

NOD 데이터를 위한 새로운 버퍼링 기법

박용운, 백건효, 서원일, 김영주, 정기동
부산대학교 전자계산학과

New Data Buffering Scheme for News On Demand

Yong-Woon Park, Keon-Hyo Baek, Won-Il Seo, Young-Ju Kim, Ki-Dong Chung
Dept. of Computer Science, Pusan National University

요약

본 연구에서는 실시간 데이터와 비 실시간 데이터가 복합적으로 존재하는 뉴스 데이터에 적합하도록 버퍼 캐쉬를 실시간 데이터와 비 실시간 데이터 영역으로 분할 한 후, 로그 데이터를 이용하여 접근 가능성이 높은 실시간 뉴스데이터를 프리팹칭하여 둬으로써 실시간 뉴스 데이터의 운영을 효과적으로 할 수 있는 새로운 버퍼 캐쉬 알고리즘을 제안한다. 이 방식을 사용함으로써 전체 뉴스 요청 건수 중 30% 이상의 요청 건수들이 디스크를 접근하지 않고 버퍼의 데이터를 접근함으로써 버퍼링 기법을 사용하지 않은 경우보다 실시간 지원에 필요한 디스크 접근 수를 줄일 수 있다.

1. 서론

통신 기술의 발전과 멀티미디어의 등장으로 종전의 텍스트 처리 방식의 데이터 운영에서 벗어나 멀티미디어 데이터를 이용한 응용들이 등장하고 있다. 이 중에서도 일반 사용자들에게 가장 접근해있는 응용으로는 VOD(Video On Demand)와 NOD(News On Demand)등을 들 수 있다. 그러나 학교나 연구기관 또는 기업을 중심으로 행하여지고 있는 연구의 대부분은 VOD를 중심으로 이루어지고 있으며 NOD에 대한 연구는 몇몇 학교나 연구기관 또는 기업에서 부분적인 테마 즉, 디지털 라이브러리 구축 또는 새로운 메타포어의 정의 등을 중심으로 행해지고 있을 뿐이다. 그러나 본 연구에서 다루게 될 NOD는 동영상이나 오디오 데이터를 다룬다는 점에서 VOD와 유사한 점이 많지만 또한 다른 면도 많으며 그 중요한 몇 가지 차이점을 들어보면

다음과 같다.

첫째, 데이터의 길이이다. 통상적인 비디오 프로그램의 재생시간은 80분 전후이다. 그러나 뉴스 데이터의 경우 하나의 뉴스가 길어도 3분을 초과하는 경우는 극히 드물다.

만약 MPEG II의 경우 정상적인 재생을 위해서는 초당 8 Mbit 정도가 필요하므로 80분의 비디오 프로그램이 약 5기가 바이트 정도의 저장 공간이 필요한 반면 뉴스 데이터의 경우 180메가 바이트 정도의 저장 공간 만이 필요하다.

둘째, 데이터의 수명(life span)이다. 비디오 프로그램의 경우 매일 새로운 프로그램이 출시되기 보다는 일주일 내지는 보름 등의 비교적 긴 주기로 출시된다. 그러므로 사용자들의 비디오 프로그램 선호도도 그 주기와 함께 서서히 변하는 속성을 가진다. 반면 뉴스 데이터의 경우 이들이 경과한 뉴스의 경

우는 거의 요구될 가능성이 없다[97 이].

셋째, 데이터의 증가량이다. 비디오 데이터의 경우에는 일단 새로운 프로그램이 들어오면 디스크 등의 저장 장치에 저장한 다음 신작 프로그램이 들어올 때 까지는 프로그램의 선호도에 별다른 이상이 없는 한 변동 없이 디스크에 상주할 수 있다. 그러나 뉴스 데이터의 경우 매일 또는 만나절을 주기로 새로운 데이터가 쏟아져 들어오므로 만약 이 데이터를 모두 디스크에 저장한다고 가정한다면 엄청난 양의 데이터 저장 공간이 필요하다.

넷째, 데이터의 다양성이다. 비디오 데이터의 경우 일부 메타 데이터를 제외하고는 대부분이 동영상 데이터인 반면 NOD 데이터는 동영상 뿐만 아니라 오디오, 텍스트 그리고 이미지 데이터로 이루어지며 경우에 따라서는 이러한 데이터들이 한번의 요청에 복합적으로 나타나는 경우도 있다.

다섯째, 데이터의 재 사용률이다. 비디오 데이터의 경우 인기가 높다고 하더라도 데이터의 크기가 방대한 만큼 전체 데이터를 전부 캐싱하지 못한다. 메모리 가격이 떨어졌다고는 하지만 수 기가에 이르는 비디오 데이터를 전부 캐싱하기는 어려우나 뉴스 데이터의 경우 완벽한 예측은 불가능해도 실제 재 사용률에 가까운 예측은 가능하다.

[97 이]의 조사에 따르면 인터넷 신문 접근 유형을 조사한 결과 총 뉴스 데이터 중 정치, 사회, 스포츠, 경제를 접근하는 비율이 전체의 70% 이상을 차지하고 있다. 또한 인터넷의 일반화로 뉴스 데이터를 인터넷을 통하여 접근하는 사용자들이 점차 늘어나고 국내외의 각종 언론사에 의해서 뉴스만을 전문으로 다루는 사이트들이 증가하고 있으며 이들이 지원하는 데이터도 점차로 텍스트나 이미지 뉴스에서 탈피하여 점차적으로 동영상이나 오디오 데이터를 제공하고 있다. [97 Tew]에 따르면 향후 5년 안에 인터넷을 통하여 접근되는 데이터의 50%가 실시간 데이터가 된다고 한다. 이러한 NOD 데이터의 특성을 고려할 때 NOD 시스템 구축 시에는 VOD와는 다른 데이터 처리 기법이 필요함을 알 수 있다. 특히 그 중에서도 데이터의 재사용과 디스크 지연 시간을 줄이기 위해 사용되는 데이터

버퍼링 기법은 재래식 데이터에 사용되는 기법 뿐만 아니라 VOD 데이터의 기법과도 다른 새로운 방식의 데이터 버퍼링 기법이 필요하며 본 연구에서는 이러한 뉴스 데이터의 버퍼링 기법을 제안한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 멀티미디어 데이터의 버퍼링과 관련된 연구들을 소개한다. 제 3 장에서는 본 논문에서 제안하고 있는 NOD 용 데이터 버퍼링 정책 및 알고리즘을 소개한다. 제 4 장에서는 이 논문에서 제안하는 버퍼링 정책을 시뮬레이션을 통하여 버퍼링 정책을 사용하지 않았을 경우와 비교, 분석한다. 마지막 장인 5 장에서는 결론을 내리고 향후 연구 방향에 대해서 논한다.

2. 관련 연구

멀티미디어 데이터 즉, 동영상, 오디오 등의 데이터의 캐싱이나 버퍼링에 관한 연구는 데이터의 재 사용률을 높이는 측면 보다는 디스크 스케줄링을 원활히 하여 디스크의 사용률을 높이기 위한 측면으로 연구의 초점이 맞추어져 있었으며 그 중에서도 NOD 데이터의 버퍼링에 관한 연구는 거의 없다. 실시간 데이터의 버퍼링 연구의 몇 가지 사례를 들면 다음과 같다. [95 Tew]에서는 일정한 시간 구간을 두어 버퍼링하는 구간 캐싱 방식으로 임의의 비디오 프로그램이 요청되었을 때 해당 비디오 오브젝트를 디스크로부터 읽어와서 캐싱한 다음 어느 일정한 시간 동안 그 비디오 오브젝트를 요구하는 사용자가 있을 때에는 후속 사용자는 디스크에서 데이터를 읽어오지 않고 캐싱된 데이터를 읽으며 선행 사용자는 캐싱을 계속해 나가고 그렇지 않을 때에는 캐싱을 멈추는 메커니즘을 제시하고 있다. 이 경우는 요청되는 비디오 프로그램이 다양하고 캐싱된 프로그램을 요청하는 사용자 수가 그다지 많지 않을 때에는 방대한 메모리만을 요구할 뿐 효율은 높지 않다. 또한 [95 Kun]의 경우에는 뉴스 데이터를 대상으로 하고 있으나 데이터의 재 사용률에 초점을 맞추기 보다는 디스크의 효율적인 사용을 위한 부가적인 메커니즘 쪽에 초점을 맞추고 있

다. [97 Wei]에서는 비디오 데이터의 버퍼링을 재사용보다는 입출력 시에 발생할 수 있는 병목 현상 해결을 위하여 여러 비디오 요청들 사이에 버퍼를 어떻게 적절히 분배하여야 전체 성능 향상을 꾀할 수 있는 지에 초점을 맞추고 있다. 이상의 몇 가지 예에서 보듯이 순수하게 뉴스 데이터의 특성을 고려하여 디스크의 효율성을 높이고 데이터의 재사용률을 높이기 위한 뉴스 데이터의 버퍼 캐쉬에 대한 연구는 없으며 일반적인 멀티미디어 데이터를 기준으로 연구들이 진행되고 있다.

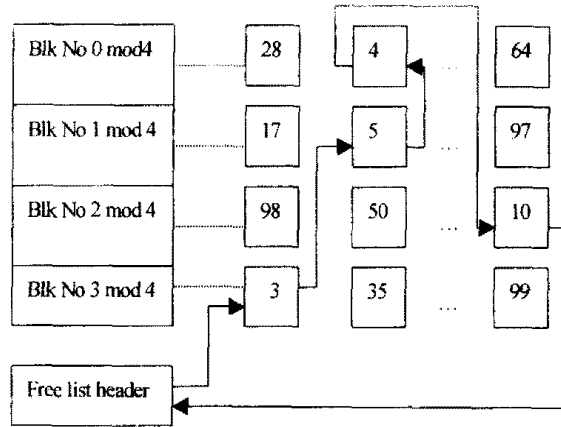
3. 뉴스 데이터를 위한 버퍼 캐쉬 메커니즘

3.1 재래식 버퍼 캐쉬를 이용한 멀티미디어 데이터의 캐싱

그림 1은 기존 유닉스 시스템의 버퍼캐쉬의 운영 방법을 설명한다. 사용자가 데이터의 읽기 또는 쓰기 요청이 있으면 그림 1에서처럼 먼저 버퍼캐쉬를 찾아가서 블록 번호와 장치번호로 해싱 함수를 이용하여 해당 블록이 존재하는 버퍼 리스트로 찾아가게 되어있으며 해당 블록이 버퍼 캐쉬에 존재하지 않을 때는 디스크 읽기를 하여 버퍼 캐쉬로 불러온다. 이러한 재래식 버퍼 캐쉬운영 방법은 사무계산용 데이터처럼 크기가 비교적 작고 데이터의 시간적, 공간적 국부성이 높은 경우에는 시스템 전체의 수행 능력을 증가시킬 수 있다. 그러나 뉴스 데이터의 경우 데이터가 단순한 텍스트나 수치 데이터 뿐만 아니라 이미지, 오디오 및 동영상의 데이터가 복합적으로 존재하며 하나의 뉴스 항목에 대한 데이터의 크기가 크고 항목 자체의 수도 다양하며 특정 시간대에 집중적으로 몰리는 경향이 있다. 또한 데이터의 국부성도 비디오 데이터 보다는 높지만 사무계산용 데이터의 국부성 보다는 낮다. 따라서 뉴스 데이터를 기존의 버퍼 캐쉬를 통하여 운영한다면 빈번한 메모리 복사에 의한 오버헤드가 증가할 가능성이 크다. 디스크 운영 측면에서 보면 재래식 데이터의 경우 버퍼 캐쉬에서 히트가 되지 않았을 때 디스크 읽기를 하지만 재래식 데이

터는 반드시 실시간으로 접근 되지 않아도 되기 때문에 디스크 큐에서의 어느 정도의 지연이 문제가 되지 않는다

해싱 큐 헤더



[그림 1] 기존 시스템의 버퍼캐쉬

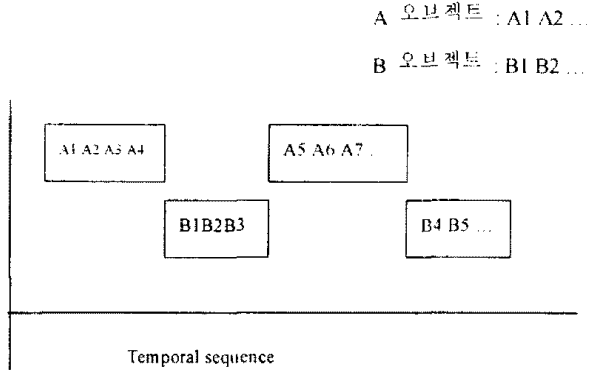
터는 실시간으로 지원이 되어야 하기 때문에 버퍼 캐쉬에서 데이터를 찾지 못할 경우 디스크 읽기를 해야 하지만 만약 그때 디스크에 가용할 만한 대역폭이 존재하지 않으면 데이터 읽기에 지연이 생기게 된다. 따라서 사용자 측면에서 보면 정상적인 데이터를 받아보지 못할 가능성이 있다.

3.2 뉴스 데이터를 고려한 새로운 버퍼캐쉬 운영 기법

앞에서 언급하였듯이 2005년에 이르면 인터넷을 통하여 접근되는 데이터 중 절반 가량이 멀티미디어형의 데이터가 된다[97 Tew]. 뉴스 데이터는 텍스트 뿐만 아니라 비디오, 오디오 등의 실시간 성격을 지닌 데이터들도 포함하고 있으므로 버퍼 캐쉬의 운영 방식도 뉴스 데이터의 성격을 고려하여 새로운 버퍼 캐쉬 운영 방법이 고안되어야 한다.

동영상 뉴스 데이터의 경우 한 개의 오브젝트가 차지하는 블록의 수는 멀티미디어 데이터의 최적 블록 사이즈로 알려진 256 키로 바이트를 블록 단위로 계산하면 2분 짜리 MPEG II 뉴스 데이터는

120 Mbytes/256Kbytes = 480 개의 블록이 된다. 따라서 블록 단위로 접근을 하게되면 그림 2와 같이 디스크에서 버퍼로 읽어오는 순서에 의해 한 오브젝트의 중간 블록이 버퍼 재할당에 의해 다른 오브젝트로 넘어가는 경우가 생긴다.



[그림 2] 시간에 따른 버퍼 읽기의 예

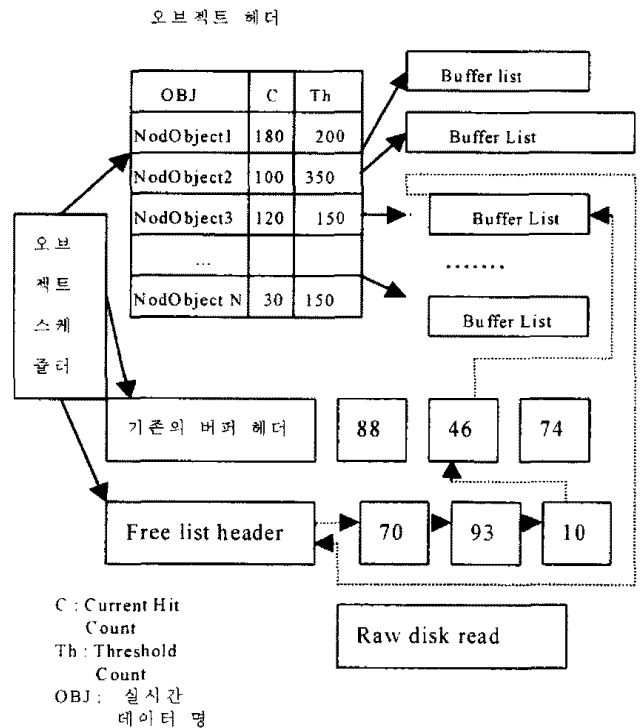
즉, 동영상 뉴스 데이터 A(A1 A2 A3...) 와 B(B1 B2 B3 ...)가 버퍼 캐쉬에 그림 2와 같은 시간 순서에 의해 디스크로부터 읽혀 들어와 있다고 하자. 만약 새로운 데이터에 대한 요청이 있어 디스크로부터 해당 데이터를 읽어서 버퍼링할 자유 리스트의 버퍼가 부족하여 이미 할당 되어있는 버퍼를 사용해야 한다면 기존 LRU 방식에 의해 B4, B5, A5, A6, A7이 새로운 데이터에 할당된다. 이 때 데이터 A를 원하는 요청이 있을 경우 A1, A2, A3은 아직 버퍼 캐쉬에 존재하므로 버퍼 캐쉬로부터 데이터를 읽지만 A4, A5, A6의 경우는 다른 데이터에 할당되었으므로 해당 블록을 읽어오기 위하여 디스크 읽기를 해야 한다. 이때 디스크에 가용 대역폭이 있는 경우에는 문제가 되지 않지만 없는 경우에는 필연적으로 읽기 지연이 발생하여 정상적인 데이터의 읽기가 불가능하게 된다. 따라서 실시간 뉴스 데이터의 경우에는 버퍼 캐쉬의 자유 리스트로 블록을 되돌려 줄때 블록 단위로 되돌려 주지 않고 오브젝트 단위로 되돌려 주는 것이 버퍼 캐쉬 운영에 효율적이다.

이러한 점을 고려할 때 뉴스 데이터를 실시간 데이터와 비 실시간 데이터로 분류하여 실시간 응답을 요구하는 데이터의 경우에는 기본적으로 단순

디스크 읽기를 통하여 지원하는 것이 디스크 스케줄링 관점에서도 유리하다. 그러나 뉴스 데이터의 경우 데이터의 재사용율을 보면 사무계산용 트랜잭션의 경우 보다는 떨어지지만 비디오 데이터의 경우 보다는 높은 수준을 유지한다.

인터넷 신문 사용자 접근 유형 조사를 참조하면 전체 뉴스 데이터의 40%를 차지하는 정치,경제, 스포츠, 문화 데이터가 전체 접근 횟수의 70%를 차지한다[97 이]. 따라서 그림 3처럼 사용자 접근 패턴을 미리 조사하여 이를 근거로 접근 가능성이 높은 데이터의 경우 미리 프리팹칭을 통하여 버퍼 리스트에 저장한 다음, 사용자의 요청에 따라 프리팹칭 되어 있는 데이터의 경우 버퍼 리스트에서 읽어오며 프리팹칭 되지 않은 데이터의 경우 디스크에서 읽기를 한다.

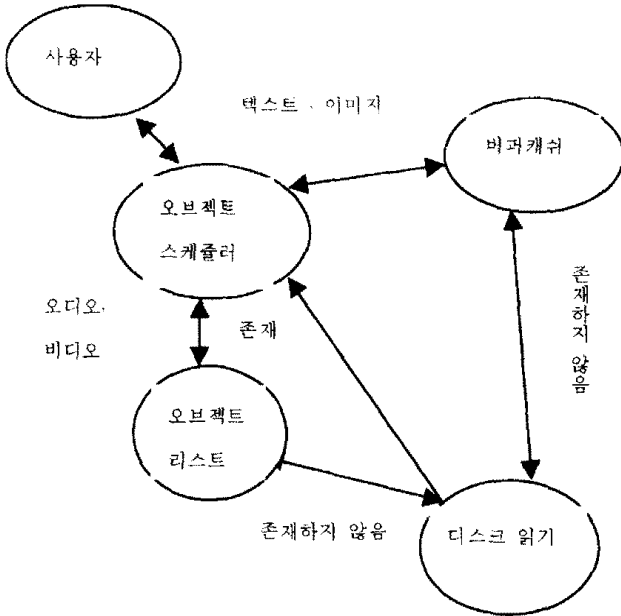
그림 3의 자료구조에서 C는 현재 뉴스 데이터 오브젝트 접근 횟수를 나타내며, Th는 접근 횟수의 Threshold로써 C가 Th를 초과하면 리스트를 자유 리스트로 반납한다.



[그림 3] 멀티미디어 데이터의 버퍼 캐쉬

제안하는 버퍼 캐쉬 운영 방안에서는 비 실시간의 데이터인 텍스트 기사나 이미지 데이터의 경우

는 기존의 버퍼 캐쉬 할당 알고리즘을 따른다. 그림 4에 그림 3의 뉴스 데이터에 적합하도록 재구성된 버퍼 캐쉬 알고리즘을 이용한 데이터의 흐름을 개략적으로 표현 하였다. 그림 5는 새로운 버퍼 관리 알고리즘을 개략적으로 표현한 것이다.



[그림 4] 새로운 버퍼 캐쉬를 이용한 데이터의 흐름

```

Do While until no more user
    receive user request
    if user request is for non real time data
        go to conventional buffer cache to process
    else go to object_list
End do while

Function object_list
    If the data is in the object list
        Increase C by and compare it to Th to check
        whether it can be candidate for buffer stealing
    else
        go to the disks for read and transfer data to the
        user directly with out buffering the data
    [그림 5] 새로운 버퍼 캐쉬 알고리즘
  
```

[그림 5] 새로운 버퍼 캐쉬 알고리즘

4. 시뮬레이션

4.1 디스크 수의 계산

비디오나 오디오 데이터와 같은 경우 실시간 처리 이므로 데이터의 저장 방법에 따라 전체 처리율이 영향을 받는다.

일반적으로 데이터를 저장할 디스크의 수를 결정할 때 두 가지 모드로 나누어진다[95 Tew]. 그 첫째가 디스크 암 바운드(disk arm bound)로 데이터를 저장할 디스크의 수를 정할 때 동시에 디스크를 접근하는 사용자를 지원하는데 필요한 대역폭을 기준으로 계산하는 방식으로써 실시간으로 데이터를 처리하기 위해서는 이 방식이 적합하다. 예를 들면 지원해야 할 총 사용자 수가 N_{user} 이면 이때 요구되는 디스크 장치의 숫자 N_{disk} 는 아래와 같이 계산할 수 있다.

$$N_{disk} = \lceil N_{user} * P_{rate} / D_{bd} \rceil \quad [식 1]$$

D_{bd} : 디스크의 대역폭, P_{rate} : play back rate

두 번째가 디스크 공간 바운드 방식으로 저장에 필요한 디스크의 수를 결정할 때 저장할 데이터의 양으로 결정하는 방법으로써 아래와 같이 계산된다.

$$N_{disk} = \lceil T_{data} / D_{space} \rceil \quad [식 2]$$

T_{data} : 총 데이터량, D_{space} : 디스크 하나의 저장 공간

뉴스 데이터의 경우 실시간 데이터와 비실시간 데이터가 복합적으로 나타나기 때문에 뉴스 데이터의 저장과 접근에 필요한 디스크의 수는 1)과 2) 식을 합친 식으로 표현 할 수 있다.

$$N_{disk} = \lceil N_{user} * P_{rate} / D_{bd} \rceil + \lceil T_{data} / D_{space} \rceil \quad [식 3]$$

만약 버퍼 캐쉬를 이용하여 접근이 예상되는 비디오, 오디오 오브젝트에 대하여 프리팹칭을 하여 오브젝트 리스트를 운영한다면 필요한 디스크의 수는 다음과 같다.

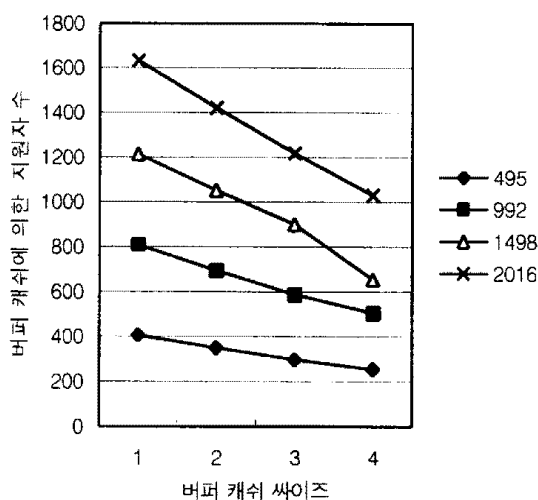
$$N_{disk} = (N_{user} * P_{rate} / D_{obj}) + [T_{rate} / D_{space}] * (1 - p) \quad [식 4]$$

p : 원하는 오브젝트가 오브젝트 리스트에 존재 할 확률

(0 ≤ p ≤ 1, p는 접근 예상치가 높거나 미디어 버퍼 메모리 사이즈를 크게 할수록 높아진다.)

4.2 오브젝트 리스트를 이용한 시뮬레이션

본 연구에서 제안하는 알고리즘의 평가를 위하여 유닉스 시스템 환경에서 C언어로 프로그램을 작성하여 시뮬레이션을 하였다. 요청 분포는 실시간 데이터의 측정에 널리 사용되는 지수 분포 함수를 사용하였다. 측정 단위는 디스크 수가 아닌 요청 건수로 하였으며 단순 디스크 읽기를 하였을 때의 전체 디스크 요청 건수와 본 알고리즘을 사용하였을 때의 디스크 접근 건수를 비교하였다. 그림 6은 동시 NOD 데이터 요청 건수를 500부터 2,000까지 그리고 오브젝트 리스트의 크기를 112.5 MB에서 450 MB까지 변화시키면서 총 사용자 중에서 디스크 읽기를 하지 않고 오브젝트 리스트의 데이터만으로 지원할 수 있는 요청 건수를 측정하였다. 오브젝트 리스트에 프리팹칭할 데이터는 [97 이]의 조사에 따라 결정하였다.



[그림 6] 오브젝트 리스트를 사용한 디스크 접근 수의 변화

결과에 따르면 오브젝트 리스트를 운영함으로써 전체 요청의 약 30% - 35% 정도가 디스크 읽기를 하지 않고 오브젝트 리스트의 데이터를 통해서 지원됨을 알 수 있다.

5. 결론 및 향후 연구 방향

뉴스 데이터는 실시간 데이터와 비 실시간 데이터가 혼합된 형태로 존재하며 VOD 데이터에 비하여 높은 시간적, 공간적 국부성이 있다. 또한 실시간 뉴스 데이터의 경우 고 대역폭의 디스크 전송 속도가 필요하므로 디스크 접근에 의한 병목 현상을 해결하고 디스크의 접근 효율을 높이기 위해서는 버퍼캐시를 이용한 데이터의 재사용과 디스크 접근을 최소화 하여야 한다. 그러나 기존의 유닉스 시스템의 버퍼 캐시 알고리즘은 실시간 데이터의 지원에는 문제가 있다. 따라서 본 연구에서는 실시간 유형의 뉴스 데이터를 지원하기 위한 새로운 버퍼 캐시 알고리즘을 제안하고 새로운 알고리즘을 사용함으로써 디스크 교통량의 30% 정도를 줄일 수 있음을 시뮬레이션을 통하여 밝혔다. 향후 연구로는 본 연구에서는 오브젝트 리스트에 프리팹칭할 데이터를 이전의 로그 데이터를 분석하여 정적으로 로드한 것이다. 하지만 이러한 방법은 동적으로 뉴스 기사의 선호도가 변할 수 있는 상황에서는 최적일 수는 없다. 따라서 향후 프리팹칭 데이터를 결정함에 있어서 정적인 프리팹칭 뿐만 아니라 동적으로 변화하는 사용자 패턴을 분석하여 반영할 수 있는 전략에 관하여 연구하는 것이 중요하다.

[참고 문헌]

- [97 이] 이주경, 박용운, 김영주, 정기동, "NOD 데이터를 위한 데이터 배치 방법", 한국 정보 과학회 추계 학술 발표회, 1997
- [97 Tew] Renu Tewari, Harrick M. Vin, Asit Dan & Dinkar Sitaram, "Resource Based Caching for Web Servers", Tech Report of CS of U. of Texas at Austin, 1997

[97 Wei] Weifeng Shi et al, "Trading memory for disk bandwidth in Video on Demand", Tech report, Dept. of CS at USC, 1997

[95 Tew] Renu Tewari, Asit Dan, et al "Buffering and Caching in Large-Scale Video Servers", Tech-Reports . Texas Austin, 1995

[95 Kun] Kun Lung Wu, Philip S Yu, " Consumption based Buffer management for Maximizing System Throughputs of News On Demand Multimedia system", 1995