

클러스터드 주문형 비디오 시스템에서 효율적인 저장 기법

김 종 훈*

An Efficient Storage Technique in Clustered VOD Systems
Jong-Hoon Kim

— < 목 차 > —

- I. 서 론
- II. 주문형 비디오 시스템의 환경
- III. 효율적인 저장 기법
- IV. 실험
- V. 결 론

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

* 제주교육대학교 컴퓨터교육학과 전임강사

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. 서 론

주문형 비디오 서버에서 단위 시간에 보다 많은 클라이언트들에게 그들이 원하는 서비스를 제공하기 위해서는 여러 가지 요인들이 함께 고려되어야 하는데, network 환경, 버퍼링(buffering) 정책, 디스크 스케줄링(disk scheduling) 정책, 배치(placement) 정책 등이 그것이다. 본 연구에서는 이중에서도 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. 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 차가 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 영상 압축 데이터의 특성을 함께 고려하여 검색 효율을 높이고자 한다.

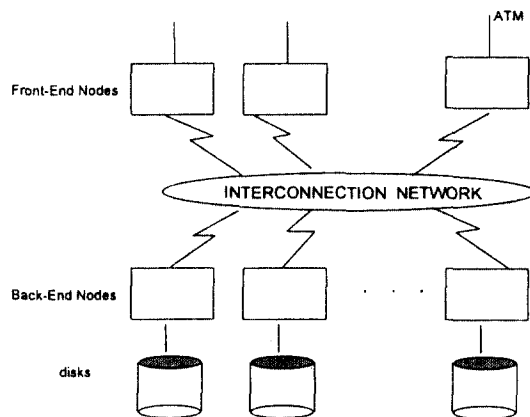
본 논문의 구성을 살펴보면, II장에서는 주문형 비디오 서버의 환경 중에서도 본 연구의 배경이 되고 있는 클러스터드 주문형 비디오 서버와 기존의 배치 정책들에

대해 간략히 기술하고 III장에서는 디스크들의 그룹화 방법과 그에 따라 저장될 압축 영상 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 시스템의 환경

1. 클러스터드 주문형 비디오 서버의 구조

클러스터드 비디오 서버는 [그림 1]에서와 같이 상호 연결된 다수의 노드와 디스크들로 구성된 2-계층 구조를 이루는 것이 보통인데, front-end 노드는 전송을 담당하고 back-end 노드는 저장을 담당한다. 클라이언트로부터 비디오 데이터에 대한 요청이 들어오면 front-end 노드에서 승인 여부를 결정한 뒤, 요청을 스트림 형태로 만들어 네트워크를 통해 back-end 노드로 전달한다. 각 비디오 객체들은 블록 단위로 나뉘어져 시스템내의 여러 디스크들에 걸쳐 분산되어 저장되는데, 스트림 형태의 요청이 들어오면 back-end 노드에서는 해당 데이터를 검색하여 front-end 노드로 전송한다. front-end 노드에서는 스트림 버퍼가 존재하여 전송 받은 데이터를 임시로 저장하며 비디오 데이터를 요구되는 재생률(playback rate)로 외부 네트워크를 통해 해당 클라이언트에게 전송한다[2].



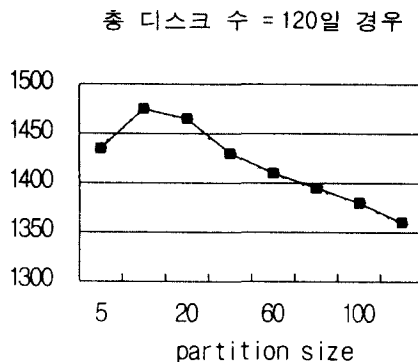
[그림 1] 클러스터드 비디오 서버의 2-계층 구조

이와 같은 환경 하에서 현재 디스크의 속도와 네트워크의 속도만을 놓고 본다면 하나의 클라이언트가 원하는 정보를 검색하여 전달하기에는 충분하다. 하지만 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 경향 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 배열내의 다양한 디스크들의 성능에 관계없이 저장하고자 하는 데이터를 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.



[그림 2] partition size와 동시 지원 가능한 고객 수와의 관계

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. 참고로 [3]에서는 디스크 그룹의 크기(partition size)에 따라 동시에 지원 가능한 고객의 수를 실험에 의해 보이고 있는데 그 결과는 [그림 2]와 같다. 이 그림에서도 알 수 있는 바와 같이 디스크 그룹의 크기가 수십 개에서 수백 개에 이르게 되면 오히려 동시에 지원 가능한 스트림의 수는 떨어지는 것을 알 수 있다. 이와 같은 배치 기법들의 문제 해결을 위해 weighted striping이라는 새로운 배치 기법이 제안되었다. 이 기법은 디스크마다 그 성능을 기준으로 가중치를 계산하고 가중치 값에 비례하는 데이터 양을 디스크에 할당하도록 하는 스트라이핑 기법이다. 이 방법을 통해 디스크들은 각기 성능에 따라 작업 부하가 조절되므로 디스크 이용률을 더욱 높일 수 있게 되었고 그에 따라 기존의 배치 방법보다 더 많은 스트림들을 처리할 수 있게 되었다 [1]. 따라서 본 연구에서는 이 weighted striping 기법을 응용하여 대형 디스크 배열 구조의 저장 시스템에 적용시키고자 한다.

Ⅲ. 효율적인 저장 기법

1. 연구 동기

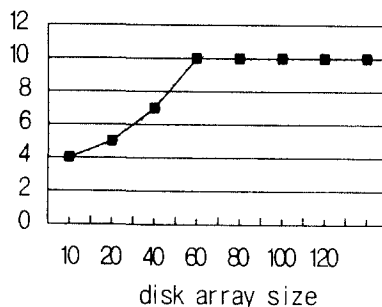
주문형 비디오 서버에서 클라이언트들에게 요구되는 재생률로 영상 데이터를 제공하기 위해서는 압축하여 저장하는 것이 필수적이거나 압축 후에도 그 크기는 작지 않다. 한 예로 MPEG-2로 압축된 2시간 짜리 영상 데이터인 경우 약 4GB 정도의 용량을 차지한다. 따라서 서버내의 저장 시스템을 생각해 보면 기존에 나와 있는 디스크들의 용량으로는 수십 내지 수백 개의 디스크가 필요하다고 볼 수 있다.

이러한 다수의 디스크들은 디스크 배열의 형태로 존재하는데, 저장 시스템 역시 계속 변화와 확장을 거듭하게 되므로 다양한 성능의 디스크들이 공존하게 된다. 또한 여기에 저장되는 영상 압축 데이터 역시 다양한 특성을 가지게 되는데 검색시 요구되는 평균 재생률(playback rate)이나 peak시의 재생률 그리고 인기도 등이 각

기 다르다. 따라서 주문형 비디오 서버가 보다 많은 클라이언트들에게 서비스하기 위해서는 디스크들의 다양한 성능과 함께 저장될 데이터들의 다양한 특성도 함께 고려하여 저장하는 것이 필요하다고 본다.

2. 디스크 배열의 구성

앞 절에서도 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 영상 데이터들을 striping 방식[8]으로 배치시킨다. 이때 디스크 분할 방법은 디스크들의 입출력 대역폭 성능이 유사한 것들을 같은 그룹으로 묶는데 이렇게 함으로써 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 3]과 같다.



[그림 3] 디스크 배열의 크기에 따른 최적의 Partition size

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 조정된다.

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, 이때 사용되는 기호에 대한 정의는 <표 1>과 같다.

<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 (|  $W_i - W_{i+1}$  | ≤  $\epsilon$ )
```

```

    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)
    if Partition_Group_Size[i] = ( $\delta \cdot \text{Optimal\_Partition\_Size}[D] + \rho$ )
    {
        if  $\rho < \theta [D]$ 
            Partition Partition_Group Partition_Group[i] into  $\delta$  groups
        [else
            Partition Partition_Group Partition_Group[i] into  $\delta + 1$  groups
        ]
    }

```

3. 영상 압축 데이터의 배치

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

현재 영상 데이터를 압축하는 가장 대표적인 표준으로는 MPEG을 들 수 있다. MPEG 표준과 같이 가변 비트율 압축 기법에 의해 압축된 영상 데이터들은 프레임 단위로 인코드되어 디스크 상에 저장된다. 이 프레임들은 I 프레임, P 프레임, B 프레임으로 구성되며 이들의 크기는 모두 다르다[9]. 이와 같이 다양한 크기를 갖는 프레임들을 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 검색 시간이 상대적으로 짧다는 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 검색 시간이 짧은 고정 크기 블록을 사용한다.

다음은 본 연구의 주된 관심인 영상 압축 데이터를 적절한 디스크 그룹에 배치하는 방법에 대해 기술한다. 제안된 배치 정책의 기본 개념은 임의의 영상 데이터에 대해 이를 저장할 수 있는 능력을 지닌 디스크 그룹들을 선정하고 이 중에서도 가장 입출력 성능이 뛰어난 디스크 그룹에 영상 데이터를 할당하자는 것인데, 본 배치 정책에서는 가장 인기도가 높은 데이터부터 할당을 해 나감으로 클라이언트로부터 잦은 요청이 발생하는 인기있는 영상 데이터일수록 더 빠른 검색과 전송을 할 수 있는 디스크 그룹에 할당되게 된다. 단, 본 연구에서 제안한 배치 정책에서는 저장될 영상 데이터들에 대한 인기도가 주어진다 가정하에서 시작한다. 주문형 비디오 서버내의 저장 시스템에서는 주기적으로 혹은 필요에 따라 서비스할 데이터의 내용을 변경하거나 확장해야 하므로 영상 데이터들에 대한 인기도는 이전의 서비스 제공 결과를 이용하거나 사용자들에 대한 경향 예측으로 구해질 수 있다.

영상 압축 데이터를 적절한 디스크 배열에 배치하는 기본 알고리즘은 다음과 같으며 여기서 사용된 기호는 <표 2>와 같다.

<표 2> 영상 데이터의 배치에 대한 기호정의

- V_k : 영상 데이터 k 의 인기도에 따른 가중치
- S_k : 영상 데이터 k 의 스트라이핑 단위 크기
- VS_k : 영상 데이터 k 의 총 크기
- Pbr_k : 영상 데이터 k 의 peak bit rate
- P_i : 디스크 그룹 i 에서 현재 영상 데이터 k 의 단위 블록 S_k 를 배치하는데 사용 가능한 디스크 수
- D_i : 디스크 그룹 i 에서 현재 영상 데이터 k 를 배치하는데 사용 가능한 디스크들의 집합
- G_i : 현재 영상 데이터 k 를 배치할 수 있는 조건을 만족하는 디스크 그룹들의 집합
- US_{ij} : i 번째 디스크 그룹 내의 j 번째 디스크의 사용 가능한 블록의 크기

영상 데이터에 대한 배치 알고리즘

단계 1 : 압축된 영상 데이터들의 인기도에 비례하도록 가중치를 구한다. 즉, 영상 데이터 j 와 k 가 각각 1분과 10분에 한번씩 요청이 발생하는 인기도를 가졌다면, $W_j = 10 \cdot W_k$ 이다. $V_k = \text{weight of video data } k, 1 \leq k \leq N$

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) 영상 데이터 k 에 대해 배치 가능한 디스크 그룹들을 모두 찾는 단계로서, 디스크 그룹 i 에 대한 영상 데이터 k 의 저장 조건은 다음을 만족해야 한다. (디스크 그룹 i 에서 영상 데이터 k 를 단위 시간동안 검색·전송하는 양 \geq 영상 데이터의 peak시의 재생률) & (디스크 그룹 i 의 총 가용 공간 크기 \geq 비디오 데이터 k 의 총 크기)

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

ii) 영상 데이터 k 를 배치 가능한 디스크 그룹들 중 입출력 대역폭 성능이 가장 뛰어난 디스크 그룹을 선택한다.

$$\max_{G_i \ni i} \{ (\min_{D_i \ni j} W_j) \cdot P_i \} \quad (3)$$

iii) 영상 데이터 k 를 선택된 디스크 그룹 i 에 배치하는데, 이때 weighted striping 정책을 사용한다. 즉, 각 디스크들의 입출력 대역폭 성능에 비해 현재 저장되어 있는 데이터의 양이 가장 작은 디스크를 선택하여 다음 단위 블록을 저장한다[1].

$$\min_{D_i \ni j} \{ C_j \cdot W_j \} \quad (4)$$

단, C_j 는 디스크 j 에 이미 할당되어 있는 데이터 양

(4)의 조건을 만족하는 디스크 j 에 영상 데이터 k 의 다음 블록을 배치한다.

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

IV. 실험

1. 실험 방법 및 환경

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

<표 2> 모의 실험에 사용되는 영상 압축 데이터

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

실험에서 사용되는 영상 데이터로는 [4]에서 얻은 MPEG-1으로 압축된 영상 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 프로 등 15편의 영상 데이터들인데, 이들에 대한 자세한 사양은 <표 3>과 같다[4]. 제안된 배치 정책에서는 영상 데이터의 인기도를 사용하게 되는데 실험의 편의를 위해 데이터들의 인기도는 표에서 나타난 순서와 같다고 가정하고 영상 데이터 k 의 스트라이핑 단위 크기 S_k 는 100 KB로 한다. 또한 실험에서는 클러스터드 주문형 비디오 서버의 저장 시스템 내에 대형의 디스크 배열을 설정하기 위해 <표 4>를 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 3> 디스크 배열을 구성하는 디스크의 종류

디스크모델명	저장능력 (MB)	평균 탐색 시간 (ms)	평균회전율 (ms)	단위데이터의 전송시간 (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

여기서는 대표적으로 디스크 배열의 크기가 20 일 때, 제안된 배치 정책의 과정에 대해 설명하고 마지막으로 전체적인 실험 결과를 보이도록 한다.

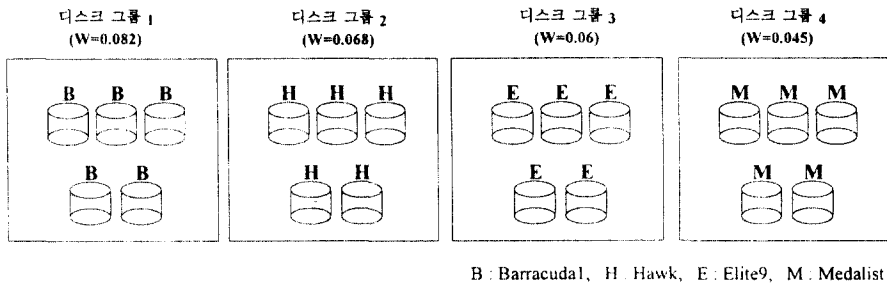
· 디스크 배열의 크기가 20 일 때 배치 정책의 예

실험에 사용되는 디스크 배열은 <표 4>에 소개된 디스크들으로써 각각 5개 씩 총 20개의 디스크를 가정한다. 먼저 실험에 사용된 디스크 배열에 대해 디스크 그룹화 작업을 수행하면 각 디스크들의 종류마다 입출력 대역폭을 기준으로 한 가중치를 구해야 하는데 Elite9, Medalist, Hawk, Barracuda1의 순으로 각각 0.06, 0.045, 0.068, 0.082 이 되며 이때 최적의 디스크 그룹의 크기는 5가 되므로([그림 3] 참조)

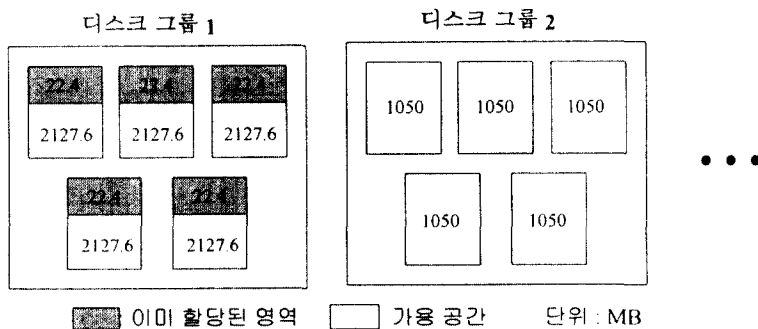
최종적으로는 [그림 4]와 같은 모양의 디스크 그룹화 작업이 이루어진다.

다음에는 배치 작업을 수행하기 위해 <표 3>에서 가장 인기 있다고 가정한 asterix를 제일 먼저 저장 대상으로 선택한다. 이 실험에서 사용한 asterix 데이터의 총 크기는 프레임의 평균 크기 X 프레임 수로 계산했을 때 22,348(bits/frame) X 40000 (frames) ≃ 112 MB 정도이며 peak bit rate=1.85 (Mbps)이다.

이 데이터를 배치시킬 수 있는 디스크 그룹들을 선정하기 위해 III장에서 제시한 식 (2)에 각 디스크들의 성능 수치를 대입시켜 그룹들을 선정한 뒤, 식 (3)을 사용해 이 중에서도 가장 입출력 대역폭 성능이 뛰어난 그룹을 선택하여 여기에다 영상 데이터를 배치시킨다. 현재는 아직 어떤 데이터도 저장되어 있지 않은 상태이므로 당연히 모든 그룹들 중 가장 성능이 우수한 디스크 그룹 1에 배치시킨다. 디스크 그룹 1에 asterix를 배치시키고 난 후의 상태가 [그림 5]이다.



[그림 4] 디스크 배열의 그룹화 예(디스크 배열의 크기=20일 때)



[그림 5] asterix의 배치 후 디스크 그룹 상태

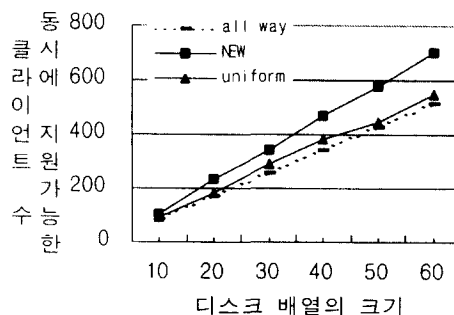
두 번째로 배치시킬 영상 데이터는 스포츠 경기인 atp(tennis)인데 현재 디스크 배열의 상태는 [그림 5]와 같으므로 식 (2)과 (3)에 의해 가장 유력한 디스크 그룹으로 꼽을 수 있는 것이 디스크 그룹 1과 2이다. 이때 현재 상태에서 이 두 디스크 그룹에 대한 성능을 식 (3)에 의해 비교해 볼 수 있는데 그 결과는 다음과 같다.

디스크 그룹 1의 현재 성능은 $\{\min_{D1 \ni j} W_j\} \cdot P_1 = 0.082 \times 5 = 0.41$ 이고, 그룹 2의 성능은 $0.068 \times 5 = 0.34$ 이므로 역시 디스크 그룹 1이 선택된다.

2. 실험 결과

마지막으로 앞에서 기술한 실험 환경을 세 가지 배치 정책에 적용한 결과를 보인다. 즉, 본 논문에서 제안한 배치 정책과 모든 디스크들을 하나의 대형 디스크 배열로 구성하여 배치시키는 all way striping 정책 그리고 디스크 성능의 고려 없이 영상 데이터 일정량을 항상 차례대로 디스크마다 배치시키는 uniform striping 정책을 실험을 통해 비교한다.

이들 세 정책을 사용하여 저장 시스템을 구성하였을 때, 데이터 요청시 동시에 지원 가능한 클라이언트의 수를 구하면 [그림 6]과 같은 결과를 얻는다. 따라서 결과적으로 본 연구에서 제안한 디스크 배열의 구조와 배치 정책을 사용했을 때 검색 효율이 향상됨을 알 수 있다.



[그림 6] 다양한 배치 정책이 동시에 지원 가능한 클라이언트 수의 비교

V. 결 론

클러스터드 주문형 비디오 서버는 보통 수십 개 이상의 단일 디스크 서버들로 구성되므로 서버내의 저장 시스템 역시 대형의 디스크 배열 형태를 이룬다. 이와 같이 대용량의 저장 시스템을 갖는 주문형 비디오 서버는 많은 클라이언트들의 요청에 대해 서비스하게 되지만 특히 인기 있는 영상 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 영상 데이터의 저장시, 보다 효율적인 검색을 이루어 낼 수 있도록 저장 시스템을 구성하고 이에 따른 배치 정책을 제안한다. 즉, 서버내의 대형 디스크 배열을 디스크 성능별로 나누어 그룹화하고 성능이 우수한 디스크 그룹일수록 자주 요청되는 영상 데이터들을 배치함으로써 데이터의 검색 효율을 높이도록 한다. 또 영상 데이터 저장 시에도 선택된 디스크 그룹내의 디스크들마다 성능을 비교하여 상대적으로 효율이 높은 디스크에 먼저 배치함으로써 보다 효율적인 검색을 지원하도록 한다. 이와 같은 배치 정책을 모의 실험으로 통해 수행해 본 결과 IV장에서 제시한 바와 같이 기존의 배치 정책에 비해 뚜렷한 성능 향상을 가져왔음을 알 수 있다.

◆ 참고 문헌 ◆

1. Yuewei Wang, David H. C. Du, "Weighted Striping in Multimedia Servers," IEEE multimedia systems, pages 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, pages 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, pages 1-16, 1997.
6. Mathias Rantenberg, Helmut Rzehak, Interactive Distributed Multimedia Systems and Services, Springer, pages 265-268, 1996.
7. Andrew Heybey, Mark Sullivan, Paul England, "Calliope : A Distributed, Scalable Multimedia Server," Proceedings of the USENIX 1996 Annual Technical Conference, Jan. 1996.
8. Peter M. Chen, Edward K. Lee, David A. Peterson, "RAID: High-Performance, Reliable Secondary Storage," ACM Computing Surveys, 26(2): 152-158, June 1994.
9. 정제창, 최신 MPEG, 교보문고, pages 117-198, 1997.