



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년06월09일
(11) 등록번호 10-2263040
(24) 등록일자 2021년06월03일

(51) 국제특허분류(Int. Cl.)
G06F 3/06 (2006.01) G06F 11/20 (2006.01)
(52) CPC특허분류
G06F 3/0649 (2013.01)
G06F 11/2046 (2013.01)
(21) 출원번호 10-2019-0047476
(22) 출원일자 2019년04월23일
심사청구일자 2019년04월23일
(65) 공개번호 10-2020-0124070
(43) 공개일자 2020년11월02일
(56) 선행기술조사문헌
JP2015088069 A*
(뒷면에 계속)

(73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
성균관대학교산학협력단
경기도 수원시 장안구 서부로 2066 (천천동, 성균관대학교내)
(72) 발명자
신동균
서울특별시 강남구 역삼로 314, 305동 1004호(역삼동, 개나리 푸르지오)
한규화
경기도 수원시 장안구 화산로285번길 12, 101동 1106호(율전동, 화남아파트)
(74) 대리인
리앤목특허법인

전체 청구항 수 : 총 14 항

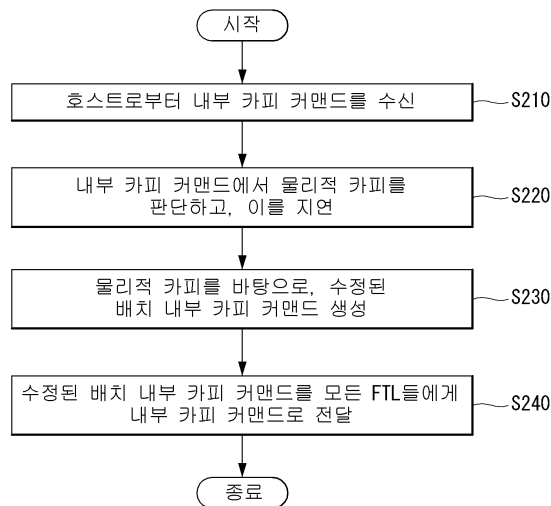
심사관 : 이후락

(54) 발명의 명칭 멀티 코어 솔리드 스테이트 드라이브의 운용 방법

(57) 요약

본 발명에 의한 멀티 코어 솔리드 스테이트 드라이브의 운용 방법은 소스 LPN 및 목적지 LPN을 포함하는 내부 카피 커맨드를 호스트로부터 수신하는 단계, 서로 다른 FTL들 간의 내부 카피 커맨드를 물리적 카피로 정의하고, 물리적 카피를 지연시키는 단계, 및 물리적 카피의 소스 LPN을 소스 LPN이 포함된 PPN로 대체하여, 수정된 배치 내부 카피 커맨드를 생성하는 단계를 포함한다.

대표도 - 도2



(52) CPC특허분류

- G06F 12/0253 (2013.01)
- G06F 3/0616 (2013.01)
- G06F 3/064 (2013.01)
- G06F 3/0659 (2013.01)
- G06F 3/0679 (2013.01)

(56) 선행기술조사문헌

- JP2007164650 A*
- YINGBIAO YAO, 외 5인. "An Advanced Adaptive Least Recently Used Buffer Management Algorithm for SSD". IEEE Access, 2019.03.12. vol. 7, doi: 10.1109/ACCESS.2019.2904639, (제33494면 내지 제33505면)*
- KR1020180076765 A*

- JP2018142240 A
- KR1020120102923 A
- KR1020130112755 A
- WO2015114809 A1
- KR1020170099746 A
- KR1020120107126 A

*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업

과제고유번호	S-2018-1129-000
부처명	-
과제관리(전문)기관명	-
연구사업명	삼성전자-성균관대 클러스터 연구과제
연구과제명	성균관대-삼성전자 클러스터 연구과제(차세대 SDDC(Software Defined Data Center)를 위한 초고속 SSD 기반 저장 시스템 구조 연구)
기여율	1/1
과제수행기관명	성균관대학교
연구기간	2018.02.01 ~ 2019.01.31

명세서

청구범위

청구항 1

둘 이상의 FTL들을 포함하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법에 있어서,

소스 LPN 및 목적지 LPN을 포함하는 내부 카피 커맨드를 호스트로부터 수신하는 단계;

서로 다른 상기 FTL들 간의 상기 내부 카피 커맨드를 물리적 카피로 정의하고, 상기 물리적 카피를 지연시키는 단계;

상기 물리적 카피의 상기 소스 LPN을 상기 소스 LPN이 포함된 PPN로 대체하여, 수정된 배치 내부 카피 커맨드를 생성하는 단계; 및

상기 수정된 배치 내부 카피 커맨드에 기초하여 가비지 컬렉션을 수행하되, 상기 목적지 LPN으로 설정된 제1 논리적 페이지에 단방향 비트를 세팅하고, 상기 제1 논리적 페이지를 포함하는 블록에 대해 기준 카운트를 증가시켜 가비지 컬렉션 대상에서 제외하는 단계를 포함하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 2

제 1 항에 있어서,

상기 물리적 카피를 지연시키는 단계는 지연된 물리적 카피 리스트를 생성하는 단계를 더 포함하고,

상기 물리적 카피는,

변경 가능한 P2L 맵핑 테이블 및 상기 지연된 물리적 카피 리스트를 이용하여 수행되어, 하나의 PPN에 다수의 카피 동작이 발생하는 것을 방지하는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 3

제 1 항에 있어서,

상기 내부 카피 커맨드를 수신하는 단계는,

상기 내부 카피 커맨드의 소스 LPN을 소스 PPN 정보로 수정함으로써, 수정된 내부 카피 엔트리를 생성하는 단계를 포함하는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 4

제 3 항에 있어서,

상기 내부 카피 엔트리는,

수정된 상기 내부 카피 커맨드로부터의 논리적 카피 및 물리적 카피를 L2P 맵핑 변경을 수행함에 따라 생성되는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 5

제 1 항에 있어서,

상기 수정된 배치 내부 카피 커맨드를 상기 FTL들에게 내부 카피 커맨드로 전달하는 단계를 더 포함하는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 6

제 1 항에 있어서,

상기 물리적 카피를 정의한 이후에,

패딩을 사용하여 상기 물리적 카피를 로직 카피로 변환하는 단계;

상기 패딩을 사용한 횟수가 미리 설정된 임계치 이상일 경우, 가장 최신의 상기 로직 카피를 상기 물리적 카피의 위치로 복구시키는 단계를 포함하는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 7

제 1 항에 있어서,

상기 내부 카피 커맨드를 수신하는 단계 이후에,

상기 FTL들에게 전달된 상태에서 내부 카피 동작이 완료되지 않은 상기 내부 카피 커맨드와 새롭게 스케줄링 할 읽기 요청이 연관된 경우, 내부 카피 커맨드에 대한 내부 카피 동작을 완료한 이후에 해당 읽기 요청을 상기 FTL들에게 전달하는 것을 특징으로 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 8

제 1 항에 있어서,

상기 내부 카피 커맨드를 수신하는 단계 이후에,

상기 내부 카피 커맨드보다 이전에 수신된 쓰기 요청을 상기 FTL들에게 전달한 이후에, 상기 내부 카피 커맨드를 상기 FTL들에게 전달하는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 9

제 1 항에 있어서,

상기 멀티 코어 솔리드 스테이트 드라이브의 운용 방법은

서든 파워 오프(SPO) 발생을 감지하는 단계; 및

상기 SPO 발생에 대응하여, 각각의 상기 FTL들은 복구 가능한 내부 카피 그룹을 선정하고, 독립적으로 복구 동작을 수행하는 SPO 복구 과정을 더 포함하는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 10

제 9 항에 있어서,

상기 SPO 발생에 대응하여,

old-PPN 정보와 목적지 LPN 정보를 바탕으로 내부 카피 커맨드와 쓰기 명령의 redo 및 undo를 수행하는 단계를 더 포함하는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 11

둘 이상의 FTL을 포함하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법에 있어서, 제1 논리적 페이지를 소스 LPN으로 하고, 제2 논리적 페이지를 목적지 LPN으로 설정한 내부 카피 커맨드를 수신하는 제1 단계;

적어도 1 차례 이상의 상기 내부 카피 커맨드에 의해서 상기 제1 및 제2 논리적 페이지가 동일한 물리적 페이지를 공유하는 제2 단계; 및

유효한 상기 물리적 페이지와 연관된 LPN 맵핑을 검색하여 가비지 컬렉션을 수행하는 제3 단계를 포함하고,

상기 제3 단계는,

상기 제2 논리적 페이지에 단방향 비트를 세팅하고, 상기 제2 논리적 페이지가 세팅된 블록에 대해서는 기준 카운트를 증가시켜서 가비지 컬렉션의 대상에서 제외시키는 단계를 포함하는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 12

제 11 항에 있어서,

상기 제2 단계는, 상기 제1 논리적 페이지가 지시하는 물리적 페이지가 다른 LPN과 맵핑되어 있음을 표시하는 플래그 비트를 세팅된 이후에 수행하는 단계를 포함하고,

상기 제3 단계는, 상기 물리적 페이지의 상기 플래그 비트가 저장된 대역의 영역을 읽어서 상기 제1 논리적 페이지를 검색함으로써, 가비지 컬렉션을 수행하는 단계를 포함하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 13

제 11 항에 있어서,

상기 제2 단계는, 상기 내부 카피 커맨드에 대한 카피 동작 및 상기 제1 논리적 페이지가 덮어쓰기(Overwrite) 된 이후에 수행되고,

상기 제3 단계는, 유효 비트맵을 통해서 상기 물리적 페이지가 유효한 페이지라고 인지하는 단계를 더 포함하는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

청구항 14

제 11 항에 있어서,

상기 제2 단계는, 상기 제2 논리적 페이지를 소스 LPN으로 하고, 제3 논리적 페이지를 목적지 LPN으로 설정된 내부 카피 동작을 수행하는 단계를 포함하는 것을 특징으로 하는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법.

발명의 설명

기술 분야

[0001] 본 발명은 멀티 코어 솔리드 스테이트 드라이브의 운용 방법에 관한 것으로, 오버헤드를 줄이면서, 내부 카피 커맨드의 원자성을 보장할 수 있는 멀티 코어 솔리드 스테이트 드라이브의 운용 방법에 관한 것이다.

배경 기술

[0002] 플래시 메모리는 저전력, 고성능, 높은 내구성의 장점을 가진다. 최근 플래시 메모리의 가격이 하락하고, 용량이 커짐에 따라 낸드 플래시 메모리 기반의 저장장치인 솔리드 스테이트 드라이브(Solid State Driver; 이하 SDD)가 기존 하드 디스크 드라이브를 대체하고 있다. SSD는 여러개의 NAND 플래시 칩, DRAM 버퍼와 컨트롤러로 구성되며, 멀티 채널과 멀티 웨이 구조를 활용하여 여러 칩에 동시에 읽기/쓰기 연산이 가능하다.

[0003] 둘 이상의 플래시 변환 계층(Flash Translation Layer; 이하, FTL)이 동작하는 멀티 코어(Multi-Core) SSD는 싱글 코어 SSD에서 나타나지 않는 문제점을 갖고 있다. 멀티 코어 SSD는 L2P 맵핑(Logical to Physical mapping) 변경만으로는 다른 코어 간의 내부 카피를 처리하기 불가능하다. 또한, 멀티 코어 SSD는 하나의 내부 카피 커맨드(Internal Copy command)가 두 개의 FTL에 연관될 수 있기 때문에, 배치 내부 카피 커맨드(batch IC command)의 원자성을 보장하기 힘든 상태이다.

발명의 내용

해결하려는 과제

[0004] 본 발명은 전술한 문제 및 다른 문제를 해결하는 것을 목적으로 한다.

[0005] 본 발명은 다른 코어 간의 내부 카피 처리를 원활하게 할 수 있는 멀티 코어 SSD의 운용 방법을 제공하기 위한 것이다.

[0006] 또한, 본 발명은 배치 내부 카피 커맨드의 원자성을 보장할 수 있는 멀티 코어 SSD의 운용 방법을 제공하기 위한 것이다.

과제의 해결 수단

[0007] 상기 기술적 과제를 달성하기 위해, 본 발명에 의한 멀티 코어 솔리드 스테이트 드라이브의 운용 방법은 소스

LPN 및 목적지 LPN을 포함하는 내부 카피 커맨드를 호스트로부터 수신하는 단계, 서로 다른 FTL들 간의 내부 카피 커맨드를 물리적 카피로 정의하고, 물리적 카피를 지연시키는 단계, 및 물리적 카피의 소스 LPN을 소스 LPN이 포함된 PPN로 대체하여, 수정된 배치 내부 카피 커맨드를 생성하는 단계를 포함한다.

발명의 효과

- [0008] 본 발명에 의한 멀티 코어 솔리드 스테이트 드라이브의 운용 방법에 의하면, 서로 다른 코어 간의 내부 카피를 지연시킴으로써 오버헤드(overhead)를 최소화할 수 있다.
- [0009] 또한, 본 발명에 의한 멀티 코어 솔리드 스테이트 드라이브의 운용 방법에 의하면, 물리적 카피 기반의 블록 할당 기법을 통해 SSD의 FTL들이 증가하여도, 다수의 FTL 내의 로직 카피(logical copy)로 처리할 수 있다. 이를 통해서 NAND에 쓰기 동작 횟수를 줄여 NAND의 수명을 증가시킬 수 있다.
- [0010] 또한, 본 발명에 의한 멀티 코어 솔리드 스테이트 드라이브의 운용 방법에 의한 스케줄링은 내부 카피 커맨드가 오래 소요되는 FTL의 지연에 영향을 개선시킬 수 있다.
- [0011] 또한, 본 발명에 의한 멀티 코어 솔리드 스테이트 드라이브의 운용 방법에 의하면, 멀티 코어 SSD에서 배치 내부 카피 커맨드(batch IC command)의 원자성을 보장함으로써, 트랜잭션(transactional)한 방식으로 내부 카피 커맨드의 처리가 가능하며, 내부 카피 커맨드를 사용하는 어플리케이션(application)은 자체적인 로그(log)를 통해 트랜잭션(transactional)한 반응을 제공하기 위한 오버헤드를 피할 수 있다.
- [0012] 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과로 제한되지 않으며, 언급하지 않은 다른 효과들은 아래의 기재로부터 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

- [0013] 도 1은 본 발명에 의한 멀티 코어 솔리드 스테이트 드라이브를 나타내는 도면이다.
- 도 2는 본 발명에 의한 내부 카피 동작을 나타내는 순서도이다.
- 도 3은 내부 카피 커맨드를 처리하는 일례를 나타내는 도면이다.
- 도 4는 본 발명의 실시 예에 따른 가비지 컬렉션 방법을 나타내는 도면이다.
- 도 5는 본 발명에 의한 블록 할당 방법을 나타내는 도면이다.
- 도 6은 본 발명에 의한 동기화 된 복구 과정의 일례를 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0014] 이하 설명하는 기술은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는 바, 특정 실시 예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.
- [0015] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.
- [0016] 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 설시된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계 동작 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.
- [0017] 도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에도 다른 구성부가 담당하는 기능 중 일부 또는 전

부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.

- [0018] 또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 과정들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.
- [0019] 도 1은 본 발명에 의한 멀티 코어 솔리드 스테이트 드라이브를 나타내는 도면이다.
- [0020] 도 1을 참조하면, 본 발명에 의한 멀티 코어 솔리드 스테이트 드라이브는 호스트 어플리케이션(100) 및 메모리 디바이스(200)를 포함한다. 호스트 어플리케이션(100)은 호스트 인터페이스 컨트롤러(110) 및 제1 및 제2 플래시 변환 계층들(121,122)을 포함한다.
- [0021] 호스트 인터페이스 컨트롤러(Host interface controller; 이하, HIC)(110)는 호스트로부터 수신한 커맨드에 응답하여, 메모리 디바이스의 동작을 제어한다. 그리고, HIC(110)는 제1 및 제2 플래시 변환 계층(Flash Translation Layer; 이하 FTL)들(121,122)로 전달하는 입/출력 요청(I/O request)의 처리 순서를 결정한다.
- [0022] 본 발명은 다수의 FTL들(121,122)이 별도의 코어(core)에서 동작하는 멀티 코어 SSD를 기반으로 한다. 도 1 및 도 3은 제1 FTL(121) 및 제2 FTL(122)만이 도시되어 있지만, FTL의 개수는 이에 한정되지 않는다. FTL들(121,122) 각각은 입력받은 논리 주소를 물리 주소로 변환하는 작업을 수행한다. 또한, FTL들(121,122)은 SSD의 유효 공간을 확보하는 가비지 컬렉션(Garbage collection; GC) 동작을 수행한다.
- [0023] 멀티 코어 SSD를 기반에서 내부 카피 동작을 수행하기 위한 인터페이스(Interface)를 살펴보면 다음과 같다.
- [0024] 내부 카피 커맨드(Internal Copy Command)는 IC(src_LPN, dst_LPN)로 정의될 수 있다. 이 때, src_LPN은 소스 LPN을 의미하고, dst_LPN은 목적지 LPN을 의미한다. LPN은 로직 페이지 번호(LP: Logical Page Number)을 의미한다.
- [0025] 내부 카피 커맨드는 배치 내부 카피(batch IC) 방식으로 전달되어 오버헤드를 줄일 수 있다. 내부 카피 커맨드와 결부된 데이터는 IC(array of (src_LPN, dst_LPN))으로 전달된다.
- [0026] 배치 내부 카피 커맨드는 내부 카피 커맨드의 어레이(array)를 제외하고 별도의 메타데이터를 포함할 수 있다. 메타데이터는 그룹 아이디(group ID), 그룹 사이즈(group size), 내부 카피 엔트리 카운트(IC entry count), 해쉬값(hash value)을 포함할 수 있다. 배치 내부 카피 커맨드는 미리 설정된 디지털 데이터로 구성되고, 내부 카피 그룹(IC command group)은 다수의 배치 내부 카피 커맨드들로 이루어진다. 내부 카피 그룹은 원자적으로 처리되어야 하는 내부 카피 엔트리(IC entry)들의 집합이다. 그룹 아이디는 배치 내부 카피 커맨드가 어떤 내부 카피 그룹에 속하는지를 정의한다. 그룹 사이즈는 내부 카피 그룹을 구성하는 배치 내부 카피 커맨드의 수를 의미한다. 내부 카피 엔트리 카운트는 해당 배치 내부 카피 커맨드가 포함한 내부 카피 커맨드의 개수를 의미한다. 해쉬값은 일반 읽기/쓰기(read/write)가 배치 내부 카피 커맨드와 연관된 입/출력(I/O)인지 판단하는데 사용된다.
- [0027] 배치 내부 카피 커맨드는 일반 쓰기 명령과 같은 방식으로 SSD에게 전달된다. 이때, 배치 내부 카피 커맨드를 일반 쓰기 명령과 구분하기 위해서, 배치 내부 카피 커맨드의 논리적 어드레스 범위는 일반 쓰기와 구분된다.
- [0028] 내부 카피 커맨드와 연관된 읽기/쓰기 명령 간의 처리 순서를 살펴보면 다음과 같다.
- [0029] 내부 카피 커맨드와 연관된 쓰기 명령(write command)은 HIC(110)에 입력되는 순서와 동일하게 메모리 디바이스(200)에 쓰기 동작이 수행된다.
- [0030] 내부 카피 커맨드와 연관된 읽기 명령(read command)은 내부 카피 커맨드 처리 이후에 수행할 수 있도록 지연된다. HIC(110)의 입/출력 스케줄러(I/O scheduler)는 읽기 요청 및 쓰기 요청을 지연시키기 위해서, 선택적 분배 지연(selective-dispatch-delay; SDD) 방법 또는 전체 분배 지연(all-dispatch-delay; ADD) 방법을 이용할 수 있다. 선택적 분배 지연 방법은 내부 카피 커맨드와 연관된 읽기/쓰기 요청만을 지연시키는 방식이다. 전체 분배 지연 방법은 내부 카피 커맨드가 분배되는 중에는 모든 읽기 요청 및 쓰기 요청을 지연시키는 방식이다.
- [0031] HIC(110)는 내부 카피 커맨드의 데이터를 탐색하여, 내부 카피 커맨드와 일반 읽기/쓰기 요청이 연관되었는지 확인한다. HIC(110)는 내부 카피 커맨드의 메타데이터로 저장된 해쉬값을 이용하여, 내부 카피 커맨드와 일반 읽기/쓰기 요청이 연관되었는지 확인한다.

- [0032] 도 2는 본 발명에 의한 내부 카피 동작을 나타내는 순서도이고, 도 3은 내부 카피 커맨드를 처리하는 일례를 나타내는 도면이다.
- [0033] 도 2 및 도 3을 참조하여, 내부 카피 동작을 살펴보면 다음과 같다.
- [0034] 제1 단계(S210)에서, HIC(110)는 호스트로부터 내부 카피 커맨드를 수신한다.
- [0035] 제2 단계(S220)에서, HIC(110)는 물리적 카피를 판단하고, 물리적 카피를 지연시킨다.
- [0036] HIC(110)는 내부 카피 커맨드를 FTL들(121,122)에게 전달하기 이전에, 물리적 카피(physical copy; 이하, PC) 처리 과정을 수정한다. 이를 위해서, HIC(110)는 배치 내부 카피 요청(batch IC request)에서 물리적 카피 횟수(PC operation)를 찾는 동작을 수행한다. 만약, 내부 카피 커맨드에서 소스 LPN과 목적지 LPN이 동일한 FTL에 속할 경우, L2P 맵핑만을 수정하여 내부 카피 동작을 수행할 수 있다. 내부 카피 커맨드에서 각각의 소스 LPN과 목적지 LPN이 같은 FTL에 속하지 않은 경우에는 단순히 L2P 맵핑만으로 내부 카피 동작을 처리하기 곤란하기 때문에 별도의 과정이 필요로 하다.
- [0037] 이를 위해서, HIC(110)는 내부 카피 커맨드에서 각각의 소스 LPN과 목적지 LPN이 같은 FTL에 속하지 않은 경우를 물리적 카피로 판단한다. 물리적 카피는 FTL들(121,122)간의 데이터 카피(data copy) 동작이 수행되어야 하기 때문에, HIC(110)는 배치 내부 카피(batch IC)의 처리시간을 최소화하기 위해서 물리적 카피를 지연시킨다.
- [0038] 제3 단계(S230)에서, 물리적 카피를 바탕으로 수정된 배치 내부 카피 커맨드를 생성한다.
- [0039] 수정된 배치 내부 카피 커맨드를 생성하기 위해서, HIC(110)는 물리적 카피 엔트리(PC entry)의 소스 LPN의 논리영역을 FTL들(121,122)로부터 전달받은 물리적 페이지 번호(Physical Page Number; 이하, PPN)로 대체하고, 지연된 물리적 카피 리스트에 (dst_LPN, src_PPN) 정보를 추가한다. 구체적으로, HIC(110)는 모든 물리적 카피 엔트리(PC entry)에 대해 FTL들(121,122)에게 get_PPN 명령을 전달한다. get_PPN 명령은 소스 LPN이 저장된 물리 페이지(physical page) 번호를 요청하는 것이다. 예컨대, 도 1에서 보는 바와 같이 HIC(110)는 "(12,23)"의 내부 카피 엔트리(IC entry)를 바탕으로 "get_PPN(12)"를 제1 FTL(121)로 제공한다. 그리고 HIC(110)는 "PPN 100"을 제1 FTL(121)로부터 회신받는다.
- [0040] 지연된 물리적 카피 리스트는 물리적 카피가 처리되기 전에 전달된 읽기 요청을 소스 LPN이 포함된 FTL로 재정렬시키기 위해 유지된다.
- [0041] 그리고 HIC(110)는 수정된 내부 카피 커맨드에 대한 정보를 "지연된 물리적 카피 리스트(delayed PC list)"에 추가한다.
- [0042] 제4 단계(S240)에서, HIC(110)는 마지막으로 수정된 배치 내부 카피 커맨드(batch IC command)를 모든 FTL들(121,122)에게 내부 카피 커맨드로 전달한다.
- [0043] FTL들(121,122)은 수정된 내부 카피 커맨드를 전달받아 논리적 카피(Logical Copy; LC)와 물리적 카피 모두 L2P 맵핑 변경으로 카피 동작을 수행한다.
- [0044] FTL들(121,122)이 물리적 카피를 처리하기 위한 맵핑을 살펴보면 다음과 같다.
- [0045] 일반적인 SSD는 대역외(Out of Band; OOB) 영역에 P2L 맵핑(Physical to Logical mapping)을 기록하기 때문에, 쓰기 동작을 한 이후에 수정이 불가능하다.
- [0046] 멀티 코어 SSD는 내부 카피 동작을 수행하는 과정에서 하나의 물리 페이지 번호(Physical Page Number; 이하 PPN)에 2개 이상의 LPN이 매칭될 수 있고, 이에 따라 하나의 PPN은 여러번 카피 동작이 발생할 수 있다. 따라서, 멀티 코어 SSD에서 일반적인 대역외(Out of Band; OOB) 영역에 P2L 맵핑을 기록하는 방식을 적용하기에는 곤란하다.
- [0047] 본 발명은 하나의 PPN에 다수의 카피 동작이 발생하는 것을 방지하기 위해서 변경이 가능한 P2L 맵핑 테이블을 이용한다. 본 발명은 전체 물리적 페이지, 즉 모든 NAND flash page들에 대해 1 별의 P2L 맵핑 테이블을 유지하고, DRAM의 일부 공간을 P2L 맵핑의 CMT(cached mapping table)로 활용한다.
- [0048] 일반적인 SSD의 가비지 컬렉션(Garbage Collecton) 동작은 물리적 페이지(physical page)의 OOB 영역을 읽어 해당 물리적 페이지가 어떤 LPN과 맵핑되는지를 판단한다. 이때, L2P 맵핑은 일대일 방식이기 때문에, 물리적 페이지는 1개의 LPN과 맵핑이 된다.

- [0049] 앞서 살펴본 바와 같이, 내부 카피 커맨드는 소스 LPN과 목적지 LPN으로 정의될 수 있다. 그리고 본 발명은 적어도 1 차례 이상의 내부 카피 커맨드에 의해서 소스 LPN의 논리적 페이지와 목적지 LPN의 논리적 페이지가 동일한 물리적 페이지를 공유할 수 있다. 본 발명에 따른 가비지 컬렉션은 유효한 물리 페이지와 연관된 여러 LPN 정보를 검색한다. 이를 위한 본 발명에 의한 가비지 컬렉션 동작의 구체적인 실시 예들을 살펴보면 다음과 같다.
- [0050] 도 4는 가비지 컬렉션 동작의 실시 예들을 나타내는 도면이다.
- [0051] 도 4의 (a)는 SSD에서 내부 카피 커맨드의 L2P 및 P2L이 유지되는 모습을 나타내는 도면이다. 즉, 도 4의 (a)는 내부 카피 동작을 수행하지 않은 상태에서, 하나의 PPN과 하나의 LPN이 맵핑된 상태를 나타내고 있다.
- [0052] 도 4의 (b)는 도 4의 (a)에서 IC(L0,L1)이라는 내부 카피 커맨드의 요청에 따라 내부 카피 동작을 수행한 것을 나타내는 도면이다.
- [0053] 도 4의 (b)를 참조하면, 내부 카피를 수행하기 위해서, 제1 논리적 페이지(L0)와 물리적 페이지가 맵핑된 L2P 맵핑, 및 물리적 페이지와 제2 논리적 페이지(L1)이 맵핑된 P2L 맵핑을 추가한다. L2P 맵핑은 "(L1, P0)"로 정의되고, P2L 맵핑은 "(P0, L1)"으로 정의된다. 이하, 제1 논리적 페이지(L0)를 "L0"라고 지칭하고, 제2 논리적 페이지(L1)를 "L1"이라고 지칭하기로 하며, 물리적 페이지를 "P0"라고 지칭하기로 한다.
- [0054] 내부 카피가 수행된 페이지를 가비지 컬렉션하는 방법은 다음과 같다.
- [0055] "L0"의 L2P 맵핑은 카피된 비트(copied bit)라고 지칭되는 플래그 비트(flag bit)를 세팅하고, 플래그 비트는 OOB에 저장된 P2L 맵핑을 통해 "L0"를 지시하는 PPN이 다른 LPN과 맵핑되어 있는 것을 나타낸다. 따라서, LPN이 카피된 경우, 가비지 컬렉션은 "P0"를 카피하기 위해 "P0"의 OOB를 읽어 "L0"를 찾을 수 있다. 그리고 "L0"에 카피된 비트가 세팅되어 있기 때문에, DRAM에 저장된 P2L 맵핑을 통해 L1을 찾을 수 있다. 이를 통해, 가비지 컬렉션 동작이 "P0"를 다른 위치로 이동시킬 때, 모든 LPN의 L2P 맵핑을 업데이트하는 것이 가능하다.
- [0056] LPN이 복사되지 않은 경우, 즉 물리적 페이지의 OOB 영역에 있는 LPN이 해당 페이지에 맵핑되어 있으면 P2L 맵핑 테이블에 액세스 할 필요가 없다.
- [0057] 도 4의 (c)는 도 4의 (b)상태에서 "L0"에 대한 쓰기 동작을 수행한 결과를 도시하고 있다. 이 경우, "L0"는 수정된 L2P 맵핑을 통해서 새로운 PPN을 지시한다. "L1"은 "P0"의 OOB 영역에 남아있는 (P0,L0)의 P2L 맵핑되기 때문에, 가비지 컬렉션은 유효 비트맵(valid bitmap)을 통해 "P0"가 유효한 페이지라는 것을 인지한다.
- [0058] 그리고 가비지 컬렉션은 OOB에서 "L0"를 검색하고, "L0"의 L2P를 확인한다. 이때, 유효 비트맵은 세팅된 상태이지만, "L0"의 L2P는 P0를 지시하지 않기 때문에, FTL들(121,122)은 "L0"가 LPN에 맵핑되지 않는 것으로 인식하고 "P0"의 DRAM에 저장된 P2L 맵핑을 통해 L1을 얻을 수 있다. 마찬가지로, GC가 P0를 옮기는 작업을 수행시, L1의 L2P 맵핑을 업데이트하여 "P0"를 옮길 수 있다.
- [0059] 도 4 (d)는 도 4의 (c)상태에서 IC(L1,L2)를 수행한 결과를 도시하고 있다. 도 4의 (d)를 참조하면, IC(L1,L2)는 일대다(one-to-many) P2L 맵핑 상황을 야기한다.
- [0060] 본 발명의 FTL들(121,122)에서 P2L 맵핑은 일대일(one-to-one) 방식만을 허용하고, 일대다(one-to-many) 맵핑 방식은 L2P 맵핑에서만 허용된다. 따라서, P2L 맵핑 방식은 "L1"과 제3 논리적 페이지에 해당하는 "L2" 중 하나의 페이지와 매칭될 수 있다. 이 경우, "L1"은 "P0"를 가리키지만, "P0"를 기반으로 "L1"을 찾을 방법이 존재하지 않는다. 따라서, L1에는 단방향 비트(unilateral bit)를 세팅하고, "L1"이 세팅된 블록은 기준 카운트(ref count)를 증가시키므로써, 해당 상태를 가진 PPN이 포함된 NAND 플래시 블록은 가비지 컬렉션의 대상에서 제외된다.
- [0061] 즉, 단방향 비트가 세팅된 블록 "L1"에 대한 덮어쓰기 동작이 수행될 때까지 가비지 컬렉션의 희생(victim)에서 제외된다.
- [0062] 도 4의 (e)는 도 4의 (d) 상태에서 지연된 물리적 카피를 수행한 이후의 L2P 맵핑과 P2L 맵핑 상태를 나타내는 도면이다.
- [0063] 도 4의 (e)를 참조하면, L2는 단방향 비트를 세팅하며, P2L은 L3를 지시한다. 이때, L2는 P0를 기반으로 찾을 수 없는 페이지이기 때문에 기준 카운트(RC)를 1 증가시키며, L3는 다른 FTL이 참조하는 페이지이기 때문에 기준 카운트(RC)를 1 증가시킨다. 해당 블록은 L1과 L2에 대한 덮어쓰기(over-write) 동작이 수행되고 L3에 대한

지연된 물리적 카피 동작이 처리될 때까지 가비지 컬렉션의 희생에서 제외된다.

- [0064] 앞서 살펴본 바와 같이, 본 발명은 내부 카피 동작을 수행하는 과정에서, 물리적 카피 동작을 지연시킴으로써 오버헤드를 줄일 수 있다. 하지만, 물리적 카피 동작을 지연시키는 것만으로는 읽기/쓰기 동작을 포함하는 NAND flash operation을 줄이지는 못한다.
- [0065] 본 발명은 NAND flash operation을 줄이기 위해서, 소스 페이지와 목적지 페이지를 동일한 FTL에 할당(allocation)한다. 이처럼, 소스 페이지와 목적지 페이지를 동일한 FTL에 할당하기 위해서, 본 발명은 "물리적 카피 적응형 파일 시스템 레벨 블록 할당(PC-aware file system-level block allocation)" 기법을 제안한다. 이를 위해서는 multi-core SSD가 내부에서 논리 주소 공간을 FTL들(121,122)에게 어떻게 분배하는지에 대한 정보를 호스트에게 공유해 주는 것이 필요하다.
- [0066] 도 5는 본 발명에 의한 블록 할당 방법을 나타내는 도면이다. 도 5는 EXT4 파일 시스템에서 본 발명에 따른 내부 카피 방법을 적용하는 과정에서 블록 할당 방법의 실시 예를 도시하고 있다.
- [0067] 도 5의 (a)는 일반적인 블록 할당 기법을 나타내는 도면이다.
- [0068] 도 5의 (b)는 물리적 카피로 처리될 페이지들이 패딩(padding)을 통해 모두 논리적 카피(Logical Copy; LC)로 전환되는 방법을 나타내는 도면이다. 논리적 카피는 소스 LPN과 목적지 LPN이 동일한 FTL 코어에 속한 것을 지칭한다. 본 명세서는 논리적 페이지를 각 FTL 코어에 "풀리 스트라이핑(fully striping)"하는 방식을 도시하고 있다. 도 5에서 "home location"은 추후에 내부 카피를 통해서 복사되는 목적지 LPN을 지칭한다. 예컨대, 도 5의 (b)는 "home location"이 15인 페이지를 "LPN 107"번에 할당함으로써 같은 FTL 코어에서 처리가 가능한 것을 도시하고 있다.
- [0069] 패딩 방법을 이용하는 과정에서, FTL들(121,122)이 많은 환경에서는 PC를 제거하기 위해 필요한 패딩의 개수가 많아지는 문제가 있다. 또한 패딩으로 공간을 소모하면, 공간을 재요청(reclaim)하기 위한 체크 포인트(checkpoint) 동작이 더 자주 발생할 수 있다.
- [0070] 도 5의 (c)는 본 발명에 의한 블록 할당 방법을 나타내는 도면이다. 본 발명에 의한 블록 할당 방법은 물리적 카피 인지 블록 할당(PC-aware block allocation) 기법을 이용하여, 패딩을 제한적으로 사용하는 블록 할당 방법을 나타내고 있다.
- [0071] 도 5의 (c)를 참조하면, 본 발명에 의한 블록 할당 방법은 먼저, 호스트는 공간을 최소화하면서, 패딩을 사용하여 모든 물리적 카피를 LC로 변환한다. 이후, 패딩을 임계치(threshold) 보다 많이 사용한 경우, 가장 마지막에 위치한 LC를 원래 자리로 돌리는 작업을 수행한다. 가장 마지막에 위치한 LC는 패딩에 의해 이동된 물리적 카피를 지칭한다.
- [0072] 이처럼 본 발명은 모든 물리적 카피에 대해서 패딩을 사용하지 않고, 제한적으로 패딩을 사용함으로써 과부하를 줄일 수 있다.
- [0073] SSD는 호스트가 전달한 I/O의 transfer 순서와 NAND에 적히는 순서를 같도록 보장하는 것이 필요하다. 이를 위한 HIC(110)의 스케줄링 정책은 다음과 같다.
- [0074] 내부 카피 요청(IC request)은 쓰기 요청(write request) 보다 높은 우선 순위를 갖는다. HIC(110)에서 쓰기 요청의 처리 순서가 섞일 수 있다. 스케줄이 결정된 요청은 FTL들(121,122)에 전달된다. FTL들(121,122)는 읽기 큐(read queue)와 쓰기 큐(write queue)를 개별적으로 보유하고 있고, 읽기 순서를 우선적으로 처리한다. 따라서, FTL들(121,122)은 늦게 전달받은 읽기 동작이 쓰기 동작 보다 먼저 처리될 수 있다. 내부 카피 요청(IC request)은 FTL들(121,122)에게 쓰기 요청(write request)으로 전달된다. FTL들(121,122)에서 큐(queue)의 엔트리(entry)들은 선입선출(first-in-first-out; FIFO)방식으로 처리된다.
- [0075] 이를 전제로, 본 발명은 다음과 같은 규칙으로 내부 카피 커맨드와 읽기 요청 간의 처리 순서를 결정한다.
- [0076] 첫째로, 카피-후-읽기(Read-After-Copy; RAC)는 내부 카피 커맨드와 해당 내부 카피 커맨드 보다 늦게 전달된 연관된 읽기 동작 사이의 순서를 규정한다. 즉, 읽기 요청(read request)은 내부 카피 동작이 완료된 이후에 수행된다. 이를 위해서, HIC(110)는 FTL들(121,122)에게 전달했지만 내부 카피 동작이 끝나지 않은 내부 카피 커맨드와 새롭게 스케줄링할 읽기 요청(read request)이 연관된 경우, 해당 읽기 요청(read request)을 내부 카피 커맨드 처리 이후에 FTL들(121,122)에게 전달한다.
- [0077] 두 번째로, 쓰기-후-카피(Copy-After-Write; CAW)는 내부 카피 커맨드 보다 먼저 전달된 연관된 쓰기 요청과 내

부 카피 커맨드 간의 처리 순서 제약사항이 있다. HIC(110)에서는 내부 카피 커맨드 보다 먼저 전달된 연관된 쓰기 요청이 있는지 검색한 후, 연관된 쓰기 요청을 먼저 FTL들(121,122)에게 전달하고, 내부 카피 커맨드를 FTL들(121,122)로 전달한다. 이 과정에서 FTL들(121,122)은 쓰기 큐(write queue)에 삽입(insert)된 순서를 지키기 때문에, 내부 카피 커맨드는 쓰기 요청(write request)의 처리를 기다릴 필요 없이 FTL들(121,122)에게 전달하는 것이 가능하다.

[0078] 비정상적 전원 불량 상황에서 배치 내부 카피 커맨드의 원자성을 보장할 수 있는 방법을 살펴보면 다음과 같다.

[0079] 본 발명은 내부 카피 커맨드를 웨이트-온-트랜스퍼(Wait on Transfer; WoT) 방식으로 처리한다. 내부 카피 커맨드를 전달받은 FTL들(121,122)은 각자 독립적으로 자신에게 전달된 내부 카피 커맨드를 수행한다. 하나의 그룹으로 묶인 내부 카피 커맨드들 모두 처리한 경우, FTL들(121,122)은 내부 카피 그룹의 체크 포인트(check Point; 이하, CP)를 NAND에 기입함으로써 해당 그룹의 내부 카피가 종료되었음을 마킹한다. 서든 파워 오프(sudden-power-off; 이하, SPO) 같은 크래쉬(crash) 현상이 발생하면, NAND에 기록한 내부 카피 그룹의 체크 포인트(IC group CP)를 통해 복구 가능한 내부 카피 그룹을 판단할 수 있다.

[0080] SPO 현상을 감지하였을 때, 이를 회복하는 방법은 다음과 같다.

[0081] 본 발명은 HIC(110)가 FTL들(121,122)의 복구를 중재한다.

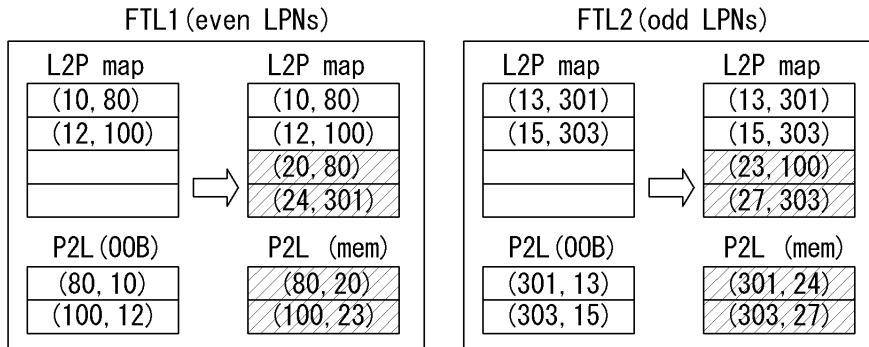
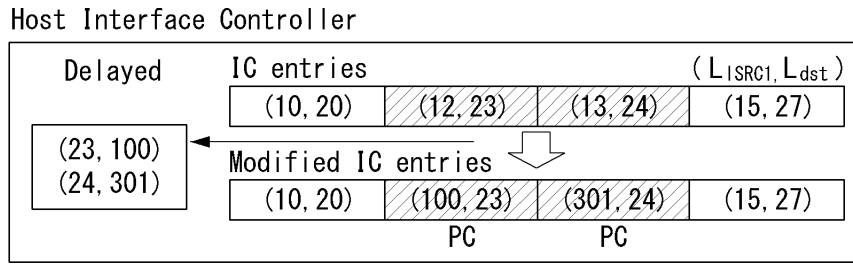
[0082] FTL들(121,122) 각각은 복구 가능한 내부 카피 그룹이 다를 수 있다. 내부 카피 커맨드의 원자성을 보장하기 위해서는 모든 FTL들(121,122)이 어떤 내부 카피 그룹에 대해 처리가 완료되어야 한다. SPO 상황이 발생한 경우, HIC(110)는 모든 FTL들(121,122)에게 복구 가능한 마지막 내부 카피 그룹이 몇 번인지에 대한 정보를 수집한다. 그리고 HIC(110)는 FTL들(121,122)이 복구 가능한 내부 카피 그룹을 선정하여 FTL들(121,122)에게 전달하고, 각 FTL들(121,122)은 독립적으로 복구 동작을 수행한다. 예컨대, 제2 FTL(122)가 내부 카피 5번까지 복구가 가능하다면, 제1 FTL(121)이 내부 카피 5번까지 복구를 수행할 수 없기 때문에 내부 카피 5번에 대한 원자성은 보장할 수 없다. 따라서, HIC(110)는 제2 FTL(122)가 내부 카피 5번까지 복구 가능할지라도, 내부 카피 3번까지만 복구를 수행한다. 그리고 본 발명의 FTL들(121,122)은 old-PPN 정보와 목적지 LPN 정보를 바탕으로 내부 카피 커맨드와 쓰기 명령의 redo 및 undo를 수행한다.

[0083] 두 번째로, 내부 카피 그룹의 원자성을 보장하기 위해서는 내부 카피 커맨드뿐만 아니라 쓰기 요청에 대해서도 undo와 redo 동작이 가능해야 한다. "undo"는 해당 쓰기 동작을 다시 수행하는 것을 지칭하고, "redo"는 해당 쓰기 동작을 취소하는 것을 지칭한다. 일반적인 SSD에서는 SPO 상황에 redo하기 위해서 스페어(spare) 영역에 P2L 맵핑 엔트리를 기록한다. 이에 반해서, 본 발명은 쓰기 동작의 undo를 수행할 수 있도록 OOB 영역에 해당하는 스페어 영역에 old-ppn 정보도 함께 기록한다. "old PPN 정보"는 undo 동작을 수행하기 위해 필요한 정보이다. FTL들(121,122)은 내부 카피 커맨드의 undo와 redo를 가능하게 하기 위해서, 각 내부 카피 커맨드의 처리가 끝나면 NAND 플래시 메모리에 old_ppn과 목적지 LPN(dst_lpn)에 대한 정보를 기록한다. FTL들(121,122)은 해당 정보들을 사용하여, 내부 카피 커맨드와 쓰기 명령의 redo 및 undo가 가능하다. 내부 카피의 redo 동작은 SPO 상황 이후에 내부 카피 커맨드에서 변경한 L2P 맵핑 테이블 정보를 반영할 경우에 수행된다. 내부 카피의 undo 동작은 HIC(110)가 최종 복구하여야 할 내부 카피 보다 많이 수행된 L2P 맵핑 테이블 정보를 갖는 경우에 수행된다.

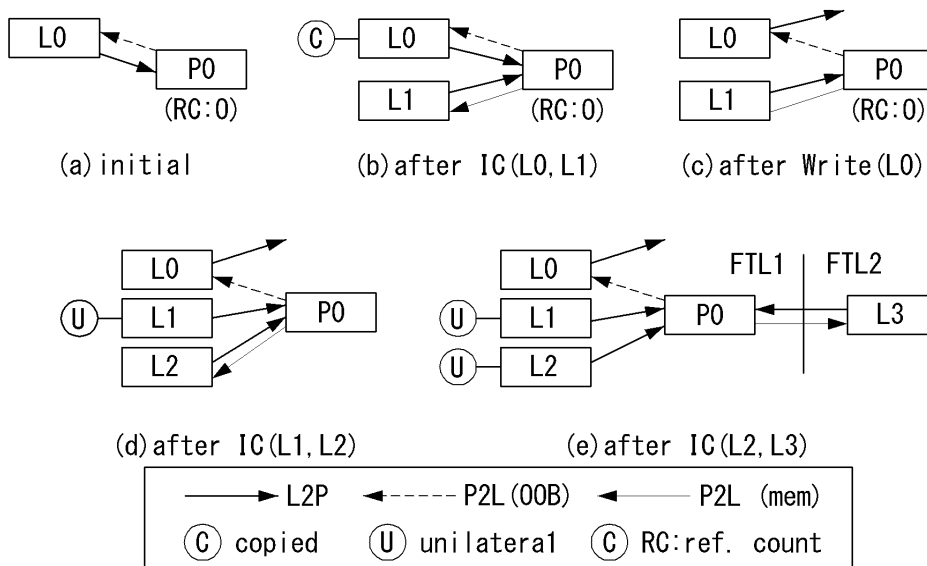
[0084] 도 6은 앞서 설명한 본 발명에 의한 동기화 된 복구 과정(synchronized recovery)의 일례를 나타내는 도면이다.

[0085] 도 6을 참조하면, 제1 FTL(FTL1)과 제2 FTL(FTL2)은 각각 내부 카피 그룹 2번(ICG2)까지 복구가 가능하며, 제3 FTL(FTL3)과 제4 FTL(FTL4)은 내부 카피 그룹 1번(ICG1)까지 복구가 가능하다. 따라서 HIC(110)는 동기화된 그룹 아이디(synchronized group ID)를 내부 카피 그룹 1번(ICG1)으로 선정한다. 제1 FTL(FTL1)의 경우, 내부 카피 그룹 2번(ICG2)을 처리한 이후에 L2P 맵핑을 저장한 예를 나타내고 있다. 따라서, 내부 카피 그룹 2번(ICG2)과 그 이후에 처리한 쓰기 동작에 대한 undo를 수행한다. 제2 FTL(FTL2)은 내부 카피 그룹 2번(ICG2)까지 처리한 후 크래쉬가 발생하였지만, 내부 카피 그룹 2번(ICG2)을 수행하기 전에 L2P 맵핑을 저장한 예를 나타내고 있다. 따라서, 마지막 저장한 L2P 맵핑에서부터 내부 카피 그룹 2번(ICG2)을 처리하기 직전까지 redo 작업을 수행한다. 제3 FTL(FTL3)은 내부 카피 그룹 1번(ICG1)을 수행한 후, crash가 발생하였으며, 내부 카피 그룹 2번(ICG2)에 대한 수행은 하지 않은 상태를 나타내고 있다. 따라서 제3 FTL(FTL3)은 SPO 복구 과정에서, 내부 카피 그룹 1번(ICG1)과 크래쉬 이전까지 수행했던 쓰기 명령에 대한 복구를 수행한다. 제4 FTL(FTL4)은 크래쉬가 발생하기 전에 내부 카피 그룹 1번(ICG1)까지 처리하였으며, 내부 카피 그룹 1번(ICG1)을 수행한 후, L2P 맵핑을 플러쉬(flush) 한 상태를 나타내고 있다. 따라서 제4 FTL(FTL4)은 크래쉬 이전까지 수행한 쓰기 명령들

도면3

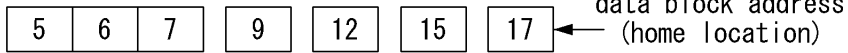


도면4



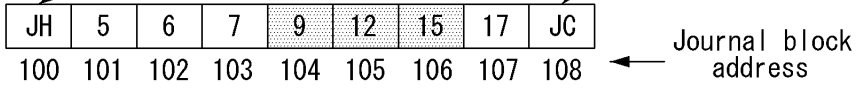
도면5

The page list of a running transaction

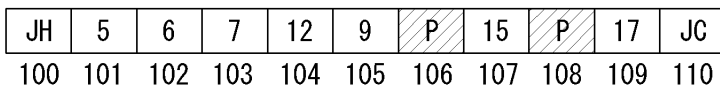


Jrnl header block

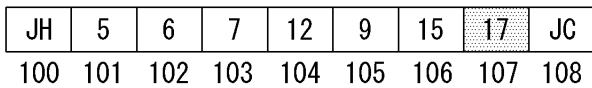
Jrnl commit block



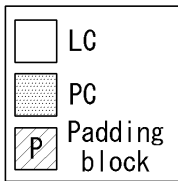
(a) normal block allocation



(b) PC-aware block allocation w/Padding



(c) PC-aware block allocation w/Padding reduction



도면6

