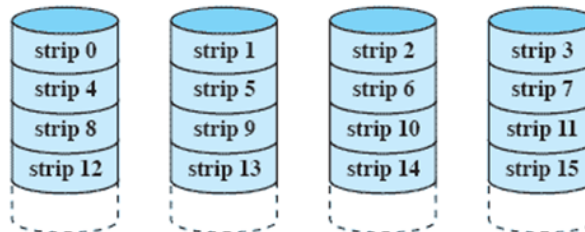


RAID

- 다중 디스크를 사용하여 다양한 방식으로 데이터를 구성할 수 있고 신뢰성을 향상시키기 위한 중복성을 여러 방법으로 추가
- RAID(Redundant Array of Independent Disks)
- RAID 방식은 0~6까지 7개의 레벨로 구성
- 이 레벨들은 계층적 관계는 없지만 다음에서 보여주는 3가지 공통된 특성을 공유하는 서로 다른 설계 구조가 있음
 1. RAID는 운영체제에 의해서 단일 논리적인 드라이브로 판단되는 물리적 디스크 드라이브의 집합이다. → 실제 디스크는 여러개
but OS에 의해서
하나의 논리 디스크로 보임
 2. 데이터는 배열의 물리적 드라이브에 스트라이핑(striping)이라는 기법을 사용하여 분산
 3. 중복된 디스크 용량은 패리티 정보를 저장하기 위해 사용되는데, 이것은 디스크 실패가 발생했을 때 데이터 복구를 보장
- 두 번째와 세 번째의 특성의 세부 사항들은 RAID 레벨별로 차이가 있음
- RAID 0과 RAID 1은 세 번째 특성을 지원하지 않음
- RAID란 용어는 원래 버클리의 캘리포니아 대학의 연구원들이 [PATT88]이란 논문에서 처음으로 사용

RAID 레벨 0

- 성능 향상을 위한 중복 데이터 포함하고 있지 않기 때문에 RAID 레벨 0은 진정한 RAID 군에 속하지 않음
- 하지만 슈퍼컴퓨터 상의 어떤 응용들은 성능과 용량이 주된 관심사이고 신뢰성 향상보다는 저비용을 더 중요시 됨
- RAID 0에서 사용자 데이터와 시스템 데이터는 배열의 모든 디스크에 골고루 분산
- 이 데이터의 분산은 하나의 대용량 디스크를 사용하는 것보다 탁월한 이점 있음
- 서로 다른 데이터 블록을 요구하는 두 개의 입출력 요청이 대기 중이라면, 요청된 블록들이 다른 디스크 상에 있을 가능성이 큼
- 따라서 두 개의 입출력 명령을 동시에 내릴 수 있어 입출력 큐 대기 시간을 줄이게 된다.
- 그러나 모든 RAID 레벨의 경우처럼 RAID 0은 단순히 데이터를 디스크 배열에 분산시키는 것 이상의 배치를 수행
- 데이터는 스트립(strip)이라는 단위로 나뉘어져 이용 가능한 디스크들에 분산

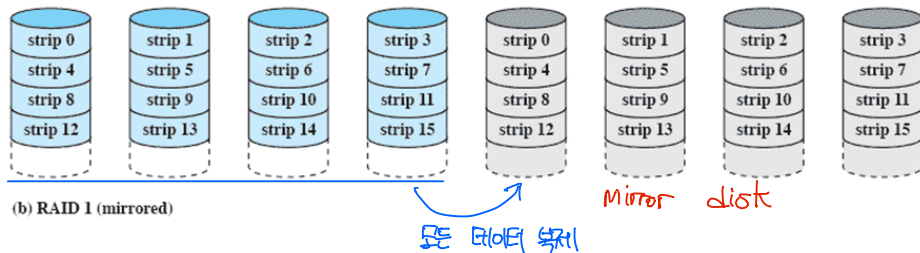


(a) RAID 0 (non-redundant)

- 모든 사용자 데이터와 시스템 데이터가 하나의 논리적 디스크에 저장
- 이 논리적 디스크는 스트립으로 나누어짐
- 스트립은 물리적 블록이나 섹터, 혹은 다른 단위일 수도 있음
- 이 스트립은 라운드-로빈 방식으로 RAID 배열의 연속적인 물리적 디스크에 매핑
- 각 원소 스트립들이 각기 다른 물리 디스크에 매핑 되고 논리적으로는 연속적인 스트립의 집합을 스트라이프(stripe)라고 부름
- n개의 디스크를 가진 디스크 배열에서 첫 n 개의 논리적 스트립들은 n 개의 물리 디스크의 첫 번째 스트립들에 각각 저장되며, 첫 번째 스트라이프를 형성 4개의 disk → 첫 4개의 논리 strip들은 첫 번째 스트라이프에 저장 (0, 1, 2, 3)
- 두 번째 n 개 스트립들은 각 디스크의 두 번째 스트립들로 분산
- 나머지도 같은 방식으로 분산
- 하나의 입출력 요청이 여러 개의 논리적으로 연속된 스트립들을 접근할 때, n 개까지의 스트립들이 동시에 처리될 수 있어 입출력 전송 시간을 감소시키는 장점이 있음

RAID 레벨 1

- RAID 1은 중복을 획득하는 방법 면에서 RAID 2에서 6까지의 레벨들과 차이가 있음
- RAID 1을 뺀 나머지 RAID의 구성에서는 중복을 도입하는데 특정 형식의 패리티 계산이 사용되는 반면, RAID 1의 중복은 모든 데이터를 복사하는 간단한 방법을 사용



- RAID 0에서처럼 데이터 스트라이핑이 사용
- RAID 1의 경우 각 논리적 스트립은 두 개의 별도의 물리적 디스크에 매핑 되어 배열의 모든 디스크는 동일한 데이터를 저장하는 미러 디스크(mirror disk)를 갖게 됨
- 비록 드문 경우지만 RAID 1은 데이터 스트라이핑 없이 구현될 수도 있음

RAID 1 장점

1. 읽기 요청은 요청된 데이터를 포함하는 두 개의 디스크 중 탐색시간과 회전 지연 시간의 합이 최소인 디스크에서 서비스될 수 있음
 2. 쓰기 요청은 서로 대응하는 스트립들 모두 갱신해야 하지만 이 작업은 병렬적으로 행해질 수 있음. 따라서 쓰기 성능은 두 개의 쓰기 중에서 더 느린 것으로 결정
- ✓ 즉 탐색시간과 회전 지연 시간의 합이 더 큰 것에 영향을 받음
- ✓ RAID 1에는 쓰기에 의한 불이익이 없음
3. 실패 시 복구 작업이 간단
- ✓ 드라이브가 실패하면 그 드라이브에 저장된 데이터는 두 번째 드라이브에서 접근가능하다.

$$2^P \geq d + p + 1$$

RAID 1 단점

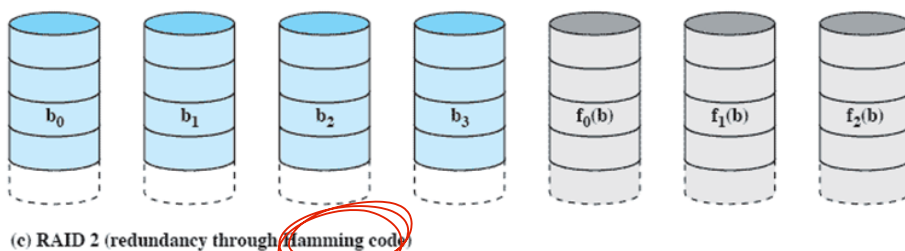
- 비용
- 논리적 디스크 크기의 두 배의 공간을 필요

최소 parity bit

$$2^P \geq d + p + 1$$

d: 디스크 수
p: 패리티 수

RAID 레벨 2

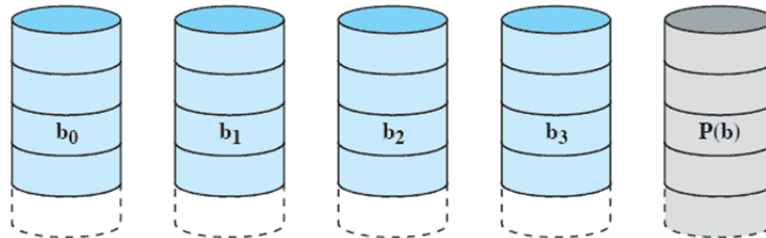


d=4이므로
최소 p: 3
 $2^P \geq 5 + p$

- RAID 2와 3 레벨은 병렬 접근 기술을 사용
- 병렬 접근 배열에서 모든 구성 디스크들은 매 입출력 요청 실행에 참여
- 일반적으로 각 디스크 헤드가 주어진 시간에 각 디스크의 동일한 위치에 있도록 개별 드라이브들의 축은 동기화됨
- 다른 RAID 구성에서처럼, 데이터 스트라이핑이 사용
- RAID 2와 3에서는 작은 스트립 사용 - 한 바이트 또는 한 워드 정도
- RAID 2에서는 각 데이터 디스크 상의 상응하는 비트들에 대해 에러-교정 코드가 계산
- 이 코드의 비트들은 다중 패리티 디스크 상의 상응하는 비트 위치에 저장
- 일반적으로 해밍 코드가 사용

- 이 코드는 단일 비트 에러들을 교정하고 이중 비트 에러들을 검출할 수 있음
- 장점
- 비용
- RAID 2는 많은 디스크 에러가 발생하는 환경에서 효과적
- 개별 디스크와 디스크 드라이브가 높은 신뢰성을 갖게 되면, RAID 2는 필요 이상의 선택이며 구현할 필요가 없음

RAID 레벨 3



(d) RAID 3 (bit-interleaved parity)

- RAID 3은 RAID 2와 비슷한 방식의 구조임
- 다른 점은 RAID 3이 디스크 배열의 크기에 상관없이 오직 하나의 중복 디스크만을 필요로 한다는 것
- RAID 3은 작은 스트립들로 데이터를 분산시키고 데이터를 병렬로 접근
- RAID 3에서는 에러 교정 코드 대신에, 하나의 단순한 패리티 비트가 모든 데이터 디스크 상의 동일한 위치에 있는 각 비트들의 집합에 대해 계산
- 중복(redundancy)
 - 드라이브 실패가 발생하면 패리티 드라이브가 접근되고 나머지 장치들로부터 데이터가 재구성
 - 실패한 드라이브가 교체되면 손실 데이터는 새로운 드라이브에 복구되어 동작이 재개
 - 데이터 재구성
 - ✓ X0부터 X3이 데이터를 저장하고 X4가 패리티 디스크인 다섯 개의 드라이브를 가진 디스크 배열 가정
 - ✓ i 번째 비트의 패리티는 다음과 같이 계산

$$X4(i) = X3(i) \oplus X2(i) \oplus X1(i) \oplus X0(i)$$

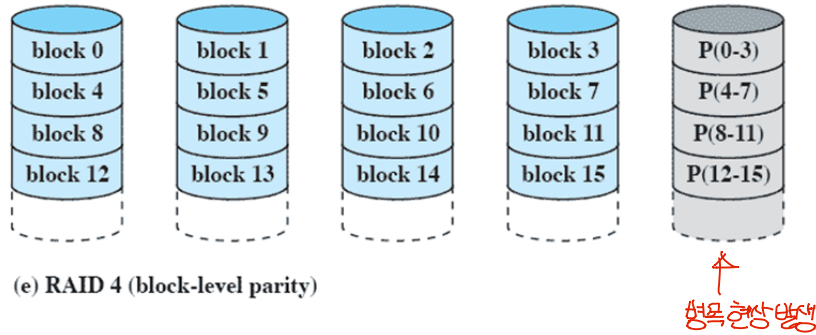
여기에서 \oplus 는 배타적-OR 함수(exclusive-OR function)이다.

- ✓ 드라이브 X1이 실패할 경우
- ✓ $X4(i) \oplus X1(i)$ 를 앞의 공식의 양쪽 변에 더하면 다음 식 구함

$$X1(i) = X4(i) \oplus X3(i) \oplus X2(i) \oplus X0(i)$$

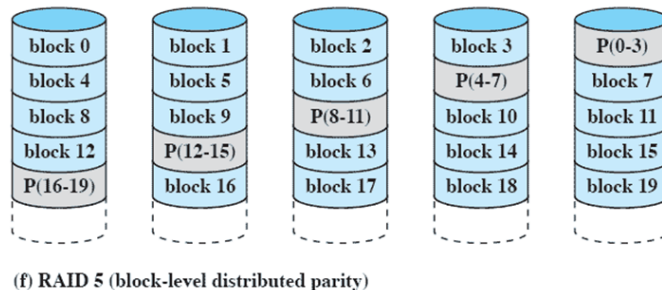
- ✓ 따라서 X1에 저장된 각 데이터 스트립의 내용은 배열의 나머지 디스크들 상의 상응하는 스트립 내용으로부터 재생성할 수 있음
- ✓ 이 원리는 RAID 3부터 6까지 모두 적용된다.
- 장점
 - 매우 높은 데이터 전송률, 비용 유리
- 단점
 - 한 번에 오직 하나의 입출력 요청만 실행
 - 트랜잭션 중심 환경에서는 성능이 나빠진다.

RAID 레벨 4



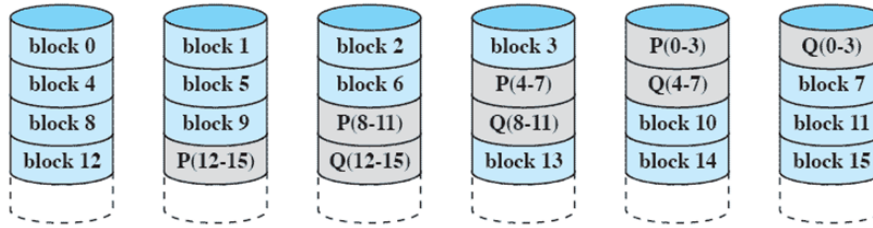
- 4에서 6의 RAID 레벨은 독립된 접근기술 사용
- 독립적 접근 배열에서는 각 구성 디스크가 독립적으로 작동되어 별개의 입출력 요청들이 병렬적으로 수행될 수 있음
- 따라서 높은 입출력 요청률을 요구하는 응용에는 독립적인 접근 배열이 더 적합하고, 상대적으로 높은 데이터 전송률을 요구하는 응용에는 덜 적합
- RAID 4 ~ 6 레벨에서는 큰 스트립 사용
- RAID 4에서는 비트별 패리티 스트립(bit-by-bit parity strip)이 각 데이터 디스크 상의 대응하는 스트립에 대해 계산되고 이 패리티 비트들은 패리티 디스크 상의 대응하는 스트립에 저장
- 단점
 - 작은 크기의 데이터에 대해 입출력 쓰기 요청이 발생하면, 쓰기 불이익
 - 쓰기가 일어날 때마다, 배열 관리 소프트웨어는 사용자 데이터뿐만 아니라 대응하는 패리티 비트들까지도 갱신
 - 모든 쓰기 연산은 항상 패리티 디스크를 접근해야 하며 따라서 패리티 디스크의 병목

RAID 5 레벨



- RAID 5는 RAID 4와 유사한 유사
- 차이점은 RAID 5는 패리티 스트립을 모든 디스크에 분산
- RR 방식 주로 사용
- n 개의 디스크를 가진 배열의 경우, 처음 n 개의 스트라이프를 위한 n 개의 패리티 스트립은 서로 다른 디스크 상에 존재하며 이 패턴이 반복
- 이렇게 패리티 스트립이 모든 드라이브들에 걸쳐 분산되면, RAID 4에서 볼 수 있는 단일 패리티 디스크의 잠재적인 입출력 병목현상을 피할 수 있음

RAID 6 레벨



(g) RAID 6 (dual redundancy)

- RAID 6 구성에서는, 두개의 다른 패리티 계산이 수행되어 서로 다른 디스크의 별도 블록에 저장
- 사용자 데이터로 N개의 디스크를 요구하는 RAID 6 배열은 N+2개의 디스크로 구성 \Rightarrow 비용↑
- P와 Q는 두 개의 서로 다른 데이터 검사 알고리즘
- 사용자 데이터를 포함한 두개의 디스크가 실패하더라도 데이터를 재생성 가능
- 이 중 하나는 RAID 4와 5에서 사용된 배타적-OR 연산
- 나머지 하나는 독립적인 데이터 검사 알고리즘
- 장점
- 높은 가용성
- 단점
- 데이터가 분실되려면 세 개의 디스크가 MTTR(mean time to repair, 수리될 때까지의 평균 시간) 간격 동안 모두 실패해야 하나 RAID 6에서는 쓰기 연산 때마다 두개의 패리티 블록에 영향을 미치기 때문에 상당한 쓰기 불이익이 있음
- 쓰기의 경우 RAID 5보다 30%이상 성능 저하 \Rightarrow 쓰기 성능↓

카테고리	레벨	설명	요구되는 디스크	데이터 가용성	큰 입출력 데이터 전송 능력	작은 입출력 요구율
스트라이핑	0	중복 없음	N	단일 디스크보다 낮음	매우 높음	읽기, 쓰기 모두에서 매우 높음
미러링	1	미러됨	2N, 3N, etc.	RAID 2,3,4,5 보다 높고, 6 보다 낮음	읽기는 단일 디스크보다 높고, 쓰기는 단일 디스크와 비슷함	읽기는 단일 디스크의 최대 두 배, 쓰기는 단일 디스크와 비슷함
병렬접근	2	해밍코드에 의한 중복	N+M	단일 디스크보다 상당히 높고, RAID 3,4,5 보다 높음	모든 레벨들 중 가장 높음	단일 디스크의 거의 두 배임
	3	비트 인터리브드 패리티	N+1	단일 디스크보다 상당히 높고, RAID 2,4,5와 대등	모든 레벨들 중 가장 높음	단일 디스크의 거의 두 배임
	4	블록 인터리브드 패리티	N+1	단일 디스크보다 상당히 높고, RAID 2,3,5와 대등	읽기는 RAID 0과 비슷함. 쓰기는 단일 디스크보다 상당히 낮음	읽기는 RAID 0 와 비슷함 쓰기는 단일 디스크보다 상당히 낮음
독립접근	5	블록 인터리브드 분산 패리티	N+1	단일 디스크보다 상당히 높고, RAID 2,3,4와 대등	읽기는 RAID 0과 비슷함. 쓰기는 단일 디스크보다 낮음	읽기는 RAID 0 와 비슷함 쓰기는 일반적으로 단일 디스크보다 낮음
	6	블록 인터리브드 이중 분산 패리티	N+2	모든 레벨들 중 가장 높음	읽기는 RAID 0과 비슷함 쓰기는 RAID 5보다 낮음	읽기는 RAID 0 와 비슷함 쓰기는 RAID 5 보다 상당히 낮음