



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0091628
(43) 공개일자 2013년08월19일

(51) 국제특허분류(Int. Cl.)
G06F 11/16 (2006.01) G06F 12/16 (2006.01)
(21) 출원번호 10-2012-0155018
(22) 출원일자 2012년12월27일
심사청구일자 없음
(30) 우선권주장
13/368,725 2012년02월08일 미국(US)

(71) 출원인
엘에스아이 코포레이션
미국 캘리포니아주 95131, 새너제이, 라이더 파크
드라이브 1320
(72) 발명자
네어 나만
미국 캘리포니아주 95035 밀피타스 에이디800 바
버 레인 1621
레 카이 엠
미국 조지아주 30093 노어크로스 아노크 샤클레포
드 로드 4165
(74) 대리인
제일특허법인

전체 청구항 수 : 총 20 항

(54) 발명의 명칭 RAID의 개선된 리빌드 시스템 및 방법

(57) 요약

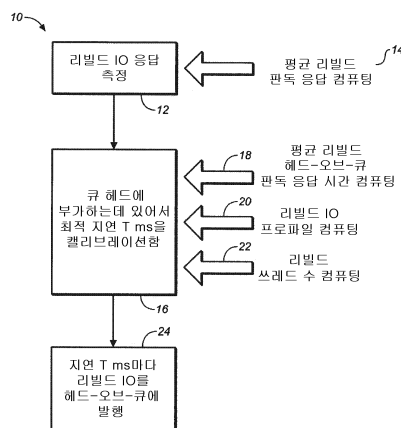
호스트 IO 상태 하에서 개선된 RAID 리빌드를 위한 시스템 및 방법은 리빌드 시간을 크게 개선하고 호스트 IO 고갈을 방지한다. RAID의 일부인 드라이브 내의 큐는 리빌드 IO 요청 및 호스트 IO 요청을 저장하는데 사용되며, 리빌드 IO는 드라이브 큐의 헤드에 발행된다. 드라이브로의 리빌드 요청은 지연 시간만큼 지연된다. 이러한 지연으로 인해서, 본 발명에 따라서 의도하지 않는 부작용이 발생하지 않게 되며 이로써 이러한 부작용이 일으킬 수 있는 RAID 시스템에 대한 호스트 IO 고갈이 방지된다.

이 지연은 디스크 헤드 응답 시간, 리빌드에 할당되는 시간, 리빌드 쓰레드 수, 드라이브 거동, 리빌드 IO 프로파일, 워크로드 및 로컬리티를 포함하는 호스트 IO 프로파일, 일 바람직한 실시예에서 매 70초로 주기화되는 일련의 이벤트의 기간 등과 같은 다수의 변수로부터 경험적으로 컴퓨팅된다.

리빌드 IO 프로파일은 리빌드가 호스트 IO에 비해서 얼마나 신속하게 발생하는지 및 리빌드가 호스트 IO 동작에 어느 정도의 영향 또는 임팩트를 주는지에 대한 가령 "노 임팩트", "로우 임팩트", "하이 임팩트" 및 "최대 임팩트"와 같은 다수의 옵션으로서 제공되는 리빌드 레이트를 갖는다.

중한 호스트 IO 상태 하에서의 소정의 구성에서는, 본 발명의 방법 및 장치를 사용하는 리빌드 시간은 종래 기술에 비해서 10 배 정도 개선될 수 있다.

대표도 - 도1



특허청구의 범위

청구항 1

호스트 IO 상태 하에서, 개선된 RAID(Redundant Array of Independent Disks) 리빌드(rebulid)를 위한 다수의 드라이브를 갖는 저장 시스템으로서,

상기 저장 시스템 내의 드라이브의 리빌드 IO 응답을 측정하는 펌웨어와,

리빌드 IO 요청 및 호스트 IO 요청을 포함하는, 상기 드라이브에 대한 큐와,

상기 큐로의 리빌드 IO의 발행을 지연시키는 펌웨어를 포함하는

저장 시스템.

청구항 2

제 1 항에 있어서,

상기 지연을 캘리브레이션하는 펌웨어를 더 포함하는

저장 시스템.

청구항 3

제 2 항에 있어서,

상기 캘리브레이션하는 펌웨어는 상기 큐로부터의 평균 리빌드 응답 시간, 전체 시간에 대한 리빌드에 할당되는 시간 및 리빌드 쓰레드 수를 포함하는 파라미터에 따라서 상기 지연을 캘리브레이션하는

저장 시스템.

청구항 4

제 2 항에 있어서,

상기 펌웨어가 얼마나 신속하게 리빌드할지를 특징하는 리빌드 레이트를 갖는 펌웨어를 더 포함하며,

상기 리빌드 레이트는 최소치 및 최대치를 가지며,

상기 최대치에 비해서 최소치에서, 리빌드 IO보다 호스트 IO에 우선 순위가 부여되는

저장 시스템.

청구항 5

제 4 항에 있어서,

상기 펌웨어 내에서 리빌드 레이트는 구성 가능하며 이로써 상기 펌웨어로 하여금 상기 최소치 내지 최대치 범위의 몇 개의 사전 결정된 값들 사이에서 변하는 리빌드 레이트를 갖게 하는

저장 시스템.

청구항 6

제 2 항에 있어서,

상기 펌웨어는 판독 응답 시간, 리빌드 할당 시간, 사용자가 선택한 리빌드 IO 프로파일, 구성 가능한 리빌드 IO 프로파일 및 리빌드 쓰레드 수로 구성된 그룹으로부터 선택된 파라미터에 따라서 상기 지연을 캘리브레이션 하는

저장 시스템.

청구항 7

제 6 항에 있어서,

주기적 방식으로 상기 큐의 헤드에 리빌드 IO를 발행하는 펌웨어를 더 포함하며

상기 주기적 방식의 일 주기는 다수의 스테이지를 포함하며,

상기 다수의 스테이지는,

상기 지연을 도입하지 않고서 상기 큐 내에 리빌드 IO를 통상적인 방식으로 발행함으로써 리빌드를 진행하는 제 1 스테이지와,

리빌드 레이트와 응답 시간 중 적어도 하나가 리빌드 레이트 및 응답 시간에 대한 사전 결정된 값보다 작을 때에 수행되어 상기 지연 T를 컴퓨팅하는 제 2 스테이지와,

T ms마다 상기 큐의 프론트에 리빌드 IO를 발행하는 제 3 스테이지

를 포함하는 주기를 포함하는

저장 시스템.

청구항 8

제 7 항에 있어서,

상기 제 2 스테이지는 $T = C/nR - \text{resp}$ 에 의해서 상기 지연 T를 컴퓨팅하며,

상기 $nR = r\text{IOPS} / \text{리빌드 쓰레드 수}$ 이며,

상기 $r\text{IOPS} = rt/\text{resp}$ 이며,

상기 C는 총 시간으로서 시간 상수이며,

상기 리빌드 쓰레드 수는 상기 RAID 시스템 내에서 리빌드 쓰레드의 수이며,

상기 rt 는 상기 총 시간 (C)에 대한 리빌드 할당 시간이며,

상기 resp 는 상기 큐의 헤드에서 평균 헤드-오브-큐 리빌드 판독 응답 시간인

저장 시스템.

청구항 9

제 7 항에 있어서,

상기 펌웨어는 RAID 1, RAID 2, RAID 3, RAID 4, RAID 5, RAID 6, 혼성 RAID, 카피백 동작을 갖는 저장 시스템 및 소거 코드를 사용하는 저장 시스템으로 구성된 그룹으로부터 선택된 저장 시스템에서 동작하며,

상기 혼성 RAID는 상기 RAID 1, RAID 2, RAID 3, RAID 4, RAID 5, RAID 6 및 RAID 0의 조합을 포함하는

저장 시스템.

청구항 10

저장 시스템 내의 다수의 드라이브에 대한 RAID(Redundant Array of Independent Disks) 리빌드(rebulid) 방법으로서,

호스트 IO 상태 하에서 RAID 시스템 리빌드 동작 시에 리빌드 IO 응답을 측정하는 단계와,

상기 RAID 내의 일 드라이브의 큐의 헤드로의 리빌드 IO의 발행을 지연시키는 지연을 컴퓨팅하는 단계와,

상기 큐의 헤드에 리빌드 IO를 발행하고 상기 리빌드 IO의 실행을 상기 지연만큼 지연시키는 단계를 포함하는

RAID 리빌드 방법.

청구항 11

제 10 항에 있어서,

판독 응답 시간, 리빌드 할당 시간, 사용자가 선택한 리빌드 IO 프로파일, 구성 가능한 리빌드 IO 프로파일 및 리빌드 쓰레드 수로 구성된 그룹으로부터 선택된 파라미터에 따라서 상기 지연을 컴퓨팅하는 단계를 더 포함하는

RAID 리빌드 방법.

청구항 12

제 11 항에 있어서,

최소 리빌드 레이트 내지 최대 리빌드 레이트로부터 상기 리빌드 할당 시간을 선택하는 단계를 더 포함하는

RAID 리빌드 방법.

청구항 13

제 12 항에 있어서,

구성 가능한 리빌드 프로파일로부터 상기 리빌드 할당 시간을 선택하는 단계를 더 포함하며,

상기 구성 가능한 리빌드 프로파일은 상기 최소 리빌드 레이트 내지 최대 리빌드 레이트의 범위에 존재하는 다수의 리빌드 레이트를 포함하는

RAID 리빌드 방법.

청구항 14

제 13 항에 있어서,

상기 단계들은 주기적 방식으로 수행되며,

상기 주기적 방식의 주기는,

상기 지연을 도입하지 않고서 상기 큐 내에 리빌드 IO를 통상적인 방식으로 발행함으로써 RAID 저장 시스템에 대한 리빌드를 진행하는 제 1 스테이지와,

리빌드 레이트와 응답 시간 중 적어도 하나가 리빌드 레이트 및 응답 시간에 대한 사전 결정된 값보다 작을 때

에 수행되어 ms 단위의 상기 지연 T를 컴퓨팅하는 제 2 스테이지와,
지연 T ms마다 상기 큐에 리빌드 IO를 발행하는 제 3 스테이지를 포함하는
RAID 리빌드 방법.

청구항 15

제 10 항에 있어서,
판독 응답 시간, 리빌드 할당 시간, 사용자가 선택한 리빌드 IO 프로파일, 구성 가능한 리빌드 IO 프로파일 및
리빌드 스레드 수를 포함하는 파라미터에 따라서 상기 지연을 컴퓨팅하는 단계를 더 포함하는
RAID 리빌드 방법.

청구항 16

제 15 항에 있어서,
상기 측정 단계, 상기 컴퓨팅 단계 및 상기 발행 단계는 상기 RAID 저장 시스템에서 주기적 방식으로 발생하는
RAID 리빌드 방법.

청구항 17

제 16 항에 있어서,
상기 단계들은 주기적 방식으로 수행되며,
상기 주기적 방식의 주기는,
상기 리빌드 IO를 드라이브의 큐의 헤드에 발행하지 않고서 상기 큐 내에 리빌드 IO를 통상적인 방식으로 발행
함으로써 상기 RAID 저장 시스템에 대한 리빌드를 진행하는 제 1 스테이지와,
리빌드 레이트와 응답 시간 중 적어도 하나가 리빌드 레이트 및 응답 시간에 대한 사전 결정된 값보다 작을 때
에 수행되어 ms 단위의 상기 지연 T를 컴퓨팅하는 제 2 스테이지와,
T ms마다 상기 드라이브의 큐의 헤드에 리빌드 IO를 발행하는 제 3 스테이지를 포함하는
RAID 리빌드 방법.

청구항 18

제 17 항에 있어서,
상기 방법은 RAID 1, RAID 2, RAID 3, RAID 4, RAID 5, RAID 6, 혼성 RAID, 카피백 동작을 갖는 저장 시스템,
소거 코드를 사용하는 저장 시스템, 물리적 드라이브에 IO를 발행하는 것을 채용하는 저장 시스템, 매체 스캔을
포함하는 저장 시스템, 일관성 체크를 포함하는 저장 시스템, 초기화를 포함하는 저장 시스템 및 포맷팅을 포함
하는 저장 시스템으로 구성된 그룹으로부터 선택된 저장 시스템에서 사용되며,
상기 혼성 RAID는 상기 RAID 1, RAID 2, RAID 3, RAID 4, RAID 5, RAID 6 및 RAID 0의 조합을 포함하는
RAID 리빌드 방법.

청구항 19

호스트 IO 상태 하에서 RAID 리빌드를 위한 장치로서,

RAID 저장 시스템에서 리빌드 IO 응답을 측정하는 수단과,

상기 RAID 저장 시스템에 대한 리빌드 IO 요청 및 호스트 IO 요청을 저장하기 위해서 큐잉(queuing)하는 수단과,

큐의 헤드로의 상기 리빌드 IO 요청의 발행을 지연시키는 지연을 컴퓨팅하는 수단-상기 지연은 밀리초 단위의 지연 시간을 포함함-과,

상기 큐의 헤드에 리빌드 IO를 발행하고 상기 지연 시간만큼 상기 리빌드 IO의 실행을 지연시키는 수단과,

상기 RAID 시스템에 대한 리빌드 레이트를 설정하는 수단을 포함하며,

상기 측정 수단, 컴퓨팅 수단 및 발행 및 지연 수단은 주기적 방식으로 동작하며,

상기 주기적 방식의 주기의 제 1 기간에서, 상기 측정 수단은 상기 RAID 시스템이 통상적인 방식으로 동작할 때에, 상기 RAID 시스템의 응답 레이트 및 응답 시간 중 어느 하나가 사전 결정된 값보다 클 때까지, 상기 RAID 시스템의 평균 리빌드 응답 시간을 컴퓨팅하고,

상기 주기적 방식의 주기의 제 2 기간에서, 상기 컴퓨팅 수단은 상기 드라이브 큐의 헤드에 부가된 지연 인스턴션과 함께 상기 RAID 시스템이 동작하고 있을 때의 평균 리빌드 응답 시간, 상기 설정 수단의 리빌드 레이트 및 상기 RAID 시스템 내의 쓰레드 수를 컴퓨팅함으로써 최적 지연을 컴퓨팅하며,

상기 주기적 방식의 주기의 제 3 기간에서, 상기 발행 및 지연 수단은 상기 제 2 기간에서 컴퓨팅된 최적 지연만큼 상기 드라이브 큐의 헤드로의 리빌드 IO 요청의 발행을 지연시키는

RAID 리빌드 장치.

청구항 20

제 19 항에 있어서,

상기 제 2 기간에서 상기 지연을 컴퓨팅 수단은 상기 최적 지연 T를 $T = C/nR - resp$ 에 의해서 컴퓨팅하며,

상기 $nR = rIOPS /$ 리빌드 쓰레드 수이며,

상기 $rIOPS = rt/resp$ 이며,

상기 C는 총 시간으로서 시간 상수이며,

상기 리빌드 쓰레드 수는 상기 RAID 시스템 내에서 리빌드 쓰레드의 수이며,

상기 rt는 상기 총 시간 (C)에 대한 리빌드 할당 시간이며,

상기 resp는 상기 큐의 헤드에서 평균 헤드-오브-큐 리빌드 관독 응답 시간인

RAID 리빌드 장치.

명세서

기술분야

[0001] 본 발명은 전반적으로 RAID(Redundant Array of Independent Disks) 분야에서의 복구 또는 리빌드(rebuild)에 관한 것이다.

배경기술

[0002] 본 발명은 전반적으로 RAID에 관한 것이다. RAID는 다수의 디스크 드라이브 성분들을 하나의 논리적 유닛으로 결합하는 가상화 형태의 저장 기술로서 특히 네트워크에서 디스크 고장에 의해서 유발되는 에러를 줄이기 위해서 주로 사용된다. 데이터는 데이터 스트라이핑(data striping)으로 알려진 순차적 방식으로 해서 몇 개의 디스

크 상에 저장되는 블록들로 분할된다. 패리티 블록(parity block)이 에러를 체크하고 디스크 드라이브 중 하나가 고장날 시에 데이터를 재구성하기 위한 수단을 형성하며, 패리티 리던던시(parity redundancy)를 형성한다.

[0003] 적절하게 구성된 RAID는 몇몇 이점이 있다. RAID의 이점은 리던던시(RAID 0 구성 저장)를 통한 높은 데이터 보안성, 폴트 톨러런스(fault tolerance), 개선된 데이터 액세스(데이터 가용성), 대량의 인접 디스크 공간을 생성하기 위한 증가된 통합 능력 및 개선된 성능을 포함한다. RAID은 고가의 하드웨어가 필요하며 특정한 유지 관리가 필요하다.

[0004] 특히 혼성(네스트형(nested)) RAID 시스템을 포함한다면 수많은 타입의 RAID가 존재하지만, 표준적으로는 RAID 레벨 0 내지 RAID 레벨 6이 있으며 이들은 디스크 기반 저장 시스템을 위한 모든 통상적인 데이터 맵핑 및 보호 방식들을 성공적으로 규정할 수 있다. RAID를 위한 다른 분류 시스템들은 고장 저항형 시스템(드라이브 고장으로 인해서 데이터 손실을 방지하는 시스템), 고장 허용형 시스템(임의의 단일 성분 또는 다중 성분의 고장으로 인해서 데이터 액세스 손실을 방지하는 시스템) 및 재앙 허용형 시스템(어느 하나가 저장된 데이터로의 액세스를 제공하는 2 개 이상의 독립된 구역들로 구성된 시스템)을 포함한다.

[0005] 몇몇 잘 알려진 RAID 레벨은 RAID 레벨 0, 레벨 5 및 레벨 6을 포함한다. 레벨 0은 리던던시가 없으며 패리티 또는 미러링(mirroring)도 없는 블록 레벨 스트라이핑 방식이다. 이 레벨 0은 개선된 성능 및 추가적인 저장 능력을 제공하지만 폴트 톨러런스를 구비하지 않고 있다. RAID 레벨 0에서, 블록들은 각각의 드라이브에 동일한 섹터 상에 동시에 기록되며 보다 작은 섹션의 데이터들이 각 드라이브로부터 병렬로 판독되며 대역폭을 증가시킨다. RAID 레벨 0은 에러 체크를 구현하지 못하며 따라서 어떠한 에러도 교정될 수 없다. 이 레벨 0은 성능을 증가시키기 위해서 혼성 RAID 시스템들에서 사용될 수 있다. RAID 레벨 5은 분산된 패리티를 가지면서 블록 레벨 스트라이핑 방식이며, 패리티는 데이터를 따라서 분산되어 있으며 하나의 드라이브를 제외하고 모든 드라이브가 실제 사용을 위해서 존재하며 하나의 드라이브는 단일 드라이브 고장시를 위해서 예비된다. 단일 드라이브 고장 시에, 어레이는 파괴되지 않고 임의의 후속 데이터 판독 사항이 분산된 패리티로부터 계산되며 이로써 드라이브 고장이 엔드 유저에게 영향을 주지 않게 된다. 그러나, 단일 드라이브 고장은 고장난 드라이브가 대체되고 관련 데이터가 복구될 때까지 전체 어레이 성능을 저감시킨다. RAID 레벨 6은 이중 분산된 패리티를 갖는 블록 레벨 스트라이핑 방식이며 2 개의 드라이브 고장에 대한 폴트 톨러런스를 제공한다. 이 어레이는 2 개의 드라이브가 고장나도 계속 동작할 수 있다. RAID 레벨 6의 장점은 대형 RAID 그룹들이 실현될 수 있게 하는 것인데 이는 큰 용량의 드라이브들은 단일 드라이브 고장으로부터 리빌드 또는 복구되는데 필요한 시간을 증가시키기 때문에 중요하다.

[0006] 카피백(copyback)은 디스크 내용을 대체 디스크에 카피함으로써, 동작하는 어레이 멤버 디스크를 다른 디스크 멤버로 대체하는 것이다. 이 카피백은 고장 성분이 해당 어레이를 고장 및 악화시키기 전에 이 고장 성분을 대체하거나 어레이에 대한 특정 물리적 구성을 복구하는데 사용되며 어레이를 감소시키지 않고서 이루어진다.

[0007] 통상적인 스핀들 기반 하드 드라이브 내에 저장된 2 차 메모리는 전기자에 의해서 유지되는 자기 헤드에 의해서 판독되는 데이터를 저장하는 다수의 회전 디스크 플래터를 갖는다. 최신 디스크들은 통상적으로 몇 개의 헤드 및 플래터를 갖는다. 단일 입출력 동작이 완료되기 위해서, 전기자는 자기 헤드를 섹터에 있어서 데이터를 유지하고 있는 플래터 트랙으로 이동시켜야 하는데 이는 시킹(seeking) 프로세스로 지칭되며 시크 시간(seek time)이 소비되며, 이어서 소망하는 섹터가 자기 헤드 아래에서 회전할 때까지 대기해야 하는데 이는 회전 레이턴시로 지칭되는 대기 시간을 요한다. 이러한 펌웨어, 하드웨어 또는 소프트웨어에 의해서 유발되는 시간 및 지연은 드라이브 응답 시간을 포함한다.

[0008] IOPS(Input/Output Operation Per Second, 아이 오프스로 발음됨)는 HDD(하드 디스크 드라이브) 또는 SAN(저장 구역 네트워크)과 같은 컴퓨터 저장 장치를 벤치마크하는데 사용되는 통상적인 성능 측정 단위이다. 저장 장치 제조자들에 의해서 공지되는 IOPS 값들은 실제의 애플리케이션 성능을 보장하지 않으며 IOPS는 (인텔에 의해서 최초로 개발된) Iometer와 같은 애플리케이션으로 측정될 수 있다. 임의의 시스템 구성에서 가능한 특정 IOPS 값은 판독 및 기록 동작의 균형, 순차적 액세스 및 랜덤 액세스 패턴의 혼합, 워커 쓰레드(worker thread) 및 큐 깊이 값, 데이터 블록 크기와 같은 다양한 변수 및 시스템 구성, 저장 드라이버, OS 백그라운드 동작 등과 같은 다른 요인들에 따라서 크게 변할 것이다.

[0009] RAID 어레이가 하나 이상의 하드 드라이브 고장을 가질 때에, 이 하드 드라이브들은 복구 또는 리빌드되어야 한다. 리빌드 동작을 구성하는 IO 동작은 리빌드 IO이며 RAID 그룹 내에서 하드 드라이브들의 정상적 동작과 같은 통상적인 비리빌드(non-rebuild) 동작에 대한 IO 동작은 호스트 IO이다. RAID 시스템에서, 호스트 IO 상태 하에서 리빌드 성능은 크게 영향을 받는다. 이는 리빌드 동작이 디스크 그룹 내의 모든 나머지 디스크들을 판독하는

바를 요구하고 각 디스크가 리빌드 동작을 서비스하고자 해야 하기 때문이다. 이와 더불어, 각 드라이브 모델은 자신의 IO 큐를 최적화하기 위해서 자신의 고유한 방법을 가지며 이 방법은 독점적인 면이 있으며 드라이브 시크(drive seek)를 최소화하기 위해서 IO들을 재배열하기 때문이다. 따라서, 리빌드 IO는 통상적으로 가장 많이 재배열되는 것이 되기 때문에 크게 영향을 받으며 높은 레이턴시를 경험하게 된다. 이는 직접적으로 리빌드 성능에 영향을 주고 시스템은 단지 1TB 데이터를 리빌드하기 위해서 8 내지 30일을 요할 수 있다. 이러한 긴 리빌드 시간으로 인해서 RAID 그룹은 긴 기간 동안 연장된 약화된 호스트 IO 성능에 노출되게 되어서 2 차 또는 3 차 드라이브 고장이 발생할 수 있으며 이로써 전체 RAID 그룹은 오프라인 상태가 되어서 데이터 손실이 발생할 수 있다.

[0010] 모든 RAID 시스템들은 통상적으로 제어기에 의해서 제어 및 관리되는 다수의 IO 큐들을 가지며 각 드라이브는 그 자신의 IO 큐들을 갖는다. 본 발명은 후자, 즉 32 또는 64 커맨드 깊이를 가질 수 있는 개별 드라이브 내의 IO 큐에 관한 것이다. 리빌드 IO는 큰 레이턴시 또는 응답 시간을 경험하는데, 그 이유는 리빌드 IO가 통상적으로 드라이브 큐 내의 나머지 호스트 IO와 동일한 로컬리티(locality) 또는 지형학적 위치를 공유하지 않고 드라이브 헤드에 일 LBA(Logical Block Address)에서 다음 LBA에 도달하는데 아주 멀리 탐색할 필요가 없도록 그룹핑된 디스크 상의 공통 섹터 영역 또는 클러스터로서 규정된 로컬리티를 공유하지 않기 때문이며, 이로써 리빌드 IO 동안에 악영향이 존재하게 된다. 리빌드 동작에 있어서는 데이터를 재구성하기 위해서 전체 드라이브가 관독되어야 하기 때문에, 대부분의 리빌드 동작에 있어서, 리빌드 IO는 호스트 IO와 로컬리티를 공유하지 않을 것이다. 모든 시스템들은 통상적으로 드라이브로의 리빌드 IO 및 호스트 IO 레이트를 제어하지만, 일단 이 IO들이 해당 드라이브로 넘겨지면, 이 드라이브는 이를 인수해야 하고 본 명세에서 기술되는 바와 같이 이 드라이브는 RAID 제어기에 의해서 컴퓨팅된 레이트들을 스큐하며 이로써 리빌드 IO 고갈 상황이 발생하게 된다.

[0011] 드라이브들은 리빌드되고 있는 가령 가상 그룹의 스트라이프 크기와 같은, 디스크 그룹의 IO 크기의 입도 및 가용한 시스템 자원들에 기초하여서, RAID 시스템 내의 펌웨어에 의해서 그 구현이 결정되는 일련의 하나 이상의 동시적 쓰레드 또는 프로세스로 리빌드될 수 있다.

발명의 내용

해결하려는 과제

[0012] 본 발명은 리빌드 IO의 높은 레이턴시로 인해서 리빌드 시간이 연장되는 문제를 호스트 IO 동작들을 계속적으로 서비스하는 동안에 리빌드하는 신규한 경험적 방법을 사용하여서 해결하고자 한다.

과제의 해결 수단

[0013] 따라서, 본 발명의 일 측면은 RAID 레벨 저장 시스템이 호스트 IO 상태 하에서 리빌드되고 있는 때에 리빌드 IO의 높은 레이턴시를 방지하여서, 이 시스템이 리빌드되지 않은 때와 같이 정상적 또는 통상적인 바와 같이 리빌드 및 이와 동시에 동작하게 함으로써, 결정론적 방식으로 리빌드 시간을 크게 개선할 수 있는 방법을 제공한다.

[0014] 본 발명의 다른 측면은 호스트 IO 상태 하에서 리빌드 동작과 연관된 IO를 헤드-오브-큐 SCSI 태그(head-of-queue SCSI tag)와 함께 발행하는 것이다.

[0015] 본 발명의 다른 측면은 호스트 IO 고갈 또는 데드(dead)가 발생하지 않도록 리빌드 레이트를 제어하는 것이다. 경험론적 방법에 의해서, 사용자 요청마다 호스트 IO가 서비스되고 어떠한 IO 고갈도 발생하지 않도록 보장하면서 최적의 리빌드 레이트를 제공하도록 리빌드 레이트가 올바르게 제어되고 연속적으로 조절된다.

[0016] 본 발명의 또 다른 측면은 RAID 저장 시스템에 의해서 리빌드 레이트를 제어하도록 사용되는 펌웨어에 의해서 리빌드 IO를 큐로 발행하는 바를 의도적으로 지연시킨다. 이 지연의 시간은 디스크 헤드 응답 시간, 전체 시간에 대한 리빌드에 할당되는 시간, 리빌드 쓰레드 수 및 일련의 이벤트의 전체 기간과 같은 파라미터에 기초하여서 경험적으로 컴퓨팅되며, 가령 일련의 이벤트의 전체 기간은 일 바람직한 실시예에서 매 70초로 주기화된다. 드라이브 거동, 리빌드 IO 및 호스트 IO의 워크로드 및 로컬리티를 포함하는 호스트 IO 프로파일은 드라이브의 재배열 알고리즘에 영향을 주고 리빌드 IO 레이턴시를 초래하는 중요한 요소들이다. 이러한 요소들은 리빌드 IO

의 평균 레이턴시를 측정함으로써 경험론적 알고리즘 내에 계산을 위해서 포함될 수 있다.

[0017] 본 발명의 다른 측면은 리빌드가 호스트 IO에 비해서 얼마나 신속하게 발생하는지 및 리빌드가 호스트 IO 동작에 얼마나 영향 또는 임팩트를 주는지를 나타내는 가령 "임팩트 없음", "로우 임팩트", "하이 임팩트" 또는 "최대 임팩트"와 같은 옵션 메뉴로 선택 사양적으로 제공되는 리빌드 레이트를 사용자가 선택하거나 구성할 수 있다는 것이다.

[0018] 본 발명의 다른 측면은 임의의 하드웨어에 적응되도록 자체적으로 계속하여서 재캐리브레이션되는 경험론적 리빌드 방식에 관한 것이다. 일 바람직한 실시예에서, 본 발명의 방법은 매 70초마다 반복된다.

발명의 효과

[0019] 호스트 IO 상태 하에서 개선된 RAID 리빌드를 위한 시스템 및 방법은 리빌드 시간을 크게 개선하고 호스트 IO 고갈을 방지한다. RAID의 일부인 드라이브 내의 큐는 리빌드 IO 요청 및 호스트 IO 요청을 저장하는데 사용되며, 리빌드 IO는 드라이브 큐의 헤드에 발행된다. 드라이브로의 리빌드 요청은 지연 시간만큼 지연된다. 이러한 지연으로 인해서, 본 발명에 따라서 의도하지 않는 부작용이 발생하지 않게 되며 이로써 이러한 부작용이 일으킬 수 있는 RAID 시스템에 대한 호스트 IO 고갈이 방지된다. 이 지연은 디스크 헤드 응답 시간, 리빌드에 할당되는 시간, 리빌드 쓰레드 수, 드라이브 거동, 리빌드 IO 프로파일, 워크로드 및 로컬리티를 포함하는 호스트 IO 프로파일, 일 바람직한 실시예에서 매 70초로 주기화되는 일련의 이벤트의 기간 등과 같은 다수의 변수로부터 경험적으로 컴퓨팅된다. 리빌드 IO 프로파일은 리빌드가 호스트 IO에 비해서 얼마나 신속하게 발생하는지 및 리빌드가 호스트 IO 동작에 어느 정도의 영향 또는 임팩트를 주는지에 대한 가령 "노 임팩트", "로우 임팩트", "하이 임팩트" 및 "최대 임팩트"와 같은 다수의 옵션으로서 제공되는 리빌드 레이트를 갖는다.

[0020] 중한(heavy) 호스트 IO 상태 하에서의 소정의 구성에서는, 본 발명의 방법 및 장치를 사용하는 리빌드 시간은 10배까지 개선될 수 있다.

[0021] 본 명세서에서 기술된 본 발명으로부터 개시되고 내재된 상술한 장점들 및 다른 다수의 장점들 모두는 종래 기술에 비해서 개선된 바들을 낳는다.

[0022] 본 발명의 상술한 특징 및 다른 수많은 특징 및 이와 관련된 장점들은 첨부 도면을 참조하여서 다음의 상세한 설명 부분을 고려하면 명확해질 것이다.

도면의 간단한 설명

[0023] 본 발명의 바람직한 실시예들에 대한 상세한 설명이 첨부 도면을 참조하여서 이루어질 것이다. 본 발명을 실시하는 현재 알려진 최상의 모드에 대한 상세한 설명이 이루어질 것이다. 이 설명 부분은 한정적인 방식으로 해석되지 말아야 하며 단지 본 발명의 전반적인 원리를 예시적으로 설명하기 위한 것일 뿐이다. 본 상세한 설명 부분의 소제목 및 구성은 단지 편리성을 위한 것일 뿐이며 본 발명을 한정하지 않는다.

도 1은 본 발명의 전반적인 전체 흐름도이다.

도 2는 사용자가 규정하거나 구성 가능한 리빌드 레이트 및 리빌드 IO 대 호스트 IO를 나타내고 있다.

도 3은 동작 시에 본 발명의 바람직한 실시예를 나타내는 보다 상세한 흐름도이다.

본 기술 분야의 당업자는 본 발명의 교시 사항을 사용하여 본 발명의 기술적 사상의 범위 내에서 상기 도면에서 도시된 실시예들을 변경할 수 있다. 이들 도면에서, 유사한 참조 부호를 갖는 요소들은 이전에 규정된 동일한 요소들이 여러 상이한 도면들에 걸쳐서 존재함을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0024] 본 발명의 방법 및 장치는 본 명세서에서 기술되는 기능들을 수행하는 임의의 하드웨어 장치 또는 소프트웨어 프로그램된 장치(가령, ASIC 또는 FPGA와 같은 PLD(programmable logic device)), 하드웨어 실행 펌웨어 또는 하드웨어 실행 소프트웨어일 수 있으며 소프트웨어는 메모리 내에 존재한다. 또한, 본 명세서에서 사용되는 용어 "펌웨어"는 약칭이며 ASIC, PLD, 제어기, 프로세서 또는 컴퓨터 시스템과 같은 하드웨어 및/또는 소프트웨어의 임의의 조합으로 대체되고 이와 동의어이며 임의의 컴퓨터 또는 소프트웨어 프로그램 가능한 장치에 의해서

취해져서 이 장치를 동작시키는 인스트럭션들을 포함하는 컴퓨터 프로그램을 저장하는 컴퓨터 판독 가능한 매체를 포함한다. 본 발명이 컴퓨터 시스템에 의해서 실행될 때에, 컴퓨터 시스템은 통상적으로 메모리 내에 저장된 인스트럭션들, 모니터, 마우스 및 키보드와 같은 입출력 수단 및 임의의 필요한 특정화된 하드웨어 또는 펌웨어를 실행시키는 하나 이상의 프로세서 및 이 프로세서와 협력하는 주 메모리 및 보조 메모리를 갖는다. 본 발명을 구성하는데 사용되는 임의의 소프트웨어는 임의의 개수의 클래스, 기능, 서브루틴, 객체, 변수, 템플릿, 모듈, 코드 라인들, 코드의 일부들을 소스 코드 또는 머신 언어로 가지며 본 명세서에서 기술되고 교시되는 연속적인 스테이지들로 해서 본 발명을 실행하는 "프로세스 단계들", "단계", "프로세스", "블록", "블록 단계", "애플리케이션", "인스트럭션들", "프로그램 인스트럭션들" 또는 "모듈"을 (총괄해서 그리고 일반적으로 본 명세서의 흐름도에서 도시된 바와 같이) 구성하고 독립형 소프트웨어 애플리케이션이 되거나 다른 소프트웨어 애플리케이션 내부에서 사용되거나 이에 의해서 호출될 수 있다.

[0025] 상세한 설명 부분들 중 몇몇 부분들은 프로세스, 프로시저, 로직 블록, 기능 블록 및 동작들에 대한 다른 부호적 표현들의 차원에서 제공되며, 이러한 다른 부호적 표현들은 컴퓨터, 프로세서, 제어기 및/또는 메모리 내에서 데이터 비트, 데이터 스트림 또는 파형으로 해서 이루어진다. 본 명세서에서 기술되는 프로세스, 프로시저, 박스, 로직 블록, 기능 블록, 동작 등은 일반적으로 전기 신호 형태, 자기 신호 형태, 광학 신호 형태 또는 다른 신호 형태를 취할 수 있는 물리적 양을 물리적으로 다루는 바를 포함하며, 상기 다른 신호 형태는 컴퓨터, 데이터 프로세싱 시스템 또는 로직 회로 내에서 저장, 전달, 결합, 비교 및 이와 달리 다루어질 수 있는 신호 형태이다. 이러한 신호들을 비트, 파, 파형, 스트림, 값, 엘리먼트, 심볼, 문자, 텀(term), 수 등으로서 지칭하는 것이 원칙적으로는 통상적인 사용의 측면에서 편리할 수 있다.

[0026] 본 명세서에서 기술되는 본 발명에 있어서, 통상적인 측면의 리빌드는 전술한 바와 같이 드라이브 큐 재배열로 인해서 높은 레이턴시를 경험한다. 본 명세서에서 기술되는 방법은 SCSI IO Tag Head-of-Q을 갖는 리빌드 IO를 발행하고 이 리빌드 IO를 드라이브의 큐의 헤드에 부가하여서 이 리빌드 IO를 강제 실행시키는 것이다. 이로써, 리빌드 레이턴시가 크게 감소하며 이로써 리빌드 성능도 역시 크게 개선될 수 있다. IO는 바람직하게는 어느 곳보다 큐의 전방에서 이 SCSI IO Tag Head-of-Q 또는 SCSI IO 태그 헤드-오브-큐를 사용하여서 발행되어야 한다. 이러한 방식은 높은 레이턴시 문제를 해소하지만, 2 차적 문제점을 야기한다. 먼저, 큐의 헤드에서 리빌드 요청이 너무 적극적이어서 리빌드가 매우 신속하게 실행될 수 있지만, 이 시스템의 나머지는 데드(dead)되는 상황으로 될 수 있다. 따라서, 이러한 문제점을 풀기 위해서, 본 명세서에서 기술되는 바와 같이, 큐에 대한 지연이 도입된다. 그러나, 이러한 지연은 문제점을 낳는다. 고정된 정적 지연을 사용하면, 모든 상태 하에서 최상의 결과를 얻을 수 없으며 가변 상태에서도 리빌드는 하루 또는 이틀 정도 걸리게 된다. 따라서, 이렇게 고정된 정적 지연 대신에 본 명세서에서 개시된 바와 같이 최적 방식으로 지연이 계산되어야 한다. 최적 지연은 본 명세서에서 개시된 바와 같이 수많은 변수를 커버하고 수많은 조건 하에서 행해지는 경험론적 방법에 기초한다. RAID 사용자는 호스트 IO 임팩트 레벨(호스트 IO 임팩트에 주는 영향의 정도)에 대해서는 간단하고 편리한 방식으로 다 이얼링하면 되고 본 명세서에서 개시되는 시스템이 나머지 부분을 담당하게 된다. 지연은 본질적으로 성능을 개선하지 못하지만, 본 명세서에서 개시된 바와 같이 시스템이 최소의 허용 가능한 수행을 가지고 리빌드가 여전히 신속하게 실행되도록 보장하기 위해서는 지연은 필수 사항이다. 본 명세서에서 개시되는 지연 없이도, 리빌드는 신속하게 실행될 수 있지만, 시스템 성능 또는 수행은 이하에서 더 기술될 바와 같이 매우 악화되게 된다.

[0027] 이제 도 1을 참조하면, RAID 레벨 시스템이 호스트 IO 상태 하에서 리빌드되고 있을 때에 본 발명을 사용하여서 이 시스템이 경험론적 방식으로 리빌드하도록 하는 RAID 시스템을 위한 본 발명의 전반적인 전체 흐름도가 도시되어 있다. 전반적으로, 본 발명을 실현하는 프로그램 흐름(10)은 3 단계를 갖는다. 본 발명은 펌웨어, 하드웨어, 소프트웨어 또는 이들의 임의의 조합(이하에서는, 약칭으로 펌웨어로 지칭됨) 내에 상주할 수 있다.

[0028] "리빌드 IO 응답 측정"으로 표시된 제 1 단계(12)에서, 프로그램은 평균 리빌드 IO 응답 시간을 컴퓨팅하여서 본 솔루션 또는 해법의 기초를 깬다. 가령, 이 프로그램은 본 발명에 따른 기술을 사용하지 않는 통상적인 방식을 사용하는 리빌드인 정상적인 리빌드 또는 통상적인 리빌드 동안에 하드 드라이브의 헤드가 데이터를 판독하는 응답 시간을 판독하여서 평균 응답 시간을 밀리초(ms)로 해서 컴퓨팅할 수 있다. 이를 위해서, "평균 리빌드 판독 응답 컴퓨팅"이 입력된다(14). 평균 리빌드 응답 시간은 임의의 소정의 시간에 호스트 IO 워크로드, 로컬리티 및 드라이브의 큐 관리 효율을 실제로 측정한다. 본 명세서에서 기술되는 바와 같이, 평균 리빌드 응답 시간이 소정의 경험적으로 결정된 임계치보다 작으면, 종래 기술의 통상적인 리빌드 메커니즘이 리빌드에 있어서 충분히 빠를 것이지만 그렇지 않고 임계치보다 크게 되면, 리빌드 커맨드는 보다 우수한 리빌드 성능을 위해서 즉각적 프로세싱을 위해서 RAID 드라이브의 큐의 헤드(프론트)로 가게 된다.

[0029] "큐 헤드에 부가하는데 있어서 최적 지연 T ms을 캘리브레이션함"이라고 표시된 제 2 단계(16)에서, 프로그램은

밀리초 단위로 최적 지연 T을 경험론적 방식으로 컴퓨팅하는데 이 지연은 드라이브 큐로의 리빌드 IO의 발행을 지연시킨다. 헤드-오브-큐 (큐의 헤드) 큐는 드라이브 벤더에 의해서 구현되는 개별 드라이브의 펌웨어 내의 개별 드라이브를 위한 큐이다. 이 단계 동안에, RAID 시스템은 특히 큐를 연루시키는 본 발명에 따른 비 통상적인 방식으로 리빌드를 수행한다. 이 큐는 어느 IO 요청이 어떠한 순서로 (큐 형태로) 프로세싱할지를 RAID 시스템에게 말하는데, 가령 리빌드 IO가 호스트 IO보다 먼저 프로세싱되거나 아니면 호스트 IO가 리빌드 IO보다 먼저 프로세싱되거나 2 개의 리빌드 IO가 먼저 프로세싱되고 이후에 호스트 IO가 프로세싱되는 등을 말할 수 있다. 헤드-오브-큐 태깅된 리빌드 IO는 드라이브 큐를 프로세싱/재배열하기 이전에 이 리빌드 IO를 즉시 프로세싱하도록 드라이브에게 말한다. 지연 T는 리빌드 IO가 드라이브 큐로 발행되기 이전에 이루어지는 지연의 시간이다 (단위는 밀리초임). 이러한 지연 또는 일시 중지 동안에, 어떠한 리빌드 IO도 큐의 헤드 또는 프론트에 발행되지 않는다. 이로써, 드라이브는 큐는 실행하고 헤드-오브-큐 리빌드 IO가 드라이브를 실효적으로 취하여서 큐의 나머지를 고갈시키는 것을 방지한다.

[0030] 리빌드 IO 커맨드 또는 리빌드 IO가 드라이브의 큐의 프론트에 발행되는 것을 지연하기 위해서, 얼마 정도의 최적 지연 시간이 필요한 지를 컴퓨팅하거나 지연을 캘리브레이션하는 바와 관련된 도 1의 단계(16)에서의 요소들은 몇 개의 요소들을 포함한다.

[0031] 제 1 요소는 큐를 연루시키는 헤드-오브-큐 리빌드 동안에 하드 드라이브 헤드가 데이터를 판독하기 위한 평균 리빌드 판독 응답 시간을 포함한다. 이는 "평균 리빌드 헤드-오브-큐 판독 응답 시간 컴퓨팅"으로 입력된다 (18).

[0032] 단계(16)에서의 지연 캘리브레이션과 관련된 제 2 요소는 바람직하게는 사용자가 선택하거나 구성할 수 있는 리빌드 IO 프로파일로부터 리빌드를 위해 할당된 시간을 컴퓨팅하는 바이며, 이는 "리빌드 IO 프로파일 컴퓨팅"으로 입력된다(20). 이 리빌드 프로파일은 리빌드가 얼마나 적극적으로 또는 공격적으로 진행될지를 결정하는 다수의 리빌드 레이트를 포함한다. 이 리빌드 프로파일은 또한 총 시간에 대해서 리빌드에 할당된 시간으로 간주될 수 있다. 본 명세서에서 개시되는 일 실시예에서, 사용자는 일련의 메뉴 선택 사항으로부터 일 프로파일을 선택할 수 있는데, 가령 리빌드 IO에게 큰 우선 순위가 주어지지 않고 호스트 IO 요청에 보다 큰 우선 순위가 주어지는 로우 임팩트 리빌드(Low)와 같은 최소 리빌드 레이트를 선택하거나, 동일한 우선 순위가 리빌드 IO 요청에 주어지는 하이 임팩트 리빌드(High)를 선택하거나, 호스트 IO보다 리빌드 IO에 보다 큰 우선 순위가 부여되거나 리빌드 IO에 최대 우선 순위가 부여되는 최대 임팩트 리빌드(Max)를 선택할 수 있다. 이러한 선택은 자동적으로 수행될 수도 있다. 그러나, 리빌드 프로파일에서, 이러한 우선 순위 부여는 실시예에 따라서 다르며 임의의 수의 단계적 우선 순위 부여가 존재할 수 있거나 제로 값 또는 최소 값에서 최대 값까지 연속적으로 순조롭게 변할 수도 있다. 또한, 리빌드 프로파일 레이트 및 할당 시간은 사용자 입력 없이도 자동적으로 결정되거나 사용자 입력과 관련하여서 자동적으로 결정될 수도 있다. 가령, 리빌드 프로파일 레이트 및 할당 시간은 전문가 시스템에서와 같이 이력 데이터를 사용하는 룩업 테이블로부터 자동적으로 구해질 수 있는데, 다수의 유사한 저장 시스템으로부터 수집된 이력 데이터 및/또는 특정 하드 드라이브 제조자의 데이터에 기초하여서 리빌드 프로파일이 결정될 수 있다.

[0033] 단계(16)에서의 지연 캘리브레이션 동안에 지연을 결정하는 바와 연관된 제 3 요소는 리빌드 쓰레드(rebuild thread)의 수를 컴퓨팅하는 바이며, 이는 임의의 특정 하드 드라이브 제조자에 따라서 그 구현이 상이하며 도 1에서는 "리빌드 쓰레드 수 컴퓨팅"으로서 입력된다(22). 드라이브들은 리빌드되고 있는 가령 가상 그룹의 스트라이프 크기와 같은, 디스크 그룹의 IO 크기의 입도 및 가용한 시스템 자원들에 기초하여서, RAID 시스템 내의 펌웨어에 의해서 그 구현이 결정되는 일련의 하나 이상의 데이터 쓰레드로 리빌드될 수 있다. 가령, 1 메가 스트라이프 크기 가상 드라이브에 있어서, 디스크로의 각 IO는 캐시가 사용될 것을 요구하기 때문에, 오직 하나의 쓰레드가 한 번에 발행될 수 있다. 그러나, 64 K 스트라이프 크기 가상 드라이브에서, 8 개의 쓰레드가 실행될 수 있는데 그 이유는 이러한 8 개의 IO는 디스크당 오직 512 K의 캐시를 사용할 것이기 때문이다. 이 모든 것들은 그 자체로 알려져 있으며 실시예에 따라서 변화될 수 있다.

[0034] 단계(16)에서 지연 캘리브레이션과 관련된 제 4 요소는 리빌드 할당 시간의 길이로 하나 이상의 상수를 사용하는 바이며 일 실시예에서 이러한 길이는 IO 응답에 있어서 1000 ms이지만, 일반적으로 특정 구현예에 따라서 이러한 길이는 가변적일 수 있다. 또한, 본 발명을 실시하는 전체 방법의 기간은 다른 시간으로 일정할 수 있으며 일 실시예에서 본 명세서에서 기술되는 바와 같이 70 초일 수 있지만 일반적으로 보편성이 손실되지 않는 이상 임의의 길이일 수 있다.

[0035] 마지막 단계인 제 3 단계(24)는 도 1에서 "리빌드 IO를 매 지연 T ms로 해서 헤드-오브-큐에 발행"으로 표시되

어 있으며, 이 단계에서 본 발명은 프로그램은 본 발명의 펌웨어가 제 2 단계(16)에서 결정된 지연 시간 T_{ms} 로 해서 리빌드 IO 커맨드를 큐의 프론트(헤드-오브-큐)에 발행하도록 함으로써, 이전 단계에서 컴퓨팅된 지연을 실제로 구현한다. 이러한 지연 동안에, 큐에 어떠한 리빌드 IO도 전송되지 않게 된다.

[0036] 도 1의 기술을 사용하면, 본 발명을 사용하는 RAID 시스템에서 리빌드 동안에 주목할만하게 성능이 개선된다. 그 효과를 말하자면, 시뮬레이션이 아래의 표 A 및 표 B에 나타나 있다. 몇몇 경우에, 도시된 바와 같이, 10 배 정도의 성능 개선이 이루어질 수 있다. 표 A는 본 발명에 따른 리빌드 프로세스를 나타내고, 표 B는 통상적인 리빌드 프로세스에 따른 리빌드를 나타내고 있다.

표 A: 본 발명을 사용하는 리빌드 - 헤드 오브 큐 리빌드 IOPS 및 응답 시간 (ms) 64K				
넘버	호스트 IO	IOPS	평균 응답 시간 (ms)	MB/s
1	IO 없음	220	2 - 40	13.74
2	1QD 니어	156	4	9.74
3	1 QD 파	146	9	9.12
4	16 QD 니어	140	10	8.74
5	16 QD 파	130	17	8.12
6	256 QD 니어	128	16	7.99
7	256 QD 파	123	20	7.68

표 B: 본 발명을 사용하지 않는 통상적 리빌드 - 리빌드 IOPS 및 응답 시간 (ms) 64K				
넘버	호스트 IO	IOPS	평균 응답 시간 (ms)	MB/s
1	IO 없음	1398	0	87.29
2	1 QD 니어	540	4	33.72
3	1 QD 파	422	12	26.35
4	16 QD 니어	102	77	6.37
5	16 QD 파	106	80	6.62
6	256 QD 니어	11	1200	0.69
7	256 QD 파	90	97	5.62

[0038]

[0039] 이 표들에 있어서 중요 사항은 다음과 같다: 제 1 열의 "넘버"는 고려 중인 케이스의 넘버를 말하며, 각 표에서 7 개 예시적인 케이스가 존재한다. 제 2 열에서, 호스트 IO 니어(near)에서 접미사 니어는 호스트 영역에 가까운 리빌드 영역을 말한다. 이 영역은 표 B의 제 6 행의 중한 호스트 IO 워크로드 하에서, 호스트 영역으로부터 더 멀리 떨어져 있는 리빌드 영역인, 제 2 열에서 접미사로 지정된 "과(far)" 영역보다 높은 레이턴시를 경험하는 리빌드 영역이 되는 것으로 된다. 수많은 제조자들이 펌웨어 의존적인 IO 큐들을 최적화하는 독점적 방법들을 가지기 때문에, 이러한 니어 및 과 현상에 대한 보편적인 이유는 명확하게 주어질 수 없지만, 이러한 현상이 관측될 수 있다. 제 2 열에서, "호스트 IO"는 IO 요청에 응답하는 지정된 호스트 IO이며, 다음과 같은 바와 함께 지정된다: QD는 큐 깊이를 말하며, 이는 호스트 IO 요청 동안에 얼마나 많은 호스트 IO들이 존재하는지를 나타내는 척도이다. "1 QD"는 전체 시스템에 실제로 어떠한 임팩트도 주지 않는 매우 약한 호스트 IO 드라이브 채배열 방식을 지정한다. 1 QD이면, 드라이브 IO 큐는 오직 하나의 호스트 IO를 가지며 수많은 리빌드 IO로 채워질 수 있다. 이와 대조적으로, 16 QD는 보다 통상적인 호스트 IO 워크로드이며 소정의 시간에 16 개의 IO가 남아 있다. 1 QD와는 전혀 반대되는 바는 256 QD이며, 이는 소정의 시간에 수많은 호스트 IO가 남아 있으며 중한 워크로드(heavy workload)를 가진다: 256 QD는 이례적인 것이지만 최악의 리빌드 시간을 초래한다.

[0040] 제 3 열에서, "IOPS"는 Input/Output rebuild Operations Per Second이다. 제 4 행에서, 평균 응답 시간은 리빌드 IO에 대한 드라이브의 평균 응답 시간을 밀리초(ms) 단위로 나타낸다. 마지막 열에서, "MB/s"는 리빌드 IO에 대한 데이터 전달 레이트 처리량을 초당 메가바이트 단위로 나타낸 것이다.

[0041] 표 A 및 표 B 간의 비교 결과에서 볼 수 있는 바와 같이, 본 발명을 사용하면, 통상적인 리빌드 방식에 비해서

성능이 크게 개선될 수 있다. 가령, 넘버 5의 케이스를 살펴보면, 16 QD 파의 적당한 로드인 경우에, 통상적인 리빌드에서는 응답 시간이 80 ms로 높다. 그러나, 본 발명의 리빌드 방식을 사용하면, 이 동일한 넘버 5의 케이스에 있어서, 응답 시간은 17 ms로서 통상적인 리빌드 경우에서보다 79 %가 감소되었으며, IOPS 및 처리량은 각기 106에서 130으로 그리고 6.62 MB/s에서 8.12 MB/s로 23 % 만큼 증가하였다.

[0042] 마찬가지로, 넘버 7의 경우에서와 같이 256 QD 파와 같은 중한 로드에서, 본 발명을 사용하는 것이 통상적인 리빌드 방식을 사용하는 것보다 성능이 개선되는데, 즉 IOPS가 90에서 123으로 증가하고 처리량도 5.62에서 7.68 MB/s로 37 % 만큼 증가하며, 평균 응답 시간은 97에서 보다 바람직하게는 20으로 79 % 만큼 감소하였다.

[0043] 약한 IO 로드에서 본 발명을 사용하는 바에는 이점이 없는데, 본 발명은 이를 인식하였고 본 명세서에서 더 기술될 바와 같이 이러한 바를 본 발명의 동작 방법에 고려하였다. 가령, 표 A 및 표 B에서 넘버 1 케이스에서 어떠한 리빌드 IO도 없는 경우에, 본 발명과 통상적인 방식을 비교하면, 본 발명에서는 어떠한 이점도 없으며 본 발명을 사용하면 실제로 성능이 더 안 좋아짐을 알 수 있다: 마찬가지로, 넘버 2의 1 QD 니어의 경우에도, 통상적인 방식에 비해서 본 발명을 사용하면 어떠한 이점도 없음을 알 수 있다(표 A의 넘버 2의 경우에, 50 ms의 최소 지연이 헤드-오브-큐 리빌드 요청들 간에 부가된다). 그러나, 넘버 3의 경우에 10 QD 파에서, 본 발명을 사용하면 응답 시간에 있어서 약간의 이점이 있는데 응답 시간이 12 ms에서 9 ms로 감소하지만, 처리량에 있어서는 어떠한 이점도 없는데 본 발명을 사용하면 처리량이 9.12 MB/s이지만 통상적인 기술을 사용하면 처리량이 26.35 MB/s로 보다 높다. 또한, 각 표에 있어서, "니어" 조건이 "파" 조건보다 통상적으로 응답 시간이 더 빠르지만, 큰 예외가 존재한다: 표 B에서, 통상적인 리빌드 방식에 있어서, 넘버 6의 256 QD 니어와 넘버 7의 256 QD 파를 비교하면, 중한 로드에서, 아주 높은 레이턴시가 급격하게 발생한다. 니어 IO 요청에 대한 통상적인 보다 신속한 응답 시간 대신에, 반대로 응답 시간이 매우 느려지며, 반대로 파의 경우에 평균 응답 시간은 니어의 경우에 있어서보다 12 배나 신속해지는데 니어의 경우에는 응답 시간은 1200 ms인데 비해서 파의 경우에 97 ms가 된다. 이렇게 응답 시간이 반전되는 바는 본 발명의 방법을 사용하는 표 A에서 넘버 6와 넘버 7을 비교하면 볼 수 있는 바와 같이 본 발명이 양호하게 해결할 수 있다. 통상적인 리빌드 방식인 표 B에 비해서 표 A에서 256 QD 니어의 경우의 성능은 매우 우수해지며 통상적인 리빌드 방식이 1200 ms인데 비해서 본 발명에서는 16 ms가 평균 응답 시간이 되며 이는 1200/16인 75배나 신속해진 것이며 98.7 %가 감소된 것이다. 마찬가지로, 처리량은 표 B의 0.69 MB/s에서 표 A의 7.99 MB/s로 크게 증가하며 그 크기 차를 비교하면 11배나 증가한 것이다. 마찬가지로, 표 A에서의 넘버 7을 표 B에서의 넘버 5와 비교하면, 중한 로드의 경우에도, 본 발명에 따르면 응답 시간이 20 ms인 반면에, 통상적인 리빌드 프로세스를 사용하면 적당한 정도의 로드의 경우에도 응답 시간이 20 ms로 열등해진다.

[0044] 표 A 및 표 B에서 볼 수 있는 바와 같이, 로드가 약하거나 없으면서 큐 깊이가 작은 경우에, 본 발명의 방법은 통상적인 방식에 비해서 실제로 어떠한 이점도 낳지 못한다. 이러한 이유로 인해서, 더 상세하게 설명될 바와 같이, 리빌드를 실행하는 동안에 본 발명은 리빌드 레이트가 얼마이며 응답 시간이 얼마인 바를 고려하며, 일바람직한 실시예에서는, 리빌드 레이트가 기준치에 비해서 33% 크거나 응답 시간이 45 ms보다 큰 경우에만 지연 시간을 갖는 헤드-오브-큐 리빌드를 수행한다. 이로써, 본 발명은 표 A에서 넘버 4 내지 넘버 7과 같은 보다 큰 넘버의 행에 의해서 규정된 영역에서 보다 양호하게 동작할 것이다.

[0045] 이제 도 2를 참조하면, 사용자 규정 가능하거나 구성 가능한 리빌드 레이트가 그래픽 사용자 인터페이스(GUI)(200)를 통해서 어떻게 선택될 수 있는지가 도시되어 있는데, 본 발명에서 사용자에 의해서 조작되는 이 GUI는 필요 없을 수도 있으며, 이렇게 GUI를 예시하는 것은 리빌드 레이트가 구성 가능하면서 변경 가능한 방식으로 어떻게 동작하는지를 잘 설명하기 위한 개념적 도구로서 고려되어야 하며, 실제로는 이러한 리빌드 레이트는 자동으로 선택 및 변경되고/되거나 GUI를 통하거나/하고 인간 운영자에 의해서 선택 및 변경되기보다는 고정된 시퀀스로 해서 선택될 수도 있다. GUI(200)는 사용자가 본 발명에 따라서 RAID 시스템을 어떻게 리빌드하기를 원하는지에 있어서 강도의 다양한 레벨에 대응하는 다수의 버튼(210, 215, 220, 225)을 가지며, 가령 이러한 강도의 다양한 레벨은 "로우", "하이", "맥스" 및 "노(No)"가 있으며, 이들은 각기 로우 값 또는 로우 임팩트, 하이 값 또는 하이 임팩트, 최대 값 또는 최대 임팩트 및 리빌드 없음 또는 임팩트 없음에 대응하며, 여기서 임팩트는 호스트 IO에 미치는 임팩트이다. 가령, 일바람직한 실시예에서, 이 4 개의 레벨의 리빌드 레이트는 "노 임팩트", "로우 임팩트", "하이 임팩트" 및 "최대 임팩트"로서 사용자에게 제공되며, 여기서 임팩트는 호스트 IO가 리빌드에 의해서 얼마나 영향을 받는지를 말한다.

[0046] "노 임팩트"의 경우에, 본 발명의 RAID 시스템은 통상적인 리빌드를 따라서 동작한다. 이 옵션의 경우에, 본 발명에 의해서 호스트 성능에 최소한의 임팩트가 존재하는데 간단히 그 이유는 본 발명이 사용되지 않기

때문이다. 이 옵션에서, 리빌드 시간은 사용자에게 큰 문제 거리가 되지 못한다.

[0047] 나머지 3 개의 옵션은 본 발명에 따른 리빌드 동작에 대한 것이다. "로우 임팩트"의 경우에, 호스트 IO 성능은 도 2의 막대 그래프에 의해서 도시된 바와 같이 도 2에 도시된 바람직한 실시예에서 "노 임팩트"의 1/3 만큼 임팩트를 받는데, 이 도 2에서, 로우 임팩트에서 헤드-오브-큐 리빌드 IO는 총 할당된 시간 1000 ms 중 333 ms를 차지한다. 이는 도 2에서 240의 참조 부호로 표시되어 있으며, 큐의 헤드에 발행되는 리빌드 IO(도 2에서는 "헤드-오브-큐 리빌드 IO"로 표시됨)는 1000 ms 중 333 ms가 동작 시간으로 실효적으로 주어지며 나머지 667 ms는 호스트 IO 요청(도 2에서는 "호스트 IO"로 표시됨)을 서비스하기 위한 본 발명의 펌웨어에게 할당된다. 도 2에서 참조 부호 250으로 표시된 "하이 임팩트" 옵션의 경우에, 본 발명의 리빌드 IO가 전체 1000 ms 중 500 ms, 즉 대략 절반의 시간의 실효 시간 동안 동작하며, 리빌드 IO는 큐의 헤드에 발송되며 나머지 시간은 호스트 IO에 할당된다. 도 2에서 참조 부호 260으로 표시된 "최대 임팩트" 옵션의 경우에, 본 발명은 1000 ms 중 실효 시간 666 ms 동안에 큐의 헤드에 리빌드 IO를 발행하며, 이 실효 시간은 전체 시간의 67 %이며 나머지 33 %의 시간은 호스트 IO를 서비스하는데 할당된다. 리빌드 IO들 모두가 한번에 특정 기간 동안에 발행되지 않는 대신에 호스트 IO들 간에 산재될 수 있다. 리빌드 IO들은 예상된 리빌드 레이트를 고려하여서 효과적인 서비스 시간을 달성하기 위해서 적합한 지연으로 해서 발행된다.

[0048] 도 2에서는 오직 4 개의 선택 옵션 버튼이 주어져서 성능을 이산적으로 단계별로 조절하였지만, 이 옵션의 개수는 이에 한정되지 않고 임의의 개수가 될 수 있다. 실제로, 사용자는 리빌드 레이트를 설정하기 위해서 슬라이더(230)와 같은 슬라이더 바를 사용하거나 사용자는 제로 퍼센트에서 최대 퍼센트까지 레이트를 설정할 수 있다. 또한, 리빌드 레이트는 시계가 가리키는 시각, 리빌드를 수행한 과거 경험, 제조자의 데이터 등과 같은 입력 데이터에 기초하여서 인공 지능 에이전트를 사용하여서 자동으로 설정될 수도 있다.

[0049] 이제 도 3을 참조하면, 동작 시에 본 발명의 바람직한 실시예를 나타낸 보다 상세한 흐름도가 존재한다. 도 3의 흐름도는 제로 내지 70 초의 시간에 걸쳐서 주기화되는 방식으로 캘리브레이션, 재캘리브레이션 및 리빌드가 어떻게 동작하는지를 나타내고 있으며, 이러한 70초의 기간은 T0 내지 T70으로 표시되며, 이 전체 70초 기간은 다시 T0 내지 T5의 제 1 스테이지, T5 내지 T10의 제 2 스테이지 및 T10 내지 T70의 제 3 스테이지로 분할된다. 도 3에 도시된 바와 같은 본 발명의 방법은 총 70 초로 구성되는 3 개의 스테이지로 분할되지만, 일반적으로 본 명세서에서 교시된 바에 따라서 임의의 기간이 선택될 수 있다. "정상적인 리빌드 IO 발행"이라고 표시된 제 1 스테이지(310)는 T0 내지 T5의 5 초 동안 계속되는 측정 스테이지이다. "리빌드 IO를 헤드-오브-큐에 발행"이라고 표시된 제 2 스테이지(315)는 T5 내지 T10의 5 초 동안 계속되며 리빌드 IO 요청을 헤드-오브-큐에 발행하는 이전에 그 사이에 이루어지는 지연 델타 (또는 T)를 계산하는 스테이지이며 조기에 설명된 바와 같이 이는 리빌드 레이트를 완화시키기 위한 것이다. 이러한 지연은 헤드 응답 시간, 리빌드 할당 시간 및 리빌드 쓰레드 수와 같은 파라미터들을 고려한다. "매 지연 ms로 해서 리빌드 IO를 헤드-오브-큐에 발행"이라고 표시된 마지막 스테이지인 제 3 스테이지(320)는 T10 내지 T70의 60 초 동안 지속되며 매 델타 ms의 지연으로 해서 헤드-오브-큐에 리빌드 IO를 발행함으로써 제 2 스테이지에서 계산된 지연을 구현하며, 이 계산된 지연 델타는 리빌드 IO 실행이 큐의 프론트 또는 헤드에 발행된 이후로부터 지연되는 시간이다. 이어서, 이 사이클 또는 주기가 반복된다. 이로써, 본 발명의 방법은 경험론적 방식으로 제공된 파라미터들에 따라서 재발행 IO들을 동적으로 재캘리브레이션한다.

[0050] 이로써, 전체 70 초의 5 초이므로 $5/70 = 7\%$ 을 차지하는 제 1 스테이지(T0 내지 T5)에서, 본 발명에 따른 RAID 시스템에서 동작하는 하드웨어 및/또는 소프트웨어(이하에서는 약칭으로 펌웨어로 지칭됨)가 리빌드가 통상적인 방법을 사용하여서 수행되는 경우, 즉 "정상적인 리빌드 IO 발행"(310)에서 표시된 바와 같은 정상적인 리빌드 IO로서 지칭되는, 본 발명을 포함하지 않는 통상적인 기술을 사용하여서 리빌드가 수행되는 경우의 평균 리빌드 응답 시간을 컴퓨팅한다(도 3에서는 박스(325)로 해서 "평균 리빌드 관독 응답 시간 컴퓨팅"으로 표시됨). 이어서, "리빌드 레이트 < 33 % || 응답 시간 < 45 ms ?"로 표시된 결정 박스(330)에서 표시된 바와 같이, 펌웨어는 통상적인 리빌드 하에서 리빌드 레이트가 33 %보다 작은지의 여부(33 % 보다 작다면, 사용자가 호스트 IO 요청에 어떠한 임팩트도 주지 않고자 함을 표시함) 또는 리빌드 레이트가 45 ms보다 작은지의 여부(45 ms보다 작으면, 통상적인 리빌드 방법이 이 시스템에 대해서 최적임을 표시함)를 체크할 것이다. 그렇다면, 결정 분기점 "예"로 표시된 바와 같이 펌웨어는 계속하며 이 프로세스는 "아니오"가 결정 박스(330)에서 만족될 때까지 제 1 스테이지를 반복하며, 결정 박스(330)에서 "아니오"가 되면, 프로세스는 제 2 스테이지(T5 내지 T10)로 진행한다.

[0051] 전체 70 초의 5 초이므로 $5/70 = 7\%$ 을 차지하는 제 2 스테이지(T5 내지 T10)에서, 펌웨어는 전술한 바와 같이 리빌드 IO 고갈/ 리빌드 IO의 높은 레이트를 방지하기 위해서 큐 내의 리빌드 IO 커맨드들 간에 필요한 최적

지연을 컴퓨팅할 것이다. 이를 위해서, 펌웨어는 그 내에 있는 실제 환경으로부터의 다양한 파라미터들을 컴퓨팅하는데, 도 3의 박스(340)에서 도시된 바와 같이, 제 2 스테이지 전체 기간 동안 평균 리빌드 헤드-오브-큐 관독 응답 시간을 컴퓨팅하며 이 스테이지의 종료 시에 이 평균을 취할 것이다. 다양한 요소들이 전술한 바와 같이 이러한 컴퓨팅에서 고려될 수 있다.

[0052] 먼저, 시스템이 큐의 헤드에 리빌드를 발행할 때에 하드 드라이브 헤드의 평균 관독 응답 시간은 5 초 동안 컴퓨팅되며 펌웨어에 의해서 파라미터로서 관독된다. 이 큐는 RAID 시스템 내의 특정 드라이브를 위한 큐 내에 리빌드 IO 및 호스트 IO를 저장하기 위한 종래 기술에서 알려진 바이머 리빌드 IO 요청은 큐의 헤드 또는 프론트(도 3의 헤드-오브-큐)로 간다. 리빌드 IO 요청은 호스트 IO 요청과 함께 큐로 전송되며 리빌드 IO 요청들 간에 지연이 부여되며 100 ms의 디폴트 지연이 초기에 선택되거나 이 프로세스가 이전의 70 초 사이클로 반복되었으면 제 2 스테이지에서 계산된 이전 지연 시간으로 된다. 하드 드라이브 헤드의 관독 응답 시간은 이 스테이지 동안에 펌웨어에 의해서 관독되어서 평균 처리된다. 이 평균화된 파라미터는 도 3에서 "평균 리빌드 헤드-오브-큐 관독 응답 시간(resp) 컴퓨팅"으로 표시된 박스(340)에서 "resp"로 표시되어 있다.

[0053] 둘째로, 리빌드 프로파일이 결정되고 리빌드 할당 시간이 호스트 IO 요청 및 리빌드 IO 요청을 모두 포함하는 총 시간에 대해서 확정된다. 사용자 및/또는 시스템이 전술한 바와 같이 33 % 내지 65 %의 레이트를 갖는 "로우 임팩트"를 선택하면, 이 파라미터가 변수 "리빌드 할당 시간(rt)"로 선택되는데, 여기서 "rt"는 다양한 리빌드 프로파일 옵션에 대한 ms 단위로서 (1) 33% 내지 65% 리빌드 레이트(로우 임팩트 옵션)에서는 333 ms에 대응하고, (2) 66% 내지 98% 리빌드 레이트(하이 임팩트 옵션)에서는 500 ms에 대응하고, (3) 99% 내지 100% 리빌드 레이트(최대 임팩트 옵션)에서는 666 ms에 대응한다. 이러한 다양한 값들은 실시예에 따라서 특정되며 전술한 바와 같이 임의의 개수의 값이 될 수 있으며 단지 3 개로만이 아니라 단계적 레벨로 선택되거나 최소치에서 최대치까지 연속적으로 선택될 수도 있다.

[0054] 셋째로, 리빌드 IOPS(rIOPS)가 다음과 같은 공식에 따라서 도 3의 박스(340)에서 표시된 바와 같이 컴퓨팅된다:

[0055]
$$\text{리빌드 IOPS(rIOPS)} = \text{rt}/\text{resp}$$

[0056] 여기서, rt는 이전 단계에서 구해진 리빌드 할당 시간이며, resp는 전술한 바와 같은 평균 리빌드 헤드-오브-큐 관독 응답 시간을 말한다.

[0057] 넷째로, 아래의 공식에 따라서, 도 3의 박스(340)((rIOPS) 리빌드 쓰레드 수)에서 파라미터 리빌드 수(nr)가 고려 중인 RAID 시스템 내에서의 리빌드 쓰레드의 개수로 상기 리빌드 IOPS를 나눔으로써 컴퓨팅된다:

[0058]
$$\text{리빌드 수(nr)} = \text{rIOPS}/\text{리빌드 쓰레드 수}$$

[0059] 마지막으로, T5 내지 T10의 제 2 스테이지의 종료 시에, 리빌드들 간에서 전개될 실제 지연이 다음과 같은 공식에 따라서 컴퓨팅된다:

[0060]
$$\text{리빌드 IO 간의 지연 시간 델타 (또는 T)} = 1000/\text{nr} - \text{resp}$$

[0061] 여기서, 1000은 ms 단위로 일정한 시간이다. 이러한 시간 상수에 대해서는 임의의 값이 선택될 수 있다; IOPS는 초당이며 사용할 양호한 상수는 1 초에 대응하는 1000 ms이다. 그러나, 이러한 시간 상수가 변경되면, rIOPS도 역시 변경될 것이며, 이는 본 명세서의 교시 사항으로부터 본 기술 분야의 당업자에게 이해될 것이다. 따라서, 도 3에 도시된 T5 내지 T10으로부터의 이러한 2 스테이지 컴퓨팅이 임의의 다른 기간에 걸쳐서도 역시 수행될 수 있다.

[0062] 다시 한번 말하자면, nr는 리빌드 수이며 resp는 평균 리빌드 헤드-오브-큐 관독 응답 시간이다.

[0063] 지연 시간 델타 또는 T가 컴퓨팅되면, 본 발명의 방법은 60초를 나타내는 T10 내지 T70의 제 3 스테이지로 진행되며, 이 스테이지에서, 이전에 계산된 실제 지연 델타가 리빌드 IO 및 호스트 IO를 포함하는, RAID 시스템의 개별 드라이브 큐의 헤드로 리빌드 IO를 발행하는 것을 지연하는데 사용된다. 리빌드 IO 요청이 드라이브로 발행되기 이전에 각 리빌드 IO는 펌웨어에 의해서 지연 시간 델타만큼 지연된다. 큐에서 이러한 지연을 사용하면 전술한 바와 같이 호스트 IO 고갈이 저감되는 효과가 있다. 이 기간의 종료 시점에, 즉 T70에서, 캘리브레이션 이 시간 T0에서 새롭게 시작하며 프로세스가 반복된다. 이로써, 본 발명은 하드웨어의 실제 상태가 지연을 컴퓨팅할 때에 시간에 따라서 변화해도 이를 경험적이면서 기회적으로 고려하게 된다.

[0064] 본 명세서에서 기술되는 본 발명의 방법 및 장치는 RAID 레벨 2, 3, 4를 포함하는 RAID 레벨 1 내지 6 중 임의의 RAID 레벨에서 사용될 수 있으며 혼성 RAID 시스템에서 사용되고 소거 코드를 사용하는 새로운 타입의 고장

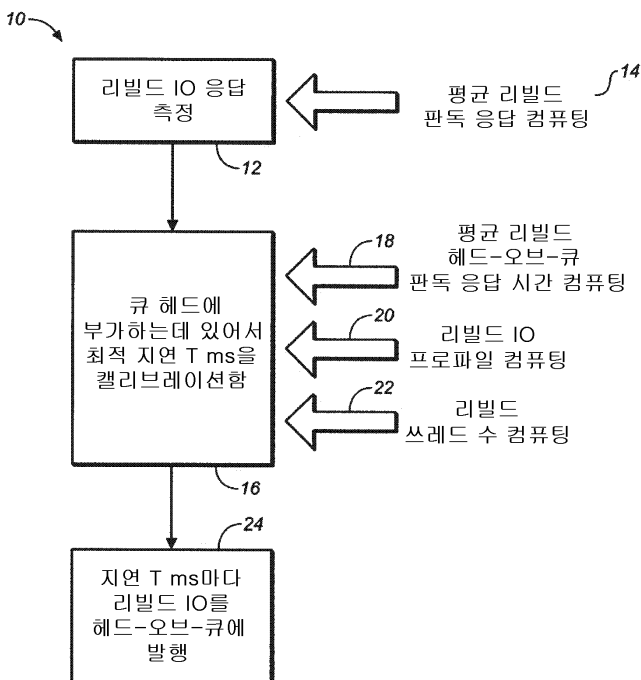
안전 및 리턴던트 저장 기술에서도 사용될 수 있다. 또한, 본 발명의 방법 및 장치는 카피백 동작에서 사용되거나, 물리적 드라이브에 IO를 발행할 필요가 있으면서 예측 가능한 방식으로 완료되고 호스트 IO에 관리 가능한 임팩트를 주는 임의의 미션 크리티컬 프로세스(mission critical process)에서 사용될 수 있으며, 이러한 프로세스는 다음으로 한정되지 않지만 매체 스캔, 일관성 체크, 초기화, 포맷팅 등을 포함한다.

[0065] 본 발명의 기술적 사상의 범위 내에서 본 명세서의 교시로부터 본 기술 분야의 당업자는 다양한 수정, 추가 또는 제거 등을 할 수 있다. 또한, 본 발명의 다양한 특징들이 기술되었지만, 이러한 특징의 서브 세트가 본 발명의 가장 기본적인 것만 남는 버전에서 사용될 수 있다. 따라서, 본 발명의 범위는 청구 범위에 의해서만 한정된다.

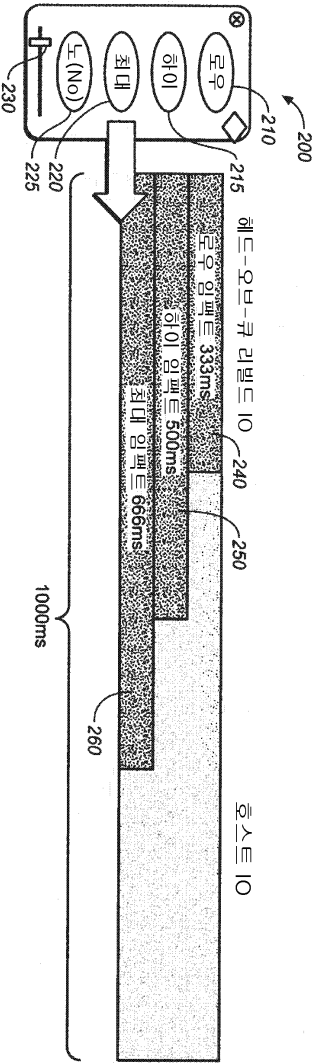
[0066] 본 발명의 범위는 모든 수정, 추가 또는 제거로 확장될 수 있으며 이하에서 기술되는 청구 범위에 의해서만 규정된다.

도면

도면1



도면2



도면3

