



(19) 대한민국특허청(KR)
 (12) 공개특허공보(A)

(11) 공개번호 10-2015-0115924
 (43) 공개일자 2015년10월14일

- (51) 국제특허분류(Int. Cl.)
G06F 12/02 (2006.01) *G06F 3/06* (2006.01)
- (52) CPC특허분류
G06F 12/0246 (2013.01)
G06F 3/061 (2013.01)
- (21) 출원번호 10-2015-7024222
- (22) 출원일자(국제) 2014년02월06일
 심사청구일자 없음
- (85) 번역문제출일자 2015년09월04일
- (86) 국제출원번호 PCT/US2014/014971
- (87) 국제공개번호 WO 2014/124064
 국제공개일자 2014년08월14일
- (30) 우선권주장
 13/763,491 2013년02월08일 미국(US)
- (71) 출원인
마이크로소프트 테크놀로지 라이센싱, 엘엘씨
 미국 위성던주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이
- (72) 발명자
카라모브 세르게이
 미국 위성던주 98052-6399 레드몬드 원 마이크로
 소프트 웨이 엘씨에이-인터넷내셔널 패턴즈 마이크
 로소프트 코포레이션 내
칼라간 데이비드 마이클
 미국 위성던주 98052-6399 레드몬드 원 마이크로
 소프트 웨이 엘씨에이-인터넷내셔널 패턴즈 마이크
 로소프트 코포레이션 내
- (74) 대리인
김태홍, 김진희

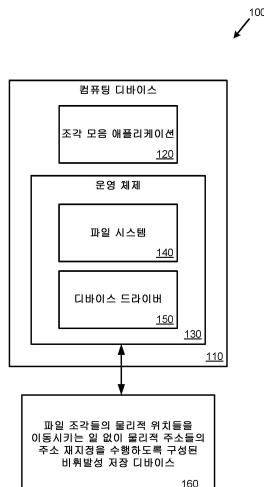
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **비휘발성 저장 디바이스를 위한 메모리의 주소 재지정**

(57) 요 약

비휘발성 저장 디바이스 상의 조각난 파일의 메모리가 연속된 물리적 메모리 주소들로 주소 재지정될 수 있는 반면, 비휘발성 저장 디바이스 상에 저장된 조각난 파일의 파일 조각들의 물리적 위치는 메모리가 주소 재지정된 후에 동일하게 유지된다. 논리 블록 주소 지정(LBA) 맵핑 테이블은 주소 재지정된 연속된 물리적 메모리 주소들에 기초하여 업데이트될 수 있다.

대 표 도 - 도1



(52) CPC특허분류

G06F 3/0616 (2013.01)

G06F 3/064 (2013.01)

G06F 3/0679 (2013.01)

G06F 3/0688 (2013.01)

명세서

청구범위

청구항 1

적어도 일부가 컴퓨팅 디바이스에 의해 수행되는, 비휘발성 저장 디바이스 상의 조각난 파일(fragmented file)에 대해 메모리의 주소 재지정(readdressing)을 수행하는 방법에 있어서,

상기 컴퓨팅 디바이스에 의해, 상기 조각난 파일 - 상기 조각난 파일의 파일 조각(file fragment)들은 복수의 비연속된 물리적 주소(noncontiguous physical address)들에 걸쳐 분산되어 있고 상기 비휘발성 저장 디바이스 내의 복수의 물리적 위치들에 저장되어 있음 - 의 상기 메모리를 주소 재지정하라는 명령(command)을 상기 비휘발성 저장 디바이스로 송신하는 단계; 및

상기 컴퓨팅 디바이스에 의해, 상기 조각난 파일의 상기 메모리가 주소 재지정되었다 - 상기 메모리는 연속된 물리적 주소들로 주소 재지정됨 - 는 응답을 상기 비휘발성 저장 디바이스로부터 수신하는 단계

를 포함하고,

상기 파일 조각들의 상기 복수의 물리적 위치들은 상기 조각난 파일의 상기 메모리가 주소 재지정된 후에 동일하게 유지되는 것인, 메모리의 주소 재지정을 수행하는 방법.

청구항 2

비휘발성 저장 디바이스에 있어서,

처리 유닛; 및

비휘발성 메모리

를 포함하고;

상기 비휘발성 저장 디바이스는 조각난 파일에 대해 메모리를 주소 재지정하기 위한 동작들을 수행하도록 구성되며, 상기 동작들은,

상기 조각난 파일 - 상기 조각난 파일의 파일 조각들은 복수의 비연속된 물리적 주소들에 걸쳐 분산되어 있고 상기 비휘발성 저장 디바이스 내의 복수의 물리적 위치들에 저장되어 있음 - 의 상기 메모리를 주소 재지정하라는 명령을 수신하는 것; 및

상기 파일 조각들 각각에 대해, 연속된 물리적 메모리 주소를 상기 파일 조각에 할당하는 것을 포함하고;

상기 파일 조각들의 상기 복수의 물리적 위치들은 상기 조각난 파일의 상기 메모리가 주소 재지정된 후에 동일하게 유지되는 것인, 비휘발성 저장 디바이스.

청구항 3

제2항에 있어서, 상기 비휘발성 저장 디바이스는 솔리드 스테이트 드라이브(SSD, solid state drive)인 것인, 비휘발성 저장 디바이스.

청구항 4

제2항에 있어서, 상기 비휘발성 저장 디바이스는 상변화 메모리 디바이스(phase change memory device)인 것인, 비휘발성 저장 디바이스.

청구항 5

제2항에 있어서, 상기 동작들은, 상기 파일 조각들 각각에 대해, 상기 파일 조각이 할당되어야 하는 물리적 메모리 주소에 다른 데이터가 위치되어 있는 경우, 상기 다른 데이터에 새로운 메모리 주소를 할당하는 것을 더 포함하는 것인, 비휘발성 저장 디바이스.

청구항 6

제5항에 있어서, 상기 다른 데이터에 상기 새로운 메모리 주소를 할당하는 것은 상기 다른 데이터의 상기 물리적 메모리 주소를 상기 파일 조각의 메모리 주소와 스와핑(swapping)하는 것을 포함하는 것인, 비휘발성 저장 디바이스.

청구항 7

제5항에 있어서, 상기 다른 데이터에 상기 새로운 메모리 주소를 할당하는 것은 상기 다른 데이터에 사용되지 않은 메모리 주소를 할당하는 것을 포함하는 것인, 비휘발성 저장 디바이스.

청구항 8

제2항에 있어서, 상기 연속된 물리적 메모리 주소를 상기 파일 조각에 할당하는 것은,
상기 조각난 파일의 시작 블록에 대해, 고유의 물리적 메모리 주소를 할당하는 것, 및
상기 조각난 파일의 하나 이상의 후속 블록들에 대해, 공유가능 물리적 메모리 주소들을 할당하는 것을 포함하는 것인, 비휘발성 저장 디바이스.

청구항 9

제2항에 있어서, 상기 조각난 파일의 상기 메모리를 주소 재지정하라는 명령을 수신하는 것은,
조각화 정도에 기초하여 주소 재지정할 가능성이 가장 많은 후보 파일을 결정하는 것, 및
상기 가능성이 가장 많은 후보 파일을 상기 조각난 파일로서 선택하는 것을 포함하는 것인, 비휘발성 저장 디바이스.

청구항 10

컴퓨팅 디바이스로 하여금, 조각난 파일에 대해 메모리를 주소 재지정하기 위한 동작들을 수행하게 하는 컴퓨터 실행가능 명령어들을 저장하는 컴퓨터 판독가능 저장 매체에 있어서,

상기 동작들은,

상기 조각난 파일 - 상기 메모리는 연속된 물리적 주소들로 주소 재지정되어 있음 - 의 상기 메모리가 주소 재지정되었다는 응답을 비휘발성 저장 디바이스로부터 수신하는 것; 및

상기 주소 재지정된 연속된 물리적 주소들에 기초하여 가상 매핑 테이블(virtual mapping table)을 업데이트하는 것을 포함하고,

상기 조각난 파일의 파일 조각들의 물리적 위치들은 상기 조각난 파일의 상기 메모리가 주소 재지정된 후에 동일하게 유지되며,

운영 체제에 대한 논리 블록 주소 지정(LBA, logical block addressing) 매핑 테이블은 상기 주소 재지정된 연속된 물리적 주소들에 기초하여 업데이트되지 않고,

상기 LBA 매핑 테이블은 상기 가상 매핑 테이블과 통신하는 것인, 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 발명은 비휘발성 저장 디바이스를 위한 메모리의 주소 재지정에 관한 것이다.

배경 기술

[0002] 파일들이 저장 디바이스 상에서 반복적으로 쓰기 및 소거가 됨에 따라, 이들 파일은, 시간의 경과에 따라, 조각화되어, 저장 디바이스의 성능을 떨어뜨릴 수 있다. 이 성능 문제를 완화시키는 데 도움을 주기 위해, 저장 디바이스에 대해 디스크 조각 모음(disk defragmentation)이 수행될 수 있다. 디스크 조각 모음은 저장 디바이스

상에서의 파일 조각들을 연속된 위치들로 이동시키는 것에 의해 저장 디바이스 상에서의 파일들의 조각화를 감소시키는 동작을 지칭하며, 그로써 파일 조각들 모두를 읽기 또는 쓰기하기 위해 필요한, 저장 디바이스와 중앙 처리 장치(CPU, central processing unit) 메모리 사이의 입출력(I/O) 트랜잭션들의 횟수를 감소시킨다.

[0003] 솔리드 스테이트 드라이브(SSD, solid state drive)와 같은 비휘발성 저장 디바이스들이, 회전하는 자기 및 광 드라이브와 같은, 전통적인 하드 디스크 드라이브 대신에 또는 그에 부가하여, 저장 디바이스로서 점점 더 사용되고 있다. 전통적인 하드 디스크 드라이브에서는 조각 모음이 효과적으로 사용될 수 있지만, 비휘발성 저장 디바이스에 대해 조각 모음을 사용하는 것은 문제가 될 수 있는데, 그 이유는 이러한 비휘발성 저장 디바이스가 디바이스에 대한 반복된 소거 동작으로 인해 마모를 겪을 수 있기 때문이다. 비휘발성 저장 디바이스가, 그의 신뢰성이 손상되기 전에, 소거 및 쓰기가 될 있는 횟수가 제한되어 있기 때문에, 비휘발성 저장 디바이스의 디스크 조각 모음은 디스크 성능 대 저장 디바이스의 수명 간의 트레이드오프를 겪는다.

발명의 내용

[0004] 이 발명의 내용은 이하에서 발명을 실시하기 위한 구체적인 내용에 추가로 기술되는 일련의 개념들을 간략화된 형태로 소개하기 위해 제공된다. 이 발명의 내용은 청구된 발명 요지의 주요 특징들 또는 필수적인 특징들을 언급하기 위한 것이 아니며, 청구된 발명 요지의 범주를 제한하기 위해 사용되기 위한 것도 아니다.

[0005] 비휘발성 저장 디바이스에서 메모리 주소를 재배열하는 기법들 및 도구들이 기술되어 있다. 예를 들어, 데이터를 저장 디바이스 상의 그의 물리적 위치로부터 이동시키는 일 없이 메모리 주소가 주소 재지정(readdress)될 수 있다. 저장 디바이스는 운영 체제에 투명한 방식으로 메모리 주소를 주소 재지정할 수 있다. 대안적으로, 운영 체제는 최적화를 수행하여 최적화된 저장 디바이스에 대한, 예컨대, 매펑 테이블을 수정하라는 명령을 저장 디바이스에 대해 발행할 수 있다.

[0006] 예를 들어, 비휘발성 저장 디바이스 상의 조각난 파일에 대해 메모리의 주소 재지정을 수행하는 방법이 제공될 수 있다. 본 방법은 조각난 파일의 메모리를 주소 재지정하라는 명령을 비휘발성 저장 디바이스로 송신하는 단계 - 조각난 파일의 파일 조각들은 복수의 비연속된 물리적 주소(noncontiguous physical address)들에 걸쳐 분산되어 있음 -, 및 조각난 파일의 메모리가 연속된 물리적 주소들로 주소 재지정되었다는 응답을 비휘발성 저장 디바이스로부터 수신하는 단계를 포함한다. 파일 조각들의 물리적 위치는 메모리가 주소 재지정된 후에 동일하게 유지된다.

[0007] 다른 예로서, 비휘발성 저장 디바이스는 본 명세서에 기술된 동작들을 수행하도록 구성될 수 있다. 예를 들어, 비휘발성 저장 디바이스는 조각난 파일의 메모리를 주소 재지정하라는 명령을 수신하고, 조각난 파일의 파일 조각들 각각에 대해, 연속된 물리적 메모리 주소를 파일 조각에 할당할 수 있다. 파일 조각들의 물리적 위치는 메모리가 주소 재지정된 후에 동일하게 유지된다.

[0008] 또 다른 예로서, 시스템으로 하여금 본 명세서에 기술된 동작들을 수행하게 하는 컴퓨터 실행가능 명령어들을 저장하는 컴퓨터 판독가능 저장 매체가 제공될 수 있다. 예를 들어, 컴퓨터 판독가능 저장 매체는 조각난 파일의 메모리가 연속된 물리적 주소들로 주소 재지정되었다는 응답을 비휘발성 저장 디바이스로부터 수신할 수 있고, 주소 재지정된 연속된 물리적 주소들에 기초하여 가상 매펑 테이블(virtual mapping table)을 업데이트할 수 있다. 파일 조각들의 물리적 위치는 메모리가 주소 재지정된 후에 동일하게 유지된다. 운영 체제에 대한 논리 블록 주소 지정(LBA, logical block addressing) 매펑 테이블은 주소 재지정된 물리적 주소들에 기초하여 업데이트되지 않고, LBA 매펑 테이블이 가상 매펑 테이블과 통신한다.

[0009] 본 명세서에 기술되는 바와 같이, 각종의 다른 특징들 및 장점들이 원하는 바에 따라 본 기술들에 포함될 수 있다.

도면의 간단한 설명

[0010] 도 1은 예시적인 운영 환경의 블록도.

도 2는 메모리의 주소 재지정을 수행하는 예시적인 방법의 플로우차트.

도 3은 메모리의 주소 재지정을 수행하는 예시적인 방법의 플로우차트.

도 4a, 도 4b, 및 도 4c는 물리적 주소들을 주소 재지정하는 예들을 나타낸 도면.

도 5a 및 도 5b는 메모리의 물리적 위치들을 이동시키지 않으면서 물리적 주소들을 주소 재지정하는 일례를 나

타낸 도면.

도 6a 및 도 6b는 LBA 매핑 테이블과 물리적 주소들 간의 매핑의 일례를 나타낸 테이블.

도 7은 일부 기술된 실시예들이 구현될 수 있는 예시적인 컴퓨팅 시스템을 나타낸 도면.

도 8은 본 명세서에서 설명되는 기술들과 관련하여 사용될 수 있는 예시적인 모바일 디바이스를 나타낸 도면.

발명을 실시하기 위한 구체적인 내용

예 1 - 예시적인 개요

[0011] 이하의 설명은 비휘발성 저장 디바이스 상에서 물리적 메모리 주소들을 주소 재지정하는 기법들 및 해결책들에 관한 것이다. 예를 들어, 조각난 파일의 메모리의 물리적 주소들이, 메모리를 저장 디바이스 상에서 그의 물리적 메모리 위치들로부터 이동시키는 일 없이, 주소 재지정될 수 있다.

[0012] 메모리 주소들을 주소 재지정하는 것에 의해, 조각난 파일의 파일 조각들이 연속된 메모리 주소들로 주소 재지정될 수 있어, 보다 효율적인 파일 작업들(예컨대, 파일의 검색)을 가능하게 한다. 예를 들어, 파일의 파일 조각들이 연속된 메모리 주소들에 위치되는 경우, 운영 체제는 파일을 검색하기 위해 비휘발성 저장 디바이스에 단일의 요청을 하거나 다수의 요청들을 패킹(pack)할 수 있을 것이다. 다른 한편으로, 파일이 비연속된 메모리 주소들에 위치되어 있는 경우, 운영 체제는 파일을 검색하기 위해 저장 디바이스에 다수의 요청을 해야만 할지도 모른다.

[0013] 비휘발성 저장 디바이스의 디스크 조각 모음이 어쩌면 유사한 효과를 달성할 것이다. 저장 디바이스를 조각 모음하는 것에 의해, 파일 조각들이 조각 모음 후에 연속된 물리적 메모리 위치들에 위치되도록, 파일 조각들이 저장 디바이스 상의 실제의 물리적 메모리 위치들 간에 이동될 것이다. 그렇지만, 조각 모음이 SSD와 같은 비휘발성 저장 디바이스의 유효 수명을 단축시킬 수 있는데, 그 이유는 각각의 조각 모음 동작이 파일 조각들을 저장 디바이스에서 여기저기로 이동시키기 위해 다수의 소거 및 쓰기 동작들을 필요로 하여, 저장 디바이스에 대한 마모(wear)를 증가시킬 것이기 때문이다. 부가의 마모의 문제점은 업계에서 쓰기 증폭(write amplification)이라고 알려져 있는 현상으로 인해 데이터를 단지 소거하고 쓰는 것보다 더 크다. 쓰기 증폭은 메모리에 다시 쓰기 이전에 메모리가 소거되어야 하는 시나리오를 말한다. 데이터는 전형적으로, 예를 들어, 4 내지 8킬로바이트 크기인 페이지 크기로 쓰기되는 반면, 소거될 블록(소거 블록)은 전형적으로 크기가 훨씬 더 크다(예를 들어, 128킬로바이트 또는 일부 고밀도 저장 디바이스들에서는 십지어 몇 MB). 따라서, 데이터를 쓰거나 이동시킬 때, 예를 들어, 512바이트의 데이터가 쓰기되는 경우라도, 이 쓰기로 인해 훨씬 더 큰 데이터 블록을 이동시키고 소거해야만 할지도 모른다. 이러한 드라이브를 조각 모음하는 것이, 기본적인 의미를 모를 때, 저장 디바이스가 신뢰성 있게 동작할 수 있는 시간을 단축시킬 수 있다는 것과, 본 명세서에 기술된 실시예들이 조각화를 여전히 주기적으로 제거하면서 저장 디바이스가 신뢰성 있게 동작할 수 있는 시간을 연장시킬 수 있다는 것을 잘 알 것이다.

[0014] 회전하는 자기 및 광 드라이브에 대한 조각 모음은, 읽기 및 쓰기 헤드가 물리적으로 다른 파일 조각들의 근방에 있을 때 일어나는 I/O 파일라인에서의 최적화를 달성하기 위해, 파일 조각들이 드라이브 상에서 새로운 인접 위치들로 물리적으로 이동되는 것을 필요로 한다. 본 명세서에 기술된 실시예들은 콘텐츠가 저장되어 있는 주소 지정 가능 위치들을, 콘텐츠를 새로운 물리적으로 인접한 위치들로 실제로 복사할 필요 없이, 수정하는 것에 의해 I/O 패턴들을 최적화하기 위해 운영 체제가 비휘발성 저장 디바이스들을 어떻게 이용할 수 있는지를 보여줄 것이다. 비휘발성 저장 디바이스는 콘텐츠를, 관련 콘텐츠가 저장되어 있는 상이한 위치들의 탐색 주소(lookup address)를 논리적으로 인접해 있도록 수정하는 것에 의해 최적화될 수 있는 주소 지정 가능 위치에, 저장한다. 본 명세서에 기술된 실시예들은 저장 디바이스에 대한 조기 마모의 악영향을 초래함이 없이 유사한 I/O 성능 장점을 추가로 제공하고, 사진을 저장하는 것 또는 영화를 재생하는 것과 같은, 최종 사용자 작업 대신에 저장 시스템 콘텐츠의 상당한 양을 재배열하는 것과 연관된 최종 사용자 충격 및 전력의 소비를 피한다.

예 2 - 예시적인 비휘발성 저장 디바이스

[0015] 본 명세서에서 사용되는 바와 같이, 비휘발성 저장 디바이스는 전원을 켜 둘 필요 없이 그의 정보를 유지하는 임의의 반도체 기반 저장 디바이스를 지칭한다. 예를 들어, 비휘발성 저장 디바이스는 SSD(solid state drive), USB 플래시 드라이브, 칩 상의 내장 메모리, 상변화 메모리 디바이스(phase change memory device), 또는 임의의 다른 유형의 비휘발성 반도체 기반 저장소일 수 있다. 본 명세서에 기술된 실시예들은 또한, 데이터를 상이한 저장 페이지들로 실제로 복사할 필요 없이 블록 또는 페이지 주소 재지정을 통해 블록들을 순차적

레이아웃으로 재정렬하기 위해 본 명세서에 기술된 메커니즘들을 사용하여, 순서 있는 정보(ordered information)가, 랜덤 액세스 메모리(RAM, Random Access Memory)과 같이, 조각화로 인해 분산될 수 있는 임의의 시나리오에서 사용될 수 있다.

[0018] 본 명세서에서 사용되는 바와 같이, 비휘발성 메모리는 반도체 기반 저장소를 지칭하고, 따라서 자기 저장 디바이스(예컨대, 하드 디스크 드라이브) 또는 광 저장 디바이스(예컨대, CD 또는 DVD 매체)를 포함하지 않는다.

예 3 - 물리적 메모리의 주소 재지정

[0020] 자기 또는 광 저장 디바이스와 달리, 비휘발성 저장 디바이스는 데이터를 선형적으로 읽지 않는다. 예를 들어, 자기 저장 디바이스에서는, 읽기 및 쓰기 헤드가 플래터(platter) 상의 위치로 이동하고, 플래터가 회전할 때, 그 플래터로부터 정보를 읽어낸다. 자기 저장 디바이스가 플래터 상의 다른 위치에 있는 데이터를 읽고자 하는 경우, 읽기 및 쓰기 헤드가 새로운 위치로 이동해야 한다. 자기 저장 디바이스의 물리적 주소들은 플래터(들) 상의 위치들에 기초하여 배열되어 있다.

[0021] 다른 한편으로, 비휘발성 저장 디바이스는 읽기 및 쓰기 헤드를 사용하지 않고, 그 대신에, 개개의 트랜지스터들의 상태를 판정하는 것에 의해 정보를 읽을 수 있다. 전압이 트랜지스터에 가해질 때, 전류 흐름이 이진 데이터로서 검출된다. 병렬로 있는 많은 상이한 트랜지스터들에서 이 동작이 수행될 수 있다. 이들 디바이스가 물리적 읽기/쓰기 헤드를 특정의 위치로 이동시키는 것과 연관된 자연 시간을 겪지 않지만, 이들 디바이스는 운영 체제 및 애플리케이션이 보다 작은 트랜잭션들을 많이 사용할 때보다 데이터를 검색하거나 저장하기 위해 보다 적은 수의 그러나 보다 대규모의 액세스를 행할 때 성능 이점들을 나타낸다. 예를 들어, 성능 및 전력 소비 관점에서 볼 때, 하나의 연속된 순차적 파일 읽기 요청에 매핑되는 1 MB 청크를 읽는 것이 동일한 파일 페이로드를 검색하기 위해 512 바이트씩 2,000번 액세스를 수행하는 것보다 더 낫다. 본 명세서에 기술된 실시예들을 이용하는 시스템들은, 연속된 블록들을 해제(free up)하는 대신에, 데이터를 상이한 일련의 블록들로 덤프(dump)하는 것에 의해 높은 쓰기 속도를 제공할 수 있는데, 그 이유는 데이터가 물리적으로 인접한 주소 지정 가능 블록들에 실제로 위치되어 있는 것처럼 주소 지정되어 버리기 때문이다.

[0022] 그렇지만, 비휘발성 저장 디바이스를 사용하는 컴퓨팅 디바이스들, 및 저장 디바이스들 자체는 보통 비휘발성 저장 디바이스를 자기 저장 디바이스와 유사한 방식으로, 즉 선형 방식으로 읽어야 하는 것처럼, 취급한다. FTL(flash translation layer)은 데이터가 특정의 물리적 위치들에 있는 것처럼 보이게 할 수 있고, FTL은 물리적 메모리 주소와 비휘발성 저장 디바이스 상에서의 물리적 위치 간의 매핑을 추적한다. 이와 같이, 비휘발성 저장 디바이스는, 트랜지스터 위치들이 선형적으로 읽혀지는 것처럼 보일 수 있도록, 물리적 주소들을 트랜지스터 위치들에 할당한다.

[0023] 그렇지만, 저장 디바이스에서의 특정의 트랜지스터들의 물리적 위치들과 물리적 주소들에 대한 공통의 매핑 방식이 없다 - 즉, 물리적 주소가 저장 디바이스 상의 임의의 위치에 매핑될 수 있고, 근방의 물리적 주소들이 근방의 물리적 위치들에 매핑될 필요가 없으며, 그 대신에, 각각의 반도체 저장 디바이스 제조업체가 물리적 메모리 주소들을 저장 디바이스에 할당하는 그 자신의 방식을 제시할 수 있다 -. 예를 들어, 어떤 제조업체들은 물리적 메모리 주소들을 저장 디바이스에 하드 코딩(hard-code)할 수 있는 반면, 다른 제조업체들은, 예컨대, 마모 평준화(wear leveling)를 위해 메모리 주소들을 동적으로 할당할 수 있다.

예 4 - 예시적인 운영 환경

[0025] 본 명세서에서의 예들 중 임의의 예에서, 메모리 주소들을 주소 재지정하기 위한 운영 환경(100)이 제공될 수 있다. 도 1은 예시적인 운영 환경(100)을 나타낸 도면이다. 예시적인 운영 환경(100)은 조각 모음 애플리케이션(120) 및 운영 체제(130)를 포함하는 컴퓨팅 디바이스(110)를 포함한다. 예를 들어, 컴퓨팅 디바이스(110)는 휴대폰 또는 태블릿 컴퓨터와 같은, 모바일 컴퓨팅 디바이스일 수 있다.

[0026] 운영 체제(130)는 비휘발성 저장 디바이스(160)와 통신한다. 운영 체제(130)는 파일 시스템(140) 및 디바이스 드라이버(150)를 포함한다. 파일 시스템(140)은 비휘발성 저장 디바이스(160) 상에서의 파일들의 위치를 유지하고, 비휘발성 저장 디바이스(160)에의 액세스를 관리한다. 예를 들어, 파일 시스템(140)은, Microsoft Corporation이 그의 Windows 운영 체제를 위해 개발한 파일 시스템인, NTFS(New Technology File System)일 수 있다. 디바이스 드라이버(150)는 비휘발성 저장 디바이스를 제어하고, 운영 체제(130)와 비휘발성 저장 디바이스(160) 사이의 통신을 처리한다.

[0027] 도 1에서, 컴퓨팅 디바이스(110) 및 비휘발성 저장 디바이스(160)는 예시를 위해 개별적인 구성요소들로서 도시되어 있다. 그렇지만, 컴퓨팅 디바이스(110) 및 비휘발성 저장 디바이스(160)가 동일한 디바이스일 수 있다는

것을 잘 알 것이다.

[0028] 도 1에서, 운영 체제(130)는 비휘발성 저장 디바이스(160)와 통신하는 파일 시스템(140) 및 디바이스 드라이버(150)를 포함한다. 그렇지만, 운영 체제(130)는 비휘발성 저장 디바이스(160)와 통신하는 다른 구성요소들을 포함할 수 있다. 하나의 실시예에서, 메모리를 주소 재지정하라는 명령은 이러한 다른 운영 체제 구성요소들 중 하나로부터 올 수 있다.

[0029] 일례에서, 컴퓨팅 디바이스(110)는 조각 모음 애플리케이션(120)을 포함할 수 있다. 조각 모음 애플리케이션(120)이 도 1에서 운영 체제(130) 밖에 있는 것으로 도시되어 있지만, 조각 모음 애플리케이션(120)이 파일 시스템(140)에 통합되도록 또는 디바이스 드라이버(150)에 포함되도록 수정될 수 있다는 것을 잘 알 것이다. 게다가, 일부 실시예들에서, 조각 모음이 비휘발성 저장 디바이스(160) 자체 내에 통합될 수 있다.

[0030] 조각 모음 애플리케이션(120)이 컴퓨팅 디바이스(110) 상에서 실행될 때, 조각 모음 애플리케이션(120)은 저장 디바이스의 주소 재지정을 완수하기 위해 메모리 주소들을 주소 재지정하라고 비휘발성 저장 디바이스(160)에 명령할 수 있다. 조각 모음 애플리케이션(120)은 파일 시스템(140)에 저장된 각각의 파일이 디바이스 드라이버(150)를 통해 비휘발성 저장 디바이스(160)에서의 저장 주소들에 어떻게 매핑되는지를 검사할 수 있다. 조각 모음 애플리케이션(120)이 파일이 구성가능한 또는 미리 정해진 수(1, 2, 20 등) 조각들에 걸쳐 저장되어 있는 것으로 판정할 때, 조각 모음 애플리케이션(120)은 주소 재지정 방식을 실시할 수 있다. 다른 실시예들에서, 조각 모음 애플리케이션(120)은 자주 액세스되는 파일, 또는 임의의 수의 다른 허리스틱(heuristic)들(파일 크기, 시스템 파일, 사용자 파일, 기타 등등)과 같은 기준들을 사용할 수 있다. 조각 모음 애플리케이션(120)은 조각난 파일 주소 위치들을 갖는 파일 시스템(140) 및 디바이스 드라이버(150)를 통해 저장 디바이스(160)에 명령을 발행할 수 있고, 새로운 조각나지 않은(또는 덜 조각난) 주소 위치(들)를 갖는 응답을 다시 수신할 수 있다. 예를 들어, 어떤 파일이 15 개의 비연속된 저장 주소들에 걸쳐 분산되어 있는 것으로 발견되는 경우, 주소 재지정 후에, 파일 시스템은 그 파일을 15 개의 연속된 저장 주소 위치들로 간주한다. 파일 시스템(140)은 이어서 파일에 대한 읽기 또는 쓰기를 위해 순차적 액세스를 수행할 수 있고, 이는 각각의 조각을 검색하고 조립하기 위한 15 개의 개별 트랜잭션들보다 훨씬 더 빠르다. 본 명세서에 기술된 실시예들은 저장 디바이스(160)가 파일 조각들을 이용가능한 비어 있는 저장소(free storage)로 복사하지 않고, 그 대신에 디바이스 드라이버(150) 및 파일 시스템(140)이 보다 효율적인 전송 모드에서 동작하도록 저장 블록들을 연속된 주소 지정 가능 범위로 단순히 주소 재지정하는 것에 의해 주소 재지정을 완수하는 방법을 설명한다. 다른 실시예들에서, 조각 모음 애플리케이션(120)은, 파일 시스템(140) 및 디바이스 드라이버(150)를 통해, 파일이 제공된 파일 저장 주소들의 목록을 사용하여 연속적으로 되어야 한다고 저장 디바이스(160)에 간단히 명령할 수 있다. 명령이 성공 응답을 수신하는 경우, 파일 시스템은 새로운 주소 위치(들)를 사용해야 한다는 것을 알게 되는 반면, 오류 응답을 수신하는 경우, 나중에 주소 재지정을 재시도할 수 있다.

[0031] 대안적으로, 조각 모음 애플리케이션(120)이 비휘발성 저장 디바이스(160)에 존재할 수 있고, 운영 체제(130)는 조각 모음 애플리케이션(120)을 주기적으로 실행하라고 비휘발성 저장 디바이스(160)에 명령할 수 있으며, 비휘발성 저장 디바이스(160)가, 그 대신에, 저장 디바이스(160) 자체에 대해 메모리의 주소 재지정을 수행할 수 있다. 이 예에서, 저장 디바이스(160)는 파일 시스템(140)으로부터, 파일들의 목록 및 파일들이 저장되어 있는 조각 위치들과 같은, 정보를 제공받는다. 비휘발성 저장 디바이스(160)는, 주소 재지정을 완료한 후에, 파일 내용의 새로운 위치들을 설명하는 정보로 응답할 수 있고, 변경들을 디바이스 드라이버(150) 및 파일 시스템(140)에 업로드한다. 파일 시스템(140)은 이어서, 파일 블록들을 읽기 및 쓰기가 될 때, 주소 재지정된 위치들에 있는 파일 조각들에 대해 새로운 주소들을 사용할 것이다.

[0032] 일례에서, 디바이스 드라이버(150)가 조각 모음 애플리케이션(120)을 포함할 수 있거나, 디바이스 드라이버 밖에 있는 조각 모음 애플리케이션(120)이 비휘발성 저장 디바이스(160)를 조각 모음하거나 주소 재지정하는 루틴을 호출할 수 있다. 대안적으로, 디바이스 드라이버(150)가 주소 재지정 동작을 시작하는 것은 물론, 저장 디바이스(160)를 컴퓨팅 디바이스(110)에 통신 연결시키는 버스를 통해 저장 위치들을 주소 재지정하는 데 사용되는 특수 프로토콜 명령들로 통신하기 위해 그 자신의 조각 모음 애플리케이션(120)을 가질 수 있다.

예 5 - 주소 재지정을 수행하는 방법

[0034] 도 2는 비휘발성 저장 디바이스(160) 상의 조각난 파일의 메모리의 주소 재지정을 수행하는 예시적인 방법(200)의 플로우차트이다.

[0035] 210에서, 조각난 파일의 메모리를 주소 재지정하라는 명령이 비휘발성 저장 디바이스(160)로 송신된다.

[0036]

주소 재지정 명령(210)의 목적은 비휘발성 저장 디바이스(160)에게 본질적으로 랜덤한 I/O 액세스 패턴인 것처럼 보이는 몇 개의 비연속적으로 주소 지정된 블록들에 걸쳐 분산되어 있는 파일을 보다 적은 수의(예컨대, 하나의) 순차적 액세스로 변환하는 것이다. 본 명세서에 기술된 실시예들은 데이터를 새로운 저장 위치들로 물리적으로 복사할 필요 없이 저장 위치들의 주소 재지정을 완수하는데, 데이터를 새로운 저장 위치들로 물리적으로 복사하는 것은 주소 재지정보다 더 많은 전력을 사용하고, 저장소 수명에 부정적 영향을 미치며, 파일들을 조각 모음하는 것을 목표로 저장 내용을 운영 체제로 그리고 다시 저장 부분으로 복사하는 상당히 더 긴 I/O 사이클들을 유입시켜, 통상의 운영 체제 거동들 또는 사용자가 실행하고자 하는 애플리케이션들과 연관된 작업들을 방해할 수 있다.

[0037]

220에서, 조각난 파일의 메모리가 주소 재지정되었다는 응답이 비휘발성 저장 디바이스(160)로부터 수신된다.

[0038]

230에서, 파일 시스템(140)은 파일 조각들이 어디에 주소 지정되어 있는지를 유지하는 그의 내부 기록을 업데이트할 것이다. 일부 실시예들에서, 파일 시스템(140)은 210에서 명령이 송신될 때 그의 기록들을 업데이트하고, 220에서 성공 응답을 수신하지 않는 경우 주소 재지정 트랜잭션을 롤백(roll back)할 수 있다. 다른 실시예들에서, 파일 시스템(140)은 응답(220)에서 반환되는 새로운 주소 블록들에 기초하여 대응하는 주소 재지정 변경들을 커밋(commit)하거나 행하라는 응답을 수신할 때까지 기다릴 수 있다. 예를 들어, 응답(220)은 명령(210)에서 주소 재지정되도록 요청된 블록들에 대한 새로운 매핑들을 포함할 수 있고, 파일 시스템(140)이 230에서 업데이트될 때, 블록들에 대한 최종적인 합의된 주소 지정이 완료된다.

[0039]

일례에서, 파일 시스템(140)이 업데이트된 후에, 컴퓨팅 디바이스(110)는 이제 주소 재지정된 메모리를 반영하는 동작들을 수행할 수 있다.

[0040]

일례에서, 컴퓨팅 디바이스는 연속된 물리적 주소들을 포함하는 이제 주소 재지정된 메모리를 사용하여 추가의 명령을 비휘발성 저장 디바이스(160)로 송신할 수 있다. 예를 들어, 컴퓨팅 디바이스는 연속된 물리적 주소들에 있는 파일을 검색하기 위해 단일의 요청 또는 다수의 요청들의 팩(pack)을 송신할 수 있다. 파일이 연속된 물리적 주소들에 위치되어 있기 때문에, 컴퓨팅 디바이스의 동작들의 수가 감소된다. 비휘발성 저장 디바이스(160)에 의해 사용되는 내부 캐싱 메커니즘들이 보다 효율적으로 이용될 수 있는데, 그 이유는 주소 재지정 이후의 저장 요청이 데이터에 대한 연속된 순차적 요청으로서 구현될 수 있기 때문이다. 보다 작은 랜덤한 읽기 및 쓰기보다 보다 큰 순차적 읽기 및 쓰기에 본질적인 성능 이점들은 SD 카드, eMMC 디바이스, MMC, 및 SSD 드라이브와 같은, 최근의 저장 디바이스의 성능 벤치마크의 문서들에 의해 잘 입증되고 있다.

[0041]

도 3은 비휘발성 저장 디바이스(160) 상의 조각난 파일의 메모리의 주소 재지정을 수행하는 예시적인 방법(300)의 플로우차트이다. 도 3에 도시된 단계들은 도 2에 도시된 단계들에 대응한다. 310에서, 조각난 파일의 메모리를 주소 재지정하라는 명령이 수신된다.

[0042]

320에서, 연속된 물리적 메모리 주소들이 조각난 파일의 메모리에 할당된다. 즉, 이전에 복수의 비연속된 물리적 메모리 주소들에 위치되어 있던 파일 조각들 각각이 연속된 물리적 메모리 주소들로 주소 재지정된다.

[0043]

325에서, 비휘발성 저장 디바이스(160)는 주소 재지정 변경 처리의 오류를 반환할 수 있고, 시스템은 주소 재지정 변경이 행해지지 않고 주소 재지정이 중단(abort)되는 340으로 진행할 것이다. 주소 재지정이 성공적인 경우, 시스템은 330으로 진행할 것이다. 비휘발성 저장 디바이스(160)가 명령을 완료할 수 없는 경우, 운영 체제(130)는, 340으로 도시된 바와 같이, 파일 시스템(140)이 주소 재지정을 하지 않는 것의 일부로서 오류를 수신할 수 있다.

[0044]

330에서, 비휘발성 저장 디바이스(160)는 파일 조각들에 대한 새로운 주소 위치들로 [디바이스 드라이버(150) 및 파일 시스템(140)을 포함하는] 운영 체제(130)에 응답할 수 있다. 일부 실시예들에서, 컴퓨팅 디바이스(110)는 운영 체제(130)에 응답하기 위해 단계(330)를 수행할 필요가 없을 수 있는데, 그 이유는 비휘발성 저장 디바이스(160)가 명령을 간단히 완료하기 때문이다. 다른 실시예들에서, 응답이 블록들이 주소 재지정되었다는 성공 응답이기만 하면 될 수 있다.

[0045]

일부 실시예들에서, 주소 재지정 논리가 주소 재지정을 조각 모음되도록 만들기 위해 수정될 수 있는 블록들 및 이용가능한 블록들 전부를 추적하는 운영 체제(130)의 일부로서 포함될 수 있다. 다른 실시예들에서, 운영 체제는 비휘발성 저장 디바이스(10)가 블록들을 관리하도록 요청하고 매우 조각난 것으로 알고 있는 파일이 주소 재지정되도록 간단히 요구할 수 있으며, 새로운 블록 매핑들을 포함하는 응답을 기대하고 있다.

[0046]

마지막으로, 일부 실시예들에서, 주소 재지정은 파일의 원래 시작 블록 주소(original starting block address

s)를 유지할 것이고, 주소 재지정은 시작 주소 이후에 주소 지정되는 모든 후속 저장 블록들을 연속적이도록 만들 것이고, 따라서 그들이 순차적 액세스인 것처럼 보이지만; 후속 블록들은, 다른 파일들에 속하는 것으로 계산될 수 있는 주소들과 비교하여, 고유 주소들을 실제로 갖지 않을 수 있다. 이것은 2 개의 파일에 들어 있는 블록들이 외부 관찰자에게 블록들을 겹치게 하는 것처럼 보일 때 성긴 주소 지정(sparse addressing)을 포함하는 주소 재지정 해결책으로서 나중에 도 4c에서 상세히 설명될 것이다.

[0047] 도 1과 관련한 예시적인 구현에서, 비휘발성 저장 디바이스(160)는 조각난 파일의 메모리가 주소 재지정되었다는 응답(330)을 송신할 수 있지만, 응답이 다시 송신될 필요는 없다. 예를 들어, 비휘발성 저장 디바이스(160)는 조각난 파일을 주소 재지정하라는 명령을 수신하기만 할 수 있고, 운영 체제(130), 파일 시스템(140), 디바이스 드라이버(150) 또는 조각 모음 애플리케이션(120)은 비휘발성 저장 디바이스(160)가 정상적으로 동작하고 있는 경우 주소 재지정이 성공적으로 완료된 것으로 가정할 것이다.

[0048] 예를 들어, 도 1을 참조하면, 메모리를 주소 재지정하라는 명령이 파일 시스템(140), 디바이스 드라이버(150), 또는 비휘발성 저장 디바이스(160)로부터 나올 수 있다. 일 실시예에서, 비휘발성 저장 디바이스(160)에 명령을 제공하는, 컴퓨팅 디바이스(110)에서의 다른 운영 체제 구성요소가 있을 수 있다. 다른 실시예에서, 비휘발성 저장 디바이스(160)에 명령을 제공하는 별도의 구성요소가 운영 체제(130)와 비휘발성 메모리 사이에 존재할 수 있다. 조각 모음 애플리케이션(120)은 컴퓨팅 디바이스(110), 운영 체제(130), 파일 시스템(140), 및 디바이스 드라이버(150) 중 하나 이상에 존재할 수 있다. 조각 모음이 어디에서 시작되는지의 선택은, 다양한 공급업체들이 주소 재지정 해결책의 품질 및 비용을 어떻게 구현하는지에 기초하여 어느 애플리케이션 모델을 설치할지를 선택할 수 있는 시스템의 설계자에 달려 있다.

[0049] 도 1과 관련한 예시적인 구현에서, 메모리를 주소 재지정하라는 명령이 비휘발성 저장 디바이스(160)에 의해 수신된다. 그렇지만, 명령이 어느 조각난 파일들이 주소 재지정될 필요가 있는지를 명시하지 않을 수 있다. 예를 들어, 명령이 비휘발성 저장 디바이스(160)에 대한 조각 모음 요청의 일부일 수 있다. 이 경우에, 비휘발성 저장 디바이스(160)는 파일들의 조각화 정도에 기초하여 주소 재지정할 가능성이 가장 많은 후보 파일을 결정하고 그 파일을 주소 재지정하도록 선택할 수 있다. 그렇지만, 주소 재지정될 조각난 파일이 가장 많이 조각난 파일일 필요는 없다. 예를 들어, 비휘발성 저장 디바이스(160)는 운영 체제에 의한 파일의 액세스 빈도, 파일의 물리적 메모리 주소들의 위치, 또는 임의의 다른 기준들에 기초하여 가장 가능성 있는 후보 파일을 결정할 수 있다. 비휘발성 저장 디바이스(160)는 파일 시스템(140)으로부터 조각들을 갖는 모든 파일들의 목록을 제공받을 수 있거나, 디바이스 드라이버(150) 또는 운영 체제(130) 또는 심지어 조각 모음 애플리케이션(120)에 의해 추적되는 조각들을 갖는 모든 파일들의 목록을 제공받을 수 있다.

[0050] 비휘발성 저장 디바이스(160)는 본 명세서에 개시된 방법들 중 임의의 방법을 사용하여 주소 재지정을 수행할 수 있지만, 그 방법들로 제한되지 않는다. 비휘발성 저장 디바이스 상의 조각난 파일의 메모리를 주소 재지정하는 임의의 방법이 수행될 수 있다.

예 6 - 예시적인 가상 매핑

[0052] 도 4a 및 도 4b는 물리적 메모리 주소들을 주소 재지정하는 일례를 나타낸 도면이다. 이 예에서, 조각난 파일의 파일 조각들이 복수의 비연속된 물리적 주소들에 걸쳐 분산되어 있다. 예를 들어, 조각난 파일이 메모리 주소들 1, 3, 4 및 7에 위치되어 있는 것으로 가정하자. 저장 디바이스가 조각난 파일을 연속된 물리적 메모리 주소들로 주소 재지정하라는 명령을 수신하면, 저장 디바이스는 파일이 어느 물리적 메모리 주소들에 주소 재지정되어야 하는지를 결정한다. 이 예에서, 파일이 메모리 주소 1에서 시작하여 주소 재지정되지만, 그 대신에, 임의의 물리적 메모리 주소에서 시작하여 주소 재지정될 수도 있다.

[0053] 이 예에서, 이전의 메모리 주소 3이 새로운 메모리 주소 2로 주소 재지정된다. 그렇지만, 이전의 메모리 주소 2는 다른 데이터를 포함할 수 있다. 이와 같이, 메모리 주소들이 스와핑된다 - 즉, 이전의 메모리 주소 3이 새로운 메모리 주소 2로 주소 재지정되고, 이전의 메모리 주소 2가 새로운 메모리 주소 3으로 주소 재지정된다 -. 조각난 파일의 나머지 메모리 주소들 모두에 대해 이것이 반복된다. 최종 결과는 이전의 메모리 주소들 3, 4 및 7이 새로운 메모리 주소들 2, 3 및 4로 주소 재지정되어, 조각난 파일의 메모리가 이제 연속된 물리적 메모리 주소들에 주소 지정될 수 있게 하고, 이전의 메모리 주소 2가 새로운 메모리 주소 7로 주소 재지정되는 것이다. 파일 시스템(140)에 의해 관리되는 클러스터 또는 섹터 크기가 저장 디바이스(160)에서의 블록 크기들과 거의 1:1 관계에 있는 시나리오들에서, 이 구현은 도 4a에 도시된 것과 아주 유사하다. 파일 시스템(140)에 의해 관리되는 섹터 크기 및 클러스터 크기가 조각 모음을 수행하기 위해 데이터를 실제로 복사하는 일 없이 저장 위치들을 주소 재지정하는 기본 원리들을 위해 1:1 관계를 가질 필요가 없고, 이미 실시 중인 조각 모음 해결책

과 비교하여 데이터의 복사 및 쓰기가 더 적다는 것을 잘 알 것이다.

[0054] 그렇지만, 물리적 메모리 주소들이 꼭 스와핑될 필요는 없고, 그 대신에, 사용되지 않은 메모리 주소들로 주소 재지정될 수 있다. 예를 들어, 도 4b에서, 이전의 메모리 주소 2는 이용가능한 메모리 주소 100(예컨대, 비어 있는 이용가능한 메모리 주소)으로 주소 재지정될 수 있다. 조각난 파일의 다른 메모리 주소들이 이어서 연속된 물리적 주소들로 주소 재지정될 수 있다.

[0055] 도 4c는 파일에 대한 원래의 고유 시작 블록 주소를 유지하는 주소 재지정 메커니즘의 대안의 실시예를 설명한다. 이 실시예에서, 주소 재지정은 시작 주소 이후에 주소 지정된 후속 저장 블록들 모두를 연속적으로 만들고, 따라서 그들이 파일 시스템(140) 또는 운영 체제(130)에게 순차적 액세스인 것처럼 보이지만; 두 번째 및 후속 블록들은 다른 파일들에 속하는 것으로 계산될 수 있는 주소들과 비교하여 고유 주소들을 실제로 갖지 않을 수 있다(예컨대, 두 번째 및 후속 블록들은 공유가능 물리적 메모리 주소들을 가질 수 있다). 파일이 고유 시작 주소 및 블록 단위로 된 특정의 길이만을 사용하여 검색되기 때문에, 그리고 파일이 항상 순차적 액세스이기 때문에, 초기 고유 블록 주소 이후의 콘텐츠의 스트림은 명확하게 주소 지정될 수 있다. 예를 들어, 도 4c에서, 상부의 일련의 블록들은 주소 재지정 이전의 상태를 나타내고; 블록 1에서 시작하고 블록 3 및 블록 5에 부가 조각들을 포함하여 총 길이가 3 블록(각각, 조각들 a.txt₁, a.txt₂, a.txt₃으로 도시되어 있음)인 파일 "a.txt"를 나타내고 있다. 시스템은 또한 블록 2에서 시작하고 블록 4 및 블록 6에 조각들(각각, b.txt₁, b.txt₂, b.txt₃으로 도시되어 있음)로서 저장된 콘텐츠를 가지는 파일 "b.txt"를 가질 수 있다. 파일 "a.txt"에 대한 주소 재지정 명령이 파일 시스템(140)에 의해 송신되고 블록들 1, 3, 5에 있는 파일이 순차적으로 되도록(즉, 블록 1에서 시작하여 길이가 3 블록이도록) 주소 재지정을 명령할 수 있다. 이 주소 재지정은 블록 1에 있는 콘텐츠는 그대로 둘 것이지만, 이어서 블록 3을 블록 2로서 주소 재지정하고(블록 2가 블록 1 읽기에 뒤따라올 때에만) 블록 5를 블록 3으로서 주소 재지정할 것이다(블록 3이 블록 1 및 블록 2의 읽기에 뒤따라올 때에만). 따라서, 주소 재지정 후에, 외부 소스로부터 순차적으로 액세스될 때, 파일 "a.txt"는 블록들 1, 2, 및 3에 저장되어 있다(도 4c의 하반부에 도시되어 있음). 비휘발성 저장 디바이스(160)는, 파일 시스템(140)이 단일 길이 블록 동안 블록 2 또는 3을 읽으려고 시도하는 경우, 시스템이 블록 1을 읽고 그 후에 단일의 순차적 액세스의 일부로서 선택적으로 블록들 2 및 3을 읽는 것에 의해서만 파일을 검색해야 한다는 것을 알고 있기 때문에, 오류를 보고할 것이다. 콘텐츠가 명령을 사용하여 블록 1에서 시작하여 3의 길이로 순차적으로 주소 지정되어 있으면 하다면, 콘텐츠가 파일 시스템(140) 및 저장소 드라이버(150)에 독자적으로 제공된다. 파일 시스템(140)은 또한 블록 2에서 시작하고 블록들 4 및 6을 포함하는 파일 "b.txt"가, 도 4c에 도시된 것과 유사한 메커니즘을 사용하여 그 파일이 블록들 2, 3, 4로 되는 것으로 가정하여, 주소 재지정되도록 명령한다. 파일 시스템(140)과 같은 외부 관찰자는 "a.txt" 및 "b.txt" 둘 다의 시작 주소 및 길이가 블록들 2 및 3에 있는 콘텐츠를 공유하고 따라서 콘텐츠가 동일한 저장 블록들을 차지하고 있는 것으로 계산할 수 있다. 그렇지만, "a.txt" 및 "b.txt"에의 액세스가 항상 고유의 주소들에서 시작하는 순차적 액세스이기 때문에, 저장 부분이, 블록 1에서 시작하여 3 블록 길이의 액세스를 수신할 때에는, 파일 "a.txt"에만 매핑되어 있는 고유의 콘텐츠를 전달할 것이고, 블록 2에서 시작하는 3 블록 길이의 요청을 수신할 때에는, 파일 "b.txt"에 대한 콘텐츠만을 검색할 것임을 비휘발성 저장 디바이스(160)가 이해하고 있기 때문에, 그렇지 않을 수 있다. 비휘발성 저장 디바이스(160)는 블록 주소 1 또는 2를 임의의 다른 파일들에 제공하지 않을 것이고, 일부 실시예들에서, 파일 시스템(140)은 파일을 뒤져서 중첩되어 있는 블록들에 액세스하지 않고 시작 주소를 사용하여 파일들에 액세스해야 한다는 것만을 알고 있다. 또한, 일부 실시예들에서, 파일 시스템(140)은 블록 3, 4, 5 또는 6의 시작 주소를 가지는 시작 저장 블록을 수신하지 않을 것인데, 그 이유는 이 블록들이 파일들 "a.txt" 및 "b.txt"에 의해 실제로 사용되고 있기 때문이다. 대안의 실시예들이 블록 3, 4, 5 또는 6의 시작 주소들을 제공할 수 있지만; 파일 시스템(140)에 제공되는 총 주소 블록들이 비휘발성 저장 디바이스(160)의 저장 용량을 초과하지 않을 것이다. 게다가, 도 4c는, 이용가능한 시작 블록 주소로 3을 제공하기 위해, 비어 있는 블록 7이 주소 재지정되는 것을 도시하고 있다. 파일 시스템(140)이 새로운 파일 "c.txt"를 1 블록 쓰기로서 주소 3에 저장해야 하는 것으로 가정하면, 파일 시스템(140)은, 주소 2에서 시작하는 3 블록 액세스를 읽을 때, 파일 "b.txt" 콘텐츠를 수신할 것으로 예상하고, 주소 3에서 시작하는 단일 블록 읽기를 발행할 때 "c.txt" 콘텐츠를 획득할 것으로 예상할 것이다. 일부 실시예들에서, 이 유형의 주소 재지정은 순차적 방식으로 읽혀지는 파일들에 대해서만 수행될 것이다. 일부 실시예들에서, 파일 시스템(140)은 이러한 방식으로 주소 재지정되는 이러한 종류의 파일들을 탐색하지 않도록 속성들을 설정할 수 있다.

[0056] 도 4a 내지 도 4c의 예들 모두가 파일 시스템(140)에 의해 추적되는 파일 조각들에 대한 개별 저장 단위들(즉, 클러스터)과 저장 블록들[비휘발성 저장 디바이스(160)에 의해 제공되는 개별 저장 단위, 즉 블록 또는 섹터]

간의 1:1 매핑을 나타내고 있다는 것을 잘 알 것이지만; 본 명세서에 기술된 실시예들에서, 파일 조각들이 저장 블록의 서브 부분(sub portion)을 차지하는 것은 물론, 각각의 클러스터에 저장된 파일 조각들이 몇 개의 주소 지정 가능 저장 블록들에 걸쳐 매핑되는 것을 쉽게 알 수 있을 것이다. 즉, 파일 조각(클러스터)와 저장 블록(섹터)의 비는 1:1, 2:1, 1:2, 1:16, 16:1 등일 수 있다. 주소 재지정 처리 동안 페이지 일부(partial page)가 이동될 필요가 있을 때 데이터의 복사가 최소한으로 행해지면서 저장에 대한 고유 주소가 제공되도록 하기 위해 주소 재지정 원리들이 간단히 확장된다. 그리고 주소 재지정 전후에 파일들 "a.txt" 및 "b.txt"에 의해 소비되는 저장 공간이 일정한 채로 유지되어 비휘발성 저장 디바이스(160)의 6개의 블록을 소비할 것이다.

[0057] 비휘발성 저장 디바이스(160)에서의 메모리의 물리적 위치들을 이동시키는 일 없이 (예컨대, 도 4a 및 도 4b를 참조하여 앞서 기술된 바와 같이) 물리적 메모리 주소들의 주소 재지정이 수행된다. 도 5a 및 도 5b는 도 4a 및 도 4b에 기술된 주소 재지정 동작들이 메모리의 물리적 위치들을 이동시키는 일 없이 어떻게 수행되는지를 나타낸 도면이다. 이전의 예를 생각해보면, 이전의 메모리 주소들 1, 3, 4 및 7이 새로운 메모리 주소들 1, 2, 3 및 4로 주소 재지정된다. 그렇지만, 메모리 디바이스 상에서의 조각난 파일의 파일 조각들의 실제의 물리적 위치들은 이동되지 않는다. 이 예를 참조하면, 도 5a는 주소 재지정 이전의 메모리 주소들 1, 3, 4 및 7을 나타낸 것이다. 도 5a에 도시된 바와 같이, 비휘발성 저장 디바이스(160)는 메모리 주소들 1, 3, 4 및 7에 대응하는 파일 조각들을, 510에 "LOC1" 내지 "LOC4"로서 간략화된 형태로 도시되어 있는, 비휘발성 저장 디바이스(160) 내의 특정의 물리적 위치들에 저장한다. 도 5b는 메모리 주소 재지정이 수행된 후의 메모리 주소들을 나타낸 것이다. 도 5b에 도시된 바와 같이, 메모리 주소들이 이제 연속적(주소들 1, 2, 3, 4)이도록 주소 재지정이 수행되었다. 또한, 도 5b에 도시된 바와 같이, 주소 재지정이 수행되었더라도, 비휘발성 저장 디바이스(160)에서의 메모리의 물리적 위치들이 변경되지 않았다. 따라서, 예를 들어, 비록 이전의 메모리 주소 3이 새로운 메모리 주소 2로 주소 재지정되었지만, 메모리의 물리적 위치가 변하지 않았다.

[0058] 일부 실시예들에서, 이러한 주소 변환(address translation)을 수행하고 재매핑(remapping)을 지원하는 소프트웨어 및/또는 하드웨어가 비휘발성 저장 디바이스(160) 내부에 포함될 수 있다. 다른 실시예들에서, 재매핑은 파일 시스템(140), 저장소 드라이버(150), 및 비휘발성 저장 디바이스(160)에 걸친 분산형 해결책일 수 있다. 예를 들어, 파일 시스템(140)이 논리 블록과 물리 블록 간의 매핑을 추적하고, 이 변경을 적용하는 재매핑 해결책을 비휘발성 저장 디바이스(160)에 제공할 수 있다. 다른 실시예들에서, 저장소 드라이버는, 자신이 알기로, 파일 시스템(140)이 저장 디바이스(160)에서의 저장 블록들에 매핑한 주소들 간의 변환을 수행할 수 있고, 따라서, 파일 시스템(140)이 재매핑을 모르는 상태로, 재매핑을 저장 디바이스에 제공한다.

예 7 - 예시적인 매핑 테이블

[0060] 도 6a 및 도 6b는 LBA 매핑 테이블과 도 4a 및 도 4b에서의 조각난 파일의 물리적 주소들 간의 매핑의 일례를 나타낸 테이블이다. 예를 들어, LBA 매핑 테이블은 비휘발성 저장 디바이스(160) 상의 물리적 주소들에 논리 주소들을 할당하기 위해 운영 체제(130)에 의해 사용될 수 있다. 이전의 예에서, 물리적 주소들이 주소 재지정되었기 때문에, LBA 매핑 테이블이 주소 재지정된 메모리에 기초하여 업데이트된다. 이와 같이, 예를 들어, 도 6a에서, LBA 0000은 물리적 주소 1을 가리키고, LBA 0001은 물리적 주소 3을 가리키며, LBA 0002는 물리적 주소 4를 가리키고, LBA 0003은 물리적 주소 7을 가리킨다. 주소 재지정 후에, 도 6b에 도시된 바와 같이, LBA 0000은 물리적 주소 1을 가리키고, LBA 0001은 물리적 주소 2를 가리키며, LBA 0002는 물리적 주소 3을 가리키고, LBA 0003은 물리적 주소 4를 가리킨다.

[0061] 도 6a 및 도 6b에서 도시된 바와 같이, 메모리의 주소 재지정을 반영하기 위해 LBA 매핑 테이블이 업데이트될 수 있다. 그렇지만, LBA 매핑 테이블이 꼭 업데이트될 필요는 없다. 예를 들어, LBA 매핑 테이블과 저장 디바이스 사이에 가상 매핑 테이블이 존재할 수 있다. 가상 매핑 테이블이 메모리의 주소 재지정의 새로운 정보로 업데이트될 수 있다. LBA 매핑 테이블이 주소를 탐색할 때, 업데이트된 가상 매핑 테이블이 주소 재지정된 물리적 주소들을 가리킬 수 있고, LBA 매핑 테이블은 이러한 주소 재지정이 일어났다는 것을 모르고 있다. 이 경우에, LBA 매핑 테이블은 주소 재지정된 물리적 메모리 주소들에 대한 정보를 포함하는 가상 매핑 테이블과 통신한다.

예 8 - 예시적인 컴퓨팅 환경

[0062] 도 7은 기술된 혁신들이 구현될 수 있는 적당한 컴퓨팅 환경(300)의 일반화된 일례를 나타낸 것이다. 이 혁신들이 다양한 범용 또는 특수 목적 컴퓨팅 시스템들에서 구현될 수 있기 때문에, 컴퓨팅 환경(700)은 용도 또는 기능의 범주에 관한 어떤 제한을 암시하기 위한 것이 아니다. 예를 들어, 컴퓨팅 환경(700)은 각종의 컴퓨팅 디바이스들(예컨대, 데스크톱 컴퓨터, 태블릿 컴퓨터, 서버 컴퓨터, 태블릿 컴퓨터, 미디어 플레이어, 게임 시스

템, 모바일 디바이스 등) 중 임의의 것일 수 있다.

[0064] 도 7을 참조하면, 컴퓨팅 환경(700)은 적어도 하나의 처리 유닛들(710, 715) 및 메모리(720, 725)를 포함한다. 도 7에서, 이 기본적인 구성(730)은 파선 내에 포함되어 있다. 처리 유닛들(710, 715)은 컴퓨터 실행가능 명령어들을 실행한다. 처리 유닛은 범용 CPU(central processing unit), ASIC(application-specific integrated circuit) 내의 프로세서, 또는 임의의 다른 유형의 프로세서일 수 있다. 다중 처리 시스템에서, 처리 능력을 증가시키기 위해 다수의 처리 유닛들이 컴퓨터 실행가능 명령어들을 실행한다. 예를 들어, 도 7은 중앙 처리 유닛(710)은 물론, 그래픽 처리 유닛 또는 코프로세싱 유닛(co-processing unit)(715)을 나타내고 있다. 유형적 메모리(tangible memory)(720, 725)는 처리 유닛(들)에 의해 액세스가능한, 휘발성 메모리(예컨대, 레지스터, 캐시, RAM), 비휘발성 메모리(예컨대, ROM, EEPROM, 플래시 메모리 등), 또는 이 둘의 어떤 조합일 수 있다. 메모리(720, 725)는 본 명세서에 기술된 하나 이상의 혁신들을 구현하는, 처리 유닛(들)에 의한 실행에 적당한 컴퓨터 실행가능 명령어들의 형태로 된 소프트웨어(780)를 저장한다.

[0065] 컴퓨팅 시스템은 부가의 특징들을 가질 수 있다. 예를 들어, 컴퓨팅 환경(700)은 저장소(740), 하나 이상의 입력 디바이스들(750), 하나 이상의 출력 디바이스들(760), 및 하나 이상의 통신 연결들(770)을 포함한다. 버스, 제어기 또는 네트워크와 같은 상호연결 메커니즘(도시 생략)은 컴퓨팅 환경(700)의 구성요소들을 상호연결시킨다. 전형적으로, 운영 체제 소프트웨어(도시 생략)는 컴퓨팅 환경(700)에서 실행되는 다른 소프트웨어에 대한 운영 환경을 제공하고, 컴퓨팅 환경(700)의 구성요소들의 활동들을 조정한다.

[0066] 유형적 저장소(740)는 이동식 또는 비이동식일 수 있고, 정보를 비일시적 방식으로 저장하는 데 사용될 수 있고 컴퓨팅 환경(700) 내에서 액세스될 수 있는, 자기 디스크, 자기 테이프 또는 카세트, CD-ROM, DVD, 또는 임의의 다른 매체를 포함한다. 저장소(740)는 본 명세서에 기술된 하나 이상의 혁신들을 구현하는 소프트웨어(780)에 대한 명령어들을 저장한다.

[0067] 입력 디바이스(들)(750)는 키보드, 마우스, 펜, 또는 트랙볼 등의 터치 입력 디바이스, 음성 입력 디바이스, 스캐닝 디바이스, 또는 컴퓨팅 환경(700)에 입력을 제공하는 다른 디바이스일 수 있다. 비디오 인코딩의 경우, 입력 디바이스(들)(750)는 아날로그 또는 디지털 형태로 비디오 입력을 받아들이는 카메라, 비디오 카드, TV 투너 카드, 또는 유사한 디바이스, 또는 비디오 샘플을 컴퓨팅 환경(700)으로 읽어들이는 CD-ROM 또는 CD-RW일 수 있다. 출력 디바이스(들)(760)는 컴퓨팅 환경(700)으로부터의 출력을 제공하는, 디스플레이, 프린터, 스피커, CD-라이터(CD-writer), 또는 다른 디바이스일 수 있다.

[0068] 통신 연결(들)(770)은 통신 매체를 통해 다른 컴퓨팅 엔터티로의 통신을 가능하게 한다. 통신 매체는 컴퓨터 실행가능 명령어들, 오디오 또는 비디오 입력 또는 출력, 또는 다른 데이터와 같은 정보를 피변조 데이터 신호(modulated data signal)로 전달한다. 피변조 데이터 신호는 신호의 특성을 중 하나 이상이 정보를 그 신호에 인코딩하는 방식으로 설정 또는 변경된 신호이다. 제한이 아닌 예로서, 통신 매체는 전기, 광, RF, 또는 다른 반송파를 사용할 수 있다.

[0069] 혁신들이 일반적으로 컴퓨팅 시스템에서 실제 또는 가상의 목표 프로세서 상에서 실행되는 프로그램 모듈들에 포함된 것과 같은 컴퓨터 실행가능 명령어들과 관련하여 기술될 수 있다. 일반적으로, 프로그램 모듈은 특정의 작업들을 수행하거나 특정의 추상 데이터 형식들을 구현하는 루틴, 프로그램, 라이브러리, 객체, 클래스, 구성 요소, 데이터 구조 등을 포함한다. 프로그램 모듈들의 기능이 다양한 실시예들에서 원하는 바에 따라 프로그램 모듈들 간에 결합되거나 분할되어 있을 수 있다. 프로그램 모듈들에 대한 컴퓨터 실행가능 명령어들은 로컬 또는 분산 컴퓨팅 시스템 내에서 실행될 수 있다.

[0070] "시스템" 및 "디바이스"라는 용어는 본 명세서에서 서로 바꾸어 사용될 수 있다. 문맥이 명백히 달리 나타내지 않는 한, 용어가 컴퓨팅 시스템 또는 컴퓨팅 디바이스의 유형에 대한 어떤 제한도 암시하지 않는다. 일반적으로, 컴퓨팅 시스템 또는 컴퓨팅 디바이스는 로컬이거나 분산되어 있을 수 있고, 본 명세서에 기술된 기능을 구현하는 소프트웨어를 갖는 특수 목적 하드웨어 및/또는 범용 하드웨어의 임의의 조합을 포함할 수 있다.

예 9 - 예시적인 모바일 디바이스

[0071] 도 8은 각종의 선택적인 하드웨어 및 소프트웨어 구성요소들(전체적으로 802로 나타내어져 있음)을 포함하는 예시적인 모바일 디바이스(800)를 나타낸 시스템도이다. 모바일 디바이스 내의 임의의 구성요소들(802)은 임의의 다른 구성요소와 통신할 수 있지만, 예시의 편의를 위해, 연결들 모두가 도시되어 있지는 않다. 모바일 디바이스는 각종의 컴퓨팅 디바이스들[예컨대, 셀폰, 스마트폰, 핸드헬드 컴퓨터, PDA(Personal Digital Assistant) 등] 중 임의의 것일 수 있고, 셀룰러, 위성 또는 다른 네트워크와 같은, 하나 이상의 이동 통신 네트워크들

(804)과 무선 양방향 통신을 할 수 있다.

[0073]

예시된 모바일 디바이스(800)는 신호 코딩, 데이터 처리, 입출력 처리, 전력 제어, 및/또는 다른 기능들과 같은 작업들을 수행하기 위한 제어기 또는 프로세서(810)(예컨대, 신호 프로세서, 마이크로프로세서, ASIC, 또는 다른 제어 및 처리 논리 회로)를 포함할 수 있다. 운영 체제(812)는 구성요소들(802)의 할당 및 사용과 하나 이상의 애플리케이션들(814)에 대한 지원을 제어할 수 있다. 애플리케이션들(814)은 통상의 모바일 컴퓨팅 애플리케이션들(예컨대, 이메일 애플리케이션, 달력, 연락처 관리자, 웹 브라우저, 메시징 애플리케이션) 또는 임의의 다른 컴퓨팅 애플리케이션을 포함할 수 있다. 애플리케이션 저장소에 액세스하기 위한 기능(813)이 또한 애플리케이션들(814)을 획득하고 업데이트하는 데 사용될 수 있다.

[0074]

예시된 모바일 디바이스(800)는 메모리(820)를 포함할 수 있다. 메모리(820)는 비이동식 메모리(822) 및/또는 이동식 메모리(824)를 포함할 수 있다. 비이동식 메모리(822)는 RAM, ROM, 플래시 메모리, 하드 디스크, 또는 다른 공지된 메모리 저장 기술들을 포함할 수 있다. 이동식 메모리(824)는 플래시 메모리 또는 GSM 통신 시스템들에서 공지되어 있는 SIM(Subscriber Identity Module) 카드, 또는 "스마트 카드"와 같은 다른 공지된 메모리 저장 기술들을 포함할 수 있다. 메모리(820)는 운영 체제(812) 및 애플리케이션들(814)을 실행하기 위한 데 이터 및/또는 코드를 저장하는 데 사용될 수 있다. 예시적인 데이터는 하나 이상의 유선 또는 무선 네트워크들을 통해 하나 이상의 네트워크 서버들 또는 다른 디바이스들로 송신되고 그리고/또는 그로부터 수신되는 웹 페이지, 텍스트, 영상, 사운드 파일, 비디오 데이터, 또는 다른 데이터 집합을 포함할 수 있다. 메모리(820)는 IMSI(International Mobile Subscriber Identity)와 같은 가입자 식별자(subscriber identifier), 및 IMEI(International Mobile Equipment Identifier)와 같은 장비 식별자(equipment identifier)를 저장하는 데 사용될 수 있다. 이러한 식별자는 사용자 및 장비를 식별하기 위해 네트워크 서버로 전송될 수 있다.

[0075]

모바일 디바이스(800)는 터치스크린(832), 마이크(834), 카메라(836), 물리적 키보드(838) 및/또는 트랙볼(840)과 같은 하나 이상의 입력 디바이스들(830), 그리고 스피커(852) 및 디스플레이(854)와 같은 하나 이상의 출력 디바이스들(850)을 지원할 수 있다. 다른 가능한 출력 디바이스들(도시 생략)은 압전 또는 다른 햅틱 출력 디바이스들을 포함할 수 있다. 일부 디바이스들은 2개 이상의 입출력 기능을 할 수 있다. 예를 들어, 터치스크린(832) 및 디스플레이(854)는 단일의 입출력 디바이스로 결합될 수 있다.

[0076]

입력 디바이스들(830)은 NUI(Natural User Interface)를 포함할 수 있다. NUI는 사용자가, 마우스, 키보드, 리모콘 등과 같은 입력 디바이스들에 의해 부과되는 인위적 제약조건들 없이, "자연적인" 방식으로 디바이스와 상호작용할 수 있게 하는 임의의 인터페이스 기술이다. NUI 방법의 예는 음성 인식, 터치 및 스타일러스 인식, 화면 상에서는 물론 화면에 인접하여 제스처 인식, 공중 제스처, 머리 및 눈 추적, 음성 및 발화, 시각, 터치, 제스처, 및 기계 지능에 의존하는 것을 포함한다. NUI의 다른 예는 가속도계/자이로스코프를 사용하는 움직임 제스처 검출, 얼굴 인식, 3D 디스플레이, 머리, 눈, 및 시선 추적, 물입형 증강 현실 및 가상 현실 시스템 - 이들 모두는 보다 자연스런 인터페이스를 제공함 - 은 물론, 전기장 감지 전극(EEG 및 관련 방법)을 사용하여 두뇌 활동을 감지하기 위한 기술들을 포함한다. 이와 같이, 하나의 특정 예에서, 운영 체제(812) 또는 애플리케이션들(814)은 사용자가 음성 명령을 통해 디바이스(800)를 동작시킬 수 있게 하는 음성 사용자 인터페이스(voice user interface)의 일부로서 음성 인식 소프트웨어를 포함할 수 있다. 게다가, 디바이스(800)는 게임 애플리케이션에 입력을 제공하기 위해 제스처를 검출하고 해석하는 것과 같은, 사용자의 공간 제스처를 통한 사용자 상호작용을 가능하게 하는 입력 디바이스 및 소프트웨어를 포함할 수 있다.

[0077]

기술 분야에서 잘 알고 있는 바와 같이, 무선 모뎀(860)은 안테나(도시 생략)에 결합될 수 있고, 프로세서(810)와 외부 디바이스들 사이의 양방향 통신을 지원할 수 있다. 모뎀(860)은 총칭하여 도시되어 있고, 이동 통신 네트워크(804)와 통신하기 위한 셀룰러 모뎀 및/또는 다른 무선 기반 모뎀들[예컨대, 블루투스(864) 또는 WiFi(862)]을 포함할 수 있다. 무선 모뎀(860)은 전형적으로 단일의 셀룰러 네트워크 내에서의, 셀룰러 네트워크들 간의, 또는 모바일 디바이스와 PSTN(public switched telephone network) 간의 데이터 및 음성 통신을 위한 GSM 네트워크와 같은, 하나 이상의 셀룰러 네트워크들과 통신하도록 구성되어 있다.

[0078]

모바일 디바이스는 적어도 하나의 입출력 포트(880), 전원(882), GPS(Global Positioning System) 수신기와 같은 위성 항법 시스템 수신기(satellite navigation system receiver)(884), 가속도계(886), 및/또는 USB 포트, IEEE 1394(Fire Wire) 포트, 및/또는 RS-232 포트일 수 있는 물리적 커넥터(890)를 추가로 포함할 수 있다. 예시된 구성요소들(802)이 필수적이지도 모두 포함되는 것도 아닌데, 그 이유는 임의의 구성요소들이 삭제될 수 있고 다른 구성요소들이 추가될 수 있기 때문이다.

[0079] 예 10 - 예시적인 구현

[0080]

개시된 방법들 중 일부의 동작이, 제시의 편의상, 특정의 순차적 순서로 기술되어 있지만, 이하에 기재되는 특정의 표현(language)에 의해 특정의 순서가 요구되지 않는 한, 이러한 기재 방식이 재배열을 포함한다는 것을 잘 알 것이다. 예를 들어, 순차적으로 기술되는 동작들이, 어떤 경우에, 재배열되거나 동시에 수행될 수 있다. 더욱이, 간단함을 위해, 첨부 도면이 개시된 방법들이 다른 방법들과 함께 사용될 수 있는 다양한 방식을 나타내지 않을 수 있다.

[0081]

개시된 방법들 중 임의의 방법이 하나 이상의 컴퓨터 판독가능 저장 매체 상에 저장되고 컴퓨팅 디바이스(예컨대, 컴퓨팅 하드웨어를 포함하는 스마트폰 또는 다른 모바일 디바이스를 비롯한, 임의의 이용가능한 컴퓨팅 디바이스) 상에서 실행되는 컴퓨터 실행가능 명령어들로서 구현될 수 있다. 컴퓨터 판독가능 저장 매체는 컴퓨팅 환경 내에서 액세스될 수 있는 임의의 이용가능한 유형적 매체[예컨대, 하나 이상의 광 매체 디스크, 휘발성 메모리 구성요소(DRAM 또는 SRAM 등), 또는 비휘발성 메모리 구성요소(플래시 메모리 또는 하드 드라이브 등)와 같은, 비일시적 컴퓨터 판독가능 매체]이다. 예로서, 도 7을 참조하면, 컴퓨터 판독가능 저장 매체는 메모리 (720 및 725), 및 저장소(740)를 포함한다. 예로서, 도 8을 참조하면, 컴퓨터 판독가능 저장 매체는 메모리 (820, 822, 및 824)를 포함한다. 컴퓨터 판독가능 저장 매체라는 용어는 신호 및 반송파와 같은, 통신 연결들(예컨대, 770, 860, 862, 및 864)을 포함하지 않는다.

[0082]

개시된 기법들을 구현하기 위한 컴퓨터 실행가능 명령어들 중 임의의 것은 물론, 개시된 실시예들의 구현 동안 생성되고 사용되는 임의의 데이터가 하나 이상의 컴퓨터 판독가능 매체 상에 저장될 수 있다. 컴퓨터 실행가능 명령어들은, 예를 들어, 전용 소프트웨어 애플리케이션 또는 웹 브라우저 또는 다른 소프트웨어 애플리케이션(원격 컴퓨팅 애플리케이션 등)을 통해 액세스되거나 다운로드되는 소프트웨어 애플리케이션의 일부일 수 있다. 이러한 소프트웨어는, 예를 들어, 단일의 로컬 컴퓨터(예컨대, 임의의 적당한 상업적으로 이용가능한 컴퓨터)에서 또는 하나 이상의 네트워크 컴퓨터들을 사용하는 네트워크 환경에서[예컨대, 인터넷, 원거리 통신망, 근거리 통신망, 클라이언트-서버 네트워크(클라우드 컴퓨팅 네트워크 등), 또는 다른 이러한 네트워크를 통해] 실행될 수 있다.

[0083]

명확함을 위해, 소프트웨어 기반 구현들의 특정의 선택된 측면들만이 기술되어 있다. 기술 분야에 공지되어 있는 다른 상세는 생략되어 있다. 예를 들어, 개시된 기술이 임의의 특정의 컴퓨터 언어 또는 프로그램으로 제한되지 않는다는 것을 잘 알 것이다. 예를 들어, 개시된 기술이 C++, Java, Perl, JavaScript, Adobe Flash, 또는 임의의 다른 적당한 프로그래밍 언어로 작성된 소프트웨어로 구현될 수 있다. 마찬가지로, 개시된 기술이 임의의 특정의 컴퓨터 또는 유형의 하드웨어로 제한되지 않는다. 적당한 컴퓨터 및 하드웨어의 특정의 상세는 공지되어 있으며, 본 개시 내용에 상세히 기재될 필요가 없다.

[0084]

또한, 본 명세서에 기술된 임의의 기능이 소프트웨어 대신에, 적어도 부분적으로, 하나 이상의 하드웨어 논리 구성요소들에 의해 수행될 수 있다는 것을 잘 알 것이다. 제한이 아닌 예로서, 사용될 수 있는 예시적인 유형의 하드웨어 논리 구성요소들은 FPGA(Field-programmable Gate Array), ASIC(Program-specific Integrated Circuit), ASSP(Program-specific Standard Product), SOC(System-on-a-chip system), CPLD(Complex Programmable Logic Device) 등을 포함한다.

[0085]

게다가, (예를 들어, 컴퓨터로 하여금 개시된 방법들 중 임의의 것을 수행하게 하는 컴퓨터 실행가능 명령어들을 포함하는) 소프트웨어 기반 실시예들 중 임의의 것이 적당한 통신 수단을 통해 업로드, 다운로드, 또는 원격적으로 액세스될 수 있다. 이러한 적당한 통신 수단은, 예를 들어, 인터넷, 월드 와이드 웹, 인트라넷, 소프트웨어 애플리케이션, 케이블(광섬유 케이블을 포함함), 자기 통신, 전자기 통신(RF, 마이크로파, 및 적외선 통신을 포함함), 전자 통신, 또는 다른 이러한 통신 수단을 포함한다.

[0086]

개시된 방법들, 장치들, 및 시스템들이 결코 제한하는 것으로 해석되어서는 안된다. 그 대신에, 본 개시 내용은, 단독으로 그리고 서로와의 다양한 컴비네이션 및 서브컴비네이션으로 된, 다양한 개시된 실시예들의 모든 신규의 비자명한 특징들 및 측면들에 관한 것이다. 개시된 방법들, 장치들, 및 시스템들은 그의 임의의 특정의 측면 또는 특징 또는 조합으로 제한되지 않으며, 개시된 실시예들은 임의의 하나 이상의 특정의 장점들이 존재하거나 문제점들이 해결되어야 하는 것을 요구하지 않는다.

[0087]

대안

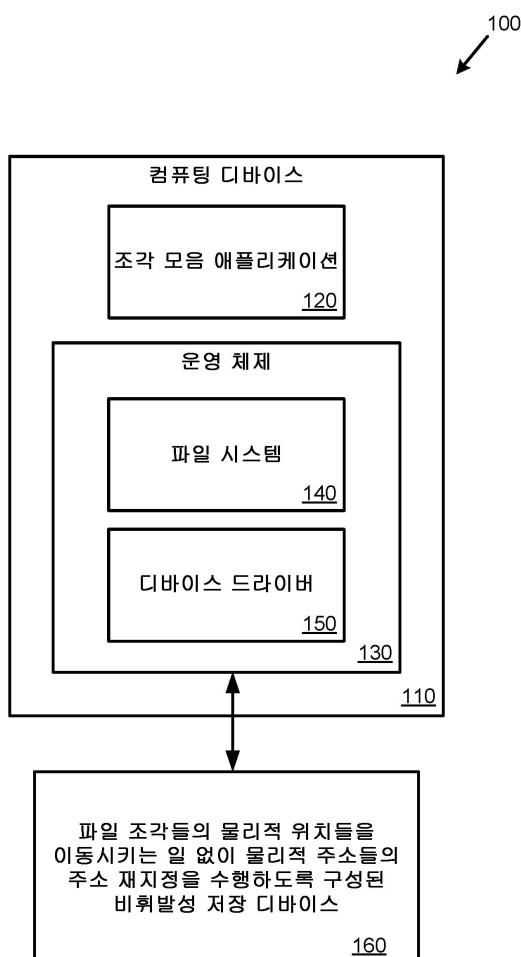
[0088]

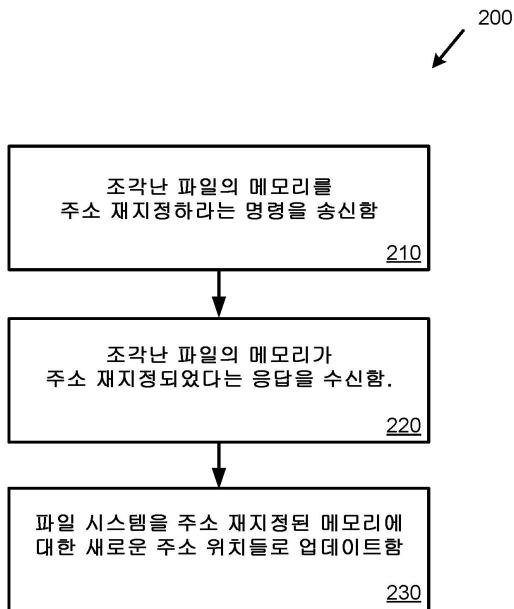
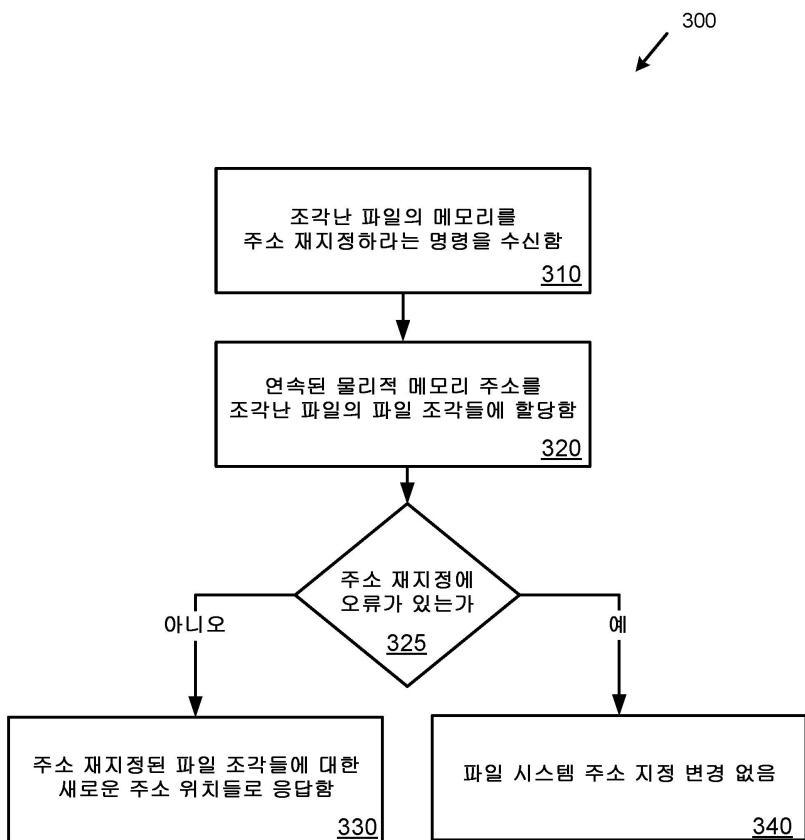
임의의 예로부터의 기술들이 다른 예들 중 임의의 하나 이상에 기술된 기술들과 결합될 수 있다. 개시된 기술의 원리들이 적용될 수 있는 많은 가능한 실시예들을 바탕으로, 예시된 실시예들이 개시된 기술의 예이고 본기술의 범주에 대한 제한으로서 해석되어서는 안된다는 것을 잘 알 것이다. 오히려, 개시된 기술의 범주는 이하

의 청구범위에 의해 포함되는 것을 포함한다. 따라서, 이 청구범위의 범주 내에 속하는 모든 것을 본 발명으로서 청구한다.

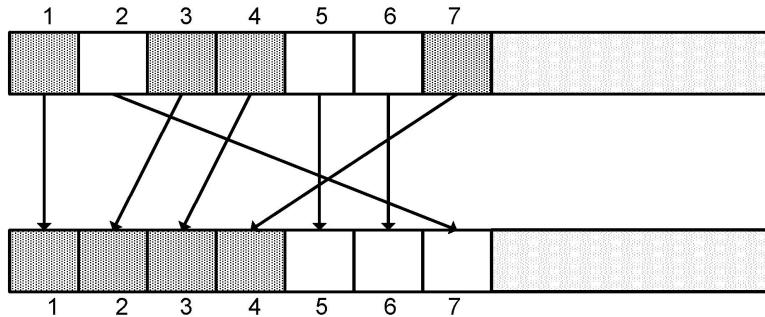
도면

도면1

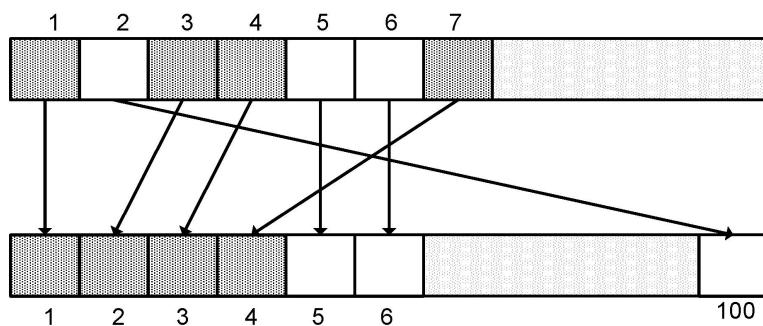


도면2**도면3**

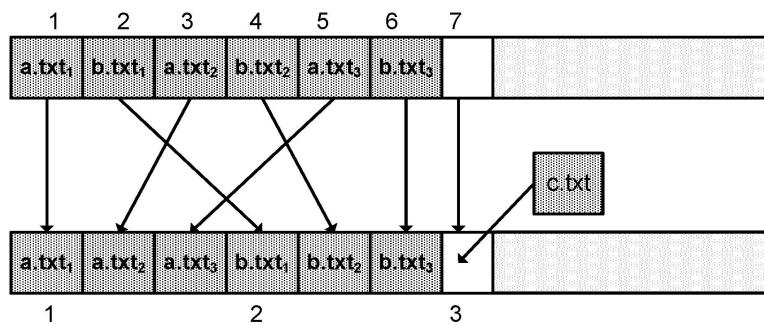
도면4a



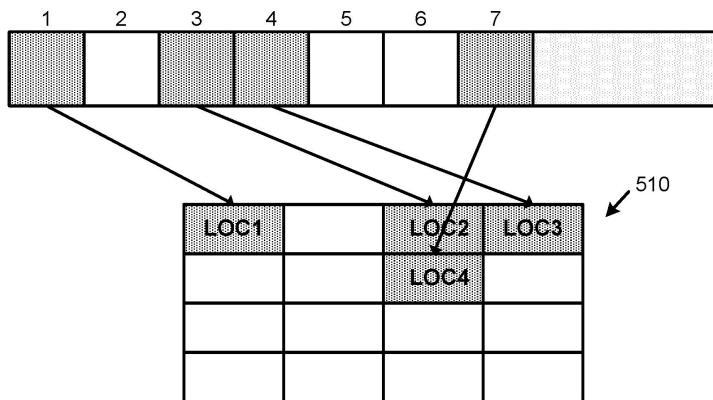
도면4b



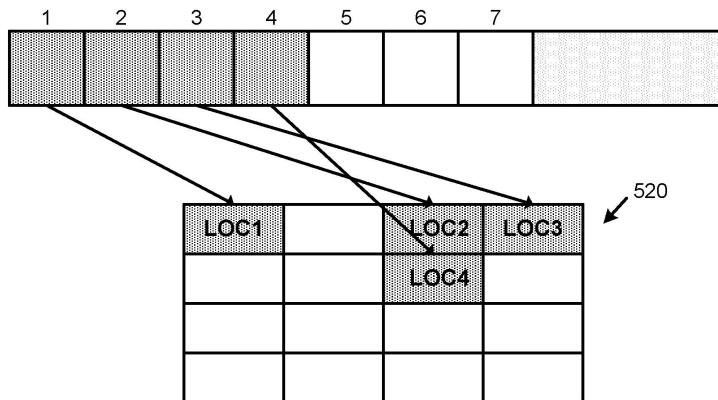
도면4c



도면5a



도면5b



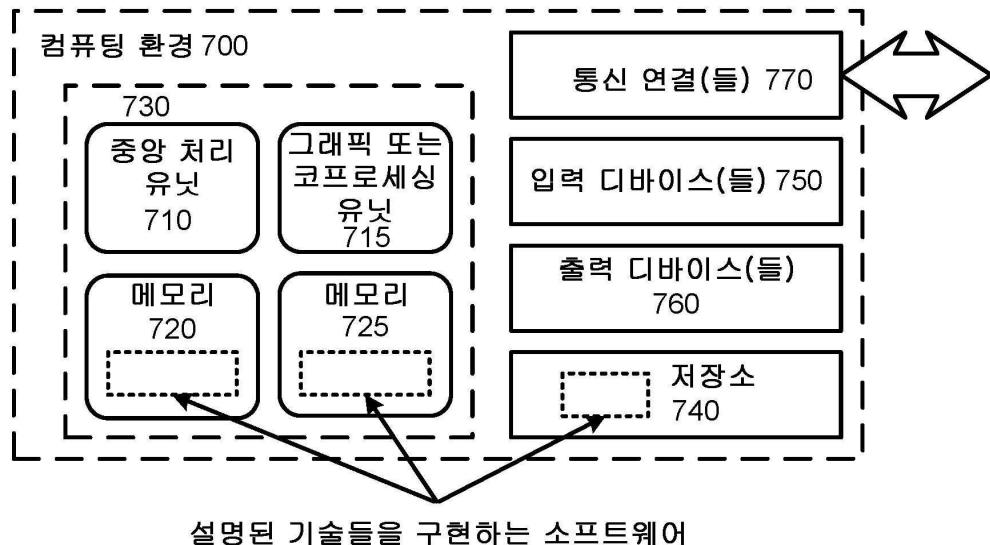
도면6a

LBA	물리적 주소
0000	1
0001	3
0002	4
0003	7

도면6b

LBA	물리적 주소
0000	1
0001	2
0002	3
0003	4

도면7



도면8

