

클러스터드 주문형 비디오 서버에서 비디오 스트림의 검색효율을 높이는 배치정책

정희원 안 유 정*, 원 유 헌*

A placement Policy improving Retrieval Efficiency of video streams in Clustered VOD Servers

You-Jung Ahn* Yoo-Hun Won* *Regular Members*

요 약

주문형 비디오 서버의 중요한 목표는 일정시간 내에 보다 많은 클라이언트들에게 그들이 원하는 서비스를 제공하는 것이다. 이와 같이 효율적이고 빠른 정보 제공을 위해서는 여러 가지 정책들이 함께 고려되어야 하지만, 정보 저장시의 효율적인 정보의 배치는 검색의 효율을 높이는 중요한 요인이 된다. 본 연구에서는 클러스터드 주문형 비디오 서버에서 비디오 데이터들을 저장 시스템에 저장할 때, 보다 효율적인 검색 성능을 제공할 수 있는 배치 정책을 제안한다. 제안된 배치 정책은 대규모의 디스크 배열을 구성하는 다양한 성능의 디스크들을 평가하여 유사한 성능을 갖는 것들끼리 디스크 그룹을 구성하고 각 디스크 그룹마다 그 성능에 적절한 특성을 지닌 비디오 데이터를 할당한다. 또한 제안된 정책의 검증에 위해 모의 실험을 통해 제안된 정책과 기존의 배치 정책을 비교함으로써 검색 효율이 개선됨을 보인다.

ABSTRACT

One of the most important goals in VOD servers is to provide services to more clients with services which clients request. In order to provide services efficiently and rapidly, though considering a few of policies, efficient placement of data when they are stored is direct cause to improve efficiency of retrievals. In this paper, we propose a efficient placement policy, encoded video data being stored in clustered VOD servers. In the proposed placement policy, partitioning a large disk array into smaller disk groups which consists of a few of disks with similar performances, specially disk I/O bandwidth. In last chapter, we compare proposed placement policy with conventional policies, and show the results of improved performances with proposed policy.

I. 서론

주문형 비디오 서버에서 단위 시간에 보다 많은 클라이언트들에게 그들이 원하는 요인들이 함께 고려되어야하는데, network 환경, 버퍼링(buffering) 정책, 디스크 스케줄링(disk scheduling) 정책, 배치(placement) 정책 등이 그것이다. 본 연구에서는 이 중에서도 디스크상의 배치정책을 통해 위의 목표에 접근하고자 한다.

주문형 비디오 서버의 특성상 비디오 데이터에 대해 클라이언트의 요청을 받아 해당 데이터를 검색하여 전송하는 과정은 실시간 작업을 요구하지만, 이 데이터를 적절한 디스크에 배치하여 저장하는 과정은 비 실시간 작업으로 이루어질 수 있다. 따라서 압축된 형태로 만들어진 비디오 데이터들의 특성과 이들이 저장될 저장 장치의 성능을 충분히 고려하여 보다 효율적인 배치를 하는 것은 데이터의 검색시의 효율을 높일 수 있다.

* 홍익대학교 컴퓨터공학과

논문번호 : 99050-0218 접수일자 : 1999년 2월 18일

클러스터드 주문형 비디오 서버의 저장 시스템은 다양한 입출력 대역폭과 저장 능력을 갖는 디스크들 수십 내지 수백 개가 디스크 배열 형태로 구성되는데, uniform striping이나 all-way striping과 같은 전통적인 배치 기법은 고도의 성능을 가진 디스크들을 충분히 이용하지 못한다. 특히 한 디스크 배열 내에서 디스크들의 성능 차이가 클 경우, 지원 가능한 비디오 스트림의 수는 가장 성능이 낮은 디스크의 입출력 대역폭에 의해 제한된다^[1]. 따라서 본 연구에서는 주문형 비디오 서버 내의 디스크들의 다양한 성능과 비디오 데이터의 특성을 함께 고려하여 검색 효율을 높이고자 한다.

본 논문의 구성을 살펴보면, II장에서는 본 연구의 배경이 되고 있는 클러스터드 주문형 비디오 서버의 구조와 기존의 배치 정책들에 대해 간략히 기술하고 III장에서는 디스크들의 그룹화 방법과 그에 따라 저장될 비디오 데이터의 특성에 따른 배치 정책을 제안한다. IV장에서는 모의 실험을 통해 제안된 배치 정책과 기존의 배치 정책을 비교함으로써 개선된 결과를 보이고 V장에서 결론을 맺는다.

II. 주문형 비디오 서버의 환경

1. 클러스터드 주문형 비디오 서버의 구조

클러스터드 비디오 서버는 그림 1에서와 같이 상호 연결된 다수의 노드와 디스크들로 구성된 2-계층 구조를 이루는 것이 보통인데, front-end 노드는 주문형 비디오 서버 내 외부로의 데이터 전송을 담당하고 back-end 노드는 저장과 검색을 담당한다^[2]. 대부분 주문형 비디오 서버는 수십 또는 수백 개의 데이터 검색과 전송을 동시에 해야 하며, 특히 인기 있는 비디오일 경우 특정 시간대에 클라이언트들의 요구가 집중될 수 있다. 이러한 문제는 사용자의 경험 예측과 클라이언트 시스템의 버퍼링 등을 통해서도 어느 정도 극복할 수 있겠으나 본 연구에서는 저장 시스템에서의 배치 정책을 통해 문제의 해결에 접근하고자 한다.

2. 기존의 배치 정책들

클러스터드 주문형 비디오 서버의 저장 시스템은 대규모의 디스크들이 디스크 배열의 형태로 구성되는데, 전형적으로 서버가 사용하는 기존의 배치 기법으로는 uniform striping과 all-way striping이 있다. uniform striping은 디스크 배열 내의 다양한 디스크들의 성능에 관계없이 저장하고자 하는 데이터

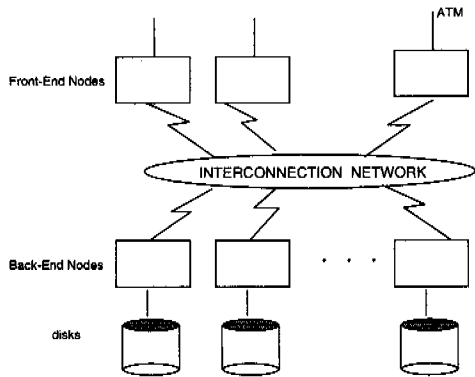


그림 1. 클러스터드 주문형 비디오서버의 구조

를 균등하게 나누어 저장하는 방식이다. 그러나 같은 스트라이핑 그룹내의 디스크들이 입출력 대역폭이나 저장 능력이 같다면 문제가 없지만, 성능이 다를 경우 지원 가능한 스트림의 수는 제일 입출력 대역폭이 낮은 디스크의 성능에 의해 제한되고, 그보다 성능이 높은 디스크들은 제한된 만큼 디스크 이용률이 저하된다.

all-way striping은 디스크 배열을 구성하는 디스크 수에 관계없이 한 스트림을 모든 디스크에 분산 저장하는 방법이다. 그러나 한 디스크 그룹을 구성하는 디스크 수가 무조건 많다고 전체적인 검색 효율이 높은 것은 아니다. [3]에서는 디스크 그룹의 크기(partition size)에 따라 동시에 지원가능한 고객의 수를 실험에 의해 보이고 있는데 그 결과는 그림 2와 같다. 이 그림에서도 알 수 있는 바와 같이 디스크 그룹의 크기가 수십 개에서 수백 개에 이르게 되면 오히려 동시에 지원가능한 스트림의 수는 떨어지는 것을 알 수 있다.

총 디스크 수 = 120일 경우

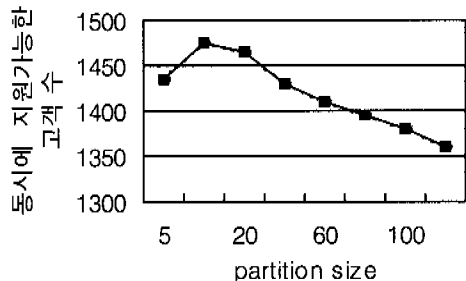


그림 2. partition size와 동시 지원 가능한 고객 수와의 관계

이와 같은 배치 기법들의 문제 해결을 위해 wei-

ghted striping이라는 새로운 배치 기법이 제안되었다. 이 기법은 디스크마다 그 성능을 기준으로 가중치를 계산하고 가중치 값에 비례하는 데이터 양을 디스크에 할당하도록 하는 스트라이핑 기법이다. 그림 3은 이 기법에 대한 기본 알고리즘을 설명하고 있다. 그림에서와 같이 이 기법은 각 디스크마다 입출력 대역폭 성능을 기준으로 가중치를 계산하고 저장하고자 하는 데이터를 스트라이핑 블록으로 나누어 각 블록마다 현재 디스크 상태를 고려하여 성능에 비해 가장 적은 양의 데이터가 할당되어 있는 디스크에 현재의 블록을 할당한다. 이렇게 하므로써 디스크들은 각기 성능에 따라 작업 부하가 조절되므로 검색 시 디스크 이용률을 더욱 높일 수 있게 되고 따라서 검색 효율도 높아진다^[1]. 그러나 멀티미디어 데이터의 특성상 하나의 비디오 데이터에 대한 검색 요청시, 해당 파일에 대해 연속적인 블록들을 순차적으로 읽어야하는데, 그림 3에서 보면 weighted striping에서는 연속적인 블록 1, 2, 3, 4, 5, 6, 7 ... 를 디스크 1, 2, 3, 3, 2, 1, 3 ...에 할당하게 되므로 다음 블록을 읽을 때 걸리는 탐색 시간과 회전 지연 시간은 uniform striping 기법을 사용했을 때 보다 오히려 더 걸릴 수 있다. uniform striping의 경우, 그림 3의 연속적인 블록들 1, 2, 3, 4, 5, 6, 7 ...은 디스크 1, 2, 3, 4, 5, 6, 7 ...에 저장 되게 되므로 특별히 각 블록들에 대한 위치 정보를 알아내는데 초과 시간이 들지 않는다.

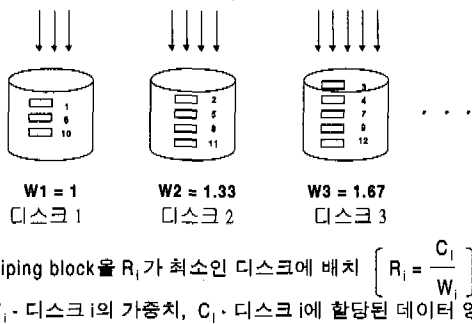


그림 3. weighted striping 기법의 기본 알고리즘

따라서 본 연구에서는 이 weighted striping 기법에서와 같이 저장 시스템내의 다양한 디스크들의 성능을 충분히 이용하면서도 멀티미디어 데이터들의 연속적인 특성을 이용하여 보다 탐색 시간을 줄일 수 있는 배치 정책을 제안하여 대용량의 검색 서버인 주문형 비디오 서버에 적용하고자 한다.

III. 검색의 효율을 높이는 배치 정책

1. 연구 동기

주문형 비디오 서버에서 클라이언트들에게 요구되는 재생물로 비디오 데이터를 제공하기 위해서는 압축하여 저장하는 것이 필수적이나 압축 후에도 그 크기는 작지 않다. 한 예로 MPEG-2로 압축된 2시간 짜리 비디오 데이터인 경우 약 4GB 정도의 용량을 차지한다. 보통 주문형 비디오 서버에서는 이와 같은 비디오 데이터들 수 백개 이상을 제공하고 있으므로 이와 같은 대용량의 데이터를 저장하기 위해 자기 디스크만으로 저장 시스템을 구성하는 것은 경제적인 문제가 따른다. 따라서 보통은 광 디스크 주크 박스와 같은 경제적인 저장장치와 고속의 자기 디스크를 혼용하여 저장 시스템을 구성한다. 이때 자기 디스크는 광 디스크의 캐쉬와 같이 사용되는데, 데이터들은 먼저 광 디스크에 저장되어 있다가 필요시에 자기 디스크로 올려진다. 이때 LRU와 같은 교체정책이 사용되는데, 이 같은 교체 과정에서 각 비디오 데이터들에 대한 고객들의 요청 정도를 알 수 있다. 특히 최근 고객들로부터의 평균 요청 간격(즉 인기도), 요구가 집중되는 시간 대 등의 정보를 데이터 저장 시에 이용하면 좋은 검색 효율을 얻을 수 있다.

이와 같은 계층적인 저장 장치 구조를 사용하더라도 기존에 나와있는 디스크들의 성능으로 고려해 볼때 수십 개 이상의 자기 디스크들이 필요하며 이 디스크들은 소프트웨어적으로 구성된 디스크 배열 형태로 구성된다. 저장 시스템 역시 계속 변화와 확장을 거듭하게 되고 VLSI 기술의 발달로 점점 더 성능이 좋은 디스크들이 추가됨에 따라 저장 시스템 내에는 다양한 성능의 디스크들이 공존하게 된다. 따라서 이와 같은 저장 시스템의 환경 내에서 보다 검색 효율을 높이려면 이 디스크들의 성능을 검색 시에 충분히 활용하고 저장하고자 하는 비디오 데이터들의 특성도 함께 고려하여 저장하는 것이 필요하리라고 본다.

따라서 본 논문에서는 먼저 디스크들의 성능을 고려하여 저장 시스템을 재구성하고 여기에 데이터들의 다양한 특성을 함께 고려하여 배치시키는 정책을 제안한다. 이때 고려되는 데이터들의 특성으로는 데이터 압축 과정에서 얻어지는 각 데이터의 평균 전송 대역폭, peak시의 재생률, 총 크기 등의 정보와 데이터 교체 과정에서 얻어지는 최근 이 데이

터에 대한 평균 요청 간격을 저장 시에 이용하도록 한다. 이때 저장 시스템을 재구성하는 작업은 저장 서버가 처음 구성될 때나 디스크들이 추가되어 저장 서버가 확장 또는 변경될때만 수행되며 일단 저장 시스템이 재구성되고 나면 데이터들의 배치 과정만 반복된다.

2. 디스크 배열의 구성

앞 절에서 언급한대로 디스크 배열의 크기와 동시에 지원 가능한 스트림의 수는 비례하지 않는다. 오히려 대형 디스크 배열을 하나의 디스크 그룹으로 하여 데이터를 배치하면 성능이 더 떨어지는 것을 알 수 있다.

따라서 본 연구에서는 대규모의 디스크 배열을 보다 작은 규모의 여러 개의 디스크 그룹으로 분할하고 분할된 디스크 그룹마다 독립된 비디오 데이터들을 RAID에서의 block interleaving 방식^[7]으로 배치시킨다. 이때 디스크 분할 방법은 디스크들의 입출력 대역폭 성능이 유사한 것들을 같은 그룹으로 묶는데 이렇게 하므로써 그룹 내에서 디스크들 간의 데이터 검색 및 전송 시간의 차이를 줄여 성능이 좋은 디스크들을 충분히 활용할 수 있다. 앞 절에서도 언급한 바와 같이 디스크 그룹 내에서 디스크들 간의 입출력 대역폭 성능차가 심하면 멀티미디어 데이터의 특성상 성능이 좋은 디스크는 느린 디스크의 검색 및 전송 속도를 기다려야 하기 때문이다.

그러나 전체 디스크 배열 상에 성능이 유사한 디스크들이 많을 경우, 다수의 디스크들이 어느 한 그룹으로 집중될 수 있는데, 이때 디스크 분할시 각 디스크 그룹의 크기를 어떻게 정할 것인가 하는 것은 매우 중요한 문제이다. 이에 대해 [3]에서는 전체 디스크 배열의 크기에 따라 동시에 지원 가능한 비디오 스트림의 수를 최대화할 수 있는 디스크 그룹의 크기(partition size)를 실험을 통해 제시하고 있으며 그 결과는 그림 4와 같다. 따라서 본 연구에서는 디스크 배열 분할 시, [3]에서 제안한 최적의 디스크 그룹의 크기를 이용하는데, 각 디스크 그룹은 입출력 대역폭 기능이 유사한 디스크들로 묶일 뿐 아니라 최적의 그룹 크기(partition size)를 갖도록 재조정된다.

총 디스크의 수를 D , 디스크 i 의 평균 탐색 시간을 $Seek_i$, 평균 rotational latency를 Lat_i , 단위 데이터 전송시간을 Tf_i 라고 할 때, 디스크 그룹화 과정에서 사용되는 기본 알고리즘은 다음과 같으며, 이

때 사용되는 기호에 대한 정의는 표 1과 같다.

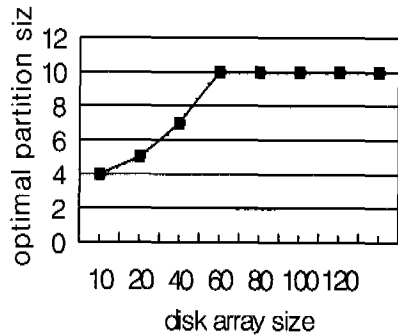


그림 4. 디스크 배열의 크기에 따른 최적의 partition size

표 1. 디스크 그룹화 과정에 대한 기호 정의

- D : 디스크들의 총 수
- W_i : 디스크 i 의 가중치
- Group_number : 디스크들의 그룹 수
- Partition_Group[j] : 디스크 그룹 j 에 속하는 디스크들의 집합
- Partition_group_Size[j] : 디스크 그룹 j 의 디스크 수
- Optimal_partition_Size[D] : 총 디스크 배열의 크기가 D 일 때 최적의 partition size
- $\theta[D]$: 총 디스크 배열의 크기가 D 일 때 독립적으로 존재할 수 있는 최소의 디스크 그룹 크기

● 디스크 그룹화 알고리즘

단계 1 : 각 디스크들에 대해 그들의 입출력 대역폭 성능을 기준으로 가중치를 구한다.

즉, 가중치 W_i 는 디스크 i 가 단위 시간에 검색하여 전송할 수 있는 데이터의 양을 나타낸 것이므로 W_i 가 클수록 성능이 우수한 디스크이다.

$$W_i = \frac{1}{Seek_i + Lat_i + Tf_i} \quad (1 \leq i \leq D) \quad (1)$$

단계 2 : 디스크들을 가중치가 유사한 것들끼리 그룹화한다.

for ($i = 1$ to D)

if ($|W_i - W_{i+1}| \leq \epsilon$)

Group W_{i+1} into

Partition_Group[Current_Group]

else

```
Add 1 to Current_Group
Group  $W_{i+1}$  into
Partition_Group[Current_Group]
```

```
단계 3 : 구성된 디스크 그룹들의 크기가 최적의
디스크 크기에 근접하도록 디스크 그룹을 재조정
for (i = 1 to Current_Group) {
Partition_Group_Size[i] =
 $\delta \cdot \text{Optimal\_Partition\_Size}[D] + \rho$ 
if  $\rho < \theta [D]$ 
Partition Partition_Group[i] into  $\delta$  groups
else
Partition Partition_Group[i] into  $\delta + 1$  groups
} /* for */
```

3. 데이터의 배치 과정

앞 절에서는 디스크들을 성능별로 그룹화하여 독립된 여러 개의 디스크 그룹을 구성하였다. 이 절에서는 각 비디오 데이터에 적합한 성능을 갖는 디스크 그룹을 선택하여 배치하는 방법을 제안한다.

비디오 데이터들을 디스크 배열 상에 저장할 때 고려해야 할 기본 사항으로는 스트라이핑 단위의 크기와 이들의 배치 방법이라고 볼 수 있다. 먼저 스트라이핑 단위 블록의 크기는 고정적이거나 가변적일 수 있는데, 양쪽 모두 장단점을 지닌다.

먼저 고정 크기의 단위 블록을 사용하는 경우에는 블록마다 동일한 크기를 가지므로 구현이 쉬우며 일정 크기를 계속 검색하므로 탐색 시간이 상대적으로 짧은 장점을 가지고 있다. 반면 가변 크기의 단위 블록에는 고정된 수의 프레임들이 저장되므로 단위 블록들의 크기가 가변적인데, 이 경우 탐색 시간과 회전 지연상의 과부하가 있을 수 있으나 load balancing을 유도할 수 있다^[5]6]. 본 연구에서는 동시에 지원 가능한 비디오 스트림의 수를 극대화하는데 목표를 두고 있으므로 탐색 시간이 짧은 고정 크기 블록을 사용한다.

다음은 본 연구의 주된 관심인 비디오 데이터를 적절한 디스크 그룹에 배치하는 방법에 대해 기술한다. 제안된 배치 정책의 기본 개념은 임의의 비디오 데이터에 대해 이를 저장할 수 있는 능력을 지닌 디스크 그룹들을 선정하고 이 중에서도 가장 검색 효율을 높일 수 있는 디스크 그룹에 비디오 데이터를 배치한다. 각 데이터마다 가장 적합한 디스크 그룹을 선택하기 위해 배치 알고리즘이 사용되는데 이 알고리즘에는 앞에서 소개된 데이터들의 특성들과 각 디스크 그룹들의 특성이 입력되고 그

결과로 한 개의 디스크 그룹이 선택된다. 이때 선택된 디스크 그룹에 해당 데이터를 배치하기 위해 uniform striping 정책이 사용된다. 앞의 II장에서 소개한 weighted striping 정책은 스트라이핑 그룹내의 디스크들의 성능이 다양할 경우 각 디스크들의 성능을 최대한 이용하기 위한 정책이었으나 본 배치 정책에서는 이미 같은 스트라이핑 그룹 내에 유사한 성능의 디스크들을 배치시켰으므로 이와 같은 환경 하에서는 weighted striping 정책을 사용하더라도 결국 uniform striping 정책과 거의 같은 배치 결과를 받게 되며 uniform striping 정책을 사용하므로써 배치 작업을 간단히 할 뿐만 아니라 연속된 블록들의 검색 시 탐색 시간과 회전 지연 시간도 상대적으로 줄일 수 있다. uniform striping 정책을 사용하므로써 얻게되는 또 하나의 장점은 RAID와 같은 redundant disk array 구조를 구성하기 쉽다는 것이다. 각 디스크 그룹마다 독립된 RAID 레벨 4나 5 구조를 갖도록 구성할 수 있는데 이때 어느 구조를 선택할 것인가는 각 서버의 환경에 따라 다를 수 있다. 쓰기 작업이 빈번히 발생하는 환경에서 RAID 레벨 4 구조를 사용하면 parity 디스크에 병목현상이 발생할 수 있으므로 RAID 레벨 5가 더 적합할 것이다.

본 연구에서 제안한 배치 정책에서는 저장될 비디오 데이터들에 대한 인기도가 주어진다 가정 하에서 시작한다. 이 인기도는 앞에서 언급한 바와 같이 3차 저장장치와 자기 디스크 사이에서 데이터를 교체하는 과정에서 얻어지는 정보로써 최근 이 데이터에 대한 고객들의 평균 요청 간격을 의미한다. 주문형 비디오 서버내의 저장 시스템에서는 주기적으로 혹은 필요에 따라 서비스할 데이터의 내용을 변경해야 하는데 이 작업이 빈번히 일어날수록 보다 정확한 인기도 값을 얻을 수 있다.

비디오 데이터를 적절한 디스크 배열에 배치하는 기본 알고리즘은 다음과 같으며 여기서 사용된 기호는 표2와 같다.

● 비디오 데이터에 대한 배치 알고리즘

단계 1 : 압축된 비디오 데이터들의 인기도에 비례하도록 가중치를 구한다.

즉, 비디오 데이터 j와 k가 각각 1분과 10분에 한번씩 요청이 발생하는 인기도를 가졌다면,

$$V_j = 10 \cdot V_k \text{ 이다.}$$

$$V_k = \text{video data } k \text{의 가중치, } 1 \leq k \leq N$$

단계 2 : 데이터들의 가중치 값을 내림차순으로

정렬한다.

Sort V_1, V_2, \dots, V_N in descending order

단계 3 : 가중치가 가장 큰 데이터부터 할당을 시작한다.

Select V_k with maximum weight in sorted list
(V_1, V_2, \dots, V_N)

i) 비디오 데이터 k에 대해 배치 가능한 디스크 그룹들을 모두 찾는 단계로서, 디스크 그룹 i에 대한 비디오 데이터 k의 저장 조건은 다음을 만족해야 한다.

(디스크 그룹 i에서 비디오 데이터 k를 단위 시간 동안 검색·전송하는 양 \geq 비디오 데이터의 peak 시의 재생률) and (디스크 그룹 i의 총 가용 공간 크기 \geq 비디오 데이터 k의 총 크기)

$$\text{if } (\min_{D_i} W_j \cdot P_i \geq P_{br_k}) \text{ and } (\sum_{D_i} US_{ij} \geq VS_k) \quad (2)$$

ii) 비디오 데이터 k에 대해 다음 R_k 를 계산한다.

$$R_k = IO_bandwidth_k \times \frac{Run_length_k}{V_k} \quad (3)$$

iii) i)에서 선택된 디스크 그룹들 중 다음 값이 최소인 그룹을 선택한다.

$$\min_{\text{디스크 그룹}} \left\{ \frac{\sum_k \text{디스크 그룹에 이미 할당된 데이터들의 } R_k}{\text{디스크 그룹 i의 IO bandwidth}} \right\} \quad (4)$$

iv) 비디오 데이터 k를 선택된 디스크 그룹 i에 uniform striping 정책을 사용하여 배치한다.

단계 3을 반복하여 다음 우선 순위의 비디오 데이터에 대해서도 디스크 그룹을 찾아 배치시킨다.

/ 비디오 데이터에 대한 배치 알고리즘 끝 /

위의 알고리즘을 살펴보면 비디오 데이터들에 대해 인기도가 높은 비디오부터 먼저 디스크 그룹을 선택하여 배치하도록 하고 있는데 단계 3을 보면 디스크 그룹의 선택 조건을 알 수 있다. 먼저 해당 데이터에 대해 저장 가능한 모든 그룹을 선택하고 이 중에서도 가장 검색 효율을 높일 수 있는 그룹을 최종적으로 선택하는데 그 조건은 단계 3의 ii), iii)에서와 같다. ii)에서 계산되는 R_k 는 데이터 k가

단위 시간에 여러 고객으로부터 평균적으로 요청되는 데이터 양이다. 예를 들어 총 상영 시간이 2시간이며 고객으로부터 평균적으로 1시간 간격으로 요청되는 비디오 데이터 k가 있다고 가정하고 이 비디오 데이터 k의 전송 대역폭을 4 Mbps 라고 하자. 비디오 데이터의 전송 대역폭이란 이 영화가 고객으로부터 한번 요청되었을 때 정상적으로 서비스되기 위해 단위 시간당 검색되어 전송할 평균 데이터 양을 의미한다. 이와 같은 비디오 데이터 k는 임의의 시간에 평균적으로 늘 두번의 요청을 받고 있다고 할 수 있고 이 데이터에 대해 임의의 시간에 초당 요청되는 데이터 양은 8 Mbits로 볼 수 있다. 이 수치가 바로 이 데이터 k의 R_k 가 되는 것이다. 즉 R_k 가 큰 데이터일수록 인기도가 높고 단위 시간당 요청되는 데이터 양도 많다는 것을 알 수 있다. R_k 가 구해지면 이 데이터에 대해 가장 높은 검색 성능을 제공할 수 있는 디스크 그룹을 선택하는데, iii)에서는 각 디스크 그룹마다 그 그룹에 이미 할당된 비디오 데이터들의 R_k 들의 총 합을 그 그룹의 입출력 대역폭 성능으로 나눈 값을 구해 이 값이 가장 작은 디스크 그룹을 최적의 그룹으로 선택한다. 즉 이 의미는 현재 해당 디스크 그룹의 입출력 대역폭 성능에 비해서 그 그룹이 단위 시간에 검색하여 전송해야 하는 평균 데이터 양이 가장 적은 디스크 그룹을 선택하자는 것이다.

표 2. 비디오 데이터의 배치에 대한 기호정의

· V_k - 비디오 데이터 k의 인기도에 따른 가중치
· VS_k - 비디오 데이터 k의 총 크기
· P_{br_k} - 비디오 데이터 k의 peak bit rate
· P_i - 디스크 그룹 i에서 현재 비디오 데이터 k의 단위블록 S_k 를 배치하는데 사용 가능한 디스크 수
· D_i - 디스크 그룹 i에서 현재 비디오 데이터 k를 배치하는데 사용 가능한 디스크들의 집합
· G_i - 현재 비디오 데이터 k를 배치할 수 있는 조건을 만족하는 디스크 그룹들의 집합
· US_{ij} - i번째 디스크 그룹 내의 j 번째 디스크의 사용 가능한 블록의 크기
· $IO_bandwidth_k$ - 비디오 데이터 k의 전송 대역폭
· Run_length_k - 비디오 데이터 k의 상영시간
· R_k - 비디오 데이터 k에 대해 단위시간당 평균적으로 요청되는 데이터 량

IV. 실험

1. 실험 방법 및 환경

이 장에서는 III장에서 제시한 배치 기법을 평가하기 위해 압축된 비디오 데이터와 다양한 크기의 디스크 배열을 설정하여 모의 실험을 수행한다.

실험에서 사용되는 비디오 데이터로는 [4]에서 얻은 MPEG-1으로 압축된 비디오 데이터들로서, 각 데이터들은 40000개의 프레임들로 구성되는데 이는 각각 약 30분 정도의 상영 분량이다. 데이터들을 분류해 보면 4편의 영화와 3편의 스포츠 경기, 2편의 만화 영화, 2편의 TV 토크쇼, 뉴스, 코미디 프로그램 등 15편의 비디오 데이터들인데, 이들에 대한 자세한 사양은 표 3과 같다[4].

표 3. 모의 실험에 사용되는 비디오 데이터

제목	압축률 X:1	평균 프레임 크기 (bits)	전송 대역폭 Mbps	peak 비트율 Mbps	평균요청 간격 (sec)
asterix	119	22,348	0.59	1.85	10
atp(tennis)	121	21,890	0.55	1.58	10
dino(movie)	203	13,078	0.33	1.01	10
lambs(movie)	363	7,312	0.18	0.85	20
mr.bean	150	17,647	0.44	1.76	20
mtv	134	19,780	0.49	2.71	20
news	173	15,358	0.38	2.23	30
race	86	30,749	0.77	3.24	30
settop	305	6,031	0.15	0.27	40
simpsons	143	18,576	0.46	1.49	60
soccer	106	25,110	0.63	2.29	300
starwars	130	15,599	0.36	4.24	600
talk show1	183	14,537	0.36	1.00	600
talk show2	148	17,914	0.49	1.40	1800
terminator	243	10,904	0.27	0.74	3600

제안된 배치 정책에서는 비디오 데이터의 인기도를 사용하게 되는데 실험의 편의를 위해 데이터들의 인기도는 표에서 나타난 평균 요청 간격에 의해 결정되고 비디오 데이터 k의 스트라이핑 단위 크기 Sk는 100KB로 한다. 이는 실험에 의해 선택된 값으로서, 디스크 배열의 크기가 10일때 스트라이핑 블록의 크기를 50KB~250KB로 증가시켜가면서 실험에서 비교한 3가지 배치정책들에 적용하여 보았다.

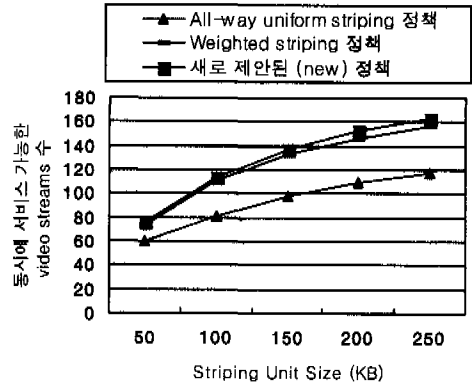


그림 5. 스트라이핑 블록의 크기에 따른 배치 정책들의 성능 비교

그 결과는 그림 5와 같은데 50KB~250KB 사이에서는 크기가 커질수록 세 정책 모두 검색 성능이 높아졌으며 스트라이핑 블록의 크기가 커질수록 기존 정책들에 비해 새로 제안된 배치 정책의 검색 성능이 점점 더 높아짐을 알 수 있었다. 따라서 본 논문의 실험에서는 기존의 정책들과의 사이에 비교적 성능 차이가 덜 나는 100KB를 단위 블록의 크기로 사용하였다.

또한 실험에서는 클러스터드 주문형 비디오 서버의 저장 시스템 내에 대형의 디스크 배열을 설정하기 위해 표4를 통해 보이고 있는 4 종류의 디스크들을 사용하는데, 제안된 배치 정책의 성능을 다양하게 평가하기 위해 디스크 배열의 크기를 10, 20, 30, 40, 50, 60으로 늘려가면서 그때마다의 성능 효율을 측정한다.

표 4. 디스크 배열을 구성하는 디스크의 종류

디스크 모델명	저장 능력 (MB)	평균 탐색 시간 (ms)	평균 회전 지연율 (ms)	단위 데이터의 전송시간 (ms/KB)
Elite9	9090	11	5.56	0.144
Medalist	631	14	7.87	0.2
Hawk	1050	9	5.54	0.142
Barracuda 1	2150	8	4.17	0.13

2. 실험 결과

본 절에서는 앞의 실험 환경을 세 가지 배치 정책에 적용한 결과를 보인다. 앞 절에서 제시한 비디오 데이터와 디스크들로 대형 디스크 배열을 구성

한 뒤, 본 논문에서 제안한 배치 정책과 기존의 정책들, 즉 all-way uniform striping과 weighted striping 정책에 적용시켜 검색 성능을 비교한다. 이들 세 정책을 사용하여 저장 시스템을 구성하였을 때, 데이터 요청 시 동시에 지원 가능한 비디오 스트림 수에 의해 성능을 비교해보면 그림 6과 같은 결과를 얻는다.

그림 6의 결과를 보면 all-way uniform striping 정책을 적용했을 때 가장 낮은 검색 성능을 보이고 있음을 알 수 있고 weighted striping 정책과 본 논문에서 제안한 배치 정책(NEW) 사이의 성능도 디스크 배열의 크기가 커질수록 차이가 더 나고 있음을 알 수 있다. 따라서 결과적으로 제안된 배치 정책을 사용하여 데이터를 저장하면 기존의 저장 시스템의 구조와 배치 정책을 사용했을 때 보다 검색 성능이 향상됨을 알 수 있다.

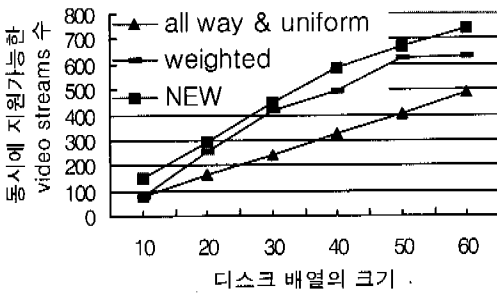


그림 6. 배치정책에 따른 검색성능의 비교

V. 결론

클러스터드 주문형 비디오 시스템의 저장 서버는 보통 수십 개 이상의 디스크들이 소프트웨어적으로 구성된 디스크 배열 형태로 구성된다. 이와 같이 대용량의 저장 시스템을 갖는 주문형 비디오 서버는 많은 클라이언트들의 요청에 대해 서비스하게 되지만 특히 인기있는 비디오 데이터일 경우에는 단 시간 내에도 많은 요청이 집중될 수 있으므로 동시에 보다 많은 비디오 스트림을 검색하여 서비스하는 일은 주문형 비디오 서버의 최고 목표 중의 하나일 수 있다.

따라서 본 연구에서는 이와 같은 검색 효율의 향상을 위해 비디오 데이터의 저장 시, 보다 효율적인 검색을 이루어낼 수 있도록 저장 시스템을 구성하고 이에 따른 배치 정책을 제안한다. 즉, 서버내의 대형 디스크 배열을 디스크 성능별로 나누어 그룹

화하고 데이터 배치 시에 디스크 그룹의 성능과 데이터들의 특성을 함께 고려하여 검색 효율을 높이고자 한다. 각 비디오 데이터들은 여러 디스크 그룹 중 한 그룹을 선택하고 그 안에서는 RAID에서의 block interleaving 방식으로 배치된다.

제안된 배치 정책은 실험 결과에서 제시한 바와 같이 기존의 배치 정책에 비해 뚜렷한 검색 성능의 향상을 가져왔으며 RAID 레벨 4나 5에서와 같은 redundant disk array를 구성하기 용이하게 하므로써 보다 신뢰성있는 저장 서버를 구성할 수 있다는 장점을 갖는다. 단, 본 실험에서는 기존의 데이터 배치 정책들과의 검색 성능 비교를 단순화하기 위해 RAID 레벨 4나 5에서와 같은 패리티 정보의 저장은 고려하지 않았으나, 실제 저장 서버의 설계 시에는 서버의 신뢰성을 위해 환경에 따라 오류 정정을 위한 적절한 정보 관리가 중요한 고려 사항이 된다.

참고 문헌

- [1] 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.
- [2] Renu Tewari, Daniel M. Dias, Rajat Mukherjee, Harrick M. Vin, "High Availability in Clustered Multimedia Servers", Proceedings of the USENIX 1996 Annual Technical Conference, Jan. 1996.
- [3] Prashant J. Shenoy, Harrick M. Vin, "Efficient Striping Techniques for Multimedia File Servers", Univ. of Texas at Austin, Technical Report CS-TR-96-27, pp. 1-21, 1997.
- [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, Harrick M. Vin, "Symphony : An Integrated Multimedia File System", Technical Report TR-97-09, pp. 1-16, 1997.
- [6] Mathias Rantenberg, Helmut Rzehak, Interactive Distributed Multimedia Systems and Services, Springer, pp. 265-268, 1996.
- [7] 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, 1994.

안 유 정(You-jung Ahn)

정회원



1991년 : 이화여자대학교

전자계산학과 학사

1991~93년 : 금융결제원 재직

1995년 : 이화여자대학교

전자계산학과 석사

1995~현재 : 홍익대학교 컴퓨터

공학과 박사 과정

<주관심 분야> 멀티미디어 시스템

원 유 현(Yoo-hun Won)

정회원

1972년 : 성균관대학교 수학과 학사

1975년 : 한국과학원 전자계산학과 석사

1985년 : 고려대학교 이학박사

1975~76년 : 한국과학기술연구소 연구원

1986~87년 : R.P.I. 교환 교수

1976~현재 : 홍익대학교 컴퓨터공학과 교수

<주관심 분야> 프로그래밍 언어론, 멀티미디어 시스템, 실시간 언어