

Zone 구조 디스크에서 효율적 구역 재구성 방법

김 종 희[†] · 최 경 희^{††} · 정 기 현^{†††}

요 약

최근 많이 사용되고 있는 많은 고속 대용량의 디스크는 복수 구역을 갖는 존 구조를 가지고 있다. 이와 같은 존 구조 디스크에서는 여러 개의 구역이 존재하고, 각각의 구역(Zone)마다 한 트랙에 존재하는 섹터 수와 대역폭이 다르다. 따라서 모든 트랙이 동일한 수의 섹터를 가지는 기존의 디스크를 모델로 사용한 연구들을 존 구조를 갖는 디스크에 적용하고 라운드 로빈 방식으로 저장된 비디오 데이터를 SCAN 기법을 이용하여 읽어 낼 때, 전체 디스크의 전송 속도는 가장 느린 안쪽 구역의 전송 속도로 정해지거나, 데이터 배치 시 저장 공간의 과도한 낭비를 발생시킨다. 이런 단점을 보완하기 위해서 본 논문에서는 디스크의 물리적 구역을 분할과 합병을 통하여 논리적 구역으로 재구성시키는 방법과 재구성된 디스크에서의 최적 블록 전송 크기 결정 방법을 제안한다.

An Efficient Zone Reconstruction Method on the Zone-Structured Disk

Jong-Hee Kim[†] · Kyung-Hee Choi^{††} · Gi-Hyun Jung^{†††}

ABSTRACT

Many popularly used recent high-speed and high-capacity disks use the Zone structure in which a disk consists of multiple zones. A zone-structured disk has multiple zones, and the bandwidth and the number of sectors in each zone are different from each other. When the previous studies that modeled disks based on the assumption that the number of sectors in all tracks are same are applied to the zone-structured disk and the video data stored in a round-robin manner are read out with the SCAN technique, the transfer rate of the disk is fixed with that of the innermost zone. And excessive disk space is wasted either. To resolve the problem, this paper proposes a method for reconstructing the physical zones into logical zones by split and merge operations and develops a method for determining optimum transfer block size.

1. 서 론

실시간 주문형 비디오(VOD: Video On Demand) 서버에서는 많은 데이터(압축된 비디오 데이터)를 저장하고, 사용자의 요구에 따라서 저장된 데이터를 대

용량 저장 장치에서 읽어 통신망을 통해 그 사용자에게 전달하게 된다. 이때 서버가 제공하는 서비스는 연속성을 가져야 하며, 서버는 가능한 많은 데이터를 저장하고 그 데이터를 가능한 많은 사용자들에게 동시에 제공할 수 있어야 한다. 대용량 저장 장치로는 하드디스크, 확장된 메모리 및 마그네틱 테이프를 이용한 다양한 장치등이 이용되고 있다. 논문[1]에서는 이들의 장단점, 가격 대 성능비 및 응용 분야에 따른 선택시 고려사항이 잘 보고되어 있다. 이들 중에서 가격

* 본 연구는 정보통신 연구관리공단 대학기초지원연구사업의 지원을 받아 행해졌습니다.
† 정 회 원 : 대신정보통신 연구원
†† 정 회 원 : 아주대학교 정보 및 컴퓨터공학부 교수
††† 정 회 원 : 아주대학교 전자공학부 교수
논문접수: 1998년 4월 10일, 심사완료: 1998년 12월 12일

대 성능비 및 구현상의 이점이 있는 RAID와 같은 하드디스크를 대용량 저장 장치로 많이 사용하고 있다. [2]-[3]

하드 디스크를 대용량 저장 장치로 사용하는 서버에서 디스크의 사용율을 높이고 출력 대역폭을 향상시키기 위한 다양한 데이터 저장 방안들이 제시되었다. 이들의 대부분은 하나의 비디오펜을 여러 개로 조개어 하나의 프로세싱 단위하에 종속되어 있는 여러 개의 디스크에 나누어 저장하는 “스트라이핑(Striping)” 기법이나 변형된 스트라이핑 기법을 이용하고 있다. [4]-[6] 또한 디스크의 내용을 효과적으로 사용하거나 읽어내기 위한 디스크 스케줄링을 위한 많은 연구가 진행되어 왔다. [7]에서는 양방향 VOD 서버에서 완전한 VCR 명령어등을 효과적으로 수행하기 위한 방법이 제시되었고, [4]에서는 디스크 스케줄링 방법으로 SCAN 기법이 다른 방식보다 많은 사용자에게 서비스가 가능함을 보여주고 있다. SCAN 기법은 한 서비스 구간 내에 있는 데이터 리드 명령을 모아서(SCSI 인 경우 가능) 디스크 지름 방향으로 재정렬하고 디스크 헤드가 바깥쪽에서 안쪽으로 한 방향으로만 이동하게 하여 디스크 헤드의 seek time을 최소화하여 전체적인 데이터 출력율을 높이는 방법이다. 이는 디스크 데이터를 읽어내는데 seek time이 매우 큰 비중을 차지함 [8]을 고려할 때 효과적인 방법으로 여겨진다. [2]에서는 탐색시간을 최소화하는 RR-SCAN 방식을 제안하고 있고, [9]에서는 다양한 스케줄링 방법을 비교 평가하고 있다.

위에서 열거한 대다수의 연구들에서는 트랙당 섹터수가 동일한 일반적인 디스크를 모델링하고 있다. 그러나 최근 많은 고속 대용량의 디스크 드라이브는 제한된 면적에 보다 많은 데이터를 저장하기 위해서 “구역 기법(Zoning technique)”을 사용하고 있다. [8] 구역 구조를 사용하는 디스크에서는 각각의 구역(Zone)마다 한 트랙에 존재하는 섹터수가 다르다. 즉, 물리적으로 큰 면적을 가진 디스크의 바깥쪽의 트랙이 적은 면적의 안쪽 트랙보다 많은 섹터를 가지게하여 디스크 용량하고 전체적인 대역폭을 증가하기 위한 이 기법은 구현시 제어 방법이 고정 섹터를 갖는 과거의 디스크에 비해 다소 복잡하다는 단점이 있음에도 불구하고 많이 이용되고 있다.

동일한 트랙당 섹터수를 가지는 디스크를 모델로

사용한 연구들은 디스크의 데이터 대역폭이 트랙의 위치와 무관하게 하나의 고정된 값을 갖게 되므로, 이들 연구의 결과를 존 구조를 갖는 디스크에 사용할 경우 존 구조의 장점을 살리지 못하게 된다. 즉, 디스크 내 여러 존 중 대역폭이 가장 적은 맨 안쪽 구역의 대역폭이 디스크의 전체적인 대역폭으로 고정되어 결과적으로 전체 대역폭을 최소로 사용하게 된다. 따라서 디스크가 최소의 대역폭으로 사용되기 때문에 서버가 동시에 서비스 할 수 있는 사용자 수의 감소를 가져오며, 이는 서버 성능의 저하를 유발하게 된다. 또한 SCAN 방식을 사용할 때, 이와 같은 단일 대역폭 모델은 저장 공간의 과도한 낭비를 유발시킨다.

이런 단점을 보완하기 위해서 본 논문에서는 존 구조 디스크에 SCAN 방법을 적용하고, 디스크의 물리적 구역을 논리적 구역으로 재구성 시켜 존 구조 디스크의 특성을 살려 대역폭을 극대화하기 위한 방법과 이때, 디스크 전체 대역폭을 최대로 할 수 있는 전송 블록의 크기를 결정하는 방법을 제안한다. 또한 구역의 논리적 재구성 방식을 사용할 때, 물리적 저장공간의 낭비크기와 가변 크기의 데이터 블록(Variable Block Size) 전송시의 전송 블록 크기와 디스크 공간 낭비를 실험을 통해 분석한다.

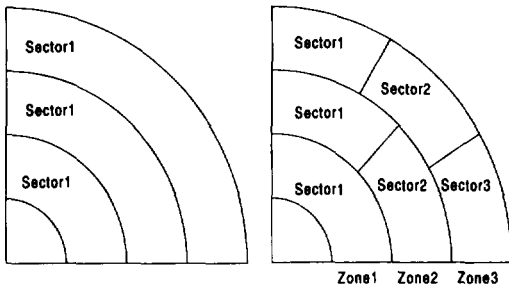
본 논문의 구성은 다음과 같다. 제 2장에서는 존 구조 디스크에 관하여 간단히 설명하고, 3장에서는 존 구조 디스크에서 구역의 재구성 방식과 디스크 공간 낭비를 최소화 하기 위한 최적 블록 전송크기 결정 방법을 제안한다. 4장에서는 제안된 알고리즘의 실험결과를 보이고, 5장에서는 결론과 향후 연구 과제에 대해서 기술한다.

2. 존 구조 디스크 구조

전통적인 하드 디스크에서는 트랙의 위치와 상관없이 트랙당 존재하는 섹터 수가 고정되어 있다. 트랙의 위치와 무관하게 트랙당 존재하는 섹터수를 동일하게 하기 위해, 모든 트랙의 섹터수는 가장 적은 트랙의 길이를 가지는 가장 안쪽의 트랙에 할당될 수 있는 섹터 수로서 결정되어지며, 이는 바깥쪽 트랙들의 공간을 낭비하는 결과를 가져온다.

이와 같은 디스크의 낭비를 줄이기 위해 최근의 고

속 대용량의 하드 디스크에서는 존 기법이라 불리는 새로운 방법을 사용하며, 존 구조를 가지는 디스크에서는 가변 트랙 사이즈 포맷을 사용한다. 즉, 존 구조 방식의 디스크에서는 트랙의 크기가 큰 바깥쪽의 트랙이 트랙의 크기가 작은 안쪽의 트랙보다 많은 수의 섹터를 할당받는다. 따라서 플랫터(platter)의 표면에 저장되는 데이터는 불 균일한 크기로 저장된다. 따라서 이를 제어하기 위한 방법은 복잡해지지만, 디스크의 표면의 저장 효율은 증가된다. (그림 1)에 트랙당 동일한 섹터수를 갖는 종래의 단일 구조의 디스크와 구역에 따라 트랙당 섹터수가 다른 존 구조 디스크를 간략히 도시하여 나타내었다. 하나의 존은 여러 개의 트랙으로 구성되며, 같은 존에 속한 트랙은 같은 수의 섹터를 가진다.



(a) 일반 디스크 구조 (b) 존 디스크 구조
(a) Traditional disk structure (b) Zone disk structure

(그림 1) 디스크 구조
(Fig. 1) Disk structure

존 구조를 가진 하드 디스크 드라이브에서 통상 3에서 20개 정도의 구역(동일 섹터수를 갖는 연속된 트랙 집합)을 가지며, 한 트랙당 섹터수는 3.5인치 디스크 경우에는 각각 제조 회사에 따르지만, 약 60에서 120정도의 섹터를 가지게 된다. 이렇게 함으로써, 이전의 디스크보다 존구조는 약 25%이상의 저장 용량의 증가를 가져올 수 있다. 구역 구조를 갖는 디스크의 예로 Quantum DSP3210, Seagate Baracuda ST15150, HP C2247을 <표 1>에서 비교하여 도시하였다. <표 1>에서는 내부 전송 속도에서 가장 안쪽의 구역과 가장 바깥쪽의 구역의 속도가 약 2배 가량 차이를 보임을 알 수 있고, 이는 디스크 효율 향상에 직접적으로 영향을 미친다.

<표 1> 존 구조 디스크 드라이브 예
<Table 1> An example of zone-structure disk drive

#of Zone		2	4	8	15
#of Sector/Track		95-146	59-119	.	122-232
Internal Transfer Rate (MB/sec)	Inner	5.75	2.7	2.02	5.6
	Outer	9	5.5	3.4	10.7
External Transfer Rate (MB/sec)	Fast SCSI-2	10	10	10	10
	Fast Wide SCSI-2	20	20	20	20

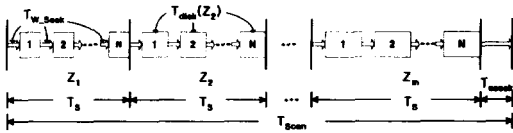
3. 논리적 구역의 재구성과 최적 블록크기

비디오 서버가 특정 비디오를 요구한 사용자의 요구를 받아 들여 비디오 전송을 시작하면 연속적인 디스플레이를 보장하기 위해서는 전송되는 비디오 데이터의 출력율은 사용자의 디스플레이 속도보다 더 빠르거나 최소한 같아야 한다. 이와 같은 대전제를 바탕으로 주어진 M개의 물리적인 구역을 가진 디스크를 논리적으로 재구성하여 전체 디스크의 출력율을 높이기 위해 본 논문에서는 물리적 구역을 논리적으로 병합하거나 분리하여 디스크 공간 낭비를 최소화하는 구역으로 만든다. <표 2>는 본 논문에서 사용하게 될 기호와 정의의 예를 보여주고 있다.

<표 2> 본 논문에서 사용된 기호와 정의
<Table 2> Definition of terminologies used in the paper

사용자의 전송 요구 속도
디스크에 존재하는 물리적 존의 수
디스크의 i번째구역 ($1 \leq i \leq M$)
디스크의 전체 크기
디스크의 i번째 구역에서의 대역폭
디스크의 i번째 구역의 크기
I번째 구역의 블록 크기(in VARB)
블록 크기(in FIXB)
재 구성된 i번째 논리적 구역의 크기
동시에 서비스가 다능한 최대 사용자수
한 구역에서의 최대 탐색시간(Rotational Latency를 포함)
I구역에서의 N 블록을 읽는데 걸리는 시간
m개의 구역에서 N 블록을 읽는데 걸리는 시간
가장 안쪽의 구역을 읽고 난 후 다음 서비스를 위해 가장 바깥쪽의 구역으로 헤드를 옮기는데 걸리는 시간
I번째 구역에서 해당 구역의 대역폭으로 B(Z)를 읽는데 걸리는 시간
S(Z)를 R(Z)로 읽어 내는 시간

여기서 M 개의 구역(Z_1, Z_2, \dots, Z_M) 중, Z_1 이 가장 바깥 구역이며, Z_M 이 가장 안쪽 구역이라 가정한다. 따라서 $R(Z_1) > R(Z_2) > \dots > R(Z_M)$ 이 성립한다. 또, 디스크에 영화 또는 미디어 X 를 저장하기 위해서는 SCAN기법에서와 같이 X 를 l 개의 블록으로 나누어 ($X_1, X_2, X_3, \dots, X_l$) 저장한다고 가정한다. 이때 나누어진 블록들은 모든 블록이 고정된 크기(Fixed Block Size : FIXB)를 갖거나 블록 마다 크기가 다른 가변 크기(Variable Block Size : VARB)를 사용하게 되며, 이 블록들을 디스크에 배치하는 방법은 각 구역에 라운드로빈 방식으로 저장하게 된다. 구역에 상관없이 배치하는 방법은 디스크가 하나의 대역폭을 가지는 기존의 디스크에서는 전체 출력율에 영향을 거의 미치지 않지만, 존 구조 디스크에서는 효과적이지 못하다. 왜냐하면, 여러 개의 비디오를 동시에 특정 디스크에 저장할 때에 존에 관계없이 저장한다면 극단적인 경우, 어떤 비디오는 나누어진 모든 블록이 모두 가장 바깥 존에 위치할 수 있는 반면, 어떤 비디오는 가장 느린 존에 위치할 수도 있다. 이러한 경우 두 비디오가 하나의 SCAN 주기에 같이 포함되어 있다면, 전체적인 출력 대역폭은 가장 안쪽 존의 가장 느린 대역폭으로 한정된다. 위와 같은 극단적인 경우가 아니더라도 여러 개의 비디오를 배치할 때 모든 비디오가 모든 존에 고루 분포하지 않으면 전체적인 대역폭 혹은 사용 가능한 디스크 용량의 손해를 감수 할 수 밖에 없다.[4] (그림 2)는 M 개의 존을 가진 디스크의 각 존에서 N 개의 블록 (N 개의 비디오 블록)을 SCAN 기법으로 읽어낼 때의 소요되는 시간 요소들을 나타내고 있다.



(그림 2) SCAN 기법으로 각 존에서 N 블록을 읽는데 소요되는 시간요소
 (Fig. 2) The required time elements to read out N blocks from each zone using the SCAN algorithm

FIXB 방법을 사용할 때는 모든 존에서 읽어내는 블록의 크기(B)가 같으므로 $T_{disk}(Z_i)$ 는 구역의 대역폭에 반비례하고, $T_s(Z_i)$ 역시 구역의 대역폭에 반비례관계에 있다. 이와 같이, 각 존의 $T_{disk}(Z_i)$ 및 $T_s(Z_i)$ 의 차이는(라운드 로빈 방식으로 비디오를 저장하는 경우)

디스크 공간을 낭비하는 결과를 가져온다. 왜냐하면, 저장 용량이 가장 적은 맨 안쪽의 존이 비디오 블록들을 저장하는데 소진되고 나면, 아직 바깥쪽의 존들에 저장 공간이 남아 있음에도 불구하고, 그 디스크에는 더 이상 데이터를 저장할 수가 없기 때문이다.

FIXB 방법에서 발생하는 디스크 저장공간의 낭비를 줄이기 위한 한 방안으로 VARB를 사용할 수 있다. [3]에서는 동일한 수의 사용자에게 동시에 비디오를 공급할 때, VARB 방법이 메모리 필요량, 낭비되어지는 디스크의 물리적 공간 등에서 FIXB 방법보다 우수함을 보이고 있다. VARB를 사용하는 경우, 각 구역의 대역폭에 따라서 블록크기($B(Z_i)$)를 설정되고, 해당 $B(Z_i)$ 에 의해 $T_{disk}(Z_i)$ 가 결정된다. [3]에서는 VARB 방법을 사용하면 디스크 공간 사용율을 증가시키는 동시에, 디스크 전체 출력율을 높이기 위해 MaxT 및 MinW를 제안하고 있다. MaxT는 디스크 공간 낭비를 감수하고 디스크 전체 출력율을 최대로 하는데 주안점을 두고 있는 기법이며, MinW는 디스크 전체 출력율을 저하를 감수하면서 디스크 공간 낭비를 최소화하는데 주안점을 두고 있는 기법이다. 두 기법에서는 물리적인 구역을 논리적으로 재구성하여 사용하고 있다. 하지만 두 기법 모두에서 디스크 전체 출력율은 이론적인 값에 크게 떨어지며, 디스크 공간의 낭비는 또한 심하다.

이상적인 경우 디스크 전체 출력율은 $\sum_{i=1}^M R(Z_i)S(Z_i)$ 이며, 디스크 공간 낭비는 없어야한다. 이를 만족시키기 위해서는 각 존의 디스크의 공간 크기와 출력율의 비를 일정하게 해야한다. 즉, VARB 방법을 사용할 때 이상적으로 필요한 구역의 크기는 다음과 같은 식을 만족할 때이다.

$$\frac{S(Z_1')}{R(Z_1')} = \frac{S(Z_2')}{R(Z_2')} = \dots = \frac{S(Z_k')}{R(Z_k')} = C \quad (\text{식.1})$$

위 (식.1)에서 C 는 상수로 이상적인 상태에서 디스크의 공간 크기와 출력율의 비를 나타내며, K 는 논리적으로 재구성된 존의 수를 나타낸다. 일반적으로 상용 디스크에서는 각 존의 디스크의 공간 크기와 출력율의 비가 일정하지 않으므로 이상적인 상태를 만족시키기 위해서는 각 구역을 재구성해야 할 필요가 발생한다. 하지만 실제 디스크 존들을 실수 크기의 이상적인 존으로 재구성하기는 불가능하므로, 본 논문에서

는 이상적인 값에 근사하는 존들로 재구성하는 방법을 제시한다.

이상적인 구역의 크기에 가까운 논리적 구역은 실제 물리적 구역들을 분할 또는 통합을 통해 재구성하여 생성한다. (식.1)에서 C값은 다음과 같이 구해진다.

$$S(Z_i') = R(Z_i') \times C$$

$$\sum_{i=1}^K S(Z_i') = C \sum_{i=1}^K R(Z_i')$$

$$C = \frac{\sum_{i=1}^K S(Z_i')}{\sum_{i=1}^K R(Z_i')} \quad (식.2)$$

하지만 C값은 실제 디스크에서 디스크 각 구역의 합을 각 구역들의 대역폭의 합으로 나눈 평균 서비스 시간(T_{avg_svr})보다 적거나 같은 최대 크기의 정수이다. 즉,

$$C \leq T_{avg_svr} \quad (식.3)$$

이때, $T_{avg_svr} = \frac{\sum_{i=1}^K S(Z_i)}{\sum_{i=1}^K R(Z_i)}$ 이다. 위 (식.2)를 만족하

는 최대 $S(Z_i')$ 값들은 전체 디스크의 낭비 공간을 최소화 하게 된다. 이와 같이 재구성된 구역들은 다음 조건들을 만족해야 한다.

- 1) $S(Z_i')$ 합은 물리적인 전체 디스크 크기를 넘을 수 없다. 즉, $\sum_{i=1}^K S(Z_i') \leq \sum_{i=1}^K S(Z_i)$
- 2) $K * C$ 값은 전체 디스크의 서비스 시간의 합보다 적어야 한다. 즉, $K \times C \leq \sum_{i=1}^K T_{svr}(Z_i)$
- 3) Z_i' 는 i 의 순서에 따라 연속된 공간에 배치된다.
- 4) Z_i' 의 대역폭은 Z_i 을 구성하는 물리적 구역들 중에서 가장 느린 구역의 대역폭이 된다.

제한하는 구역 재구성 기법에서는 물리적인 구역을 (식.1)을 만족하는 구역들로 재구성하기 위해, 물리적 구역을 여러 부분으로 나누는 구역 분할과 구역들을 논리적으로 합치는 구역 합병을 이용한다. 앞에서 열거한 조건을 만족하면서 이론적인 값에 가까운 존들을 구하는 알고리즘을 (그림 3)에 나타내었다.

```

MakeLogicalZone() {
    C= SUM(S(Zi))/SUM(R(Zi));
    Index_ref_zone = 1;
    Index_add_zone =2;
    SIndex_ref_zone = S(Z1);
    for(i=1; i ≤ M; i++) {
        Temp = C × R(ZIndex_ref_zone);
        S(Zi)=Sector_size × (Temp/Sector_size)
        If (S(Zi) > S(ZIndex_ref_zone)) {
            While (S(Zi) ≥ S(ZIndex_ref_zone)) {
                S(ZIndex_ref_zone) += S(ZIndex_ref_zone+1);
                ++Index_ref_zone;
                Temp = C × R(ZIndex_ref_zone);
                S(Zi)=Sector_size (Temp/Sector_size)
            }
        }
        SIndex_ref_zone -= S(Zi);
    }
}
    
```

(그림 3) 구역 재구성 알고리즘
(Fig. 3) Zone reconstruction algorithm

(그림 3)의 알고리즘에서 (식.1)의 C값은 (식.2)를 만족하는 최대치인 T_{avg_svr} 을 사용한다. 새로 구성될 첫 번째 논리 구역(Z_1')은 출력율이 가장 높은 물리적인 첫 번째 구역(Z_1)인 가장 바깥쪽의 구역으로부터 시작한다. 논리적 첫 번째 구역의 크기 $S(Z_1')$ 은 C값과 물리적 첫 번째 구역의 출력율인 $R(Z_1)$ 의 곱으로 정해진다. 대부분의 상용 디스크들에서 $S(Z_1)$ 은 $CR(Z_1)$ 보다 훨씬 크다. 따라서 Z_1 은 $S(Z_1')$ 과 $S(Z_1)-S(Z_1')$ 으로 분할된다. 하지만 디스크에서 읽어 내는 데이터의 크기의 최소 단위는 섹터이므로 실제 $S(Z_1')$ 은 섹터의 정수배에 가장 가까운 크기를 갖게 된다. 위 알고리즘에서 $\lfloor x \rfloor$ 는 x보다 적거나 같고 x에 가장 가까운 정수를 나타낸다. Z_2' 는 Z_1 의 나머지 부분인 $S(Z_1)-S(Z_1')$ 으로부터 시작한다. 그러므로, 앞에서 열거한 논리적 구역이 갖추어야 할 조건 중의 하나인 구역의 인접성을 그대로 유지하게 된다. $S(Z_2')$ 은 C값과 물리적 첫 번째 구역의 출력율인 $R(Z_1)$ 의 곱으로 정해진다. 만약 $S(Z_2')$ 가 $S(Z_1)-S$

(Z_1')보다 적으면, 다시 $S(Z_1) - S(Z_1')$ 은 $S(Z_1) - S(Z_1') - S(Z_2')$ 로 분할되고, 두 번째 논리 구역 또한 Z_1 내에 존재하게 된다. 따라서 첫 번째 물리 구역인 Z_1 은 3개 이상의 논리 구역으로 분할된다. 하지만 만약 $S(Z_2)$ 가 $S(Z_1) - S(Z_1')$ 보다 크면, $S(Z_1) - S(Z_1')$ 은 $S(Z_2)$ 와 병합되고 $S(Z_2')$ 는 C 값과 병합된 구역 중 가장 낮은 출력율을 가진 가장 안쪽 구역(이 경우는 Z_2)의 출력율인 $R(Z_2)$ 의 곱으로 정해진다. 왜냐하면 하나의 구역을 구성하는 모든 디스크 공간은 동일한 출력율을 가져야 하므로, 여러 개의 출력율이 다른 물리적 구역이 병합되어 하나의 논리적 구역을 형성할 때, 가장 낮은 출력율을 대표 출력율로 할 수 밖에 없기 때문이다. 동일한 방법으로 Z_3', \dots, Z_K' 를 정해간다. 정해가다가 $S(Z_{K+1}')$ 이 디스크의 남아 있는 공간 보다 크면, 논리적 재구성은 끝나게 되고 남아있는 공간은 버려지게 된다.

이때, 논리적 구역의 갯수 K 는 항상 물리적 구역의 수인 M 보다 항상 적거나 같다. 이는 C 값은 이론적인 최대치를 사용한 반면, 논리적 합병으로 이루어진 구역의 출력율을 가장 느린 출력율로 사용하고 있기 때문에 실제 논리적 구역의 총출력율이 물리적 출력율 보다 적게되나, 구역의 크기는 항상 맨 바깥쪽의 출력율을 기준으로 정해지기 때문에 실제 논리적 구역의 수는 물리적 구역의 수보다 적게된다.

제시된 알고리즘에 의해 재구성된 구역에서 VARB 방법을 사용하여 비디오 데이터를 각 구역에 저장할 때, 각 구역에서의 블록의 크기는 최대 동시 서비스 가능한 사용자 수에 의해서 (식.4)와 같이 결정될 수 있다.[10]

$$B(Z_i') = R(Z_i') \times \frac{R \times T_{w,seek} \times N \times K + R \times T_{c,seek}}{\sum_{i=1}^K R(Z_i') - R \times N \times K} \quad (식.4)$$

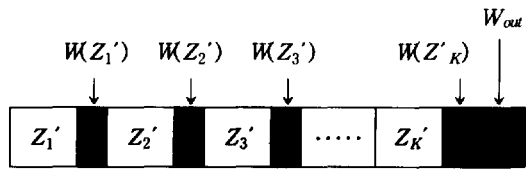
(식.4)에 의해서 결정되어지는 값은 실수 형태이다. 하지만 디스크에서 읽어 내는 최소의 단위는 섹터이므로 실제 사용되어질 블록의 크기를 결정하기 위해서 실수 값은 섹터 단위로 보정할 필요가 있다. 보정된 블록의 크기를 $B'(Z_i)$ 라 할 때, $B'(Z_i')$ 는 $Sector_size [1]$ 혹은, $Sector_size [1]$ 로 결정된다. 여기서 $l = B(Z_i') / Sector_size$ 이고 $[x]$ 는 x 보다 크거나 같

은 x 에 가장 가까운 정수를 나타낸다. 이들 두 값중 $B'(Z_i')$ 는 $B(Z_i')$ 와 오차가 적은 값으로 결정된다.

일반적으로 $B'(Z_i')$ 의 정수배가 $S(Z_i')$ 가 되지 않으므로 각 구역 내에서 디스크 공간의 낭비가 발생된다. 따라서 디스크 공간의 전체 낭비 (W_{total})는 앞에서 설명한 가장 안쪽의 물리적 구역에서 발생하는 디스크 공간의 낭비(W_{out})와 각 논리적 구역의 낭비를 포함해서 다음의 (식.5)와 같이 표현될 수 있다. (그림 4)는 이를 보여주고 있다.

$$W_{total} = \sum_{i=1}^K W(Z_i') + W_{out} \quad (식.5)$$

이 때, $W(Z_i') = S'(Z_i') - B'(Z_i') \times [S(Z_i') / B'(Z_i')]$ 이다.



(그림 4) 존 구조 디스크의 공간 낭비
(Fig. 4) The wasted disk space in zone-structure disk

4. 모의 실험 결과

앞에서 제시한 구역 재구성 알고리즘을 HP C2247 디스크에 적용하여 모의 실험한 결과를 <표 4>에 나타내었다. 실제 디스크에서 데이터를 읽어 내는 최소

<표 4> 재구성된 논리 구역
<Table 4> Reconstructed logical zones

1	3.4	324	663552	1	151.60	310471	3.4
2	3.17	112	229376	3	135.55	277597	3.04
3	3.04	76	155648				2.92
4	2.92	77	157696	5	113.25	231940	2.54
5	2.78	71	145408				
6	2.54	145	296960	7	101.21	207285	2.27
7	2.27	109	296960				
8	2.02	89	182272	W_{out}	16.28	33344	2.02

의 단위는 섹터이므로, 알고리즘에서 논리 구역의 크기는 섹터 단위로 보정된 값이다.

첫 번째 물리적 구역(Z_1)은 3부분으로 분할되어 두 부분은 논리 구역 Z_1' 과 Z_2' 로 할당되고 마지막 부분은 물리적 구역 Z_2 및 Z_3 의 일 부분과 합병되어 Z_3' 을 형성한다. 이와 같은 방법으로 재 구성된 논리 구역은 모두 8개로 물리 구역의 수와 동일하다. 논리 구역으로 재구성한 후 디스크의 낭비 공간은 W_{out} 으로 표시되는 맨 안쪽 구역의 일부분이다.

디스크 낭비 공간(W_{out})은 전체 디스크 공간의 약 1.62%가 낭비된다. 이는 알고리즘을 적용하지 않을 때 (44%)[10]에 비교하면 대폭으로 감소했음을 알 수 있다. 재구성된 디스크 전체 성능을 평가하기 위해서 디스크의 기대 대역폭(R_{EXP})을 사용한다. 기대 대역폭을 계산하기 위해서 구역에 데이터가 존재할 확률(P_i)과 기대 대역폭을 다음과 정의 한다.[10]

$$P_i = \frac{S(Z_i)}{S_{Total}}$$

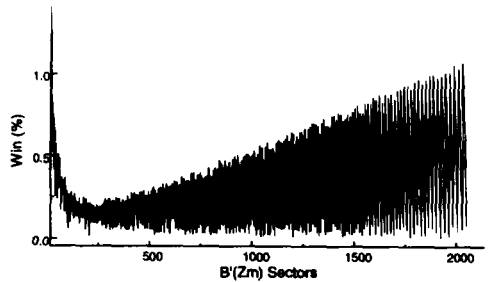
$$R_{EXP} = \sum_{i=1}^K P_i \times R(Z_i')$$

실제 디스크의 R_{EXP} 는 2.896(MB/sec)이며, VARB 방법에서의 재구성 알고리즘을 적용하기 전 R_{EXP} 와 적용 후의 R_{EXP} 는 2.817(MB/sec)[10]과 2.848(MB/sec)이다. 재구성 후의 R_{EXP} 가 실제 디스크에 가까운 값을 갖는다. 따라서 디스크 구역을 논리적으로 재구성함으로써 디스크 저장공간 낭비의 감소와 기대 대역폭의 증가를 동시에 기대할 수 있음을 알 수 있다.

전송 블록크기 $B'(Z_i')$ 값에 따라서 각 $S(Z_i)$ 영역에서 나머지 공간($W(Z_i')$)이 생기게 된다. $B'(Z_i')$ 을 1 섹터에서부터 2048섹터까지 변화 시켜 가면서 모의 실험을 통해 $\sum_{i=1}^K W(Z_i') + W_{out}$ 을 구한 결과를 (그림 5)에 보였다. 그림에서 모든 경우 디스크 낭비 공간의 크기 값이 전체 디스크의 크기의 1%이내임을 알 수 있다. 전송 블록크기 $B'(Z_i')$ 값의 적은 변화에 따라 낭비 공간의 차이가 심하게 나는 것은 다음과 같이 설명할 수 있다. 어떤 특정 $B'(Z_i')$ 값의 정수배가 $S(Z_i')$ 와 적은 차이를 보이고, 따라서 $W(Z_i')$ 가 적고 또한, $B'(Z_i')$ 에 대응하는 다른 구역의 전송 블록 크기(식.4에 의해 정해짐)에 따른 각 구역의 손실이 적은 경우

가 있는 반면, $B'(Z_i')$ 보다 크거나 적은 블록들의 경우는 반대의 상황이 발생하여 전체 디스크 공간의 낭비 차이가 많이 발생할 수 있다. 이와 같이 적은 $B'(Z_i')$ 의 변화에 각 구역의 디스크 낭비율은 민감하게 반응하므로 (그림 5)에서와 같이 낭비를 변화가 심한 모양을 보인다.

블록의 크기가 약 250섹터까지는 낭비율이 전체적으로 감소하는 경향을 보이다가 다시 꾸준히 증가하는 양상을 나타낸다. 이는 전송 블록의 크기가 커질수록 각 구역에서 발생하는 구역 조각짐(fragmentation)이 커지기 때문이다. 하지만 구역 크기가 약 250섹터 보다 적을 때는 오히려 전체적인 낭비율이 증가한다. 이는 (식.4)에 의해 구해진 전송 블록의 크기가 각 구역에서 많은 낭비를 가져오게 정해지기 때문이다.



(그림 8) 블록 크기에 따른 디스크 낭비 공간의 비율
(Fig. 8) The wasted disk space ratio with different block sizes

5. 결 론

본 논문에서는 최근 비디오 서버의 저장 매체로 사용하고 있는 고속, 대용량의 복수 구역 기록 구조 디스크를 위한 구역 재구성 알고리즘을 제안하였다. 복수 구역 기록 구조 디스크에 가변 크기 블록 전송 방법을 적용할 때, 실제 디스크에 존재하는 물리적 구역을 사용할 경우의 44%의 저장 공간의 낭비가 발생되나, 제안된 구역 재구성 알고리즘을 적용하여 논리적 구역으로 재구성함으로써 약 2%로 감소하였다. 논리적 구역으로 재구성되었을 경우에도 실제 디스크의 기대 대역폭에 가까운 기대 대역폭을 가지고 있음을 보임으로써 제안된 방법의 유용함을 알 수 있었다. 보정 과정에서 최적의 블록 크기를 선택하기 위한 알고리즘을 본 논문에서 제안하고, 최적의 블록 크기를 사용할 때

구역 내부에서 발생 할 수 있는 물리적 공간 낭비를 모의 실험을 통해 1%이내 임을 보임으로써 논리적 구역 재구성의 우수함을 보였다.

본 논문에서 제안한 알고리즘을 실제 시스템에 적용했을 경우, 본 논문에서 생략되어진 여러 가지 미미한 요소들이 디스크의 성능에 어떤 영향을 미치는가에 대한 연구가 향후 계속 진행될 것이다.

참 고 문 헌

[1] Y. Doganata and A. Tantawi, "Making a cost-effective video server," IEEE Multimedia, Vol.1, No.4, pp.22-30, Winter, 1994.

[2] C. Wu, et. al., "A Scalable architecture for video-on-demand server," IEEE Trans. on Consumer electronics, Vol.42, No.4, pp.1029-1036, Nov. 1996.

[3] W. Tseng, "A High performance video server for karaoke systems," IEEE Trans. on Consumer electronics, Vol.40, No.3, pp.329-336, Aug. 1994.

[4] Antoine N.Mourad, "Issues in the design of a storage server for video-on-demand," Multimedia Systems, No.4, pp.70-86, 1996.

[5] S. Chen, "A Performance evaluation of RAID architectures," IEEE Trans. on Computers, Vol. 45, No.10, pp.1116-1130, Oct. 1996.

[6] 조진성, 신현식, "디스크 배열을 이용한 실시간 멀티미디어 저장 서버에서의 스케줄링 기법", 한국정보과학회 논문지, 21권2, pp.221-229, 1994.

[7] C. Cheng, et. al.], "Effective utilization of disk bandwidth for supporting interactive video-on-demand," IEEE Trans. on Consumer electronics, Vol.42, No.1, pp.71-79, Feb. 1996.

[8] C. Ruemmler and J. Wilkes, "An Introduction to disk drive modeling," Computer, Vol.27, No.3, pp. 17-28, Mar. 1994.

[9] Margo Seltzer, Peter Chen, and John Ousterhout, "Disk Scheduling Revisited," Proc. of Usenix, 1990.

[10] Shahram Ghandeharizdedh and Seonho Kim, "Continous Display of Video Object Using Multiple Zone Disk :," Techncal Report USC94-592, 1994.



김 종 희

e-mail : jhkim@dist.co.kr

1996년 아주대학교 전자공학과 졸업(학사)

1998년 아주대학교 대학원 전자공학과(공학석사)

1998년~현재 대신정보통신 연구원

관심분야 : 컴퓨터 구조, 멀티미디어등



최 경 희

e-mail : khchoi@madang.ajou.ac.kr

1982년 Paul Sabatier 대학(프) 정보공학과 졸업

1982년~현재 아주대학교 정보 및 컴퓨터공학부 교수

관심분야 : 운영체제, 분산시스템, 멀티미디어, 실시간시스템



정 기 현

e-mail : khchung@madang.ajou.ac.kr

1990년 Purdue 대학(미) 전기공학과 졸업

1992년~현재 아주대학교 전자공학부 교수

관심분야 : 컴퓨터구조, VLSI 설계, 멀티미디어, 실시간시스템