파일과 스트라이프 크기에 대한 RAID5의 읽기/쓰기 성능 비교

최귀열, 박계원
재능대학

Performance Compression of RAID5 Read/write to File and Stripe Size.

Gwi-yoel Choi, Kye-won Park
*Jainueng College
E-mail : gychoi@mail.jnc.ac.kr, kye@mail.jnc.ac.kr

요 약

RAID는 디스크 배열 상에 데이터를 이중으로 저장하거나 패리티를 같이 사용하는 기법으로 디스크에 장애가 일어났을 경우 이를 복구하는 구조로 되어 있다. RAID5는 읽기와 큰 데이터 쓰기 접근을 수행하는 2차 저장 장치에서 높은 신뢰성과 비용 효과를 갖는다. 그러나 RAID5는 작은 크기의 데이터 쓰기에 대하여 패리티를 다시 계산하고 유지하는데 소비되는 오버헤드 때문에 성능이 저하된다. 본 논문에서는 패리티 로깅 기법에 의한 파일, 스트라이핑 크기에 대한 RAID5의 읽기/쓰기 성능을 비교 시스템의 성능을 개선한다.

ABSTRACT

RAID were proposed to stored double data or used to parity logging methode for error recovery. We describe a technique for automating the execution of redundant disk array operation, including recovery from errors, independent of array architecture. RAID5 provide highly reliable cost effective secondary storage with high performance for read access and large write accessed. It discusses the two architectural techniques used in disk arrays, striping across multiple disks to improve performance and redundancy to improve reliability. In this paper we compare with performance and reliability in RAID5 read/write to file and stripe size. than suggest to algorithm.

① 키워드

② RAID, Parity logging, stripe, redundancy,

Ⅰ. 서 론

최근 초고속 통신망, 멀티미디어 데이터 압축 기술, 대용량의 디스크 저장 장치와 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 입출력 병목 현상을 일으키고 있으며 이러한 현상을 해결하기 위하여 입출력 장치의 처리속도를 향상시키려는 연구가 진행되고 있다.[1],[2],[3]. 즉 기존에 사용되던 SLED(Single Large Expensive Disk)는 부피가 크고 가격이 비싸며 데이터 손실 가능성의 단점을 가지고 있

다. 이러한 단점을 개선한 RAID(Redundant Arrays of Inexpensive Disks)는 작은 디스크를 여러 개 사용하여 병렬로 구동하기 때문에 동시에 많은 양의 작업을 처리할 수 있어서 높은 데이터 처리율(data rate)을 나타내며 작업을 여러 디스크로 분산 접근하여 수행함의로써 작업을 기다리는 병목 현상을 줄일 수 있어서 높은 입출력 처리율(I/O rate)을 나타낸다. 또한 RAID는 가격이 낮고 용량이 크며 접근 시간이 빠르고 대역폭이 넓고 신뢰성이 높은 장점을 가진다.

특히 RAID5는 블록 단위로 데이터를 처리하며 패리티 블록을 각 디스크로 분산시켜 디스크 접근 병목 현상을 제거한다. RAID5는 소량의 읽기 요청과 대량의 읽기/쓰기 요청에는 매우 효율적이지만 작은 쓰기 요청인 경우는 패리티를 계산하기 위해서 읽고 수정하고 기록하는 과정을 수행하기 때문에 경사 기법에 비해서 좋은 성능을 보이지 못한다. 따라서 블록 단위 작은 쓰기를 자주 하는 OLTP(On Line Transaction Processing) 작업에서 RAID5는 작은 데이터 쓰기 문제를 일으키기 때문에 좋은 성능을 기대하기 어렵다. 이러한 데이터 쓰기 문제를 해결하기 위하여 제안된 패리티 로깅(parity logging) 기법은 디스크에서 쓰기 작업이 일어날 때 패리티 유지를 위해서 작은 크기로 빈번히 디스크를 접근하지 않고 버퍼에 저장하였다가 큰 크기의 트랙 단위로 로그 디스크에 순차적으로 저장하는 방법으로 디스크 처리 시간을 줄인다.

본 논문에서는 RAID5의 읽기/쓰기에 대하여 파일의 크기, 스트라이프의 크기, 초당 전체 I/O 처리율, 초당 MB 처리율, 평균 I/O 응답시간, 최대 I/O 응답시간 등에 대하여 알아본다.

II. RAID에 관한 연구

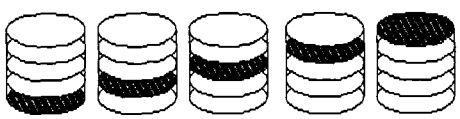
RAID에 관한 연구는 디스크 배열을 구성하는 디스크의 물리적인 성능에 대한 연구보다는 디스크 배열을 중심으로 하는 데이터 저장 구조와 데이터 복원 등의 문제에 중점을 두고 있다. Salem은 디스크 상에 저장되는 데이터의 분할(striping) 구조에 대하여 연구한 바 있고 Patterson은 디스크 데이터 분할기법을 사용하는 디스크 배열에 데이터 복원기법을 부가하고 복원에 사용되는 데이터의 관리 방법에 따라서 여러 가지 RAID 레벨에 따른 저장 기법에 관한 연구를 수행하였다. 따라서 RAID5의 성능을 개선시키기 위한 연구가 다양하게 수행되었다. RAID5에서 한번의 작은 쓰기를 위해서는 기존의 데이터 읽기 기존의 패리티 읽기 새로운 데이터 쓰기 새로운 패리티 쓰기 등 4번의 디스크 접근을 요구한다. 따라서 작은 쓰기가 많이 발생하는 OLTP 시스템에서는 성능이 크게 악화 된다. 이와 같은 단점을 보안하기 위해서 Write Buffering & Caching, Parity Caching[4]. Floating parity. Parity logging[5] 등이 제안되었다

2.1 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.[1]. 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.

2.2 RAID 레벨 5

RAID5는 [그림2.1]과 같이 구성된다. 이것은 하나의 디스크에 패리티 갱신의 부하가 집중되는 RAID4의 단점을 극복하기 위해서 패리티 블록을 분산 인터리빙(Distributed Interleaving)을 시키고 패리티 갱신에 읽기-수정-쓰기(RMW : Read-Modify-Write) 방식을 사용한다. 이러한 배치는 디스크 읽기 요구의 병렬성 및 디스크 쓰기 요구에 대한 패리티 갱신의 병목 현상을 크게 줄일 수 있고 큰 크기의 읽기/쓰기가 디스크에 일어날 때는 효과적일 수 있으나 작은 크기의 쓰기 작업 일때는 작업 부하에 비해서 많은 크기의 읽기 작업과 쓰기 작업을 행해야 하므로 작업 부하가 커지게 되는 단점이 있다. [그림2.1]은 각 디스크에 점선으로 채워진 패리티 블록이 Left-Symmetric 방식으로 배치된 것을 보여주고 있다.



[그림2.1] RAID 레벨 5

2.3 패리티 배치 방법

패리티를 어떻게 배치하느냐에 따라 성능이 달라지는데 [6]에서 제안한 패리티 배치 방법과 그들을 성능 비교한 것을 살펴보고자 한다. 패리티 스트라이프(parity stripe)란 패리티가 계산되어지는 최소의 스트라이핑 단위의 집합을 의미한다. 아래에 몇 가지의 예를 들어 설명한다.

2.3.1. RAID 레벨 4

RAID4는 [그림2.2]와 같이 패리티 정보를 하나의 특정 디스크에만 저장하는 방식으로 데이

터 스트라이핑 단위 s0, s1, s2, s3과 패리티 스트라이핑 단위 P0가 하나의 패리티 스트라이프가 된다.

s0	s1	s2	s3	P0
s4	s5	s6	s7	P1
s8	s9	s10	s11	P2
s12	s13	s14	s15	P3
s16	s17	s18	s19	P4

[그림2.2] RAID 레벨 4

2.3.2 Right-Asymmetric

RAID0 배치에서 패리티 스트라이핑 단위가 들어갈 수 있도록 수평으로 데이터 스트라이핑 단위를 밀어내는 배치 방식이다. [그림2.3]에서 볼 수 있듯이 각 연속적인 패리티 스트라이프를 보면, 패리티 스트라이핑 단위가 삽입된 지점은 오른쪽으로 하나의 스트라이핑 단위 크기로 회전한다.

P0	s0	s1	s2	s3
s4	P1	s5	s6	s7
s8	s9	P2	s10	s11
s12	s13	s14	P3	s15
s16	s17	s18	s19	P4

[그림2.3] Right-Asymmetric

2.3.3 Extended-Left-Symmetric

[그림2.4]에서 볼 수 있듯이 RAID0 배치에서 패리티 스트라이핑 단위가 들어갈 수 있도록 수직으로 데이터 스트라이핑 단위를 밀어냄으로 구해지는 배치 방법이다. 각 연속적인 패리티 스트라이프를 보면, 패리티 스트라이핑 단위가 위치한 지점은 왼쪽으로 하나의 스트라이핑 단위 크기로 회전하고 있음을 볼 수 있다.

s0	s1	s2	s3	P0
s10	s11	P2	s13	s4
P4	s21	s12	s23	s14
s20	s31	s22	P6	s24
s30	P8	s32	s33	s34

[그림2.4-a]

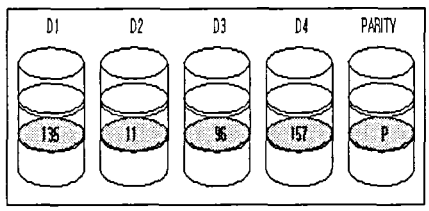
s5	s6	s7	P1	s9
s15	P3	s17	s8	s19
s25	s16	s27	s18	P5
s35	s26	P7	s28	s29
P9	s36	s37	s38	s39

[그림2.4-b]

[그림2.4] Extended-Left-Symmetric

2.4 패리티 계산 방법

RAID3에서 패리티가 결정이 되는 것은 [그림2.5]과 같다. 먼저 그림과 같이 1번 드라이브에 135가 기록되고 나머지 드라이브에 순서대로 11, 96, 157이 기록되어진다고 가정한다.
이들 숫자를 아래 [그림2.6]과 같이 이진수로 계산한 후 짝수 패리티로 계산하여 Bit 7에 1 의 숫자가 짝수이면 패리티는 0, 홀수이면 1을 각각 채워 넣으면 된다.



[그림2.5] 데이터 배치

다음의 결과들은 [6]에서 앞의 8가지 배치 방법들을 통해서 얻은 것이다. 높은 부하(high load)에 있어서 읽기 동작에서는 RAID4 배치가 가장 나쁜 성능을 보였다. 그 이유는 패리티를 여러 디스크에 분산시키지 않았기 때문이다. 그 외의 배치 방법들 간에는 큰 차이를 보이지 않고 있다. 그러므로 높은 부하에 대한 쓰기 동작에 있어서 높은 성능을 얻기 위해서는 패리티를 균등하게 분산시켜 저장하는 것이 바람직하다. 그러므로 어떠한 배치 방법을 선택하느냐는 낮은 부하에 대한 읽기 성능을 중요시 여기느냐 혹은 낮은 부하에 대한 쓰기 성능을 중요시 여기느냐에 따라 달라진다.

Binary Value of Data									
Drive	Data Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
D1	135	1	0	0	0	0	1	1	1
D2	11	0	0	0	0	1	0	1	1
D3	96	0	1	1	0	0	0	0	0
D4	157	1	0	0	1	1	1	0	1
Sum of bits		Even	Odd	Odd	Odd	Even	Even	Even	Odd
Parity values		0	1	1	1	0	0	0	1

[그림2.6] 패리티 계산 방법

III. 읽기-수정-쓰기(Read-Modify-Write)방식

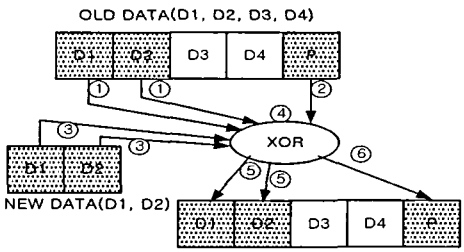
읽기-수정-쓰기 방식은 RAID5에서 패리티 그룹 내의 패리티 갱신을 위해서 사용하는 방식으로 정상적인 작동모드(Normal Mode)와 디스크 장애가 일어났을 경우 복구 모드(Degrade Mode)로 나뉘어져 작동한다.

3.1 정상적인 작동 모드에서 읽기 동작

RAID5의 정상적인 동작 모드에서 읽기 동작의 경우는 해당 디스크 요구를 각각 혹은 병렬적으로 처리하므로 RAID0에서처럼 디스크 효율을 최대한으로 사용할 수 있다.

3.2 정상적인 작동 모드에서 쓰기 동작

RAID5의 정상적인 동작 모드에서 쓰기 동작의 경우는 쓰기 동작이 일어난 블록 외에도 패리티 블록에 대한 갱신 작업을 해주어야 하므로 읽기 동작에 비해서는 효율이 떨어지게 된다. [그림3.1]에서는 이러한 동작이 일어나는 과정을 보여주고 있다



[그림3.1] 읽기-수정-쓰기 방식의 쓰기 동작

3.3. 장애 복구 모드(Degrade Mode)의 동작

디스크 장애가 발생하고 그 디스크에 대한 디스크 요청이 발생했을 경우 장애가 생긴 디스크가 속한 패리티 그룹 내의 패리티 블록과 데이터 블록들을 모두 읽어서 XOR 연산을 한다. 이와 같은 작업은 디스크 이상이 생겨도 시스템이 정지되지 않는다는 장점은 있지만 이것에 소요되는 작업 부하가 상당히 커지므로 시스템 전체에 대한 성능이 저하된다.

3.4. RAID5의 단점 보완을 위한 연구

읽기-수정-쓰기 방식을 사용하는 RAID5의 성능을 향상시키기 위해서 여러 가지 방법이 연구되어 왔다. 이 같은 방법은 크게 두 가지 부류로 나뉠 수 있는데 캐쉬를 이용해서 디스크 접근을 줄여보려는 방법과 쓰기 방법을 바꿈으로 해서 디스크 기록 시간을 단축하려는 방법으로 나눌 수 있다.

3.4.1 캐쉬를 이용한 방법

쓰기 버퍼와 캐쉬은 디스크에 대한 여러 개의 작은 쓰기 동작을 버퍼에 모아 하나의 큰 쓰기 동작으로 만들고 캐쉬를 이용해서 될 수 있으면 디스크에 대한 접근을 줄임으로 해서 작업 부하를 감소시키는 방법이다.

3.4.2 쓰기 방법을 바꾸는 방법

Floating parity는 Piggy back 방식을 패리티 갱신에 이용해서 쓰기 효율을 높인다. Piggy back 방식은 디스크가 원하는 읽기/쓰기 지점에 도달하기 전에 기록할 사항이 있으면 바로 기록을 함으로 해서 회전 지연 시간(Rotation Delay)를 줄여 디스크에 대한 쓰기 지연 시간을 줄이는 방법이다. 패리티 로깅[7]은 패리티에 대한 로그 디스크를 두어서 패리티 동작을 그 디스크에 따로 처리한 후 어느 정도 용량이 되면 한꺼번에 패리티를 갱신하는 방식이다. 이 방식은 작은 패리티 갱신 작업을 모아서 크게 만들 수 있다는 장점은 있지만 패리티 로그 디스크에 이상이 생길 경우 패리티 전체에 대한 신뢰도를 없애버릴 수 있다는 위험성이 있다.

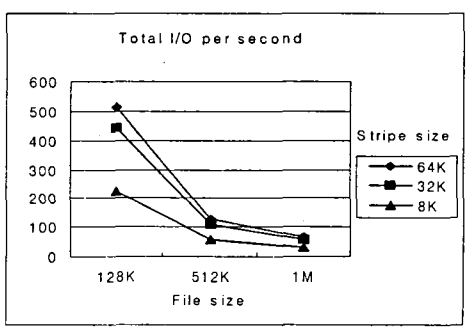
IV. 실험 및 결과

실험은 시뮬레이션 도구인 SMPL이나 RaidSim을 사용하지 않고 Host이1 환경은 CPU는 1GHz, RAM은 524MB, HDDE는IBM 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 128K, 512K, 1M로 가변하였으며 또한 스트라이프 크기도 64K, 32K, 8K로 변환시켜 실험하였다.

4.1 동영상 데이터

4.1.1 쓰기(100 %) 일 때

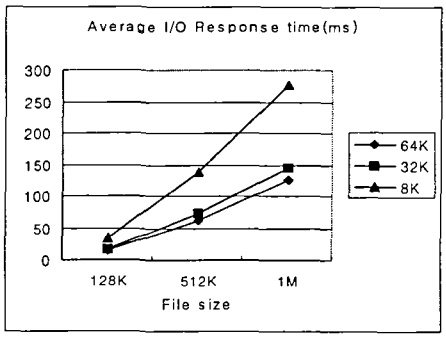
[그림4.1]에서와 같이 동영상 데이터 쓰기 100(%)일 때의 조건에서 스트라이프 크기를 64k, 32k, 8k로 가변하면서 파일 크기에 대한 초당 총 I/O 처리율을 나타낸 것이다. 결과 그래프에서와 같이 읽기보다 쓰기에서 처리율이 적으므로 쓰기가 읽기보다 많은 시간을 필요로 한다는 것을 알 수 있다.



[그림 4.1] 초당 총 I/O 처리율

[그림4.1]의 조건과 같이 실험한 평균 I/O 응답시간을 [그림4.2]에 나타내었다. 실험 결과와 같이 평균 I/O 응답 시간도 읽기보다 많이 소요

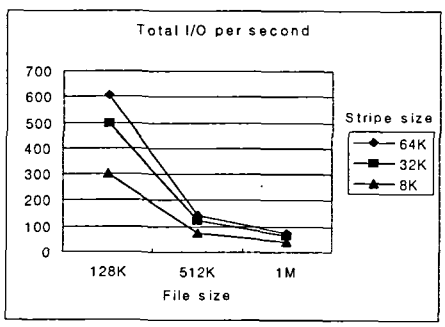
됨을 알 수 있다.



[그림 4.2] 평균 I/O 응답 시간

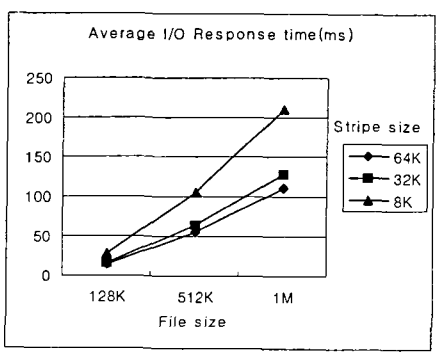
4.1.2 읽기(100 %) 일 때

[그림4.3]에 동영상 데이터 읽기 100(%)일 때의 조건을 쓰기100(%)일 때와 동일하게 한다. 결과 그래프에서와 같이 스트라이프 크기가 클때 초당 처리율이 높다는 것을 알 수 있다.



[그림 4.3] 초당 총 I/O 처리율

[그림4.4]에 평균 I/O 응답 시간 결과 그래프를 나타내었다. 역시 스트라이프 크기와 파일의 크기에 따라 특성이 변화하는 것을 알 수 있다.



[그림 4.4] 평균 I/O 응답 시간

V. 결 론

본 논문에서는 RAID5의 특징을 비교하고 신뢰도와 성능을 향상 시키는 방법에 대한 알고리즘을 설명했다. 결국 I/O 성능을 향상시키기 위해 RAID5를 이용한 병렬처리와 대용량의 2차 저장 장치의 데이터를 손상 없이 고속 전송해야 한다는 것이다. 본 논문에서는 RAID5의 읽기/쓰기에 대하여 파일의 크기, 스트라이프의 크기, 초당 전체 I/O 처리율, 초당 MB 처리율, 평균 I/O 응답시간, 최대 I/O 응답시간 등에 대하여 RAID5의 읽기/쓰기 실험 결과에 의해 쓰기에서 많은 시간이 소요되는 것을 알수 있다. 또한 전체 I/O 시스템의 성능을 좌우하는 것도 알 수 있다.

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] 이상민 외5인 "디스크 배열에서 데이터와 패리티 캐쉬의 관리". 정보과학회 가을 학술발표 논문집 *pp.827-830. 1995.*

[5] Daniel Stodolsky, Mark Holland, "Parity Logging Disk Array" ACM Transactions on Computer System, Col 12. No.3 pp.206-235. August 1994.

[6] E. Lee and R. katz. "Performance Consequences of Parity Placement in Disk Arrays" ASPLOS-4. IEEE. New York. APRIL 1991.

[7] 김근혜, 최항규, "RAID5에서 작은 쓰기 문제 해결을 위한 압축 패리티 로깅 기법" 정보통신 논문지 제3집. 1999.