

동영상 정보 검색 시스템에서 버퍼 캐시의 효율성 연구

이강희⁰, 전주탁, 류연승
한림대학교 정보통신공학부

{leekh⁰, hong4uu}@center.cie.hallym.ac.kr, ysryu@hallym.ac.kr

Study on Efficiency of Buffer Cache for Video Information Search System

Kang-Hee Lee⁰, Joo-Tak Jeon, Yeon-Seung Ryu
Div. of Information and Communication Eng. Hallym University

요 약

동영상 정보 검색 시스템은 비교적 작은 크기의 동영상 클립과 클립을 인덱싱하기 위한 키 프레임으로 구성된다. 본 논문에서는 동영상 정보 검색 시스템을 위한 버퍼 캐시에서 버퍼 교체 기법을 연구하였고, 버퍼 캐시 사용의 효율성을 연구하였다. 실험을 통해 버퍼 캐시가 좋은 성능을 가지려면 적은 수의 동영상 클립에 요청이 편중되어야 함을 알 수 있었다.

1. 서 론

1.1 배경

최근 정보통신의 발전으로 유무선 인터넷에서 실시간으로 동영상 파일을 볼 수 있게 되었으며 유무선 단말기를 사용하여 동영상 정보를 검색하고 살펴보는 동영상 정보 검색 시스템과 같은 응용이 증가할 것으로 예측되고 있다.

동영상 정보 검색 시스템은 동영상 정보를 저장하는 저장 시스템(storage system)을 가지며, 동영상 정보를 검색할 수 있는 사용자 인터페이스를 제공한다[7,9]. 일반적으로 동영상 검색은 키워드를 사용하거나 특정 객체(object)의 모양(shape), 색상(color), 크기 등의 특징(feature)을 사용하여 객체를 포함하고 있는 동영상을 검색한다. 사용자가 객체의 특징이나 키워드를 입력하면 검색 시스템은 객체를 포함하고 있는 동영상의 키 프레임(또는 대표 프레임)을 찾게 된다. 찾아진 키 프레임 중에서 하나를 선택하면 해당 동영상 정보를 상영하여 볼 수 있게 된다. 그림 1은 일반적인 동영상 정보 검색 시스템의 기능도이다.

대규모 동영상 정보 검색 시스템은 저장된 동영상의 수 및 지원하는 동시 사용자 수가 매우 많은 대용량 저장 시스템을 가진다. 저장 시스템에서 동영상 데이터는 디스크에 저장되어 있으며 참조되는 데이터는 디스크에서 읽혀져 버퍼에 저장된 후 사용자에게 전송된다. 버퍼 캐시는 참조되는 데이터를 버퍼에 저장하고 이후에 그 데이터가 재참조될 때 디스크가 아닌 버퍼에서 서비스하게 하는 방법이다. 버퍼 캐시는 디스크의 입출력을 감소 시킴으로써 데이터 입출력 요청에 대한 응답시간을 줄여 시스템 성능을 좋게 한다. 그러나, 버퍼 캐시는 가격이 비싸서 일반적으로 그 용량이 디스크 용량보다 매우 작다. 따라서, 새로이 참조되는 데이터에 의해 버퍼에 저장된 데이터가 교체되며 교체 알고리즘에 따라 버퍼 캐시

의 성능이 좌우된다. 일반적으로 버퍼 캐시는 데이터 참조의 국지성(locality of reference)이 높을 때 좋은 효율을 보인다

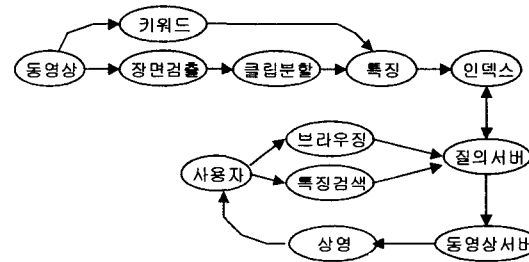


그림 1 동영상 정보 검색 시스템의 예

본 연구의 목적은 비교적 작은 크기의 동영상 객체들을 버퍼에 캐시함으로써 시스템 성능을 개선하는 방안의 효율성을 연구하는 것이다.

1.2 기존연구

지금까지 동영상 저장 시스템을 위한 많은 버퍼 관리 알고리즘들이 연구되어 왔다[1-4]. 그러나, 대부분은 영화처럼 오랜 시간동안(1시간 이상) 동영상 데이터를 상영하는 VOD(Video-on Demand) 시스템을 가정한 연구들이었다[1,2]. VOD 시스템의 경우 인터벌(interval) 기법이 매우 좋은 효율성을 보인다. 대표적으로는 인터벌 캐싱(interval caching)[2]과 distance[1] 알고리즘이 있다. 인터벌이란 같은 동영상 파일을 참조하면서 연속된 두 스트림 간의 데이터 블록으로 정의된다. 인터벌 캐싱[2]은 선행 스트림이 디스크에서 읽은 데이터를 버퍼에 유지시켜 후행 스트림이 사용하게 하는 기법이다. 연속적인 두 스트림 간의 인터벌이 캐시되어 있다면 후행 스트림은 선행 스트림이 읽은 데이터를 버퍼에서 읽게 되므

로 디스크 입출력을 줄일 수 있게 된다. 인터벌 캐싱은 작은 인터벌을 먼저 캐시하고 버퍼 공간이 부족하다면 인터벌이 가장 긴 것을 캐시에서 제거하는 교체기법을 사용한다.

인터벌 기법은 많은 수의 스트림이 같은 동영상 파일을 오랜 시간 동안 순차적으로 참조할 때 효율성이 우수하다. 그러나, 빠른 되감기, 점프 등의 VCR 연산이 많은 경우에는 성능이 떨어지게 된다. 최근에는 동영상 파일에 대해 다양한 참조 유형을 고려한 버퍼 캐시 기법에 대한 연구 결과가 발표되고 있다[5]. 한편, 범용 파일 시스템에서 LRU와 LFU를 결합한 버퍼 교체 기법도 제안되었다[6]. 본 연구에서는 비교적 짧은 동영상 클립을 서비스하는 시스템에서의 버퍼 캐시 기법의 효율성을 연구하고자 한다.

2. 시스템 모델

본 버퍼 캐시 연구에서 가정하는 동영상 정보 검색 시스템은 동영상 파일과 키 프레임(key frame)으로 구성된다. 동영상 정보 검색 시스템의 사용자는 동영상 파일을 검색하기 위해 우선 키 프레임을 구해야 한다. 키 프레임의 검색 방법은 키워드, 객체의 특징(모양, 색상 등)을 사용자가 입력하는 등의 다양한 방법이 있을 수 있다. 찾아진 키 프레임 중에서 하나를 선택하면 대응하는 동영상을 상영하여 볼 수 있다. 동영상 파일은 장면 검출(scene detection) 기법을 사용하여 작은 동영상 클립(clip)들로 분할되고 장면 검출에서 추출된 키 프레임을 사용하여 동영상 클립들을 인덱싱한다. 동영상에서 장면을 대표하는 키 프레임을 추출하기 위한 다양한 장면 검출 기법들이 연구되어 왔다[8-10].

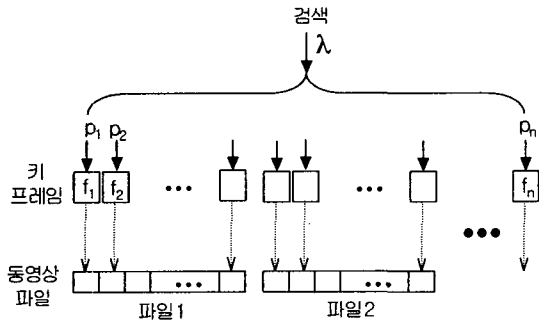


그림 2. 시스템 모델

그림 2는 본 연구의 시스템 모델을 보여주고 있다. 시스템에 저장된 동영상 파일의 개수가 M 이고 키 프레임의 개수가 n 개라고 가정한다. 시스템 전체의 검색 요청의 도착률을 λ (요청수/초)라 하자. 검색 요청 중에서 키 프레임 f_i 를 검색할 확률을 p_i 라고 하자. 이때 키 프레임 f_i 의 검색 도착률은 λp_i 이다.

3. 버퍼 캐시의 연구

3.1 워크로드 모델링

버퍼 캐시의 성능을 조사하기 위해서는 워크로드 모델링이 선행되어야 한다. 여기에는 사용자 검색 요청의 도착률과 키 프레임 선택의 분포를 결정해야 한다. 그러나, 이를 위한 실제(real) 워크로드를 구하기 어렵기 때문에 일반적으로 적용되고 있는 분포함수를 사용한다. 검색 요청의 도착간 시간 간격의 분포는 (식 1)과 같은 지수 분포(exponential distribution)를 사용한다.

$$f(x) = \frac{1}{\lambda} e^{-x/\lambda} \quad (1)$$

요청의 도착률 λ 이 클수록 짧은 시간에 요청이 집중됨을 의미한다. 키 프레임의 선택 확률은 (식 2)와 같은 기하 분포(geometric distribution)를 사용한다. p_i 은 n 개의 키 프레임에서 키 프레임 f_i 를 요청할 확률이다.

$$p_i = \frac{(1-\alpha)\alpha^i}{(1-\alpha^n)} \quad (2)$$

(식 2)의 기하분포에서 파라미터 α 값을 변화시키면 요청의 편중(skew)도를 조절할 수 있다. 그림 3은 $\alpha=0.9, 0.99, 0.999$ 인 경우의 분포를 보여주고 있다. $\alpha=0.9$ 인 경우 소수 키 프레임에 검색 요청이 집중하며, $\alpha=0.999$ 인 경우 모든 키 프레임에 균등하게(uniform) 검색 요청이 일어난다.

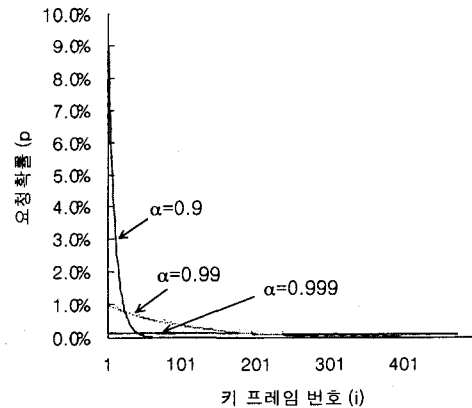


그림 3 기하분포에서 파라미터 별 분포도

3.2 버퍼 교체 정책

대표적인 버퍼 교체 정책인 LFU, LRU, 인터벌 캐싱 알고리즘을 고려할 수 있다. LFU는 데이터의 참조 횟수가 적은 것을 교체하고, LRU는 가장 오래동안 참조되지 않은 것을 교체하며, 인터벌 캐싱은 두 스트림 간의 인터벌이 작은 것을 교체한다. 본 연구에서는 동영상 클립의 길이를 10초로 가정하고 있으므로 같은 클립에 대해 10초 내에 많은 참조가 발생하는 경우에 효율적이다. 인터벌 캐싱의 경우 한 클립에 10초 내로 요청이 있어야

인터벌이 형성된다. 따라서, 인터벌 캐싱이 효과적이라면 비교적 짧은 기간 내에 동일한 클립에 요청이 집중되어야 한다.

교체되기로 결정된 클립을 교체할 때, 클립 단위 교체 방법과 블록 단위 교체 방법이 있다. 클립 단위 교체 방법은 클립 내의 모든 블록을 교체하는 방법이며, 블록 단위의 교체 방법은 클립 내의 블록 단위로 교체하는 방법이다. 클립 단위 교체 방법은 클립이 참조되는 동안 버퍼에 있는 클립 내의 블록이 참조되지 않더라도 교체될 수 없다. 클립의 길이가 길다면(예, 20-30분) 클립 단위 교체 방법보다 클립의 블록 단위로 교체하는 것이 버퍼 성능(버퍼 사용률을 높임으로써 캐시 적중률이 좋아짐)이 더 좋을 것이다. 블록 단위 교체 방법에서, 클립 내의 블록은 순차적으로 접근되는 특성을 이용하면 클립 내의 블록의 교체 정책은 MRU가 적합하다.

본 연구에서는 클립의 길이가 비교적 짧기 때문에(10초), 클립 단위의 LRU 교체 방법의 효율성을 살펴본다.

3.3 실험

각 동영상의 상영률은 1.5 Mbps로 가정하였다. 또한 동영상 파일은 데이터 블록으로 저장되며 각 블록의 크기는 200 KB (1초 상영분의 데이터)를 가정하였다. 동영상 클립은 10,000개이고, 각 동영상 클립의 길이는 10초로 고정값을 가정하였다. 즉, 동영상 클립의 데이터 크기는 2 MB가 된다. 버퍼의 크기는 60 MB(30 클립)이다.

검색 요청의 도착 간격($1/\lambda$)은 10초, 30초, 50초, 100초를 사용하였고 키 프레임의 선택 확률은 기하분포를 사용하고 파라미터 α 는 0.9, 0.99, 0.999를 사용하였다.

그림 4는 $1/\lambda$ 과 α 의 변화에 대한 캐시 적중률(hit ratio)을 보여주고 있다. 동영상 클립에 대한 편중도(skew)가 클 때는 버퍼 적중률이 높았고, 편중도가 작을 때는 버퍼 적중률이 좋지 않음을 알 수 있다. 따라서, 클립 단위의 LRU 교체 방법을 사용할 때 동영상 정보에 대한 편중도에 따라 성능이 좌우됨을 알 수 있었다.

4. 결 론

초고속 유무선 인터넷이 통합되고 있으며 대규모의 동시 사용자들이 동영상 정보를 검색할 수 있는 동영상 저장 시스템 응용 분야가 등장하고 있다. 동영상 정보 검색 시스템은 비교적 작은 크기의 동영상 클립과 클립을 인덱싱하기 위한 키 프레임을 저장하고 있다.

본 논문에서는 대규모의 동영상 정보 검색 시스템을 위한 버퍼 캐시에서 버퍼 교체 기법을 연구하였고, 버퍼 캐시 사용의 효율성을 연구하였다. 실험을 통해 버퍼 캐시가 좋은 성능을 가지려면 적은 수의 동영상 클립에 요청이 편중되어야 함을 알 수 있었다. 본 논문에서는 기존의 교체 기법에서 버퍼 캐시의 효율성을 알아보았으며 향후에는 효율성을 높일 수 있는 버퍼 교체 기법에 대한 연구를 진행할 계획이다.

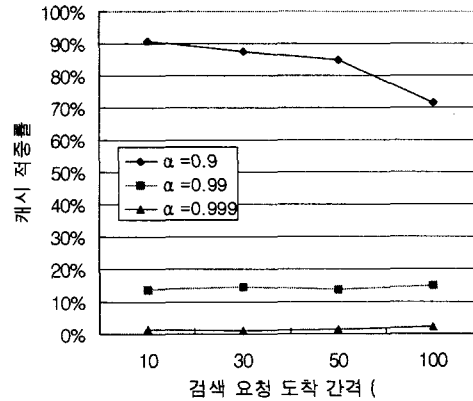


그림 4 실험 결과

5. 참고문헌

- [1] B. Özden, R. Rastogi, and A. Silberschatz, "Buffer Replacement Algorithms for Multimedia Storage Systems," Proc of IEEE International Conference on Multimedia Computing and Systems, Jun. 1996.
- [2] A. Dan and D. Sitram, "Buffer Management Policy for an On-Demand Video Server," IBM Research Report, RC 19347, Yorktown Heights, NY. 1993.
- [3] K. Wu and P. S. Yu, "Consumption-Based Buffer Management for Maximizing System Throughput of a Multimedia System," Proc. of IEEE International Conf. on Multimedia Computing and Systems, Jun. 1996.
- [4] T. Raymond, and Y. Jinhai, "An Analysis of Buffer Sharing and Prefetching Techniques for Multimedia Systems," Multimedia Systems, pp. 55-69, Jun. 1996.
- [5] Y. Ryu, K. Cho, Y. Won and K. Koh, "Intelligent Buffer Cache Management in Multimedia Data Retrieval," Lecture Notes in Artificial Intelligence, Springer-Verlag, Vol. 2366, pp. 462-471, Jun. 2002.
- [6] D. Lee, J. Choi, J. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. Kim, "LRFU: A Spectrum of Policies that Subsumes the LRU and LFU Policies," IEEE Transactions on Computers, Vol. 50, No.12, pp.1352-1361, 2001.
- [7] C. Taskiran, C. Bouman, and E. J. Delp, "The ViBE Video Database System: An update and further studies," Proc. of the SPIE/IS&T Conference on Storage and Retrieval for Media Databases, Jan. 2000.
- [8] C. Ngo, T. Pong and H. Zhang, "On clustering and retrieval of video shots," ACM Multimedia, 2001.
- [9] Hanjalic, A., Lagendijk, IL. and Biemond, J. "Automated High-level Movie Segmentation for Advanced Video Retrieval System," IEEE Trans. on Circuits and System for Video Technology, pp. 580-588, 1999.
- [10] J. Gauch, S. Gauch, S. Bouix and X. Zhu, "Real Time Video Scene Detection and Classification," Information Processing and Management, 35/3, 1999.