

# 참조 정보를 이용한 RAID5 제어기의 부하 균형 반출 기법

장 윤 석<sup>†</sup> · 김 보 연<sup>‡</sup>

## 요 약

RAID5 제어기의 디스크캐쉬에 저장된 쓰기 요구들은 반출 알고리즘에 의하여 디스크로 쓰여진다. 반출 동작은 병렬 입출력 요구들에 대한 응답 성능에 많은 영향을 주기 때문에 RAID5 제어기의 성능을 향상시키기 위하여 여러 가지 반출 기법들이 연구되었다. 이들 반출 기법들 중 부하 균형 반출 기법은 디스크배열을 이루는 디스크들간의 부하 분포를 고려하여 동적으로 디스크간 부하 균형을 유지하므로 입출력 부하가 큰 경우에 우수한 응답 성능을 보인다. 그러나 부하 균형 반출 알고리즘은 디스크들간의 부하 균형을 유지하는 데에 중점을 두고 있기 때문에 일부 디스크들에 작은 크기의 데이터에 대한 부하가 집중될 경우, 쓰기 데이터들의 참조 지역성에 의한 이득을 충분히 이용하지 못한 채로 디스크간의 부하 균형을 유지하려고 하므로 병렬 입출력 성능이 효과적으로 향상되지 어렵다. 본 연구에서는 부하 균형 반출 기법에 디스크들의 참조 정보를 이용하여 반출 우선 순위를 결정하는 참조-부하균형 반출 기법을 적용한 새로운 RAID5 제어기를 제안하고, 시뮬레이션을 통하여 성능을 평가함으로써, 참조-부하균형 반출 기법이 기존의 부하 균형 반출 기법 이하의 RAID5 제어기의 입출력 성능을 개선할 수 있음을 입증하였다.

## The Load Balancing Destage Algorithm of RAID5 Controller using Reference History

Yun-Seok Chang<sup>†</sup> · Bo-Yeon Kim<sup>‡</sup>

## ABSTRACT

Write requests which stored in disk cache of the RAID5 controller should be destaged to disk arrays according to the destage algorithm. As the response performance of the parallel IO request is being hit by the effect of the destage, several destage algorithms have been studied to enhance the performance of the RAID5 controller. Among the destage algorithms, the load balancing destage algorithm has better performance than other destage algorithms when system load is highly increased. But the load balancing destage algorithm gives priority to load balance among the disks in disk arrays, therefore, when some disks are affected by the very heavy system load caused by small data requests, the load balancing destage algorithm cannot enhance the performance of parallel IO requests effectively since it makes effort to maintain the load balance without the benefit of the locality of the write requests. This paper proposes a new RAID5 controller that applies reference-load balancing destage algorithm which decides the destage priority based on the reference history and load distribution of the disks. The simulation results show that RAID5 controller with the reference-load balancing destage algorithm has better performance than previous load balancing destage algorithm.

※ 이 논문은 1998학년도 대전대학교 학술연구비 지원에 의한 것임

† 중신회원 대전대학교 이공대학 컴퓨터공학과 교수

‡ 순회원 경희대학교 동서의학대학원 교수

논문접수 1999년 9월 20일, 심사완료 2000년 1월 31일

## 1. 서 론

최근의 대형, 대용량 컴퓨터 시스템은 병렬처리 능력을 가지는 병렬 컴퓨터로 구성되고 있다. 이들 병렬 컴퓨터 시스템들은 대량의 데이터들에 대한 입출력 성능을 높이기 위하여 대용량, 고속의 입출력 성능 가지는 RAID(Redundant Arrays of Inexpensive Disks)와 같은 병렬 디스크 시스템을 점차로 많이 사용하고 있다 [3, 6, 7, 9]. 이 중에서 RAID5(RAID level 5)는 제어기에서 디스크캐쉬를 능동적으로 사용함으로써 그 응답 특성과 데이터 신뢰도가 다른 레벨에 비하여 가격 대비 성능이 우수하기 때문에 많은 상용 RAID 시스템들이 이를 지원하고 있으며, RAID5를 기초로 하는 병렬 디스크 구조와 병렬 디스크 제어기에 대한 다양한 연구들이 수행되고 있다[4, 12, 14]. RAID5 제어기는 호스트로부터 발생하는 입출력 요구들 중, 읽기 요구에 대해서는 여러 디스크에 대한 병렬 데이터 접근을 보장하므로 대량 읽기(Burst read) 요구에 대해서 우수한 응답 특성을 가지고 있다. 그러나 쓰기 요구에 대해서는 쓰기 동작에 소요되는 오버헤드가 매우 크기 때문에 이를 줄이기 위하여 비휘발성 메모리(Non-volatile RAM)로 쓰기 캐쉬를 구성하고, 적절한 반출 기법에 의하여 쓰기 캐쉬에 저장된 쓰기 요구들을 디스크로 반출(Destage)함으로써 쓰기 요구에 대한 RAID5의 응답 성능을 높이고 있다[8, 10, 13].

반출 기법은 쓰기 요구들을 선별하고, 디스크로 스케줄하는 알고리즘을 포함한다. 디스크 스케줄링 알고리즘은 디스크 입출력 성능에 매우 큰 영향을 끼치는 성능 요소이므로, RAID5 제어기의 반출 기법을 쓰기 요구들의 특성에 따라서 적절히 설계하면 입출력 성능을 효과적으로 향상시킬 수 있다[13]. 기존에 연구된 반출 기법에는 개개의 디스크에 대한 디스크 스케줄링 알고리즘을 개선함으로써 반출 성능을 높이는 반출 기법들과 디스크 배열 내의 디스크들을 집단적인 관점에서 다루어 디스크들간의 부하 균형을 이루는 데에 중점을 두는 부하 균형 반출 기법이 있다. 이 중에서도 부하 균형 반출 기법은 디스크배열을 구성하는 디스크들간의 부하 분포의 변화에 석용하는 특성을 기존의 반출 기법에 추가하였기 때문에 부하가 큰 경우, 가장 우수한 응답 성능을 보인다[2, 5, 16]. 그러나 부하 균형 반출 기법은, 비록 각 디스크의 관점에서는 디스크캐쉬의 지역성을 적절히 활용하지만, 개개의 디스크들의 현재 부

하 크기만을 고려하여 부하 균형을 유지하려고 하기 때문에 디스크들의 참조 패턴에 의하여 발생하는 디스크간의 참조 지역성을 무시하는 반출 동작을 자주 수행하는 문제점을 가지고 있다. 특히 작은 데이터에 대한 입출력 요구들이 크게 증가함으로써 일부 디스크들의 부하가 불균형적으로 매우 높을 경우에는, RAID5 제어기가 디스크간 부하 균형을 유지하기 위하여 참조 지역성은 높지만 해당 디스크의 부하는 낮은 쓰기 요구들을 반출시키는 경우의 수가 크게 증가된다. 이는 RAID5 제어기의 디스크캐쉬의 효율을 저하시키고, 결과적으로 입출력 요구에 대한 응답 성능을 저하시키게 된다.

일반적으로 디스크 장치에서 접근되는 입출력 요구들은 디스크로의 접근 분포에 의하여 형성되는 디스크 참조 지역성을 가지고 있으므로 RAID5로의 입출력 부하가 매우 높을 경우, 단순히 디스크간 부하 균형만을 고려할 것이 아니라 디스크캐쉬에서의 참조 지역성을 최대한 이용하면서 이를 적절히 이용하면 보다 효과적으로 쓰기 요구들을 디스크로 반출할 수 있고, 결과적으로 병렬 입출력의 효율을 증대시켜서 입출력 성능을 보다 향상시킬 수 있다. 따라서 본 논문에서는 RAID5를 구성하는 개개의 디스크들에 대한 참조 정보들을 각 디스크별로 독립적으로 유지하고, 반출 우선 순위를 결정할 때 이들 참조 정보들을 사용하는 반출 기법을 적용한 새로운 RAID5 제어기를 설계하였다. 또한 새로운 RAID5 제어기에 대한 모의 실험 도구를 구현하고, 시뮬레이션을 통하여 기존의 반출 기법을 적용한 RAID5 제어기들과의 성능을 비교 평가함으로써 입출력 부하가 매우 크고, 작은 데이터들로 인하여 특정 디스크에 부하가 집중되는 경우에도 비교적 적은 비용으로 균형하게 디스크간 부하 균형을 유지하면서 효율적으로 쓰기 요구들을 반출할 수 있음을 보였다.

본 논문은 다음과 같이 구성되었다. 제2장에서는 기존에 연구된 반출 기법들에 대한 관련 연구들을 고찰하고, 제3장에서는 참조 정보를 사용하는 부하 균형 반출 기법을 제안하고, 이를 적용한 RAID5 제어기를 설계하였다. 제4장에서는 제안된 반출 기법과 기존의 반출 기법들을 적용한 RAID5 제어기의 성능을 평가하기 위한 성능 평가 모델을 구성하고 시뮬레이션 도구의 구현 내용에 대하여 기술하였다. 제5장에서는 시뮬레이션 도구를 사용하여 성능을 평가한 결과 및 비교 평가 결과를 나타내고 제6장에서 결론과 향후에 병렬 입

출력 시스템의 연구 방향에 대한 내용을 기술하였다.

## 2. 관련 연구

기존에 연구된 디스크 반출 기법들로는 최소 비용 스케줄링 기법, Hi-low marking 기법, Linear threshold 스케줄링 기법, 그리고 부하 균형 반출 (LBD : Load balancing destage) 기법 등이 있다. 각 반출 기법들은 이전의 기법들을 포함하면서 이전 기법들의 문제점들에 대한 해결책을 제시하고 있다. 최소 비용 기법은 쓰기 요구를 반출시키는 평균 비용이 적기 때문에 부하가 낮은 경우에 유리하지만 참조 지역성을 고려하지 않기 때문에 쓰기 캐쉬의 장점을 적극적으로 이용하지 못하며, 최소 비용 계산을 위한 오버헤드가 크다는 문제를 가지고 있다. 또한 Hi-low marking 기법에서는 지역성을 활용할 수 있는 캐쉬의 데이터들이 조기에 축출(Flush)되는 현상을 방지하기 위하여 저장 한계를 두고 반출 동작을 수행하므로 참조 지역성을 이용할 수 있지만 저장 한계의 설정 값에 따라서 참조 지역성에 의한 디스크캐쉬 효율이 매우 크게 달라지는 문제가 있다[1]. 이들에 비하여 Linear threshold 스케줄링 기법은 쓰기 요구를 반출할 때 현재 캐쉬 내의 유효 데이터 점유율에 따라서 반출 속도를 조절함으로써 이전의 반출 기법의 장점들과 함께 디스크캐쉬에서의 참조 지역성을 최대한으로 이용할 수 있는 장점을 가지고 있다[13].

그러나 이들 반출 기법들은 디스크캐쉬 내의 모든 입출력 요구들을 하나의 집합으로 취급하기 때문에 단일 디스크를 사용하는 컴퓨터 시스템에서는 효과적이지만 RAID5와 같은 병렬 디스크 시스템에서는 최적이라고 할 수 없다. 이는 디스크배열을 구성하는 디스크간 부하 분포가 균일하지 못하고 비대칭적인 경우에 부하가 높은 일부 디스크로 인하여 병렬 입출력 처리 시간, 특히 병렬 읽기에 대한 처리 시간이 지연되는 문제를 고려하지 않고 있기 때문이다. 따라서 RAID5 제어기에서 쓰기 요구를 반출할 때에는 가능한 한 디스크간 부하 균형을 유지할 필요가 있다. 부하 균형 반출 기법은 쓰기 요구들을 각 디스크별로 유지하고 Linear threshold 스케줄링 기법에 의하여 반출되되, 부하가 가장 적은 디스크에 해당되는 쓰기 요구를 먼저 반출하도록 한다[2]. 디스크들간의 부하 분포는 수시로 변동하므로 이전에 연구된 반출 기법에서는 하나의 큰 읽

기 요구가 여러 읽기 요구들로 분할되어 병렬로 처리될 때, 부하가 집중된 디스크에서의 처리가 지연되어 결국 병렬 읽기 요구에 대한 처리 시간이 길어질 수 있다. 그러나 부하 균형 반출 기법은 각 디스크간 부하를 가능한 한 균등하게 조정하여 특정 디스크에서 큰 지연 시간을 가지지 않도록 함으로써 병렬 입출력 요구들의 처리 지연 시간을 줄일 수 있는 장점을 가지고 있다.

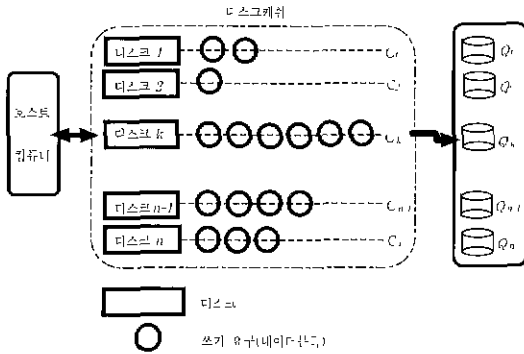
## 3. 참조 정보를 기반으로 하는 부하 균형 반출 기법

### 3.1 부하 균형 반출 기법의 문제점

(그림 1)은 부하 균형 반출 기법의 동작을 나타낸다. 쓰기 캐쉬 내에 저장된 쓰기 요구들은 반출될 디스크별로 LRU 순으로 리스트 체인(List chain)을 형성한다. 반출할 기회가 되면 RAID5 제어기는 Linear threshold 스케줄링 기법에 의하여 쓰기 요구들을 반출한다. 이때 디스크배열의 디스크들 중 대기 큐의 길이  $Q_k$ 가 가장 작은 값을 가지는 디스크에 해당되는 리스트 체인 ( $C_k$ )의 쓰기 요구가 반출되며 매번 반출될 때마다  $Q_k$  값들을 비교하게 된다[2, 16].

그러나 호스트로부터 발생되는 입출력 요구들이 모두 병렬 입출력 요구인 것은 아니다. 만약에 작은 크기의 데이터에 대한 단일 입출력 요구들이 일부 특정 디스크들( $D_{hot}$ )에 매우 심하게 집중되는 상태에서 부하 균형 기법이 적용되면, RAID5 제어기는 부하가 상대적으로 작은 디스크들( $D_{cool}$ )에 해당되는 쓰기 요구들을 우선적으로 반출하게 된다. 이 경우, 호스트로부터  $D_{hot}$ 에 전달될 입출력 요구들에는 읽기 요구뿐 아니라 쓰기 요구도 많아지므로 이 디스크에 해당되는 리스트 체인의 길이가 증가하면서 쓰기 요구들이 캐쉬 내에 머물게 되는 시간이 증가되는 반면에  $D_{cool}$ 의 리스트 체인들에 연결된 쓰기 요구들은 캐쉬 내에 머무는 시간이 짧아지게 된다. 따라서 쓰기 캐쉬 차원에서의 LRU 데이터는  $D_{hot}$ 에 존재할 확률이 증가되지만 반출될 확률이 낮아지고,  $D_{cool}$ 에는 MRU 데이터가 존재하게 될 확률은 증가되지만 반출될 확률이 높아지게 된다. 따라서 부하가 매우 집중되는 입출력 특성을 가지는 응용이 수행될 경우에는 부하 균형 반출 기법 자체가 참조 지역성에 의한 이득을 감소시키는 결과를 가지오게 된다. 이러한 결과는 디스크캐쉬의 효율을 떨어뜨리고

멀티미디어 데이터와 같은 대용량의 데이터 처리를 위주로 하는 시스템이 아닌 병렬 처리 시스템의 경우 전체적인 입출력 성능을 저하시키게 된다.



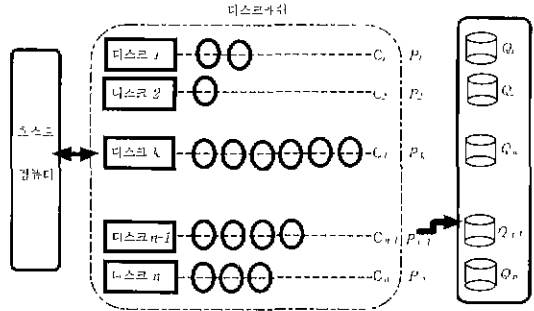
(그림 1) 부하 균형 반출 기법 개요

이러한 문제를 개선하기 위해서는 부하 균형 반출 기법을 적용할 때, 각 디스크들의 부하 분포 외에 입출력 요구들의 참조 지역성도 고려하여 반출 우선 순위를 결정하고, 이 우선 순위에 따라서 쓰기 요구들을 반출하도록 할 필요가 있다. 이와 같이 반출 우선 순위를 결정하기 위해서는 RAID5 제어기는 캐쉬에 저장된 쓰기 데이터들에 대한 참조 정보를 이용하여야 한다

3.2 참조-부하균형 반출 기법

부하 균형 반출 기법에서는 쓰기 요구들을 디스크별로 LRU 리스트를 만들고, 반출할 디스크의 결정은 물리적인 입출력이 수행될 때마다 전송되는 디스크 큐 길이들의 탐색으로 완료될 수 있으므로  $O(1)$ 의 복잡도로 반출 기법이 동작될 수 있다. 그러나 여기에 반출 우선 순위를 부여하려면 우선 순위를 계산할 때마다 리스트 탐색과 재배열을 동시에 수행하여야 하므로 그 복잡도는  $O(n^2)$ 으로 급격히 증대된다. 반출 우선 순위 계산의 복잡도를 줄이기 위해서는 부하 균형 반출 기법에 참조 지역성을 고려할 때 가능한 한 최소한의 참조 정보만을 활용하여야 한다. 참조 정보로는 디스크 별로 반출될 쓰기 요구, 즉 각 디스크  $i$ 에 해당되는 리스트 체인의 마지막에 연결된 쓰기 요구, 즉 반출 요구에 대한 참조 순위  $Freq(i)$ 을 사용하였다. 이는 반출 요구들의 LRU 리스트상의 위치를 순서대로 정렬한 순서 위치  $k$ 를 정규화한 값으로 나타내며, 가장 작은

$Freq(i)$  값을 가지는 디스크에 해당되는 쓰기 요구가 반출 우선 순위가 가장 높다.



(그림 2) 참조 정보를 포함하는 우선 순위 반출 기법

$$Freq(i) = k / n \quad (0 < Freq(i) \leq 1)$$

$k$ : 디스크  $i$ 로 반출될 데이터의 LRU 리스트상의 위치 순서(1, 2, 3, ...,  $n$ )  
 $n$ : 디스크의 수

참조 순위와 마찬가지로 각 디스크  $i$ 에 대한 부하 순위  $Load(i)$ 도 각 디스크  $i$ 의 부하 크기를 순서대로 정렬한 순서 위치  $w$ 를 정규화한 값으로 나타내며, 이는 다음과 같이 계산할 수 있다.

$$Load(i) = w / n \quad (0 < Load(i) \leq 1)$$

$w$ : 디스크  $i$ 의 부하 크기에 따른 순서 (1, 2, 3, ...,  $n$ )

쓰기 요구도 시간 지역성에 의하여 최근의 캐쉬 적중률이 높은 데이터 블록들은 이후에도 적중될 확률이 높기 때문에 해당 디스크의 부하가 상대적으로 낮다고 하여 기계적으로 반출시키게 되면 캐쉬 적중 실패로 인한 오버헤드가 입출력 처리 지연시간보다도 커질 수 있다. 따라서 적절한 반출 요구를 결정하기 위해서는 다음과 같이  $Load(i)$  값과  $Freq(i)$ 를 동시에 고려하는 반출 우선 순위 계산 방법을 사용한다

$$P(i) = Load(i) + Freq(i)$$

$P(i)$  : 디스크  $i$ 에 해당하는 리스트 체인의 반출 우선 순위  
 $Load(i)$  : 디스크  $i$ 의 부하 순위  
 $Freq(i)$  : 디스크  $i$ 의 참조 순위

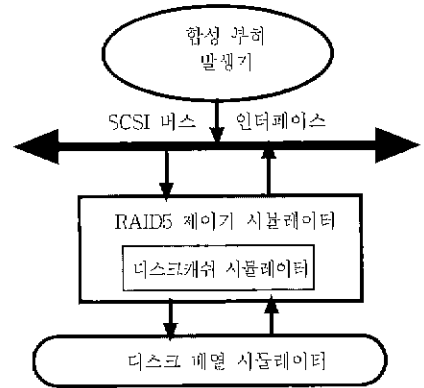
$Load(i)$  값과  $Freq(i)$  값이 각각 어떻게 변화하는가에 따

라서 반출 우선 순위는 달라질 수 있다.  $Freq()$  값이 0으로 수렴하면 반출 우선 순위는 부하 균형 반출 기법의 결과와 같게 되고,  $Load()$  값이 0으로 수렴하면 단순히 LRU 방법만을 적용하는 결과가 된다. 여기서 각 디스크에 대한 부하 순위와 참조 순위는 각각 성구화된 값을 사용함으로써 부하의 종류에 따른 참조 순위 변화량과 부하 순위 변화량의 상대적인 크기에 관계없이 반출 우선 순위를 계산할 수 있도록 한다. 이와 같은 방법으로 반출 우선 순위를 결정하여 반출시키는 기법을 참조-부하균형 반출(R-LBD: Reference-LBD) 기법이라고 한다. (그림 2)는 참조-부하균형 반출 기법에 의하여 참조 정보를 포함하여 반출 우선 순위를 계산한 결과에 따라서 부하 균형 반출을 수행하는 과정을 나타낸다. 그림에서는  $P_{k-1}$  값, 즉 디스크  $k-1$ 의 우선 순위가 가장 높게 계산되었을 경우 디스크  $k-1$ 의 리스트 체인에 해당되는 데이터 블록들을 반출하게 된다.  $Load()$ 와  $Freq()$  값들은 매번 반출이 수행될 때마다 갱신된다. 이와 같이 참조 정보를 포함하여 계산된 우선 순위에 의하여 반출을 수행할 경우 부하 균형 반출 기법의 장점을 그대로 이용하면서 반출 요구들에 대한 참조 지역성을 활용될 기회를 높일 수 있으므로 병렬 읽기의 성능을 증대시킬 수 있다.

#### 4. 시뮬레이션 환경

참조-부하균형 반출 기법을 적용한 RAID5 제어기의 성능을 측정하고, 이전의 반출 기법들과의 성능을 비교, 평가하기 위한 성능 평가 도구로 본 연구에서는 다양한 반출 기법을 사용하는 RAID5 제어기 시뮬레이터를 구현하였다. 이를 위하여 RAID 연구 초기부터 사용된 Berkeley와 CMU의 RAID 시뮬레이션 프로그램인 raidSim을 이용하였다[9, 12]. 본 연구에서는 이 raidSim을 기본으로 하여 시뮬레이션 모델을 설정하고, 디스크캐쉬를 포함하는 RAID5 제어기를 구현한 RAID5 모델을 (그림 3)과 같이 재구성하였다. 여기서 합성 부하 생성기는 Berkeley의 부하 생성기를 기반으로 구현하였고, CMU의 RAID5 시뮬레이터에 Dincro-III 캐쉬 시뮬레이터를 기반으로 구현된 디스크캐쉬 시뮬레이터를 결합하여 하나로 구현한 RAID5 제어기 시뮬레이터를 새로 구성하였다. 그리고 디스크매일은 HP의 디스크 모델링 기법을 적용하여 구현한 다음 RAID5 제어기 시뮬레이터에 연결되도록 구현함으로써 RAID5 제어기 시

뮬레이터로부터 사건-구동(Event-driven) 방식에 의하여 구동될 수 있도록 하였다[9, 13].



(그림 3) 시뮬레이션 모델의 구성

#### 4.1 합성 부하 발생기

RAID5 제어기의 반출 기법을 개선함으로써 병렬 디스크 시스템의 성능이 향상됨을 증명하기 위해서는 특정한 조건 아래에서만 나타나는 입출력 패턴에 대한 성능이 아니라 다양한 조건하에서 나타나는 입출력 패턴들에 대하여 성능을 비교하여야만 객관적인 비교 평가를 수행할 수 있다. 따라서 RAID5 제어기의 성능을 평가하는 데에 사용할 시뮬레이터의 입력은 입출력의 종류, 크기, 분포, 부하 발생률 등의 여러 입출력 요소들을 반영할 수 있어야 한다. 이를 위해서는 실제의 병렬 처리 시스템으로부터 입출력 트레이스를 추출하여야 하지만 이와 같은 입출력 트레이스를 얻기는 사실상 매우 어렵다. 따라서 대부분의 RAID5 관련 연구에서는 인위적으로 입출력 부하를 발생시켜 주는 부하 생성기를 구현하여 시뮬레이터의 입력으로 사용하고 있다. 따라서 본 연구에서도 이전의 연구에서 많이 사용하는 Berkeley의 raidSim에 구현되었던 합성 부하 발생기에 입출력 종류, 즉 읽기와 쓰기의 발생 비율과, 입출력 크기의 발생 비율, 즉 작은 데이터와 멀티미디어 데이터의 비율을 조절할 수 있는 기능들을 추가함으로써 다음과 같은 부하 요소들을 조절하여 필요한 조건의 부하를 발생시킬 수 있도록 합성 부하 발생기를 수정하였다.

- 입출력의 종류 (read/write), 발생 비율 및 발생 확률(probability)

- 입출력 크기의 분포(deterministic / exponential)
- 입출력 크기의 발생 비율(small data/ multimedia data)
- 지역성을 발생시키기 위한 지역 공간(local region)의 위치와 크기

raidSim의 합성 부하 발생기와 같이 수정된 합성 부하 발생기도 디스크매열상의 모든 데이터 블록에 대한 입출력 요구들을 임의로 발생시키기보다는 다중 프로세스 그룹들에 의하여 입출력 요구의 일부가 지역성을 가지도록 하고 있다. 각 프로세스 그룹은 부하 요소들을 개별적으로 임의로 조정할 수 있다. 각 프로세스 그룹은 30개의 프로세스들로 구성된다. 입출력 종류의 발생 비율과 입출력 크기의 발생 비율을 조정하는 기능이 추가된 것은 이전의 RAID5 연구들에서 제외되었던 연속적인 데이터에 대한 조건을 작은 데이터에 대한 처리 환경과 동시에 고려할 수 있도록 함으로써 최근의 대형 컴퓨터 시스템들의 입출력 행태에 대한 보다 현실적인 성능 평가가 가능하게 하기 위한 것이다. 그리고 raidSim을 사용한 이전의 연구들과의 비교 분석이 용이하도록, 데이터 크기 이외의 모든 조건값과 여러 파라미터들은 이전 연구들과 동일한 값들을 사용하였다[9, 12, 13, 16].

#### 4.2 디스크 배열 시뮬레이터

RAID5 디스크 시스템은 보통, 동일한 특성을 가진 다수의 디스크들을 이용하여 병렬 입출력을 수행할 수 있도록 디스크 배열로 구성된다. 따라서 RAID5의 디스크 배열을 시뮬레이션하기 위해서는 개개의 디스크 장치에 대한 시뮬레이션 환경을 구축하여 RAID5 제어기와 결합하여야 한다. 본 논문에서는 디스크 장치를 시뮬레이션하기 위하여 먼저 디스크 모델을 설계하고 모델에 따른 동작을 수행하는 시뮬레이션 도구를 구현하였다[11]. 구현된 시뮬레이션 프로그램에서 적용된 디스크 모델은 HP의 모델링 기법을 사용하였으며 여기에는 동작 시간 계산을 위한 요소 즉, 데이터에 대한 탐색 시간(Seek time), 회전 지연 시간(Rotational delay), 데이터 전송 시간(Data transfer time)들과 디스크 동작을 제어하기 위한 요소 즉, 입출력 인터페이스(I/O interface), 입출력 큐, 트랙 버퍼(Track buffer), 디스크 스케줄링 기법(Disk scheduling algorithm) 등이 고려되었다[15].

#### 4.3 RAID5 제어기 시뮬레이터

##### 4.3.1 RAID5 제어기 모델

RAID5의 시뮬레이션을 정확하게 수행하기 위해서는 RAID5를 구성하는 각 구성 요소들의 구조와 상호간의 동작 관계를 명확하게 모델링 하여야 한다. RAID5는 여러 디스크들을 집합적으로 구동하므로 디스크 배열의 제어 기능에 대한 모델이 필요하게 된다. 그러므로 RAID5의 시뮬레이션을 위해서는 일반적인 디스크 구성 요소에 대한 모델 외에 RAID5 제어기에 대한 모델이 추가된 시뮬레이션 도구가 요구된다. 여기에는 SCSI 버스 인터페이스 모델, RAID 분할기(Striper) 모델, 페리티 제어기 모델, 그리고 디스크캐쉬 모델이 포함된다[8, 14]. 본 연구에서 RAID5 분할기와 페리티 제어기 모델은 Berkeley의 radSim을, SCSI 버스 인터페이스 모델은 HP의 방법을 사용하여 시뮬레이션 도구를 구현하였으며 본 논문의 중심이 되는 디스크캐쉬 모델은 Varma의 방법에 의하여 구현하였다[13]. 디스크캐쉬 모델은 디스크캐쉬 운영 기법과 반출 기법들을 포함하고 있으며 이전에 연구된 반출 기법들과 본 논문에서 제안한 참조 정보를 사용하는 참조-부하균형 반출 기법이 구현되었다.

##### 4.3.2 디스크캐쉬 시뮬레이터

본 연구에서는 빈출 입출력 요구를 처리하는 반출 기법들의 성능을 서로 비교 분석하기 위하여 기존의 연구들에서 제안된 여러 반출 기법들과 참조 정보를 사용하는 반출 기법을 디스크캐쉬 시뮬레이터에 구현하였다. 구현한 디스크캐쉬 시뮬레이터는 Dinero-III 캐쉬 시뮬레이터를 기본으로 하고 여기에 여러 반출 기법들을 추가하여 구현되었다. 반출 입출력 동작들은 background로 수행된다. 특히 부하 균형 반출 기법과 참조 정보를 사용하는 반출 기법을 구현하기 위하여 본 연구에서는 제3장에서 기술한 바와 같이 각 디스크별 큐 길이를 동적으로 파악하는 자료 구조인 리스트 체인을 사용하였다. 리스트 체인은 각 디스크 상에서 입출력을 수행할 때마다 새로이 변경된다. 부하 균형 반출 기법과 참조 정보를 이용한 반출 기법은 이 자료구조를 참조하여 쓰기 캐쉬로부터 반출될 데이터를 결정한다.

#### 5. 시뮬레이션 및 결과

##### 5.1 시뮬레이션 환경 구성 요소

시뮬레이션에 사용한 RAID5 모델의 디스크배열은

모두 20개의 HP 97560 디스크를 사용하여 2차원 배열로 구성하였다. 이는 2개의 열로 구성되며 각 열은 10개의 디스크들을 포함한다. 디스크상의 데이터는 4KB의 기본 블록 크기(Block size)를 가지며 개개의 디스크 입출력은 기본 블록 크기의 배수로 수행된다. 분할 단위(Striping Unit)의 크기는 OLTP(On-line transaction proccssing)로부터 멀티미디어 응용에 이르기까지의 다양한 응용이 적용되는 경우를 시뮬레이션할 수 있도록 4KB~64KB내에서 변화하도록 하였으며 시뮬레이션에서 명시되지 않은 경우에는 16KB로 설정하였다. 전체 디스크 공간이 20GB를 넘는 크기이므로 읽기 캐쉬의 크기는 전체 디스크 공간의 1% 이상인 256MB로 설정하였다. 쓰기 캐쉬의 크기는 16MB로부터 128MB 까지 변화시킬 수 있도록 하였다. 두 캐쉬는 모두 기본적으로는 LRU 교체 전략을 사용하며 캐쉬 블록의 크기도 4~64KB의 범위에서 변화시키면서 시뮬레이션을 수행할 수 있으나 여기서는 8KB 블록으로 설정하였다. 그리고 입출력 데이터의 크기는 32KB를 기준으로 하고, 입출력 요구의 구성비는 임의로 변화시키지 않는 경우에는 읽기 60%, 쓰기 40%의 비율로 구성되었다[13]. 기타 다른 조건값들은 이전의 raidSum을 사용하는 연구들과 동일하게 설정하였다.

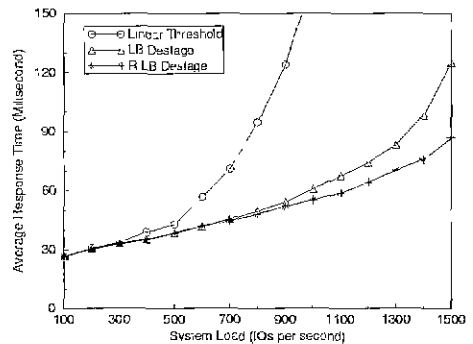
5.2 성능 평가 요소

빙털 디스크 시스템은 매우 많은 구성 요소들을 포함하고 있기 때문에 시뮬레이션을 통하여 평가하여야 하는 요소도 매우 다양하고, 여러 측면에서 성능을 평가하여야 하므로 매우 많은 실험 결과들을 사용하여 체계적인 분석을 하여야 할 필요가 있다. 이를 위하여 이전의 여러 연구에서 사용하였던 대표적인 성능 평가 요소들로는 평균 입출력 응답 시간, 단위 시간당 최대 처리량, 대량 입출력 데이터에 대한 처리 지속 특성 등이 있다. 그러나 컴퓨터 시스템의 성능에 직접적으로 영향을 미치는 가장 중요한 성능 평가 요소는 입출력 요구, 특히 읽기 요구에 대한 평균 응답 시간(Average response time)이다 따라서 본 논문에서는 시뮬레이션 환경을 구성하는 여러 환경 변수들을 다양하게 변화시키면서 참조 정보를 사용하는 부하 균형 반출 기법과 다른 반출 기법들, 특히 부하 균형 반출 기법을 사용한 RAID5 제어기의 읽기 요구에 대한 응답 시간의 변화를 측정하여 비교, 분석함으로써 입출력 성능을 평가하였다 또한 디스크캐쉬에서의 캐쉬 적중률과 캐쉬

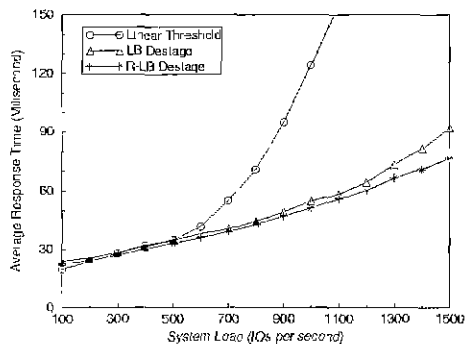
점유율을 분석함으로써 반출 기법에 따른 디스크캐쉬의 성능을 평가하고, 디스크캐쉬의 성능과 입출력 응답 성능과의 관계를 평가함으로써 참조 정보를 사용하는 부하 균형 반출 기법에 효과적인 디스크캐쉬 구성 방법을 제시하였니

5.3 실험 결과

본 논문에서 구현한 모의 실험 도구를 사용하여 수행한 시뮬레이션은 기존 연구 중 Linear threshold 스케줄링 기법 및 부하 균형 반출 기법과, 본 논문에서 구현한 참조-부하균형 반출 기법을 사용한 RAID5 제어기의 입출력 성능을 다양한 조건하에서 비교, 평가하는 것이다. 이를 위하여 부하를 증가시키면서 시뮬레이션 환경의 각종 환경 구성 요소들의 변화에 따른 시뮬레이션을 수행하고, 시뮬레이션의 수행 결과로 도출되는 각 반출 기법들의 상대적인 입출력 성능을 비교, 평가하였다.



(a) Write Cache Size=16MB

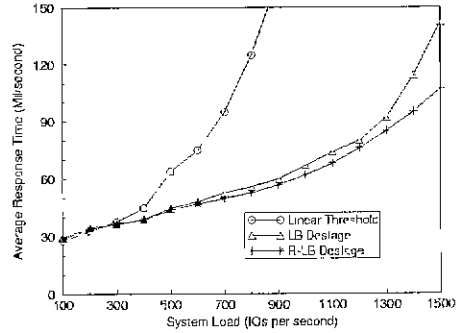


(b) Write Cache Size=128MB

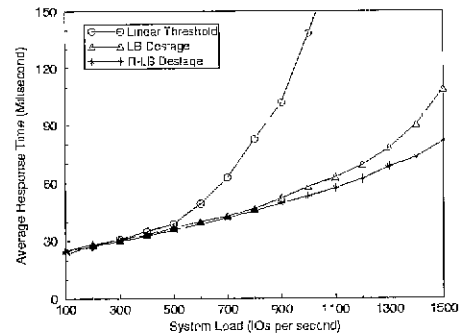
(그림 4) 부하의 증가에 따른 응답 시간의 변화

(그림 4)에서 (그림 7)까지는 각 반출 기법별로 부하 (System load)의 증가에 따른 읽기 요구에 대한 응답 시간의 변화를 나타낸다. (그림 4)에서는 쓰기 캐쉬의 크기가 각각 16MB와 128MB인 경우에 부하의 증가에 따른 성능 변화를 보여준다. 여기서 Linear threshold 스케줄링 기법은 부하가 증가함에 따라서 평균 응답 시간이 급격히 증가하고 있으며, 부하 균형 반출 기법은 부하가 크지 않을 때에는 비교적 선형적인 응답 특성을 보이고 있으나 부하가 매우 큰 경우에는 역시 지수적인 응답 특성을 나타내고 있다 이는 합성 부하 발생기가 부하를 발생시킬 때 디스크 공간의 일정 영역에 대한 접근을 많이 수행함으로써 지역성을 만드는 부분 공간을 가지도록 설계되었기 때문이다. 따라서 일부 디스크에 집중적인 입출력이 발생될 확률이 비교적 높기 때문에 부하가 매우 높을 때에 부하 균형 반출 기법이 지역성을 가지는 반출 요구들을 반출할 확률이 많아지게 된다. 이러한 반출 요구들은 디스크에서의 큐 길이를 증가시켜 이후에 디스크 큐에 대기하는 읽기 요구들의 대기 시간을 증가시키고, 결국 평균 응답 시간이 증가된다. 그러나 참조-부하균형 반출 기법은 지역성을 가지는 반출 요구들을 보존할 확률을 높일 수 있으므로 반출 요구들이 반출된 후, 즉시 재 저장될 확률을 줄여서 지수적인 응답 특성을 최소화한다. 쓰기 캐쉬의 크기가 작을수록 이러한 경향은 분명해지지만, 이는 부하의 특성, 즉 읽기-쓰기 요구의 비율이나 입출력 데이터의 크기에 의하여 크게 달라진다.

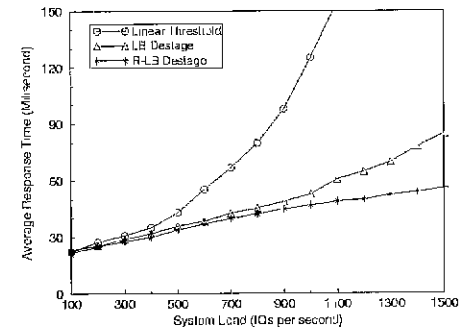
(그림 5)는 부하에 포함된 읽기 요구와 쓰기 요구의 비를 변화시키면서 부하의 크기를 증가시켰을 때, 읽기 요구에 대한 응답 시간의 변화를 나타내고 있다. 여기서 쓰기 캐쉬의 크기는 64MB로 설정하였다. 어느 반출 기법이나 읽기 요구가 많아질수록, 즉 쓰기 요구의 비가 적을수록 반출 기법이 응답 성능에 미치는 효과는 적어진다. 이는 대부분의 읽기 요구들이 요구 읽기(demanded read)이기 때문에 디스크 큐에서의 우선 순위가 높고, 이에 따라서 상대적으로 수가 적은 반출 요구들로는 디스크배열의 전체적인 부하 균형을 유지하기 어렵기 때문이다. 따라서 디스크간 부하 균형을 고려하지 않는 Linear threshold 스케줄링 기법을 제외하고는, 쓰기 요구의 비가 증가하면 부하 균형 기법에 의한 응답 시간 개선 효과가 분명해진다. (그림 5(c))의 결과는 쓰기 요구가 읽기 요구에 비하여 대단히 많고 부하가 매우 높을 때에는 참조-부하균형 반출 기법



(a) Read Write = 80 : 20



(b) Read : Write = 60 : 40

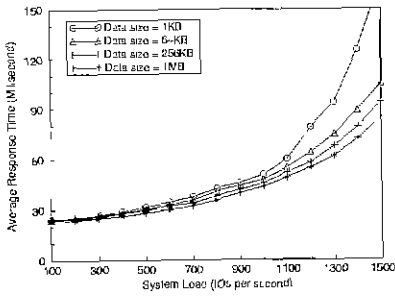


(c) Read . Write = 30 70

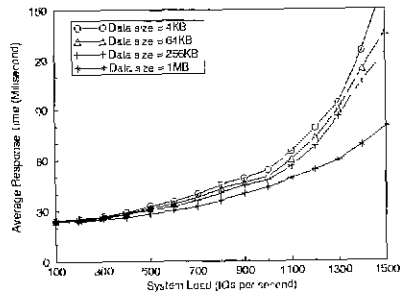
(그림 5) 입출력 요구의 구성비에 따른 응답 시간의 변화

이 비교적 우수한 응답 성능을 보인다. 이는 참조-부하균형 반출 기법 (그림 4)의 결과처럼 반출 요구를 반출-재저장하는 확률을 효과적으로 줄일 뿐만 아니라 부수적으로 쓰기 요구들 중 반출될 확률이 높은 쓰기 요구들의 캐쉬 적중률을 높여서 디스크로의 물리적인 반출 요구의 수를 줄여 주기 때문이다.

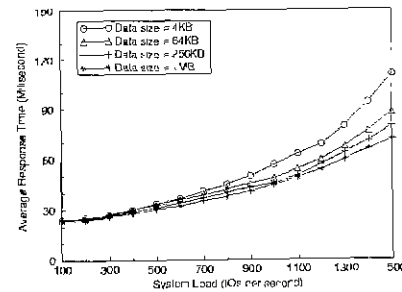




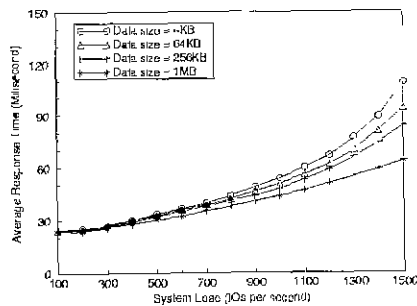
(a) Load Balancing, Stripe Unit = 32KB



(b) Load Balancing, Stripe Unit = 512KB



(c) Reference-Load Balancing, Stripe Unit = 32KB



(d) Reference-Load Balancing, Stripe Unit = 512KB

(그림 6) 입출력 데이터의 크기와 분할 단위에 따른 응답 시간의 변화

(그림 6)은 시뮬레이션에서 RAID의 분할 단위를 일정 크기로 설정한 경우, 입출력 데이터의 크기를 변화시켰을 때의 부하 균형 반출 기법과 참조-부하균형 반출 기법에서의 읽기 요구에 대한 응답 시간의 변화를 나타낸다. 합성 부하 발생기는 기준 입출력 크기를 중심으로 정규분포를 따르는 입출력 데이터들을 생성하므로 기준 크기에 따라서 다양한 크기의 입출력 데이터가 혼합된 부하를 발생시킨다. 시뮬레이션에 적용된 RAID5 모델은 입출력 데이터의 크기가 분할 단위를 초과하면 RAID5 데이터에서 하나의 입출력 요구를 병렬 입출력 요구로 변환하여 처리하도록 설계되어 있다. 따라서 입출력 요구의 기준 입출력 크기가 분할 단위보다 작은 경우에는 개별 입출력으로 처리된다. 따라서 기준 입출력 크기가 작은 경우에는 개별 입출력 요구들이 많아지고, 또한 합성 부하 발생기에 의하여 생성되는 부하는 지역성을 가지고 있으므로, 이들 개별 입출력 요구들이 일부 디스크들로 집중되는 경향을 나타낼 확률이 높다. 그리고 기준 입출력 크기가 큰 경우에는 병렬 입출력, 특히 병렬 읽기 요구들이 많이 발생되는데 RAID5의 데이터 저장 구조의 특성상 부하 중에서 병렬 읽기가 많고, 이들 병렬 읽기들이 동시에 접근하는 디스크들의 수가 많을수록 부하 균형을 이루기 쉽다. 그러나 기준 입출력 크기가 분할 단위에 접근하는 경우에는 개별 입출력 요구들과 병렬 입출력 요구들이 혼합되므로 합성 부하 발생기의 지역성 정도에 따라서 부하가 일부 디스크에 집중될 수 있다.

시뮬레이션 결과, 입출력 요구의 기준 크기가 분할 단위보다 작거나 분할 단위를 초과할 경우에는 부하가 집중되는 디스크에서의 대기 지연으로 인하여 응답 성능이 악화될 수 있지만 부하 균형을 유지하는 기법 때문에 비교적 일정한 성능을 유지한다. 그러나 부하 균형 반출 기법은 부하가 매우 커지면 입출력 요구의 크기가 분할 단위의 크기와 같은 경우, 응답 성능이 저하된다 즉, 작은 데이터 입출력 요구에 의한 부하가 매우 크게 증가되면 부하 균형 반출 기법은 부분적으로 부하가 집중되는 디스크로 인하여 지역성이 있는 반출 요구들이 반출된 후 다시 쓰기 캐쉬에 재저장되었다가 또다시 반출됨으로 인하여 상대적으로 쓰기 요구들의 비율이 증대된 효과를 나타내게 되고, 결국에는 (그림 5(c))와 유사한 결과를 도출하게 된다. 그러나 참조-부하균형 반출 기법에서는 재저장 확률을 줄이므로 쓰기 요구들이 입출력 요구에서 차지하는 비율이 상대

적으로 증가되는 효과를 제거할 수 있다. 따라서 참조-부하균형 반출 기법에서는 데이터의 크기가 분할 단위의 크기에 근접할 경우, 부하의 크기가 매우 크게 증가하더라도 응답 성능은 부하 균형 반출 기법에 비하여 상대적으로 일정하게 유지할 수 있음을 나타내고 있다

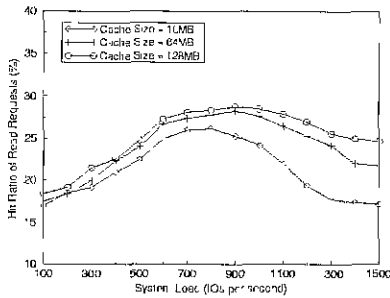
(그림 4)와 (그림 6)의 결과들은 디스크간 부하 균형을 유지하는 기법의 성능이 캐쉬의 성능과 직접적인 관련이 있음을 나타내고 있다. (그림 7)은 디스크캐쉬 중 쓰기 캐쉬의 크기를 변화시켰을 때 쓰기 캐쉬에서의 읽기 적중률을 부하의 증가에 따라서 나타낸 것이다. 부하 균형 반출 기법에서는 부하가 증가됨에 따라서 캐쉬 적중률이 증가하지만 부하가 매우 큰 경우에는 캐쉬 적중률이 오히려 저하된다. 이는 부하가 증가함에 따라서 쓰기 캐쉬에서 강제적인 반출이 일어나기 때문이기도 하지만, 3장에서 설명한 것처럼, 일부 디스크에 부하가 집중될 경우, 상대적으로 부하가 작은 디스크에 해당되는, 지역성을 가지는 쓰기 요구들이 반출될 확률이 실제로 증가되기 때문이다. 반면에 참조-부하균형 반출 기법의 경우에는 이러한 경우가 가능한 한 발

생되지 않도록 쓰기 요구들을 반출하기 때문에 부하가 매우 큰 경우에도 부하 균형 반출 기법에 비하여 상대적으로 캐쉬 적중률을 높게 유지할 수 있음을 보여 주고 있다

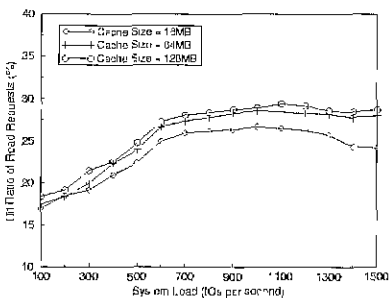
이러한 캐쉬 적중률의 차이는 곧 입출력 응답 성능, 특히 읽기 요구에 대한 응답 성능에 큰 영향을 미친다. 참조-부하균형 반출 기법에 의하여 캐쉬에 저장된 쓰기 요구들에 대한 캐쉬 적중률이 높아지면 결국 디스크로의 물리적인 입출력 요구의 수가 줄어들게 된다. 이는 디스크에서의 부하를 줄일 수 있지만 지역성을 가지는 쓰기 요구들을 유지하기 위하여 부하 균형 반출 기법에 비해서는 상대적으로 디스크간 부하 균형을 고르게 유지하기는 어렵다. 그러나 시뮬레이션의 결과는 부하가 지역성을 가지는 경우, 지역성에 의한 이득을 유지하면서 부하 균형을 이루는 것이 RAID5의 전체적인 성능을 향상시킬 수 있음을 나타내고 있다.

### 6. 결 론

최근의 대형 메인 컴퓨터 시스템들은 많은 수의 고속의 프로세서들을 사용함으로써 단일 프로세서를 사용하는 컴퓨터 시스템들에 비하여 디스크 저장 장치의 부하가 상대적으로 볼 뿐만 아니라, 작은 크기의 데이터들에 대한 입출력 부하가 매우 높은 경우, 디스크 배열을 이루는 디스크들간의 부하 분포도 균일하지 않은 경향이 높다. 이러한 경향은 병렬 읽기 요구에 대한 처리 시간을 지연시키므로 이를 줄이기 위하여 부하 균형 반출 기법이 적용된 RAID5 제어기를 필요로 하게 되었다 그러나 부하 균형 반출 기법은 부하가 매우 크고 작은 데이터에 대한 입출력 요구의 증가로 인하여 일부 디스크로 부하 집중될 때, 디스크캐쉬에 저장된 반출 요구들에 대한 참조 지역성을 무시한 캐 쓰기 요구들을 반출하므로 디스크캐쉬의 효율을 저하시켜서 결과적으로 병렬 디스크 제어기의 입출력 성능을 저하시키게 된다 그러나 부하 균형 반출 기법에서 각 디스크들의 참조 정보를 유지하고, 이를 바탕으로 쓰기 요구들에 대한 지역성을 사용하는 참조-부하균형 반출 기법을 적용한 병렬 디스크 제어기를 RAID5 제어기에 적용하면 디스크간 부하 균형을 유지하면서 디스크캐쉬의 지역성을 효과적으로 활용할 수 있도록 반출 요구들을 반출할 수 있다. 이를 위하여 본 논문에서는 참조-부하균형 반출 기법을 사용하는 RAID5 제어기



(a) Load Balancing Destage



(b) Reference-Load Balancing Destage

(그림 7) 디스크캐쉬의 크기에 따른 캐쉬 적중률의 변화

모형을 구성하고 시뮬레이션을 통하여 이전의 반출 기법들과의 상대적인 입출력 성능을 비교, 평가하였다. 성능 평가 결과, 참조-부하균형 반출 기법을 사용하는 RAID5 제어기는 지역성을 이용할 수 있는 반출 요구들이 이전의 부하 균형 기법에 의하여 강제로 반출되어 쓰기 요구들이 디스크캐쉬에서 반출-재저장되는 동작이 반복되는 확률을 효과적으로 줄임으로써, 작은 크기의 데이터에 대한 부하가 매우 높은 경우에도 디스크캐쉬의 성능 저하를 방지하여 RAID5의 입출력 성능을 향상시킬 수 있었다.

### 참 고 문 헌

- [1] P. Biswas, K. Ramakrishnan and D. Towsley, "Trace-Driven Analysis of Caching Policies for Disks," *Proceedings of the 1993 ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pp.13-23, 1993.
- [2] Y. S. Chang and C. S. Kim, "The Design and Performance Evaluation of the RAID5 Controller Using the Load-balanced Destage Algorithm," *Proceedings of the International Conference of the Parallel and Distributed Systems*, pp.28-34, 1997.
- [3] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz and D. A. Patterson, "RAID : High Performance, Reliable Secondary Storage," *ACM Computing Surveys*, Vol.26, No. 2, pp.145-188, 1994.
- [4] P. M. Chen, "Striping in a RAID Level 5 Disk Array," *Proceedings of the 1995 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pp.136-145, 1995.
- [5] B. Y. Kim and Y. S. Chang, "Improved RAID5 controller using the load-balanced destage algorithm," *Electronics Letters*, Vol.34 No.3, pp.248-250, 1998.
- [6] J. Menon and D. Mattson, "Performance of Disk Arrays in Transaction Processing Environments," *Proceedings of the 12th International Conference on Distributed Computing Systems*, pp.302-309, 1992.
- [7] J. Menon and J. Courtney, "The Architecture of a Fault-Tolerant Cached RAID Controller," *Proceedings of the 20th Annual International Symposium on Computer Architecture*, pp.76-86, 1993.
- [8] J. Menon, "Performance of RAID 5 Disk Arrays with Read and Write Caching," *Distributed and Parallel Databases*, Vol.17, No.1-2, pp.129-139, 1993.
- [9] D. A. Patterson, P. M. Chen, G. A. Gibson and R. H. Katz, "Introduction to Redundant Arrays of Inexpensive Disks (RAID)," *IEEE CompCon '89*, pp.112-117, 1989.
- [10] C. Ruemmler and J. Wilkes, "UNIX Disk Access Patterns," *Proceedings of the Winter 1993 USENIX Conference*, pp.405-420, 1993.
- [11] C. Ruemmler and J. Wilkes, "An Introduction to Disk Drive Modeling," *IEEE Computer*, Vol.27, No. 3, pp.17-28, 1994.
- [12] D. Stodolsky, G. Gibson and M. Holland, "Parity Logging: Overcoming the Small Problem in Redundant Disk Arrays," *Proceedings of the 20th Annual International Symposium on Computer Architecture*, pp.64-75, 1993.
- [13] A. Varma and Q. Jacobson, "Destage Algorithms for Disk Arrays with Non-Volatile Caches," *Proceedings of the 22th Annual International Symposium on Computer Architecture*, pp.83-95, 1995.
- [14] J. Wilkes, R. Golding, C. Staelin, and T. Sullivan, "The HP AutoRAID hierarchical storage system," *15th Symposium on Operating System Principles*, 1995.
- [15] B. R. Worthington, G. R. Ganger, and Y. N. Pakt, "Scheduling Algorithms for the Modern Disk Drives," *Proceedings of the 1994 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pp.241-251, 1994.
- [16] 장윤식, 김증상, "RAID5의 성능 향상을 위한 부하 균등 destage 기법", *정보과학회논문지(A) 제25권 1호*, pp.83-93, 1998.



장 윤 석

e-mail : cosmos@sparc.snu.ac.kr  
1988년 서울대학교 물리학과 졸업  
(이학사)  
1990년 서울대학교 대학원 컴퓨터  
공학과 졸업(공학석사)  
1998년 서울대학교 대학원 컴퓨터  
공학과 졸업(공학박사)

1994년~1998년 대진대학교 컴퓨터공학과 전임강사  
1998년~현재 대진대학교 컴퓨터공학과 조교수  
2000년~현재 Visiting Scholar in Dept. of EE-Systems,  
University of Southern California  
관심분야 : 컴퓨터구조, 클러스터 컴퓨터, RAID 등



김 보 연

e-mail kumby@nms.kyunghee.ac.kr  
1989년 이화여자대학교 전자계산  
학과 졸업(이학사)  
1991년 서울대학교 대학원 컴퓨터  
공학과 졸업(공학석사)  
1998년 서울대학교 대학원 혈동  
과정 의용생체공학전공  
졸업(공학박사)

1998년~1999년 서울대학교 의학연구원 선임연구원  
1999년~현재 경희대학교 동서의학대학원 연구교수  
관심분야 : 알고리즘, 인공지능, 의료경보시스템 등