NOD(News On Demand) 데이터의 인기도와 생명주기를 이용하는 시간 윈도우에 기반한 캐시 재배치 기법

최태욱°, 박성호, 김영주, 정기동
부산대학교 전자계산학과

Time Window based Cache Replacement Strategy using Popularity and Life Cycle of News-on-Demand Data

Tae-uk Choi, Sung-Ho Park, Young-ju Kim, Ki-dong Chung
Department of Computer Science, Pusan National University

요 약

뉴스기사를 구성하는 NOD 데이터는 VOD(Video on Demand)데이터와는 달리 미디어의 종류 및 크기, 시간적인 접근 지역성, 사용자와 상호 작용성 등의 차이점을 가질 뿐만 아니라 새로운 뉴스기사가 수시로 생성되고 사용자가 인기도가 높은 기사와 최신의 뉴스기사에 더 많이 접근하는 특성을 가진다. 본 논문에서는 실제 서비스중인 전자신문의 로그파일을 분석하여 NOD 뉴스기사의 인기도가 Zipf 분포와 다름을 보이고, 뉴스기사의 생명주기(Life Cycle)에 따른 접근 확률분포 제시한다. NOD 데이터의 접근 편기성으로 인하여 데이터 캐싱을 통한 NOD 서버의 성능 향상을 기대할 수 있으나 뉴스기사의 생명주기가 짧고 접근시간대별로 사용자 접근형태가 변하는 등의 이유로 단순히 인기도만 고려한 캐싱은 빈번한 데이터 재배치 문제로 인해 높은 캐시 관리비용을 야기한다. 따라서 본 논문에서는 뉴스 기사의 접근 편기성에 나타나는 인기도(popularity)와 생명주기를 조합한 척도를 제안하고 이를 이용한 재배치 기법을 제안한다.

1 서론

컴퓨터 및 네트워크, 그리고 통신기술의 발달과 멀티미디어 관련기술의 발전으로 인하여 활자인쇄의 신문을 통한 뉴스정보보다는 멀티미디어를 통한 뉴스정보에 대한 요구가 증가하고 있으며 이에 'NOD(News On Demand)' 멀티미디어 응용에 관심이 집중되고 있다. 뉴스기사를 구성하는 NOD 데이타는 VOD(Video On Demand)의 비디오 데이터와는 달리 미디어의 다양성 상대적으로 작은 크기, 사용자와의 빈번한 상호작용 등의 주요한 차이점을 가진다[박 97].

특히 NOD 데이타는 다음과 같은 특성들을 가진다. 첫째로, 뉴스 기사가 수시로 생성된다. 비디오 프로그램의 경우 보통 그 생성주기가 일주일에서 보름단위 정도이다. 따라서 사용자의 선호도 또한 그 생성주기를 따라서 천천히 변한다. 그러나 뉴스기사의 경우 하루에도 수시로 생성되기 때문에 그 생명주기가 비디오 데이터에 비해 매우 짧다. 둘째, 사용자는 새로운 기사를 선호한다. 좋은 비디오 프로그램은 상당히 지속적으로 사용자들이 선호하나, NOD 뉴스 기사의 경우는 3 일이 지나면 거의 접근되지 않는다[이 97]. 셋째, 사용자가 한번에 선택하는 데이터의 개수가 다르다. 비디오 데이타의 경우 시청하는데 시간이 많이 필요하기 때문에 사용자는 대부분 한두 편 정도의 비디오를 보지만, NOD 데이타는 몇 분이면 하나의 기사를 볼 수 있기 때문에 사용자는 보통 한번에 여러 개의 기사를 본다. 넷째, NOD 데이타는 시간적 공간적 접근 지역성을 가진다. 사용자는 특정 인기 있는 기사를 주로 액세스하고, 또한 특정 시간대에 집중적으로 접근하는 경향이 있다

상기의 NOD 데이타의 접근 지역성으로 인하여 데이터의 캐싱을 통한 NOD 서버의 성능 향상을 기대할 수 있다. 그러나 뉴스기사는 생성주기가 짧고 접근시간대 별로 사용자 접근형태가 변하는 등의 이유로 단순히 인기도 모델을 바탕으로 한 캐싱은 빈번한 데이터 재배치 문제를 야기시킨다. 따라서 본 논문에서는 실제 운영중인 전자신문의 로그파일을 분석하여 NOD 데이타의 인기도(Popularity) 모델과 생명주기(Life Cycle) 모델을 제시하고 이를 바탕으로 뉴스기사의 중요도에 관한 새로운 척도를 제시한다. 그리고 이를 이용하여 시간 윈도우(Time Window)를 기반으로 한 NOD 서버의 캐시 재배치 기법을 제안한다.

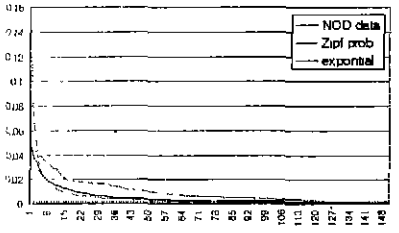
2. 관련 연구

멀티미디어 데이터에 대한 캐싱 정책을 세우는 데 인기도와 생명주기(Life Cycle)의 모델이 매우 중요하다. [Scott 94]는 실제 비디오 인기도 분포가 Zipf 분포 보다는 double exponential 에 가깝다고 주장하였으며, 생명주기를 고려해서 캐싱 전략을 제시하고 있다. [Car97]은 계층적인(hierarchical) VOD 시스템을 위한 Long-term Movie Popularity 모델을 제시하고 있는데, 3 개의 파라메타를 가진 지수분포의 변형이다. 이 두 논문은 생명주기가 큰 VOD 데이타를 분석한 것으로 NOD 데이타에는 적합하지 않다. [Jam95]는 인지적인 분석을 통해 WWW 서버에서는 빈도수(frequency)보다는 최신도(recency)가 더 중요한 척도임을 주장하고, 최신의 데이터를 캐시에 배치하는 알고리즘을 제안했다. 그러나 단순히 최신도만을 가지고 NOD 데이타를 분석하기 무리다. [박 97]는 사전에 사용자의 접근패턴을 조사하여 접근 가능성이 높은 비디오 오브젝트 전체를 프리페칭하는데, 이때의 재배치 척도로서 기사의 임계값(최대 접근 횟수)과 현재 접근횟수의 차이를 사용한다. 그러나 임계값을 정확히 알 수 없다는 단점이 있다

3. 뉴스기사의 인기도(Popularity)와 생명주기(Life Cycle)

3.1 인기도(Popularity)

비디오 데이터의 인기도 모델로서 Zipf 분포를 들 수 있는데, 이것은 가장 인기 있는 비디오부터 순위(rank)를 줄 때, 그 순위를 가진 비디오가 접근될 확률을 나타내는 분포이다. 그러나 현재 운영되고 있는 부산일보 전자신문의 로그파일을 분석한 결과 뉴스기사의 인기도 모델이 Zipf 분포와 상당한 차이가 난다.

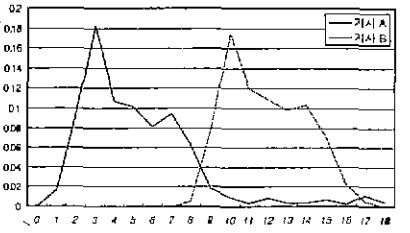


[그림 1] NOD 뉴스기사의 인기도 분포

[그림 1]을 보면 NOD 데이터가 상위 부분에서 Zipf 분포에 비해 비교적 완만하게 감소하고 있음을 알 수 있는데, 이는 한 명의 사용자가 NOD 서버에 접속하면 여러 개의 기사를 같이 보기 때문에 상대적으로 상위부분의 기사들이 함께 접근되는 경우가 많기 때문이다. 또한 NOD 데이타의 확률 분포는 exponential 분포 보다는 편기성이 큼을 알 수 있다. 비교를 위해 Zipf 분포의 θ가 없는 원래의 분포를 사용했으며, exponential 분포의 λ값은 0.01 을 사용하였다.

3.2 생명주기(Life cycle)

[그림 2]는 생성주기에 따른 접근확률의 분포를 나타내는데, 생성된 직후 접근확률이 가장 높고 그 다음 서서히 감소하는 경향을 가진다. 이것은 사용자가 오래된 기사보다 새로운 기사를 선호한다는 뉴스기사의 특성에서 기인한다. 또한, 캐시 안에 있는 기사 A 의 접근확률이 감소하고 있고 기사 B 가 새로 생성되어 접근확률이 증가할 때 두 확률곡선이 만나는 시점이 두 기사를 바꾸어서 재배치할 최적의 시점이 된다.



[그림 2] 생명주기에 따른 접근 확률 분포

본 논문에서는 뉴스기사의 인기도(Popularity)를 나타내는 척도로써 빈도수(frequency)를 사용하고, 생명주기(Life Cycle)의 척도로서 빈도수의 증감량을 사용한다. 즉 빈도수의 변화가 큰 폭으로 증가한 기사는 생성된 지 얼마 안되었음을 나타내고, 감소하는 기사들은 생성 후 어느 정도 되었음을 나타내고, 거의 변화가 없는 경우는 아주 오래된 기사임을 나타낸다. 따라서 이러한 빈도수와 생명주기를 같이 고려할 때 최적의 재배치 시점을 찾을 수 있다.

4. 시간 윈도우(Time Window)에 기반한 재배치 기법

비디오데이터와는 달리 뉴스기사 데이터는 수시로 생성되고 사용자는 최신의 뉴스를 선호하기 때문에 캐시를 자주 갱신하여야 한다. 또한 전통적인 데이터와는 달리 3~5 분 정도의 동영상 뉴스기사의 경우 그 크기가 수십 Mbyte 이므로 재배치의 비용이 매우 크다. 따라서 NOD 서버에서는 이러한 그 비용을 줄이는 것이 중요하다. 본 논문에서는 NOD 데이타의 특성을 이용하여 재배치 척도와 재배치 기법을 제안한다

4.1 LBF (Lifecycle Based Frequency)

본 논문에서는 빈도수(frequency)를 계산하기 위해서 일정한 시간 간격으로 시간 윈도우를 나누고 이 동안 발생하는 요구들의 수를 누적하여, 그 윈도우 i 에서의 빈도수(f_i)를 구하고 다음과 같이 빈도수의 예상 증감량(r_i)을 다음과 같이 계산한다.

$$\begin{aligned} r_i &= SF\,(f_{i-1} - f_i) + (1 - SF\,)r_{i-1} \\ &= SF\,(f_{i-1} - f_i) + SF\,(1 - SF\,)(f_{i-2} - f_{i-1}) + \\ &\quad SF\,(1 - SF\,)^2(f_{i-3} - f_{i-2}) + \ldots + (1 - SF\,)^n r_0 \end{aligned} \qquad [\text{식 } 1]$$

[식 1]에서 SF(Smoothing Factor)는 현재 윈도우의 빈도수 증감량에 대한 가중치를 나타내며, r_i 값은 모든 윈도우들의 빈도수 증감량에 대한 가중평균을 의미한다.

따라서 재배치 척도(LBF_i)는 현재 윈도우의 빈도수(f_i)와 예상 증감량(r_i)을 더하여 다음과 같이 나타낼 수 있다.

$$LBF_i = f_i + r_i \quad \text{[식 2]}$$

4.2 재배치 기법

하나의 시간 윈도우가 끝날 때, 각각의 기사에 대해 LBF_i를 계산하고 그 값이 높은 기사 순으로 캐시 안의 값이 낮은 기사와 바꾸어 재배치를 한다. 그리고 재배치된 기사들은 다음 시간 윈도우가 끝날 때까지 캐시 안에서 유지된다. [그림 3]은 재배치 알고리즘을 나타낸다.

```
While(1) {
  If(하나의 윈도가 끝날 때){
    While(윈도안의 모든 기사){
      r_i = SF(f_i − f_{i−1}) + (1 − SF)r_{i−1};
      LBF_i = f_i + r_i;
    }
    윈도정보를 기사 table 에 추가한다;
    기사 table 을 LBF_i 순으로 정렬한다;
    while(캐시에 들어갈 상위 n 기사)
      if (이미 캐시안에 존재) continue;
      else 캐시안의 가장 작은 LBF_i 를
        가지는 기사와 재배치;
}}
```

[그림 3] 재배치 알고리즘

5 모의 실험

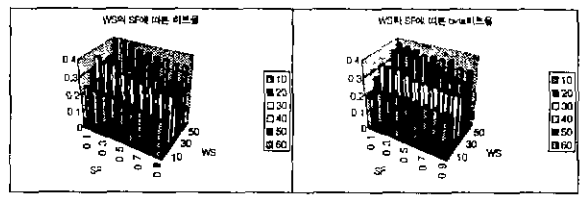
본 실험은 Sun Ultra Sparc 워크스테이션에서 C 언어로 수행하였고 실험 데이터는 부산일보 전자신문의 로그파일을 그대로 사용하였다. 캐시 크기는 256Mbyte 로 가정하였고, 데이터의 종류는 Text, Image, Video 3 가지를 사용하였으며, 그 크기와 비율은 [그림 4]와 같다.

종류	크기	비율
Text	10K	50%
Image	40K	30%
Video	30M	20%

[그림 4] 실험 데이터

5.1 WS 와 SF 에 따른 히트율 비교

[그림 4]는 WS(Window Size)와 SF(Smoothing Factor)에 따른 Hit ratio 와 Byte hit ratio 를 나타낸 것으로, WS 가 60 분, SF 가 0.1 일 때, 가장 성능이 좋음을 알 수 있다. 이는 부산일보 로그파일내의 기사들의 생명주기가 매끄럽지 못하여 smoothing 효과가 클 때, 성능이 높아짐을 나타낸다.



[그림 4] WS 와 SF 에 따른 hit-ratio & byte hit-ratio

5.2 LRU 와의 성능비교

재배치가 많이 일어날수록 재배치를 위한 디스크 접근횟수가 증가하며, NOD 데이타와 같이 그 크기가 클 경우 디스크 접근량은 더 커진다. 따라서, 이러한 디스크 접근횟수와 접근량을 줄임으로써 시스템의 성능향상을 기대할 수 있다. [그림 5]는 LRU 와의 성능비교를 나타내는데, Hit ratio 와 Byte hit ratio 는 비슷하지만 디스크 접근횟수와 디스크 접근량에서 5 배 이상의 차이를 남을 알 수 있다.

	Hit ratio	Byte hit ratio	Disk 접근수	Disk 접근량
LRU	0.404	0.379	5148	33139M
LBF	0.399	0.376	919	5414M

[그림 5] LRU 와의 성능 비교

6. 결론 및 향후 과제

본 논문에서는 NOD 뉴스데이타의 여러 가지 특성 중 접근 편기성에 나타나는 인기도와 생명주기의 척도들을 조합하여 새로운 척도를 제시하고 재배치 비용을 줄일 수 있는 시간 윈도우에 기반한 캐시 재배치 방법을 제안하였다. 그 결과 제한된 자원하에서 빈도수와 생명주기를 함께 고려한 기법이 LRU 기법보다는 디스크 접근회수와 디스크 접근량을 5 배 이상 줄일 수 있었다. 향후에는 NOD 데이타의 인기도 분포와 최근도 분포들의 모델을 추정하려고 한다.

[참고 논문]

[Jam95] James E. Pitkow, Margaret M.Recker, A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns, In the Proceeding of the Second International WWW Conference, 1995

[Sco95] Scott A. Barnett, Gary J.Anido, H.W.Beadle. Caching Policies in a Distributed Video-on-Demand System, In the Proceeding of Australian Telecommunication Networks and Applications Conference, 1995

[Car97] Carsten Griwodz, Michael Bar, Lars C Wolf, Long-term Movie Popularity Models in Video-on-Demand Systems or The Life of an on-Demand Movie, The fifth ACM International Multimedia Conference, 1997

[박 97] 박용운, 백건효, 서원일, 김영주, 정기동, NOD 데이타를 위한 새로운 버퍼링 기법, 한국방송공학회 학술 대회, 1997

[이 97] 이주경, 박용운, 김영주, 정기동, NOD 데이타를 위한 데이터 배치방법, 한국정보과학회 추계학술 발표대회, 1997