SDIO에서 RAID 레벨 5의 효율적인 구현

김호진°, 황인철, 맹승렬, 윤현수
한국과학기술원 전산학과
{hojin°, ichwang, maeng, hyoon}@camars.kaist.ac.kr

Efficient Implementation of RAID Level 5 on Single Disk I/O

Hojin Ghim°, In-Chul Hwang, Seungryoul Maeng, Hyunsoo Yoon
Division of Computer Science, Dept. of EECS, Korea Advanced Institute of Science and Technology

요 약

단일 입출력 디스크(SDIO)는 클러스터 환경에서 빠르고 신뢰성있는 분산 저장장치를 제공한다. 단일 입출력 디스크는 주로 RAID 레벨 0이나 레벨 1을 사용하는데, RAID 레벨 5를 사용함으로써 좋은 성능과 좋은 신뢰도, 그리고 높은 디스크 용량 사용률을 얻을 수 있다. 그러나 RAID 레벨 5를 사용함으로써 네트웍 오버헤드 문제, 작은 데이터의 기록 성능 문제, 동시 기록 문제가 발생한다.

위의 세 가지 문제를 해결하거나 최소화하기 위하여 본 논문에서는 Parity Cumulating이라는 접근방법을 제시한다. Parity Cumulating은 패리티의 계산을 두 개의 노드로 분산시킴으로써 네트웍 오버헤드를 줄이고, 패리티를 버퍼에 저장하고 디스크에 작업이 없을 때 처리함으로써 작은 데이터의 기록 성능을 높이며 동시 기록시 일관성을 지킨다.

1. 서 론

클러스터 시스템(Cluster System)[1]은 여러 대의 Personal Computer(PC) 혹은 Workstation을 빠른 네트웍으로 연결하여 병렬로 처리함으로써 비교적 낮은 가격으로 높은 성능을 얻을 수 있는 시스템이다.

클러스터 시스템에서 I/O 하부 시스템의 성능을 높이고 안정성을 보장하기 위하여 Single Disk I/O(SDIO)[2]를 사용한다. SDIO는 클러스터의 각 노드의 디스크들을 하나의 가상 디스크로 사용하여 여러 개의 물리적 디스크에 대해 병렬 I/O를 수행함으로써 성능을 높이고 데이터를 중복되게 저장함으로써 안정성을 얻는다.

기존의 SDIO는 주로 RAID[3] 레벨 0과 레벨 1을 사용한다. 이는 RAID 레벨 0과 레벨 1의 구현이 간단하고 노드간 메시지 전송 오버헤드가 크지 않기 때문이다. 그러나 RAID 레벨 0은 성능이 좋지 중복된 데이터가 전혀 없기 때문에 클러스터 시스템의 규모가 커지면 신뢰도가 크게 낮아진다. RAID 레벨 1은 신뢰도가 높으며 성능도 좋지만 디스크 공간의 50%밖에 사용하지 못하므로 대용량 데이터의 저장을 어렵게 한다.

SDIO에 RAID 레벨 5를 사용하면 기존의 SDIO에 비해 디스크 공간을 효율적으로 사용하면서 동시에 높은 성능과 신뢰도를 얻을 수 있다. 그러나 RAID 레벨 5는 한 블락의 기록 작업에 네 번의 디스크 액세스를 필요로 하기 때문에 작은 데이터의 기록 작업이 현저하게 느리다는 단점을 가지고 있다. 파일에 대한 메타데이터를 유지해야 하는 파일시스템이나 작은 파일을 사용하는 응용 프로그램에서는 작은 데이터의 기록이 빈번히 발생하므로 성능이 크게 저하될 수 있다. 따라서 RAID 레벨 5를 SDIO에 사용하기 위해서는 작은 데이터의 기록에서의 성능을 향상시킬 필요가 있으며, 클러스터 시스템에 알맞게 최적화시켜야 한다.

SDIO에서 RAID 레벨 5를 사용할 때 발생하는 문제점은 Network Overhead 문제, Small Write 문제, Write Sharing 문제의 세 가지로 정리할 수 있다. 본 논문에서는 이 문제점들을 정의하고 문제점들을 해소하거나 최소화하는 방법으로 Parity Cumulating을 제안한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 먼저 관련된 기존 연구를 소개하고, 3장에서는 본 연구의 기초가 되는 SDIO를 설명한다. 4장에서는 SDIO에 RAID 레벨 5를 실제로 적용할 때의 문제점에 대해서 알아보고 해결 방법을 제시한다. 5장에서는 4장에서 제시한 해결 방법이 어떤 성능을 나타내는지를 성능 분석과 실험 결과를 통해 알아보겠다. 마지막으로 6장에서는 본 논문의 결론을 맺고 향후 연구 과제에 대해 기술한다.

2. 관련 연구

2.1 Redundant Array of Distributed Disks (RADD)

RADD[4]는 장애 발생시에도 데이터를 접근할 수 있게 하기 위하여 지역적으로 떨어진 곳에 데이터를 분산시키고 신뢰도와 성능을 높이는 방법이다.

RADD에서는 데이터를 RAID 레벨 5와 같은 방법으로 각 사이트에 분산하고, 하나의 패리티 그룹에 하나의 패리티 블락 외에 또 하나의 예비 블락을 준비하여 한 곳에서 장애가 발생하면 예비 블락을 사용하도록 한다. 따라서 예비 블락을 저장하기 위해 디스크 공간의 낭비가 발생하게 된다. 또한 지역적으로 떨어져 있기 때문에 네트웍 지연시간의 불규칙적으로 발생하므로 각 디스크 작업 요청에 시스템에서 유일한 아이디를 부여한다. 유일한 아이디의 사용을 위하여

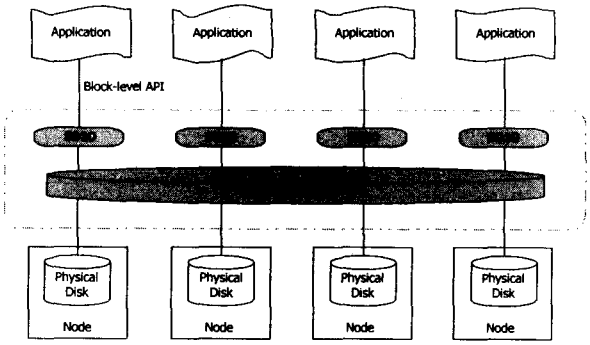

그림 1 SDIO 서비스 개념

사이트들 간의 동기화가 빈번히 일어나므로 그에 따른 오버헤드도 발생하게 된다.

본 논문에서 제시한 Parity Cumulating은 RADD에 비해 디스크 공간의 사용률이 높으며 시스템에서 유일한 아이디를 유지하는 오버헤드가 없으므로 더 좋은 성능을 보일 수 있다.

2.2 RAID-x

RAID-x[5]는 클러스터 환경에서 SDIO를 구현하고 신뢰도와 성능을 높이기 위하여 제안된 방법이다. 여기서는 데이터 블락을 여러 디스크로 분산시키기 위해 Orthogonal Striping and Mirroring(OSM) 방법을 사용한다. 디스크의 앞쪽 반에는 데이터 블락이 RAID 레벨 0과 같은 방법으로 저장되고, 뒤쪽 반에는 미러 블락 디스크별로 순차적으로 저장된다.

이 방법은 Parity Cumulating에 비해 신뢰도가 약간 낮으며 디스크 공간 사용률은 크게 낮다.

3. Single Disk I/O (SDIO)

SDIO[2]는 클러스터 환경에서 여러 노드에 장착되어 있는 디스크들을 하나의 디스크처럼 사용하도록 해주는 SSI 서비스의 일종이다. SDIO는 장치 드라이버 수준에서 구현되어 있어 응용 프로그램에서 하나의 디스크를 사용하듯이 SDIO를 사용하면 데이터가 클러스터 시스템에 흩어져 있는 디스크에 분산되어 저장된다.(그림 1)

이처럼 하나의 가상 디스크를 제공함으로써 응용 프로그램은 데이터의 실제 위치에 신경쓸 필요 없이 전체 클러스터 시스템의 디스크를 모두 사용할 수 있다. 또한 여러 개의 디스크를 병렬 작업에 활용하여 하나의 디스크에서보다 더욱 높은 성능을 얻을 수 있으며, 데이터를 중복 저장하여 필연적으로 발생할 수밖에 없는 디스크의 장애에서도 데이터의 유실을 방지할 수 있다.

SDIO는 장치 드라이버 수준에서 구현되어 있으므로 블락 단위의 Application Programming Interface(API)에서 투명성이 보장되므로 응용 프로그램이 보기에 물리적인 하드디스크와 완전히 같다. 따라서 SDIO에서 제공하는 가상 디스크에 일반적인 디스크에 사용되는 파일시스템을 수정 없이 사용하는 것이 가능하다. 파일시스템을 사용하는 응용 프로그램들도 마찬가지로 소스코드 수정이나 재 컴파일이 필요 없으므로 이진 코드 수준의 호환성이 제공된다.

SDIO는 여러 노드에서 수행중인 여러 응용 프로그램에 의해 동시에 접근될 때의 일관성을 보장하지는 않는다. SDIO는 응용 프로그램에 블락 단위의 API를 제공하는데 블락 단위의 API는 파일 단위의 API와는 달리 한 블락에 대한 작업 요청이 이어지는 다른 작업 요청과 연관성이 없다. 여러 노드에서 여러 응용 프로그램의 동시 접근을 일관성 있게 허용하기 위해서 SDIO 위에 분산 파일시스템을 사용할 수 있다.

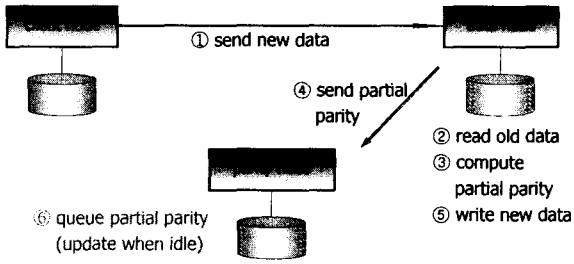

그림 2 Parity Cumulating

4. RAID 레벨 5를 SDIO에 구현할 때의 문제점 및 접근 방법

4.1 RAID 레벨 5를 SDIO에 구현할 때의 문제점

SDIO에서 RAID 레벨 5를 사용할 때에 나타날 수 있는 문제점은 다음의 세 가지로 정리된다.

■Network Overhead 문제 : RAID 레벨 5는 RAID 레벨 0이나 레벨 1에 비해 한 번의 작업을 위해 디스크를 여러 번 접근해야 하므로 네트웍에 의한 시간 지연이 크다. 기본적으로 기존 데이터, 새로운 데이터, 기존 패리티, 새로운 패리티의 전송을 위해 총 네 번의 네트웍 메시지가 필요하다.

■Small Write 문제 : RAID 레벨 5에서 하나의 블락을 기록하려면 패리티의 일관성 유지를 위해 두 번의 읽기와 두 번의 쓰기, 총 네 번의 디스크 접근이 필요하다.

■Write Sharing 문제 : 하나의 기록 작업이 완전히 끝나기 전에 같은 패리티와 관련된 또다른 기록 작업이 시작되면 패리티의 일관성이 깨질 수 있다.

4.2 RAID 레벨 5를 SDIO에 구현할 때의 접근 방법

SDIO에서 RAID 레벨 5를 사용하기 위해서 그림 2에서 설명하는 Parity Cumulating을 사용할 수 있다.

RAID 레벨 5에서의 패리티는 기존 데이터, 새로운 데이터, 기존 패리티를 XOR 연산하여 만들어진다. Parity Cumulating에서는 데이터 노드에서 기존 데이터와 새로운 데이터를 XOR 연산하여 부분 패리티를 생성한 후 부분 패리티를 패리티 노드로 전송한다. 이 때 네트웍 메시지는 두 번 보내지므로 기본 방법에 비해 Network Overhead 문제가 반으로 감소되었다.

보내진 부분 패리티는 패리티 노드의 메모리에 임시 저장된다. 이후 패리티 노드의 디스크에 작업이 없을 때 디스크에 있는 기존 패리티와 메모리의 부분 패리티를 XOR 연산하여 새로운 패리티를 만들어서 디스크에 기록하게 된다. 기존 패리티를 읽고 새로운 패리티를 기록하는 작업이 실제 디스크 작업시간 이후에 이루어지므로 Small Write 문제는 크게 줄어든다.

부분 패리티를 메모리에 저장할 때 같은 패리티 블락에 대한 부분 패리티가 이미 메모리에 있으면 두 개의 부분 패

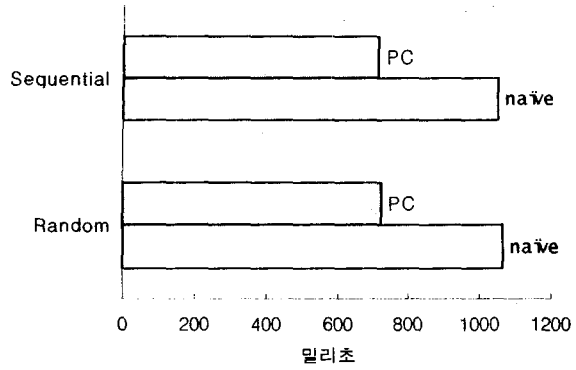

그림 3 작업 소요시간

리티를 XOR 연산하여 하나의 부분 패리티로 합친다. 이렇게 중첩된 부분 패리티를 기존 패리티와 XOR 연산하면 두 개의 부분 패리티를 각각 연산하였을 때와 같은 결과를 얻을 수 있다. 이렇게 부분 패리티를 중첩시킴으로써 부분 패리티 저장으로 인한 메모리 사용을 줄일 수 있으며, 또한 부분 패리티의 저장 시점에 관계없이 패리티의 일관성을 유지할 수 있으므로 Write Sharing 문제가 해결된다.

5. 분석과 실험에 의한 성능 평가

5.1 분석에 의한 성능 평가

RAID 레벨 5의 기록 작업에 가장 기본적인 방법을 사용할 때 한 블락 기록 작업 수행시간은 다음과 같다.

$$4M+X+4P$$

여기서 N은 디스크 개수 즉 클러스터 노드의 수이며, M은 요청메시지와 응답메시지의 전송에 소요된 시간, X는 XOR 연산 시간, P는 디스크 액세스 시간이다.

Parity Cumulating의 한 블락 기록 작업 수행시간은 다음과 같다.

$$(2-\frac{1}{N-1})M+(1+s)X+2P$$

여기서 s는 부분 패리티가 중복될 확률이다.

대부분의 클러스터 시스템에서 N은 1보다 훨씬 크고, s는 0에 가까우므로 Parity Cumulating의 성능은 기본 방법에 비해 크게 좋아졌다.

5.2 실험에 의한 성능 평가

실험 환경은 다음과 같은 노드 4개로 이루어진 클러스터 시스템이다.

CPU	Intel Pentium IV 1.8GHz
Memory	512MB
Hard disk	IBM 60GB
OS	Linux (kernel 2.4.20-8)
Disk access time(1KB)	1112ms
Network transfer time(1KB)	74.5ms

먼저 세 개의 블락에 해당하는 크기의 데이터를 디스크의 무작위로 선택된 위치에 기록하는 응용 프로그램을 네 개의 노드에서 동시에 실행시켜 네 개의 응용 프로그램이 SDIO 장치 파일에 기록한 총 시간을 모두 더한 결과를 측정하였다. 다음으로 같은 형태의 실험을 연속된 세 블락 크기의 데이터가 아닌 무작위 위치의 세 블락의 데이터를 기록하는 시간을 측정하였다.

기본적인 방법에서 연속 데이터는 평균 1052ms, 무작위 데이터는 평균 1063ms가 소요되었다. 반면 Parity Cumulating을 사용했을 때 연속 데이터가 평균 715ms, 무작위 데이터는 평균 725ms로 각각 32%, 32%의 성능 향상을 나타냈다. 이 결과는 그림 3에 나타나 있다.

6. 결론 및 향후 과제

클러스터 시스템[1]에서 Single Disk I/O(SDIO)[2]는 I/O 작업에서 사용자의 편의성과 높은 성능, 데이터의 안정성을 제공한다. 본 논문에서는 SDIO에서 RAID[3] 레벨 5를 효과적으로 사용하는 방법을 제공함으로써 기존에 사용되던 RAID 레벨 0이나 레벨 1에 비해 보다 높은 성능과 안정성, 그리고 높은 디스크 공간 사용률을 얻을 수 있게 한다.

이를 위하여 Network Overhead 문제, Small Write 문제, Write Sharing 문제를 정의하고, Parity Cumulating을 사용하여 위의 세 문제를 어떻게 해소 혹은 최소화하는지를 보인다.

실험 결과 Parity Cumulating은 기본적인 방법에 비해 약 32%의 성능 향상을 보였다.

향후 계속될 수 있는 과제로는 RAID 레벨 5의 성능을 높이기 위해 기존에 제안된 여러 가지 기법들을 SDIO에 적용해 보고 실제 클러스터 환경에 맞도록 개선하는 것이 있다. 디스크의 수가 수십 개에 달하는 대교모의 클러스터에서의 RAID 레벨 5의 성능 병화를 알아보고 이 경우에 성능을 최적화하는 방법을 연구해볼 수도 있다. 또한 RAID 레벨 5를 사용하는 SDIO가 클러스터에 노드나 디스크가 추가되고 제거되는 상황에서도 유연하게 대처할 수 있는 방안을 마련하는 과제를 수행할 수 있다.

1) R. Buyya, "High Performance Cluster Computing: Architectures and Systems, Volume 1", Prentice Hall PTR, 1999
2) 황인철, 김동환, 김호진, 맹승렬, 조정완, "단일 디스크 입출력을 위한 커널 모듈 프로토타입의 설계 및 구현", 한국정보과학회 2003년도 추계학술발표논문집, 2003
3) D. A. Patterson, G. A. Gibson and R. Katz, "A Case for Redundant Arrays of Inexpensive Disks", SIGMOD International Conf. on Data Management, pp109-106, 1988
4) M. Stonebraker and G.A. Schloss, "Distributed RAID—A New Multiple Copy Algorithm", In Proceedings of the Sixth International Conf. on Data Engineering, pp 430-437, 1990
5) K. Hwang, H. Jin and R. Ho, "RAID-x: A New Distributed Dist Array for I/O-Centric Cluster Computing", In Proceedings of 9th IEEE International Symp. on High Performance Distributed Computing, pp279-286, 2000