(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0145151
(43) 공개일자 2020년12월30일

(51) 국제특허분류(Int. Cl.)
G06F 12/02 (2018.01) *G06F 12/0802* (2016.01)
G06F 3/06 (2006.01)
(52) CPC특허분류
G06F 12/0246 (2013.01)
G06F 12/0292 (2013.01)
(21) 출원번호 10-2019-0073753
(22) 출원일자 2019년06월20일
심사청구일자 없음

(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
성균관대학교산학협력단
경기도 수원시 장안구 서부로 2066 (천천동, 성균관대학교내)
(72) 발명자
임상우
경기도 용인시 기흥구 금화로11번길 10(상갈동, 금화마을주공3단지아파트)
신동군
서울특별시 강남구 역삼로 314, 305동 1004호 (역삼동, 개나리 푸르지오)
(74) 대리인
리앤목특허법인

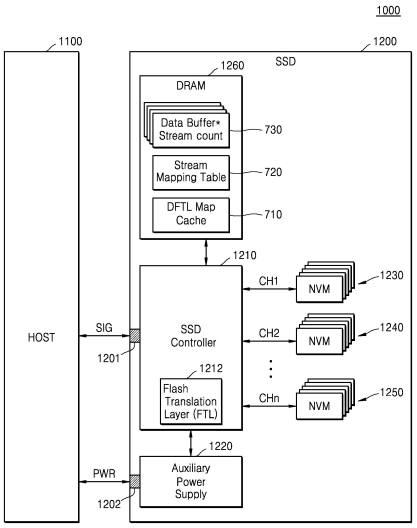
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **맵핑 정보를 축약하는 플래시 변환 계층 구조를 이용하여 메모리 리소스를 관리하는 데이터 저장 장치**

(57) 요 약

맵핑 정보를 축약하는 플래시 변환 계층(FTL) 구조를 이용하여 메모리 리소스를 관리하는 데이터 저장 장치가 제공된다. FTL 구조는, 호스트의 입출력 요청들에 대한 전체 논리 주소 공간을 n개의 가상 논리 주소 스트림들로 분할하고, 가상 논리 주소 스트림들의 스트림 속성에 따라 예비 클러스터 맵핑 테이블을 생성하고, 예비 클러스터 맵핑 테이블에 대하여 k-평균 클러스터링 알고리즘을 수행하여 축약된 클러스터 맵핑 테이블을 생성하고, DFTL 방식을 이용하여 상대적으로 자주 참조되는 축약된 클러스터 맵핑 테이블의 일부로 구성되는 캐시드 클러스터 맵핑 테이블을 생성한다. FTL 구조는, 축약된 클러스터 맵핑 테이블에 의해 휘발성 메모리 장치의 데이터 버퍼 공간에서 맵핑되지 않은 물리 주소 스트림들에 할당되었던 데이터 버퍼들의 공간을 DFTL 캐시 맵 공간으로 확장한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 12/0802 (2013.01)
G06F 3/064 (2013.01)
G06F 3/0679 (2013.01)

명 세 서

청구범위

청구항 1

호스트의 논리 주소 스트림들과 불휘발성 메모리 장치들의 물리 주소 스트림들 사이의 맵핑 정보를 관리하는 플래시 변환 계층(FTL) 구조에 있어서,

상기 호스트로부터 수신되는 입출력 요청들에 대한 전체 논리 주소 공간을 n(n은 정수)등분하여 n개의 가상 논리 주소 스트림들로 설정하고, 상기 n개의 가상 논리 주소 스트림들의 선택된 스트림 속성에 따라 상기 n개의 가상 논리 주소 스트림들을 k (k는 n보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 예비 클러스터 맵핑 테이블을 생성하도록 구성되는 스트림 클러스터링 매니저 모듈; 및

상기 예비 클러스터 맵핑 테이블에 대하여 적응적 k-평균 클러스터링 알고리즘을 수행하여, 상기 n개의 가상 논리 주소 스트림들을 c(c는 k 보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 축약된 클러스터 맵핑 테이블을 생성하도록 구성되는 적응적 k-평균 클러스터링 알고리즘 모듈을 포함하는 플래시 변환 계층 구조.

청구항 2

제1항에 있어서, 상기 플래시 변환 계층 구조는,

상기 축약된 클러스터 맵핑 테이블에 대하여 DFTL(Demand-based FTL) 방식을 이용하여 상기 축약된 클러스터 맵핑 테이블에서 상대적으로 자주 참조되는 맵핑 테이블의 일부로 구성되는 캐시드 클러스터 맵핑 테이블을 생성하도록 구성되는 DFTL 캐시 맵 매니저 모듈을 더 포함하는 플래시 변환 계층 구조.

청구항 3

제2항에 있어서, 상기 DFTL 캐시 맵 매니저 모듈은,

상기 축약된 클러스터 맵핑 테이블을 이용하여 (k-c)개의 맵핑되지 않은 물리 주소 스트림들에 할당되어 있던 휘발성 메모리의 데이터 버퍼 공간을 상기 캐시드 클러스터 맵핑 테이블을 저장하도록 확장하는 것을 특징으로 하는 플래시 변환 계층 구조.

청구항 4

제1항에 있어서, 상기 스트림 클러스터링 매니저 모듈은,

상기 n개의 가상 논리 주소 스트림들에 대응하는 하나 이상의 스트림 속성들을 추적하여 획득하고, 획득된 스트림 속성들 중 하나를 선택하여 상기 선택된 스트림 속성으로 제공하도록 구성되고,

상기 획득된 스트림 속성들은 상기 n개의 가상 논리 주소 스트림들의 예상 수명, 빈도, 시간적 지역성, 순차성, 입출력 크기, 입출력 격리 레벨 및 QoS(Quality of Service) 요구 사항과 관련되는 속성들인 것을 특징으로 하는 플래시 변환 계층 구조.

청구항 5

제1항에 있어서, 상기 적응적 k-평균 클러스터링 알고리즘 모듈은,

상기 n개의 가상 논리 주소 스트림들의 상기 선택된 속성을 기반으로, 상기 k개의 물리 주소 스트림들 사이의 중심을 계산하고, 상기 k개의 물리 주소 스트림들끼리의 거리를 계산하고, 가까운 물리 주소 스트림들을 합병하는 동작을 반복 수행하여 상기 축약된 클러스터 맵핑 테이블을 생성하도록 구성되는 것을 특징으로 하는 플래시 변환 계층 구조.

청구항 6

불휘발성 메모리 장치들;

휘발성 메모리 장치; 및

호스트의 논리 주소 스트림들과 상기 불휘발성 메모리 장치들의 물리 주소 스트림들 사이의 맵핑 정보를 관리하는 플래시 변환 계층(FTL) 구조를 이용하여, 상기 호스트로부터 수신되는 입출력 요청들에 응답하여 상기 불휘발성 메모리 장치들 및 상기 휘발성 메모리 장치를 제어하는 콘트롤러를 포함하고,

상기 플래시 변환 계층 구조는,

상기 호스트의 입출력 요청들에 대한 전체 논리 주소 공간을 n(n은 정수)등분하여 n개의 가상 논리 주소 스트림들로 설정하고, 상기 n개의 가상 논리 주소 스트림들을 k (k는 n보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 예비 클러스터 맵핑 테이블을 생성하고, 상기 예비 클러스터 맵핑 테이블에 대하여 적응적 k-평균 클러스터링 알고리즘을 수행하여 상기 n개의 가상 논리 주소 스트림들을 c(c는 k 보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 축약된 클러스터 맵핑 테이블을 생성하도록 구성되는 데이터 저장 장치.

청구항 7

제6항에 있어서, 상기 플래시 변환 계층 구조는,

상기 축약된 클러스터 맵핑 테이블에 대하여 DFTL(Demand-based FTL) 방식을 이용하여 상기 축약된 클러스터 맵핑 테이블에서 상대적으로 자주 참조되는 맵핑 테이블의 일부로 구성되는 캐시드 클러스터 맵핑 테이블을 생성하도록 구성되는 것을 특징으로 하는 데이터 저장 장치.

청구항 8

불휘발성 메모리 장치들, 휘발성 메모리 장치 및 콘트롤러를 포함하고, 상기 콘트롤러는 호스트로부터 수신되는 입출력 요청들에 응답하여 상기 불휘발성 메모리 장치들 및 상기 휘발성 메모리 장치를 제어하도록 구성되는 데이터 저장 장치의 동작 방법에 있어서,

상기 콘트롤러에 의해, 상기 호스트의 입출력 요청들에 대한 전체 논리 주소 공간을 n(n은 정수)등분하여 n개의 가상 논리 주소 스트림들로 설정하는 단계;

상기 콘트롤러에 의해, 상기 n개의 가상 논리 주소 스트림들을 k (k는 n보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 예비 클러스터 맵핑 테이블을 생성하는 단계;

상기 콘트롤러에 의해, 상기 예비 클러스터 맵핑 테이블에 대하여 적응적 k-평균 클러스터링 알고리즘을 수행하여 상기 n개의 가상 논리 주소 스트림들을 c(c는 k 보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 축약된 클러스터 맵핑 테이블을 생성하는 단계; 및

상기 콘트롤러에 의해, 상기 축약된 클러스터 맵핑 테이블에 대하여 DFTL(Demand-based FTL) 방식을 이용하여 상기 축약된 클러스터 맵핑 테이블에서 상대적으로 자주 참조되는 맵핑 테이블의 일부로 구성되는 캐시드 클러스터 맵핑 테이블을 생성하는 단계를 포함하는 데이터 저장 장치의 동작 방법.

청구항 9

제8항에 있어서, 상기 데이터 저장 장치의 동작 방법은,

상기 콘트롤러에 의해, 상기 캐시드 클러스터 맵핑 테이블을 상기 휘발성 메모리 장치의 DFTL 맵 캐시 공간에 저장하는 단계;

상기 콘트롤러에 의해, 상기 축약된 클러스터 맵핑 테이블을 상기 휘발성 메모리 장치의 스트림 맵 테이블 공간; 및

상기 콘트롤러에 의해, 상기 호스트의 상기 입출력 요청들의 데이터를 데이터 버퍼 공간에 저장하는 단계를 더 포함하는 것을 특징으로 하는 데이터 저장 장치의 동작 방법.

청구항 10

제9항에 있어서, 상기 데이터 저장 장치의 동작 방법은,

상기 콘트롤러에 의해, 상기 축약된 클러스터 맵핑 테이블을 이용하여 (k-c)개의 맵핑되지 않은 물리 주소 스트림들에 할당되어 있던 상기 데이터 버퍼 공간을 상기 DFTL 맵 캐시 공간으로 확장하는 단계를 더 포함하는 것을 특징으로 하는 데이터 저장 장치의 동작 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 플래시 메모리 기반의 데이터 저장 장치에 관한 것으로서, 더욱 상세하게는 맵핑 정보를 축약하는 플래시 변환 계층 구조를 이용하여 메모리 리소스를 관리하는 데이터 저장 장치 및 그것의 동작 방법에 관한 것이다.

배 경 기 술

[0002] 운영 체제(OS, Operating System) 나 파일 시스템(file system)으로부터 플래시 메모리 기반의 데이터 저장 장치로 접근하고자 하는 요청을 수행하기 위해서, OS 나 파일 시스템과 플래시 메모리 사이에는 플래시 변환 계층(FTL, Flash Translation Layer)이 존재한다. 플래시 메모리는 페이지 단위의 읽기 및 쓰기, 블록 단위의 소거(erase)를 기본 동작으로 하고, 배드 블록을 허용하고, 수명이 제한되는 등의 제약들이 있다. 제자리에 데이터를 덮어쓸 수 없는, 즉 제자리 갱신이 불가능한 플래시 메모리의 제약 사항을 극복하기 위하여, FTL은 호스트에서 제공되는 논리 주소(logical address)와 플래시 메모리의 물리 주소(physical address) 사이의 관계를 정의하는 맵핑 정보를 포함한다. FTL은 맵핑 정보를 이용하여 호스트의 논리 주소를 플래시 메모리의 물리 주소로 변환하는 주소 변환 동작을 수행한다.

[0003] 데이터 저장 장치는 그 내부에 휘발성 메모리, 예컨대 DRAM(Dynamic Random Access Memory) 또는 SRAM(Static Random Access Memory)를 포함한다. 휘발성 메모리는 호스트가 쓰기 요청한 데이터를 저장한 후 플래시 메모리에 쓰도록 구성되는 데이터 버퍼 기능을 수행하고, FTL의 맵핑 정보를 저장하도록 구성되는 FTL 맵핑 기능을 수행할 수 있다.

[0004] FTL은 호스트가 쓰기 요청한 데이터를 휘발성 메모리에 저장한 후, 호스트에게 해당 호스트 쓰기 요청에 대하여 쓰기 완료 응답을 할 수 있다. FTL의 쓰기 완료 응답 이 후 휘발성 메모리에 있는 데이터를 플래시 메모리에 쓰지 않은 상태에서 쓰기 요청된 데이터와 동일한 논리 주소 공간에 대해 호스트 읽기 요청이 도착하면, FTL은 휘발성 메모리의 캐시에 보관되어 있는 최신 데이터로 읽기 요청을 처리할 수 있다.

[0005] 한정된 리소스인 휘발성 메모리를 효율적으로 관리하여 FTL 캐시 기능이 확장(scalability)될 수 있다면, 응답 속도 개선을 통해 데이터 저장 장치의 성능을 향상시킬 수 있을 것이다.

발명의 내용

해결하려는 과제

[0006] 본 발명의 목적은 맵핑 정보를 축약하는 플래시 변환 계층(FTL) 구조를 이용하여 메모리 리소스를 관리하는 데이터 저장 장치 및 그것의 동작 방법을 제공하는 데 있다.

과제의 해결 수단

[0007] 본 발명의 실시예들에 따른 플래시 변환 계층(FTL) 구조는, 호스트의 논리 주소 스트림들과 불휘발성 메모리 장치들의 물리 주소 스트림들 사이의 맵핑 정보를 관리한다. 플래시 변환 계층 구조는, 호스트로부터 수신되는 입출력 요청들에 대한 전체 논리 주소 공간을 n(n은 정수)등분하여 n개의 가상 논리 주소 스트림들로 설정하고, n개의 가상 논리 주소 스트림들의 선택된 스트림 속성에 따라 n개의 가상 논리 주소 스트림들을 k (k는 n보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 예비 클러스터 맵핑 테이블을 생성하도록 구성되는 스트림 클러스터링 매니저 모듈과, 예비 클러스터 맵핑 테이블에 대하여 적응적 k-평균 클러스터링 알고리즘을 수행하여 n개의 가상 논리 주소 스트림들을 c(c는 k 보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 축약된 클러스터 맵핑 테이블을 생성하도록 구성되는 적응적 k-평균 클러스터링 알고리즘 모듈을 포함한다.

[0008] 본 발명의 실시예들에 따른 데이터 저장 장치는, 불휘발성 메모리 장치들,

[0009] 휘발성 메모리 장치 및 호스트의 논리 주소 스트림들과 불휘발성 메모리 장치들의 물리 주소 스트림들 사이의 맵핑 정보를 관리하는 플래시 변환 계층 구조를 이용하여 호스트로부터 수신되는 입출력 요청들에 응답하여 불휘발성 메모리 장치들 및 상기 휘발성 메모리 장치를 제어하는 콘트롤러를 포함한다. 플래시 변환 계층 구조는, 호스트의 입출력 요청들에 대한 전체 논리 주소 공간을 n(n은 정수)등분하여 n개의 가상 논리 주소 스트림들로 설정하고, n개의 가상 논리 주소 스트림들을 k (k는 n보다 작은 수)개의 물리 주소 스트림들에 맵핑하

는 예비 클러스터 맵핑 테이블을 생성하고, 예비 클러스터 맵핑 테이블에 대하여 적응적 k-평균 클러스터링 알고리즘을 수행하여 n개의 가상 논리 주소 스트림들을 c(c는 k 보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 축약된 클러스터 맵핑 테이블을 생성한다.

[0010] 본 발명의 실시예들에 따른 불휘발성 메모리 장치들, 휘발성 메모리 장치 및 콘트롤러를 포함하고, 콘트롤러는 호스트로부터 수신되는 입출력 요청들에 응답하여 불휘발성 메모리 장치들 및 휘발성 메모리 장치를 제어하도록 구성되는 데이터 저장 장치의 동작 방법은, 콘트롤러에 의해 호스트의 입출력 요청들에 대한 전체 논리 주소 공간을 n(n은 정수)등분하여 n개의 가상 논리 주소 스트림들로 설정하는 단계, 콘트롤러에 의해 n개의 가상 논리 주소 스트림들을 k (k는 n보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 예비 클러스터 맵핑 테이블을 생성하는 단계, 콘트롤러에 의해 예비 클러스터 맵핑 테이블에 대하여 적응적 k-평균 클러스터링 알고리즘을 수행하여 n개의 가상 논리 주소 스트림들을 c(c는 k 보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 축약된 클러스터 맵핑 테이블을 생성하는 단계, 및 콘트롤러에 의해 축약된 클러스터 맵핑 테이블에 대하여 DFTL(Demand-based FTL) 방식을 이용하여 축약된 클러스터 맵핑 테이블에서 상대적으로 자주 참조되는 맵핑 테이블의 일부로 구성되는 캐시드 클러스터 맵핑 테이블을 생성하는 단계를 포함한다.

발명의 효과

[0011] 본 발명의 실시예들에 따른 플래시 변환 계층(FTL) 구조 및 이를 포함하는 데이터 저장 장치는, 멀티 스트림을 지원하고, 적응적 k-평균 클러스터링 알고리즘을 이용하여 축약된 클러스터 맵핑 테이블을 생성하고, 축약된 클러스터 맵핑 테이블에 의해 휘발성 메모리 장치의 데이터 버퍼 공간에서 맵핑되지 않은 물리 주소 스트림들에 할당되었던 데이터 버퍼들의 공간을 DFTL 캐시 맵 공간으로 확장함에 따라, 캐시 미스율을 줄이고, 호스트 읽기 및 쓰기 동작 시 호스트 데이터의 논리 주소를 물리 주소로 변환하는 맵핑 정보 즉, 메타 데이터의 읽기 및 쓰기 동작 횟수를 줄여서 데이터 저장 장치의 성능을 향상시킬 수 있다.

도면의 간단한 설명

[0012] 도 1은 본 발명의 실시예들에 따른 데이터 저장 장치를 포함하는 시스템을 개념적으로 설명하는 도면이다.

도 2는 도 1의 SSD에 포함되는 불휘발성 메모리 장치의 일 예를 설명하는 블록도이다.

도 3a 내지 도 3c는 도 2의 불휘발성 메모리 장치 내 메모리 셀 어레이를 설명하는 도면들이다.

도 4는 본 발명의 실시예들에 따른 도 1의 FTL 구조를 설명하는 도면이다.

도 5 및 도 6은 도 4의 FTL 구조에 의한 맵핑 구조들을 설명하는 도면들이다.

도 7 및 도 8은 본 발명의 실시예들에 따른 FTL 구조에 따른 휘발성 메모리 장치의 데이터 맵핑을 설명하는 도면들이다.

도 9 내지 도 11은 본 발명의 실시예들에 따른 FTL 구조를 포함하는 데이터 저장 장치의 동작 방법을 설명하는 도면들이다.

도 12a 내지 도 12c 그리고 도 13은 본 발명의 실시예들에 따른 FTL 구조를 다양한 워크로드들에 수행한 성능 평가들을 보여주는 도면들이다.

발명을 실시하기 위한 구체적인 내용

[0013] 도 1은 본 발명의 실시예들에 따른 데이터 저장 장치를 포함하는 시스템을 개념적으로 설명하는 도면이다.

[0014] 도 1을 참조하면, 시스템(1000)은 호스트(1100)와 데이터 저장 장치(1200)를 포함한다. 데이터 저장 장치(1200)는 SSD(Solid State Drive)로 구성되고, 시스템(1000)은 SSD 시스템일 수 있다. 이하, 데이터 저장 장치(1200)는 SSD(1200)로 통칭한다.

[0015] SSD(1200)는 멀티 스트림(Multi Stream)을 지원할 수 있다. 멀티 스트림은 쓰기 증폭(write amplication)을 감소시키고 SSD(1200)의 성능을 향상시키기 위해, SSD(1200)가 유사한 특성들(characteristics) 또는 속성들(attributes)에 따라 보다 효율적으로 데이터를 배치하도록 할 수 있다. 쓰기 증폭은 SSD(1200)에서 불휘발성 메모리 장치(NVM)에 협정된(committed) 쓰기들 횟수와 호스트(1100)로부터 오는 쓰기들 횟수의 비율로 설명될 수 있다. 높은 쓰기 증폭은 SSD(1200)에 랜덤 쓰기들(random writes)에 대한 문제를 일으켜서 쓰기 성능을 감소시킬 수 있고, 불휘발성 메모리 셀들의 마모를 증가시킬 수 있으며, 이에 따라 SSD(1200)의 내구성

(endurance)을 저하시킬 수 있다.

[0016] 호스트(1100)는 SSD(1200) 내에 스트림들을 명확히 개방시킬 수 있으며, 상이한 속성들을 갖는 데이터를 개별적으로(separately) 저장하는 인터페이스를 갖는 어플리케이션들 및 오퍼레이팅 시스템들을 제공할 수 있다. 이러한 개별 데이터 저장소들을 "스트림들"이라고 한다. 스트림들은 서로 다른 데이터 쓰기들이 언제 서로 연관되거나 비슷한 수명을 갖는지를 나타내는데 사용될 수 있다. 즉, 개별 데이터 쓰기들의 그룹은 멀티 스트림들의 일부일 수 있으며, 그리고 각 스트림은 오퍼레이팅 시스템 또는 대응하는 어플리케이션에 의해서 할당된 스트림 ID로 식별될 수 있다. 유사한 특성들 또는 속성들을 갖는 데이터는 고유한 스트림 ID로 할당되어, 그 스트림 ID에 대응하는 데이터가 SSD(1200) 내 동일한 블록에 기입될 수 있도록 한다.

[0017] 그런데, 호스트(1100)에서 실행되는 멀티 스트림을 위한 스트림 배치는 어플리케이션을 수정하거나 이전에 사용된 데이터의 로그를 테이블 또는 대기열들(queues)에서 관리함으로써 제공될 수 있다. 이는 높은 유지 관리 오버헤드로 인해 비효율적이다.

[0018] 한편, 멀티 스트림 가능 SSD(1200)에서 서로 관련있는 쓰기 동작들의 스트림들을 적응적으로 조절하여 낸드 플래시 블록에 물리적으로 함께 기록되게끔 배치한다면, 더 많은 유연성을 가능하게 하고, 쓰기 증폭을 감소시킬 수 있고, SSD(1200)의 성능을 향상시킬 수 있을 것이다.

[0019] SSD(1200)는 낸드(NAND) 플래시 기반의 SSD로 구현될 수 있고, 신호 커넥터(1201)를 통해 호스트(1100)와 신호(SIG)를 주고 받고, 전원 커넥터(1202)를 통해 전원(PWR)을 입력받을 수 있다. SSD(1200)는 SSD 콘트롤러(1210), 보조 전원 장치(1220), 복수의 불휘발성 메모리 장치들(NVMs, 1230~1250) 및 휘발성 메모리 장치(1260)를 포함할 수 있다.

[0020] SSD 콘트롤러(1210)는 호스트(1100)로부터 수신된 신호(SIG)에 응답하여 복수의 불휘발성 메모리 장치들(NVMs)을 제어할 수 있다. 신호(SIG)에는 커맨드, 어드레스, 데이터 등을 포함할 수 있다. SSD 콘트롤러(1210)는 호스트(1100)의 커맨드에 따라 복수의 불휘발성 메모리 장치들(NVMs)에 데이터를 기입하거나 복수의 불휘발성 메모리 장치들(NVMs)로부터 데이터를 독출할 수 있다.

[0021] SSD 콘트롤러(1210)는 복수의 불휘발성 메모리 장치들(NVMs)의 읽기 및 쓰기 동작들을 관리 및/또는 제어할 수 있다. 불휘발성 메모리 장치(NVM)에서 페이지는 읽기 및 쓰기 동작들의 단위이고, 블록은 소거 동작의 단위이다. 불휘발성 메모리 장치(NVM)는 덮어쓰기 동작을 지원하지 않기 때문에, 페이지에 기록된 데이터를 수정하기 위해서는 해당 페이지가 속한 블록 내의 모든 유효한 데이터(유효한 페이지)를 다른 빈 블록에 복사하고, 이전 블록을 소거하는 과정을 필요로 한다. 이러한 과정은 다수의 페이지 복사(페이지 읽기 및 쓰기) 및 소거 동작 과정을 수반하기 때문에 전반적인 불휘발성 메모리 장치(NVM)의 성능을 감소시킨다.

[0022] SSD 콘트롤러(1210)는 불휘발성 메모리 장치(NVM)의 읽기 및 쓰기 동작을 관리하는 FTL(1212)을 포함할 수 있다. FTL(1212)은 주소 맵핑 기능과 가비지 콜렉션 기능으로 구성될 수 있다. 주소 맵핑 기능은 호스트(1100)로부터 덮어쓰기 요청을 받은 경우, 해당 데이터를 원래 페이지에 덮어쓰는 대신, 다른 빈 페이지에 기록함으로써 부가적인 페이지 복사 및 소거 동작을 감소시키는 역할을 할 수 있다. 이를 위하여, 주소 맵핑 기능은 맵핑 테이블을 휘발성 메모리 장치(1260) 내에 유지해야 하고, 맵핑 테이블을 통해 호스트(1100)로부터 받은 논리 주소를 불휘발성 메모리 장치(NVM)의 물리 주소로 맵핑할 수 있다. 주소 맵핑 기능은 덮어쓰기에 따른 불필요한 읽기, 쓰기, 소거 동작들을 줄여주지만, 최신 데이터보다 오래된 데이터를 저장하고 있는 페이지들(무효화된 페이지들)을 다수 발생시킬 수 있다. 무효화된 페이지들로 인해 불휘발성 메모리 장치(NVM)의 저장 공간이 낭비되는 것을 막기 위해, FTL(1212)은 주기적으로 무효화된 페이지들을 삭제해야만 하는데, 이를 가비지 콜렉션 기능이라 칭한다.

[0023] FTL(1212)은 호스트 데이터 나 호스트 데이터의 맵핑 정보를 나타내는 맵핑 테이블 및 FTL(1212)이 관리하는 메타 데이터 각각에 대하여 가상 물리 주소 공간 상의 낸드 플래시 블록들을 이용하여 제자리 갱신이 가능한 주소 공간을 구현할 수 있다.

[0024] FTL(1212)은, 호스트 데이터를 복수의 불휘발성 메모리 장치들(1230~1250)에 쓰면 필수적으로 호스트 데이터의 논리 주소를 복수의 불휘발성 메모리 장치들(1230~1250)의 물리 주소로 변환하는 맵핑 정보를 맵핑 테이블에 생성하거나 갱신해야 한다. 또한, FTL(1212)은 해당 쓰기 동작에 의해 무효화된 페이지에 대한 유효성 정보를 나타내는 비트맵을 맵핑 테이블에 생성하거나 갱신해야 한다. FTL(1212)은 호스트(1100)로부터 전달된 읽기 및 쓰기 명령을 처리하기 위하여, 맵핑 테이블을 이용하여 주소 변환을 수행하는 소프트웨어, 하드웨어, 또는 이들의 조합으로 다양하게 구현될 수 있다.

[0025] 보조 전원 장치(1220)는 전원 커넥터(1202)를 통해 호스트(1100)와 연결될 수 있다. 보조 전원 장치(1220)는 호스트(1100)로부터 전원(PWR)을 입력받아 SSD 콘트롤러(1210)에 전원(PWR)을 공급할 수 있다. 보조 전원 장치(1220)는 SSD(1200) 내부에 위치할 수도 있고, SSD(1200) 외부에 위치할 수도 있다. 예를 들면, 보조 전원 장치(1220)는 SSD 시스템(1000)의 메인 보드에 위치하며 SSD(1200)에 보조 전원을 제공할 수 있다.

[0026] 복수의 불휘발성 메모리 장치들(1230~1250, NVM)은 SSD(1200)의 저장 매체로서 사용될 수 있다. 복수의 불휘발성 메모리 장치들(1230~1250) 각각은 플래시 메모리 장치를 포함할 수 있다. 복수의 불휘발성 메모리 장치들(1230~1250)은 복수의 채널들(CH1~CHn)을 통해 SSD 콘트롤러(1210)와 연결될 수 있다. 하나의 채널에는 복수의 불휘발성 메모리 장치들(NVM)이 연결될 수 있고, 하나의 채널에 연결되는 복수의 불휘발성 메모리 장치들(NVM)의 개수는 웨이(Way) 또는 뱅크(Bank)로 정의될 수 있다.

[0027] 휘발성 메모리 장치(1260)는 SSD 콘트롤러(1210)에 대한 입출력 버퍼로서 동작한다. 휘발성 메모리 장치(1260)는 DFTL 맵 캐시 공간(710), 스트림 맵핑 테이블 공간(720) 및 데이터 버퍼 공간(730)으로 구성될 수 있다. 휘발성 메모리 장치(1260)는 DRAM 또는 SRAM로 구현될 수 있다.

[0028] 쓰기 모드에서, SSD 콘트롤러(1210)는 호스트로부터 수신되는 호스트 데이터를 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)에 저장하고, 데이터 버퍼 공간(730)에 저장된 호스트 데이터를 복수의 불휘발성 메모리 장치들(1230~1250)에 기입할 수 있다. 실시예에 따라, SSD 콘트롤러(1210)는 호스트(1100)로부터 쓰기 요청이 들어오면, 불휘발성 메모리 장치들(NVMs)의 사양들 예컨대, 3-비트 멀티 레벨 셀(TLC) 및 HSP(High Speed Program) 방법에 기초하여 일정 양의 호스트 데이터를 모으기 위한 용도로 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)을 이용할 수 있다. 또한, 읽기 모드에서, SSD 콘트롤러(1210)는 복수의 불휘발성 메모리 장치들(1230~1250)로부터 독출되는 독출 데이터를 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)에 저장하고, 데이터 버퍼 공간(730)에 저장된 독출 데이터를 호스트(1100)에 제공할 수 있다.

[0029] SSD 콘트롤러(1210)는, 호스트 데이터의 논리 주소들과 복수의 불휘발성 메모리 장치들(1230~1250)의 물리 주소들 맵핑 정보를 관리하는 FTL(1212) 구조를 이용하여, 호스트(1100)로부터 수신되는 입출력 요청들에 응답하여 복수의 불휘발성 메모리 장치들(1230~1250) 및 휘발성 메모리 장치(1260)를 제어할 수 있다.

[0030] SSD 콘트롤러(1210)는, 호스트의 입출력 요청들에 대한 전체 논리 주소 공간을 n(n은 정수)등분하여 n개의 가상 논리 주소 스트림들로 설정할 수 있다. SSD 콘트롤러(1210)는, k-평균 클러스터링 알고리즘을 수행하여 n개의 가상 논리 주소 스트림들을 k (k는 n보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 예비 클러스터 맵핑 테이블(P_MAP, 도 5)을 생성할 수 있다. SSD 콘트롤러(1210)는, 예비 클러스터 맵핑 테이블(P_MAP)에 기초하여 k개의 물리 주소 스트림들 각각을 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)에 할당할 수 있다. 이에 따라, 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)은 k개의 물리 주소 스트림들에 대응되는 k개의 데이터 버퍼들을 포함할 수 있다.

[0031] SSD 콘트롤러(1210)는, 예비 클러스터 맵핑 테이블(P_MAP)에 대하여 적응적 k-평균 클러스터링 알고리즘을 수행하여 n개의 가상 논리 주소 스트림들을 c(c는 k 보다 작은 수)개의 물리 주소 스트림들에 맵핑하는 축약된 클러스터 맵핑 테이블(C_MAP, 도 5)을 생성할 수 있다. SSD 콘트롤러(1210)는, 축약된 클러스터 맵핑 테이블(C_MAP)을 휘발성 메모리 장치(1260)의 스트림 맵핑 테이블 공간(720)에 저장할 수 있다. 이에 따라, 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)은 c개의 물리 주소 스트림들에 대응되는 c개의 데이터 버퍼들을 포함할 수 있다.

[0032] SSD 콘트롤러(1210)는, 축약된 클러스터 맵핑 테이블(C_MAP)에 기초하여 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)에서 사용되지 않는, 즉 (k-c)개의 맵핑되지 않은 물리 주소 스트림들에 할당되었던 빈 공간을 보유할 수 있다. SSD 콘트롤러(1210)는, 메모리 장치(1260)의 데이터 버퍼 공간(730)에서 (k-c)개의 맵핑되지 않은 물리 주소 스트림들에 할당되었던 빈 공간을 DFTL 맵 캐시 공간(710)으로 확장할 수 있다.

[0033] 도 2는 도 1의 SSD(1200)에 포함되는 불휘발성 메모리 장치(NVM)의 일 예를 설명하는 블록도이다.

[0034] 도 2를 참조하면, 불휘발성 메모리 장치(NVM)는 메모리 셀 어레이(122), 어드레스 디코더(123), 제어 로직 블락(124), 페이지 버퍼부(125), 데이터 입출력 회로부(126) 그리고 전압 생성부(127)를 포함할 수 있다.

[0035] 메모리 셀 어레이(122)는 워드 라인들(WL), 스트링 선택 라인들(SSL), 그라운드 선택 라인들(GSL) 및 비트 라인들(BL)에 연결될 수 있다. 메모리 셀 어레이(122)는 워드 라인들(WL), 스트링 선택 라인들(SSL) 및 그라운드 선택 라인들(GSL)을 통해 어드레스 디코더(123)에 연결되고, 비트 라인들(BL)을 통해 페이지 버퍼부(125)에 연

결될 수 있다. 메모리 셀 어레이(122)는 복수의 메모리 블록들(BLK1 내지 BLKn)을 포함할 수 있다.

[0036] 일 실시예에서, 메모리 셀 어레이(122)는 기판 상에 3차원 구조 (또는 수직 구조)로 형성되는 3차원(three-dimensional) 메모리 셀 어레이일 수 있다. 이 경우, 메모리 셀 어레이(122)는 서로 적층되어 형성되는 복수의 메모리 셀들을 포함하는 수직 메모리 셀 스트링들을 포함할 수 있다. 다른 실시예에서, 메모리 셀 어레이(122)는 기판 상에 2차원 구조 (또는 수평 구조)로 형성되는 2차원(two-dimensional) 메모리 셀 어레이일 수 있다.

[0037] 3차원 메모리 셀 어레이는 실리콘 기판 위에 배치되는 활성 영역과, 메모리 셀들의 동작과 관련된 회로로서 상기 기판 상에 또는 상기 기판 내에 형성된 회로를 가지는 메모리 셀 어레이들의 적어도 하나의 물리적 레벨에 모놀리식으로 형성된다. 상기 용어 "모놀리식"은 상기 어레이를 구성하는 각 레벨의 층들이 상기 어레이 중 각 하부 레벨의 층들의 바로 위에 적층되어 있음을 의미한다. 본 발명의 기술적 사상에 의한 일 실시예에서, 3차원 메모리 셀 어레이는 적어도 하나의 메모리 셀이 다른 메모리 셀의 위에 위치하도록 수직 방향으로 배치된 낸드 스트링들을 포함한다. 상기 적어도 하나의 메모리 셀은 전하 트랩층을 포함할 수 있다. 미국 특허공개공보 제7,679,133호, 미국 특허공개공보 제8,553,466호, 미국 특허공개공보 제8,654,587호, 미국 특허공개공보 제8,559,235호, 및 미국 특허출원공개공보 제2011/0233648호는 3차원 메모리 어레이가 복수 레벨로 구성되고 워드라인들 및/또는 비트 라인들이 레벨들 간에 공유되어 있는 3차원 메모리 어레이에 대한 적절한 구성들을 상술하는 것들로서, 본 명세서에 인용 형식으로 결합된다.

[0038] 메모리 블록들(BLK1~BLKn) 각각은 복수의 메모리 셀들 및 복수의 선택 트랜지스터들을 포함할 수 있다. 메모리 셀들은 워드라인들(WL)에 연결되고, 선택 트랜지스터들은 스트링 선택 라인들(SSL) 또는 그라운드 선택 라인들(GSL)에 연결될 수 있다. 메모리 블록들(BLK1~BLKn) 각각의 메모리 셀들은 1-비트 데이터를 저장하는 싱글 레벨 셀들 또는 M(M은 2 또는 그 보다 큰 정수)-비트 데이터를 저장하는 멀티 레벨 셀들로 구성될 수 있다.

[0039] 어드레스 디코더(123)는 메모리 셀 어레이(122)의 복수의 메모리 블록들(BLK1 내지 BLKn) 중 하나를 선택할 수 있고, 선택된 메모리 블록의 워드라인들(WL) 중 하나를 선택할 수 있고, 복수의 스트링 선택 라인들(SSL) 중 하나를 선택할 수 있다.

[0040] 제어 로직 블락(124)은 커맨드(CMD), 어드레스(ADDR) 및 제어 신호(CTRL)를 기초로 하여, 메모리 셀 어레이(122)에 대한 프로그램, 독출 및 소거 동작을 수행하기 위한 각종 제어 신호를 출력할 수 있다. 제어 로직 블락(124)은 어드레스 디코더(123)에 로우 어드레스(X-ADDR)를 제공할 수 있고, 데이터 입출력 회로부(126)에 칼럼 어드레스(Y-ADDR)를 제공할 수 있고, 전압 생성부(127)에 전압 제어 신호(CTRL_Vol)를 제공할 수 있다.

[0041] 페이지 버퍼부(125)는 동작 모드에 따라 기입 드라이버로서 또는 감지 증폭기로서 동작할 수 있다. 독출 동작시, 페이지 버퍼부(125)는 제어 로직 블락(124)의 제어에 따라 선택된 메모리 셀의 비트라인(BL)을 센싱할 수 있다. 센싱된 데이터는 페이지 버퍼부(125) 내부에 구비되는 래치들에 저장될 수 있다. 페이지 버퍼부(125)는 제어 로직 블락(124)의 제어에 따라 래치들에 저장된 데이터를 데이터 입출력 회로부(126)로 덤핑할 수 있다.

[0042] 데이터 입출력 회로부(126)는 불휘발성 메모리 장치(NVM)의 외부로부터 입출력 라인(I/O)을 통해서 제공되는 명령어(CMD), 어드레스(ADDR), 제어 신호(CTRL) 및 데이터(DATA)를 일시 저장할 수 있다. 데이터 입출력 회로부(126)는 불휘발성 메모리 장치(NVM)의 읽기 데이터를 일시 저장하고, 지정된 시점에 입출력 라인(I/O)을 통해서 외부로 출력할 수 있다.

[0043] 전압 생성부(127)는 전압 제어 신호(CTRL_Vol)를 기초로 하여 메모리 셀 어레이(122)에 대한 프로그램, 독출 및 소거 동작을 수행하기 위한 다양한 종류의 전압들을 생성할 수 있다. 구체적으로, 전압 생성부(127)는 워드 라인 전압(VWL), 예를 들어, 프로그램 전압, 독출 전압, 패스 전압, 소거 검증 전압 또는 프로그램 검증 전압 등을 생성할 수 있다. 또한, 전압 생성부(127)는 전압 제어 신호(CTRL_Vol)를 기초로 하여 스트링 선택 라인 전압 및 그라운드 선택 라인 전압을 생성할 수 있다. 또한, 전압 생성부(127)는 메모리 셀 어레이(122)에 제공할 소거 전압을 생성할 수 있다.

[0044] 도 3a 내지 도 3c는 도 2의 불휘발성 메모리 장치 내 메모리 셀 어레이를 설명하는 도면들이다. 도 3a 및 도 3b는 도 2의 불휘발성 메모리 장치(NVM) 내 메모리 셀 어레이(122)에 포함되는 제1 메모리 블록(BLK1)의 등가회로도와 사시도를 나타낸다. 도 3a 및 도 3b에 도시된 메모리 셀 어레이(122)는 기판 상에 3차원 구조로 형성되는 3차원 메모리 셀 어레이를 나타내고, 메모리 셀 어레이(122)에 포함되는 복수의 메모리 셀 스트링들 또는 낸드 스트링들은 기판과 수직한 방향으로 형성될 수 있다.

[0045] 도 3a을 참조하면, 제1 메모리 블록(BLK1)은 낸드 스트링들(NS11~NS33), 워드 라인들(WL1~WL8), 비트 라인들(BL1~BL3), 그라운드 선택 라인들(GSL1~ GSL3), 스트링 선택 라인들(SSL1~SSL3) 및 공통 소스 라인(CSL)을 포

함할 수 있다. 낸드 스트링들(NS11~NS33)은 비트라인들(BL1, BL2, BL3)과 공통 소스 라인(CSL) 사이에 연결되고, 낸드 스트링들(NS11~NS33) 각각은 직렬로 연결된 스트링 선택 트랜지스터(SST), 복수의 메모리 셀들(MCs) 및 그라운드 선택 트랜지스터(GST)를 포함할 수 있다. 낸드 스트링들(NS11~NS33) 각각이 8개의 메모리 셀들(MCs)을 포함하는 것으로 도시되어 있으나, 본 발명은 이에 한정되지 않는다.

[0046] 스트링 선택 트랜지스터(SST)는 대응하는 스트링 선택 라인(SSL1, SSL2, SSL3)에 연결될 수 있다. 복수의 메모리 셀들(MC)은 각각 대응하는 워드 라인(WL1~WL8)에 연결될 수 있다. 그라운드 선택 트랜지스터(GST)는 대응하는 그라운드 선택 라인(GSL1, GSL2, GSL3)에 연결될 수 있다. 스트링 선택 트랜지스터(SST)는 대응하는 비트라인(BL1, BL2, BL3)에 연결되고, 그라운드 선택 트랜지스터(GST)는 공통 소스 라인(CSL)에 연결될 수 있다. 동일 높이의 워드 라인 (예를 들면, WL1) 공통으로 연결되고, 스트링 선택 라인(SSL1, SSL2, SSL3) 및 그라운드 선택 라인(GSL1, GSL2, GSL3)은 각각 분리될 수 있다.

[0047] 낸드 스트링들(NS11~NS33) 중에서 제1 행의 낸드 스트링들(NS11, NS12, NS13)은 제1 스트링 선택 라인(SSL1)에 공통으로 연결될 수 있다. 제2 행의 낸드 스트링들(NS21, NS22, NS23)은 제2 스트링 선택 라인(SSL2)에 공통으로 연결될 수 있다. 제3 행의 낸드 스트링들(NS31, NS32, NS33)은 제3 스트링 선택 라인(SSL3)에 공통으로 연결될 수 있다. 제1 행의 낸드 스트링들(NS11, NS12, NS13)은 제1 플레인을 구성할 수 있고, 제2 행의 낸드 스트링들(NS21, NS22, NS23)은 제2 플레인을 구성할 수 있고, 제3 행의 낸드 스트링들(NS31, NS32, NS33)은 제3 플레인을 구성할 수 있다. 본 실시예에서, 제1 메모리 블록(BLK1)은 3개의 비트라인들(BL1~BL3)에 연결되는 3개의 플레인들을 보여준다.

[0048] 제1 메모리 블록(BLK1) 내 낸드 스트링들의 행들 및 열들의 수는 증가 또는 감소할 수 있다. 낸드 스트링들의 행들의 수가 변경됨에 따라, 플레인들의 수가 변경될 수 있다. 낸드 스트링들의 열들의 수가 변경됨에 따라, 낸드 스트링들의 칼럼들에 연결되는 비트라인들의 수, 그리고 하나의 스트링 선택 라인에 연결되는 낸드 스트링들의 수 또한 변경될 수 있다. 낸드 스트링들의 높이는 증가 또는 감소될 수 있다. 예를 들어, 낸드 스트링들 각각에 적층되는 메모리 셀들의 수는 증가 또는 감소될 수 있다.

[0049] 도 3b를 참조하면, 제1 메모리 블록(BLK1)은 기판(SUB)에 대해 수직 방향으로 형성되어 있다. 기판(SUB)은 제1 도전형(예를 들어, p 타입)을 가지며, 기판(SUB) 상에 제1 방향(y)을 따라 신장되고, 제2 도전형(예를 들어, n 타입)의 불순물들이 도핑된 공통 소스 라인(CSL)이 제공된다. 인접한 두 공통 소스 라인들(CSL) 사이의 기판(SUB)의 영역 상에, 제1 방향(y)을 따라 신장되는 절연막들(IL)이 제3 방향을 따라 순차적으로 제공되며, 절연막들(IL)은 제3 방향(z)을 따라 특정 거리만큼 이격된다. 인접한 두 공통 소스 라인들(CSL) 사이의 기판(SUB)의 영역 상에, 제1 방향(y)을 따라 순차적으로 배치되며, 제3 방향(z)을 따라 절연막들(IL)을 관통하는 필라들(pillars)(P)이 제공된다. 예를 들어, 필라들(P)은 절연막들(IL)을 관통하여 기판(SUB)과 컨택할 것이다. 구체적으로, 각 필라(P)의 표면층(surface layer)(S)은 제1 타입을 갖는 실리콘 물질을 포함할 수 있고, 채널 영역으로 기능할 수 있다. 한편, 각 필라(P)의 내부층(I)은 실리콘 산화물과 같은 절연 물질 또는 에어 갭(air gap)을 포함할 수 있다.

[0050] 인접한 두 공통 소스 라인들(CSL) 사이의 영역에서, 절연막들(IL), 필라들(P) 및 기판(SUB)의 노출된 표면을 따라 전하 저장층(CS)이 제공된다. 전하 저장층(CS)은 터널링 절연층, 전하 트랩층 및 블로킹 절연층을 포함할 수 있다. 또한, 인접한 두 공통 소스 라인들(CSL) 사이의 영역에서, 전하 저장층(CS)의 노출된 표면 상에, 선택 라인들(GSL, SSL) 및 워드 라인들(WL1 내지 WL8)과 같은 게이트 전극(GE)이 제공된다. 복수의 필라들(P) 상에는 드레인 컨택들(DR)이 각각 제공된다. 예를 들어, 드레인 컨택들(DR)은 제2 도전형을 갖는 불순물들이 도핑된 실리콘 물질을 포함할 수 있다. 드레인들(DR) 상에, 제2 방향(x)으로 신장되고 제1 방향(y)을 따라 특정 거리만큼 이격되어 배치된 비트 라인들(BL1 내지 BL3)이 제공된다.

[0051] 도 3c는 도 3a에 도시된 복수의 메모리 셀들(MCs)의 문턱 전압 산포를 예시적으로 보여주는 산포도이다. 도 3c에서, 가로축은 문턱 전압(Vth)을 나타내고, 세로축은 메모리 셀들(MC)의 개수를 나타낸다.

[0052] 도 3c를 참조하면, 복수의 메모리 셀들(MCs) 각각은 3-비트 멀티 레벨 셀(TLC)일 수 있다. 복수의 메모리 셀들(MCs)은 소거 상태(E)를 가질 수 있다. HSP(High Speed Program) 방법을 이용하여, 소거 상태(E)의 복수의 메모리 셀들은 소거 상태(E) 및 제1 내지 제7 프로그램 상태들(P1~P7) 중 하나를 갖도록 프로그램될 수 있다. 본 발명의 범위가 이에 한정되는 것은 아니며, 메모리 셀들 각각은 2-비트 멀티 레벨 셀(MLC) 또는 4-비트 이상의 비트를 저장하는 멀티 레벨 셀(QLC)일 수 있다.

[0053] 도 4는 본 발명의 실시예들에 따른 도 1의 FTL(1212) 구조를 설명하는 도면이다. 도 5 및 도 6은 도 4의

FTL(1212) 구조에 의한 맵핑 구조들을 설명하는 도면들이다.

[0054] 도 4를 참조하면, FTL(1212) 구조는 호스트(1100, 도 1)에서 제공되는 호스트 데이터에 대응하는 논리 주소 공간을 복수의 불휘발성 메모리 장치들(NVMs, 도 1)의 물리 주소로 주소 변환하도록 구성될 수 있다. FTL(1212) 구조는 스트림 클러스터링 매니저(410), 적응적 k-평균 클러스터링 알고리즘 모듈(420) 및 DFTL 캐시 맵 매니저(430)를 포함할 수 있다. 스트림 클러스터링 매니저(410), 적응적 k-평균 클러스터링 알고리즘 모듈(420) 및 DFTL 캐시 맵 매니저(430) 각각은, 기능 및 논리적으로 분리될 수 있음을 나타내는 것이며, 반드시 구성들 각각이 별도의 물리적 장치로 구분되거나 별도의 코드로 작성되는 것을 의미하는 것은 아니다.

[0055] FTL(1212) 구조에서, 모듈 및/또는 매니저는 호스트 데이터, 맵핑 데이터 또는 스트림 특성들과 같은 메타 데이터를 처리하므로, 처리 모듈이라고 지칭될 수 있다. 처리 모듈은, ROM 디바이스들과 같은 기록 불가능 저장 매체 상에 영구적으로 저장된 프로그램 코드들이거나, 플로피 디스크, 자기 테이프, CD들, RAM 디바이스들 및/또는 다른 자기 및 광학 매체와 같은 비일시적인 기록 가능 저장 매체 상에 변경 가능한 프로그램 코드이거나, 또는 인터넷 또는 전화 모뎀 라인들과 같은 전자 네트워크에서와 같이 통신 매체를 통해 컴퓨터로 전달된 프로그램 코드들과 같은 형태로 인식될 수 있다.

[0056] 실시예에 따라, 처리 모듈은, 소프트웨어 실행 가능 객체에서 또는 명령어들에 응답하는 프로세서에 의한 실행을 위해 인코딩된 명령어들의 세트로서 구현된 컴퓨터 프로그램 제품으로 제공될 수 있다.

[0057] 실시예에 따라, 처리 모듈은, 주문형 집적 회로들(ASICs), 필드 프로그래머블 게이트 어레이들(FPGAs), 상태 머신들, 제어기들, 또는 다른 하드웨어 컴포넌트들 또는 디바이스들과 같은 하드웨어 컴포넌트들, 또는 하드웨어, 소프트웨어 및 펌웨어 컴포넌트들의 조합을 사용하여 전제적으로 또는 부분적으로 구현될 수 있다.

[0058] 스트림 클러스터링 매니저(410)는 호스트 데이터를 분석하여 상이한 스트림들에 그룹화 또는 클러스터링할 수 있다. 호스트 데이터는 호스트 데이터에 대응하는 논리 주소 공간으로 설명될 수 있다. 스트림 클러스터링 매니저(410)는 호스트(1100)에서 제공되는 전체 논리 주소 공간을 n (n은 가상 논리 주소 클러스터들의 개수) 등분하여 n개의 가상 논리 주소 스트림들(Lstreams)을 설정할 수 있다. 예시적으로, 스트림 클러스터링 매니저(410)는 호스트(1100)에서 제공되는 전체 논리 주소 공간이 16GB 인 경우, 도 5에 도시된 바와 같이, 16GB 논리 주소 공간을 200 등분하여 200개의 가상 논리 주소 스트림들(Lstream 0 ~ Lstream 199)을 설정할 수 있다.

[0059] 스트림 클러스터링 매니저(410)는 n개의 가상 논리 주소 스트림들(Lstreams)에 대응하는 하나 이상의 스트림 속성들을 추적(trace)하여 획득할 수 있다. 획득될 수 있는 스트림 속성들에는 예를 들면, 예상 수명(life time), 빈도(frequency), 시간적 지역성(temporal locality), 순차성(sequentiality), 입출력(I/O) 크기, 입출력(I/O) 격리 레벨, QoS(Quality of Service) 요구사항 및/또는 업데이트 카운트(update count) 등이 포함될 수 있다.

[0060] 예상 수명의 속성은, 가상 논리 주소 스트림들(Lstreams)이 시간 단위로 측정되어 임의의 특정 가상 논리 주소 스트림이 무효화되기 전 소정의 시간 (예를 들면 10초) 동안 또는 일정 횟수만큼 쓰인 양을 기준으로 SSD(1200)에 저장될 것으로 기대되는지에 해당할 수 있다. 또는, 예상 수명은 임의의 특정 가상 논리 주소 스트림이 ″ 핫(hot)″ 인지 아니면 ″ 콜드(cold)″ 인지의 관점에서 측정될 수 있다. 여기에서, ″ 핫″ 은 해당 가상 논리 주소 스트림의 수명이 짧다는 것을 의미하고, ″ 콜드″ 는 수명이 길다는 것을 의미한다. 예상 수명은 핫 정도(hotness) 또는 콜드 정도(coldness)에 따라 측정될 수 있다.

[0061] 실시예에 따라, 예상 수명의 속성은 호스트(1100)의 중복 쓰기 요청에 기초하여 특정 가상 논리 주소 스트림이 오버라이트되면 예상 수명의 속성을 나타내는 비트맵이 ″ 0″ 으로 클리어(clear)되고, 비트맵 ″ 0″ 의 합산된 업데이트 카운트 값으로 나타낼 수 있다. 업데이트 카운트 값은 중복 쓰기 횟수로 볼 수 있다. 업데이트 카운트 값이 상대적으로 큰 가상 논리 주소 스트림들은 자주 쓰이는 핫 스트림들을 나타내고, 상대적으로 작은 업데이트 카운트 값을 갖는 가상 논리 주소 스트림들은 자주 쓰이지 않는 콜드 스트림들을 나타낼 수 있다.

[0062] 빈도의 속성은, 호스트 데이터 입력의 결과로서 특정 가상 논리 주소 스트림들(Lstreams)이 얼마나 자주 억세스되었는지에 해당할 수 있다. 빈도의 속성을 획득하기 위하여, 임의의 특정 가상 논리 주소 스트림의 억세스 수가 미리 설정된 임계값 (예를 들면, 4번) 보다 많은 경우, 그 특정 가상 논리 주소 스트림에 대응하는 빈도 속성을 나타내는 비트맵이 ″ 1″ 로서 할당되고, 그렇지 않으면 ″ 0″ 으로 할당될 수 있다.

[0063] 시간적 지역성의 속성은, 특정 가상 논리 주소 스트림들(Lstreams)이 얼마나 최근에 억세스되려고 시도되었는지에 해당할 수 있다. 시간적 지역성의 속성을 획득하기 위하여, 최대 타임스탬프(timeatamp)의 미리 설정된 임계값 (예를 들면, 50%의 임계값) 이후에 억세스되는 특정 가상 논리 주소 스트림들은 시간적 지역성 속성을 나

타내는 비트맵에 ″1″이 할당되어 최근의 것으로 표시될 수 있고, 그렇지 않으면 ″0″으로 할당될 수 있다. 타임스탬프는 해당 가상 논리 주소 스트림에 데이터를 저장하기 위한 생성 시간, 수정 시간, 읽기 시간 등의 정보를 포함할 수 있다.

[0064] 순차성의 속성은, 호스트 데이터 입력이 순차적인 순서로 특정 가상 논리 주소 스트림들에 억세스하고자 시도하는지의 여부에 해당할 수 있다. 순차성의 속성을 획득하기 위하여, 소정의 미리 정의된 윈도우 크기에서 이전 특정 가상 논리 주소 스트림들을 검색(look up)할 수 있다. 일치하는 것이 발견되면, 순차성 속성을 나타내는 비트맵은 ″1″로 할당되고, 그렇지 않으면 ″0″으로 할당될 수 있다.

[0065] 입출력(I/O) 크기의 속성은, 호스트 데이터 입력이 한번에 둘 이상의 페이지들에 기입되거나 독출되는지 여부에 대응할 수 있다. 입출력(I/O) 격리 레벨의 속성은, 호스트 데이터 입력이 불휘발성 메모리 장치들(NVMs)의 채널, 웨이, 플레인 또는 블록 레벨로 다른 데이터와 격리된 데이터를 필요로 하는지 아니면 필요로 하지 않는지 여부에 해당할 수 있다. QoS 요구사항의 속성은 데이터에 대한 입출력 요청들에서 사용된 대역폭 또는 기입되거나 독출된 데이터에 대한 레이턴시 요구 사항들에 대응할 수 있다.

[0066] 스트림 클러스터링 매니저(410)는 n개의 가상 논리 주소 스트림들(Lstreams)에 대응하는 속성들 중 선택된 속성을 기반으로 n개의 가상 논리 주소 스트림들(Lstreams)을 복수의 불휘발성 메모리 장치들(NVMs)의 k(k은 물리 주소 클러스터들의 개수)개의 물리 주소 스트림들(Pstream)로 변환할 수 있다. 설명의 편의를 위하여, 복수의 불휘발성 메모리 장치들(1230~1250)은 16개의 물리 주소 스트림들(Pstream 0 ~ Pstream 15)로 할당되는 것으로 가정한다.

[0067] 본 실시예에서, 스트림 클러스터링 매니저(410)는 n개의 가상 논리 주소 스트림들(Lstreams)에 대응하는 속성들 중 예상 수명을 기반으로 FTL(1212) 구조에서 맵핑 단위를 관리하면서 사용하는 유효한 비트맵 자료 구조를 이용한다.

[0068] 스트림 클러스터링 매니저(410)는, 200개의 가상 논리 주소 스트림들(Lstream 0 ~ Lstream 199)의 예상 수명 속성의 비트맵 ″0″ 업데이트 카운트를 표 1과 같이 획득할 수 있다.

표 1

[0069]

Lstream Index	Bitmap ″0″ Update Counts
0	10
1	500
2	2
3	1000
…	…
198	50000
199	0

[0070] 스트림 클러스터링 매니저(410)는, 표 1의 가상 논리 주소 스트림들(Lstream 0 ~ Lstream 199)의 비트맵 ″0″ 업데이트 카운트가 비슷한 가상 논리 주소 스트림들을 그룹화 또는 클러스터링하여 해당 물리 주소 스트림들(Pstream 0 ~ Pstream 15)에 맵핑되도록 구성될 수 있다. 도 5에 도시된 바와 같이, 스트림 클러스터링 매니저(410)는, 200개의 가상 논리 주소 스트림들(Lstream 0 ~ Lstream 199)에 대한 비슷한 업데이트 카운트를 갖는 가상 논리 주소 스트림들(Lstreams)끼리 클러스터링하는 k-평균 클러스터링 알고리즘을 수행할 수 있다. 예시적으로, k=16 이라고 가정하면, 스트림 클러스터링 매니저(410)는, 200개의 가상 논리 주소 스트림들(Lstream 0 ~ Lstream 199)을 16개의 물리 주소 스트림들(Pstream 0 ~ Pstream 15)에 맵핑되도록 주소 변환하는 예비 클러스터 맵핑 테이블(P_MAP)을 생성할 수 있다.

[0071] k-평균 클러스터링 알고리즘은 해당 가상 주소 스트림(Lstream)이 해당 물리 주소 스트림(Pstream)에 속할 확률과 해당 가상 주소 스트림(Lstream)과 해당 물리 주소 스트림(Pstream)의 중심 간의 거리의 곱에 대체로 비례하는 기법이다. k-평균 클러스터링 알고리즘은, 수학식 1로 주어지는 목적 함수 J를 최소화하는 클러스터링 기법을 실행할 수 있다.

수학식 1

[0072] $$J=\sum_{j=0}^{k-1}\sum_{i=0}^{n-1}(P(Cj\mid Xi))^{\Phi}\cdot D_{ij}^{2}$$

[0073] 여기에서, i, j는 각각 가상 논리 주소 스트림들(Lstreams)의 인덱스 및 물리 주소 스트림들(Pstreams)의 인덱스를 나타낸다. n은 가상 논리 주소 스트림들(Lstreams)의 개수며, k는 클러스터링을 통하여 얻고자하는 클러스터들의 개수를 나타내는 파라미터들이다. 본 실시예에서, n=200, k=16으로 설정될 수 있다. ϕ는 퍼지(fuzzy) 지수로서, ϕ=0 이면 일반적인 k-평균 클러스터링 알고리즘에 따른 목적 함수를 나타낸다. 실시예에 따라, ϕ는 1보다 크게 설정될 수 있다.

[0074] 수학식 1에서, $(P(Cj\mid Xi))^{\Phi}$ 은 해당 가상 주소 스트림(Lstream) Xi가 Cj를 중심으로 하는 클러스터에 속할 확률을 의미하고, D_{ij}^{2} 는 수학식 2로 주어지는 마할라노비스 거리(Mahalanobis distance)를 나타낸다.

수학식 2

[0075] $$D_{ij}=\sqrt{(Xi-Cj)^{T}S_{q}^{-1}(Xi-Cj)}$$

[0076] 마할라노비스 거리는 중심과의 거리가 표준 편차의 몇배인지를 나타내는 값이다. 중심은 가상 논리 주소 스트림들(Lstreams)의 선택된 속성을 기반으로 한 물리 주소 스트림들(Pstreams)의 전체 평균으로 볼 수 있다. 마할라노비스 거리가 높다면 중심에서 멀리 떨어져 있다고 볼 수 있고, 낮다면 중심에 가깝다고 볼 수 있다.

[0077] k-평균 클러스터링 알고리즘은, 초기화 단계에서 수학식 3을 사용하여 균일(uniform) 확률 분포로 확률 값을 할당할 수 있다.

수학식 3

[0078] $$\sum_{j=0}^{k-1}P(Cj\mid Xi)=1,$$

[0079] k-평균 클러스터링 알고리즘에서, 수학식 1의 목적 함수 J를 최소화하는 $P(Cj\mid Xi)$ 및 Cj 는, 목적 함수 J를 각각 $P(Cj\mid Xi)$ 및 Cj 로 편미분한 결과를 0으로 하는 값으로서, 수학식 4 및 수학식 5이 주어질 수 있다.

수학식 4

[0080] $$P(Cj\mid Xi)=\frac{(1/Dij)^{1/(\Phi-1)}}{\sum_{r=0}^{k-1}(1/Dij)^{1/(\Phi-1)}}$$

수학식 5

[0081]
$$Cj=\frac{\sum_{i=0}^{n-1}(P(Cj \mid Xi))^{\Phi}\cdot Xi}{\sum_{i=0}^{n-1}(P(Cj \mid Xi))^{\Phi}}$$

[0082] k-평균 클러스터링 알고리즘은, 충분히 수렴되는 데이터가 확보될 때 (예컨대, 반복 횟수를 소정의 횟수 이상한 경우, 클러스터의 구성원들이 변동이 없어진 경우, 목적 함수 J의 값이 미리 설정된 임계치 보다 작은 경우)까지 수학식 4 및 5의 계산을 반복 수행할 수 있다.

[0083] 스트림 클러스터링 매니저(410)는, 표 1의 예상 수명 속성의 비트맵 ″0″ 업데이트 카운트를 기반으로 k-평균 클러스터링 알고리즘을 수행하여, 200개의 가상 논리 주소 스트림들(Lstream 0 ~ Lstream 199)을 16개의 물리 주소 스트림들(Pstream 0 ~ Pstream 15)에 맵핑되도록 주소 변환하는 예비 클러스터 맵핑 테이블(P_MAP)을 표 2와 같이 생성할 수 있다.

표 2

[0084]

Lstream Index	Pstream Index
0	0
1	1
2	0
3	2
…	…
198	15
199	0

[0085] 도 5를 참조하면, 표 2의 예비 클러스터 맵핑 테이블(P_MAP)에 의해, Lstream 0 가상 논리 주소 스트림은 Pstream 0 물리 주소 스트림으로 변환되고, Lstream 1 가상 논리 주소 스트림은 Pstream 1 물리 주소 스트림으로 변환되고, Lstream 2 가상 논리 주소 스트림은 Pstream 0 물리 주소 스트림으로 변환되고, Lstream 3 가상 논리 주소 스트림은 Pstream 2 물리 주소 스트림으로 변환될 수 있다. Lstream 4 내지 Lstream 197 가상 논리 주소 스트림들 각각은 Pstream 0 내지 Pstream 15 물리 주소 스트림들로 변환될 수 있다. Lstream 198 가상 논리 주소 스트림은 Pstream 15 물리 주소 스트림으로 변환되고, Lstream 199 가상 논리 주소 스트림은 Pstream 0 물리 주소 스트림으로 변환될 수 있다.

[0086] 적응적 k-평균 클러스터링 알고리즘 모듈(420)은, n개의 가상 논리 주소 스트림들(Lstreams)과 k개의 물리 주소 스트림들(Pstreams) 사이의 맵핑 정보를 나타내는 예비 클러스터 맵핑 테이블(P_MAP)에 대하여 적응적 k-평균 클러스터링 알고리즘이 수행되도록 구성될 수 있다.

[0087] 적응적 k-평균 클러스터링 알고리즘 모듈(420)은 n개의 가상 논리 주소 스트림들(Lstreams)의 선택된 속성을 기반으로, k개의 물리 주소 스트림들(Pstreams) 사이의 중심(centroid, 또는 평균)을 계산하고, k개의 물리 주소 스트림들(Pstreams)끼리의 거리를 계산하고, 가까운 물리 주소 스트림들(Pstreams)을 합병(merge)할 수 있다. 적응적 k-평균 클러스터링 알고리즘 모듈(420)은 n개의 가상 논리 주소 스트림들(Lstreams)과 맵핑에 사용되는 물리 주소 스트림들(Pstreams)의 수가 줄어들도록 반복 수행될 수 있다. 적응적 k-평균 클러스터링 알고리즘 모듈(420)은, 가까운 물리 주소 스트림들(Pstreams)을 합병(merge)하여 클러스터링하고, 클러스터링되지 않는 물리 주소 스트림들은 새로운 클러스터들로 할당할 수 있다.

[0088] 따라서, 적응적 k-평균 클러스터링 알고리즘 모듈(420)은, 예비 클러스터 맵핑 테이블(P_MAP)의 n개 가상 논리 주소 스트림들(Lstreams)에 맵핑되는 k개 물리 주소 스트림들(Pstreams)을 축약하여, n개의 가상 논리 주소 스트림들(Lstreams)이 c(c는 k 보다 작은 수)개의 물리 주소 스트림들(Pstreams)로 변환되도록 하는 축약된 클러스터 맵핑 테이블(C_MAP, 도 6)을 생성할 수 있다.

[0089] 실시예에 따라, 적응적 k-평균 클러스터링 알고리즘 모듈(420)은, 호스트(1100)에서 쓰기 동작 시에 일정 타임

스탬프를 기반으로 워크로드마다 적응적으로 얻어진 각 클러스터간 최소 거리, 평균 거리 또는 최대 거리 등을 기준으로 k-평균 클러스터링 알고리즘에서 사용하는 k 값을 c 값으로 조절할 수 있다.

[0090] 적응적 k-평균 클러스터링 알고리즘 모듈(420)은, 도 6에 도시된 바와 같이, 200개의 가상 논리 주소 스트림들(Lstream 0 ~ Lstream 199)을 8개의 물리 주소 스트림들(Pstream 0 ~ Pstream 7)로 변환하는 축약된 클러스터 맵핑 정보(C_MAP)를 생성할 수 있다.

[0091] 예시적으로, 도 6의 축약된 클러스터 맵핑 테이블(C_MAP)에 의해, Lstream 0 가상 논리 주소 스트림은 Pstream 0 물리 주소 스트림으로 변환되고, Lstream 1 가상 논리 주소 스트림은 Pstream 1 물리 주소 스트림으로 변환되고, Lstream 2 가상 논리 주소 스트림은 Pstream 0 물리 주소 스트림으로 변환되고, Lstream 3 가상 논리 주소 스트림은 Pstream 1 물리 주소 스트림으로 변환될 수 있다. Lstream 4 내지 Lstream 197 가상 논리 주소 스트림들 각각은 Pstream 0 내지 Pstream 7 물리 주소 스트림들로 변환될 수 있다. Lstream 198 가상 논리 주소 스트림은 Pstream 7 물리 주소 스트림으로 변환되고, Lstream 199 가상 논리 주소 스트림은 Pstream 0 물리 주소 스트림으로 변환될 수 있다.

[0092] 적응적 k-평균 클러스터링 알고리즘 모듈(420)에 의해 축약된 클러스터 맵핑 테이블(C_MAP)의 크기(k개 물리 주소 스트림들)는 예비 클러스터 맵핑 테이블(P_MAP)의 크기(c(c<k)개의 물리 주소 스트림들)보다 작을 것이다. FTL(1212) 구조는 휘발성 메모리 장치(1260)의 스트림 맵핑 테이블 공간(720)에 축약된 클러스터 맵핑 테이블(C_MAP)을 저장할 수 있다. 이에 따라, 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)에는 축약된 클러스터 맵핑 테이블(C_MAP)의 c개의 물리 주소 스트림들에 대응되는 c개의 데이터 버퍼들이 유지될 수 있다. 그리고, 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)에는 (k-c)개의 맵핑되지 않은 물리 주소 스트림들에 대응하는 데이터 버퍼 공간이 비어 있을 것이다.

[0093] FTL(1212) 구조는 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)에서 (k-c)개의 맵핑되지 않은 물리 주소 스트림들에 대응하는 데이터 버퍼 공간을 DFTL 맵 캐시 공간(710)으로 확장하도록 구성될 수 있다. 즉, FTL(1212) 구조는 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)에서 사용되지 않은 물리 주소 스트림들의 수를 적응적으로 변경시켜 DFTL 맵 캐시 공간(710)으로 이용할 수 있다.

[0094] 다시 도 4를 참조하면, FTL(1212) 구조는 호스트(1110, 도 1)에서 제공되는 호스트 데이터에 대응하는 논리 주소 공간을 복수의 불휘발성 메모리 장치들(NVMs, 도 1)의 물리 주소로 변환하는 맵핑 테이블을 관리할 수 있다. FTL(1212) 구조는, 맵핑 테이블의 단위에 따라, 페이지 단위로 맵핑하는 페이지 맵핑 방법, 블록 단위를 기준으로 하는 블록 맵핑 방법, 페이지 맵핑 방법과 블록 맵핑 방법의 장점들을 취한 하이브리드 맵핑 방법을 수행할 수 있다.

[0095] 페이지 맵핑 방법은 쓰기의 가장 작은 단위인 페이지를 기준으로 맵핑하기 때문에, 불휘발성 메모리 장치들(NVMs)의 저장 공간이 클수록 맵핑 테이블의 크기가 커지고, 맵핑 테이블을 유지하는데 필요한 휘발성 메모리 장치(1260, 도 1)의 저장 공간 또한 늘어나야 하는 문제점이 있다. 이 문제점은 필요한 맵핑 테이블 일부만을 휘발성 메모리 장치(1260)에 적재하는 DFTL(Demand-based FTL) 방식으로 해결될 수 있다.

[0096] DFTL 방식은, 페이지 맵핑 방법과 동일하게 페이지 단위로 맵핑 테이블을 관리할 수 있다. 그러나, DFTL 방식은 맵핑 테이블을 모두 휘발성 메모리 장치(1260)에 로딩(loading)하는 대신에 캐싱(caching)할 수 있다. DFTL 방식은 캐시된 맵핑 테이블을 휘발성 메모리 장치(1260)에 로딩하여, 주소 변환시에 페이지 맵핑 기법과 동일한 결과를 발생시키면서 휘발성 메모리 장치(1260)의 사용량을 감소시킬 수 있다. 그러나, DFTL 방식은 해당 주소가 캐시된 맵핑 테이블에 존재하지 않을 경우, 해당 주소를 불러오기 위해 추가적인 불휘발성 메모리 장치(NVM)의 읽기, 쓰기 동작이 일어남에 따라 오버헤드가 발생한다는 단점이 있다. 이러한 단점을 해결하기 위하여, FTL(1212) 구조는 DFTL 캐시 맵 매니저(430)를 이용하여 접근 성공률을 높이기 위한 캐시 교체 정책을 실행할 수 있다.

[0097] DFTL 캐시 맵 매니저(430)는 적응적 k-평균 클러스터링 알고리즘 모듈(420)에 의해 생성된 축약된 클러스터 맵핑 정보(C_MAP)를 이용하여, 맵핑되지 않은 물리 주소 스트림들에 할당되었던 휘발성 메모리(1260)의 데이터 버퍼 공간(730)의 크기만큼을 DFTL 캐시 맵 공간(710)으로 늘려주거나 줄여주는 기능을 수행할 수 있다.

[0098] DFTL 캐시 맵 매니저(430)는 축약된 클러스터 맵핑 테이블(C_MAP)에서 자주 참조되는 맵핑 테이블의 일부를 휘발성 메모리 장치(1260)의 DFTL 캐시 맵 공간(710)에 저장하고, 나머지 맵핑 테이블은 불휘발성 메모리 장치(NVM)에 저장하도록 구성될 수 있다.

[0099] DFTL 캐시 맵 매니저(430)는, 캐시 교체 정책을 이용하여 DFTL 캐시 맵 공간(710)의 캐시 라인에 유효한 상태

또는 캐시 히트(cache hit)임을 제공할 수 있다. 캐시 교체 정책으로는 LRU(Least Recently Used) 방식, LFU(Least Frequently Used) 방식, 랜덤(Random) 방식, 선입 선출(First In First Out: FIFO) 방식 등이 있다.

[0100] LRU 방식은 가장 덜 최근에 사용된 캐시 라인 또는 블록을 교체(또는 만료(expire))할 수 있다. 예를 들면, 매번 휘발성 메모리 장치(1260)에 대한 접근이 이루어질 때마다 유효한 캐시 라인에 대한 LRU 비트를 갱신할 수 있다. 최근에 접근된 순서를 나타내는 LRU 비트는 캐시 라인 교체가 일어날 때, 가장 덜 최근에 사용된 블록(또는 가장 오래 머문 블록)을 알 수 있는 정보로 사용될 수 있다. LFU 방식은 휘발성 메모리 장치(1260)에 저장된 후 가장 적게 사용된 블록을 교체할 수 있다. 랜덤 방식은 휘발성 메모리 장치(1260)의 임의의 블록을 선택하여 교체할 수 있다. FIFO 방식은 휘발성 메모리 장치(1260)에 저장된 지 가장 오래된 블록을 교체할 수 있다.

[0101] 도 7 및 도 8은 본 발명의 실시예들에 따른 FTL 구조에 따른 휘발성 메모리 장치(1260)의 데이터 맵핑을 설명하는 도면들이다.

[0102] 도 7을 참조하면, FTL(1212) 구조에서 제공되는 가상 논리 주소 스트림들(Lstream 0 ~ Lstream 4)과 휘발성 메모리 장치(1260) 사이의 데이터 맵핑을 보여준다. Lstream 3 및 Lstream 4 가상 논리 주소 스트림들 각각은 휘발성 메모리 장치(1260)의 DFTL 맵 캐시 공간(710)의 AD3 및 AD4 메모리 영역에 저장되어 있다. Lstream 0 내지 Lstream 2 가상 논리 주소 스트림들은 휘발성 메모리 장치(1260)의 스트림 맵핑 테이블 공간(720)을 통해 불휘발성 메모리 장치들(NVMs)의 물리 주소 스트림들(Pstreams)과 맵핑될 수 있다.

[0103] 도 7에서, 휘발성 메모리 장치(1260)의 스트림 맵핑 테이블 공간(720)에는 스트림 클러스터링 매니저(410, 도 4)에서 제공되는 축약된 클러스터 맵핑 테이블(C_MAP)이 저장되어 있다.

[0104] 도 8에 도시된 바와 같이, 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)은, k개의 물리 주소 스트림들에 할당된 데이터 버퍼들의 공간에서 축약된 클러스터 맵핑 테이블(C_MAP)에 의해 c개의 물리 주소 스트림들에 할당되는 데이터 버퍼들의 공간만 사용됨을 볼 수 있다. 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)에서 맵핑되지 않은 (k-c)개의 물리 주소 스트림들에 할당되었던 데이터 버퍼들의 공간은 DFTL 캐시 맵 매니저(430, 도 4)에 의해 DFTL 캐시 맵 공간(710)으로 확장될 수 있다. 이에 따라, FTL(1212) 구조는 높은 가용성 능력(high-availability capability)과 확장성(scalability)을 제공할 수 있다.

[0105] DFTL 캐시 맵 공간(710)이 확장됨에 따라, DFTL 캐시 맵 공간(710)에 로드될 수 있는 데이터가 증대될 수 있다. FTL(1212) 구조는 DFTL 캐시 맵 공간(710)에 로드된 데이터에 대해서는 불휘발성 메모리 장치들(NVMs)에 억세스하지 않아도 되기 때문에, 레이턴시를 줄일 수 있다. 이에 따라, SSD(1200, 도 1)는 빠른 응답 속도로 성능을 향상시킬 수 있다.

[0106] 도 9 내지 도 11은 본 발명의 실시예들에 따른 FTL(1212) 구조를 포함하는 데이터 저장 장치의 동작 방법을 설명하는 도면들이다.

[0107] 도 1 내지 도 8과 연계하여, 도 9를 참조하면, FTL(1212) 구조는 호스트의 n개 논리 주소 스트림들과 불휘발성 메모리 장치들의 k(k는 n 보다 작은 수)개 물리 주소 스트림들 사이의 맵핑 정보를 관리할 수 있다. SSD(1200)의 FTL(1212) 구조는 호스트(1100)로부터 수신되는 입출력 요청들에 대한 전체 논리 주소 공간을 식별할 수 있다(S910). FTL(1212) 구조는 전체 논리 주소 공간에 대한 축약된 클러스터 맵핑 테이블(C_MAP)을 생성할 수 있다(S920). FTL(1212) 구조가 축약된 클러스터 맵핑 테이블(C_MAP)을 생성하는 과정(S920)은, 도 10에서 구체적으로 설명된다.

[0108] 도 10을 참조하면, FTL(1212) 구조는 전체 논리 주소 공간을 n등분하여 n개의 가상 논리 주소 스트림들(Lstreams)로 설정할 수 있다(S921). FTL(1212) 구조는 n개의 가상 논리 주소 스트림들(Lstreams)에 대응하는 적어도 하나의 스트림 속성을 추적하여 획득할 수 있다(S922).

[0109] FTL(1212) 구조가 n개의 가상 논리 주소 스트림들(Lstreams)에 대응하는 적어도 하나의 스트림 속성을 추적하여 획득하는 과정(S922)는, 도 11에 도시된 바와 같이, 예상 수명을 추적하여 획득하는 단계(S1101), 빈도를 추적하여 획득하는 단계(S1102), 시간적 지역성을 추적하여 획득하는 단계(S1103), 순차성을 추적하여 획득하는 단계(S1104), 입출력(I/O) 크기를 추적하여 획득하는 단계(S1105), 입출력(I/O) 격리 레벨을 추적하여 획득하는 단계(S1106) 및/또는 QoS 요구 사항을 추적하여 획득하는 단계(S1107) 중 어느 하나에 대응할 수 있다.

[0110] 다시, 도 10으로 돌아가서, FTL(1212) 구조는 적어도 하나의 스트림 속성을 기반으로 n개의 가상 논리 주소 스트림들(Lstreams) 중 비슷한 스트림 속성을 갖는 가상 논리 주소 스트림들을 판별할 수 있다(S923). FTL(1212)

구조는 비슷한 스트림 속성을 갖는 가상 논리 주소 스트림들끼리 클러스터링하고(S924), k-평균 클러스터링 알고리즘을 이용하여 클러스터링된 n개의 가상 논리 주소 스트림들(Lstreams)을 k개의 물리 주소 스트림들(Pstreams)에 맵핑할 수 있다(S925). FTL(1212) 구조는 n개의 가상 논리 주소 스트림들과 k개의 물리 주소 스트림들 사이의 맵핑 정보를 나타내는 예비 클러스터 맵핑 테이블(P_MAP)을 생성할 수 있다(S926).

[0111] FTL(1212) 구조는 예비 클러스터 맵핑 테이블(P_MAP)에 대하여 적응적 k-평균 클러스터링 알고리즘을 수행하고, n개의 가상 논리 주소 스트림들이 c(c는 k 보다 작은 수)개의 물리 주소 스트림들로 변환되도록 하는 축약된 클러스터 맵핑 테이블(C_MAP)을 생성할 수 있다(S927).

[0112] 다시, 도 9로 돌아가서, FTL(1212) 구조는 축약된 클러스터 맵핑 테이블(C_MAP)에 기초하여, 휘발성 메모리 장치(1260)의 데이터 버퍼 공간(730)에서 맵핑되지 않은 물리 주소 스트림들에 할당되었던 데이터 버퍼들의 공간을 DFTL 캐시 맵 공간(710)으로 확장할 수 있다(S930).

[0113] 도 12a 내지 도 12c 그리고 도 13은 본 발명의 실시예들에 따른 FTL 구조를 다양한 워크로드들에 수행한 성능 평가들(evaluations)을 보여주는 도면들이다. 도 12a 내지 도 12c에서, X축은 타임스템프를 나타내고, 왼쪽 Y축은 축약된 클러스터 맵핑 테이블(C_MAP)에 의해 사용되는 물리 주소 스트림들의 개수를 나타내고, 오른쪽 Y축은 축약된 클러스터 맵핑 테이블(C_MAP) 및 DFTL 캐시 맵 매니저(430)에 의한 DFTL 맵 캐시 공간(710)의 크기를 나타낸다. 도 12a 내지 도 12c 그리고 도 13은 FTL 맵핑 공간(720)의 최대 사용 가능한 공간 크기를 4MB 로 제한 상태로 시뮬레이션 한 결과를 보여준다.

[0114] 도 12a는 Virtual 워크로드의 타임스탬프에 따른 축약된 클러스터 맵핑 테이블(C_MAP)에 의해 변화되는 물리 주소 스트림들의 개수와 크기를 보여준다. Virtual 워크로드의 경우, 타임스탬프가 증가함에 따라 초기 16개의 물리 주소 스트림들에서 시작된 물리 주소 스트림들의 개수가 2~10개의 물리 주소 스트림들로 상대적으로 크게 변동하는 갖는 것을 관찰할 수 있다. 이에 따라, Virtual 워크로드의 경우, 축약된 클러스터 맵핑 테이블(C_MAP)에 의해 차지하는 DFTL 맵 캐시 공간(710)의 크기도 타임스탬프가 증가함에 따라 2.5~3.5 MB 정도의 큰 변동률을 갖는 것을 관찰할 수 있다.

[0115] 도 12b는 FIU 워크로드의 타임스탬프에 따른 축약된 클러스터 맵핑 테이블(C_MAP)에 의해 변화되는 물리 주소 스트림들의 개수와 크기를 보여준다. FIU 워크로드의 경우, 타임스탬프가 증가함에 따라 초기 16개의 물리 주소 스트림들에서 시작된 물리 주소 스트림들의 개수가 4~8개의 물리 주소 스트림들로 상대적으로 작게 변동하는 갖는 것을 관찰할 수 있다. 이에 따라, FIU 워크로드의 경우, 축약된 클러스터 맵핑 테이블(C_MAP)에 의해 차지하는 DFTL 맵 캐시 공간(710)의 크기는 타임스탬프가 증가해도 3MB 정도인 것을 관찰할 수 있다.

[0116] 도 12c는 MSR 워크로드의 타임스탬프에 따른 축약된 클러스터 맵핑 테이블(C_MAP)에 의해 변화되는 물리 주소 스트림들의 개수와 크기를 보여준다. MSR 워크로드의 경우, 타임스탬프가 증가함에 따라 초기 16개의 물리 주소 스트림들에서 시작된 물리 주소 스트림들의 개수가 3~9개의 물리 주소 스트림들로 변동하는 것을 관찰할 수 있다. 이에 따라, MSR 워크로드의 경우, 축약된 클러스터 맵핑 테이블(C_MAP)에 의해 차지하는 DFTL 맵 캐시 공간(710)의 크기는 타임스탬프가 증가함에 따라 2.5~3MB 정도인 것을 관찰할 수 있다.

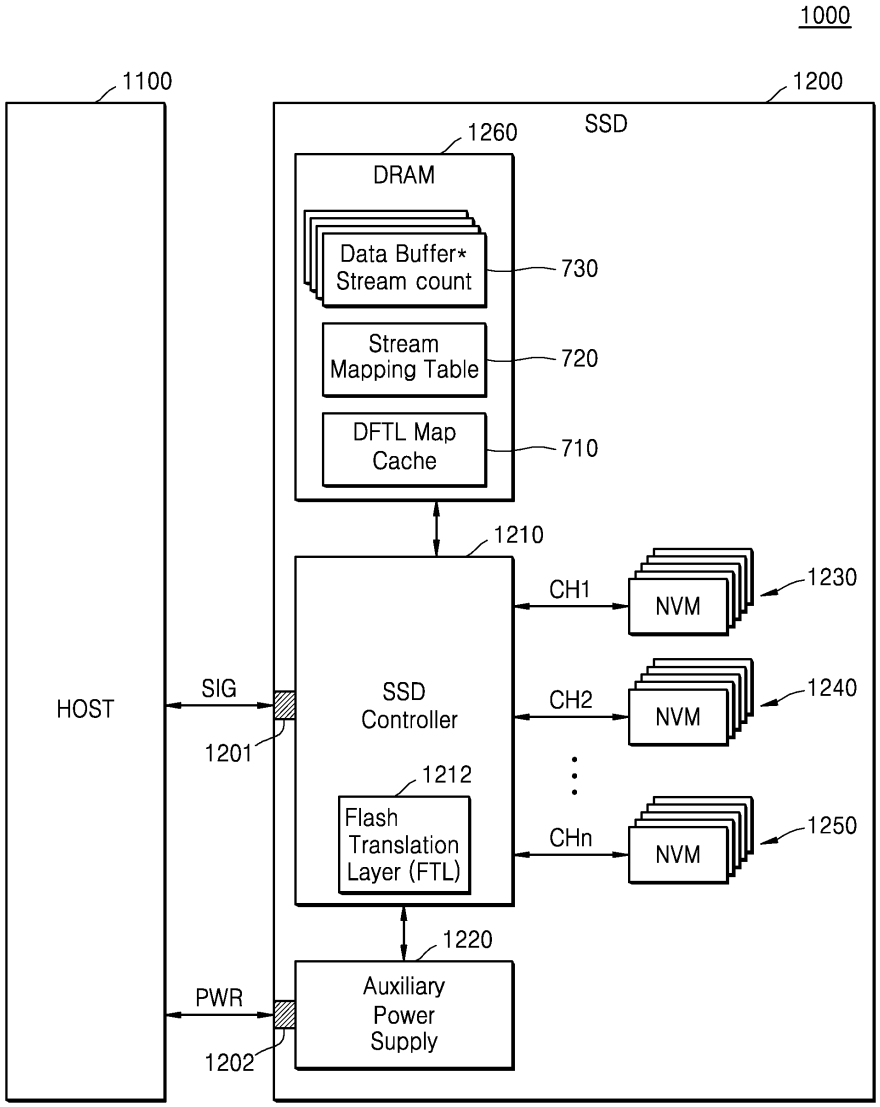
[0117] 도 12a 내지 도 12c에 도시된 바와 같이, FTL(1212) 구조는 Virtual 워크로드에 비해 FIU 및 MSR 워크로드들에서 상대적으로 작은 수의 물리주소 스트림들을 포함하고, 상대적으로 작은 DFTL 맵 캐시 공간(710)을 차지하는 것을 알 수 있다.

[0118] 도 13 에서는, 도면의 간소화를 위하여, FTL(1212, 도 4) 구조에 의해 생성되는 예비 클러스터 맵핑 테이블(P_MAP)은 16streams로 표시하고, 적응적 k-평균 클러스터링 알고리즘 모듈(420)에 의해 생성되는 축약된 클러스터 맵핑 테이블(C_MAP)에 따라 적응적으로 스트림들을 조절해서 수행한 결과를 Dstream(Dynimic stream)으로 표시한다. 도 13을 참조하면, 16streams과 Dstream을 다양한 워크로드들(Virtual, FIU, MSR)에서 수행했을 때, DFTL 맵 캐시 공간(710)의 캐시 미스율을 보여준다. 캐시 미스율은 Virtual, FIU, MSR 워크로드 각각 16streams 대비 상당히 줄어드는 것을 관찰할 수 있다. 휘발성 메모리 장치(1260)에서 Dstream에 의해 적응적으로 줄어든 데이터 버퍼 공간(730)이 DFTL 맵 캐시 공간(710)으로 확장됨에 따라, 캐시 미스율이 줄어든 것으로 이해할 수 있다.

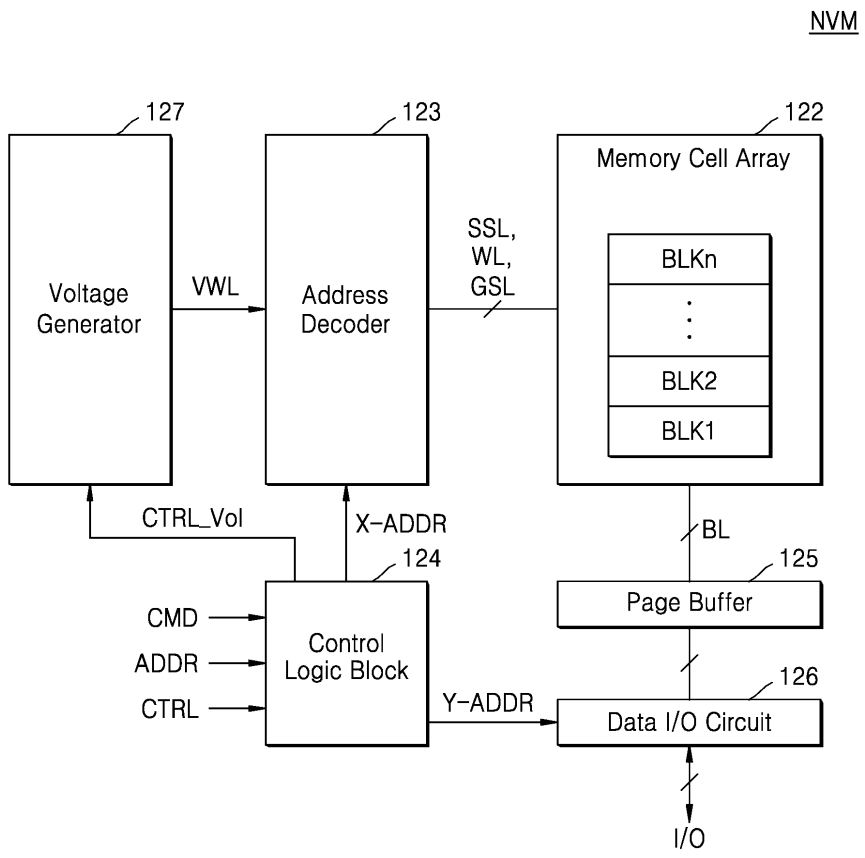
[0119] 본 개시는 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 다른 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 개시의 진정한 기술적 보호 범위는 첨부된 특허청구범위의 기술적 사상에 의하여 정해져야 할 것이다.

도면

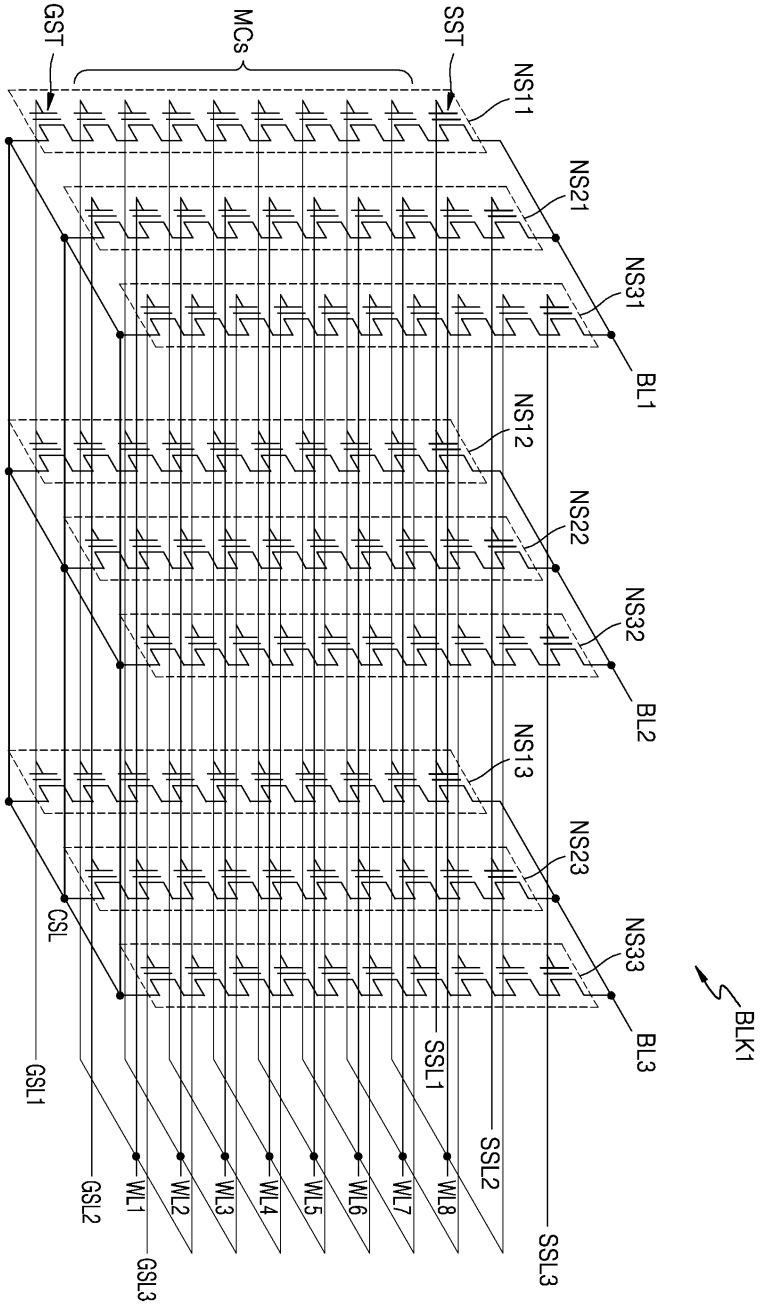
도면1


1000
1100
1200
SSD
1260
DRAM
Data Buffer*
Stream count
730
Stream
Mapping Table
720
DFTL Map
Cache
710
1210
CH1
NVM
1230
HOST
SIG
SSD
Controller
CH2
NVM
1240
1201
1212
Flash
Translation
Layer (FTL)
CHn
NVM
1250
1220
PWR
Auxiliary
Power
Supply
1202

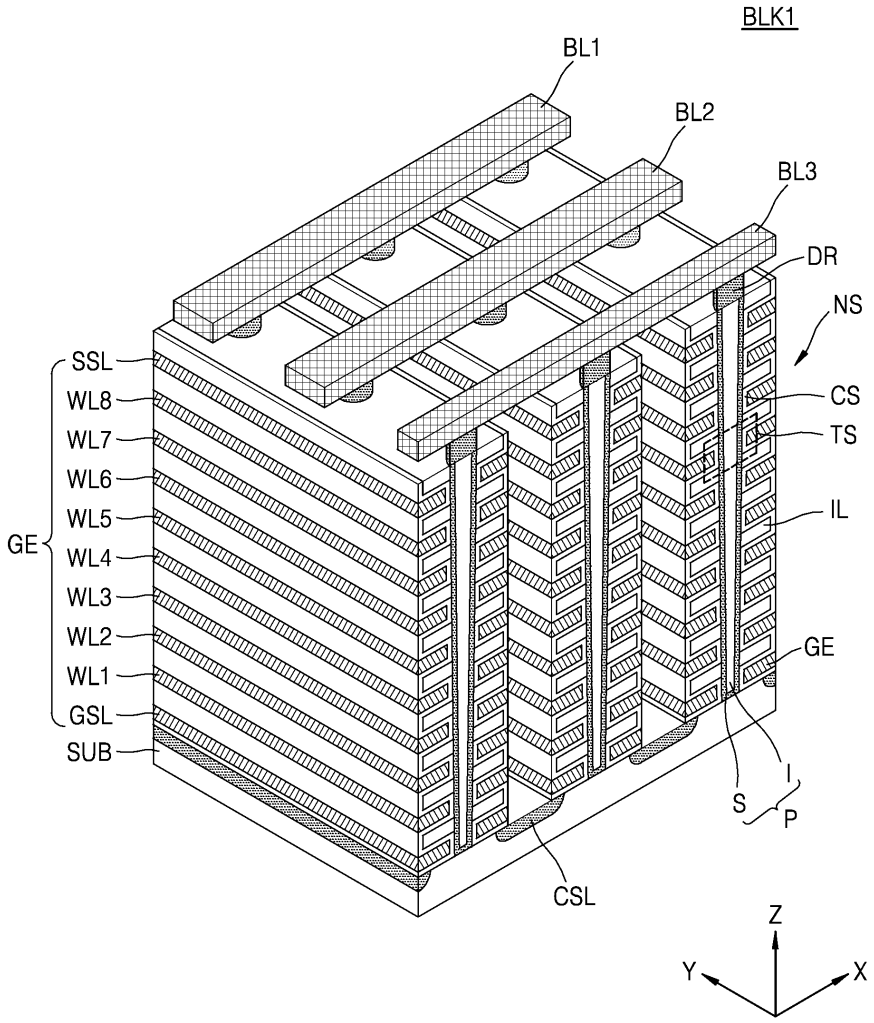
도면2


NVM
127
123
122
Memory Cell Array
Voltage Generator
VWL
Address Decoder
SSL, WL, GSL
BLKn
BLK2
BLK1
CTRL_Vol
X-ADDR
BL
124
125
Page Buffer
CMD
ADDR
CTRL
Control Logic Block
Y-ADDR
126
Data I/O Circuit
I/O

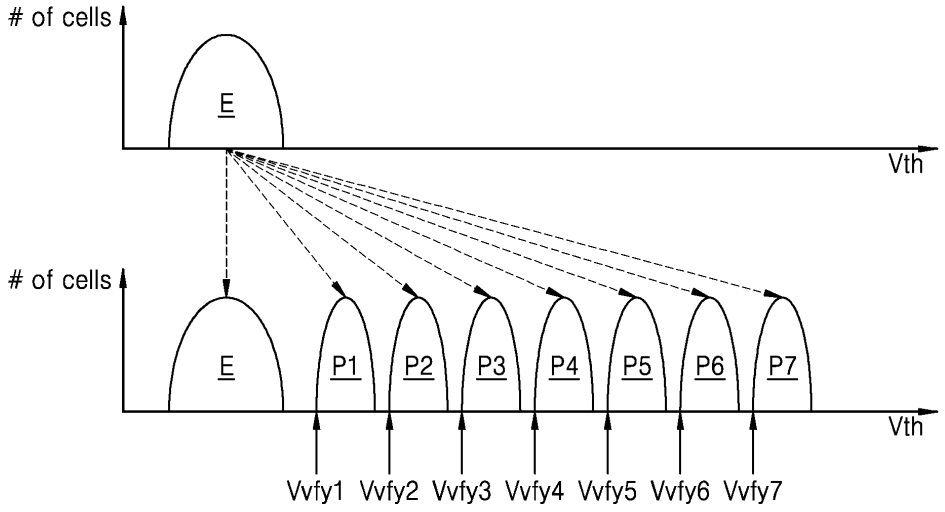
도면3a


BLK1
BL1
BL2
BL3
NS11
NS21
NS31
NS12
NS22
NS32
NS13
NS23
NS33
SSL1
SSL2
SSL3
WL1
WL2
WL3
WL4
WL5
WL6
WL7
WL8
GSL1
GSL2
GSL3
CSL
GST
MCs
SST

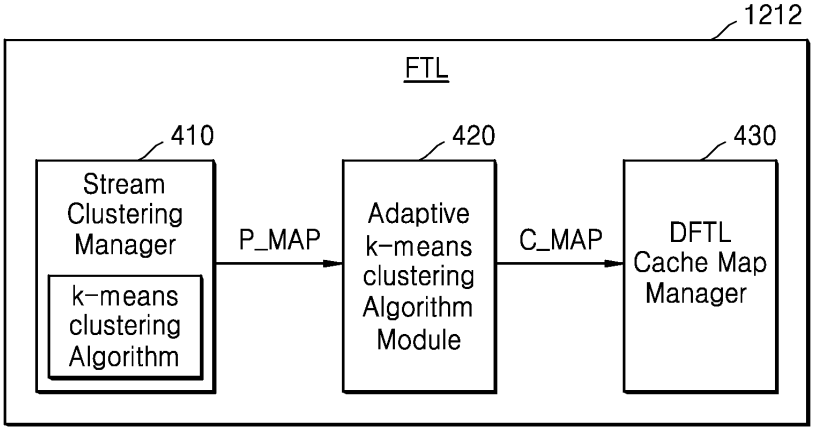
도면3b


BLK1
BL1
BL2
BL3
DR
NS
CS
TS
IL
GE
I
S
P
CSL
SSL
WL8
WL7
WL6
WL5
WL4
WL3
WL2
WL1
GSL
SUB
GE
Z
Y
X

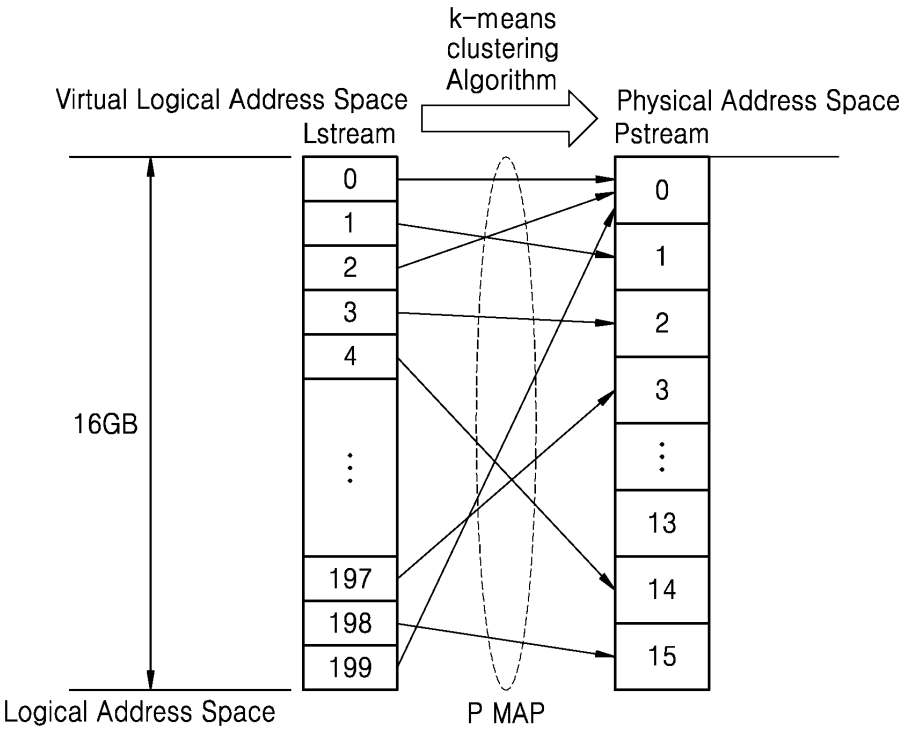
도면3c


# of cells
E
Vth
# of cells
E
P1
P2
P3
P4
P5
P6
P7
Vth
Vvfy1
Vvfy2
Vvfy3
Vvfy4
Vvfy5
Vvfy6
Vvfy7

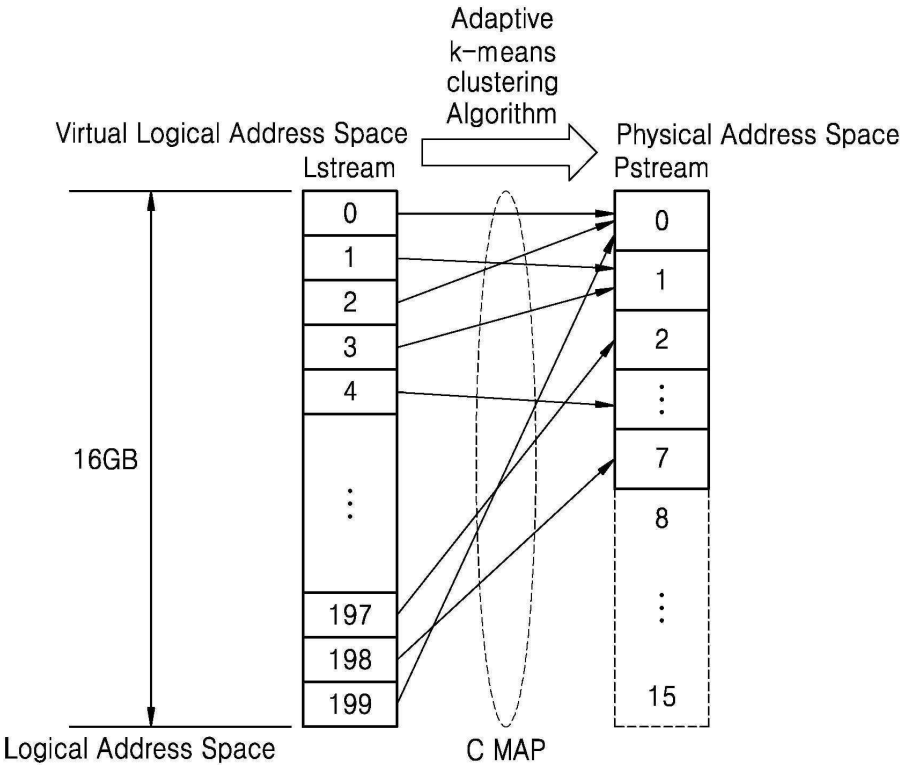
도면4


1212
FTL
410
Stream Clustering Manager
k-means clustering Algorithm
P_MAP
420
Adaptive k-means clustering Algorithm Module
C_MAP
430
DFTL Cache Map Manager

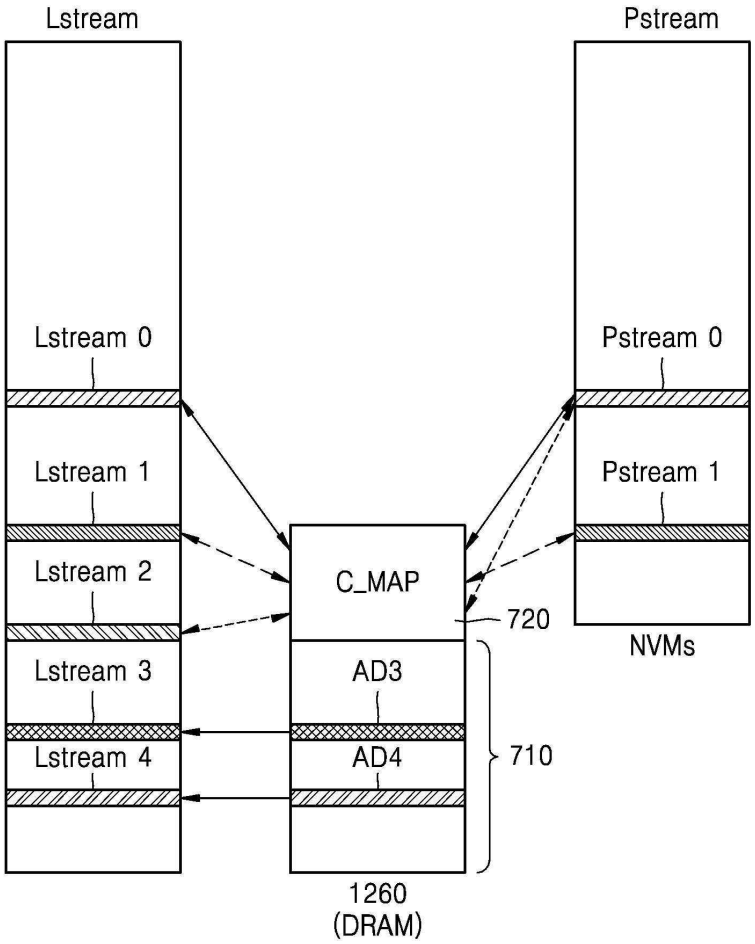
도면5


k-means clustering Algorithm
Virtual Logical Address Space
Lstream
Physical Address Space
Pstream
16GB
0
1
2
3
4
197
198
199
0
1
2
3
13
14
15
Logical Address Space
P MAP

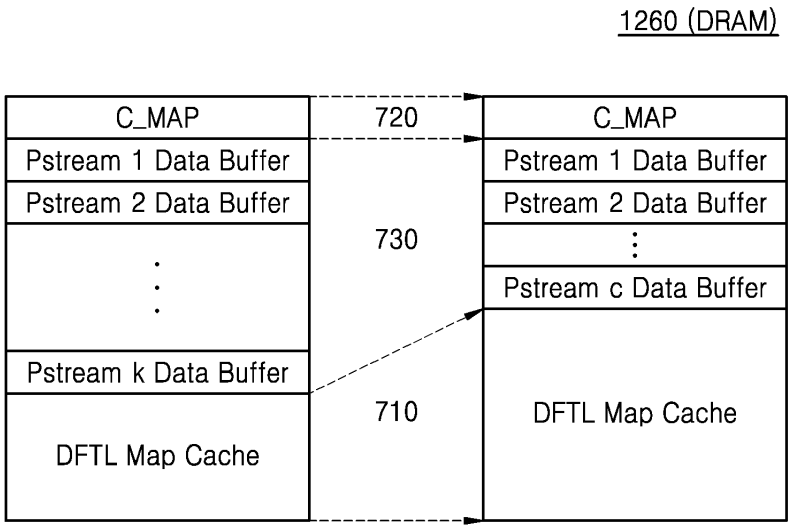
도면6


Adaptive
k-means
clustering
Algorithm
Virtual Logical Address Space
Lstream
Physical Address Space
Pstream
16GB
0
1
2
3
4
197
198
199
0
1
2
7
8
15
Logical Address Space
C MAP

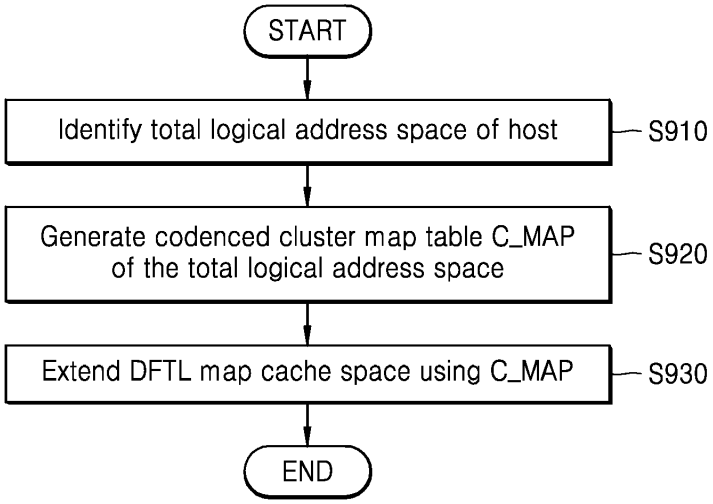
도면7


Lstream
Pstream
Lstream 0
Lstream 1
Lstream 2
Lstream 3
Lstream 4
Pstream 0
Pstream 1
C_MAP
720
NVMs
AD3
AD4
710
1260
(DRAM)

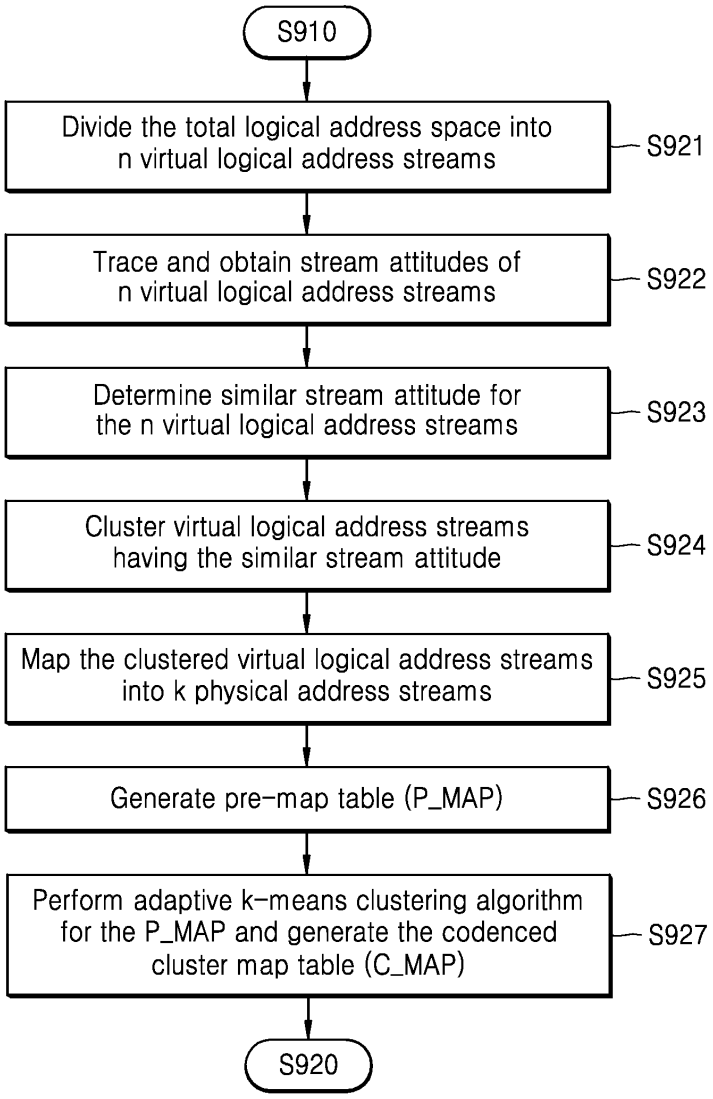
도면8


1260 (DRAM)
C_MAP
Pstream 1 Data Buffer
Pstream 2 Data Buffer
Pstream k Data Buffer
DFTL Map Cache
720
730
710
C_MAP
Pstream 1 Data Buffer
Pstream 2 Data Buffer
Pstream c Data Buffer
DFTL Map Cache

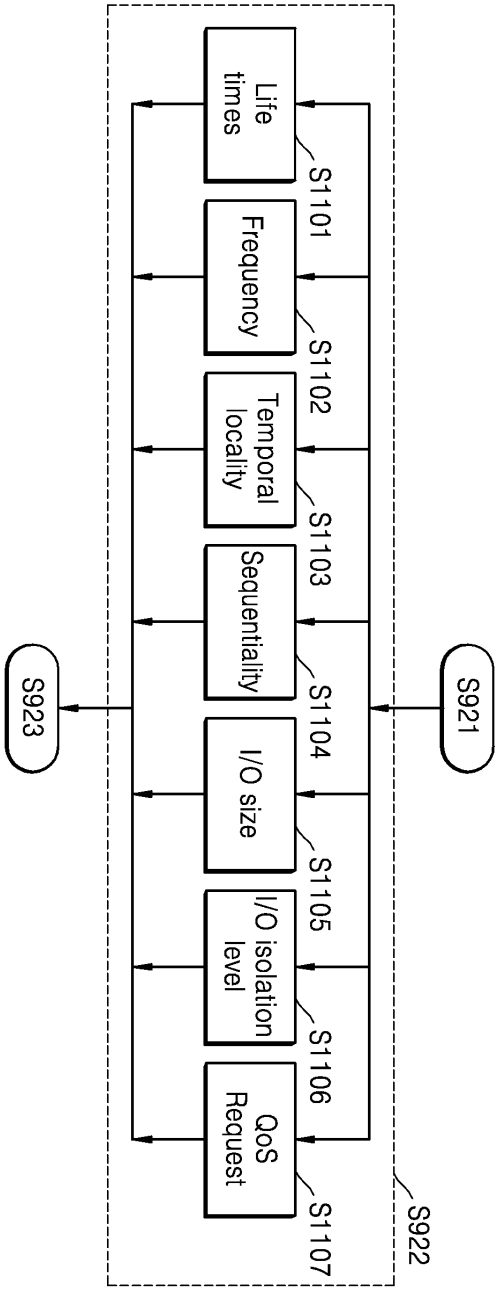
도면9


START
Identify total logical address space of host
S910
Generate codenced cluster map table C_MAP of the total logical address space
S920
Extend DFTL map cache space using C_MAP
S930
END

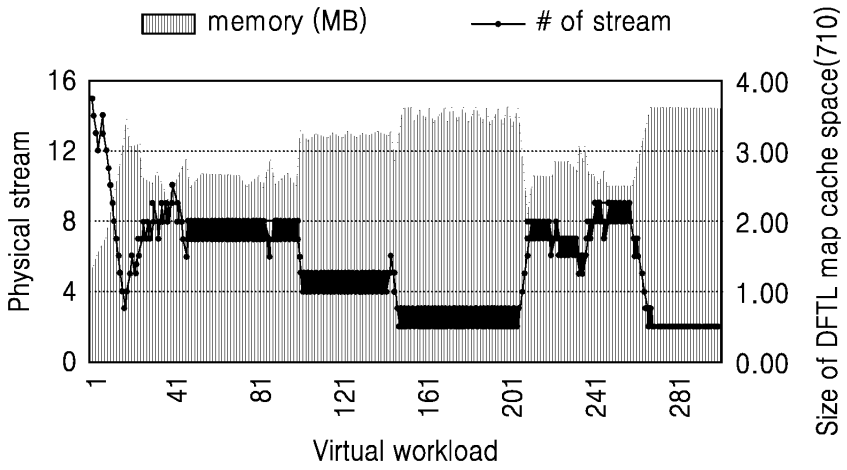
도면10


S910
Divide the total logical address space into n virtual logical address streams
S921
Trace and obtain stream attitudes of n virtual logical address streams
S922
Determine similar stream attitude for the n virtual logical address streams
S923
Cluster virtual logical address streams having the similar stream attitude
S924
Map the clustered virtual logical address streams into k physical address streams
S925
Generate pre-map table (P_MAP)
S926
Perform adaptive k-means clustering algorithm for the P_MAP and generate the codenced cluster map table (C_MAP)
S927
S920

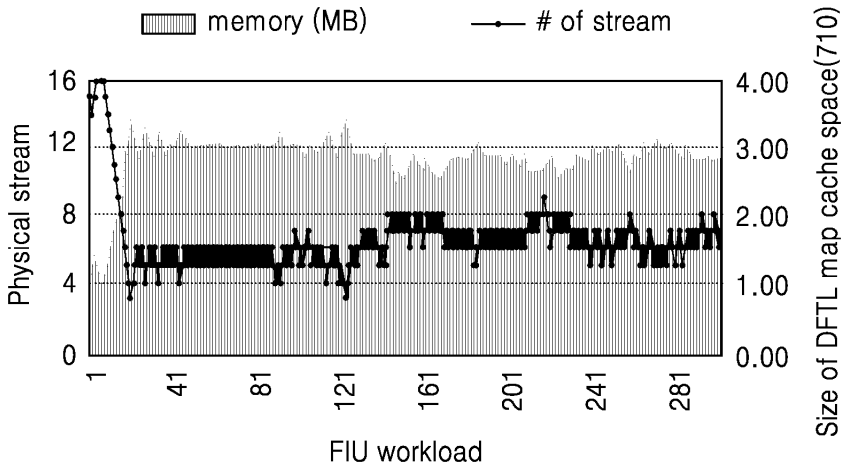
도면11


S921
S922
S923
S1101
Life times
S1102
Frequency
S1103
Temporal locality
S1104
Sequentiality
S1105
I/O size
S1106
I/O isolation level
S1107
QoS Request

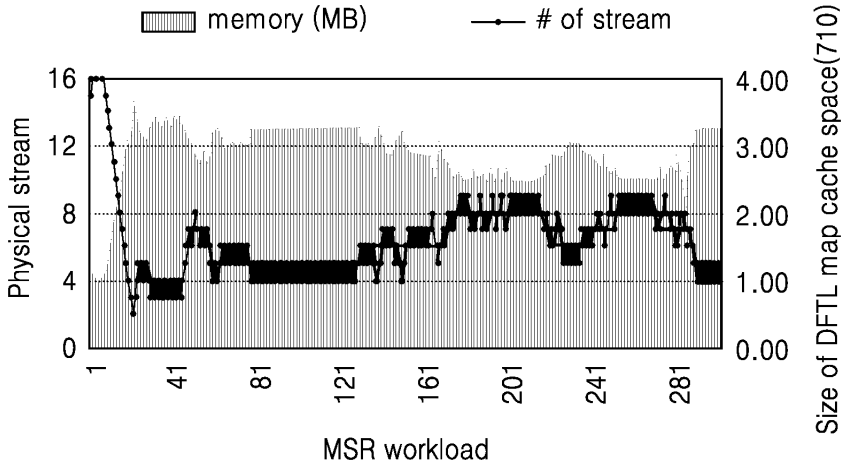
도면12a



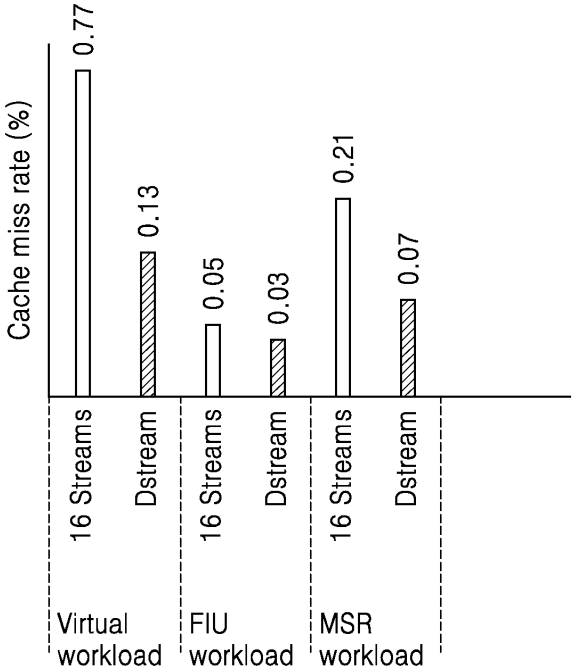
도면12b



도면12c



도면13


Cache miss rate (%)
0.77
0.13
0.05
0.03
0.21
0.07
16 Streams
Dstream
16 Streams
Dstream
16 Streams
Dstream
Virtual workload
FIU workload
MSR workload