

서로 다른 분산 비디오 저장 서버를 위한 데이터 할당과 부하 조정

송상우^o 정병수
경희대학교 전산학과

swsong@jupiter.kyunghee.ac.kr jeong@nms.kyunghee.ac.kr

Data Allocation and Load Balancing for Different Distributed Video Server

Sang-Woo Song^o Byung-Soo Jeong
Dept. of Computer Science, Kyunghee University

요 약

비디오 저장 서버를 설계하는데 가장 주안을 두어야 할 점은 서버의 이용률을 증가시키고 그에 따른 비용을 최소화 해야 한다는 점이다. 이를 위한 서버로 분산 비디오 저장 서버가 주로 이용되고 있고, 그 목적으로는 균등한 비디오 할당과 부하 조정(load balancing)이 제안되어 왔다. 이에 대한 방안의 하나로 디스크 기반의 비디오 저장 서버 클러스터에서 부하 시프팅(load shifting)이 그 목적을 이루기 위한 한가지 방법으로 연구되었다. 이것은 동일한 용량의 서버로 구성된 비디오 서버 클러스터에서 비디오를 다른 서버에 복사해 놓고, 사용되고 있는 비디오 요구시 그 비디오에 대한 요구를 다른 서버의 복사본으로 이동시킴으로써 서버들에 대해 균등한 비디오 할당을 하고 서버나 비디오의 추가와 같은 동적인 변화에 대해 데이터 할당 및 부하 조정을 감안하여 제안된 것이다. 이 논문에서는 기존의 동일한 서버만을 사용한 것과는 달리 용량이 서로 다른 서버나 비디오를 추가할 경우에 요구되는 최적의 비디오 데이터 할당과 부하 조정에 관한 전략에 관하여 설명한다.

1. 서론

최근에 컴퓨터와 통신 기술의 발전은 분산 멀티미디어 어플리케이션이 비디오 저장 서버에 중요한 역할을 할 수 있게 하였다. 비디오 서버를 설계하는데 중요한 논쟁점 중의 하나는 확장성 문제이다[1]. 비디오 서버들이 현재의 모든 사용자 요구를 처리할 만한 충분한 능력을 가졌을지라도 요구되는 비디오를 갖는 컴퓨터들이 그 요구를 처리할만한 능력을 이용할 수 없는 경우에는 몇몇의 사용자 요구는 처리되지 못할 수 있다. 이런 저급 컴퓨터들의 제한된 저장 공간과 능력 때문에 비디오 파일의 적당한 할당과 그들 사이의 사용자 요구를 균형화 시키는 것은 매우 중요하다. 사용자 행동을 오랫동안 관찰한 통계학적 결과를 기대 요구 패턴이라 하고 비디오 할당의 기초로 사용한다. 그러나 비디오의 사용자 요구는 본래 동적이기 때문에 실제 요구 패턴은 기대 요구 패턴과는 다르다.

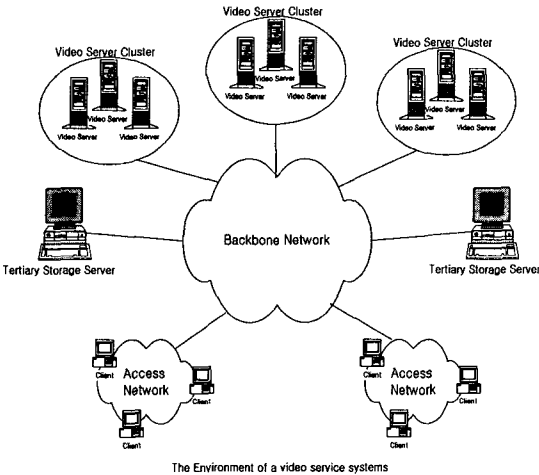
데이터 할당의 첫 번째 가능한 해법은 비디오 파일을 모든 서버에 stripe 하는 것이다[2,3]. 이 연구의 이점은 사용 가능한 서버가 사용자 요구들을 지원할 수 있기 때문에 최대의 부하 조정을 얻을 수 있다는 것이다. 그러나 이 방법은 서버 하나의 손실이 전체 시스템에 손실을 가져오기 때문에 바람직하지 못하다는 단점을 가지고 있다.

또한 시스템 동기화 오버헤드와 복잡한 시스템 제어 매커니즘이라는 단점이 있다. 이에 대한 해결 방법으로 서버 손상시의 대처 능력과 부하 조정을 목적으로 비디오를 몇몇의 서버에 복사하여 사용자의 요구를 다른 서버로 이동시키는 전략이 제안되었다. 본 논문에서는 분산 비디오 저장 서버에서, 동일한 용량의 서버에 대한 데이터 할당과 부하 조정을 제안한 기존의 방법과는 달리, 서로 다른 용량의 새로운 서버나 비디오를 추가할 경우에 필요한 최적의 데이터 할당과 부하 조정을 얻는 정책에 관해 연구하였다.

2. 관련연구

지금까지 비디오 데이터를 복사하여 사용하는데 필요한 기본적인 정책들이 제안되어 왔다. Serpanos et al.[4]은 실제 요구가 기대 요구와 동일하다는 가정하에서 정적인 부하 조정을 얻기 위한 초기 할당에 관해 제안했지만 기대되는 부하는 단기간의 요구 패턴을 나타내지 않는다는 점과 비디오 액세스 확률이 정확히 예상되지 않는다는 모순을 보였다. Wolf et al.은 디스크를 노드로 두 디스크 사이의 한 쌍의 에지를 비디오 복사본으로 표현하는 그래프를 모델링하여 멀티디스크 서버의 사용자 요구를 균형화시키는 DASD dancing algorithm 을 제안했지만 각각의 비디오 복사본이 같아야 하고 시프팅 단계의 수의 감소를 고려하지 않는다는 단점이 있다. [7]에서는 분산 비디오 서버에 비디오를 복사하여 사용자

요구 도착시 부하 시프팅을 사용하여 요구를 이동시킴으로써 최적의 데이터 할당과 부하 조절을 얻는 기법에 관해 소개하였다. 그러나 이 연구는 동일한 용량의 서버상에 적용하였고 서버를 추가하는 것도 동일한 용량의 서버만 가능하다. 분산 비디오 서버의 구조는 [그림1]과 같다. tertiary storage server 에는 모든 비디오들이 저장되고, 인기도가 높은 비디오[6] 들은 cluster video server 에 저장 된다. 클러스터링은 IP 주소나 도메인명으로 정한다.



[그림1] 분산 비디오 서버의 구조

각각의 비디오의 복사본의 수 R_i 는 자원 할당 알고리즘(resource allocation algorithm)[5]에 의해 결정한다. 비디오 복사본의 수를 결정하는데는 조건이 있다. 비디오 복사본의 수는 정수여야 하고 서버의 수보다는 같거나 작아야 한다는 조건이 있다.

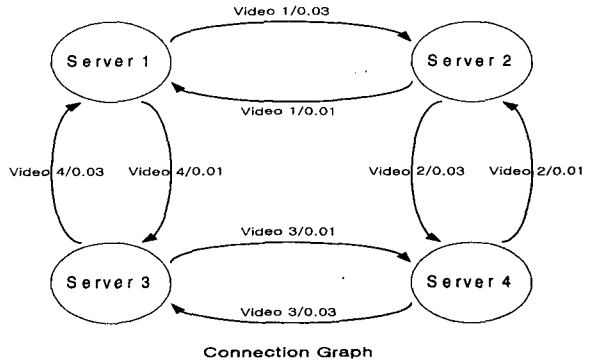
$$R_i = \frac{Pr_i}{\sum_{j=1}^M Pr_j} \cdot c \cdot s$$

여기서 R_i 는 비디오 복사본의 수이고 Pr_i 는 캐쉬된 비디오의 액세스 확률, K 는 캐쉬된 비디오의 수, S 는 서버들의 수, C 는 비디오의 디스크 저장 용량이다. 각각의 비디오 복사본들의 액세스 확률은 액세스 확률 할당 알고리즘(access probability allocation algorithm)에 의해서 구하고 모든 복사본들의 액세스 확률들의 합은 비디오의 액세스 확률과 같다.

$$Pr_i = \sum_{j=1}^{R_i} Pr_i^j$$

초기 데이터 할당은 연결 그래프라고 불리는 방향성 그래프에 의해 표현되고, [그림2]와 같이 나타낸다. 여기서 서버는 그래프의 노드이고 두 서버 사이의 에지는 양쪽 서버에 저장된 비디오 복사본들을 가리킨다. 각각의 에지는 비디오 복사본의 액세스 확률로 정의할 수 있는 가중치를 갖는다.

양쪽의 비디오 복사본들이 서로 다른 액세스 확률을 가진다면 그 에지는 각기 다른 가중치를 갖게 된다. 다시 말해서 에지의 가중치는 노드의 바깥쪽으로부터 안쪽으로 사용자 요구가 시프트 될 확률을 의미하는 것이다.



[그림2] Connection Graph

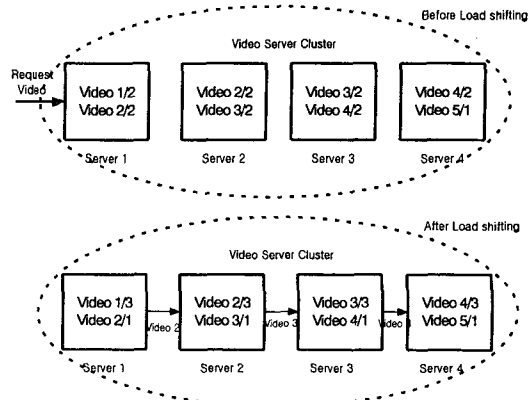
서버에 있는 모든 비디오 복사본들이 액세스 될 확률을 connectivity 라고 하며, 1번째 서버의 connectivity를 C_1 로 표시하고 connectivity 클수록 그 서버의 비디오 액세스 확률은 커진다. 이를 구하는 식은 다음과 같다.

$$C_i = \sum_{l=1}^M \sum_{j=1}^S V_{i,l} V_{l,i} Pr_i^{n(i,l)}$$

M 은 시스템의 비디오의 수, $n(i,l)$ 는 1번째 서버의 i 번째 비디오의 복사본을 나타내는 매핑 함수이다. 모든 서버의 connectivity가 비슷할 때 즉 $C_1 \approx C_2 \approx C_3 \approx \dots \approx C_S$ 일 경우 최상의 부하 조절을 얻을 수 있다 실제로 $C_1 \cdot C_2 \cdot C_3 \cdot \dots \cdot C_S$ 의 경우에 최대값이 나오게 되고 최적의 부하 조절을 얻게 된다.

3. 부하 시프팅(Load Shifting) 기법

부하 시프팅의 목적은 실제 액세스 확률과 기대 액세스 확률 사이의 차이를 조절하는 것이다. [그림3]는 이 기법을 그림으로 표시한 것이다.



[그림3] Load Shifting Scheme

Server 4를 제외한 모든 서버가 사용되고 있고 비디오 1을 원하는 사용자의 요구가 도달한다면 비디오 1은 Server 1에만 존재하므로 부하 시프팅이 발생하게 된다. 서버 1의 비디오 2에 대한 요구는 서버 2로, 서버 2의 비디오 3에 대한 요구는 서버 3으로, 그리고 서버 3의 비디오 4에 대한 요구는 서버 4로 shifting 하게 됨으로써 결국 사용자 요구는 기다릴 필요가 없고 서버들 간에는 부하 조정을 이루게 된다.

4. 초기 할당 기법

시스템에 새로운 서버를 추가하는 경우 비디오를 재할당하는 방법에는 두가지가 있다. 기존 서버의 비디오는 무관하게 추가 시킨 서버나 비디오에 대해서만 재할당을 하는 부분 재할당 정책(partial reallocation policy)과 기존 서버의 비디오들까지 모두 재할당하는 전체 재할당 정책(total reallocation policy)이 있다. 부분 재할당 정책은 서버 전체를 업데이트 하는 것이 아니기 때문에 서버들 상에 최적의 connectivity를 보장할 수는 없다. 이에 반해 서버들의 비디오 전체를 재할당하는 방법은 최적의 connectivity를 얻을 수 있다.

	Server A (storage space =3)	Server B (storage space =3)	Server C (storage space =3)	Server D (storage space =4)	Total access probability	Number of replicas
Original file allocation						
Video1	v	v	v		0.5	4
Video2	v	v	v		0.3	3
Video3	v	v			0.2	2
Video4						
New file allocation by partial reallocation policy						
Video1	v	v	v	v	0.4	5
Video2	v	v	v	v	0.3	4
Video3	v	v		v	0.2	3
Video4				v	0.1	1
New file allocation by total reallocation policy						
Video1	v	v	v	v	0.4	5
Video2	v	v	v	v	0.3	4
Video3	v	v		v	0.2	3
Video4			v		0.1	1

[표1] 새로운 서버(ServerD)와 비디오(Video4) 추가 시 두가지 정책에 의한 파일의 위치

위의 [표1]은 비디오 3개를 저장할 수 있는 저장공간을 가진 3개의 비디오 서버에 비디오 4개를 저장할 수 있는 비디오 서버를 추가할 경우, 부분 재할당 정책과 전체 재할당 정책을 사용하여 비디오들을 서버에 재할당 한 것을 비교하여 표로 나타낸 것이다. 각각의 비디오의 액세스 확률과 비디오 복사본의 수들을 통해 각 서버들의 connectivity를 얻은 후 이를 곱하여 최적의 부하 조정 상태를 구할 수 있다.

- 부분 재할당 정책의 connectivity = $0.9 \cdot 0.9 \cdot 0.7 \cdot 1.0 = 0.567$
- 전체 재할당 정책의 connectivity = $0.9 \cdot 0.9 \cdot 0.8 \cdot 0.9 = 0.5832$

위와 같이 부분 재할당 정책을 사용하여 모든 서버의 connectivity를 구한 후 곱하여 얻은 값은 0.567이고, 전체 재할당 정책을 사용하여 얻은 값은 0.5832이다. 이 결과에서와 같이 전체 재할당 정책을 사용했을 때 최대의 connectivity를 구할 수 있고, 이때가 각 서버의 비디오들이 액세스 될 확률이 비슷해지는 최적의 부하 조정을 얻을 수 있는 상태임을 확인할 수 있다.

5. 결론

본 논문에서는 부하 시프팅을 지원하는 분산 비디오 서버에서 기존의 서버와 용량이 다른 서버나 비디오를 추가할 경우에 요구되는 최적의 비디오 데이터 할당과 부하 조정에 관해 연구하였다. 구체적으로 부분 재할당 정책과 전체 재할당 정책을 통해 각각의 connectivity를 구하였고 이 값들을 비교해 봄으로써 최적의 부하 조정 상태를 확인해 보았다. 소프트웨어의 발전에 못지 않게 하드웨어의 기술도 급속히 발전하고 있는 지금 용량이 다른 분산 서버의 최적화에 대한 정책의 연구는 매우 중요하다. 실제적으로 부분 재할당 정책에 비해 전체 재할당 정책이 가장 최적화된 비디오 데이터 재할당을 할 수 있으나 그에 따른 비용 등이 문제가 되기 때문에 두가지 정책을 상황에 따라 적절히 혼용하여 사용하는 새로운 방안이 제시되어야 한다.

6. 참고문헌

- [1] C.Freedman and D.Dewitt, The spiffi scalable video-on-demand server, in ACM SIGMOD, 1995.
- [2] R.Tewari, D.Dias, R.Mukherjee, and H.Vin, High Availability in Clustered Video Server, Technical Report RC 20108, IBM Research Division, T.J.Watson Research Center, June 1995.
- [3] R.Tewari, R.mukherjee, D.Dias, and H.Vin, Design and performance tradeoffs in clustered video servers, in 3rd IEEE international Conference on Multimedia Computing and Systems, 1996, 144-150.
- [4] D.N.serpanos, L.Georgiadis, and T.Bouloutas, Mnpacking: A Load and Storage Balancing Algorithm for Distributed Multimedia Servers, Technic al Report RC 20410, IBM Research Division, T.J.Watson Research Center, March 1996.
- [5] T.Lbarkai and N.Katoh, Resource Allocation Problems-Algorithmic Approaches, MIT Press, Cambridge, MA, 1988
- [6] Dan.A, D.Sitaram and P.Shahabuddin Scheduling Policies for an On -Demand Video Server with Batching. IBM Research Report, RC 19381, Yorktown Heights, NY, 1993.
- [7] shiao-Li Tsao, Meng Chang Chen, Ming-Tat Ko, Jan-Ming Ho, and Yueh -Min Huang, Data Allocation and Dynamic Load Balancing for Distributed Video Storage Server, Journal of Visual Communication and Image Representation, Vol. 10, No. 2, Jun 1999, pp. 197-218 (doi:10.1006/jvci.1999.0420)