어야 한다. [식 2.1]은 웹 문서의 단위 크기 당 지연시간<sup>1)</sup>( $d_i/s_i$ )을 그 문서의 참조율( $p_i$ )로 정규화시켜 성능 척도 DSR에 맞게 웹 문서의 가치를 평가하도록 Value를 정의한 것이다. 이러한 모델 하에서 교체 정책의 우수성은  $p_i$ 를 얼마나 잘 예측하는가에 좌우된다. 본 연구에서는 참조율에 예측값  $p_i$ 를 과거 참조 기록과 웹 요구의 특성, 그리고 사용자 가중치를 요소로 하여 이를 조합하여 [식 2.2]와 같은 수식으로 나타낸다.

$$p_i = L(i) \cdot W(i) + cB(i) \quad [식 2.2]$$

- $L(i)$  : 문서 i의 과거 참조 기록의 미래 참조 예측값
- $W(i)$  : 문서 i의 요구 특성 분석을 통한 미래 참조

예측값

- $B(i)$  : 문서 I의 사용자의 문서 중요도의 미래 참조 예측값

·  $c$  : 상수

본 연구에서는  $L(i)$ 를 과거  $k$ 번 참조되는 동안의 기록을 이용하도록 [식 2.3]과 같이 정의한다[3].

$$L(i) = \frac{k}{(t - t_k)} \quad [식 2.3]$$

·  $k$  : 과거 참조 회수

·  $t$  : 현재 시각

·  $t_k$  :  $k$ 번 이전에 참조되었을 때의 시각

[식 2.3]에서  $H(i)$ 는 과거  $k$ 번 참조되는 동안의 평균적인 참조율을 의미한다. 여기서 실제 구현에서는 매번 캐시 교체가 있을 때마다 캐시 상의 모든 문서에 대해 현재 시각  $t$ 를 재 설정해서  $p_i$ 를 새로 구해야 한다. 이는 작업 부하가 크므로 특정 시간 간격에 대해 주기적으로  $p_i$ 를 재계산하는 근사 방법을 사용한다.  $k$  값, 즉 과거 참조 기록을 어떻게 사용할 것인가에 대해서는 여러 가지 방법이 있다. 캐시 상의 문서들에 대해서만 참조 회수  $k$ 와  $t_k$ 를 유지하는 방법을 사용하기로 한다.

$$W(i) = \frac{R_i}{R_t} \quad [식 2.4]$$

·  $R_i$  : 특정 문서의 요구 패턴 수치

·  $R_t$  : 전체 요구 패턴 수치

[식 2.4]의  $W(i)$ 는 웹 문서에 대한 요구 특성을 분석한 값으로 전체 요구 패턴에 대한 특정 문서의 요구 패턴의 일치율을 나타낸다. 이 경우 푸리에 변환, 신경망, 유전 알고리즘 등의 방법을 적용하여 표현할 수 있다.

$$cB(i) = c \left(\frac{I_i}{I_t} \cdot D\right) \quad [식 2.5]$$

·  $I_i$  : 문서 뷰어 프로세서와의 interactive-action 값

·  $I_t$  : 문서 뷰어 프로세서와의 interactive-action동안 지속 시간

·  $c$  : 상수

[식 2.5]의  $cB(i)$ 는 특정 문서에 대한 사용자의 반응 유형을 분석한 값이다.

$$D = \begin{cases} -1, & \text{if Value}(i) \text{ is increasing} \\ +1, & \text{if Value}(i) \text{ is decreasing} \end{cases} \quad [식 2.6]$$

[식 2.6]은 인자 D에 대한 값을 표현하고 있다. 인자 D는 방향성을 나타내는 값으로 증가 추세인 경우 +1의 값을 가지고 감소 추세의 경우 -1의 값을 가진다.

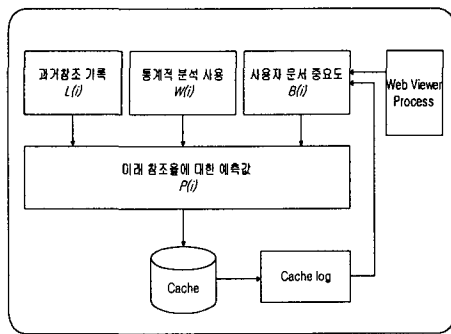
## III. 가중화 디스크 알고리즘

1) 한 번 캐시 직중 시 해당 캐시 바이트가 줄일 수 있는 지연 시간

### 3.1 사용자 반응 유형 분석 방법

[식 2.5]에서  $cB(i)$ 는 특정 웹 문서에 대한 사용자의 반응 특성을 반영하는 파라미터로서 제안하는 클라이언트 웹 캐시를 위한 부분이며 웹 뷰어 프로세서와의 사용자 행동을 분석, 기록하여 휴리스틱 방법을 적용할 수 있다.

제안하는 캐시 교체 정책은 특정 문서에 대한 사용자의 interactive-action의 종류를 구분하고 문서 중요도에 따른 차등화 된 가중치를 부여한다. 이러한 가중치를 바탕으로 사용자의 제각기 다른 문서 중요도를 미래 참조에 예측, 반영하는 것이 가능하다. 다음 [그림 3.1]은 미래 참조 예측값  $pi$ 의 메커니즘을 나타내고 있다.



[그림 3.1] 미래참조 예측값  $pi$ 의 메커니즘

사용자 문서 중요도  $B(i)$ 는 캐시 로그의 정보를 이용한다. 캐시 로그는 URL, 문서의 크기, 문서의 타입, 문서의 요청 시간, 근원지에서 인출해 오는데 걸린 시간, 리퀘스터(requestor) 등의 정보를 담고 있다.

### 3.2 UACA(User Action Check Agent)

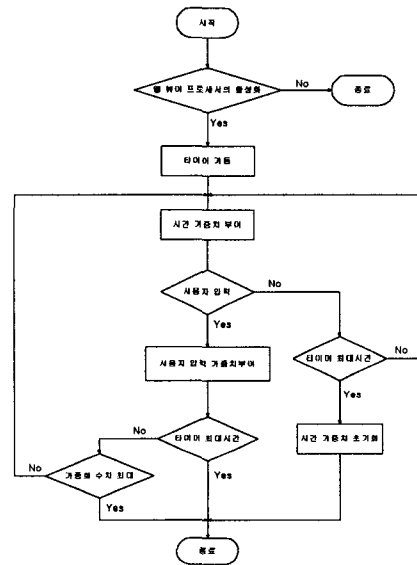
[그림 3.1]에서의 “사용자 문서 중요도“에 대한 사용자 가중치를 추출해 내기 위해서는 특정 문서에 대한 웹 뷰어 프로세서와 사용자간의 인터페이스 양을 통해 알 수 있다. 이러한 사용자의 휴리스틱 입력을 바탕으로 본 논문에서는 UACA(User Action Check Agent)라 정의하는 에이전트를 사용한다. UACA는 다음 정보를 입력받는다.

- 캐시 로그(Cache log)
- 사용자의 직접적인 입력
- UACA 자체의 사용자 모니터링

캐시 로그는 캐시 메모리에서 캐시-인(cache-in)과 캐시-아웃(cache-out)된 캐시 로그 히스토리의 정보를

가지고 있다. UACA는 기본적으로 캐시 로그를 통해 입력받은 정보와 사용자 모니터링을 통해 정보를 갱신하게 된다.

문서에 대한 중요도 측정에 영향을 미치는 사용자의 행동으로는 여러 가지 요인이 있을 수 있다. UACA는 웹 뷰어 프로세서 상에서의 사용자의 마우스 입력과 키보드 입력 행위를 모니터링하고 이를 웹 문서에 대한 가중치로 부여하여, 저장하는 역할을 담당하는 부분이다. UACA에 대한 개략적인 순서도는 [그림 3.2]과 같다.



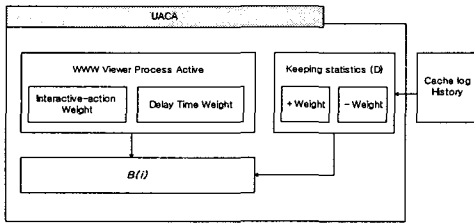
[그림 3.2] UACA 기본 동작 순서도

UACA는 다음 세 부분으로 구성된다. 첫 번째 부분은 활성화 체크 부분이다. UACA의 구동 여부는 웹 뷰어 프로세서의 활성화에 따라 결정된다. 두 번째 부분은 타이머 카운트 부분이다. UACA가 활성화되면 타이머에 의해 작업 시간에 대한 가중치가 누적된다. 작업 시간이 길수록 가중치가 누적되지만 최대값 이상의 가중치는 부여되지 않는다. 또한 사용자 입력 없이 타이머가 한계시간을 초과하게 되면 사용자가 작업을 중단한 경우로 간주하고 누적된 시간 가중치를 초기화한 후 종료한다. 그리고 사용자 입력이 있으면 시간 가중치와 사용자의 입력 가중화 수치를 누적하여 부가한다. 세 번째 부분은 사용자 행동을 모니터링하는 부분이다. 이 부분은 사용자의 웹 뷰어 프로세서 상에서의 마우스 입력과 키보드의 키 입력을 모니터링 한다. 사용자의 마우스나 키보드의 입력이 많을수록 높은 가중치

가 누적된다.

보통 사용자의 입력은 마우스와 키보드의 사용으로 이루어지며, 마우스와 키보드의 조합에 대한 입력 방식도 구분하여 처리한다. 여기서 작업의 종류에 따라 다른 가중치를 부여한다. 사용자 가중치 입력의 예외 상황으로 웹 문서들 중 마우스를 이용한 '자바 애플릿 게임', '자바 애플릿 채팅', '글쓰기' 등의 경우에는 가중치 최대 값으로 처리한다.

한편, 특정 문서에 대한 사용자 interactive-action의 양이 많다 하더라도 항상 그 문서의 내용이 유지 되는 것은 아니라 가장 빈번히 갱신되어야 하는 경우가 있다. 가령 실시간으로 뉴스 정보를 제공해 주는 CNN의 웹 문서 정보는 매우 빈번히 교체되어야 할 것이다. 반대로 법률 정보를 제공하는 웹 문서의 경우처럼 내용이 자주 바뀌지 않는 곳을 이용하는 사용자의 경우에는 해당 문서에 대한 interactive-action의 양은 많지만 캐시 된 웹 문서는 상대적으로 교체가 되지 않아야 할 것이다. 두 경우 모두 사용자의 interactive-action이 많다고 가정하면 웹 문서가 중요하다라는 점에는 공통적이지만 그 교체 정책에 있어서는 전혀 반대의 양상을 가지게 된다. 이를 위해서 고안된 인자가 바로 [식 2.5]의 D이다.



[그림 3.3] 인자 B(i) 결정 메커니즘

웹 뷰어 프로세서의 활성화와 함께 사용자 입력과 시간 가중치, 그리고 캐시 로그를 통한 문서 가중화 유형을 통해서 인자 D가 정해진다. 인자 D는 이전의 캐시 유지 기록을 살펴보았을 때 해당 문서가 유지되는 추세에 있는지 아니면 교체되는 추세에 있는지를 판단하여 앞에서 계산된 사용자 interactive-action의 양에 방향성을 부여하게 된다. 이 방향성에 의해서 B(i)의 방향성이 조절된다.

[그림 3.3]은 B(i)의 메커니즘을 보인다. 해당 문서가 캐시 장소에 보관되는 추세에 있다면 사용자 interactive-action의 양으로 계산된 가중치가 양의 값을 가지게 되어 결과적으로 전체 참조 예측값은 커지게 되며, 반대로 제거되는 추세에 있다면 가중치가 음

의 값을 가지게 되어 결국 전체 참조값의 예측값은 작아지게 된다. c는 B(i)값의 조정을 위한 상수 값이다. B(i)값이 미세하게 변화하였을 경우 이를 일정한 상수 값으로 곱하여 값의 크기를 크게 하게 위해 사용된다. 이 부분을 통해서 사용자 반응 유형의 값을 조절하는 할 수 있다.

#### IV. 결론

제안된 캐시 교체 정책의 구조적 특징은 다양한 사용자 환경을 그 기반으로 하여 일반적 웹 특성만을 고려하거나 특정 환경에서 적용 가능한 기존의 캐시 교체 정책과는 달리 다양한 웹 사용자에게 대한 클라이언트 환경에서의 능동적이고 유동적인 캐시 교체 정책을 설계한다.

본 논문은 기존의 캐시 교체 정책의 장점을 포함하고 단점을 과거 참조 기록과 통계적 분석 사용, 사용자의 문서 중요도를 미래 참조에 대한 예측값으로 사용하여 일률적인 캐시 환경에서의 교체 정책이 가변적인 캐시 교체 정책이라는 점에서 효율성이 높다고 할 수 있다.

#### 참고문헌

- [1] S. Willians, M. Abrams, C. R. Standridge, G. Abdulla, and E. A. Fox, "Removal policies in network caches for world-wide web document", In Proceedings of ACM SIGCOMM '96, Aug. 1996.
- [2] P. Lorenzetti, L. Rizzo, and L. Vicisano, "Replacement Policies for a Proxy Cache", 1996.
- [3] P. Scheuermann, J. Shim, and R. Vingralek, "A Case for Delay-Conscious Caching of Web Documents", In Proceedings of the 6th International WWW Conference, 1997.
- [4] S. Willians, M. Abrams, C. Standridge, G. Abdulla, and E. Fox, "Removal policies in network caches for world-wide web documents", In proceedings of the ACM SIGCOMM '96, 1996.