

# MZR 디스크를 채택한 VOD 저장서버의 효율적인 비디오 블록 배치방법

임 형 룡<sup>†</sup> · 김 정 원<sup>††</sup> · 김 영 주<sup>††</sup> · 정 기 동<sup>†††</sup>

## 요 약

본 논문에서는 MZR 디스크의 특성과 VOD에서 사용자들의 접근 편기성을 고려한 비디오 블록의 효율적 배치방법을 제안한다. 또 제안된 방법에 대하여 VOD 서버의 모델링과 모의실험을 통하여 성능을 평가하였다. 기본적인 방법은 인기도에 따라 Zipf 분포를 고려하여 MZR 디스크상에 선형배치나 편기된 휴리스틱(SH) 방법으로 배치 시키는 것이다. 제안된 방법의 타당성을 검증하기 위해서, 2 대의 디스크를 설치한 워크스테이션에서 다양한 편기요인 하에서의 성능을 실험하였다. 실험결과 여기에서 제시한 방법이 임의배치 방법보다 좋은 성능을 보여주고 있음을 알 수 있다. 디스크 그룹으로 배치방법을 확장하기 위하여 동시에 지원 가능한 최대 사용자수와 사용자당 필요한 버퍼의 크기를 분석하였다. 제안된 방법의 수행력에 관련한 변수로는 디스크헤드 스케줄링 방법, 배치방법, 스트라이핑 단위 등을 고려하였으며 실험결과 본 논문에서 제시한 방법이 효율적임을 알 수 있다.

## An Effective Video Block Placement Strategy on VOD Storage Server with MZR Disks

Hyoung-Roung Lim<sup>†</sup> · Jeong Won Kim<sup>††</sup> · Young Ju Kim<sup>††</sup> · Ki Dong Chung<sup>†††</sup>

## ABSTRACT

In this paper, we propose an efficient video block placement scheme that utilize the current disk product that has MZR disk characteristic and users' skewed access pattern on VOD. Also, we evaluate its performance through simulation and modeling of VOD server. The basic placement rule is to place on MZR disks by LP, SHP methods according to the Zipf distribution of popularity. To verify the proposed scheme, we examined its performance on workstation with 2 MZR disks under varied skewed factors. The proposed placement scheme showed better response time than the random method. To extend proposed placement scheme to disk group, we analyzed the theoretical maximum numbers of concurrent users and the required buffer size per user. For performance parameters for the proposed scheme, we considered the disk head scheduling methods, the placement methods and the striping unit. The result of experiments showed that the proposed scheme was effective.

† 정 회 원: 시스템공학연구소

†† 준 회 원: 부산대학교 전자계산학과

††† 총신회원: 부산대학교 전자계산학과

논문접수: 1997년 6월 24일, 심사완료: 1997년 11월 4일

## 1. 서 론

최근 정보기술의 눈부신 발달로 인하여 멀티미디어 데이터를 표준규격으로 처리하여 컴퓨터에 저장하였다가 초고속 망을 통하여 원거리의 사용자에게 서비스할 수 있는 VOD(Video On Demand) 서버가 제안되고 있다. VOD 서버는 주문형 비디오, 홈 쇼핑, 화상회의, 원격교육 및 진료, 전자 도서관, 주문형 신문 등과 같은 많은 응용 분야에서 사용되어진다. 보통 MPEG-2로 압축된 90 분의 비디오 데이터인 경우에 약 2 GB의 저장 공간이 필요하며 한 명의 사용자를 위해 2Mbps 이상<sup>1)</sup>의 전송 대역폭이 요구된다[1]. 따라서 VOD 서버는 다수의 비디오 데이터를 저장하고 동시에 많은 사용자의 요구를 서비스하기 위해서는 대량의 저장공간과 높은 전송률을 보장하는 시스템 구조를 가져야 한다. 그리고 제한된 VOD 서버의 저장능력과 전송능력을 최대한 활용하여 많은 수의 사용자를 동시에 서비스할 수 있는 효율적인 비디오 데이터 배치방법이 필요하다. 또한 VOD 서버에서 사용자들이 비디오 데이터에 접근하는 형태를 살펴보면, 모든 비디오에 균등하게 접근되지 않고 특정 비디오에 편기 되어 된다. VOD에서 사용자들의 비디오에 대한 접근 빈도수는 일반적으로 인기도가 상위 50%에 속하는 비디오에 대한 접근 요구가 전체 요구의 80~90%를 차지하고 있다.

본 논문의 주제와 관련된 기존 연구로는 하나의 디스크에서 저장된 오브젝트의 특성에 관계없이 지속적인 전송률을 보장하여 부하 균형을 제공하고자 하는 logical tracks과 track pairing[6] 방법이 있다. 이 방법들은 비디오 블록의 접근 빈도에 상관없이 지속적인 전송률을 보장하는데 MZR디스크의 가변적인 전송률을 고정적인 전송률로 서비스하는 방법이다. 그러나 액세스 빈도가 높은 비디오 블록들에 편기가 발생할 때 전송률이 낮은 안쪽 존에 배치가가능성이 있으므로 MZR디스크의 대역폭을 충분히 이용할 수 없다. 그리고 여러 개의 디스크에서 오브젝트가 할당받은 대역폭과 저장공간 사이에 균형을 이루고자 하는 DSR(Dynamic Segmentation Replication)[7], BSR

(Bandwidth to Space Ratio)[8] 방법 등이 있다. 지금까지 연구된 이러한 방법들에서는 멀티미디어 오브젝트의 인기도와 MZR 디스크 존의 가변적인 전송률과의 관계가 전혀 고려되지 않았다.

따라서, 본 논문은 VOD 서버에서 비디오 데이터를 MZR 디스크에 배치할 때 사용자의 비디오에 대한 접근 형태와 MZR 디스크의 특성을 고려하여 접근 빈도가 높은 비디오 블록에 대한 디스크 헤드의 탐색 시간(seek time)을 확률적으로 감소시키는 비디오 블록배치 알고리즘을 제안한다. 제시된 알고리즘의 효과를 검증하기 위하여 MZR 디스크 상에서 성능을 측정하였다. 또 이를 더욱 발전시켜 디스크 배열 상에서 서비스 라운드 시간, 최대 지원 가능한 사용자의 수, 요구되는 버퍼 크기에 대한 수치적인 식을 유도하고 VOD 서버를 모델링하여 모의실험을 행하고 그래프를 통하여 결과를 나타내었다.

## 2. 연구 배경

본 연구는 내용기반 동영상 검색 시스템의 비디오 저장서버의 설계를 위한 선행연구로, 저장 서버에 도착하는 새로운 사용자 요구의 허가여부를 결정하는 허용제어(Admission control)의 중요한 변수가 된다. 본 실험에서는 접근형태의 편기성(skew)을 변화시키면서 탐색시간(seek time)에 대한 전송시간(transfer time)의 비율을 측정하였다. 또 MZR 디스크에서 존의 전송률을 고려한 새로운 배치방법을 적용했을 때 성능 향상이 있는지를 확인하기 위한 실험을 하였다. 또 MZR 디스크의 특성을 검사하기 위해 워크스테이션<sup>2)</sup>에 MZR의 특성을 갖는 Quantum Fireball 540S 디스크[9]와 동일한 크기의 CAD 디스크<sup>3)</sup>를 설치하여 1,000개 블록을 1,000회 읽는 raw 디스크 읽기 연산을 수행하였다(표 1)).

실험 결과 MZR 특성을 갖는 디스크에서는 바깥쪽 트랙으로 갈수록 응답시간이 선형적으로 증가하는 특성이 있음을 알 수 있다. 따라서 비디오 블록의 접근 빈도수와 디스크 존의 전송률을 고려하여 비디오 블록을 MZR 디스크에 배치하는 방법이 필요하게 된다.

1) 프레임 크기는 780×480이며 초당 30프레임을 상영할 경우임.

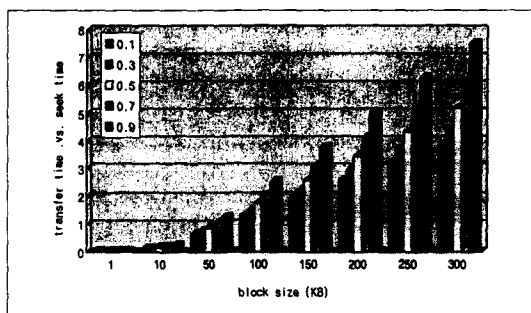
2) 모델명: Axil 320, CPU: Superscalar SPARC(60Mhz), 메모리: 32MB

3) 모델명: Digital DSP 3053L

<표 1> CLD, CAD 디스크에서의 읽기연산 결과

Disk	Quantum Fireball 540S	일반 CAD 디스크
Outermost Cylinder	54.1ms	125.3ms
Middle Cylinder	94.5ms	123.9ms
Innermost Cylinder	130.2ms	124.4ms

(그림 1)은 MZR 디스크에서 블록크기에 따른 탐색시간과 전송시간의 비율을 Zipf 분포<sup>4)</sup>로 표현한 그래프이다. 이 실험에서 바깥쪽 실린더가 접근확률이 높게 된 이유는 접근빈도가 높은 비디오 블록을 바깥쪽 존에 배치했기 때문이며, 또한 바깥쪽 존의 저장용량이 안쪽 존에 비해 2배 이상이기 때문이다. 일반적으로 디스크의 응답시간 중에서 탐색시간이 전체 서비스 시간의 많은 부분을 차지하고 있다. (그림 1)에서는 블록크기가 10KB 이하인 경우에는 전송시간이 탐색시간에 비해 적은 부분을 차지한다. 그러나 블록의 크기가 50KB 이상이 되면 전체 서비스 시간 중에서 전송시간이 차지하는 비율이 탐색시간을 초과하게 된다. 따라서 MZR 디스크와 같이 존간의 전송률이 다른 경우에 전송률을 고려하여 비디오 블록을 배치해야 한다. 또한 RAID3에서 사용하는 바이트 단위의 스트라이핑보다는 RAID5와 같은 블록단위의 스트라이핑이 탐색시간의 감소로 인해 좋은 응답시간을 얻을 수 있게 된다[10]. 스트라이핑 단위가 커지면 필요한 버퍼크기가 증가하여 시스템 가격이 올



(그림 1) 블록크기에 따른 전송시간 대 탐색시간의 비율

라하지만 최근의 메모리 가격의 하락 등에 비추어 볼 때 블록 단위의 스트라이핑 방법이 바람직한 방향이라 할 수 있겠다.

### 3. MZR 디스크에서 비디오 블록 배치 방법

#### 3.1 접근 프로파일과 성질

사용자의 접근 빈도와 MZR 디스크의 가변적인 전송률을 바탕으로 비디오 블록의 새로운 배치 방법을 제안하기 위해 두 가지 사항을 고려한다. 첫째는 사용자의 접근 형태를 나타내는 접근 프로파일(Access profile)이고, 둘째는 MZR 디스크의 특성들이다.

본 논문에서는 과거에 사용자들이 사용한 누적된 정보나 Zipf 분포에 의해 결정되는 비디오별 인기도를 나타내는 접근 프로파일<sup>5)</sup>이 있다고 가정한다. <표 2>는 일정 시간동안 100명의 사용자들이 10편의 비디오에 접근한 횟수를 나타낸 것으로 비디오의 인기도를 반영하고 있다. 접근 프로파일에서 예측한 블록의 접근 확률이 디스크 큐에 들어 있는 각 비디오 블록의 분포와 비슷하다<sup>5)</sup>면 디스크에 비디오 블록을 배치할 때 접근 빈도수를 이용하여 탐색시간을 최대한 확률적으로 감소시킬 수 있다. 즉, 같은 비디오의 블록들은 접근 프로파일의 확률에 따라 인접한 실린더에 배치되며 탐색시간을 줄일 수 있게 된다.

<표 2> 접근 프로파일의 예

인덱스	1	2	3	4	5	6	7	8	9	10
접근빈도수	39	17	11	8	6	5	4	4	3	3
확률	0.39	0.17	0.11	0.08	0.06	0.05	0.04	0.04	0.03	0.03

- 성질 1:  $AP_i \equiv DQ_i$ ,  
 $AP_i$ :  $i$ 번째 비디오가 접근될 확률  
 $DQ_i$ :  $i$ 번째 비디오에 대한 요청이 디스크에 있을 확률

일반적으로 CAD 디스크는 모든 실린더가 동일한

4) zipf 분포값이 작을수록 바깥쪽 실린더가 접근될 확률이 높다

5) 접근 확률과 디스크 큐에서의 대기 확률에 대한 실험 결과 Zipf분포의 값이 작을 경우에는 두 확률의 값에는 차이가 없으며, 값이 클 경우에는 30% 정도의 차이가 있다. 그리고 값이 0.5 미만일 경우에는 두 확률 값은 10% 정도의 오차가 있다.

저장 공간을 가지므로 디스크의 헤드가 확률적으로 빈번하게 위치할 실린더( $MAX(P_{head}^i)$ )는 전체 실린더의 중앙( $1/2$ )이 된다.

$N$ : 전체 실린더 수

$P_{head}^i$ : 디스크 헤드가  $i$ 번째 실린더에 위치할 확률,

$$\sum_{i=1}^N P_{head}^i = 1$$

$$MAX(P_{head}^i), i = \frac{N}{2}$$

그러나 MZR 디스크에서는 외곽쪽 존이 많은 섹터 수를 가지므로 헤드가 위치할 실린더는 중간 트랙에서 외곽쪽 트랙으로 이동된 위치일 것이다. 따라서 본 논문에서는 디스크 헤드가 빈번히 위치할 실린더 위치(HC)로 전체 섹터중에서 번호순으로 가운데 번호의 섹터를 포함한 실린더로 정한다.

- 성질 2:  $HC = MAX(P_{head}^i) = C_{\frac{N}{2}}$ , ( $N$ : 전체 섹터 수)

$C_i$ :  $i$ 번째 섹터가 위치한 실린더

본 논문에서는 성질 1, 2를 바탕으로 선형(linear) 배치와 편기된 휴리스틱(SH) 배치 방법을 제안한다.

### 3.2 선형배치

일반적으로 VOD에서는 최신 비디오가 인기 있으므로<sup>6)</sup> 저장 서버는 비디오 데이터를 저장할 때 이러한 특징을 고려하여 저장공간과 대역폭이 큰 존에 배치하여야 한다. 그리고 VOD 사용자의 대부분은 비디오를 연속해서 시청하므로 사용자의 요구는 실 시간성을 갖는다. 따라서 사용자들의 요구는 시간에 따라 변하는 서비스 라운드의 집합이며, VOD 서버는 서비스 라운드를 실시간에 처리할 수 있어야 한다. 설명

을 위해 이 장에서는 디스크 스케줄링 방법으로 C-SCAN 방법을 사용한다고 가정하여 다음 기호를 정의한다.

- $R_{length}$ : 임의의 순간에 하나의 서비스 라운드 내에 있는 서비스 수
- $T_{round}^i$ :  $i$ 번째 서비스 라운드의 서비스 시간

선형배치 방법의 설명을 간단히 하기 위해 모든 서비스 라운드의  $R_{length}$ 가 같다고 가정하자. 인기 있는 비디오 블록들을 존의 특성을 고려하지 않고 임의(random)로 MZR 디스크에 배치한다면  $T_{round}^i$ 이 존의 특성을 고려한 배치방법의  $T_{round}^i$ 보다 클 것이다. 이것은 임의로 배치된 인기 있는 블록들이 낮은 대역폭의 존에 할당받을 확률이 높아져서 전체 전송시간이 길어지기 때문이다. 따라서 새로운 사용자의 서비스 요청이 VOD 서버에 도착할 경우에 임의적인 배치에서는  $T_{round}^i$ 가 선형적으로 증가하여 동시지원 가능한 사용자 수가 증가할 수 없게 된다.  $T_{round}^i$ 를 감소시키기 위해서는 인기 있는 비디오 블록을 바깥쪽 실린더에 배치하는 방법이 필요하다. (그림 2.1), (그림 2.2)는 이러한 상황을 고려한 선형배치 예와 알고리즘을 나타낸 그림이다.

(그림 2.1)은 선형배치기법의 예를 보여주는 것인데, 한 디스크에 저장될 비디오들의 블록들을 인기순으로 정렬(Block<sub>0</sub>에서 Block<sub>n</sub>)하여 외곽 존인 Cyl<sub>0</sub>부터 내곽 존인 Cyl<sub>m</sub>으로 순차적으로 배치한다. 따라서 액세스 확률이 높은 블록들이 대역폭을 충분히 할당받으므로 편기가 일어날 경우 디스크의 처리율을 높일 수 있다. 그리고 (그림 2.2)의 선형배치 알고리즘에서 접근확률에 따라 빈 공간을 확보한다(step 6, 15). 이것은 특정 디스크에 새로운 사용자의 추가로 인하여 서비스 라운드의 한계시간(deadline)을 만족할 수

Outer Cylinder				Inner Cylinder		
Cyl 0	...	Cyl i-1	Cyl i	Cyl i+1	...	Cyl m
Block 0	...	Block j-1	Block j	Block j+1	...	Block n
Higher Popularity				Lower Popularity		

(그림 2.1) 선형배치의 예

6) VOD에서 가장 인기 있는 비디오에 대한 접근 확률은 30~40% 이상 된다.

```

1  Procedure Linear_Placement();
2  begin
3      sort blocks in decreasing order of access probability;
4      for(i=0; i<number_of_video; i++)
5          begin
6              calculate reserved disk space per video based on access profile;
7          end;
8      for(i=0; i<number_of_video; i++)
9          begin
10             for(j=0; j<number_of_ith_video's_blocks; j++)
11                 begin
12                     place a video block under current disk head;
13                     move disk head;
14                 end;
15             reserve empty space that calculated at step 6;
16         end;
17     end;

```

(그림 2.2) 선형배치 알고리즘

없을 때 다른 디스크에 확보된 여유 공간으로 비디오 블록을 복사하고 새로운 사용자 요구는 복사본(replica)을 가지고 있는 디스크에서 계속 서비스할 수 있게 한다.

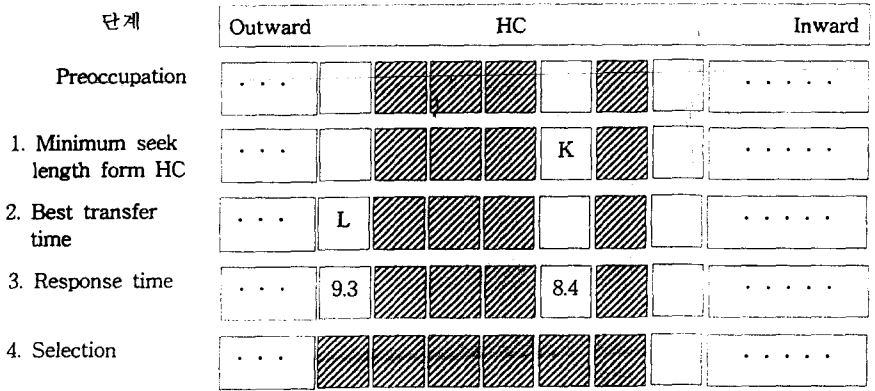
### 3.3 편기된 휴리스틱(Skewed heuristic : SH) 배치

저장서버가 허용제어를 할 때 디스크 대역폭에 여유가 있다면 추가로 사용자의 요구를 수용하게 되어 동시 지원 가능한 사용자 수가 증가된다. 저장서버가 동시에 여러 비디오 스트림을 지원하게 되는 경우 탐색시간이 많이 소비되어 전체 서비스 시간 중에서 차지하는 비중을 무시할 수 없게 된다. 한 라운드에서 각 요구의 크기가 10 KB 이하이면 탐색시간이 전체 서비스의 80% 이상을 차지하게 되므로(그림 1), 탐색을 어느 정도 휴리스틱하게 고려할 필요가 있다. 본 논문에서 제시하는 편기된 휴리스틱 배치 알고리즘은 각 존의 전송률 뿐만 아니라 탐색 시간을 고려하기 위해 3.1 절에서 언급한 휴리스틱 센타(HC)개념을 사용한다. SH방법의 핵심은 HC를 기준으로 탐색시간과 전송률을 고려하여 접근 프로파일의 확률순에 따라 비디오 블록을 배치한다. 디스크 헤드가 위치할 확률이 높은 실린더에 접근 확률이 높은 블록을 배치

하는 것은 디스크 스케줄링의 입장에서 본다면 탐색 거리를 감소시킬 수 있는 요인이 된다. 이미 비디오 블록이 저장된 실린더의 접근확률은 실린더에 저장된 블록의 접근확률과 같으며 Zipf 분포에 따라 접근된다. 따라서 순차적으로 비디오 블록에 접근한다면 이미 저장되었던 실린더에서 응답시간이 최소인 실린더에 다음 블록을 저장하면 탐색시간을 확률적으로 감소시킬 수 있다. 이것은 성질 1, 2를 이용한 것으로서 배치기법의 예와 알고리즘은 (그림 3.1), (그림 3.2)와 같다.

(그림 3.1)은 SH기법의 예를 보이고 있는데 빗금친 부분은 블록이 배치된 실린더를 의미한다. Step 1에서는 HC로부터 최소탐색거리를 가지는 실린더를 선택하는데 (그림 3.1)에서는 K실린더가 선택되어진다. Step 2에서는 가장 높은 전송률을 가지는 실린더가 선택되는데 외곽 존에 가장 근접한 실린더 L이 선택된다. Step 3에서는 이미 저장된 실린더에서 후보 실린더인 K, L실린더로의 응답시간을 계산한다. Step 4에서는 후보실린더들 중에서 응답시간에 가장 낮은 실린더를 선택하는데 (그림 3.1)에서 K실린더가 8.4로 적으므로 K실린더에 다음 블록을 저장한다.

(그림 3.2)는 SH기법의 구체적인 알고리즘이다. 먼



(그림 3.1) Skewed heuristic 배치의 예

```

1  Procedure Skewed_Huristic_Placement();
2  begin
3      move disk head to heuristic center;
4      do
5          find empty cylinders(Emp_Cyl);
6          find partially allocated cylinders(Partial_Allocated_Cyl);
7          find cylinder(BST_Cyl) that have minimum seek time from heuristic
           center of Emp_Cyl;
8          find cylinder(BTT_Cyl) that have minmum transfer time of Emp_Cyl;
9          find cylinders(Sub_PA_Cyl) that can allocate current video blocks of
           Partial_Allocated_Cyl;
10         for(i = each cylinder of Best_Seek_Time_Cyl, Best_Transfer_Cyl,
              Sub_PA_Cyl)
11             begin
12                 calculate the access time that consider seek time and transfer
                    time
13             end;
14         find cylinder(m) that have minimum access time of step 12's results;
15         update access probabily of m cylinder;
16     while( have not empty disk blocks);
17 end;

```

(그림 3.2) Skewed heuristic 배치 알고리즘

저 블록을 저장할 후보 실린더로 HC에서 빈 실린더 까지 탐색시간이 최소인 실린더 집합(단계 7), 빈 실린더 중 전송률이 최대인 실린더 집합(단계 8), 블록을 저장할 공간이 남아 있는 실린더 집합(단계 9)을 찾는다. 그리고 이미 저장된 실린더에서 후보 실린더 집합으로의 응답 시간을 다음 식에 의해 계산한다(단계 12).

$$R[i] = \sum_j P_j * ST|i-j| + S_b * T_i,$$

- $j \in \{ \text{완전히 할당된 실린더} \},$
- $P_j: j$ 번째 실린더가 접근될 확률,
- $ST|i-j|: \text{디스크헤드가 거리 } |i-j| \text{를 이동하는 시간,}$
- $S_b: \text{현재 할당할 블록크기(바이트 단위),}$

$T_i$ :  $i$ 번째 실린더의  $i$  바이트 전송시간

이미 저장된 실린더에서 후보 실린더들로의 탐색 시간을 계산할 때, VOD의 순차적인 읽기 연산을 고려하여 기존에 저장된 실린더가 선택될 확률을 곱하여 탐색시간을 확률적으로 감소시킨다. 이렇게 계산된 접근시간 중에서 최소값을 갖는 실린더에 다음 블록을 저장한다(단계 13). 연이은 다음 블록 저장을 위해서 그 실린더의 접근 확률을 갱신한다(단계 14).

#### 4. Zipf 분포에 따른 제안된 방법의 실험

##### 4.1 Zipf 분포

Zipf 분포는  $N$  개의 객체 중에 몇 개의 객체가 선택될 확률이 다른 객체에 비해 현저히 클 때 특정 객체가 선택될 확률을 구하기 위해 사용되는 것으로(그림 4)와 같은 식으로 표시된다. 단순 Zipf 분포는  $i$  번째로 빈번하게 선택될 객체의 확률은  $F_i$ 로 표시된다. 단순 Zipf 분포는 편기성 정도를 표현할 수 없어 편기성 정도를 나타내는  $\theta$  인수를 추가한다.  $\theta$  값이 0 이면 단순 Zipf 분포이고,  $\theta$  값이 1이면 일항(uniform) 분포를 나타낸다. 즉  $\theta$  값이 적을 수록 편기성이 증가하고 1에 접근할 수록 일항분포에 접근한다.

VOD에서 사용자가 비디오를 시청할 때 특정 비디오에 대한 사용자의 선호도(인기도)를 나타내기 위하여 Zipf 분포를 많이 사용한다. 여기에서는 접근 프로파일을 예측하는데 Zipf 분포를 사용하였다.

Pure Zipf Distribution

$$\text{Frequency } F_i = \frac{1}{i} \quad (1 \leq i \leq n)$$

Zipf Distribution with the skew ( $\theta$ )

$$\text{Frequency } F_i = \frac{1}{i^{1-\theta}} \quad (1 \leq i \leq n)$$

$$\text{Probability } P_i = \frac{F_i}{\sum_j F_j}$$

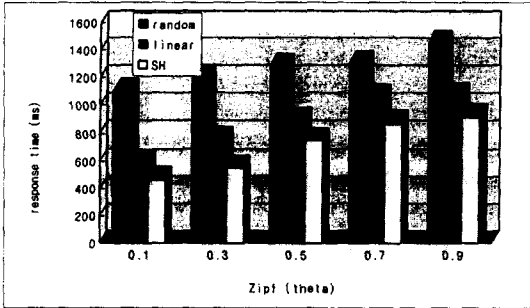
(그림 4) Zipf 분포

##### 4.2 MZR 디스크에서의 실험

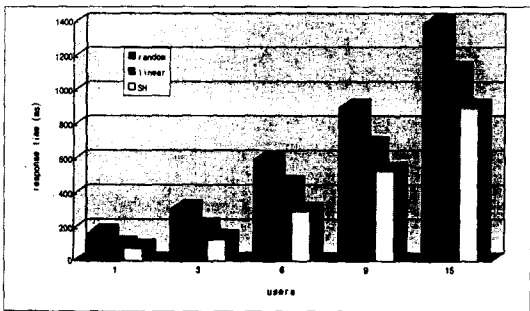
MZR 디스크에서 선형배치, 편기된 휴리스틱 배치 방법의 성능분석을 위해서 워크스테이션<sup>7)</sup>에 Quantum Fireball 540S 디스크 2대를 장착하여 실험하였다. 이 실험에서는 초당 25 프레임율 사용자의 셋탑박스가 요구한다고 가정하고 400초 분량의 비디오 데이터를 임의(random)배치, 선형배치, SH 배치방법에 따라 디스크에 저장하였다. 임의배치는 제안된 방법의 비교를 위해 임의적으로 선택된 실린더에 비디오 블록들을 배치하는 방법이다. 그리고 원격 사용자로부터의 요구를 나타내는 프로세스를 사용자 수만큼 생성하고 각 프로세스는 Zipf 분포에 의해 비디오의 블록을 저장서버에게 요구하도록 했다. 본 실험에서는 프로세스 제어를 위한 프로세스 스케줄링은 고려되지 않았다. 각 프로세스는 수행을 시작하라는 신호를 받으면 수행이 되도록 하였으며 주기적으로 저장서버에게 비디오 블록을 요구한다. (그림 5)는 3 가지 배치방법에 따라 비디오 블록을 배치하고 사용자가 Zipf 분포에 따라 비디오를 요구할 때, 스트라이핑 단위가 64KB인 경우의 평균 응답시간을 실험한 결과이다. 3 가지 배치 방법에서  $\theta$  값이 커질수록 응답시간이 늦어지는 현상을 나타낸다. 이것은 사용자가 여러 비디오를 요구하여 각 실린더로의 요구가 분산되기 때문이다. 임의 배치방법은 선형배치, SH 배치와 비교할 때 응답시간에 많은 차이를 나타낸다. 이러한 현상은 인기 있는 비디오가 저대역폭의 실린더에 배치되고 인기 없는 비디오가 고대역폭의 실린더에서 서비스되는 확률이 높아지기 때문이다.  $\theta$  값이 0.3 일 때 선형배치가 700ms, SH가 620ms의 평균 응답시간을 나타내었다. (그림 6)은 사용자의 수에 따른 3 가지 배치방법의 응답 시간을 분석한 것이다. 임의 배치방법은 사용자의 수가 증가함에 따라 급격한 응답시간의 증가를 보이고 있다. (그림 5)에서와 같이 SH 배치방법이 선형배치에 비해 좋은 응답시간을 보이고 있다. 특이한 것은 사용자의 수가 3에서 9명으로 증가할 때 SH 배치방법은 응답시간의 증가가 완만한데 비해서 선형배치 방법은 SH방법보다는 응답시간의 증가폭이 크다. 이것은 사용자 수가 증가할수록 탐색시간이 응답시간에서 차지하는 비중이 커져서 탐색시간을

7) 모델: AXIL 320, CPU: superscalr(60Mhz), MM:32MB)

고려한 SH 배치방법이 디스크헤드의 이동거리를 단축시켰기 때문이다.



(그림 5) Zipf 분포에 의한 3가지 배치기법의 응답 시간



(그림 6) 사용자 수에 의한 응답 시간

### 5. 디스크 스케줄링 방법의 해석적인 분석

모의실험을 하기 전에 전송률을 결정하는 방법과 FIFO, C-SCAN 디스크 스케줄링 방법에 의한 전송률이 VOD가 지원 가능한 사용자 수와 버퍼크기에 미치는 영향을 해석하고 수치적 결과를 도출하였다. VOD 서버는 이중버퍼(double buffer)<sup>8)</sup>를 사용한다. <표 4>는 해석적 분석에서 사용되는 기호에 대한 정의이다.

#### 5.1 MZR 디스크의 전송률

기존 디스크들은 전송률이 고정된 상수값을 가지

<표 4> 해석적 분석의 기호 정의

기호	정의
$T_{seek}^{worst}$	최대 탐색 시간
$N_{service\ list}$	하나의 서비스 라운드에서 있는 요구 수
$T_{settle}$	디스크 헤드를 안정화하는데 필요한 최대시간
$T_{rotation}^{worst}$	최대 회전 지연 시간
$S_{request}$	각 요구에서 필요한 평균 비디오 데이터 크기
$B_{size}$	사용자 당 필요한 버퍼 크기
$N_{disk}$	VOD 서버에 있는 디스크 수
$T_{disk}^{transfer\ rate}$	디스크의 전송률
$T_{play}$	클라이언트의 display 비율

므로 전송률 측면에서 지원 가능한 사용자 수를 미리 구할 수 있었다. 그러나 MZR 디스크에서는 전송률이 변수로 취급되어지므로 지원 가능한 사용자 수도 변할 수 있다. 따라서 지원 가능한 사용자 수와 버퍼 크기를 결정하는 과정에 있어서 모든 상황을 사전에 예측할 수 없으므로 전송률을 결정하는 세 가지 방법을 고려해야 한다. 즉 디스크는  $nc$  개의 실린더와  $n$  개의 존을 가지고 있으며,  $i$ 번째 존의 전송률은  $Z_i$ ,  $j$ 번째 실린더의 전송률은  $C_j$ 라고 가정한다.

$$\text{방법 1: } T_{disk}^{transfer\ rate} = \frac{\sum_{i=1}^n Z_i}{n}$$

모든 존의 전송률의 합을 존의 수로 나눈 산술 평균값을 디스크의 전송률로 결정하는 방법이다. 이 방법은 존이 선택되어질 확률이 동일할 경우에 적용 가능하나 VOD에서는 인기 있는 비디오에 접근이 집중되므로 사용하기에 적당하지 않다.

$$\text{방법 2: } T_{disk}^{transfer\ rate} = \sum_{i=1}^n Z_i * P_i^{\theta}$$

$P_i^{\theta}$ 는  $n$ 과  $\theta$ 값을 인수로 갖는 Zipf 분포에서  $i$ 번째 존이 선택되어질 확률이다. 방법 2는 전

8) 하나는 선 반입을 위해 사용되고 다른 버퍼는 클라이언트에게 데이터를 전송하기 위해 사용된다.

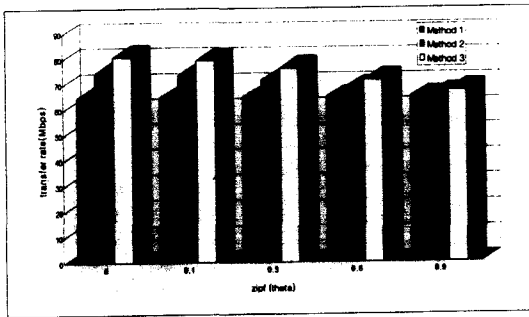


송률을 결정할 때 존에 접근될 확률이 고려되기 때문에 방법 1에 비해 사용자의 접근형태에 근접한 경우다. 특히 MZR 디스크에서 바깥쪽 존은 전송률이 높고 섹타수가 많으므로 존이 선택될 확률로 Zipf 분포를 사용하는 것은 타당하다 하겠다.

방법 3:  $T_{disk}^{transfer\ rate} = \sum_{i=1}^{nc} C_i * CP_i^6$

$CP_i^6$ 는 nc,  $\theta$  값을 인수로 가지는 Zipf 분포에서 i번째 실린더가 선택되어질 확률이다. 실린더가 접근될 확률을 고려하므로 방법 2보다 정교한 전송률을 계산할 수 있다.

(그림 7)은 위의 세 가지 방법에 대한 Quantum Fireball 540S의 전송률을 계산한 것이다. 방법 3이 방법 2보다 약 4.8Mbps 정도의 높은 전송률을 나타냈다. 허용제어는 이 여분의 전송률로 새로운 요구를 수용할 수 있다. 본 논문에서는 가장 정교하게 고려된 방법 3의 전송률을 시뮬레이션 모델로 사용하며 이 값은 HC의 전송률과 거의 같게 된다.



(그림 7) MZR 디스크에서의 전송률

5.2 FIFO

FIFO 디스크 스케줄링 방법은 한 라운드에 포함된 요구들은 큐에 들어온 순서대로 처리되기 때문에 하나의 서비스 라운드를 처리하는데 필요한 전체 디스크 탐색시간은  $\frac{T_{seek}^{worst}}{3} * N_{service\ list}^{9)}$ 이 된다. 다음 식은

하나의 서비스 라운드를 처리하는데 필요한 시간을 나타낸다.

$$T_{round} = N_{service\ list} * (\frac{T_{seek}^{worst}}{3} + T_{settle} + T_{rotation}^{worst} + \frac{S_{request}}{N_{disk} * T_{disk}^{transfer\ rate}}) \leq \frac{S_{request}}{T_{play}} \quad (1)$$

다음은 (1) 식에서 유도된 지원 가능한 사용자수이다.

$$N_{service\ list} = \left\lfloor \frac{S_{request}}{T_{play} * (\frac{T_{seek}^{worst}}{3} + T_{settle} + T_{rotation}^{worst} + \frac{S_{request}}{N_{disk} * T_{disk}^{transfer\ rate}})} \right\rfloor \quad (2)$$

(1)의 식에서 각 사용자 당 필요한 버퍼 크기를 다음 (3)식과 같이 구할 수 있다.

$$B_{size} = \left\lceil \frac{T_{play} * N_{service\ list} * N_{disk} * T_{disk}^{transfer\ rate} * (T_{seek}^{worst} + 3(T_{settle} + T_{rotation}^{worst}))}{3(N_{disk} * T_{disk}^{transfer\ rate} - T_{play} * N_{service\ list})} \right\rceil \quad (3)$$

5.3 C-SCAN

C-SCAN방식은 한 라운드 동안 처리해야 할 블록들을 실린더 순으로 정렬시켜서 헤드가 바깥쪽에서 안쪽으로 올 때, 혹은 안쪽에서 바깥쪽으로 이동할 때 서비스하고 다시 원위치로 되돌아간다. 따라서 한 라운드에서 최대 탐색거리는  $2 * T_{seek}^{worst}$ 가 된다. C-SCAN에서도 FIFO와 마찬가지로 한 라운드 검색에 걸리는 시간을 클라이언트들의 화면에 나타나는 시간보다 빨리 하여 버퍼에 선 받입시켜 전송해야 한다. C-SCAN 방식에서의 서비스 시간은 다음과 같다.

$$T_{round} = 2 * T_{seek}^{worst} + N_{service\ list} * (T_{settle} + T_{rotation}^{worst} + \frac{S_{request}}{N_{disk} * T_{disk}^{transfer\ rate}}) \leq \frac{S_{request}}{T_{play}} \quad (4)$$

9) 여러 요구를 순차적으로 처리할 때 각 요구당 디스크 헤드의 평균 이동 거리는 전체 트랙 길이의 1/3이 된다.

(4)로부터 지원 가능한 사용자 수는 다음과 같다.

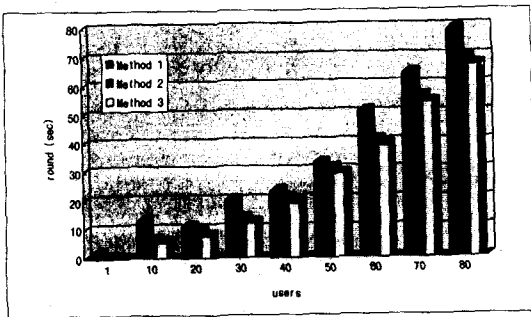
$$N_{service\ list} = \left\lfloor \frac{S_{request} - 2 \times T_{seek}^{worst} \times T_{play}}{T_{play} \times (T_{settle} + T_{rotation}^{worst} + \frac{S_{request}}{N_{disk} \times T_{transfer\ rate}^{disk}})} \right\rfloor \quad (5)$$

(4)로부터 각 사용자당 필요한 버퍼의 크기는 다음과 같다.

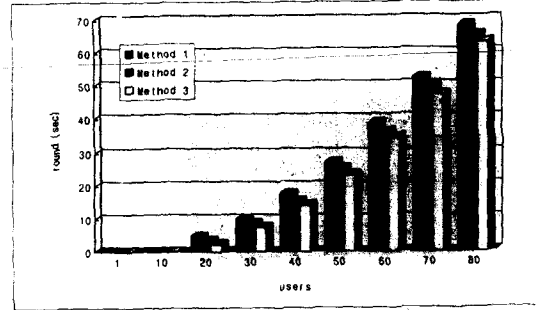
$$B_{size} = \left\lceil \frac{T_{disk}^{transfer\ rate} \times (2 \times T_{disk}^{rotation} + N_{service\ list} \times T_{settle} + N_{service\ list} \times T_{rotation}^{worst})}{\frac{N_{service\ list}}{N_{disk}} - \frac{T_{transfer\ rate}^{disk}}{T_{play}}} \right\rceil \quad (6)$$

### 5.4 수치적인 결과

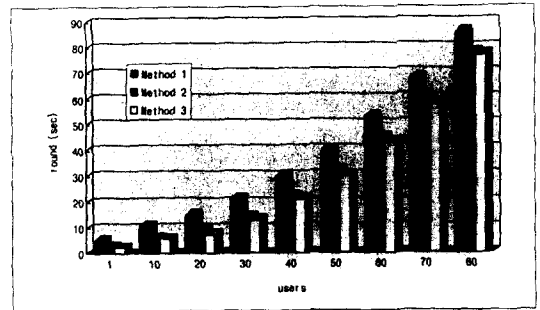
(그림 7)에서 구한 전송률을 FIFO와 C-SCAN 디스크 스케줄링 방법에 적용할 때 하나의 서비스 라운드를 처리하는데 소요되는 시간을 측정하는 것은 중요하다. 하나의 서비스 라운드를 처리하는 시간이 감소되면 VOD서버가 요구를 서비스하는 평균시간 간격이 감소하게 된다. MZR 디스크는 가변 전송률을 가지므로 특정 시점에서 VOD 서버의 부하를 측정하는 기준이 필요하다. (그림 8)과 (그림 9), (그림 10)과 (그림 11)는 MZR 디스크의 전송률을 결정하는 3가지 방식을 FIFO, C-SCAN에 적용할 때 버퍼크기와 사용자 수에 따른 하나의 서비스 라운드를 처리하는데 소요되는 시간을 보여주고 있다. 실험 결과의 공통점은 전송률이 증가할수록 서비스 라운드시간의 합이 감



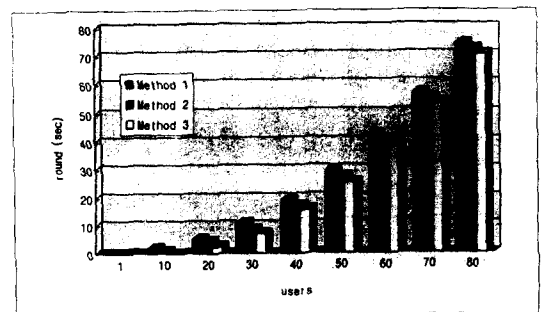
(그림 8) FIFO 스케줄링 방법에서의 서비스 라운드 시간 (버퍼 크기 : 64KB)



(그림 9) CSCAN 스케줄링 방법에서의 서비스 라운드 시간(버퍼 크기 : 64KB)



(그림 10) FIFO 스케줄링 방법에서의 서비스 라운드 시간 (버퍼 크기 : 256KB)



(그림 11) CSCAN 스케줄링 방법에서의 서비스 라운드 시간(버퍼 크기 : 256KB)

소됨을 보여준다. 또한 버퍼의 크기가 증가할 수록 전체 라운드 시간도 감소함을 알 수 있다. 또한 FIFO 방법에 비하여 C-SCAN 방법이 전체 서비스 라운드 시간이 짧게 나타나는데 이것은 FIFO의 평균 탐색거리가 C-SCAN 보다는 길기 때문이다.

### 6. 모의실험 및 성능 평가

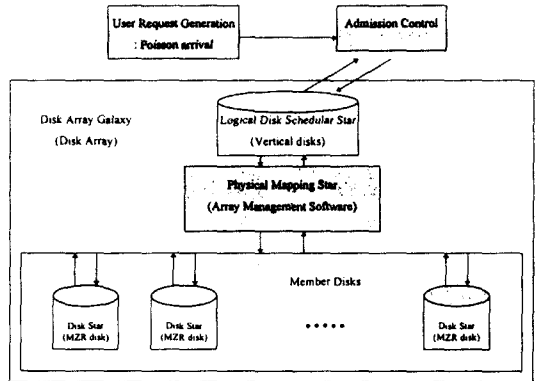
VOD 저장서버의 디스크 배열을 모델링하여 시뮬레이션을 수행하였다. 본 논문에서 사용한 각 디스크의 사양은 <표 5>와 같다.

<표 5> 모의실험에서 사용된 디스크 사양

디스크 인수	Value
포맷된 용량	549.3 MB
RPM	4,500
헤드 수	3
존 수	15
가장 안쪽 존의 섹터 수	88
가장 바깥쪽의 섹터 수	179
탐색시간	4 ms
가장 바깥쪽 존의 전송률	84.12 Mbits/sec
가장 안쪽 존의 전송률	43.03 Mbits/sec

#### 6.1 디스크 배열에 대한 모의실험 모델

성능을 측정하기 위해서 모의실험 도구로 Ptolemy [12]를 이용하였는데 Ptolemy는 이산사건을 포함한 여러 가지 형태를 모의실험 할 수 있다. 본 연구에서는 이산적인 사건을 다루는 Ptolemy DE(discrete event) Domain의 universe galaxy, star들을 사용하여 모의 실험하였다. 본 논문에서는 540MB의 저장용량을 갖는 24개의 하드 디스크(총 저장용량:12.65GB)로 구성된 하나의 디스크 배열을 대상으로 실험하였다. VOD 서버에서의 병목현상은 저장시스템의 하드디스크 부분이 가장 심하므로 모의실험에서는 다른 시스템 자원을 제외한 디스크 배열만을 모델링하였다. 본 연구에서는 RAID 5 level의 디스크 배열에서 모의 실험하기 위하여 (그림 12)와 같은 모의실험 모델을 구성하였다. 사용자의 요구는 Poisson 분포로 도착되며 admission control star는 새로 도착되는 사용자 요구의 수용여부를 결정한다. 디스크 배열 모델에서 허용제어 정책을 구현하기가 어렵기 때문에 여기에서는 미리 정해진 최대치를 넘지 않은 범위에서 서비스하는 것으로 단순화 시켰다. 먼저 사용자로부터 허용된 요구들이 디스크 배열에 비디오 블록을 요청한다



(그림 12) 모의실험 모델

고 하자.

Logical disk scheduler star는 논리적으로 구성된 디스크 클러스터에 대한 정보를 유지하고 사용자의 요구들을 자신의 큐에 넣어서 주기적으로 검색할 블록을 Physical mapping star로 보낸다. Physical mapping star는 논리적인 블록 정보를 물리적인 위치로 매핑시켜서 물리적인 블록번호에 해당하는 MZR disk star로 보낸다. MZR disk star는 자신의 스케줄링 방법에 따라 요구된 블록들을 서비스하게 된다. 이 Disk star는 <표 5>에 나타난 헤드, 각 실린더 용량, 전송률, 존의 수 등과 같은 물리적인 정보를 사용하여 요구를 처리하는데 필요한 시간동안 지연된 후에 Physical mapping star로 완료 메시지를 보낸다. Physical mapping star는 disk star로 부터 완료 메시지가 모두 도착하면 Logical disk scheduler star로 완료 메시지를 보낸다. 이 과정 전체가 하나의 서비스 라운드에 대한 검색과정을 나타낸다.

#### 6.2 모의실험 인수

모의실험에서는 블록배치 방법, 블록 크기, 디스크 스케줄링 방법을 인수로 하여 수행하였다. 블록배치는 임의배치, 선형배치, SH방법에 의해 각각 배치하였다. 블록 배치시 스트라이핑 단위를 변동하면서 성능을 측정하였다. 스트라이핑 단위 크기가 작으면 요구되는 버퍼의 크기는 감소하지만 디스크 탐색시간의 부담으로 인해 수용할 수 있는 사용자의 수가 감소된다. 스트라이핑 단위가 크면 라운드당 읽어들이

는 블록이 많아지므로 요구되는 버퍼크기의 증가로 인해 시스템의 비용은 증가되나 탐색시간의 부담<sup>10)</sup>은 줄어든다. VOD 응용과 같이 연속적인 서비스를 요구하는 환경에서 MZR 디스크에 적합한 디스크 스케줄링 방법을 선정하기 위해서 3 가지 배치방법에 따라 FIFO, C-SCAN 디스크 스케줄링 방법의 성능을 측정하였다.

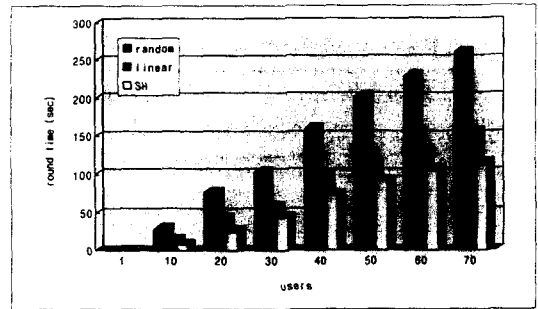
### 6.3 모의실험 결과

MZR 디스크 상에서 선형배치, SH 배치방법의 성능을 분석하기 위해 약 10분 정도의 모의실험 환경을 준비했다. 24개의 Quantum Fireball 540S 디스크에 선형배치, SH 배치방법에 따라 10개의 비디오를 저장 시켰다. 디스크의 응답시간을 계산하는데는 5장에서 통계적으로 유도된 전송률을 사용하였으며 실험은 디스크 스케줄링 방법, 스트라이핑 단위, 사용자 수에 따른 응답시간을 측정하였다.

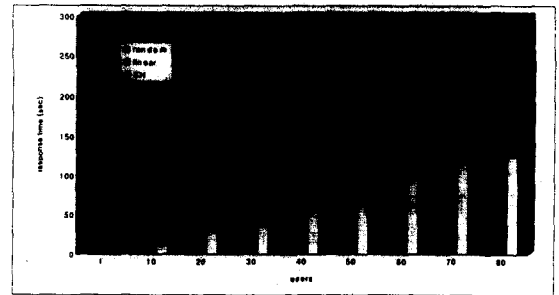
#### 6.3.1 사용자 수가 응답 시간에 미치는 영향

(그림 13)과 (그림 14)는 사용자의 수에 따른 VOD 서버의 응답시간을 분석한 그래프이다. 사용자 접근 분포는  $\theta$  값이 0.271인 Zipf 분포를 가정하였다. 사용자들은 VOD 서버에 포이센(Poisson) 분포로 단위 시간당 평균 100개가 도착한다고 가정하며, 비디오를 시청할 때 fast forward, rewind 등의 오퍼레이션 없이 연속해서 시청한다고 가정하였다. 이 실험에서는 임의 배치, 선형배치, SH 배치방법에 의해 비디오 블록을 디스크에 배치하고 사용자 수를 1명에서 80명까지 증가시켰을 때 전체 응답시간의 합을 나타내었다. FIFO 디스크 스케줄링 방식이 C-SCAN 디스크 스케줄링 방법을 채용한 경우보다 응답시간이 전체적으로 커짐을 볼 수 있다. 또한 임의로 블록을 배치한 경우는 선형배치와 SH 배치 방법에 비해 응답시간이 많이 소요됨을 알 수 있다. SH는 선형배치보다 약 8.4% 성능 향상을 보이고 있다. 그리고  $\theta$  값이 0.5까지 증가한다고 가정해 보자.  $\theta$  값이 증가하면 특정 오브젝트에 대한 접근 확률이 낮아지므로 디스크 서버 시스템의 전체 서비스 시간중 탐색 시간의 비율이 증가한다. 선형배치의 경우는 한 비디오에 소속된 비디오

들이 특정 존에 집중적으로 배치되어 있고 SH의 경우는 확률에 따라 약간의 차이가 존재한다. 즉 탐색 시간의 비율이 늘어날 수록 SH 알고리즘은 더욱 향상된 성능을 보일 것이다.



(그림 13) 사용자수가 응답 시간에 미치는 영향(FIFO, Zipf  $\theta = 0.271$ )



(그림 14) 사용자수가 응답 시간에 미치는 영향 (CSCAN, Zipf  $\theta = 0.271$ )

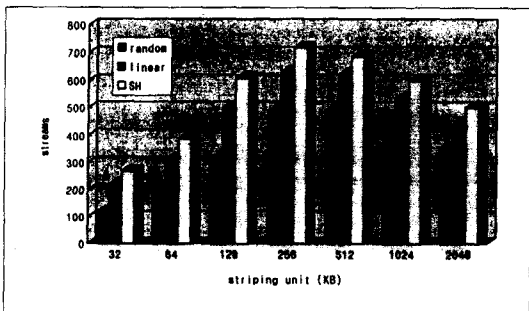
#### 6.3.2 블록의 크기가 응답 시간에 미치는 영향

(그림 15)와 (그림 16)은 스트라이핑의 단위 크기가 32KB에서 2,048KB 까지 변화할 때 지원 가능한 스트림의 수를 나타낸 것이다. 전체적으로 보면 C-SCAN 디스크 스케줄링 방법이 FIFO 방식에 비해 10% 이상의 스트림을 더 지원할 수 있는 것으로 나타났다. 스트라이핑 단위가 256KB일 때 두 방법에서 최대의 스트림수를 보이고 있다. C-SCAN의 경우 SH 배치방법이 약 800 스트림을, 선형 배치방법이 720 정도의

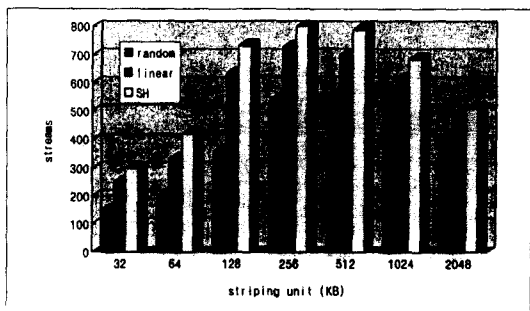
10)가변 전송율을 갖는 MZR 디스크에서는 탐색시간이 중요하다.

스트림을 지원할 수 있는 것으로 나타났다. FIFO의 경우는 SH 배치방법이 약 700 스트림을, 선형 배치방법이 640정도의 스트림을 지원할 수 있는 것으로 나타났다. FIFO와 C-SCAN에서 모두 256KB까지는 스트림의 수가 지속적으로 증가하고 256KB를 넘어 서면 점차로 스트림의 수가 감소함을 볼 수 있다.

스트라이핑 단위가 적을 때(이 모의실험의 경우 256KB이하)는 디스크의 전송률이 최대한 활용되지 못하다가 256KB에 접근할 수록 탐색시간의 부담은 최소화되어 디스크 존의 대역폭이 최대로 활용되기 때문이다. 그러나 256KB 이상 되면 전송률이 전체 서비스 시간에서 차지하는 비중이 탐색시간에 비해 상당히 커지게 되므로 응답시간의 지연을 초래하고 결과적으로 손실되는 스트림이 증가하게 된다. 따라서 총 허용되는 스트림은 감소한다.



(그림 15) 스트라이핑 단위에 따른 서비스 스트림 수(FIFO, Zipf  $\theta = 0.271$ )



(그림 16) 스트라이핑 단위에 따른 서비스 스트림 수(CSCAN, Zipf  $\theta = 0.271$ )

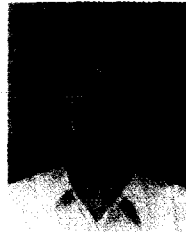
## 7. 결 론

본 논문에서는 최근 상용화되고 있는 MZR 디스크에서 비디오 블록을 효율적으로 배치하기 위한 선형배치와 편기된 휴리스틱(SH) 배치방법을 제안했다. 이 실험 결과 임의배치 보다는 선형배치와 SH 배치 방법의 성능이 상대적으로 우수함을 보였다. 이 성능향상을 바탕으로 MZR 디스크의 가장 큰 변수인 전송률을 결정하기 위해서 각 존의 산술평균, 존의 Zipf 분포에 의한 기대값, 실린더의 Zipf 분포에 의한 기대값 등 세 가지를 고려하여 실험하였다. 또 위의 전송률의 기대값이 사용자 수와 버퍼의 크기에 미치는 영향에 대하여 해석적인 측정을 하였다. 가장 정밀하게 측정된 실린더의 선택 확률을 고려한 방법이 가장 좋은 결과를 나타내었다. 또 위의 결과와 전송률의 기대값 등을 고려하여 모의실험을 수행하였다. 임의로 블록을 배치한 경우는 존의 가변적인 전송률이 고려되지 않아 선형배치와 SH 배치방법보다 상당한 성능의 저하를 나타냈다. 그리고 접근 프로파일의 확률에 의해 HC(heuristic center)로부터 전송률과 탐색시간을 고려한 SH 배치방법이 단순히 전송률만 고려한 선형 배치 방법보다 약 8.4% 이상의 성능향상을 보여 주고 있다. 따라서 MZR 디스크에서는 존의 전송률을 고려한 배치방법이 보다 많은 사용자를 수용할 수 있는 것이다. 알고리즘의 복잡도를 고려한다면 SH 배치 방법은 선형배치 방법보다 복잡하다는 단점이 있다.

## 참 고 문 헌

- [1] Francois fluckiger, Understanding Networked Multimedia: Applicationa and Technology, Prentice Hall, page 548-549.
- [2] P. Chen, D. Patterson. Maximizing Performance in a Striped Disk Array. Proceedings of ACM SIGARCH Conference on Computer Architecture, Seattle, WA, pages 322-331, May 1990.
- [3] R. H. Katz, G. Gibbson, D. Patterson. Disk System Architectures for High Performance Computing. Proceedings of the IEEE, 77(12):1842-1858, December 1989.

- [4] D. Patterson, G. Gibson, R. Katz. A Case for Redundant Array of Inexpensive Disks (RAID). ACM SIGMOD88, pages 109-116, June 1988.
- [5] Quantum Fireball 540/1080S Product Manual. 1996.
- [6] Dan, A., Sitaram, D. and Shahabuddin, P. Scheduling Policies for an On-Demand Video Server with Batching. Proceedings of ACM Multimedia 94 Conference, pages 15-21, October, 1994.
- [7] Banu Ozden, Rajeev Rastogi, Avi Silberschatz, Disk Striping in Video Server Environments, Proceedings of the international conference on multimedia computing and systems, Hiroshima, Japan, June 1996.
- [8] Ming-Syan Chen, Dilip D. Kandular, Philp S. Yu, Support for Fully Interactive Playout in a Disk-Array-Based Video Server, Proceedings of 1994 ACM Multimedia, 1994.
- [9] Yitzhak Birk, Deterministic Load-Balancing for Disk-Based Video-On-Demand Storage Servers, Proceedings of the international conference on multimedia computing and systems, Washington, DC. May 15-18, 1995.
- [10] Dan, A. M., Kienzle and D. Sitaram, Dynamic Segment Replication Policy for Load-Balancing in Video-on-Demand Servers, ACM Multimedia Systems, Vol 3, 1995.
- [11] Asit dan, Dinkar Sitaram, An Online Video Placement Policy based on Bandwidth to Space Ratio (BSR), ACM SIGMOD, 1995.
- [12] Almagest 1995, Users manual.



**임 형 룡**

1986년 2월 건국대학교 공과대학 전자공학과 졸업(학사)  
 1994년 2월 한양대학교 산업대학원 전자통신공학 전공(공학석사)  
 1997년 11월 시스템공학연구소 연구원

관심분야: 멀티미디어, VOD시스템, 컴퓨터 네트워크



**김 정 원**

1995년 부산대학교 전자계산학과 대학원, 이학사  
 1997년 부산대학교 전자계산학과 대학원, 이학석사  
 1997년~현재 부산대학교 일반대학원 전자계산학과, 박사과정

관심분야: 멀티미디어 화일 시스템, 병렬 처리



**김 영 주**

1988년 부산대학교 계산통계학과 졸업(이학사)  
 1990년 부산대학교 대학원 계산통계학과 졸업(이학석사)  
 1990년~1995년 큐닉스컴퓨터 응용시스템연구소 근무

1995년~현재 부산대학교 대학원 전자계산학과 박사과정 재학

관심분야: 멀티미디어, 병렬파일시스템, 분산처리 등



**정 기 동**

1973년 서울대학교 졸업(학사)  
 1975년 서울대학교 대학원 졸업(석사)  
 1986년 서울대학교 대학원 계산통계학과 졸업(박사)  
 1990년~1991년 MIT, South Carolina 대학 교환 교수

1978년~현재 부산대학교 전자계산학과 교수  
 1993년~현재 부산대학교 부설 컴퓨터 및 정보통신연구소 운영위원

1995년~현재 부산대학교 전자계산소장

관심분야: 병렬처리, 멀티미디어