제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

스트라이핑 기반의 병렬 접근 미러링 기법

강동재*, 김창수*, 신범주**, 김학영*
*한국전자통신연구원 컴퓨터시스템연구부
**밀양대학교 컴퓨터공학과
e-mail : djkang@etri.re.kr

Parallel Accessed Mirroring based on Stripping

Dong-Jae Kang*, Chang-Soo Kim*, Bum-Joo Shin**, Hag-Young Kim*
*Computer System Dept, Computer Software Research Lab, ETRI
**Dept of Computer Engineering, Miryang National University

요 약

멀티미디어와 인터넷의 대중화가 야기한 급격한 데이터의 증가는 테라(Tera)바이트 이상의 대용량 저장공간과 대용량 정보의 효율적인 공유를 지원하는 스토리지 시스템을 요구하고 있으며 이를 위하여 SAN 기반의 스토리지 클러스터링 시스템들이 많이 사용되고 있다. 이러한 환경에서 하드웨어 또는 소프트웨어 RAID(Redundant Array of Independent Disks)는 대용량 정보의 고성능의 입출력과 신뢰성을 위해서 필수적이 되었다.

범용적인 RAID로는 RAID-0, RAID-1, RAID-5가 주로 사용되고 있으며 각각의 레벨은 장단점을 갖는다. 본 논문에서는 RAID-0와 RAID-1이 갖는 문제점들의 보완을 위하여 변형된 RAID 레벨인 RAID-SM을 제안한다. RAID-SM은 기존의 RAID-1이 가지는 데이터의 가용성을 유지하면서 추가적인 비용 없이 RAID-0의 우수한 입출력 성능을 얻기 위한 RAID-1의 변형된 방식이다. RAID-SM의 구현을 위하여 디스크상의 데이터의 배치 및 데이터 맵핑 방식을 정의하고 RAID-SM에서의 I/O방법을 기술한다. 제안하는 RAID-SM은 멀티미디어나 GIS 데이터와 같은 읽기 연산 집약적인 시스템을 대상으로 하는 안정적인 레이드 방식이며 RAID-SM의 장점 및 성능은 본 논문에서의 실험을 통한 결과로서 제시한다.

1. 서론

인터넷이 대중화됨에 따라 데이터의 폭발적인 증가를 수용하기 위한 스토리지 시스템은 테라(Tera)바이트 이상의 대용량 저장공간을 지원하기 위하여 SAN 기반의 스토리지 클러스터링을 기반으로 구축하는 것이 효과적이다. 또한 대용량 정보의 효율적인 공유를 위하여 고성능의 입출력과 저장 데이타의 신뢰성이 보장되어야 한다. 이를 위하여 SAN 환경에서는 레이드(RAID : Redundant Array of Independant Disks)를 기반으로 한 논리 디스크(Logical Disk)가 주로 사용하며 SAN상의 물리적인 디스크들을 적용 업무에 맞는 레이드 레벨로 묶고 이를 가상화(Storage Virtualization)함으로서 지원된다. 레이드는 스토리지에 저장된 데이터를 보다 빠르고 안정적으로 접근, 관리하기 위한 방식으로 각각의 레벨은 디스크 오류시 데이터에 대한 접근 불능이나 데이터의 손실에 대한 가용성을 지원하기 위한 정책과 가상 디스크의 데이터 블록들을 물리적인 디스크로 연결하는 주소 연결 알고리즘으로 정의된다. RAID-0(stripping)의 경우는 중복 데이터를 갖지 않으므로 추가적인 디스크 비용이 없고 I/O 성능이 뛰어나다는 장점을 갖으며 동시에, 중복 데이터가 없으므로 RAID-0을 구성하는 디스크 중 하나 이상의 디스크에서 오류가 발생하면 데이터의 가용성 지원 및 복구가 불가능하다는 단점을 갖는다. RAID1(mirroring)의 경우는 중복 데이터를 위한 디스크 비용이 크지만 레이드 레벨 중에서 가장 우수한 데이터 가용성 및 복구 기능을 지원할 수 있는 레벨이며 읽기 요청에 대해서는 각각의 중복 데이터들에 대하여 독립적인 접근이 가능하므로 읽기 연산에 대한 부하 분산을 지원할 수 있다. 하지만 데이터의 가용성을 지원하기 위해서 중복 데이터들 사이의 일관성이 보장되어야 하며 쓰기 연산시에 모든 중복 데이터에 대한 원자적 갱신이 이루어짐을 보장해야 한다. 앞에서 언급한 바와 같이 RAID-0, 1을 포함한 6개의 Berkely 레이드 레벨들은 모두 장, 단점을 동시에 갖으며 적용하려는 어플리케이션에 맞는 레벨의 선택이 요구된다. 이상적인 레이드 시스

템은 데이터 가용성 및 복구 능력이 뛰어난 RAID-1의 특징을 가지면서 RAID-0와 같이 우수한 I/O 성능을 갖는 시스템이다. 따라서 RAID-1과 RAID-0를 결합한 형태인 RAID1+0도 많이 사용되지만 RAID-0을 구성하는 논리디스크들 만큼 추가적인 디스크들이 요구되므로 디스크 비용이 크다는 문제점을 갖는다.

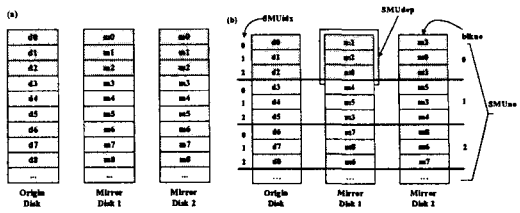
따라서 본 논문에서는 데이터 가용성 및 복구 기능이 뛰어난 RAID-1에서 추가적인 디스크 비용이 없이 RAID-0의 우수한 I/O 성능을 보완하기 위한 *RAID-SM (Stripped-Mirroring)*이라는 변형된 RAID-1방식을 제안하며 자세한 내용은 본론에서 기술하도록 한다.

2. RAID-SM(*Stripped-Mirroring*)

본 장에서는 제안하는 *RAID-SM(stripped - mirroring)*에 대하여 살펴본다. 디스크 상의 데이터 배치 및 맵핑 방식을 중심으로 *RAID-SM*에서의 데이터 입출력 방식을 기술한다.

2.1 데이터 블록의 배치

RAID-1(mirroring)에서 RAID-0(stripping)의 우수한 I/O 성능을 얻기 위하여 본 논문에서 제안하는 *RAID-SM(stripped-mirroring)*의 데이터 블록 배치는 [그림 1]의 (b)와 같으며 [그림 1]의 (a)는 일반적인 RAID-1의 데이터 블록의 배치를 나타낸다. [그림 1]에서 d는 Original Disk의 데이터를, m은 Mirrored Disk의 데이터를 나타내고 영문 소문자 뒤의 번호는 데이터 블록의 번호를 나타낸다. 따라서 동일한 번호를 갖는 블록들은 미러링의 중복 데이터(redundant data)이며 [그림 1]에서는 3개의 디스크가 미러링을 구성하는 예를 나타낸다. 기존 RAID-1의 경우, [그림 1]의 (a)에서 보는 바와 같이 중복 데이터들이 RAID-1을 구성하는 각각의 디스크들에서 동일한 데이터 블록의 위치를 갖지만 *RAID-SM*에서는 각 디스크마다 모두 다른 블록 위치와 배치 방식을 갖는다. *RAID-SM*은 RAID-1과 동일한 디스크 용량을 요구하면서 RAID-0의 우수한 I/O 성능을 얻기 위한 레이드의 구성 방식이다. *RAID-SM*에서는 기존의 RAID-0의 stripping unit과 유사한 *SMU(Stripped-Mirroring Unit)*라는 데이터 블록들의 모임이 존재하며 *SMU*는 연속적인 데이터 블록들의 모임이다. *SMU*를 구성하는 데이터 블록의 수는 *RAID-SM*을 구성하는 디스크들의 수와 동일한 값을 갖는다. *SMU*들은 RAID-0에서의 stripping unit number와 유사한 디스크내의 순서값을 갖으며 이를 *SMUno*라고 한다. *RAID-SM*을 구성하는 각각의 디스크에서 *SMUno*가 같은 *SMU*들은 동일 블록들로 구성되어지며 각 *SMU*내에서의 블록들은 서로 다른 배치를 갖는다. *SMU*를 구성하는 각각의 블록들 역시 *SMU*내에서의 순서를 갖으며 *SMUidx*라는 용어를 사용한다. *RAID-SM*을 구성하는 디스크들에서 동일 SMU의 동일 SMUidx를 접근하면 SMU를 구성하는 블록수 만큼 연속된 블록 그룹을 접근할 수 있도록 한다.



[그림 1] RAID-1과 *RAID-SM*의 데이터 블록 배치
((a)RAID-1의 블록 배치 (b)*RAID-SM*의 블록 배치)

*RAID-SM*을 구성하는 디스크 중에서 원본(Original) 디스크로 선정된 disk0의 경우는 기존 디스크상의 블록들과 동일한 배치 방식을 가지며 두 번째 디스크부터 새로운 디스크 배치 방식의 적용을 받게 된다. 새로운 배치 방식을 갖는 *RAID-SM* 에서의 블록들은 SMU 단위로 배치가 이루어지기 때문에 중복 데이터들은 동일한 SMU값을 가지고 SMUidx는 서로 다른 값을 갖게 된다. 이러한 배치 방식은 결과적으로 *RAID-SM*을 구성하는 각 디스크에 존재하는 동일한 *SMUno*의 동일한 *SMUidx*를 접근함으로서 순차적인 디스크 블록들의 접근을 가능하게 하여 RAID-0과 같이 *SMU* 크기 만큼의 RAID-1 데이터의 병렬 접근(Parallel Access)을 허용한다.

2.2 *RAID-SM*의 데이터 맵핑

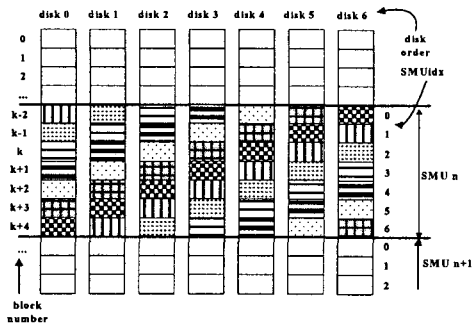
[그림 1]에서와 같이 *RAID-SM*은 기존의 RAID-1과 다른 데이터 배치 방법을 사용하고 있으므로 그에 적합한 새로운 주소 연결 방식(mapping)이 요구된다. RAID-1은 동일한 중복 데이터를 접근하기 위해서 각 디스크의 동일한 데이터 블록을 접근하지만 *RAID-SM*의 경우, 각 디스크마다 데이터 블록의 위치가 다르므로 중복 데이터를 접근하기 위한 대상 데이터 블록의 위치가 디스크마다 다른 값을 갖게 된다. 따라서 *RAID-SM*을 구성하는 디스크들 중에서 특정 디스크에서의 해당 블록 위치를 찾을 수 있는 새로운 주소 연결식의 정의가 요구된다.

[그림 2]는 *RAID-SM*을 구성하는 디스크에서 중복 데이터(redundant data) 블록의 위치를 일반화한 그림이다.

[그림 2]에서의 동일한 무늬의 블록들은 *RAID-SM*를 구성하는 디스크들에서 임의의 블록 번호 n을 갖는 동일한 중복 데이터 블록들의 위치이다. 원본 디스크에서의 임의의 블록 번호 k가 주어지는 경우, *RAID-SM*를 구성하는 각각의 디스크에서 중복 데이터 블록의 실제 디스크 블록의 주소를 구하는 식을 정리해보자.

*SMU*의 크기는 *RAID-SM*을 구성하는 디스크들의 개수와 동일한 크기를 가지므로 N으로 모두 동일하다. 원본 데이터 디스크에서 주소 k를 갖는 블록을 block(k), k 블록의 *SMUidx*를 SMUidx(k)라고 하고 [그림 2]에서 순서

값 i를 갖는 디스크의 order값을 order(i)라고 하자.



[그림 2] *RAID-SM*에서의 중복 데이터의 배치 방식

그러면, order(i)가 SMUidx(k)보다 작거나 같은 경우는 디스크 i에서 블럭 SMUidx(k) - order(i)에 원본 디스크의 block(k)와 동일한 중복 데이터가 존재하고 디스크 order가 SMUidx(k)보다 크면 SMU의 크기 N에서 order(i) - *SMUidx(k)*의 값 만큼 감소한 위치에 배치된다.

```
calculating the address of k-th block of original disk at disk i

  SMUno = k / SMU;
  SMUidx = k % SMU;
  SMUfirst = SMUno * SMU;
  if( SMUidx(k) => order(i) )
        SMUidx(k(i)) = SMUidx(k) - order(i);
  else
        SMUidx(k(i)) = SMU - (order(i) -
  SMUidx(k));
  addr(k(i)) = SMUfirst + SMUidx(k(i));
```

[알고리즘 1] Finding a address of block k in the disk i

이러한 맵핑 규칙을 알고리즘으로 표현하면 [알고리즘 1]과 같다. *SMU*는 *SMU*를 구성하는 데이터 블록의 수 N로 크기를 나타내고 SMUfirst는 특정 SMU에서의 첫 번째 블록이 갖는 주소라고 하자. 또한, 디스크 i에서 블록 k 가 가지는 SMUidx를 SMUidx(k(i)), 디스크 i에서 구하려는 블록 k의 주소를 addr(k(i))라고 하면 addr(k(i))는 *SMUno*와 *SMU*내에서의 인덱스를 구함으로서 찾을 수 있다.

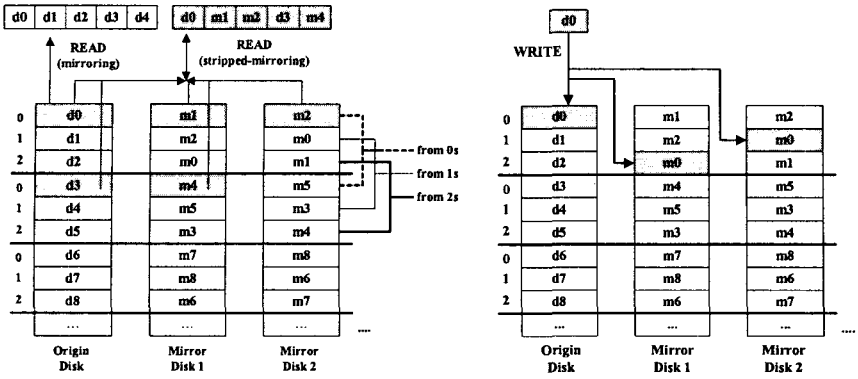
2.3. *RAID-SM*에서의 입출력

*RAID-SM(Stripped-Mirroring)*은 최대의 데이터 가용성을 지원하는 레이드 레벨인 미러링에서 우수한 I/O 성능을 보장하는 스트라이핑(stripping)의 장점을 이용하기 위한 방식이다. 이를 위하여 앞에서 디스크에서의 데이터 배치 및 데이터 맵핑을 정의하였다. 본 장에서는 *RAID-SM*의 입출력 방식에 대하여 설명하도록 한다.

*RAID-SM*은 대용량의 데이터 접근이나 다중 호스트의 환경에서 빠른 입출력을 보장하기 위한 방식으로 대부분의 멀티미디어 데이터와 같이 갱신 연산이 적고 읽기 연산 집약적인 시스템에 적합하다.

기존의 RAID-1에서는 호스트들이 데이터에 대한 접근 요청을 하는 경우, 스토리지의 부하를 분산시키기 위하여 RAID-1을 구성하는 스토리지 중에서 현재 사용하지 않는 스토리지에 I/O 요청을 할당하는 방식을 사용한다. 하지만 이는 임의 호스트가 할당된 하나의 스토리지로부터 요구되는 대용량의 데이터를 모두 읽어야 하므로 Linear 레벨의 성능을 얻을 수 있다. 즉 [그림 3]에서와 같이 일반 미러링에서는 할당된 스토리지가 Original Disk인 경우, 그로부터 요구한 데이터인 d0~d4까지를 접근하게 된다. 하지만 *RAID-SM*의 경우는 요구하는 데이터를 *RAID-SM*을 구성하는 모든 디스크로부터 동시에 접근하여 읽게 되므로 스트라이핑 방식의 입출력 성능을 유지하는 것이 가능하다. 또한 여러 호스트가 동시에 동일한 데이터를 접근하는 경우, RAID-1에서는 하나의 스토리지를 하나의 호스트에 할당함으로서 입출력의 성능을 개선하지만, *RAID-SM*은 [그림 3]와 같이 *RAID-SM*을 구성하는 단위인 *SMU*의 *SMUidx*를 각각 할당함으로서 다중 호스트의 요구에 대해서도 RAID-0의 성능을 지원할 수 있다는 장점을 갖는다. 할당할 수 있는 *SMUidx*의 수는 미러링을 구성하는 디스크의 개수와 동일하게 구성되어 있으므로 RAID-1에서 지원할 수 있는 동시 호스트의 수와 동일한 수를 지원할 수 있다. 즉 RAID-1인 미러링에서는 [그림 3]에서와 같이 3개의 디스크로 구성이 된 경우, 접근을 요구하는 호스트에 대하여 임의의 디스크를 할당 할 수 있다. *RAID-SM*에서는 SMUidx를 할당 함으로서 [그림 3]의 *from 0s, from 1s, form 2s*들로부터 각각 동일한 데이터들을 접근 할 수 있다. 읽기 연산은 요구되는 데이터 양만큼을 *SMUno*를 증가시키면서 동일한 *SMUidx*의 데이터를 스트라이핑과 동일한 방식으로 접근한다.

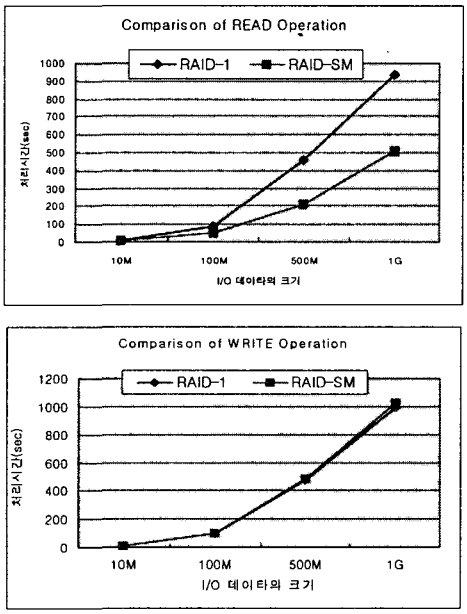
*RAID-SM*에 대한 갱신 연산 및 쓰기 연산은 2.2장에서 정의한 주소 연결 관리 알고리즘을 사용하여 접근한다. 상위에서 쓰기 및 갱신에 대한 데이터 주소는 Original Disk에 대한 주소로 요청을 하게 되므로 동시에 원자적(Atomic)으로 연산이 발생하여야 하는 Mirrored Disk들에 대한 데이터의 갱신은 [알고리즘 1]에 의하여 대상 데이터를 찾은 후에 연산을 수행한다. 쓰기 연산에 대한 전체적인 처리 과정은 RAID-1에서의 처리 과정과 동일하다. 하지만 [그림 3]에서 보는 바와 같이 각 디스크마다 동일한 중복 데이터들의 디스크 주소가 다르므로 [알고리즘 1]에 의하여 정확한 위치를 산출하는 과정이 요구된다.



[그림 3] *RAID-SM*에서의 입출력 방식

3. 실 험

본 장에서는 본론에서 제안한 *RAID-SM*의 성능을 실험을 통하여 제시한다.



[그림 4] RAID-1 과 *RAID-SM*의 R/W성능 비교

실험은 기존의 RAID-1의 성능과 제안하는 *RAID-SM*의 성능을 비교하며 [그림 4]는 읽기와 쓰기 연산에 대한 처리 성능을 비교한 도표이다. 실험은 3개의 디스크가 하나의 가상 디스크를 구성하는 환경에서 실시한 결과이다.

[그림 4]에서 보는 바와 같이, *RAID-SM*의 경우는 읽기에서 기존 RAID-1보다 30%이상의 월등한 성능 향상을 보임을 알수 있다. 쓰기 연산에서는 RAID-1과 비슷한 성능을 유지하지만 주소 연산으로 인한 다소의 성능 감소를 볼 수 있다.

4. 결 론

본 논문에서는 RAID-0의 우수한 성능과 RAID-1의 데이터 가용성이라는 장점을 동시에 적용할 수 있는 변형된 RAID 레벨인 *RAID-SM*을 제안하였다. *RAID-SM*은 읽기 연산 집약적인 시스템을 위한 RAID 레벨로서 이를 위하여 본론에서 *RAID-SM*의 구현을 위한 디스크상의 데이터 배치 및 데이터 맵핑 방식을 정의하고 *RAID-SM*에서의 I/O방법을 기술하였다. *RAID-SM*은 GIS 데이타나 멀티미디어 데이터와 같이 READ연산이 많은 시스템에서 좋은 I/O 성능을 유지할 수 있으며 WRITE 연산이 집약적인 시스템에서는 성능이 다소 떨어질 수 있다는 단점이 있다. 따라서, 향후에는 *RAID-SM*에서의 WRITE / UPDATE 연산에서의 성능을 개선하기 위한 연구가 필요하며 안정적인 스토리지 시스템을 위한 *RAID-SM*의 메타 데이터를 처리하는데 요구되는 시간의 단축을 위한 연구가 계속 진행 되어야한다.

참고문헌

[1] Paul Massiglia, "The RAID book" 6th edition, RAID Advisory Board, 1997

[2] Kai Hwang, Hai Jin, Roy Ho, "RAID-x: A New Distributed Disk Array for I/O-Centric Cluster Computing", High-Performance Distributed Comput ing, 2000. Proceedings, 2000, pp 279-286

[3] Xavier Caron, Jorg Kienzle, and Alfred Strohmeier, "Object-Oriented Stable Storage Based on Mirroring", Ada-Europe 2001, LNCS 2043, pp.278-289, 2001

[5] Hai Jin, Kai hwang, Jiangling Zhang, "A RAID Reconfiguration Scheme for Gracefully Degraded Operations" , Parallel and Distributed Processing, 1999. PDP '99. Proceedings of the Seventh Euromicro Workshop, 1999, pp.66-73