

주문형 비디오 서버에서 비디오 데이터의 인기도에 따른 분산 배치 기법

안 유 정[†] · 원 유 헌^{††}

요 약

주문형 비디오 서버의 검색 성능은 사용자들에게 가장 인기있는 몇몇 비디오 데이터들을 얼마나 빨리 그리고 많은 사용자들에게 서비스해주는가에 달려 있다. 이와 같은 서버의 검색 성능은 비디오 데이터의 배치 정책에 의해 많은 영향을 받는데, 특히 각 비디오 데이터들을 저장하고 있는 디스크들의 다양한 성능과 수 그리고 지백된 배치 방법에 의해 검색 성능은 많은 차이가 날 수 있다. 본 논문에서는 각 비디오 데이터들의 여러 가지 특성을 고려하여 데이터들을 배치시키는데 특히 비디오의 인기도에 따라 분산 정도를 다르게 하여 배치시키도록 한다. 본 논문에서 제안된 정책이 어느 정도 개선된 검색 성능을 내는지 측정하기 위해, 다양한 실험 환경을 구성하고 데이터 배치시 비디오 데이터의 인기도를 적용한 경우와 그렇지 않은 경우의 검색 성능을 측정하여 비교한다.

A Distributive Placement Policy according to Popularity of Video Data in Video-On-Demand Server

You-Jung Ahn[†] · Yoo-Hun Won^{††}

ABSTRACT

A retrieval performance of VOD server is estimated by how quickly it services popular videos to users and how many users it is able to service. Each video data is placed on heterogeneous disks and placement techniques are various, retrieval performance is under the control of these elements, so that a retrieval performance is affected by placement policy. In this paper, we place video data considering their characteristics, especially, we place videos distributively according to their popularity. To verify our policy, we make various environment of experiment, estimate a placement policy using popularity of videos and a contrary policy, and compare them.

1. 서 론

주문형 비디오 서비스는 저장 기술의 발전과 초고속 네트워크의 등장으로 더욱 진보되어 왔다. 주문형 비디오 서버에서는 네트워크를 통해 많은 고객들에게 그들이 원하는 멀티미디어 데이터를 실시간으로 서비스

하므로, 보다 많은 고객의 요청을 처리하기 위해서 서버는 대용량의 메모리와 실시간 서비스를 만족시킬 수 있는 많은 디스크들을 갖는다.

이와 같은 주문형 비디오 서버는 동시에 서비스할 수 있는 고객의 수에 의해 평가받게 되며 이와 같은 서버의 능력은 저장 매체의 접근 능력과 데이터가 저장되어 있는 방식에 의해 좌우된다. 이러한 특성으로 인해 주문형 비디오 서버는 기존의 범용 파일서버와는 다른 새로운 형태의 서버 구성이 요구된다. 일반적으로

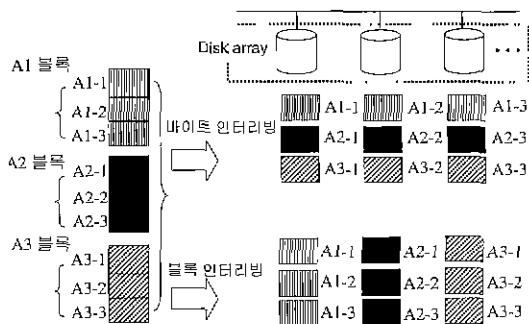
[†] 준 회 원 홍익대학교 대학원 컴퓨터공학과
^{††} 정 회 원 홍익대학교 컴퓨터공학과 교수
논문접수 2000년 1월 5일, 심사완료 2000년 1월 21일

로 주문형 비디오 서버내의 저장 시스템은 디스크 배열 형태로 구성되는데 멀티미디어 데이터들은 이와 같은 디스크 배열 상에 단위 블록으로 나누어져 분산 저장된다[2, 3]. 그러나 기존의 데이터들의 배치 정책들은 다양한 비디오 데이터들의 특성 및 저장 매체의 성능에 대한 고려 없이 일괄적인 방식으로 데이터들을 저장하여 왔다.

따라서 본 논문에서는 이와 같은 주문형 비디오 서버 환경에서 서버의 검색 성능을 좌우하는 중요한 요소인 데이터 배치 방식에 초점을 둔다 특히 인기있는 영화는 짧은 시간 동안에도 많은 요청을 받으므로 보다 많은 데이터 접근이 발생하며 이는 서버의 성능을 좌우하는 중요한 요인이 된다 그러므로 이에 대응하기 위해서는 다른 영화와 차별화 된 배치 방식이 요구된다[6]. 본 논문에서는 데이터들의 인기도 및 특성들과 저장 매체의 성능을 함께 고려하여 보다 많은 고객의 요청에 응답할 수 있는 배치 기법을 제안하고자 한다.

본 논문의 구성은 다음과 같다 2장에서는 기존에 사용되고 있는 일반적인 멀티미디어 데이터들에 대한 저장 기법에 대해 소개하고 3장에서는 본 논문에서 제안한 주문형 비디오 서버에서 비디오 데이터의 인기도에 따른 분산 배치 기법에 대해 기술한다. 제안된 배치 기법에 대한 알고리즘과 함께 상세한 설명을 덧붙인다 4장에서는 제안된 배치 정책을 평가하기 위해 실시한 실험의 환경 및 결과를 기술하고 5장에서 결론을 맺는다.

2. 멀티미디어 데이터의 저장 기법



(그림 1) 바이트 인터리빙과 블록 인터리빙 방식의 비교

주문형 비디오 서버 내의 저장 시스템은 디스크 배열 형태로 구성되는데 멀티미디어 데이터들은 이와 같은 디스크 배열 상에 분산 저장된다 멀티미디어 데이터의 특성 상 주로 사용되는 데이터 배치 방식은 바이트 인터리빙과 블록 인터리빙 방식으로서 두 방식에 대해 비교한 것이 (그림 1)이다[1]

바이트 인터리빙 방식은 물리적인 한 블록이 다수의 디스크 상에 여러 바이트나 비트로 나누어져 분산 저장되는 방식이고 블록 인터리빙 방식은 데이터가 다수의 디스크 상에 블록 단위로 나누어져 분산 저장되는 방식이다 (그림 1)에서 데이터 내의 임의의 블록인 A1이 만일 A1-1, A1-2, A1-3라는 일련의 바이트들로 구성되었을때, 바이트 인터리빙 방식에 의하면 이 세 바이트들은 각기 다른 디스크 상에 분산 저장되지만, 블록 인터리빙 방식에 의하면 하나의 블록인 A1은 한 디스크에 저장된다.

이때 멀티미디어 데이터의 특성상 블록 인터리빙 방식이 주로 사용되는데, 그 이유는 멀티미디어 데이터의 크기와 연속적인 특성 때문에 블록 인터리빙 방식이 더 디스크 탐색 시간과 저장 공간 관리의 문제를 잘 만족시키기 때문이다

그러나 한 디스크 블록의 크기는 보통 512 B에서 4 KB 정도이므로, 블록 인터리빙 방식을 사용하더라도 수 기가 바이트에 달하는 하나의 비디오 데이터는 수 백만 개에 달하는 디스크 블록으로 나뉘어 저장된다. 이때 이 블록들을 어떻게 배치시키며 어떻게 검색할 것인가의 방법들에 관한 많은 연구가 진행되어 왔다.

기존의 UNIX 시스템은 각각의 블록들을 아무 제약 없이 디스크 상의 임의의 위치에 할당하는 임의 할당 (random allocation) 방식을 사용하고 있다. 이 방식은 기억 공간의 관리가 간단하지만 연속 매체와 같이 순차적으로 접근되고 시간의 연속성이 있는 경우에는 적합하지 않다. 연속 매체들은 그 특성상 물리적으로 인접하게 배치하는 것이 좋다. 논리적으로 인접한 블록이 물리적으로 아주 다른 위치에 배치될 경우 디스크 탐색 시간의 부담 때문에 특정 수준 이상의 데이터 전송률을 보장하는 것이 어렵기 때문이다.

따라서 본 절에서는 먼저 일반 파일 시스템에서 데이터들을 배치하는 방법들에 대해 알아보고 이중에서 연속 매체에 적합한 배치 방법은 무엇인지 살펴본다. 그리고 마지막으로 현재 주문형 비디오 서버의 파일 시스템에서 주로 사용되고 있는 대표적인 데이터 배치

기법들에 대해서 살펴본 뒤 이 기법들이 갖고 있는 문제점들도 비교한다.

2.1 일반 파일 시스템에서의 데이터 배치 방식

일반 파일 시스템에서 사용되는 데이터 배치 방식으로는 연속 배치(continuous placement)와 임의 배치(random placement) 그리고 부분 연속 배치(extent-based placement) 방식이 있다.

2.1.1 연속 배치

한 파일을 구성하는 모든 블록들을 디스크 상의 물리적으로 연속된 위치에 배치하는 방식이다. 이 경우 해당 파일의 맨 처음 블록을 찾을 때만 탐색이 필요하며 그 이후의 접근에 대해서는 순차적으로 데이터 블록에 접근할 수 있으므로 접근이 빠르다는 장점을 갖는다. 그러나 이 방법의 단점은 파일 중간에 새로운 데이터를 삽입해야 할 경우 블록들 사이에 빈 공간이 없으므로 뒤의 모든 데이터 블록들을 이동시켜야 삽입할 수 있다. 뿐만 아니라 디스크 상에 빈 공간이 많이 남아 있더라도 저장하고자 하는 파일 크기에 충분한 연속 공간이 없는 경우도 자주 발생한다. 이와 같은 특성 때문에 데이터의 삽입, 삭제, 갱신이 자주 일어나는 저장 시스템에는 적합하지 않으며 읽기 전용의 환경에 적합한 배치 방법이다.

2.1.2 임의 배치

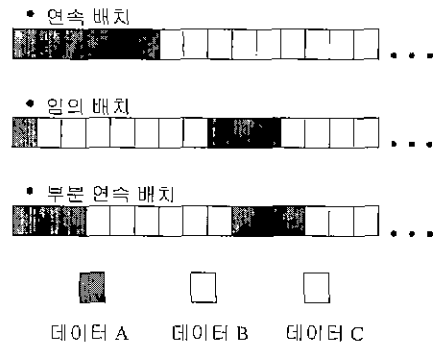
각 블록을 아무 제한 없이 디스크 상에 임의로 배치하는 방식이다 따라서 디스크 공간 관리가 용이하고 데이터들이 자주 변경되는 환경에 적합하지만 각 데이터 블록 하나 하나마다 탐색 시간이 요구되므로 연속 매체를 실시간으로 서비스해야 하는 주문형 비디오 서버에는 적합하지 않다.

2.1.3 부분 연속 배치

앞의 두 방식을 절충한 것으로 배치의 기본 단위를 블록이나 전체 파일로 하는 것이 아니라 연속된 여러 개의 블록들로 하는 것이다. 이 방법은 연속 배치의 단점들을 대부분 해결하며 연속 배치의 장점인 순차적 접근도 어느 정도 만족한다. 또한 임의 배치의 단점인 탐색 시간의 과부하 문제도 부분적으로 해결하며 장질인 저장 공간 관리의 용이성도 얻을 수 있다. 이때 연속된 블록들을 익스텐트(extent) 또는 세그먼트(segment)

라고 부르는데 익스텐트의 크기 즉 연속적으로 배치시켜야 할 블록들의 수를 얼마로 할 것인가를 결정해야 한다. 이 값의 설정은 연속 매체의 시간적 특성과 매체 재생기의 저장 시스템에 대한 데이터 요청 패턴 등에 따라 적합하게 선택되어야 한다[8].

위의 세 가지 방식에 대한 데이터 배치 예가 (그림 2)이다. 이 중에서 주문형 비디오 서버와 같이 연속 매체를 다루는 시스템에 가장 적합한 배치 방법은 부분 연속 배치이다. 기존 주문형 비디오 서버에서의 데이터 배치 정책들을 살펴보면 이 방법을 기본으로 사용하고 있다. 따라서 이 장의 끝 부분에서 다루고 있는 연속 매체들을 위한 배치 정책들은 이 부분 연속 배치를 기본으로 하고 있으며 이 정책들에서 블록이라 함은 정확히 말하면 익스텐트이다.



(그림 3) 기존 파일 시스템에서의 배치 방식 비교

따라서 일반적으로 주문형 비디오 서버에서 사용하는 데이터 배치 기법은 앞에서 언급한 블록 인티리빙을 기본으로 하되 분산 배치되는 단위 블록은 임의히 말하면 블록이 아니라 익스텐트이다

2.2 멀티미디어 데이터를 위한 대표적인 배치 정책

앞 절에서 기술한 배치 기법은 기본으로 하여 멀티미디어 데이터의 효율적인 저장을 위한 많은 배치 정책들이 연구되어져 왔다 대표적인 배치 정책으로는 all-way uniform striping 정책을 들 수 있다. 이 기법은 uniform striping과 all-way striping을 결합한 것으로서, uniform striping이란 데이터를 동일 크기의 블록으로 나누어 여러 디스크들에 동등하게 분산 배치하는 방식이고 all-way striping은 데이터를 분산 저장할

때 모든 디스크에 분산시키는 방식이다. 즉, 이 기법에서는 데이터 특성에 대한 고려 없이 모든 비디오 데이터를 일정 크기의 스트라이핑 블록으로 나누어 디스크 배열 상의 모든 디스크에 똑같이 분산 배치시킨다. 이 정책은 디스크 배열 상에서 연속된 데이터 블록들을 여러 디스크에 분산 배치시켜 놓았다가 한꺼번에 읽어낼 수 있으므로 큰 재생률을 요하는 비디오 데이터를 실시간으로 제공해야하는 주문형 비디오 서버에 적합하다.

그러나 이 기법은 저장 매체의 성능이 각기 다르거나 저장하고 있는 데이터의 인기도 차이가 많이 날 경우 자원을 충분히 이용하지 못하므로 검색 성능에 제약이 따른다.

3. 비디오의 인기도에 따른 분산 배치 정책

본 논문에서는 읽기 전용의 멀티미디어 서버인 주문형 비디오 서버 상에서, 보다 많은 고객들에게 그들이 원하는 영화를 서비스해주기 위한 배치 정책을 제안한다. 주문형 비디오 서버의 경우, 검색 성능에 가장 큰 영향을 미치는 것은 특정 시간대에 집중되는 인기있는 영화에 대한 요청들이다. 인기있는 영화의 경우, 동시에 많은 요청이 몰릴 뿐만 아니라 근소한 요청 간격 차이로 영화의 서로 다른 부분들이 동시에 검색되어 서비스되어야 한다 따라서 이들을 어떻게 처리하느냐에 따라 전체 서버의 성능이 크게 좌우된다.

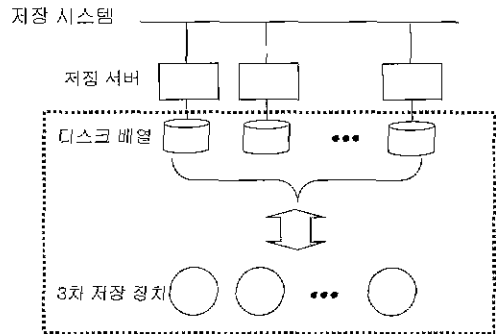
그러나 기존의 배치 정책인 all-way uniform striping은 데이터들을 블록 단위로 나누어 디스크 배열 내의 모든 디스크들 상에 일률적으로 분산 배치하기 때문에, 비디오 데이터나 저장 매체의 특성이 전혀 고려되지 않고 있다. 따라서 본 논문에서는 각 비디오 데이터의 인기도를 비롯한 특성과 저장 매체의 성능을 고려한 효율적인 배치 정책을 제안한다. 이 정책의 기본 개념은 인기도가 큰 비디오 데이터일수록 저장 매체의 입출력대역폭을 크게 함으로써 동시에 보다 많은 데이터를 검색하여 전송할 수 있게 하는 것이다.

3.1 배치 정책에 사용된 비디오 데이터의 특성

본 배치 정책에 사용된 비디오 데이터의 특성 정보는 데이터 압축 과정에서 얻어지는 해당 데이터의 총 크기, 전송대역폭 및 상영 시간과, 비디오 데이터의 반복되는 배치 과정에서 얻어지는 인기도, 즉 평균 요청 간격이다. 비디오 데이터의 전송대역폭이란 해당

비디오에 대해 고객으로부터 한 번 요청이 있을 때 단위 시간에 검색하여 전송해야하는 데이터 양을 의미한다.

또한 주문형 비디오 서버의 저장 시스템은 (그림 3)에서와 같이 디스크와 같은 2차 저장 장치와 광 디스크 같은 3차 저장 장치를 혼용하는 것이 보통이다. 즉, 대부분의 비디오 데이터는 대용량의 3차 저장 장치에 저장되어 있다가 필요시 디스크로 올려지게 되는데, 이 교체 과정에서 이 비디오 데이터에 대한 최근 평균 요청 간격 정보를 얻을 수 있으며 이것이 곧 해당 비디오 데이터의 인기도를 의미한다. 본 논문에서는 바로 이 정보를 데이터 배치 시에 이용하게 된다.



(그림 4) 주문형 비디오 저장 서버에서 비디오 교체 과정

3.2 비디오의 인기도에 따른 분산 배치 알고리즘

제안된 배치 정책에서는 기본적인 배치 기법으로, k-way uniform striping을 사용한다. 이 기법은 uniform striping과 k-way striping을 결합한 것으로서, k-way striping이란 데이터를 k개의 디스크에 분산 배치시킨다는 것을 의미한다. 이때 k는 각 데이터에 대한 평균 요청 간격 즉 인기도에 의해 결정되는데, 인기도가 큰 비디오 데이터일수록 k 값을 크게 함으로써 동시에 보다 많은 양의 데이터를 검색하여 전송할 수 있게 한다. 본 논문에서 제안한 배치 알고리즘은 다음 (그림 4)와 같다.

- 인기도에 따른 분산 배치 알고리즘

단계 1: 디스크 그룹화 과정

입출력 대역폭 성능이 유사한 디스크들을 논리적으로 같은 그룹으로 구성

단계 2 : 비디오 데이터 $x(V_x)$ 에 대한 평균 요구량 계산

$$V_x \text{ 평균요구량} = V_x \text{ 전송대역폭} \times \frac{V_x \text{ 상영시간}}{V_x \text{ 평균요청간격}} \quad (1)$$

단계 3 비디오 데이터 $x(V_x)$ 에 대한 분산 배치 폭 계산

V_x 배치량 = V_x 평균 요구량

while (V_x 배치량 $\neq 0$)

{ if disk group G_i = (2)

$$\min \left(\frac{\sum G_i \text{에 이미 할당된 비디오들의 평균 요구량}}{G_i \text{의 입출력대역폭}} \right)$$

Select G_i for placing video data x

if (G_i 의 입출력대역폭 < V_x 배치량) (3)

{ SW[x][i] = 디스크 그룹 G_i 의 총 디스크 수
 V_x 배치량 = V_x 배치량 - G_i 의 입출력대역폭
 }

$$SW[x][i] = \frac{V_x \text{ 배치량}}{G_i \text{의 단위 디스크의 입출력대역폭}}$$

else (4)

Striping_width[x] = Striping_width[x] + SW[x][i]
 } /* end of while */

while ($G_i \in$ 비디오 데이터 x 의 할당을 위해 (5)
 선택된 그룹들의 집합)

{ Place video data x on G_i by Uniform Striping
 with SW[x][i] width
 }

모든 비디오 데이터에 대해 단계 3 반복

end /* 분산 배치 알고리즘 끝 */

(그림 4) 인기도에 따른 분산 배치 알고리즘

위의 배치 알고리즘에 대해 단계별로 설명하면 다음과 같다. 먼저 단계 1은 저장 시스템 내의 디스크 배열을 유사한 성능의 디스크들로 그룹화하는 과정이다. 비디오 데이터들은 이렇게 구성된 논리적인 디스크 그룹 내에서 분산 배치되는데, 한 스트라이핑 그룹 내의 디스크들이 성능 차이가 많이 나면 좋은 성능의 디스크들을 충분히 활용할 수 없기 때문이다[5,7]

단계 2는 각 비디오 데이터에 대한 분산 배치 폭,

즉 몇 개의 디스크에 분산 배치시킬 것인가를 결정하기 위해 먼저 각 데이터에 대한 평균 요구량을 구한다. 식 (1)에서 상영시간/평균요청간격은 해당 비디오 데이터가 평균적으로 늘 요청되고 있는 횟수를 의미하므로, 평균 요구량이란 각 비디오 데이터의 인기도를 고려했을때 해당 데이터가 단위 시간에 검색되어 전송되어야 할 총 데이터 양을 의미한다. 따라서 식 (1)에서 V_x 평균 요구량이란 비디오 데이터 x 가 한 번 요청될 때 단위 시간에 검색되어 서비스되어야 할 데이터 량을 의미한다

단계 3에서는 단계 2에서 구한 평균 요구량을 근거로 해당 비디오 데이터의 분산 배치 폭을 구한다. 단계 3에서 V_x 배치량이란 비디오 데이터 x 가 디스크 상에 배치되는 과정에서 배치되어야 할 잔여 데이터 량을 의미하고 G_i 는 디스크 그룹 i 를 지칭한다. 식 (3)과 (4)에서 사용된 SW[x][i]는 비디오 데이터 x 를 배치하기 위해 선택한 디스크 그룹 i 가 있을 때 이 그룹 내에서 해당 비디오 k 의 스트라이핑 폭을 값으로 갖는다. 이때 만일 디스크 그룹 i 의 잔여 입출력대역폭이나 용량이 충분하다면 식 (4)에서처럼 한 디스크 그룹 내에 비디오 k 가 모두 배치될 수 있으나, 식 (3)에서처럼 그렇지 못할 경우에는 여러 디스크 그룹을 선택하여 각각 조금씩 배치하여야한다. 단계 3의 과정이 끝나면 각 비디오 데이터 x 의 스트라이핑 폭인 k 의 값이 결정되는데, 위의 알고리즘에서 비디오 데이터 x 에 대한 k 값은 최종적으로 Striping_width[x]에 들어가게 된다. 또한 이와 같이 k 값이 결정되면 비디오 x 는 k 개의 디스크에 uniform striping으로 배치된다.

이때 또 한 가지 중요한 점은 각 비디오들에 대한 k 값을 구할 때 비디오의 특성뿐만이 아니라 저장 매체의 성능도 함께 고려한다는 것이다. 식 (2)에서는 비디오를 배치시킬 디스크 그룹을 찾을 때 나중에 디스크들이 받을 작업 부하의 균형을 생각하여, 성능에 비해 작업 요구량이 가장 적은 비디오 데이터들이 할당되어 있는 그룹을 차례로 선택한다. 이렇게 선택된 디스크 그룹이 만일 현재 비디오 데이터 x 를 모두 수용할 만큼의 입출력대역폭을 갖는다면(식 (4)), 해당 비디오 데이터에 대해 최종적인 분산 배치 폭, k 를 설정하여 선택된 디스크 그룹 G_i 에 분산 배치시킨다. 그러나 만일 선택된 디스크 그룹의 능력이 현재 배치할 데이터의 평균 요구량을 소화해낼 수 없다면(식 (3)의 경우) 다수의 디스크 그룹을 선택하여 분산 배치하게 된다.

4. 실험

4.1 실험 환경

본 논문에서 제안한 비디오 데이터의 인기도에 따른 분산 배치 정책의 검색 성능을 증명하기 위해 모의 실험을 수행하였다. 실험에서 사용된 비디오 데이터들은 MPEG-1으로 압축된 데이터 트레이스로서, 구체적인 사항은 <표 1>과 같다[4].

본 논문에서 제안한 배치 정책에서는 비디오 데이터의 인기도를 사용하게 되는데 데이터들의 인기도는 표에서 나타난 평균 요청 간격에 의해 결정되고, 실험의 편의를 위해 이 평균 요청 간격 값은 다양한 인기도를 갖는 비디오들을 설정하기 위해 임의로 정한 값으로서, 참고로 인기있는 비디오에 대한 평균 요청 간격 값은 논문[7]의 실험에서 참조하였다.

실험에서는 저장 시스템을 구성하기 위해 네 종류의 디스크들을 사용하였는데, 디스크 각각에 대한 정보는 <표 2>와 같다. 이 디스크들을 다양하게 조합하여 총 디스크 수를 40과 60으로 구성하여 각각 실험하였으며 이들을 가지고 구성한 전체 저장 시스템의 구조는 <표 3>과 같다.

<표 1> 실험에 사용된 비디오 데이터

비디오 제목	총 크기 (GB)	평균전송대역폭 (Mbps)	상영시간 (sec)	평균요청간격 (sec)
asterix	0.5587	0.59	7200	10
atp(tennis)	0.5473	0.55	3600	10
dino(movie)	0.327	0.33	7200	10
lambs(" ")	0.183	0.18	5400	20
mr bean	0.441	0.44	3600	20
mtv	0.495	0.49	3600	20
news	0.384	0.38	3600	30
race	0.7687	0.77	7200	30
settop	0.15	0.15	3600	40
simpsons	0.4644	0.46	3600	60
soccer	0.6278	0.63	6000	300
starwars	0.39	0.36	7200	600
talk show1	0.3634	0.36	3600	600
talk show2	0.4479	0.49	3600	1800
terminator	0.2726	0.27	7200	3600

이와 같이 디스크 배열의 크기를 40과 60으로 한 이유는 <표 2>의 디스크들의 성능 상, 사용된 저장 시스템의 성능이 배치될 전체 비디오 데이터들의 요구량 보다 작은 경우와 큰 경우에 대해 각각 실험하기 위해

서이다. <표 3>과 같이 디스크 그룹을 구성할 때 크기가 60정도 되어야 <표 1>의 비디오 데이터의 요청에 대해 충분히 서비스 가능하다.

<표 2> 실험에 사용된 디스크 종류

디스크 모델명	저장 능력 (MB)	평균 탐색 시간 (ms)	평균 회전율 (ms)	단위데이터 전송 시간 (ms/KB)
Elite9	9090	11	5.56	0.144
Medalist	631	14	7.87	0.2
Hawk	1050	9	5.54	0.142
Barracuda1	2150	8	4.17	0.13

<표 3> 실험에 사용된 디스크 배열의 구성

(B : Barracuda1, H : Hawk, E : Elite9, M : Medalist)

총 디스크 배열의 크기	디스크 그룹의 구성			
	디스크 그룹 G1	디스크 그룹 G2	디스크 그룹 G3	디스크 그룹 G4
40	B 10개	H 10개	E 10개	M 10개
60	B 15개	H 15개	E 15개	M 15개

4.2 실험 결과

4.1 절에서 설명한 실험 환경을 통해 본 논문에서 제안한 배치 정책의 검색 성능을 측정하였다. 첫 번째 실험에서는 <표 1>에서 소개한 15개의 비디오 데이터를 모두 사용하였고 디스크 배열의 크기는 40으로 하였다. 이때 본 논문에서 제안한 인기도를 고려한 배치 정책과 인기도를 고려하지 않은 배치 정책에 대해 동시에 서비스 가능한 고객의 수를 측정함으로써 검색 성능을 비교하였다. 먼저 <표 4>에서는 인기도를 고려한 배치 정책을 사용하여 데이터를 배치하였을 때 각 비디오 데이터가 어느 디스크 상에 어느 정도로 분산 배치되었는지를 나타내고 있다. 이 실험의 경우, 현재 사용된 저장 시스템의 성능이 깨칠될 전체 비디오 데이터들의 요구량 보다 작아서 일부의 비디오 데이터는 배치되지 못한다.

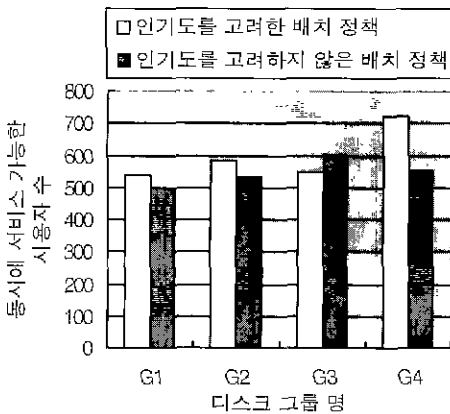
(그림 5)는 이 실험에 대한 결과를 보이고 있는데 인기도를 고려한 배치 정책을 사용하였을 때가 대체로 더 좋은 검색 성능을 내고 있다. 특히 서비스 가능한 전체 사용자 수로 비교할 때 인기도를 고려한 배치 정책의 경우 207,962명을 더 지원할 수 있다. 또한 인기도를 고려하지 않은 배치 정책의 경우, 비디오의 검색 요구량이 디스크 성능에 비례하지 않으므로 작업 부하가 균형을 이루지 못하고 있음을 알 수 있다.

〈표 4〉 비디오의 배치 상태
(총 디스크 배열의 크기=40일 경우)

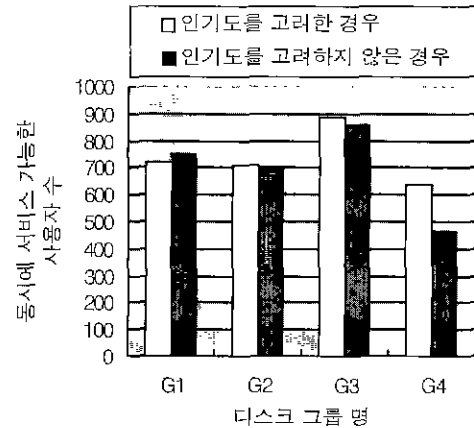
비디오 제목	스트라이핑 폭	G1	G2	G3	G4
asterix	14	10	4		
atp(tennis)	8			8	
dmo(movie)	12		2		10
lamb(")	2		2		
mr.bean	4		3	1	
mtv	x				
news	2			2	
race	x				
settop	x				
simpsons	x				
soccer	x				
starwars	1			1	
talk show1	1			1	
talk show2	1			1	
terminator	1			1	

〈표 5〉 비디오의 배치 상태
(총 디스크 배열의 크기=60일 경우)

비디오 제목	스트라이핑 폭	G1	G2	G3	G4
asterix	14	14			
atp(tennis)	8		8		
dmo(movie)	10			10	
lamb(")	3				3
mr.bean	5				5
mtv	4		4		
news	3				3
race	8		5		3
settop	1			1	
simpsons	2			2	
soccer	1	1			
starwars	1			1	
talk show1	1			1	
talk show2	1	1			
terminator	1	1			



(그림 5) 인기도 고려 여부에 따른 배치 정책의 검색 성능 비교
(디스크 배열의 크기 = 40일 경우)



(그림 6) 인기도 고려 여부에 따른 배치 정책의 검색 성능 비교
(디스크 배열의 크기 = 60일 경우)

두 번째 실험은 디스크 배열의 크기를 60으로 하였을 때 <표 1>의 비디오 데이터를 배치시킨 실험으로 내용은 앞과 유사하다. <표 5>에서는 인기도를 고려한 배치 정책을 사용하여 데이터를 배치하였을 때 각 비디오 데이터가 어느 디스크 상에 어느 정도로 분산 배치되었는지를 나타내고 있는데, 이 실험의 경우에는 현재 사용된 저장 시스템의 성능이 배치될 전체 비디오 데이터들의 요구량 보다 크므로 전체 비디오 데이

터들이 모두 배치된다.

(그림 6)은 이 두 번째 실험의 결과를 보이고 있으며 역시 인기도를 고려한 배치 정책을 사용하였을 때가 더 좋은 검색 성능을 내고 있는데, 서비스 가능한 전체 사용자 수로 비교할 때 인기도를 고려한 배치 정책의 경우 18629명을 더 지원할 수 있다. 또한 디스크 그룹들 사이의 작업 부하도 인기도를 고려한 배치 정책의 경우에 더 균형을 이루고 있음을 알 수 있다.

5. 결 론

대용량의 저장 시스템을 갖는 주문형 비디오 서버는 많은 고객들의 요청에 대해 서비스하게 되지만, 특히 인기있는 비디오 데이터일 경우에는 단 시간 내에도 많은 요청이 집중될 수 있으므로 동시에 보다 많은 비디오 데이터를 검색하여 전송할 수 있는 배치 정책이 필요하다.

본 논문에서는 인기 전용의 주문형 비디오 서버에서 동시에 보다 많은 고객들에게 검색 서비스를 제공하기 위한 방법으로 비디오 데이터의 인기도에 따른 분산 배치 기법을 제안하였다. 인기있는 비디오의 경우 고객들의 요청이 특정 시간 대에 집중될 수 있으므로 이 비디오들에 대해 예상되는 요청 정도와 데이터 특성을 배치 시에 이용하면 좋은 검색 효율을 얻을 수 있다.

제안된 배치 정책의 효율성을 평가하기 위해, 주문형 비디오 서버의 저장 시스템 환경을 구성하고 같은 환경 하에서 비디오 데이터의 인기도를 고려하지 않은 기존의 배치 정책과 비교한 결과, 제안된 배치 정책이 더 좋은 검색 성능을 보임을 확인하였다. 또한 제안된 배치 정책에서는 비디오들의 인기도에 따라 배치할 디스크 그룹을 선택할 때 기회를 균등하게 부여하므로, 디스크 그룹들간에도 작업 부하의 균형을 이루게 됨을 알 수 있었다.

참 고 문 헌

[1] Peter M Chen, Edward K. Lee, David A. Peterson, "RAID : High-Performance, Reliable Secondary Storage," ACM Computing Surveys, Vol.26, No.2, pp.152-158, June 1994.
 [2] Andrew Heybey, Mark Sullivan, Paul England, "Calliope : A Distributed, Scalable Multimedia Server," Proceedings of the USENIX 1996 Annual Technical Conference, Jan 1996.
 [3] Mathias Rantenberg, Helmut Rzehak, Interactive Distributed Multimedia Systems and Services, Springer, pp.265-268, 1996.
 [4] O Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM

systems," Univ. of Wurzburg Research Report Series No.101, Feb 1995

[5] Prashant J Shenoy, Harnck M Vin. "Efficient Striping Techniques for Multimedia File Servers," Univ. of Texas at Austin, Technical Report CS-TR-96-27, pp 1-21, 1997
 [6] Sangyup Shim, Harish Vedavyasa, Liu, David H. C. Du. "An Effective Data Placement Scheme to Serve Popular Video-On-Demand," Pacific Workshop on Distributed Multimedia Systems, Hong Kong, June 1996.
 [7] Yuewei Wang, David H. C. Du, "Weighted Striping in Multimedia Servers," Proc. of IEEE Int'l Conf. on Multimedia Computing and systems pp.102-109, June 1997
 [8] 김병호, 이윤준, 멀티미디어 개념 및 응용, pp 311-317, 홍릉과학출판사, 1997



안 유 정

e-mail : kls302@chollian.net

1991년 이화여자대학교 전자계산학과 학사

1991년~1993년 금융결제원 제직

1995년 이화여자대학교 전자계산학과 석사

1995년~현재 홍익대학교 컴퓨터공학과 박사 과정
 관심분야 : 멀티미디어 시스템, 실시간 시스템



원 유 현

e-mail : won@cs.hong.ac.kr

1972년 성균관대학교 수학과 학사

1975년 한국과학원 전자계산학과 석사

1985년 고려대학교 이학박사

1975년~1976년 한국과학기술연구소 연구원

1986년~1987년 R.P.I. 교환 교수

1976년~현재 홍익대학교 컴퓨터공학과 교수

관심분야 : 프로그래밍 언어론, 멀티미디어 시스템, 실시간 언어