論文94-31B-11-2

기록전용버퍼를 내장한 디스크배열 시스템의 성능분석

(Performance Analysis of Disk Array System with Write Dedicated Buffer)

尹 齊 鉉*, 全 昌 浩**

(Je-Hyun Yoon and Chang Ho Jeon)

要 約

중앙처리장치와 디스크장치 사이의 큰 속도 차이로 인해 발생하는 병목현상은 디스크 입출력을 수반하는 작업의 처리시간에 많은 영향을 미친다. 본 논문에서는 디스크배열 시스템의 읽기요청에 대한 응답시간을 개선하기 위하여 기록전용버퍼를 내장할 것을 제안하고 그것의 스케줄링정책과 개선효과를 정량적으로 분석한다. 시뮬레이션으로 비교분석함으로써 세 가지 스케줄링 정책 가운데 부분스트라입결합 정책이 읽기와 기록응답시간 면에서 가장 효율적이며, 특히 분산형배열 시스템에 기록전용버퍼를 내장할 경우 요청율이 높을수록 읽기요청에 대한 응답시간의 개선효과가 크다는 것을 보인다.

Abstract

Turnaround time of a job performing frequent disk I/O operations is greatly affected by I/O bottleneck which incurs due to the large gap in the speeds of I/O devices and the CPU. This paper proposes to employ a Write Dedicated Buffer(WDB) in disk arrays to improve the response time for read requests and analyzes the scheduling policies and the efficiency of the WDB. Through a series of simulations we show that, among the three policies examined, the partial stripe join(PSJ) policy is the most effective in terms of response time for both read and write requests and that, especially when using a WDB on declustered arrays, improvement of response time for read requests becomes greater as the request rate increases.

*正會員, 韓國 디지탈 콘트롤
(Korea Digital Control)
**正會員, 漢陽大學校 電子計算學科
(Dept. of Computer Science and Eng., Hanyang Univ.)
接受日字 : 1994年 1月 20日

I. 서 론

지난 수년간 VLSI 기술과 병렬처리 기술의 발달에 힘입어 중앙처리장치(CPU)와 주기억장치의 속도는 크게 향상되었으나 디스크장치와 같은 입출력장치의 경우는 기계적인 동작 특성때문에 성능향상에 한계가 있어 중앙처리장치와 입출력장치와의 속도불균형이

심화되었다.[1, 2] 또한 중앙처리장치의 처리속도가 빨라짐에 따라서 더 짧은 시간간격으로 입출력을 요청하게 되므로 입출력시스템의 느린 응답은 컴퓨터시스템의 병목요인이 되고 있다. 최근 이러한 입출력 병목현상을 해결하기 위한 연구가 활발히 진행되고 있으며, 특히, 입출력장치 중 컴퓨터시스템의 동작 중에 가장 빈번하게 사용되는 디스크 입출력장치의 성능개선을 위하여 많은 노력이 집중되고 있다.

디스크장치는 입출력 요청을 처리하기 위해 헤드를 움직이고, 원하는 섹터를 헤드 아래에 위치시키는 등의 기계적인 동작을 수반하기 때문에 디스크 측에서의 획기적인 속도 향상을 기대하기는 어렵다. 그래서 디스크장치 자체의 성능향상을 추구하기 보다는 여러 개의 디스크를 병렬로 연결하여 동시에 구동시키는 디스크배열을 많이 사용하고 있다.[3, 4] 디스크배열은 값싼 소형의 디스크를 여러 개 연결하여 데이타를 각 디스크에 교호배치(interleaving)시키고 입출력 요청 시에 각 디스크에서 동시에 데이타를 전송함으로써 데이타 전송률을 향상시킨 장치이다.[2, 5]

디스크 장치를 상대로 한 입출력작업은 읽기요청(read request)과 기록요청(write request)으로 나누어 생각할 수 있는데 그 중에서도 읽기요청의 응답시간은 그 요청을 생성시킨 작업의 반환시간(turnaround time)에 큰 비중을 차지한다. 예를 들어 디스크에 저장되어 있는 어떤 화일을 갱신하는 작업이 있다고 하자. 이 작업은 갱신할 데이타에 대해 먼저 읽기요청을 하고 데이타가 주기억장치로 들어온 다음 원하는 갱신을 한다. 그리고 그 결과에 대해 다시 기록요청을 함으로써 수행을 끝내게 된다. 따라서 작업이 읽기요청을 하고나서 그 데이타가 준비되기까지의 시간 즉, 읽기요청에 대한 응답시간은 디스크 입출력을 수반하는 작업의 반환시간을 결정하는 중요한 요소가 된다.

읽기요청의 응답시간은 디스크장치의 동작원리 상 피할 수 없는 탐색시간(seek time)과 회전지연(rotation latency) 이외에 또 다른 요인으로 길어질 수 있다. 즉, 동일한 디스크에 기록요청이 먼저 있었다면 그 기록요청은 다 처리할 때까지 읽기요청을 처리할 수 없기 때문이다. 이렇게 읽기요청에 대한 응답이 그보다 앞서 요청된 기록작업 때문에 지연되는 문제는 경우에 따라 시스템의 기능을 상실케 할 수도 있는 것이다. 예를 들면 실시간 응용분야에서 시간제약을 가진 실시간성 읽기요청이 비실시간성 기록요청 때문에 지연되어 실기되는 경우이다.

본 논문에서는 디스크배열 시스템의 읽기요청에 대한 응답시간을 개선시키기 위해 기록전용버퍼(Write Dedicated Buffer, WDB)를 내장할 것과 그것을 효율적으로 스케쥴링하는 방식을 제안하고, 그에 따라 실제 응답시간이 개선되는 정도를 정량적으로 분석한다. 기록전용버퍼는 디스크 제어장치 내에 위치하여 완료되지 않은 기록작업때문에 후속되는 읽기요청의 처리가 지연되는 것을 피하기 위하여 기록될 데이타를 임시 저장하는 기억장치이다. 일단 기록전용버퍼에 저장된 데이타는 차후에 버퍼 내의 여유공간이 부족해지거나 후속되는 읽기요청이 없을 동안에 기록된다. 이러한 기록전용버퍼의 유효성은 단일 디스크 상에서 입증된 바 있으며[6] 이 외에 하나의 요청을 처리하는데 액세스되는 디스크의 수, 동시에 동기되어 움직이는 디스크의 수, 전체 디스크의 수의 관점에서 서로 다른 구성방식 별로 디스크배열 시스템의 성능을 분석한 연구[7]와, 읽기/쓰기 요청의 크기, 읽기/쓰기 요청의 비율에 따라 RAID 시스템의 성능이 어떻게 달라지는지를 실험적으로 분석한 연구[8]가 보고되어 있다. 그러나 디스크배열 시스템에서 여러 디스크 사이에 교호배치된 데이타의 위치를 고려한 버퍼의 운용방식이나 그에 따른 버퍼의 효율을 분석한 연구결과는 없다.

본 논문의 제 2장에서는 기록전용버퍼를 도입한 디스크시스템의 구성과 기록전용버퍼의 유용성을 간단히 예시한다. 제 3장에서는 기록전용버퍼를 효율적으로 운용하기 위한 스케쥴링 정책과 동작원리를 소개한다. 그리고 제 4장에서는 세 가지의 스케쥴링 정책들을 상호 비교하고 기록전용버퍼의 효율성을 보이기 위한 시뮬레이션 방법과 그에 근거한 분석결과를 제시하고, 제 5장에서 결론을 맺는다.

Ⅱ. 기록전용버퍼를 내장한 디스크시스템

기록전용버퍼를 내장한 디스크배열 시스템의 구성도는 그림 1에 나타나 있다.

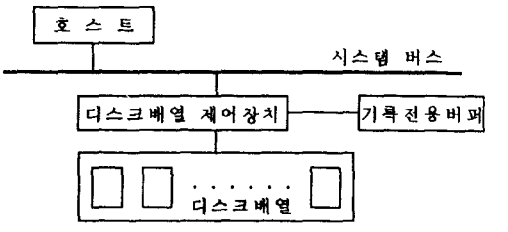


그림 1. 기록전용버퍼를 가진 디스크배열 시스템 구성

Fig. 1. A Disk Array System with Write Dedicated Buffer.

그림 1에서 보인 것처럼 기록전용버퍼는 디스크배열 제어장치에 의해 관리되는 RAM으로서 호스트로부터 요청된 디스크 입출력작업 가운데 기록요청을 임시수용하는 곳이다. 모든 기록요청을 저장하는 것이 아니라 버퍼내의 여유공간이 허락하는 한도내에서 후속되는 읽기요청이 대기상태로 머무르게 되는 경우에 저장된다. 그렇게 함으로써 그 후속되는 읽기요청을 보다 신속히 처리하게 하는 것이다.

여기서 기록전용버퍼가 있음으로 해서 읽기요청의 응답시간이 어떻게 개선되는지 예를 들면 다음과 같다. 현재의 논리시각을 100이라고 하고, 이 이후에 입출력 요청들이 write_1〈110, 40〉, read_2〈125, 30〉, write_2〈137, 40〉, read_3〈160, 40〉, … 의 순서로 들어온다고 가정하자. 여기서 write_1〈110, 40〉은 도착시각이 110인 기록요청을 말하고, 그 요청을 처리하는 데 소요되는 시간이 40이라는 것을 의미한다. 이 경우에 기록될 데이타를 기록전용버퍼에 저장함으로써 읽기요청의 응답시간을 단축시키는 효과를 그림 2에 보이고 있다.

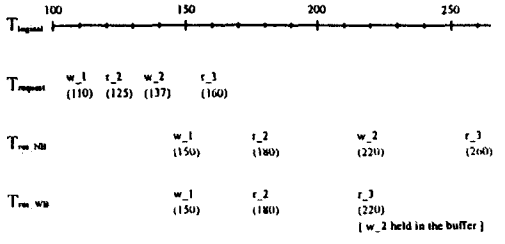


그림 2. 기록전용버퍼의 유무에 따른 응답시간 차이
Fig. 2. The Difference in Response Time between Systems with and without Write Dedicated Buffer.

그림 2에서 $T_{logical}$, $T_{request}$, $T_{res\ NB}$, 그리고 $T_{res\ WB}$는 각각 논리시간축, 요청시각, 기록전용버퍼가 없는 시스템의 응답시각, 그리고 기록전용버퍼를 내장한 시스템의 응답시각을 의미한다.

기록전용버퍼를 사용함으로써 읽기요청의 응답시간이 개선되는 효과는 read_3의 경우에서 볼 수 있다. 기록전용버퍼가 없는 시스템에서는 각 요청의 도착순서에 따라 순차적인 처리과정을 거쳐 읽기요청 read_3이 논리시각 260에서 처리완료되므로 그 응답시간이 100이 된다. 반면에 기록전용버퍼가 있는 시스템에서는 read_2가 끝나는 시각 180에서 write_2보다 먼저 read_3의 처리가 시작되므로 논리시각 220에서 완료되어 응답시간이 60으로 단축된다.

여기서 기록전용버퍼에 수용됨으로써 디스크에까지 기록되지 못한 데이타블록에 대하여 읽기요청이 후속되는 경우도 생각해 볼 수 있다. 이 경우에는 디스크 제어장치가 버퍼에 저장된 내용으로 그 읽기요청에 응답을 해준다. 단일 디스크시스템에서 기록전용버퍼의 유용성을 밝힌 연구[6]에서도 읽기요청에 대해 버퍼에 저장되어있는 내용으로 응답하지 못할 경우에 한하여 디스크처리기로 요청을 전달하는 모델을 사용하고 있다.

Ⅲ. 기록전용버퍼의 스케쥴링

기록전용버퍼를 스케쥴한다는 것은 버퍼에 수용된 기록요청들을 처리할 순서를 정하는 것이다. 이것은 통상의 디스크 스케쥴링과는 다른 문제이다. 디스크 스케쥴링의 결과로 각 디스크로 보내지는 요청들 중에서 기록요청들이 기록전용버퍼에 저장되고, 이렇게 수용된 기록요청들을 처리할 시점에 버퍼를 스케쥴하는 것을 말한다.

기록전용버퍼에 수용된 기록요청들은 버퍼 내의 여유공간이 정해진 임계치보다 작아지거나 또는 대기중인 읽기요청이 없을 때 처리된다. 여기서 만약 기록전용버퍼에 기록요청들이 남아있는 상태에서 시스템의 오류가 발생했을 때의 디스크 데이타의 완전성이나 신뢰도 문제를 생각한다면 버퍼의 용량한계나 읽기요청의 대기유무에 근거하기보다는 가능한 한 짧은 시간간격을 두고 주기적으로 버퍼 내의 데이타를 해당 디스크에 기록해 나갈 수도 있을 것이다. 그러나 시스템의 오류발생 가능성은 짧은 주기 내에서도 존재하는 만큼 별도의 결함허용장치를 구비하지 않는 한 완전한 신뢰도를 보장할 수는 없다. 그런 의미에서 기록전용버퍼를 거치게 되는 데이타의 신뢰도 문제를 다루는 것은 기록전용버퍼의 유효성을 분석하는 것과 별개의 문제라고 본다. 따라서 본 논문에서는 버퍼 내의 요청들을 처리한다면 어떤 순서로 처리하느냐에 관심을 두고 기록전용버퍼를 가장 효율적으로 스케쥴하는 방법을 찾기위하여 다음의 세 가지 스케쥴링 정책을 비교검토한다. 여기서 효율적이라 함은 기록요청과 읽기요청에 대한 평균응답시간이 가장 작게되는 것을 말한다.

(1) 선착선처리(First-Come First-Served)

선착선처리 정책은 기록전용버퍼 내의 기록요청들을 도착 순서대로 디스크로 보냄으로써 도착시간이 빠른 순서로 처리한다. 개념이 단순하여 구현하기가 가장 간단한 정책이라는 점때문에 비교대상으로 택한다.

(2) 최단대기열 우선(Shortest Queue First)

이 정책은 전체 디스크 중에서 읽기요청의 부하가

가장 작은 디스크에 대한 기록요청들을 다른 디스크에 대한 기록요청보다 먼저 처리한다. 물론 이 때 같은 디스크에 대한 읽기요청이 있다 하더라도 기록전용버퍼에 여유공간이 없을 때이므로 먼저 기록요청을 처리하게 한다. 이것은 디스크배열 시스템 전체 차원에서 지연되는 읽기요청의 수를 최소화할 수 있고, 따라서 평균 읽기응답시간을 줄일 수 있는 정책이다.

(3) 부분스트라입결합(Partial Stripe Join)

부분스트라입결합 정책은 가능한 한 배열을 구성하는 모든 디스크에서 병렬로 처리되도록 두 개 이상의 요청을 합쳐서 기록처리한다. 하나의 기록요청은 단일 블록에 대한 요청일 수 도 있고 여러 블럭에 대한 요청일 수도 있다. 여러 블럭에 대한 단일 요청이 들어오면 각 블럭들이 저장되어 있는 해당 디스크로 요청이 분산된다. 그림 3에 8개의 디스크로 구성되는 디스크배열 시스템에서 여러 디스크에 분산저장된 데이타 블럭들에 대하여 요청들이 들어온 예를 보이고 있다. 이 예를 통하여 부분스트라입결합 정책의 동작을 설명하면 다음과 같다.

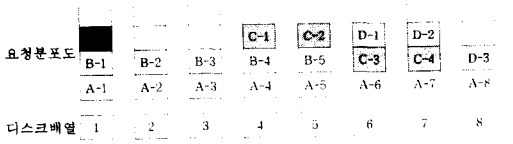


그림 3. 전체스트라입 및 부분스트라입 요청의 분포예

Fig. 3. An Example of Enqueued Full Stripe Resquests and Partial Stripe Requests.

그림 3에 보인 것은 임의의 시간간격을 두고 생성된 기록요청 A, B, C, D, E를 해당 디스크와 짝지워 보인 예이다. 여기서 여러 디스크에 걸친 하나의 요청을 스트라입(stripe)라고 한다. 요청 A와 같이 전체 디스크를 모두(A-1에서 A-8까지) 접근해야 하는 요청을 전체스트라입(full stripe) 요청이라고 하며, 요청 B, C, D, E와 같이 하나 또는 몇 개의 디스크만 접근해도 되는 것은 부분스트라입(partial stripe) 요청이라 한다.

부분스트라입결합 정책에서는 요청 A와 같은 전체스트라입 요청인 경우에는 A에 대한 처리만 해주고, B, C, D, E와 같은 부분스트라입 요청들에 대해서는 서로 다른 디스크를 접근하게 되는 요청들을 적절히 결합하여 동시에 처리되도록 스케쥴한다. 예를 들어 그림 3에서 요청 B를 처리하게 될 때에는 요청 B의 처리와 관계없는 디스크 6, 7, 8에 대한 다른 요청들도 동시에 처리되도록 한다. 여기서 가능한 대상은 요청 C의 일부(C-3과 C-4)와 요청 D의 일부(D-3)이다. 그러나 디스크 6, 7에 대한 요청 C-3와 C-4는 디스크 4, 5에 대한 요청 C-1 및 C-2와 더불어 하나의 스트라입을 이루고 있으므로 그것들과 함께 처리되지 않는 한 C-3와 C-4가 먼저 처리된다 하더라도 요청 C는 전체적으로 처리가 완료되지 않은 상태로 남게된다. 그러므로 요청 C의 일부를 선택하기보다는 요청 B와 함께 처리가 완료될 수 있는 요청 D를 먼저 스케쥴하게 된다. 본 논문에서는 비동기식 배열 시스템을 대상으로 하고 있으나 이 정책은 블럭단위의 동기화를 시도하는 것이다.

이상의 세 가지 스케쥴링 정책들의 효율을 상호비교하기 위한 시뮬레이션 방법과 그 결과는 다음 절에서 기록전용버퍼의 유효성을 분석한 결과와 함께 제시한다.

Ⅳ. 시뮬레이션 및 결과 분석

1. 시뮬레이션 모델 및 방법

본 연구에서는 디스크배열 시스템에서 기록전용버퍼를 가장 효율적으로 사용할 수 있는 스케쥴링 정책을 구하고, 또 기록전용버퍼가 있음으로해서 읽기응답시간이 얼마나 개선되는지를 분석하기 위하여 디스크배열 시스템을 C언어로 모델링하고 디스크작업 부하를 변화시켜 가면서 모의실험을 하였다.

비교대상이 된 스케쥴링 정책은 앞 절에서 설명한 세 가지이며, 기록전용버퍼의 유무를 가정하여 모델링된 디스크배열 시스템은 단순배열(traditional) 시스템과 분산형배열(declustered) 시스템 두 가지이다. 여기서 말하는 단순배열 시스템은 데이타를 저장할 때 화일의 크기에 관계없이 화일단위로 디스크에 배치하는 시스템을 말하며, 분산형배열 시스템은 데이타를 블럭단위로 배치하는 시스템으로서 하나의 화일이 블럭단위로 나뉘어져서 여러 디스크에 교호배치되는 것을 말한다. 분산형배열 시스템은 여러 블럭으로 분산배치된 대단위 화일에 대한 요청이 있을 경우에 전송속도가 높으므로 전송될 데이타 양이 많고 요청율이 낮은 작업부하에 효과적이며, 전자의 단순배열 시스템은 그와 반대로 전송될 데이타의 양이 적고 요청율이 높은 경우에 더 좋은 성능을 보인다.[7] 다른 말로 하면 분산형배열 시스템은 요청율이 높을 때 응답이 느리다는 단점을 가진 것이다. 본 논문에서는 기록전용버퍼를 사용함으로써 분산형배열 시스템이 가지는 이 단점을 개선할 수 있다는 것을 보이는데

초점을 두고 있다.

모델링 한 디스크배열 시스템은 N개의 디스크와 그 각각에 대한 쓰기요청을 저장하는 N개의 큐(앞의 그림 1에서 보였듯이 기록전용버퍼는 디스크배열 시스템 내에 하나 뿐이지만, 시뮬레이션 방법 상 여기서 N개의 큐로 분할된 모델을 사용한다.), 그리고 호스트가 보내는 논리적(logical) 요청을 각 디스크에 물리적(phsical) 요청으로 분할하여 처리를 요구하는 지명기(dispatcher)로 구성된다. 단순배열 시스템에서는 논리적 요청의 수와 물리적 요청의 수가 동일하지만 분산형배열 시스템에서는 데이타가 블럭 단위로 여러 디스크에 교호배치되므로 하나의 논리적 기록요청이라 할지라도 M블럭의 데이타에 대한 처리가 요구되는 경우 M개의 물리적 기록요청으로 분할되어 해당되는 디스크 큐에 들어가게 된다.

디스크시스템의 성능지표는 읽기요청 또는 기록요청에 대한 응답시간으로 하여 다음과 같이 정의하였다.

$$\text{응답시간}(R_t) = \text{요청 오버헤드}(R_o) + \text{탐색시간}(S_t) + \text{회전지연}(R_d) + \text{전송시간}(T_t) + \text{대기지연}(Q_d)$$

여기서 요청 오버헤드란 논리적 요청을 물리적 요청으로 변환하는데 소요되는 시간으로 전송될 데이타 양에 따라 증가한다. 실험을 통하여 요청 오버헤드는 한 개의 블럭을 처리하는데 1.3ms 정도 소요된다고 알려졌으며[8], 탐색시간과 회전지연시간은 다음과 같이 얻어진다고 보고되어 있다.[6]

$$\text{탐색시간}(S_t) = 0.6 \times \sqrt{n},$$

$$\text{회전지연}(R_d) = \frac{k}{k+1} \times T_r.$$

여기서 n은 이동할 트랙의 수이며, k는 하나의 요청을 처리하기 위해 함께 동작하는 디스크의 수이고 T_r은 디스크의 한 회전 시간이다.

이제 기록전용버퍼를 사용한다고 하면 기록요청시 버퍼에 저장되었다가 읽혀져 나오는 시간이 응답시간에 포함되어야 한다. 그 시간을 B_t라고 하면

$$Bt = Block_Size / Data_Transfer_Rate \times 2$$

로 계산할 수 있다. 블럭크기가 4KB이고 데이터전송율이 4.8MB라고 한 본 연구에서는 평균적으로 블럭당 1.6ms 씩 소요된다고 보았다.

이제까지 정의된 수식을 이용하여 읽기응답시간과 기록응답시간을 정리해 보면 아래와 같다. 여기서 어떤 읽기요청이 기록전용버퍼로부터 직접 응답되는 경우는 확률적으로 무시할 만큼 적다고 보아 고려하지 않았다.

$$\text{읽기응답시간}(R_{read}) = R_o + S_t + R_d + T_t + Q_d$$

$$\text{기록응답시간}(R_{write}) = R_o + S_t + R_d + T_t + Q_d + B_t$$

이상과 같이 정의된 입출력 요청에 대한 응답시간을 계산하는데 표 1과 표 2에 보인 디스크자체 및 시뮬레이션에 관련된 파라메터 값을 적용하였다.

표 1. 디스크 파라메터
Table 1. Disk Parameters.

디스크 파라메터	수 치
1회전 시간	16.6 ms
트랙간 헤드 이동시간	0.6 ms
디스크당 트랙 수	1000
트랙당 섹터 수	10
섹터당 바이트 수	4K 바이트
데이터 전송율	4.8M 바이트

표 2. 시뮬레이션 파라메터
Table 2. Simulation Parameters.

시뮬레이션 파라메터	수 치
디스크 수	16
요청 처리 수	10,000
읽기요청 비율	70 %
요청 도착간 시간	10~70 ms
기록전용버퍼 크기	400K 바이트
최대 요청크기	64K 바이트
블럭크기	4K 바이트

표 1에 보인 디스크 파라메터 값은 디스크의 성능에 관한 다른 연구에서 사용된 디스크 특성과 동일한 것이다.[4, 6, 8] 우리는 최대 요청크기를 64K 바이트로 제한하였는데 이것은 화일 트랜잭션의 응용분야에서 대개 적은 양의 데이터를 처리하며, 디스크 성능을 분석한 다른 문헌에서도 최대 요청크기를 제한하고 있기 때문이다.[8] 물론 실제 응용분야에서는 드물게 수 백KB까지의 크기를 가지는 요청도 있을 수 있으나 요청크기가 작고 요청율이 높은 경우에 읽기응답의 개선을 추구하는 본 연구에서는 최대 요청크기를 넘는 기록요청이 들어온다면 예외적으로 먼저 처리해 준다고 가정하고 이러한 경우가 기록전용버퍼의 사용효과에 미칠 수 있는 영향까지는 고려하지 않았다. 또한 기록전용버퍼의 여유공간의 임계치를 최대 요청크기인 64K 바이트로 하였는데 이것은 하나의 요청이 최대 크기로 들어온다 하더라도 기록전용버퍼에 저장이 가능하도록 한 것이다.

읽기요청과 기록요청으로 분류되는 디스크 입출력 작업은 그 70%가 읽기요청이라고 분석된 바 있어[9] 읽기요청과 기록요청의 비율을 7:3으로 고정시켰다. 그리고 입출력 요청의 평균도착간 시간을 10ms에서 70ms까지 변화시키면서 지수분포를 갖도록 하였으며, 요청크기는 난수발생기를 사용하여 4K 바이트에서 64K 바이트 사이의 범위에서 임의로 변화시켰다. 또한 처리될 데이타의 트랙위치도 난수발생기를 사용하여 전체 디스크에 임의분포시켰다.

2. 결과 및 분석

1) 스케줄링정책의 성능비교

분산형배열 시스템을 대상으로 하여 세 가지 스케줄링 정책에 대해 앞 절에서 설명한 방법에 따라 평균 읽기응답시간과 기록응답시간을 계산하고, 그 결과를 각각 그림 4와 그림 5에 그래프로 나타냈다.

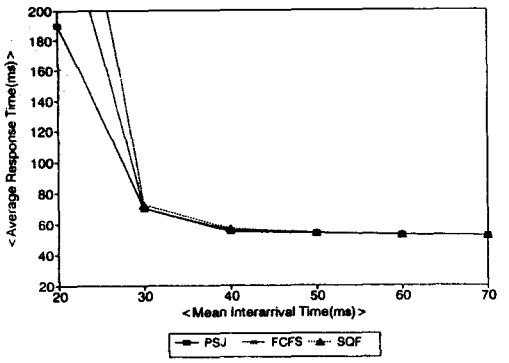


그림 4. 스케줄링 정책들의 읽기응답시간 비교

Fig. 4. Comparison of Read Response Time for Different Scheduling Policies.

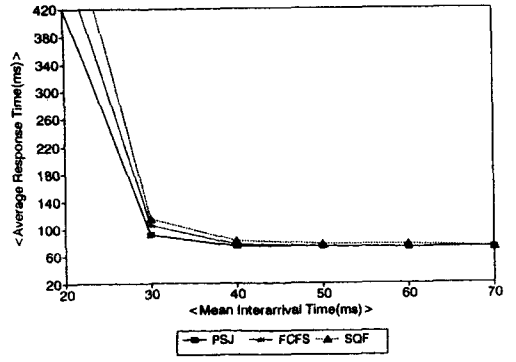


그림 5. 스케줄링 정책들의 기록응답시간 비교

Fig. 5. Comparison of Write Response Time for Different Scheduling Policies.

2) 기록전용버퍼의 유무에 따른 성능 비교

그림 6은 두 가지 유형의 디스크배열 시스템에서 기록전용버퍼를 사용한 경우와 사용하지 않은 경우에 대한 읽기응답시간을 보여주고 있으며, 그림 7은 같은 경우에서의 기록응답시간을 보여주고 있다. 이 두 그림에서 TRD와 DCL은 각각 단순배열 시스템과 분산형배열 시스템을 나타내며, 그 앞에 붙은 W_의 유무는 기록전용버퍼가 사용된 경우와 사용되지 않은 경우를 각각 의미한다.

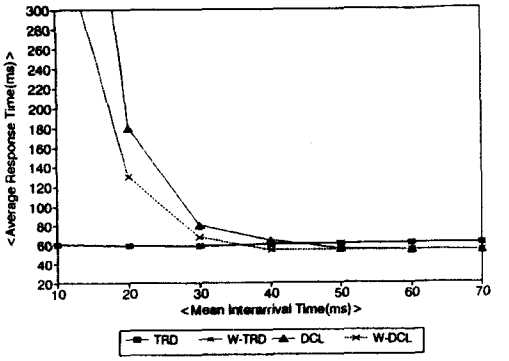


그림 6. 기록전용버퍼의 유무에 따른 읽기응답시간 비교

Fig. 6. Comparison of Read Response Time for Systems with and without Write Dedicated Buffer.

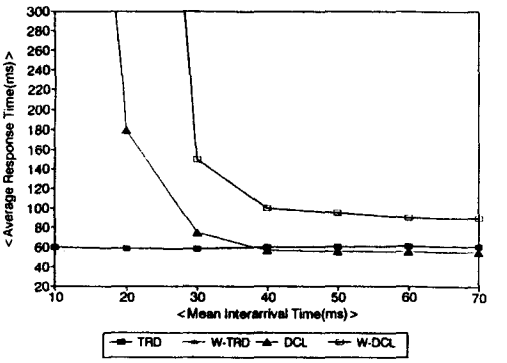


그림 7. 기록전용버퍼의 유무에 따른 기록응답시간 비교

Fig. 7. Comparison of Write Response Time for Systems with and without Write Dedicated Buffer.

두 그림에서 볼 때 단순배열 시스템의 경우에는 기록전용버퍼가 응답시간에 별다른 영향을 미치지 않음

을 알 수 있다. 이것은 단순배열 시스템에서 하나의 화일이 여러 디스크에 교호배치되는 것이 아니라 동일한 디스크에 저장되므로 하나의 요청을 처리하는데 액세스되는 디스크가 하나 뿐이며, 그러므로 읽기요청과 기록요청이 동일한 디스크에 대해 발생되지 않는 한 읽기요청과 기록요청은 서로의 처리시간에 거의 영향을 미치지 않기 때문이다. 실제 시뮬레이션에서는 난수발생기를 사용하여 읽기요청과 기록요청을 임의의 디스크에 보냈으므로 동일한 디스크에 읽기와 기록이 동시에 요청되는 경우도 포함되었다고 보지만 기록전용버퍼를 사용했을 경우와 사용하지 않았을 경우의 평균 응답시간이 그림의 Y축 상에 나타날 만큼 차이나지 않았다. 따라서 그림 6과 그림 7에서 TRD와 W-TRD에 해당하는 그래프는 서로 구분이 되지 않을 정도로 겹쳐져 있음을 참고하기 바란다.

반면에 분산형배열 시스템의 경우에는 여러 디스크에 분산된 블럭에 대한 하나의 요청이 여러 개의 부분요청으로 분할되어 각 디스크에서 처리되므로 어느 한 요청의 처리는 다른 요청의 처리에 많은 영향을 주게 된다. 따라서 요청의 도착간 시간이 변하는데에 따른 평균응답시간의 변화는 분산형배열 시스템에서 상당히 크다. 즉, 요청율이 커지게 되면 분산형배열 시스템에서는 각 디스크에서의 대기지연에 의한 영향 때문에 응답시간이 큰 폭으로 증가하게 되는데 이것은 분산형배열 시스템의 특성이면서 또한 문제점이다.

그러나 그림 6에서 볼 수 있는 바와 같이 분산형배열 시스템에서는 기록전용버퍼를 사용한 경우의 읽기요청의 응답시간은 기록전용버퍼를 사용하지 않는 경우에 비해 훨씬 좋다. 예를 들어 요청의 도착간 시간이 20ms인 경우를 보면 응답시간이 약 28%정도 개선되며, 요청율이 높을수록 그 개선효과도 커짐을 볼 수 있다. 이것은 요청율이 높은 경우 응답속도가 좋지 않은 분산형배열 시스템의 단점을 기록전용버퍼가 보완해 준다는 의미가 된다.

그림 6에서 또 주목할 만한 점 한 가지는 평균도착간 시간이 36ms를 기점으로 하여 기록전용버퍼를 사용한 두 가지 유형의 디스크배열 시스템 사이에 성능 반전현상이 있다는 것이다. 즉, 평균 도착간 시간이 36ms 이하에서는 분산형배열 시스템의 대기지연이 크므로 단순배열 시스템이 더 좋은 성능을 보이나, 그 이후에는 분산형배열 시스템의 빠른 전송속도에 의한 이득이 대기지연에 의한 성능 저하보다 우세하므로 단순배열 시스템보다 분산형배열 시스템의 성능이 좋게 나타난다. 이러한 반전 현상은 기록전용버퍼를 사용하지 않은 경우에는 43ms 정도에서 나타난다. 이로부터 우리는 기록전용버퍼가 분산형배열 시스템에서 요청율이 높을 때 대기지연으로 인해 야기되는 성능저하를 완화시켜 주며, 요청율이 낮은 경우에는 본래 대기지연이 크지 않으므로 기록전용버퍼의 효과도 눈에 띄지 않음을 알 수 있다. 이것 또한 분산형배열 시스템의 문제점이 개선되는 것을 단순배열 시스템의 응답시간을 기준으로하여 상대적으로 보여주는 것이다.

기록전용버퍼를 사용함으로써 읽기요청에 대한 응답시간은 개선되었지만 반대로 기록요청에 대한 응답시간은 그림 7에 보인 것처럼 더 나빠졌다. 이것은 읽기요청을 빨리 처리해 주기위하여 기록요청을 일시 저장한다는 기록전용버퍼의 기능을 생각할 때 당연하다고 할 것이다. 기록전용버퍼를 사용하면서도 기록응답시간이 덜 나빠지게 하자면 버퍼내의 기록요청들을 처리해 주는 다른 방법이 요구된다. 예를 들면, 기록전용버퍼의 여유공간이 부족한 경우에 버퍼에 수용된 기록요청을 처리하는 것이 아니라 디스크 스케쥴링 정책으로 이미 알려진 SCAN 또는 C_SCAN[10] 방식처럼 헤드의 현위치를 이용하는 방법도 생각해 볼 수 있을 것이다. 즉, 읽기요청을 처리한 후 헤드가 위치하고 있는 트랙에 대한 기록요청이 버퍼내에 있으면 이를 먼저 처리해 주는 것이다. 사실 우리는 시뮬레이션을 통하여 그런 정책으로 스케쥴했을 때 읽기응답시간이 어느 정도 줄어든다는 것을 확인했다. 이것은 동일 트랙에 대한 처리를 해주기 때문에 적어도 탐색시간만큼의 시간절약을 할 수 있기 때문이라고 본다. 그러나 이로 인해서 읽기요청의 응답 개선효과도 상대적으로 감소하는 것으로 나타났기 때문에 응용분야에 따라 선택할 수 있는 여지로 남겨두고 여기서는 시뮬레이션 결과를 제시하지 않기로 한다.

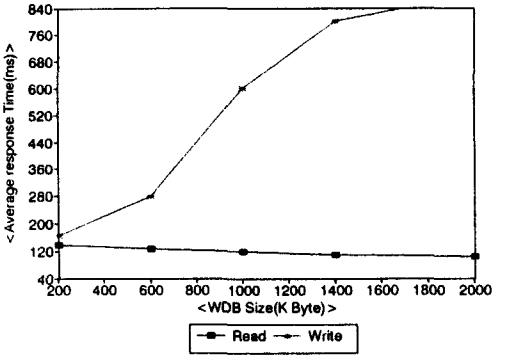


그림 8. 기록전용버퍼의 크기와 응답시간 관계
Fig. 8. The Relation between the Response Time and the Size of Write Dedicated Buffer.

3) 기록전용버퍼의 크기가 미치는 영향

이제까지의 시뮬레이션 결과는 기록전용버퍼의 크기를 400K 바이트로 고정시키고 분석한 것이었다. 그러나 버퍼의 크기에 따라 응답시간에 차이가 있을 것이므로 여기서는 기록전용버퍼의 크기가 응답시간에 미치는 영향에 대해서 분석한 결과를 제시한다. 입출력 요청의 도착간 시간을 20ms로 고정시키고 기록전용버퍼의 크기를 200~2000K 바이트까지 변화시켜가며 시뮬레이션을 수행한 결과를 그림 8에 나타냈다. 그림 8에서 기록전용버퍼가 커질수록 읽기응답시간은 소폭으로 개선되지만 기록응답시간은 버퍼의 크기 600K 바이트 이상에서부터 크게 늘어남을 볼 수 있다. 물론 버퍼의 크기가 커지면 더 많은 기록요청이 누적되므로 기록응답은 그만큼 더 지연되지만 읽기응답이 상대적으로 더 좋아지지 않기 때문에 기록전용버퍼의 크기를 무조건 크게하는 것이 의미가 없음을 알 수 있다.

V. 결 론

분산형 디스크배열 시스템에서 기록전용버퍼를 사용할 때 읽기요청에 대한 응답시간이 개선되는 정도를 분석하였다. 기록전용버퍼는 배열을 이루고 있는 각 디스크에 보내지는 읽기요청과 기록요청 중 기록요청을 일시 저장하여 나중에 처리되도록 함으로써 읽기응답 속도를 향상시켜주는 장치이다.

기록전용버퍼를 스케쥴하는 세 가지의 정책을 비교하여 분석한 결과 선착선처리 정책과 최단대기열 우선정책에 비해 부분스트라입결합 정책이 가장 빠른 읽기응답과 기록응답을 제공해 주는 것을 알 수 있었다. 또한 분산형배열 시스템은 화일 트랜잭션같이 높은 요청율을 보이는 작업 환경하에서는 대기지연시간 때문에 성능이 좋지않다는 단점이 있으나 시뮬레이션을 통하여 기록전용버퍼를 사용함으로써 읽기요청의 도착간 시간이 짧아질수록, 즉, 요청율이 높을수록 더 큰 개선효과를 보이는 것을 알았다. 그러나 단순배열 시스템의 경우에서는 대기지연이 작고 기록요청으로 인해 읽기요청이 지연되는 경우가 드물기 때문에 기록전용버퍼를 통한 입출력 요청의 응답시간은 크게 개선되지 않았다.

본 연구에서는 특히 실시간 응용분야에서 읽기요청을 우선적으로 처리함으로써 입출력 요청을 하는 트랜잭션의 수행속도를 향상시킬 수 있다는 전제 하에서 기록전용버퍼를 사용함으로써 요청율이 높은 경우에 분산형배열 시스템의 읽기요청에 대한 응답시간을 개선시킬 수 있음을 보였으나 통상의 디스크 작업요청에서 7:3으로 읽기요청이 많은 점을 감안할 때 읽기요청의 응답개선은 전체 작업수행시간을 단축하는데 큰 기여를 한다고 본다. 그러나 상대적으로 기록요청의 처리가 지연되는 부작용이 있으므로 읽기응답을 향상시키면서 동시에 기록응답시간을 보존 내지 개선시키는 방안을 향후 연구과제로 다룰 예정이다. 또한, 디스크배열 시스템은 신뢰도를 보장하기 위해 에러가 발생했을 경우 이를 복구하는데 필요한 정보를 유지하며 동작하게 되므로 이러한 점까지 감안한 응답속도 개선에 관해서 계속적인 연구가 필요하다고 본다.

參 考 文 獻

[1] H. T. Kung, "Memory Requirements for Balanced Computer Architecture," Proc. 13th Ann. Int'l Symp. Computer Architecture, pp. 49-54, 1986.

[2] R. H. Katz, G. A. Gibson and D. A. Patterson, "Disk System Architectures for High Performance Computing," Proc. of the IEEE, vol. 77, no. 12, pp. 1482-1458, Dec. 1989.

[3] K. Salem and H. Garcia-Molnia, "Disk Striping," Proc. of the Second Data Engineering Conf., pp.336-342, 1986.

[4] Michelle Y. Kim, "Synchronized Disk Interleaving," IEEE Trans. Comput., vol. C-35, no. 11, pp. 978-988, Nov. 1986.

[5] David. A. Patterson, Garth Gibson and Rabdy H. Katz, "A Case for Redundant Array of Inexpensive Disks(RAID)," Proc. of ACM SIGMOD Conf., pp. 109-116, Jun. 1988.

[6] Robert K. Abbott and Hector Garcia-Molina, "Scheduling I/O Requests with Deadlines: a Performance Evaluation," Proc. Real-Time Systems Symposium '1990, pp. 113-124, 1990.

[7] A. L. Narasimha Reddy and Prithviral Banerijee, "An Evaluation of Multiple-Disk I/O Systems," IEEE Trans. on Computers, vol. 38, no. 12, pp. 1680-1690, Dec. 1989.

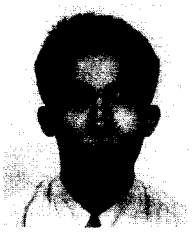
[8] Peter. M. Chen et al., "An Evaluation

of Redundant Arrays of Disk Using an Amdhal 5890," Proc. of the 1990 ACM SIGMETRICS Conf., pp. 74-85, May. 1990.

[9] John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann Publishers, Inc. 1990.

[10] A. Silberschatz, J. Peterson, and P. Galvin, Operating System Concepts, 3rd ed. Addison-Wesley, 1991.

著 者 紹 介



尹 齊 絃(正會員)

1967年 10月 8日生. 1990年 2月 한양대학교 전자계산학과 졸업. 1992年 2月 한양대학교 대학원 전자계산학과 석사학위 취득. 1992年 3月 ~ 현재 한국 디지탈컨트롤 근무. 주관심 분야는 분산제어 시스템 및 설비진단시스템 등임.

全 昌 浩(正會員)

1977年 한양대학교 전자공학과 졸업. 1977年 ~ 1979年 한국전자통신연구소 연구원. 1982年 미국 Comell 대학교 석사학위 취득. 1986年 미국 Comell 대학교 박사학위 취득. 1986年 3月 ~ 1989年 2月 성균관대학교 전기공학과 조교수. 1989年 3月 ~ 현재 한양대학교 전자계산학과 부교수. 주관심 분야는 병렬처리시스템, 성능분석 등임.