
지능형 I/O 구조를 갖는 RAID 시스템의 성능 향상을 위한 연구

최귀열 · 박계원

A Study for High Performance of Intelligent I/O Architecture of RAID System.

Gwi-yoel Choi · Kye-won Park

요 약

RAID는 디스크 배열 상에 데이터를 이중으로 저장하거나 패리티를 사용하는 기법으로 디스크에 장애가 일어났을 경우 이를 복구하는 구조로 되어 있다. 본 논문에서는 지능형 I/O 구조를 갖는 고성능 I/O 서브시스템에서 CPU로부터 대부분 I/O 작업 부하를 줄이고 동시에 I/O 성능을 향상시키는 방안에 대하여 연구한다. 지능형 I/O 장치의 디스크 스케줄링 디자인에 관한 초당 MB 처리율, 최대 I/O 응답시간 등의 실험 통하여 새로운 스케줄링 기법에 의해 성능을 개선한다.

ABSTRACT

RAID(Redundant arrays of inexpensive disks) were proposed as a way to use parallelism between multiple disks to improve aggregate I/O performance. The emerging of intelligent I/O architecture provides a standard for high performance I/O subsystems and introduces intelligence at the hardware level. With an embedded processor, intelligent I/O adaptors can offload the major I/O processing workload from the CPU and, at the same time, increase the I/O performance. This paper addresses the essential issue in the design of disk scheduling for intelligent I/O devices. In this paper we compare with MB throughput per second and maximum I/O respond time in RAID.

키워드

RAID, I/O performance, intelligent I/O

I. 서 론

대용량의 저장 공간과 고속 통신망을 갖춘 고성능 저장 장치를 필요로 하는 멀티미디어 응용에서는 데이터 전달 속도와 입출력 성능을 향상시키기 위해 디스크 배열이 사용되고 있다. 디스크 배열은 구성 방법 및 데이터 할당 방법 등에 따라 성능의 차이를 보이므로 디스크 배열을 설계할 때 해당 응용에 적합한 디스크 배열 특성 변수가 결정되어야 한다.

최근 들어 대용량의 디스크 저장 장치와 멀티미디어 데이터 압축기술 등의 컴퓨터 기술과 기가비트 전송율을 갖는 초고속 통신망 등의 발전으로 사용자에게 대용량의 음성(audio) 자료, 동화상(video) 자료 등의 멀티미디어 데이터를 서비스 할 수 있게 되었다. 최근 프로세서와 저장 장치의 처리 속도의 차이에 의해 전체 시스템의 I/O 병목 현상을 일으키고 있으며 이러한 현상을 해결하기 위하여 I/O 장치의 처리속도를 향상시키려는 연구가 진행되고 있다.[1],[2],[3]. 최근에는 큰 용량을 가진 하나의 디스크

* 청운대학교 인터넷학과

** 배재대학교 컴퓨터공학과(교신저자)

를 작은 크기의 여러 디스크들로 구성으로서 대역폭 (bandwidth) 및 처리율(throughput)을 증가시키고 빠른 응답시간(response time)을 얻을 수 있는 I/O 구조에 대한 연구가 진행되고 있다.

지능형 I/O 구조는 고성능 I/O 서브시스템의 표준과 지능형 하드웨어 레벨을 제공한다. 임베이드 프로세서 지능형 I/O 어댑터는 CPU로부터 대부분 I/O 작업부하를 줄이고 동시에 I/O 성능을 향상시킨다. 지능형 I/O 요구에서 실시간 다중디스크 스케줄링과 고효율적 알고리즘의 제한을 최소화 하는 동시에 요구의 응답을 향상시킨다.

본 논문에서는 RAID의 읽기/쓰기에 대하여 초당 MB 처리율, 최대 I/O 응답 시간 등을 비교하여 I/O의 성능을 개선한다.

II. 기존의 스케줄링 기법

멀티캐스트(Multicast)기능을 사용하여 동시간 또는 근접한 시간대에 요구되어진 요청들을 하나의 스트림으로 서비스하는 기술을 배칭이라 한다. VOD 서버당 수용 가능한 가입자의 수는 서비스 이용료에 직접적인 영향을 미치면 스케줄링 방법에 따라 크게 변화 한다. 최적의 스케줄링은 배칭을 최대화 하여 스트림의 효율을 증대 시키는 동시에 서비스 지연시간과 서비스 신청 후 지연에 따른 포기율을 최소화하고 가입자들에게 최대한의 공평성을 보장하여 주는 것이다.

2.1. VOD 서버의 스트림 스케줄링

기존에 제안된 스케줄링 기법으로는 다음과 같은 것들이 있다.

① FCFS(First-Come-First-Service) : 사용 가능한 스트림이 있으면 언제라도 즉시 요청순서에 따라 서비스를 시작하는 방법으로 프로그램의 종류에 따른 기호도에 좌우되지 않아 공평성이 뛰어나고 서비스 지연시간이 짧은 장점이 있는 방법이나 스트림 이용 효율성이 낮아 포기율이 높은 것이 단점이다.

② MFQL(Maximum Factored Queue Length) : VOD 시스템에 저장된 프로그램 별로 요청된 서비스를 저장하는 독립적인 큐를 두어 스트림 할당시 가장 큰 FQL을 가진 비디오를 스케줄하는 방법이다. 서비스 지연시간이 짧은 장점이 있으나 포기율이 크고 공평성이 떨어지는 단점이

있다.

③ BML(Batch Maxiimal Loss) : 스트림 이용 효율성을 높이기 위하여 일정기간(Tolerance Interval)을 기다린 요청을 보유한 프로그램 중에서 다음 스트림이 서비스 완료될 때까지 포기가 가장 많이 기대되는 프로그램에 스트림을 할당하는 방법이다. 낮은 포기율이 보장되나 일정기간 지연되는 동안에 여분의 스트림을 효율적으로 사용하지 못하는 단점이 있다.

2.2. 디스크 I/O 스케줄링

일반적 디스크 스케줄링은 단일 디스크 시스템에 초점을 맞추고 이것을 SCAN처럼 SSTF(shortest seek time first), C-SCAN(circular SCAN), FIFO이고 여기서 SCAN은 디스크의 한 면으로부터 다른 면까지의 요청을 서비스하고 후진동도 가능하다.

①C-SCAN은 SCAN의 변형이고 C-SCAN은 항상 한 면에서 다른 면까지 요청을 서비스하고 즉시 읽기/쓰기 헤드는 다른 면에서 도달하고 디스크는 어떤 요청의 서비스 없이 되돌아오기를 실행한다. SSTF는 항상 읽기/쓰기 헤드 위치를 마무리 할때 요청을 서비스한다.

② FIFO는 요청의 도착 순서에 따라 서비스된다. Andrews등은 [4] 단일 디스크 스케줄링의 최적화 문제를 비동기적 접근 알고리즘 제공에 대하여 발표했다. 최근 연구자들은 다양한 실시간 디스크 스케줄링에 기반한 요청제한 등에 대한 것은 제안했다. [5,6]. Chen은 가중-함수를 기반으로한 알고리즘을 Shortest Seek Time Earlist Deadline By Value(SSEDV)라고 하고 요청 제한과 그들의 탐색시간을 고려했다. Reddy와 Wyllie는 SCSI 버스의 조건을 확장하는 것을 그들의 제한조건에 가중함수를 최우선으로 정의했다. Bruno[5]등은 두-레벨 스케줄링 매카니즘을 제한적 조건에 기초한 큐와 작업은 그들의 제한에 의해 분류된다. 가장 시급한 작업은 각 제한적 조건에 기초한 큐는 시스템 큐에 삽입되어진다. 여기서 시스템 서비스 작업은 시스템 큐 안에서 SCAN 유형 안에서 이루어진다. Hwang과 Shin[6]은 제한적 동작 알고리즘을 제안했고 시급한 그룹에 기초한 요청의 실행이다. 시급한 요청은 항상 제일먼저 서비스 되고 Abbott와 Garcia-Molina는 SCAN처럼 제한적 동작 알고리즘을 제안했다. 이 알고리즘은 최종 제한적에서 요청을 우선 선별하고 다음 실린더와 현재 실린더 그리고 최종 제한적 요청의 실린더와 모든 요청을 제한적으로 서비스 한다. Chang등은 제한적-단

조 SCAN 알고리즘을 제안하고 이것은 디스크 접근의 하드웨어 제한적 장점을 갖고 여기서 디스크 접근시 작업부하는 분산되어진다.

연구자들을 통해 단일 디스크 스케줄링을 알고리즘들의 다양한 우수성을 제안하였으나 다중디스크 스케줄링에 관한 것은 거의 없다. 특히 실시간 RAID 응용은 더욱 그렇다. 실제로 Weikum과 Zabback은 RAID의 동시성에서 스트라이핑 크기의 영향에 대해 성능, 비실시간 응용을 연구했다. 연구자들의 목표는 실시간 디스크 스케줄링을 위한 지능형 I/O RAID-O 장치의 향상이고 낮은 실행시간 오버헤드를 갖고 RAID-O는 블록 스트라이핑 구조를 갖는 RAID의 표준이다. 지능형 I/O RAID-O 장치는 디스크 스케줄링을 위한 전력소모가 거의 없는 임베이드 프로세서 사용한다. 지능형 I/O RAID-O 요청에서 일반적으로 간단한 실시간 제한을 갖고 디스크 스케줄링 알고리즘은 오직 I/O 성능의 최대화만을 위한 것이 아니고 처리율 또는 응답시간의 제한적 손실을 갖는 사항에서 요청의 수를 최소화 해야 한다.

III. RAID의 특징

RAID에 관한 연구는 디스크 배열을 구성하는 디스크의 물리적인 성능에 대한 연구보다는 디스크 배열을 중심으로 하는 데이터 저장 구조와 데이터 복원 등의 문제에 중점을 두고 있다. Salem은 디스크 상에 저장되는 데이터의 분할(striping) 구조에 대하여 연구한 바 있고 Patterson은 디스크 데이터 분할기법을 사용하는 디스크 배열에 데이터 복원기법을 부가하고 복원에 사용되는 데이터의 관리 방법에 따라서 여러 가지 RAID 레벨에 따른 저장 기법에 관한 연구를 수행하였다.[1]. 또한 RAID를 구성하는 구성 인자들이 성능에 미치는 영향에 대한 연구도 최근에 들어 많이 수행되고 있다. 특히 RAID5는 분산 구조의 결합 제어 관리방식을 사용하기 때문에 다른 RAID 레벨로 구성된 저장 장치에 비하여 복잡한 데이터 저장 구조를 가지고 있다. RAID는 에러복구 코드와 데이터의 분산 방법에 따라 여러 단계로 구분할 수 있으며 지금까지 데이터와 패리티 블록의 위치 결정 분산단위 크기 연구 등 전반적인 입출력 성능을 높이기 위하여 추가 부담이 적은 패리티(parity)를 사용한다. 데이터를 모든 디스크에 분산시켜 큰 접근일 때 높은 대역폭으로 인출(fetch)할 수 있게

하였고 이에 대한 패리티를 분산시킴으로써 특정 디스크 집중을 제거하였다.

3.1. RAID 종류

RAID는 디스크 배열의 신뢰성의 문제를 해결하기 위해 Patterson 교수 등에 의해 제안된 개념으로 디스크에 저장되는 데이터의 구조가 이중으로 되어 있거나 패리티를 두어서 하나의 디스크 장애시 나머지 디스크들이 장애가 일어난 부분을 복구할 수 있는 구조로 되어 있다. RAID는 디스크에 자료를 분할하는 방법과 장애 복구 방법에 따라서 여러 가지 레벨로 나뉜다. 현재 Patterson 교수 등이 구분한 7가지의 레벨 외에 RAID 레벨 7, 레벨 10, AUTO RAID 등 레벨별의 특징을 혼합한 형태의 RAID가 소개되고 있다.

표 1. RAID구조의 특성
Table.1 Characteristic of RAID Architecture

| 레벨 | 제 목 | 전송 형태 | 성능 효과 |
|----|---|-----------|----------|
| 1 | Disk Mirroring | 병렬 | 높은 전송율 |
| 2 | Hamming Code for Error Correction | 병렬 | 높은 전송율 |
| 3 | Single Check Disk per Group | 병렬 | 높은 전송율 |
| 4 | Independent Disk Array | 독립적 전송 | 높은 I/O 율 |
| 5 | Rotating Independent Disk Array | 패리티 회전 분산 | 높은 I/O 율 |
| 6 | Two Dimensional Parity Independent Disk Array | 이중 패리티 분산 | 높은 I/O 율 |

3.2. RAID 컨트롤러

제어 시스템은 우선 순위의 예정화 모델과 실시간 동작시스템으로 되어 있다. 제어 소프트웨어는 아래와 같이 조직되어 있다. 주 제어모듈은 호스트 등으로 부터 데이터 또는 전송상태를 요구수신을 재구성성을 포함하고 호스트로부터 데이터와 전송상태를 받고 재구성 한다. 버퍼 관리모듈은 데이터 재구성모듈 데이터 재제작 모듈 주 제어모듈은 기본 임무를 생성하고 실시간 동작을 시작한다. 그 주기능은 호스트로부터 요구를 받고 요구에 의해 부 임무를 하고 데이터의 전송을 호스트 혹은 상태로부터 데이터를 전송한다. 그 기본임무는 호스트로부터 요구를 받고 그 요구를 받을 때 많은 디스크는 부대기를 드라이브

로 부터 요구한다. 재생(기록요구는 다중재생)기록 요구를 즉시하고 요구재생의 포맷은 아래와 같다.

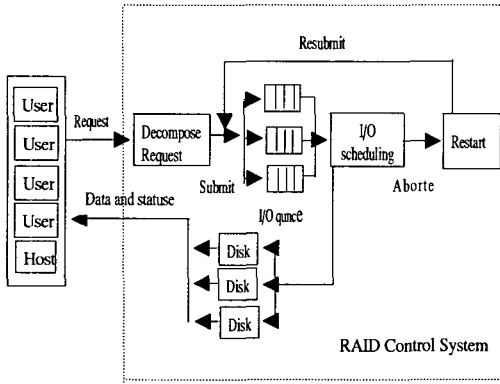


그림 1. RAID 컨트롤러 시스템
Fig. 1. RAID Controller System

재생 데이터 기록재생 먼저데이터, 재생 먼저패리티, 기록 새로운 데이터, 기록 새로운 패리티, 이같은 모든 요구는 I/O대기로부터 들어오고 그들은 다중대기 실시간 I/O 예정화 알고리즘에 의해 스케줄 된다. 데이터 재구성 모듈은 재임무를 형성하고 어떤 임무를 발견할 때 시작되고 어떤 디스크는 손실된다. 시스템으로 들어온 그레이드 모델은 새로운 디스크가 빈 디스크로 배치될 때 그 시스템은 재형성모델로 들어간다. 부임무는 그레이드 모델에서 보통 모델과 다르다.

① 재생; 재생먼저데이터 1 재생먼저데이터 N-1 빈 디스크에서 먼저 데이터의 제2의 계산에 의해서 얻어진다.

② 기록; 역세스 데이터 유닛의 임무를 원하면 부임무는 먼저데이터 1를 읽고 먼저데이터 N-1를 읽고 새로운 패리티를 기록한다. 한편 임무는 데이터 재구성모듈을 형성하고 디스크가 비어있을 때 시작된 새로운 디스크에 의해 배치가 된다. 시스템 입력은 재구성모델이고 시스템이 자유로울 때 재구성임무와 부임무회복은 비어있는 디스크의 본래 데이터이다. 회복된 빈 디스트의 데이터는 보통모델의 시스템으로 되돌아 간다.

IV. 지능형 I/O 시스템 구조

4.1. 지능형 I/O 장치구조

지능형 I/O 규격은 산업의 고급 기술자들에 의해 제안

되었고 마이크로소프트, 인텔, 휴렛패키드, 3COM, 컴팩 등에서 차세대 I/O 서비스시스템의 표준처럼 되었다. 지능형 I/O규격에 포함 된 것은 하드웨어 레벨의 기능 그리고 산업의 각 분야의 표준을 제공한다. 지능형 I/O규격은 OS 매대인에 의하고 장치의 각 분류를 위한 단일 드라이브가 진행되고 드라이브의 OS 최적을 위해 전념한다. 임베이드 프로세서를 갖는 지능형 I/O 어댑터는 CPU로 부터의 대부분 I/O진행 작업부하를 없애주고 동시에 I/O 성능을 증가시킨다. 지능형 I/O 어댑터와 호스트 사이 인터페이스는 PIC버스처럼 정의 되어진다. ACE 6850은 임베이드 된 프로세서를 가졌고 인텔 i960, 메모리, SCSI 어댑터 5 개 이상이다. 각 SISC 어댑터는 15장의 디스크가 연결되었고 지능형 I/O 어댑터의 메모리 공간은 호스트의 메모리 주소 영역에 의해 맵핑 되어지고 마치 호스트와 타아겟은 DMA에 의해 통신된다. 지능형 I/O구조의 드라이버를 둘로 분류하면 OSM(os규격모듈)과 DDM(Device Driver Module)으로 구분된다.

OSM은 호스트 옆에 구현되고 DDM은 타아겟 옆에 위치한다. OSM은 인터페이스에서 호스트 동작 시스템에 제공되고 OS 판매자에 의해 통상 제공되어진다. OSM은 DDM을 통해 메시지를 통신한다. 지능형 I/O 실시간 동작 시스템은 지능형 I/O 어댑터의 프로세서 I/O수신 요청으로부터 호스트 OSM을 통해 동작되고 디스크 서비스가 스케줄링 된다. 모든 디스크의 동작은 DDM 조정함수에 의해 초기화되어진다. DDM은 두 부분으로 구성되고 내부 서비스 모듈(ISD)과 하드웨어 디바이스 모듈(HDM)이다. HDM은 하드웨어 규격 코드로 메시지 장치 조절기를 포함하고 ISM은 하드웨어 판매자가 더욱 많은 기능을 장치에 계획하기 위한 것이다. 즉 실시간 디스크 스케줄링 혹은 자원 관리를 포함한다.[5,6].

4.2. I/O 예정화 알고리즘

컴퓨터 시스템에서 통상 속도가 가장 늦은 것은 I/O 장치이고, 그것은 I/O 동작의 효율적 향상은 RAID 시스템에서 매우 중요하다. 서비스 시간을 줄이기 위해 고성능 I/O 예정화 알고리즘이 필요하다. I/O 예정화 알고리즘의 FCFS는 가장 간단한 방법이고 그 I/O 요구는 첫 번째 들어온 것을 첫 번째 연속적되는 예정화이다. SCSI는 향상된 알고리즘이다. 디스크가 한 방향으로 이동할 때 그 방향에서의 요구는 스케줄 될때까지 요구되지 않는다. 스캔 방향이 바뀌고 서비스 요구가 바뀔 때 C_SCSEN 서클러스

캔 알고리즘이다. 이것은 스캔처럼 스캔 방향에서 요구되는 예정화 후에 그 스캔 목표로 다시 되돌아가고 서빙없이 어떤 위치에서 요구가 시작되고, 다른 스캔이 시작된다. SSTF는 서비스를 위한 현재 디스크 목표 위치를 요구한다. P스캔은 I/O 예정화의 첫 번째이다. 대기에서 모든 요구는 다중 우선 레벨로 분류된다. 이 스캔 알고리즘은 각 레벨에서 사용되고, 디스크는 현재 우선 레벨에서 요구 될 때까지 요구가 없다. 디스크 서비스가 완전히 끝나고 스케줄러는 보다 높은 요구가 어느 것인가 체크하고 서비스를 위해 기다린다. 만약 이것이 스케줄러가 보다 높은 레벨스위치라면 그것은 작은 우선 레벨보다 좋은 성능을 갖는다. FDSCSN은 실질적 테드라인 SCSN이고 이 알고리즘은 빠른 실질 테드라인을 요구하고 스캔 방향에 의해 결정된다. 만약 요구의 테드라인이 현재시간+보다 크면 서비스시간 그 테드라인은 실현된다. 현재 디스크 목표위치와 트랙은 액세스 요구 서비스 시간을 결정한다. 예정화 관점에서 모든 요구는 빠른 실현 테드라인을 가지고 있는가를 결정한다. 다음 스캔 방향은 선택되고 디스크 목표는 전진방향으로 움직이고, 그리고 모든 요구는 방향을 따라 서비스된다.

SSEDO 알고리즘은 윈도우에서 각 요구하고 주어진 웨이트는 순차적으로 증가한다. 요구의 우선순위는 현재 디스크위치 목표와 다중거리 웨이트 사이에서 위치를 요구한다. 그 요구는 가장 적은 우선순위가 선택되고, 만약 하나의 요구보다 적은 우선순위를 가지면 그 요구는 빨리 테드라인을 선택하고 그 각자 요구 우선 순위는 디스크 목표 움직인다. SSEDV 알고리즘은 SSEDO 처럼 스케줄링 규칙이 같고 계산 우선 순위의 방법은 다르다. SSEDO와 SSEDV는 다양한 스캔보다 성능이 우수하고 그것은 실시간 시스템의 성능을 향상시키고 쉽게 구현할 수 있다.

RAID에서 각 디스크의 I/O 대기는 사용자로부터 요구된 후 다시 구성되고 모든 서버 요구입력은 대기된다. 대기 중에서 요구 사이에 대한 관계는 판독기로 기록판독 기록 관계이다. [그림 4]에 나타내고 R-1과 R-2는 대기에서 요구의 관계를 나타내고 R-3는 대기중의 요구사이의 관계를 나타낸다. RAID에서 I/O요구는 실시간 I/O 예정화 알고리즘의 다중대기는 다음과 같다. 모든 디스크는 I/O대기를 갖고 대기가 들어올 때 대기와 대기의 가운데의 요구는 요구의 독립에 의해 결정된다.

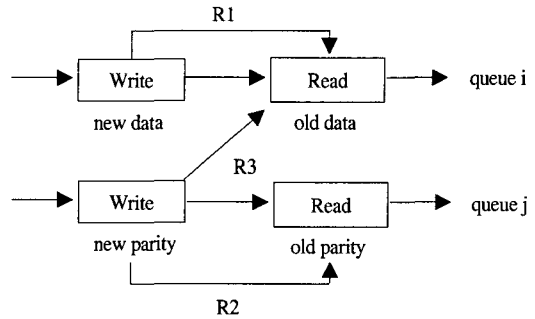


그림 2. RAID에서의 대기
Fig. 2. Queue in RAID

SSEDV 알고리즘은 모든 대기에서 사용되고, 대기에서 요구에 의존한다면 파괴가 되고 SSEDV 알고리즘에 의해 요구는 선택된다. 만약 대기가운데의 요구에 의하면 파괴가 되지 않으면 임무 그룹은 병렬 스케줄로부터 선택된다. 많은 임무는 병렬에서 무수한 스케줄이 되고 임무의 그룹을 끝낸 후 단계 3으로 점프한다. RAID는 하나의 진행시스템이고 병렬 I/O 임무를 실행한 것은 아래와 같다. 실행 임무는 데이터를 기다리고 스케줄러는 슬립상태로 들어간다. 데이터가 도착할 때까지 들어온 상태를 읽고 과정이 자유로울 때 스케줄러는 병렬 타 스케줄러에서 다른 임무를 수행한다. 새로운 알고리즘을 사용하여 많은 I/O 임무를 병렬로 스케줄하고 요구된 서비스 시간은 여러 번 중복된다. SSEDV의 알고리즘의 장점은 요구의 서비스 시간이 줄어들고 알고리즘이 RAID 성능을 향상 시킨다고 생각한다.

V. 실험 및 결과

실험은 시뮬레이션 도구인 SMPL이나 RaidSim을 사용하지 않고 Host의 환경은 CPU는 1GHz, RAM은 524MB, HDDE는 IBM 40G, SCSI Card는 Adaptec 29160 ultra 160, OS는 Win2K-pro이다. 또한 RAID 환경은 RAID Level5에서 HDD는 Maxtor 200G*8, Diamond Max plus9 ATA/133, CPU는 80308, RAID 용량은 1.36Tera이고 파일 크기는 영상 데이터에서는 128K, 512K, 1M로 문자 데이터에서는 1K, 16K, 64K로 가변하였으며 또한 스트라이프 크기도 64K, 32K, 8K로 변환시켜 실험하였다.

5.1. 영상 데이터 읽기/쓰기(100 %)

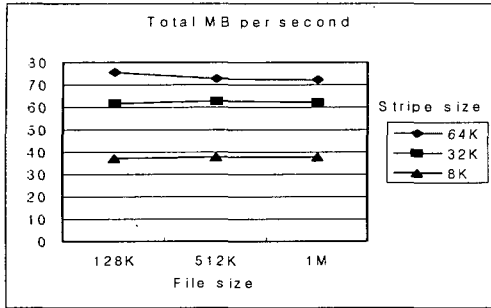


그림 3. 초당 총 MB 처리율
Fig. 3. Throughput of total MB per second

[그림3]에서와 같이 동영상 데이터 읽기/쓰기 100(%) 일 때의 조건에서 스트라이프 크기를 64k, 32k, 8k로 가변 하면서 파일 크기에 대한 초당 총 MB 처리율을 나타낸 것이다. 결과 그래프에서와 같이 파일의 크기에는 거의 관계가 없고 스트라이프의 크기에 따라 처리율이 달라짐을 알 수 있다.

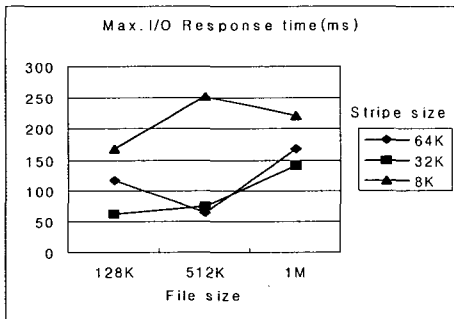


그림 4. 최대 I/O 응답시간
Fig. 4. Maximum I/O response time

[그림3]의 조건과 같이 실험한 최대 I/O 응답시간을 [그림4]에 나타내었다. 실험 결과와 같이 최대 I/O 응답 시간은 파일의 크기와 스트라이프의 크기에 따라 변화가 현저하였다. 특히 파일의 크기가 512K이고 스트라이프의 크기가 8K 일때 성능이 우수함을 알 수 있다.

5.2. 문자 데이터 읽기/쓰기(100 %)

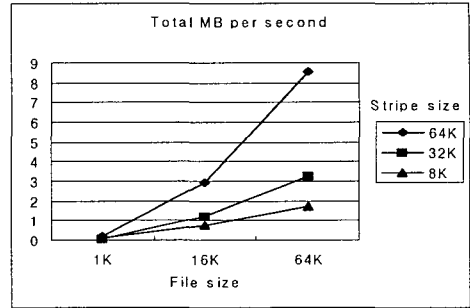


그림 5. 초당 총 MB 처리율
Fig. 5. Throughput of total MB per second

[그림5]에서와 같이 문자 데이터 읽기/쓰기 100(%) 일 때의 조건에서 스트라이프 크기를 64K, 32K, 8K로 가변 하면서 초당 총 MB 처리율을 나타낸 것이다.

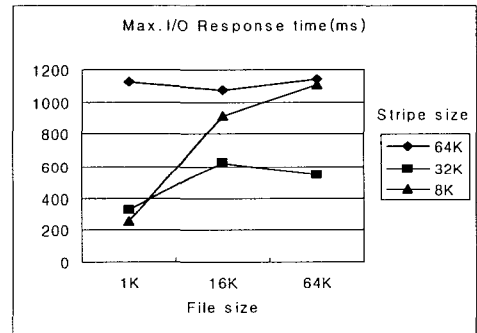


그림 6. 최대 I/O 응답시간
Fig. 6. Maximum I/O response time

[그림6]은 [그림5]와 같은 조건에서 파일 크기에 따른 스트라이프 크기에 대한 변화를 나타낸 것이다.

VI. 결 론

최근 초고속 통신망, 멀티미디어 데이터 압축기술, 대용량의 디스크 저장 장치 등의 발전으로 사용자에게 대용량의 음성(audio) 자료, 동화상(video) 자료 등의 멀티미디어 데이터를 서비스 할 수 있게 되었다. 그러나 프로세서와 저장 장치의 데이터 처리 속도의 차이에 의해 전체 시스템의 I/O 병목 현상을 일으키고 있으며 이러한 현상을

해결하기 위하여 I/O 장치의 처리속도를 향상시키려는 연구가 진행되고 있다. 그중의 하나가 다양한 실시간 디스크 스케줄링에 기반한 제안들이다.

본 논문에서는 지능형 I/O 구조를 갖는 고성능 I/O 서버 시스템의 표준과 지능형 하드웨어 레벨에서 임베이드 프로세서, 지능형 I/O 어댑터, CPU로부터 대부분 I/O 작업 부하를 줄이고 동시에 I/O 성능을 향상시키기 위해 초당 MB 처리율, 평균 I/O 응답시간 등을 비교 연구 한다.

참고문헌

- [1] D. Patterson, G. Gibson, and R. Katz, "A Case for Redundant Arrays of Inexpensive Disks(RAID)," Proceeding of the ACM SIGMAOD Conference, pp.109-116, 1988.
- [2] R. H Katz, D. A. Patterson, and G. A. Gibson, "Disk System Architectures for High Performance Computing," Proceedings of IEEE, Vol. 77, No. 12, pp.1842-1858, Dec 1989.
- [3] D. Stodolsky, G. Gibson, And M. Holland, "Parity Logging Overcoming the Small Write Problem in redundant Disk Arrays," Proceedings of the 20th Annual International Symposium on Computer Architecture, pp. 64-75, May. 1993.
- [4] M. Andrews, M.A. Bender, L. Zhang, "New Algorithms for the Disk Scheduling Problem," The 37th Annual Symposium on Foundations of Computer Science, PP. 550-559. 1996.
- [5] J. Bruno, J. Brustoloni, E. Gabber, B. Ozden ,and A. Silberschatz, "Disk Scheduling with Quality of Service Guarantees," IEEE Int. Conf. on Multimedia COMPUTING AND Systems, PP. 400-405, 1999.
- [6] K. Hwang and H. Shih, "Real-Time Disk Scheduling Based on Urgent Group and Shortest Seek Time First," The 5th Euromicro Workshop on Real-Time Systems, pp. 124-130. 1993.

저자소개

최 귀 열(Gwi-Yoel Choi)



1980년 2월 숭실대학교 전자공학과 (공학사)

1987년 8월 숭실대학교 전자공학과 (공학석사)

2002년8월 숭실대학교 전자공학과(공학박사)

1991년3월~현재 재능대학 정보통신과 교수

※관심분야: 컴퓨터 구조, VOD, 통신회로 설계, 퍼지 및 신경망.

박 계 원(Kye-Won Park)



현재 재능대학 정보통신과 교수

※관심분야: 데이터 통신, 화상통신, 이동통신.