



하이티센 2주차

RAID 개념

- RAID 란? - Redundant Array of Inexpensive(또는 Independent) Disk의 약자
- 여러개의 디스크를 하나의 디스크처럼 사용함
- 비용절감 + 신뢰성 향상 + 성능 향상의 효과를 냄
- 원래 목적은 저렴한 저용량의 디스크 여러 개를 하나의 비싼 대용량의 디스크로 사용하는 것
- 예를 들면 1TB 하드디스크 10개를 가지고 마치 10TB 하드디스크 1개처럼 사용하려는 것
- 하드웨어 RAID와 소프트웨어 RAID 두가지로 분류함

RAID 배경

- 제일 처음 제안한건 버클리 대학 연구실
- David Patterson과 Garth A Gibson 그리고 Randy Katz가 기존에 있던 개념을 정리 및 확장하여 RAID라는 용어를 만듦
- 1970-80년에는 우리가 지금 사용하는 개인 PC, 데스크탑 이렇게 사용화 되지 않았던 시절
- 커다란 메인프레임 컴퓨터가 회사에서 사용되던 시절
- 점점 개인 컴퓨터 가격이 낮아짐에 따라 사용 되기 시작했고, 하드디스크가 처음으로 메인프레임 컴퓨터가 아닌 저장장치로 사용 되기 시작했고 예전 시절보다 가격이 낮아지기 시작함
- 이 사회현상이 바로 RAID가 탄생하게 된 배경
- 사람들은 PC 하드 드라이브를 연결해 사용함으로써 메인프레임에 있는 큰 하드에서 내는 성능을 보일 수 있을 거라 생각함
- 여러 개 하드디스크를 동시에 연결해 사용하는게 에러를 유발하는 요인이 될 수 있지만 오히려 중복할 수 있으니 메인프레임 하나의 큰 드라이브보다 더 신뢰도를 높일 수 있을거라 생각했다.

하드웨어 RAID



- 하드웨어 장치로 RAID를 구성하는 것
- 하드웨어 제조업체에서 여러 개의 하드디스크를 가지고 장비를 만들어서 그 자체를 공급
- 좀 더 안정적이지만, 상당한 고가

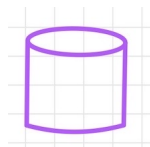
소프트웨어 RAID

- 컴퓨터에 장착된 여러 개의 디스크를 RAID로 구성하는 기능
- 이러한 기능을 운영체제에서 지원하는 방식
- 고가의 하드웨어 RAID의 대안
- 저렴한 비용으로 좀 더 안전한 데이터의 저장이 가능
- ⇒ 하드웨어 RAID보다 안정성이 떨어지기는 하지만 디스크만 추가하면 빠르게 안정적인 시스템을 구성할 수 있다는 장점

RAID LEVELS

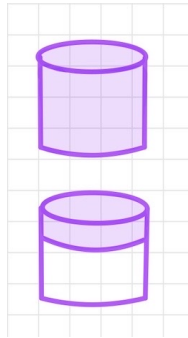
- 초기에는 RAID 방법으로 1,2,3,4,5 5개만 제안
- 그 이후에 회사들이 생기고 상용화가 되면서 다양한 RAID level이 추가가 됨

단순 볼륨



- 용량 : 디스크 개수
- 디스크 1개
- 공간 효율 100%

Linear RAID (스팬 볼륨)



- 용량 : 디스크 개수
- 디스크 2개 이상
- 첫 번째부터 저장
- 공간 효율 100%

RAID 0 (Stripping) : 신뢰도는 높아지지 않지만 성능이 좋아짐!



- 용량 : 디스크 개수
- 디스크 2개 이상
- 동시 저장
- 가장 빠름
- 공간 효율 100%
- Block Striping
- 사용 예시 : 동영상 관련된 것들 : 동영상은 디스크에서 계속 데이터를 읽어냄 → 성능이 중요 / but, 영화데이터가 없어지면 다른 곳에서 또 다운받으면 됨 → 신뢰도는 중요하지 않는데 성능이 중요한 하나의 예시

: 하나의 디스크에 데이터를 쭉쭉 저장하는게 아니라, 데이터를 블록 단위로 쪼개서 여러 디스크에 나눠 저장함

: 예를 들어 3개의 하드디스크를 사용한다고 하면 “말라이물랑이블로그” 라는 내용이 저장될 때 한 디스크에 “말라이물랑이블로그”가 다 써지는 게 아니라, 하나의 디스크가 다 차지 않았어도, 3개의 디스크 각각 동시에 “말” ”랑” ”이” 이렇게 써진다. 그 다음은 “물” ”랑” ”이”.

: 그럼 한 글자를 쓰는데 1초가 걸리다고 가정하면 한 디스크에 저장할 때는 9초가 걸리는데 이렇게 RAID 0을 사용하면 3초가 된다. 이것이 성능이 높아진다는 의미!

: 이 예시는 STRIPPING에 대한 이해를 돕기 위해 예시를 과장해서 든 것

: 실질적으로는 한 글자 단위가 아니라 블록 단위로 데이터가 저장이 된다.

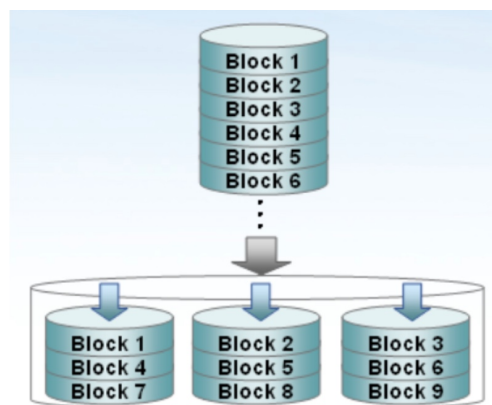
[정리]

- 이러한 스트라이핑 기술로 RAID 0은 빠른 입출력 속도를 제공
- 데이터 중복이나 패리티 없이 디스크에 분산하여 기록
- 처리속도는 빠르나 구성된 디스크 중에 하나라도 오류가 발생하면 데이터 복구가 불가

[요약]

- RAID 0은 block stripping 기법으로 성능에만 초점을 둔 방식
- 따라서 빠른 성능을 요구하되, 혹시 전부 잃어버려도 큰 문제가 되지 않는 자료를 저장하는데 적합

Block Stripping



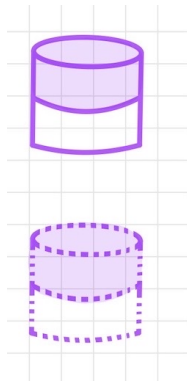
: 디스크 블록이 0 1 2 3 이렇게 여러 개 블록들이 있으면 원래는 순차적으로 0,1,2, ... 이게 하나의 디스크에 쭉 저장이 되지만

: 블록 스트라이핑을 하면 1번 블록은 첫번째 디스크에, 2번 블록은 두 번째 디스크에, 3번 블록은 세 번째 디스크에 나누어서 저장

: 보통 연속된 데이터의 접근이면 디스크 블록 0번 1번 2번 3번을 읽어라 그러면, 하나의 디스크에서 0번 읽고 1번 읽고 2번 읽고 3번 읽고 이렇게 하는 것 보다, 각각에서 디스크에서 동시에 읽으니까 성능이 좋음

- Stripping = 연속된 데이터를 여러 개의 디스크에 라운드로빈 방식으로 기록하는 기술
- 네 개의 드라이브가 있을 경우, 보통 하나의 섹터를 읽을 수 있는 시간에 네 개의 섹터를 동시에 읽을 수 있게 되는 성능

RAID 1(Mirroring) : 성능보다는 절대적으로 신뢰도가 우선시 !



- 용량 : 디스크개수 / 2
- 디스크 2개
- 동시 저장
- 결함 허용 제공
- 공간 효율 50%
- Mirroring

: 데이터가 쌍쌍으로 동일하게 저장됨

: 이렇게 하나의 데이터를 저장시키면 이거랑 똑같은 디스크를 하나 더 두는게 미러링 - 순차적

[정리]

- 미러링 기술을 사용하여 두 개의 디스크에 데이터를 동일하게 기록함
- 스트라이핑 기술은 사용하지 않고, 각 드라이브를 동시에 읽을 수 있어서 읽기 성능은 향상되나 쓰기 성능은 단일 디스크와 같음
- 디스크 오류 시 데이터 복구 능력은 탁월하지만, 중복 저장으로 인한 디스크의 낭비가 50%에 이른다.

RAID 2 (Bit Stripping, ECC) - RAID 0과 1을 합치려고 했지만,,,ㅠ

0	1	2	3	4	5	6
A0	A0	A0	A0	Px	Py	Pz
A1	A1	A1	A1	Px	Py	Pz
A2	A2	A2	A2	Px	Py	Pz
A3	A3	A3	A3	Px	Py	Pz
...

- 기록용 디스크와 데이터 복구용 디스크를 별도로 둠
- 복구방법 : Hamming code를 사용
- 에러 검출 능력이 없는 드라이브를 위해 사용하지만 4개의 디스크에 기록하기 위해서 3개의 부가데이터 디스크를 기록해야 함
- 최근 디스크다라이브들은 기본적으로 오류정정 기능을 가지고 있으므로 요즘은 거의 사용하지 않음
- 비효율적이면서 비용증가
- Bit Stripping + ECC

: 사진을 보면, RAID 2는 하나의 디스크 블록을 4개에다가 나누어 저장

: A0이 0,1,2,3에 있음

: 현재 RAID 2 에서 사용하는 건 하나의 디스크 블록 A0를 네 개의 디스크에 나눠서 저장

이 그림은 bit level stripping 한 것을 표현 ← 잘 안쓰임...

+해밍코드 :

해밍코드는 데이터비트에 몇 개의 체크비트가 추가된 코드이다.

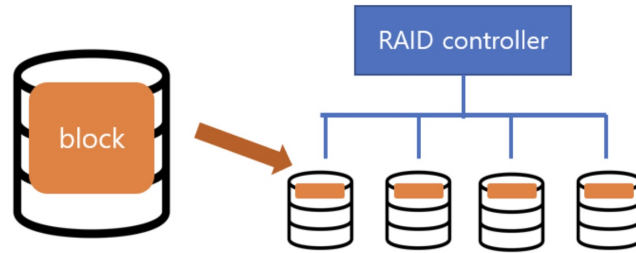
기존의 체크비트들은 수신된 데이터열에 에러가 있다 없다 정도만 확인 할 수 있었는데,

해밍코드를 이용하면 수신단은

에러비트의 위치까지 알 수 있게 할 뿐만 아니라 정정

할 수 있다!

+ 비트 스트라이핑 (Bit Stripping)



- Stripe : 줄
- 비트스트라이핑을 한건데 저 주황색 이 줄 같이 보임
- 그래서 지금 원래 디스크 하나에다가 저장을 하려면 디스크 블록 크기가 왼쪽만큼이다. 그럼 이렇게 네 개에다가 즉, 디스크 블록 하나를 이렇게 4개에다가 비트로 쪼개어 나눠 저장하는 걸 비트스트라이핑 방법이라고 한다.
- 블록이 4K라고 하면 Raid Controller가 4k를 알아서 1k 1k 1k 1k 이렇게 나눠서 저장을 하는 것
- 그럼 나중에 아까 기록한 데이터 읽어줘! 이렇게 하면 여기 여기 여기 네 개의 디스크가 동시에 읽어서 합쳐서 전달함
- 보통 요청은 데이터를 순차적으로 읽어서 줘야 하는 요청이 많은데, 이렇게 하면 동시에 읽으니까 하나에서 읽는거보다 속도가 빠름

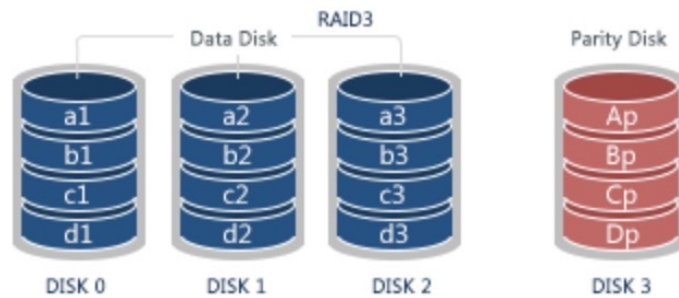
: 디스크 블록 하나를 여러 개로 쪼개서 이런 줄을 만들어서 저장하는 방법 = bit stripping

+에러 코릭팅 코드 ECC(Error Correcting Code)



- ECC는 패리티 디스크보다 디스크를 추가로 더 뒤서 해밍코드 방식으로 동작하게 함으로써 커버가 가능
- 읽을 때 마다 데이터에 에러는 없었는지 확인도 해주지만 구현하기 어려움
- 더 좋은 기법이 나와서 더이상 사용되지 않음

RAID3 (bit stripping, parity)



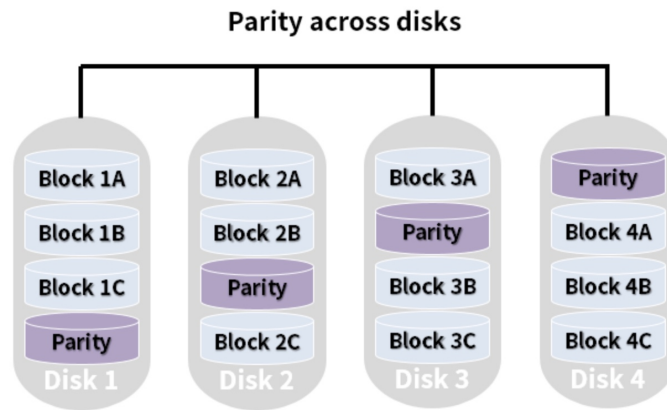
- 최소 디스크 개수 : 3
- 데이터가 비트 단위로 분산 저장되기 때문에 읽기 및 쓰기 동작이 수행될 때 배열 내의 모든 데이터 디스크들이 참여하게 된다.
- 쓰기 동작마다 패리티 비트들을 갱신해야 한다는 문제가 있음

→ 한 디스크 fail 정도면 괜찮~ → 보통 디스크가 고장나면 하나 고장나지 2개가 고장나진 않는듯~ → 그러니까 나는 ECC 처럼 막 몇개씩 갖다두지 않고 딱 하나만 더 돌려~ → 이렇게 된게 RAID 3

[요약]

- 스트라이핑 기술을 사용하며 디스크를 구성하고, 패리티 정보를 저장하기 위해 별도로 하나의 디스크를 사용
- 입출력작업이 동시에 모든 디스크에 대해 이루어지므로 입출력을 겹치게 할 수는 없다
- 보통 대형 레코드가 많은 시스템에서 사용

+패리티 디스크 Parity Disk



Parity (패리티) 란?

디스크 장애 시 데이터를 재 구축하는데 사용할 수 있는 사전에 계산된 값으로 디스크의 4개 블록 중 3개는 데이터를 저장하는데 쓰고 나머지 하나는 Parity 영역으로 둬

디스크 하나가 장애가 날 경우 Parity 영역을 이용해서 장애 난 디스크의 데이터를 복구해내는 것

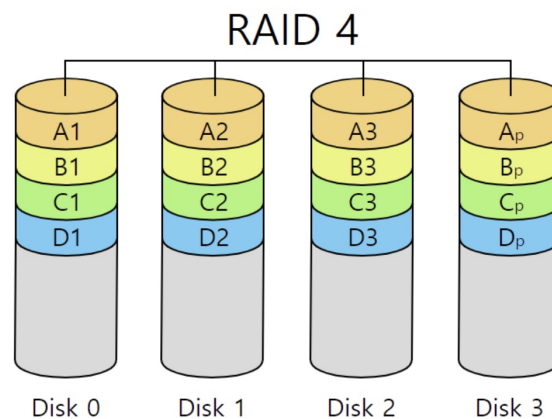
따라서 문제가 발생할 경우, 컨트롤러 가 정상적으로 운영되고 있는 다른 디스크로부터 손실 된 데이터를 가져와 복구/재생함

@ 비트레벨 스트라이핑 VS 블록레벨 스트라이핑

디스크라고 하는 것은 한 번에 가서 많이 읽어야함

비트레벨은 디스크가 조금 조금 나뉘서 읽기 때문에 성능의 향상 폭이 그렇게 크지 않음

RAID 4 (Block Striping, Parity)



- RAID 3보다 좀 더 개선된 형태

- 블록 형태의 스트라이핑 기술을 사용하여 디스크를 구성
- 단일 디스크로부터 레코드를 읽을 수 있고 데이터를 읽을 때 중첩 입출력의 장점
- 쓰기 작업은 패리티 연산을 해야하고 패리티 디스크에 저장해야 하기 때문에 입출력의 중첩이 불가능
- 패리티 디스크에 접근이 집중되어 병목(전체 시스템의 성능이나 용량이 하나의 구성 요소로 인해 제한을 받는 현상)현상 발생 가능

[정리]

- 디스크 중 하나가 fail 나면 Parity disk 로 인해 복구 가능
- 한 디스크에 접근해서 블록 데이터를 읽는 동안, 다른 디스크에서 또 다른 요청을 처리할 수 있음

: RAID2, 3에서는 데이터가 비트 단위로 분산 저장되었기 때문에 어떤 데이터를 읽거나 쓰기 위해서는 모든 데이터 디스크들을 동시에 접근

: RAID 4에서는 필요한 데이터 블록이 어느 한 디스크에 모두 저장되어 있기 때문에 각 액세스 요구가 서로 다른 디스크들에서 독립적으로 처리 가능

출처 :

<https://www.youtube.com/watch?v=yFompxWVqNw&list=PLVsNizTWUw7EJ9z-LW3lv3VC-6HI9I3hN&index=30>

<https://www.youtube.com/watch?v=Ah-jFz5xd7g&list=PLVsNizTWUw7FxQhmscBfWXJzd1xlfQWYo&index=17>

<https://ko.101-help.com/ed26dc7194-software-raid-vs-hardware-raid-caijeomi-seolmyeongdoeeossseubnida/>

<https://jhnyang.tistory.com/160?category=815411>

<https://jhnyang.tistory.com/157>

<https://jongamk.tistory.com/2199>

<https://www.dknyou.com/blog/?q=YToxOntzOjE5OiJrZXI3b3JkX3R5cGU0M6MzoiYWxsJt9&bmode=view&idx=7342927&t=board>

<https://raisonde.tistory.com/entry/RAID0부터-RAID7까지-모든-RAID구성을-알아보자>

