



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0075845
(43) 공개일자 2021년06월23일

(51) 국제특허분류(Int. Cl.)
G06F 16/27 (2019.01) G06F 16/182 (2019.01)
G06F 16/28 (2019.01) G06F 3/06 (2006.01)
H04L 29/08 (2006.01)
(52) CPC특허분류
G06F 16/27 (2019.01)
G06F 16/182 (2019.01)
(21) 출원번호 10-2020-0140924
(22) 출원일자 2020년10월28일
심사청구일자 없음
(30) 우선권주장
62/948,192 2019년12월13일 미국(US)
16/746,803 2020년01월17일 미국(US)

(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
최창호
95120 미국 캘리포니아주 산호세, 마운트 홀리 드
라이브 6622
천, 커
95133 미국 캘리포니아주 산호세 타임 워크웨이
1020
비손, 티모씨 시
95124 미국 캘리포니아주 산호세 로스우드 드라이
브 2132
(74) 대리인
하영옥

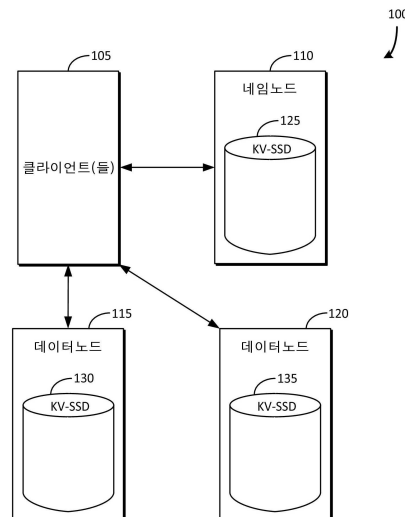
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 네이티브 키-밸류 분산 스토리지 시스템

(57) 요약

본 발명은 네이티브 키-밸류 분산 스토리지 시스템을 지원하기 위한 방법, 장치 및 시스템을 포함한다. 상기 시스템은 KV-SSD를 갖는 네임노드 및 각각 하나 이상의 KV-SSD를 포함하는 하나 이상의 데이터노드를 포함한다. 상기 시스템은 상기 네임노드 및 상기 하나 이상의 데이터노드에 통신 가능하게 연결된 클라이언트 장치를 포함한다. 상기 클라이언트 장치는 네이티브 키-밸류 스토리지 및 네트워킹 스택을 포함한다. 일부 실시예들은 블록-기반 파일 및 네이티브 키-밸류 튜플을 모두 지원하는 하이브리드 블록-기반 네이티브 키-밸류 분산 저장 시스템을 포함할 수 있다.

대표도 - 도1



(52) CPC특허분류

G06F 16/284 (2019.01)

G06F 3/067 (2013.01)

G06F 3/0688 (2013.01)

H04L 67/1097 (2013.01)

명세서

청구범위

청구항 1

키-밸류 솔리드 스테이트 드라이브(KV-SSD)를 포함하는 네임노드;

하나 이상의 KV-SSD를 각각 포함하는 하나 이상의 데이터노드; 및

상기 네임노드 및 상기 하나 이상의 데이터노드에 통신 가능하게 연결된 클라이언트 장치를 포함하며, 상기 클라이언트 장치는 네이티브 키-밸류 스토리지 및 네트워킹 스택을 포함하는, 네이티브 키-밸류 분산 스토리지 시스템.

청구항 2

제1항에 있어서,

상기 클라이언트 장치는 상기 네이티브 키-밸류 스토리지 및 네트워킹 스택에 통신 가능하게 연결된 애플리케이션 계층을 포함하고;

상기 클라이언트 장치는 동작에 대한 호출에 응답하여 키-밸류 튜플이 생성되도록 구성되고,

상기 동작에 대한 호출은 상기 애플리케이션 계층 내에서 이루어지고, 및

상기 키-밸류 튜플은 상기 하나 이상의 데이터노드의 상기 하나 이상의 KV-SSD에 저장되는, 네이티브 키-밸류 분산 스토리지 시스템.

청구항 3

제2항에 있어서,

상기 키-밸류 튜플은 상기 하나 이상의 KV-SSD 중 제1 KV-SSD 및 상기 하나 이상의 KV-SSD 중 제2 KV-SSD에 걸쳐 스트라이프 되는, 네이티브 키-밸류 분산 스토리지 시스템.

청구항 4

제3항에 있어서,

상기 스트라이프된 키-밸류 튜플은 상기 제1 KV-SSD에 저장된 제1 내부 키-밸류 튜플과 상기 제2 KV-SSD에 저장된 제2 내부 키-밸류 튜플을 포함하는, 네이티브 키-밸류 분산 스토리지 시스템.

청구항 5

제2항에 있어서,

상기 동작에 대한 상기 호출은 제1 동작에 대한 제1 호출이고;

상기 클라이언트 장치는 제2 동작에 대한 제2 호출에 응답하여 상기 키-밸류 튜플에 액세스되도록 구성되고; 및

상기 제2 동작에 대한 상기 제2 호출은 상기 애플리케이션 계층 내에서 이루어지는, 네이티브 키-밸류 분산 스토리지 시스템.

청구항 6

제5항에 있어서,

상기 제2 동작은 네이티브 키-밸류 get() 동작 또는 또는 네이티브 키-밸류 put() 동작 중 적어도 하나인, 네이티브 키-밸류 분산 스토리지 시스템.

청구항 7

제5항에 있어서,

상기 클라이언트 장치는 제3 동작에 대한 제3 호출에 응답하여 상기 키-밸류 튜플을 삭제하도록 구성되고, 상기 제3 동작에 대한 상기 제3 호출은 상기 애플리케이션 계층 내에서 이루어지는, 네이티브 키-밸류 분산 스토리지 시스템.

청구항 8

제1항에 있어서,

상기 네임노드는 데이터노드의 순서화된 목록을 포함하는 키-밸류 맵핑 섹션을 포함하고; 및

데이터노드의 순서화된 목록에는 하나 이상의 데이터노드가 포함되는, 네이티브 키-밸류 분산 스토리지 시스템.

청구항 9

제8항에 있어서,

상기 순서화된 목록은 제1 순서화된 목록이고;

제1 순서화된 목록은 제1 키-밸류 튜플과 연관되고,

상기 키-밸류 맵핑 섹션은 데이터노드의 제2 순서화된 목록을 포함하고; 및

상기 제2 순서화된 목록은 제2 키-밸류 튜플과 연관된, 네이티브 키-밸류 분산 스토리지 시스템.

청구항 10

제1항에 있어서,

상기 네임노드는 상기 하나 이상의 데이터노드 각각의 총 용량을 저장하거나 추적하도록 구성된 공간 관리 섹션을 포함하는, 네이티브 키-밸류 분산 스토리지 시스템.

발명의 설명**기술 분야**

[0001] 본 실시예는 스토리지 시스템에 관한 것으로, 보다 상세하게는 키-밸류 솔리드 스테이트 드라이브 (KV-SSD)와 같은 키-밸류 스토리지 장치를 사용하여 분산 스토리지 시스템의 전체 아키텍처에 걸쳐 키-밸류 지원을 제공하기 위한 시스템 및 방법에 관한 것이다.

배경 기술

[0002] 클라이언트에 키-밸류 인터페이스를 제공하는 일부 분산 스토리지 시스템에서는, 블록 인터페이스가 일부 SSD에 의해 노출될 수 있으므로, 키와 밸류를 블록 인터페이스에 매핑해야 한다. 이러한 개별 블록의 관리는 스토리지 시스템 스택의 여러 계층에서 수행될 수 있다. 예를 들어, 일부 분산 파일 시스템(DFS)에서 데이터노드는 블록 관리를 수행할 수 있다. 이 접근법의 한가지 한계는 클라이언트의 키-밸류 튜플이, SSD와 같은, 블록 스토리지 장치에 의해 노출될 때, 키-밸류 데이터 구조를 블록 인터페이스에 매핑하기 위해 소프트웨어의 추가 계층이 필요할 수 있다는 것일 수 있다.

[0003] 또한, 일부 마스터/슬레이브 분산 스토리지 시스템에서, 마스터 노드(즉, 네임노드)는 파일 시스템 메타데이터를 관리할 수 있다. 디렉토리는, 명명된 항목이 파일이든 디렉토리이든, 상기 명명된 항목을 내부에 저장할 수 있다. 그러나 파일은 사용자 콘텐츠를 저장할 수 있다. 사용자 콘텐츠를 저장하기 위해, 파일은 블록의 순서화된 세트를 포함하며, 각 블록은 고정된 크기이고 분산 스토리지 시스템의 스토리지 노드에 상주한다. 블록-기반 시스템에서 키-밸류 튜플을 사용하는 것은 키-밸류 데이터 구조를 블록 인터페이스에 매핑하기 위한 소프트웨어의 추가 계층을 요구하므로, 성능과 효율성에 부정적인 영향을 미친다.

발명의 내용

해결하려는 과제

[0004] 본 개시가 해결하고자 하는 과제는 키-밸류 튜플을 블록에 매핑할 필요가 없는 네이티브 키-밸류 분산 스토리지 시스템을 제공하는 것이다.

과제의 해결 수단

[0005] 본 발명은 네이티브 키-밸류 분산 스토리지 시스템을 지원하기 위한 방법, 장치 및 시스템을 포함한다. 상기 시스템은 KV-SSD를 갖는 네임노드 및 각각 하나 이상의 KV-SSD를 포함하는 하나 이상의 데이터노드를 포함할 수 있다. 상기 시스템은 상기 네임노드 및 상기 하나 이상의 데이터노드에 통신 가능하게 연결된 클라이언트 장치를 포함할 수 있다. 상기 클라이언트 장치는 네이티브 키-밸류 스토리지 및 네트워킹 스택을 포함할 수 있다.

[0006] 일부 실시예들은 블록-기반 파일 및 네이티브 키-밸류 튜플을 모두 지원하는 하이브리드 블록-기반 네이티브 키-밸류 분산 저장 시스템을 포함할 수 있다. 상기 하이브리드 시스템은 KV-SSD를 포함하는 네임노드, 각각 하나 이상의 KV-SSD를 포함하는 하나 이상의 제1 데이터노드 및 각각 하나 이상의 블록-기반 SSD를 포함하는 하나 이상의 제2 데이터노드를 포함할 수 있다. 상기 클라이언트 장치는 상기 네임노드, 상기 하나 이상의 제1 데이터노드 및 상기 하나 이상의 제2 데이터노드에 통신 가능하게 연결될 수 있다. 상기 클라이언트 장치는 하이브리드 블록-기반 네이티브 키-밸류 스토리지 및 네트워킹 스택을 포함할 수 있다.

발명의 효과

[0007] 본 개시의 네이티브 키-밸류 분산 스토리지 시스템은 키-밸류 튜플을 블록에 매핑할 필요가 없을 수 있다. 맵핑에 대한 이러한 감소된 요구는 소프트웨어 계층을 더 간단하게 할 수 있고, 개시된 시스템의 오버 헤드를 줄일 수 있으며, 더 큰 확장성을 가능하게 할 수 있다.

도면의 간단한 설명

[0008] 본 발명의 원리의 상기 및 추가 특징 및 장점은 첨부 도면을 참조하여 이루어진 다음의 상세한 설명으로부터 보다 쉽게 명백해질 것이다.

도 1은 일부 실시예에 따른 네이티브 키-밸류 분산 저장 시스템의 예시적인 다이어그램이다.

도 2a는 도 1의 네이티브 키-밸류 분산 저장 시스템의 예시도이며, 일부 실시예에 따른 예시적인 세부 사항을 도시한다.

도 2b는 도 1의 네이티브 키-밸류 분산 저장 시스템의 예시도이며, 일부 실시예들에 따른 다른 예시적인 세부 사항을 도시한다.

도 3은 일부 실시예들에 따른 키-밸류 맵핑 섹션 및 자유 목록 공간 관리 섹션을 갖는 네임노드를 포함하는 네이티브 키-밸류 분산 저장 시스템의 또 다른 예시적인 다이어그램이다.

도 4는 일부 실시예들에 따른 클라이언트 장치 및 대상 장치를 포함하는 네이티브 키-밸류 분산 저장 시스템의 또 다른 예시적인 다이어그램이다.

도 5는 일부 실시예들에 따른 네이티브 키-밸류 분산 저장 시스템의 동작의 타이밍 흐름도의 예시적인 다이어그램이다.

도 6은 일부 실시예들에 따른 하이브리드 블록-기반 네이티브 키-밸류 분산 저장 시스템의 예시적인 다이어그램이다.

도 7은 일부 실시예들에 따라 분산 저장 시스템에서 네이티브 키-밸류 튜플 요청 및 블록 기반 파일 요청의 하이브리드 조합을 처리하기 위한 기술을 도시한 흐름도이다.

도 8은 일부 실시예들에 따른 하이브리드 키-밸류 분산 저장 시스템을 포함하는 컴퓨팅 시스템의 예시적인 블록 다이어그램이다.

발명을 실시하기 위한 구체적인 내용

[0009] 이하, 본 명세서에 개시된 실시예들을 상세히 참조할 것이며, 그 예는 첨부 도면에 도시되어 있다. 다음의 상세한 설명에서, 본 발명의 개념을 완전히 이해할 수 있도록 다수의 특정 세부 사항이 설명된다. 그러나, 당업자는

이러한 특정 세부 사항없이 본 발명의 개념을 실시할 수 있음을 이해해야 한다. 다른 경우들에서, 공지된 방법들, 절차들, 컴포넌트들, 회로들 및 네트워크들은 실시예들의 양태들을 불필요하게 모호하게 하지 않기 위해 상세히 설명되지 않았다.

[0010] 본 명세서에서 제1, 제2 등의 용어가 다양한 요소를 설명하기 위해 사용될 수 있지만, 이들 요소는 이들 용어에 의해 제한되지 않아야 한다는 것이 이해될 것이다. 이러한 용어는 하나의 요소를 다른 요소와 구별하기 위해서만 사용된다. 예를 들어, 제1 인터페이스는 제2 인터페이스로 지칭될 수 있고, 유사하게, 제2 인터페이스는 본 발명의 범위를 벗어나지 않으면서 제1 인터페이스로 지칭될 수 있다.

[0011] 본 명세서에서 본 발명의 설명에 사용된 용어는 특정 실시예를 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 발명의 개념 및 첨부된 청구 범위의 설명에 사용된 바와 같이, 단수 형태는 문맥 상 명백하게 달리 나타내지 않는 한 복수 형태도 포함하도록 의도된다. 본원에 사용된 용어 "및/또는"은 하나 이상의 관련된 열거된 항목의 임의의 및 모든 가능한 조합을 지칭하고 포함하는 것으로 이해될 것이다. 본 명세서에서 사용될 때 용어 "포함하다" 및/또는 "포함하는"은 언급된 특징, 정수, 단계, 연산, 요소 및/또는 구성 요소의 존재를 명시하지만 존재를 배제하지는 않는 것으로 이해 될 것이다. 또는 하나 이상의 다른 특징, 정수, 단계, 연산, 요소, 구성 요소 및/또는 이들의 그룹의 추가. 도면의 구성 요소 및 특징은 반드시 축척대로 그려지는 것은 아니다.

[0012] 본 명세서에 개시된 실시예는 기본 스토리지 장치가 KV-SSD인 분산 저장 시스템을 포함한다. 또한 키-밸류 데이터 구조는 분산 스토리지 시스템의 클라이언트에 노출될 수 있다. 따라서, 키-밸류 인터페이스는 클라이언트로부터 개별 KV-SSD까지 보존될 수 있으므로, 키-밸류-대-블록 매핑의 필요성을 줄이거나 없앨 수 있다. 일부 실시예는 클라이언트 키-밸류 튜플을 관리하기 위한 기술을 포함한다. 예를 들어, 키-밸류 튜플은 스토리지 노드에서 하나의 KV-SSD에 상주할 수 있고, 또는 대안적으로, 키-밸류 튜플을 하나 이상의 스토리지 노드에서 여러 KV-SSD에 걸쳐 스트라이프하여 노드-레벨 병렬 처리를 활용할 수 있다.

[0013] 일부 분산 스토리지 시스템은 블록-기반 프로토콜만 사용할 수 있으며, 원격 대상 SSD의 블록 주소를 지정하여 원격 스토리지 장치에 직접 쓸 수 있다. 대조적으로, 키-밸류 인터페이스의 네이티브 지원을 제공하고 KV-SSD를 활용하기 위해, 본 명세서에 개시된 네트워킹 프로토콜은 키-밸류 인터페이스 및/또는 put(), get() 및 delete()와 같은 기능을 지원한다. 또한, 본 명세서에 개시된 바와 같이, 일부 실시예는 스토리지가 정상 블록-기반 파일 및 키-밸류 튜플 모두를 포함할 때 메타데이터를 관리하기 위한 하이브리드 스토리지 관리 기술을 포함한다. 이 하이브리드 접근법은, 또한 키-밸류 튜플에 대한 지원을 포함하도록, 블록 스토리지에 정보를 저장할 수 있는 메타데이터노드를 수정함으로써 달성될 수 있다. 키-밸류 튜플(즉, 비-하이브리드 시스템)을 저장하는 스토리지 시스템의 경우, 블록-기반 메타데이터노드가 필요하지 않을 수 있다. 대신, 클라이언트 메타데이터 또한 KV-SSD에 저장되는, 키-밸류-전용 메타데이터노드가 사용될 수 있다. 이러한 시스템에는 블록-기반 스토리지가 없을 수 있다.

[0014] 도 1은 일부 실시예에 따른 네이티브 키-밸류 분산 저장 시스템(100)의 예시적인 다이어그램이다. 시스템(100)은, 예를 들어, 하나 이상의 클라이언트(105), 네임노드(110), 제1 데이터노드(115) 및 제2 데이터노드(120)를 포함할 수 있다. 네임노드(110)는 키 위치 정보를 저장할 수 있고, 데이터노드는, 후술하는 바와 같이, 하나 이상의 KV-SSD에 키-밸류 튜플을 저장할 수 있다. 임의의 적절한 수의 데이터노드가 포함될 수 있다는 것이 이해될 것이다. 클라이언트(105)는 컴퓨터 서버, 개인용 컴퓨터, 랩톱 컴퓨터, 태블릿 컴퓨터 등과 같은 컴퓨팅 장치일 수 있다. 네임노드(110)는 하나 이상의 KV-SSD(125)을 저장할 수 있다. 제1 데이터노드(115)는 하나 이상의 KV-SSD(130)에 대한 액세스를 제공할 수 있다. 제2 데이터노드(120)는 하나 이상의 KV-SSD(135)에 대한 액세스를 제공할 수 있다. 일부 실시예에서, 네임노드(110)는 데이터노드(115 또는 120) 중 하나에 저장될 수 있으며, 이에 의해 KV-SSD(125)에 대한 필요성을 제거할 수 있다. 네이티브 키-밸류 분산 스토리지 시스템(100)은, 하나 이상의 클라이언트(105)로부터 아래로 개별 KV-SSD(예를 들어, 130 및/또는 135)까지, 전체에 걸쳐 네이티브 키-밸류 지원을 제공할 수 있다. 본 명세서에서 사용되는 용어 "네이티브(native)"는 하나 이상의 키-밸류 튜플을 생성, 액세스 및/또는 삭제하기 위해 하나 이상의 인터페이스가 제공되고, 하나 이상의 인터페이스가 시스템(100) 전체에서, 하나 이상의 클라이언트(105)로부터 아래로 개별 KV-SSD(예를 들어, 130 및/또는 135)까지, 이 기능을 제공한다는 것을 의미한다.

[0015] 도 2a는 도 1의 네이티브 키-밸류 분산 스토리지 시스템(100a)의 예시적인 다이어그램이다. 도 1은 일부 실시예에 따른 추가 세부 사항을 도시한다. 시스템(100a)을 사용하면, 일부 블록-기반 스토리지 시스템에 의해 요구될 수 있는 바와 다르게, 이 개시된 시스템이 키-밸류 튜플을 블록에 매핑할 필요가 없을 수 있다. 맵핑에 대한 이

러한 감소된 요구는 소프트웨어 계층을 더 간단하게 할 수 있고, 개시된 시스템의 오버 헤드를 줄일 수 있으며, 더 큰 확장성을 가능하게 한다. 도 2a에 도시된 바와 같이, 네임노드(110)는 키 위치 정보(205)를 저장할 수 있다. 키 위치 정보(205)는 데이터노드의 순서화된 목록으로 저장될 수 있다. 예를 들어, KEY 1은 제1 데이터노드(115) 및 제2 데이터노드(120)와 연관될 수 있다. 다른 예로서, KEY 2는 제1 데이터노드(115)에만 단지 관련될 수 있다. 동일한 키는 동일한 키 식별자를 사용하여 데이터노드에 저장될 수 있다. 예를 들어, 도시된 바와 같이 제1 데이터노드(115)에는, KEY 1 및 KEY 2가 모두 그들의 대응하는 밸류와 함께 저장될 수 있다. 도시된 바와 같이 제2 데이터노드(120)에는, KEY 1만이 대응하는 밸류와 함께 저장된다.

[0016] 몇몇 측면에서, KV-SSD는 스토리지 장치 유닛으로서 사용될 수 있다. 이와 같이, 클라이언트의 키-밸류 튜플(예를 들어, KEY 1 : FOO, BAR)은 도 2a에 도시된 바와 같이, 다수의 내부 키-밸류 튜플로 분해될 수 있다. 알 수 있는 바와 같이, 키-밸류 튜플(KEY 1 : FOO, BAR)은 제1 데이터노드(115) 및 제2 데이터노드(120)를 가로질러 분해 및 스트라이핑 될 수 있다. 다르게 말하면, KEY1 : FOO는 KV-SSD(130)에 저장될 수 있고, KEY1 : BAR은 KV-SSD(135)에 저장될 수 있다. 클라이언트 키-밸류 튜플을 여러 개의 내부 키-밸류 튜플로 나누는 것과 키-밸류 튜플을 여러 개의 고정된-크기 블록으로 나누는 것의 한 가지 차이점은 내부 키-밸류 튜플을 사용하면 그것들의 크기가 동적일 수 있다는 것이다.(예를 들어, 1, 2, N 등) 대조적으로, 종래의 블록-기반 키-밸류 튜플과 함께, 튜플을 저장하는데 필요한 블록의 수는 고정될 수 있고(즉, 키-밸류 크기에 기초할 수 있고), 이는 블록 내부의 단편화를 야기할 수 있다.

[0017] 도 2a에 도시된 예에서, KEY 2는 6개의 문자(즉, PEN CIL)를 포함하는 밸류를 가지며 제1 데이터노드(115)에 단일 키-밸류 튜플로서 완전히 저장되는 반면, KEY 1은 6개의 문자(즉, FOO BAR)를 포함하는 밸류이지만, 2개의 반쪽으로 분해되며, 반쪽 각각은 3개의 문자를 가지며, 제1데이터노드(115) 및 제2데이터노드(120)에 걸쳐 저장된다. 따라서, 개시된 시스템은 KV-SSD가 유연한 밸류 크기를 지원할 수 있기 때문에 KEY 1의 밸류를 블록으로 더 분할할 필요가 없을 수 있다. 또한, KEY 1의 밸류의 2개의 세그먼트는 네임노드(110)에서 특정 KEY 1에 대해 데이터노드가 어떻게 정렬되는지(즉, 제1데이터노드 (115), 제2데이터노드 (120))에 따라 저장될 수 있다.

[0018] 단일 키-밸류 튜플을 여러 개의 내부 키-밸류 쌍으로 분리하여, 분산 시스템 내 스토리지 노드 전반에서 I/O 작업을 병렬화할 수 있다. 다른 클라이언트 키-밸류 튜플에 대해 여러 개의 다른 I/O 작업이 있는 경우, 전체 시스템의 집계 처리량으로 키-밸류 튜플을 통해 스토리지 노드의 전반에서 병렬 처리가 달성될 수 있다. 개시된 시스템은 다수의 키-밸류 튜플을 병렬로 저장함으로써 동작의 스케일링을 구현할 수 있다. 클라이언트의 키-밸류 내부 튜플의 데이터노드 수를 늘림으로써, 단일 키-밸류 튜플의 병렬 처리가 증가한다. 개선된 지연 시간과 함께 병렬 처리를 제공하는 내부-키-밸류 튜플은 큰 밸류에 대하여 또한 달성될 수 있다.

[0019] 도 2b는 도 1의 네이티브 키-밸류 분산 스토리지 시스템(100b)의 예시적인 다이어그램이다. 도 1은 일부 실시예에 따른 상이한 예시적인 세부 사항을 도시한다. 시스템(100b)을 사용하면, 일부 블록 기반 저장 시스템에 의해 요구될 수 있는 바와는 달리, 개시된 시스템이 키-밸류 튜플을 블록에 매핑할 필요가 없을 수 있다. 맵핑에 대한 이러한 감소된 요구는 소프트웨어 계층을 더 간단하게 할 수 있고, 개시된 시스템의 오버 헤드를 줄일 수 있으며, 더 큰 확장성을 가능하게 한다. 도 2b에 도시된 바와 같이, 네임노드(110)는 키 위치 정보(205)를 저장할 수 있다. 키 위치 정보(205)는 데이터노드의 순서화된 목록(예를 들어, 115, 120 등)으로서 저장될 수 있다. 예를 들어, KEYa는 제1 데이터노드(115) 및 제2 데이터노드(120)와 연관될 수 있다. 다른 예로서, KEYb는 단지 제2데이터노드(120)와 연관될 수 있다. 동일한 키는 동일한 키 식별자를 사용하여 데이터노드에 저장될 수 있다. 예를 들어, 제1데이터노드(115)에 도시된 바와 같이, KEYa0은 대응하는 밸류 FOO와 함께 저장될 수 있다. 다른 예로서, 제2데이터노드(120)에 도시된 바와 같이, KEYa1 및 KEYb0은 각각 그것들에 대응하는 밸류 BAR 및 CAT와 함께 저장될 수 있다.

[0020] 몇몇 측면에서, KV-SSD는 스토리지 장치 유닛으로서 사용될 수 있다. 이와 같이, 클라이언트의 키-밸류 튜플(예를 들어, KEYa : FOO, BAR)은 도 2b에 도시된 바와 같이, 다수의 내부 키-밸류 튜플로 분해될 수 있다. 알 수 있는 바와 같이, 키-밸류 튜플(KEYa : FOO, BAR)은 제1데이터노드(115) 및 제2데이터노드(120)를 가로질러 분해 및 스트라이핑 될 수 있다. 다르게 말하면, KEYa0:FOO는 KV-SSD(130)에 저장될 수 있고 KEYa1:BAR은 KV-SSD(135)에 저장될 수 있다. 클라이언트 키-밸류 튜플을 여러 개의 내부 키-밸류 튜플로 나누는 것과 키-밸류 튜플을 여러 개의 고정된-크기 블록으로 나누는 것의 한 가지 차이점은 내부 키-밸류 튜플을 사용하면 그것들의 크기가 동적일 수 있다는 것이다.(예를 들어, 1, 2, N 등) 대조적으로, 종래의 블록-기반 키-밸류 튜플과 함께, 튜플을 저장하는데 필요한 블록의 수는 고정될 수 있고(즉, 키-밸류 크기에 기초하여), 이는 블록 내부의 단편화를 야기할 수 있다.

- [0021] 도 2b에 도시된 예에서, KEYb는 3개의 문자(즉, CAT)를 포함하는 밸류를 가지며 데이터노드(120)에 단일 키-밸류 튜플로서 완전히 저장되는 반면, KEYa는 6개의 문자(즉, FOO BAR)를 포함하는 밸류를 가지지만, 절반은 각각 3개의 문자를 가지며 제1데이터노드(115) 및 제2데이터노드(120)에 걸쳐 저장된다. 따라서, 개시된 시스템은 KV-SSD가 유연한 밸류 크기를 지원할 수 있기 때문에 KEYa의 밸류를 블록으로 더 분할할 필요가 없을 수 있다. 또한, KEYa 밸류의 2개의 세그먼트는 네임노드(110)에서 특정 KEYa에 대해 데이터노드가 어떻게 정렬되는지(즉, 제1데이터노드(115), 제2데이터노드(120))에 따라 저장될 수 있다.
- [0022] 도 2b에 도시된 예시적인 실시예에서, KEYa0, KEYa1, KEYb0 등은 내부 키일 수 있으며, 네임노드(110)는 키 인덱스 번호(즉, 0, 1, 2 등)에 기초하여 메타데이터를 복구할 수 있다. 클라이언트의 키-밸류 튜플을 KV-SSD 상에서 개별 튜플에 매핑할 때, 스토리지 노드 내에서 KV-SSD 상의 각 내부 키-밸류 쌍에 동일한 키가 사용될 수 있다.(즉, 각 데이터노드가 클라이언트 키-밸류 튜플의 일부를 저장할 수 있다). 키-밸류 정보 데이터 구조(예를 들어, KVInfo)에서 데이터노드의 오프셋은 클라이언트의 집계된 키-밸류 튜플에서 그 부분 밸류의 순서를 결정할 수 있다. 그러나, 데이터 검증은 가능하게 하기 위해, 키 인덱스 번호(예를 들어, 0, 1, 2, N 등)가 스토리지 노드 상의 KV-SSD에 저장된 키에 추가되어 집계된 키에서 그것의 위치를 식별하는 것을 도울 수 있다. 데이터 검증을 위해, 분산 스토리지 시스템에 여러 키-밸류 튜플이 있을 것이므로, 이는 키 인덱스 번호가 나타내는 클라이언트 키-밸류 튜플의 어느 부분을 식별하는 방법이다. 또한, 키 인덱스 번호는 시스템이 동일한 데이터노드 상에 둘 이상의 내부 키-밸류 튜플을 넣는 경우에 도움이 된다. 복원력을 높이기 위해, 네임노드(110)가 다운되어 그것의 KV-SSD에 저장된 키-밸류 정보 데이터 구조(예를 들어, KVInfo)를 잃으면 클라이언트(105)는 추가 키 인덱스 번호 정보를 사용할 수 있다. 키-밸류 스토리지 시스템(100b)으로부터의 모든 키는 대응하는 키 인덱스 번호가 나타내는 클라이언트 키-밸류 튜플의 어느 부분을 결정하기 위해 키 인덱스 번호를 사용하여 추출될 수 있다. 따라서, 키 인덱스 번호에 기초하여, 개시된 시스템(예를 들어, 100b)은 필요한 경우 분산 시스템에서 파일 및 디렉토리의 메타데이터를 재구성 할 수 있다. 더욱이, 순환 중복 검사(Cyclic Redundancy Check: CRC) 코드는 밸류가 손상되지 않고 정확한지 확인하기 위해 밸류의 끝에 포함될 수 있다.
- [0023] 도 3은 일부 실시예에 따른 네이티브 키-밸류 분산 스토리지 시스템(300)의 또 다른 예시적인 다이어그램이다. 이 예시적인 실시예에서, 클라이언트(105)는 "FOO", "BAR" 및 "BAT" 밸류를 포함하는 "KEY 1"이라고 명명되는 키를 저장 또는 액세스하게 할 수 있다. 밸류의 서브 섹션(즉, "FOO", "BAR")은 제1데이터노드(115)에 저장될 수 있고, 다른 서브 섹션(즉, "BAT")이 제2데이터노드(120)에 저장될 수 있다. 네임노드(110)는, 클라이언트 키-밸류 튜플을 KV-SSD(예를 들어, 130 및/또는 135)에 저장된 키-밸류 튜플에 매핑하는 키-밸류 매핑 섹션(305)을 포함할 수 있다. 네임노드(110)는 또한 각각의 데이터노드의 총 용량(예를 들어, 315) 및 각 데이터노드의 사용된 공간(예를 들어, 320)을 저장 및/또는 추적할 수 있는 자유 리스트 공간 관리 섹션(310)을 포함할 수 있다.
- [0024] 도 4는 일부 실시예에 따른 네이티브 키-밸류 분산 스토리지 시스템(400)의 또 다른 예시적인 다이어그램이다. 키-밸류 분산 스토리지 시스템(400)은 NVMe(non-volatile memory express), NVMe-oF(non-volatile memory express over fabrics) 등과 같은 네트워크 전송 프로토콜을 사용할 수 있다. 인피니밴드(Infiniband), TCP/IP, 또는 파이버 채널(Fibre Channel) 등과 같은 다른 적합한 네트워크 전송 프로토콜이 사용될 수 있음을 이해될 것이다. 클라이언트(105)는 스토리지 및 네트워킹 스택(405) 및 애플리케이션 계층(410)을 포함할 수 있다. 애플리케이션 계층(410)은 하나 이상의 애플리케이션(415), 커맨드 라인 인터페이스(420) 등을 포함할 수 있다. 하나 이상의 애플리케이션(415), 커맨드 라인 인터페이스(420) 또는 이와 같은 것은, 아래에서 더 묘사되어진, 네이티브 키-밸류를 파라미터로서 사용하여 put(), get(), delete() 등과 같은 특정 기능을 직접 호출할 수 있다. 사용자가 예를 들어 커맨드 라인 인터페이스(420)를 사용하여 'ls'를 입력하면, 디렉토리의 콘텐츠가 나열되고, 이는 KV-SSD에 저장된 키-밸류 튜플 자체를 포함할 수 있다. 하이브리드의 예시적인 실시 예에서, 디렉토리의 콘텐츠는 블록-기반 파일 및 키-밸류 튜플을 모두 포함할 수 있다.
- [0025] 스토리지 및 네트워킹 스택(405)은 분산 데이터 저장 인터페이스(425), 다이렉트 스토리지 및 네트워킹 라이브러리(430), SPDK(Storage Performance Development Kit)(435), 및 RDMA(Remote Direct Memory Access) 엔진(440)을 포함할 수 있다. 애플리케이션 계층(410)은 스토리지 및 네트워킹 스택(405)을 통해 타겟 노드(450)의 KV-SSD(435)에 저장된 키-밸류 튜플과 상호 작용할 수 있다. 일부 실시예에서, 분산 데이터 저장 인터페이스(425) 및 다이렉트 스토리지 및 네트워킹 라이브러리(430)는 JVM(Java virtual machine)(445)과 같은 가상 머신에서 실행된다. 다이렉트 스토리지 및 네트워킹 라이브러리(430)는 JVM(445)으로부터 SPDK(435)의 네이티브 코드까지 인터페이스를 포함할 수 있다. SPDK(435)의 사용자-레벨 라이브러리에 의해 제로 데이터 카피 데이터 전송이 촉진될 수 있으며, SPDK(435)는 클라이언트(105)의 RDMA 엔진(440)과 타겟 노드(450)의 타겟 노드의 RDMA 엔진(445)을 사용하여 다이렉트 RDMA 데이터 전송(460)과 함께 클라이언트(105)와 타겟 노드(450)를 연결

할 수 있다. 그러한 감소된 소프트웨어 스택으로, JVM(455) 내의 데이터 버퍼의 포인터는 타겟 노드(450)로 직접 이동할 수 있으며, 그 결과 전통적인 RDMA IO 데이터 경로를 따라 데이터 카피가 0이 된다. 다시 말해, 데이터 전송은 네이티브 키-밸류 스토리지 및 네트워킹 스택(405)을 통해 키-밸류 튜플과 관련된 데이터를 복사하지 않고, RDMA 데이터 전송을 사용하여 클라이언트(105)와 타겟(450) 사이에서 직접 발생할 수 있다.

[0026] 따라서, 클라이언트의 키-밸류 튜플을 구성하는 각각의 데이터 세그먼트에 대한 하나 이상의 포인터는, 예를 들어, 원격 전송 프로토콜을 사용하여, 스토리지 노드(예를 들어, 450)상의 KV-SSD(예를 들어, 435)로 직접 전송될 수 있다. 일부 실시예들에서, 키-밸류 쌍을 전송한 네트워킹-기반 프로토콜(예를 들어, NVMe-oF, Infiniband®, TCP/IP, 콰이버 채널 및/또는 이와 같은 것들)을 통해 JVM 애플리케이션으로부터 하드웨어로의 데이터 카피 요구 사항이 없을 수 있다.

[0027] 본 명세서에 기술된 일부 실시예는 컨테이너의 유형: 즉, 키-밸류 엔트리를 사용할 수 있으며, 여기서 클라이언트(예를 들어, 105)는, 일반적으로 블록-기반 파일과 연관된 open(), read(), write() close() 인터페이스 보다는, put()/get() 인터페이스를 사용하는 하나 이상의 키-밸류 튜플(예를 들어, KEY 1: VAL)을 생성한다. 키-밸류 스토리지 유형과 함께, 컨테이너는 전통적인 파일과는 다를 수 있으므로, 그것의 내부 콘텐츠가 하나 이상의 키-밸류 정보 데이터 구조(예: KVInfo)이며, 여기서 각 KVInfo 엔트리는 키, 및 해당 키를 저장하는 하나 이상의 데이터노드를 포함한다. KVInfo 당 데이터노드 수는 유연하여, 키당 다수의 데이터노드가 있을 수 있다. 이들 데이터노드에 대한 포인터는 실제로 데이터노드의 순서화된 리스트일 수 있다(예를 들어, 도 2A 및 도 2B의 205). 이는 클라이언트의 원래 밸류를 포함하는 실제 밸류의 파티셔닝을 나타낸다. 대조적으로, 블록-기반 저장 시스템에서, 파일을 저장하는데 사용된 블록의 수는 고정될 수 있다.

[0028] 데이터노드(예를 들어, 450, 115 및/또는 120)는 사용자 키-밸류 튜플의 콘텐츠를 저장할 수 있다. 이는 생성시, 키-밸류 튜플의 크기를 알 수 있고, 튜플의 콘텐츠가 스토리지 노드(즉, 데이터노드)로 전송될 수 있기 때문에, 일반적인 파일보다 RPC가 적은 이점이 있다.

[0029] 일부 실시예들에서, KV-NVMe-oF는 JVM-기반 시스템에서 제로-카피 데이터 변환을 갖는 분산 스토리지 시스템에서 직접적인 클라이언트-측 키-밸류 동작을 제공하기 위해 활용될 수 있다. 예를 들어, put(), get() 및 delete() KV-SSD 동작을 실행하도록 실제 I/O NVMe-oF 명령을 래핑(wrapping)함으로써 KV-NVMe-oF 프로토콜을 Java로 확장할 수 있다. C 프로그래밍 언어-기반 NVMe-oF 응용 프로그래밍 인터페이스(APIs)로부터 Java로의 이러한 변환은, JVM과 하위-레벨 언어간의 포인터만 복사되는 것이 요구되기 때문에, 데이터 복사를 발생시키지 않을 수 있다. 키-밸류 APIs를 클라이언트에 노출시킴으로써, 분산 스토리지 시스템 클라이언트는 클러스터에 상주하는 스토리지 노드의 특정 KV-SSD에 I/O 동작을 직접 발행할 수 있다. 심지어 중간 인터페이스로서 JVM이 없어도, 개시된 시스템의 NVMe-oF 프로토콜은 클라이언트(예를 들어, 105)로부터 데이터노드(예를 들어, 450, 115 및/또는 120)로도 활용될 수 있다.

[0030] 도 5는 일부 실시예에 따른 네이티브 키-밸류 분산 스토리지 시스템의 일부 예시적인 동작의 타이밍 흐름도(500)의 예시적인 다이어그램이다. 데이터노드(예를 들어, 데이터노드 (115))의 초기 상태는, 565에 도시된 바와 같이, 100GB의 용량이며, 예를 들어, 사용된 용량은 0GB 일 수 있다. 타이밍 흐름도(500)에 도시된 기술은 데이터-노드(예를 들어, 데이터노드 (115))가 키-밸류 네이티브 동작 호출을 사용하여 KV-SSD 용량을 네임노드(예를 들어, 110)에 등록할 수 있는 동안에 505에서 시작할 수 있다. 예를 들어, SET_KV_INFO(CAPACITY : 100G)와 같은 동작 호출은 100GB의 용량을 네임노드(예를 들어, 110)에 등록하는데 사용될 수 있다. 510에서, 네임노드는 505에서의 요청을 승인할 수 있다. 515에서, 클라이언트(105)는 키-밸류 네이티브 분산 저장 시스템(예를 들어, 도 1의 100)을 통해 요청을 전송함으로써 키-밸류를 생성할 수 있다. 예를 들어, 클라이언트(105)는 키-밸류 네이티브 동작 호출을 사용하여 특정 크기(예를 들어, 약 1GB)의 새로운 키를 생성할 수 있다. 예를 들어, 클라이언트(105)는 1GB 크기를 갖는 키 K1을 생성할 수 있는 CREATE_KV_FILE (KEY:K1, SIZE:1G)에 대한 동작 호출을 수행할 수 있다. 네임노드(110)는 515에서 전달된 동작 호출 및 파라미터를 수신할 수 있고, 520에 도시된 바와 같이, 데이터노드(예를 들어, 데이터노드 (115))와 함께 클라이언트(105)로 다시 응답할 수 있다. 525에서, 클라이언트(105)는 키-밸류 네이티브 동작 호출을 호출함으로써 새로운 밸류를 키와 연관시킬 수 있다. 예를 들어, 클라이언트(105)는 새로운 밸류 VAL을 키 K1과 연관시킬 수 있는 PUT(K1, VAL)에 대한 동작 호출을 수행할 수 있다. 데이터노드(예를 들어, 데이터노드 (115))는 525에서 전달된 동작 호출 및 파라미터를 수신할 수 있고, 530에서 요청을 승인할 수 있다. 535에서, 클라이언트는 확인 풋 메시지(535)를 네임노드(110)에 전송할 수 있다. 540에서, 네임노드(110)는 클라이언트(105)에 의한 풋(put)이 완료된 후, 570에 도시된 바와 같이, 대응하는 밸류 크기만큼 가용 용량을 감소시킬 수 있다.

- [0031] 클라이언트(105)는 키와 연관된 값에 액세스하기 위해 갱(get) 동작을 계속할 수 있다. 예를 들어, 545에서, 클라이언트(105)는 키-밸류 네이티브 동작을 사용할 수 있으며, 키-밸류 네이티브 동작은 네임노드(110)와 인터페이스할 수 있다. 예를 들어, 클라이언트(105)는 키-밸류 파일을 검사할 수 있는 CHECK_KV_FILE()에 대한 동작 호출을 수행할 수 있다. 네임노드(110)는 550에 도시된 바와 같이, 데이터노드(예를 들어, 데이터노드(115))를 클라이언트(105)로 리턴할 수 있다. 그런 다음, 555에서, 클라이언트(105)는 데이터노드(예를 들어, 데이터노드(115))로부터 밸류를 얻기 위해 키-밸류 네이티브 동작을 사용할 수 있다. 예를 들어, 클라이언트(105)는 키 K1을 얻을 수 있는 GET(K1)에 대한 동작 호출을 수행할 수 있다. 560에서, 데이터노드(예를 들어, 데이터노드(115))는 키 K1과 연관된 값을 리턴할 수 있다.
- [0032] KV-SSD(예를 들어, 130 및/또는 135)를 사용함으로써, 시스템 설계는 시스템의 하단부터 위로 및 시스템 전체에서 블록의 개념을 감소시키거나 제거할 수 있기 때문에, 블록 관리가 감소될 수 있다. 네임노드(예를 들어, 110)에 대한 하나의 예시적인 요구 사항은 스토리지 노드(예를 들어, 115 및/또는 120)상의 개별 KV-SSD(예를 들어, 130 및/또는 135)의 용량을 추적하는 것일 수 있다. 따라서, 도 5에 도시된 바와 같이, 데이터노드(예를 들어, 115 및/또는 120)가 시작될 때, 데이터노드의 용량을 네임노드에 등록할 수 있다. 그런 다음, 네임노드(예를 들어, 110)는 이 용량을 KV-SSD 단위 별로 관리한다. 네임노드(예를 들어, 110)가 특정 크기의 키-값 튜플을 저장하라는 요청을 수신하면, 어떤 데이터노드(예 : 115 및/또는 120)가 그 클라이언트의 키-밸류 튜플을 저장해야 하는지에 관한 결정을 하기 위해 네임노드(예를 들어, 110)는 KV-SSD의 맵과 KV-SSD의 가용 용량을 참고할 수 있다. 결정은 정책 기반일 수 있으며, 개시된 시스템은 예를 들어, 라운드-로빈과 같은 다수의 정책을 사용할 수 있다. 클라이언트(예를 들어, 105)가 특정 데이터노드 상에 키-밸류 튜플을 저장하게 한 후, 네임노드는 해당 데이터노드의 가용 용량을 해당 밸류의 크기만큼 감소시킬 수 있다.
- [0033] 도 6은 일부 실시예에 따른 블록 기반 및 네이티브 키-밸류 스토리지의 하이브리드 분산 저장 시스템 (600)의 예시적인 다이어그램이다. 이 예시적인 실시예에서, 분산 스토리지 시스템(600)은 기본적으로 블록-기반 파일 및 네이티브 키-밸류- 기반 튜플을 모두 처리할 수 있다. 달리 말하면, 하이브리드 분산 스토리지 시스템(600)은 블록-기반 및 네이티브 KV-SSD를 동시에 지원할 수 있다. 예를 들어, 클라이언트(105)는 하나 이상의 애플리케이션(415) 및 명령행 인터페이스(420)를 포함할 수 있으며, 이들 중 어느 하나 또는 둘은 분산 데이터 저장 인터페이스(425)와 인터페이스할 수 있다. 네이티브 키-밸류 연산 또는 명령이 키-밸류 튜플을 생성, 액세스 또는 삭제하는데 사용될 때, 실행 경로는 네이티브 키-밸류 분산 저장 시스템 인터페이스 (605)를 통해 흐르며, 그리고 네임노드(110) 및 다양한 데이터노드로 흐른다(예를 들어, 데이터노드(115) 및/또는 데이터노드(120)). 그렇지 않으면, 블록-기반 동작 또는 명령이 블록-기반 파일을 생성, 액세스 또는 삭제하는 데 사용될 때, 실행 경로는 블록-기반 분산 스토리지 시스템 인터페이스(610)를 통해 흐르고, 그리고 종래의 블록-기반 SSD(620)를 가질 수 있는 블록-기반 데이터노드(예를 들어, 데이터노드(615))로 흐른다. 따라서, 시스템(600)에 도시된 바와 같은 하이브리드 실시예에서, 일반적인 파일(즉, 고정 블록에 의해 뒷받침 되는 파일)은 동일한 파일 시스템 네임 스페이스에서 키-밸류 튜플과 함께 (즉, KV-SSD에 저장된 키-밸류 쌍에 의해 뒷받침되는 키 밸류-튜플) 결합될 수 있다.
- [0034] 블록-기반 분산 스토리지 시스템 인터페이스(610) 및/또는-블록 기반 데이터노드(615)는 블록-기반 메타데이터 노드를 포함할 수 있다. 키-밸류 튜플에 대한 메타데이터는 비-하이브리드 시스템에서 전송한 것과 같이 하이브리드 블록-기반 네이티브 키-밸류 분산 스토리지 시스템(600)에서 동일할 수 있다. 하이브리드 시스템(600)의 한 가지 차이점은 블록-기반 파일 시스템 메타데이터가 확장될 수 있다는 것이며, 이는 블록-기반 아키텍처를 사용하는 파일 및 디렉토리의 콘텐츠를 저장하는 것이 또한 블록-기반이 아닌 다른 유형의 엔티티인 키-밸류 튜플을 저장할 수 있는 키-밸류 튜플을 저장하는 것과 같다.
- [0035] 도 7은 일부 실시예들에 따라 분산 저장 시스템에서 네이티브 키-밸류 튜플 요청 및 블록-기반 파일 요청의 하이브리드 조합을 처리하기 위한 기술을 도시한 흐름도이다. 흐름도는 클라이언트가 블록-기반 분산 시스템 인터페이스를 호출할 때 702를 따르고, 클라이언트가 네이티브 키-밸류 분산 인터페이스를 호출할 때 704를 따를 수 있다.
- [0036] 705에서, 클라이언트는 분산 스토리지 시스템의 네이티브 키-밸류 인터페이스를 사용하여 새로운 키-밸류 튜플을 생성하거나 생성되도록 유발할 수 있다. 715에서, 클라이언트는 put() 동작을 호출하여 하나 이상의 KV-SSD에 특정 키에 대한 밸류를 저장할 수 있다. 725에서, 클라이언트는 하나 이상의 KV-SSD로부터 특정 키와 연관된 밸류를 얻기 위해 get() 동작을 호출할 수 있다. 735에서, 클라이언트는 하나 이상의 KV-SSD로부터 키-밸류 튜플을 삭제하기 위해 delete() 동작을 호출할 수 있다.

- [0037] 710에서, 클라이언트는 분산 스토리지 시스템의 블록-기반 인터페이스를 사용하여 새로운 블록-기반 파일을 생성하거나 생성되도록 유발할 수 있다. 720에서, 클라이언트는 put() 동작을 호출하여 하나 이상의 SSD 상의 특정 블록-기반 파일에 데이터를 저장할 수 있다. 730에서, 클라이언트는 하나 이상의 SSD로부터 특정 블록-기반 파일과 연관된 데이터를 얻기 위해 get() 동작을 호출할 수 있다. 740에서, 클라이언트는 하나 이상의 SSD로부터 블록-기반 파일을 삭제하기 위해 delete() 동작을 호출할 수 있다. 도 7의 단계들은 표시된 순서대로 수행할 필요가 없는 것으로 이해된다. 예를 들어, 단계들은 상이한 순서로 수행될 수 있고 및/또는 도시되지 않은 중간에 삽입되는 단계를 가질 수 있다.
- [0038] 도 8은 본 명세서에 개시된 바와 같은 일부 실시예들에 따른 하나 이상의 클라이언트(예를 들어, 805, 810 및/또는 815), 네임노드(110), 하나 이상의 네이티브 키-밸류 데이터노드(예를 들어, 115 및/또는 120), 및 하나 이상의 블록-기반 데이터노드(예를 들어, 825)를 포함하는 하이브리드 키-밸류 분산 스토리지 컴퓨팅 시스템(800)의 블록 다이어그램의 예시이다. 하나 이상의 클라이언트(예를 들어, 805, 810 및/또는 815)는 네임노드(110), 네이티브 키-밸류 데이터노드(예를 들어, 115 및/또는 120), 및 블록-기반 데이터노드(825)에 클라우드(820)를 통해 통신적으로 연결될 수 있다. 하이브리드 키-밸류 분산 저장 컴퓨팅 시스템(800)은 키-밸류 튜플(예를 들어, 840) 및 블록-기반 데이터(예를 들어, 850)를 전송, 수신 및 저장할 수 있다. 분산 스토리지 컴퓨팅 시스템(800)은 하나 이상의 클라이언트(예를 들어, 805, 810 및/또는 815)상의 소프트웨어 스택 전체에 걸쳐 네이티브 키-밸류 동작을 지원할 수 있고, 네이티브 KV-SSD(예를 들어, 130 및/또는 135)에 있는 키-밸류 튜플(예를 들어, 840)을 저장할 수 있다.
- [0039] 이제 도 1 내지 8을 참조한다.
- [0040] 본 명세서에 개시된 시스템 및 방법의 실시예들은 분산 스토리지 시스템에서 입력/출력(I/O) 경로 동작을 수정하여 KV-SSD로 직접 이동함으로써, 로컬 스토리지 노드에 상주하는 로컬 파일 시스템을 우회한다. 개시된 시스템은 마스터/슬레이브 분산 파일 시스템의 기존 아키텍처에 KV-SSD의 네이티브 통합을 가능하게 한다. 본 명세서에 기술된 다양한 실시예들은 네이티브 KV-SSD를 활용함으로써 분산 스토리지 시스템의 전체 아키텍처에 걸쳐 키-밸류 클라이언트 인터페이스의 네이티브 지원을 포함하여, 키-밸류 인터페이스를 분산 시스템의 기본 블록 스토리지에 매핑해야 하는 비효율성을 피한다.
- [0041] 본 명세서에 개시된 일부 실시예들은 데이터 구조(예를 들어, 도 2A 및 도 2B의 205)를 사용할 수 있으며, 이는 하나 이상의 스토리지 노드(예를 들어, 115 및/또는 120)에 저장된, 하나 이상의 내부 키-밸류 튜플의 순서화된 리스트이며, 여기서 각 내부 키-밸류 튜플이 기본적으로 KV-SSD에 저장된다. 이것은 성능을 확장하고 오버헤드를 최소화하는 유연성을 가능하게 할 수 있다. 예를 들어, 단일 클라이언트 키-밸류 튜플(예를 들어, 도 2a의 KEY 1 및/또는 도 2b의 KEYa)은 다수의 KV-SSD(예를 들어, 130 및/또는 135)에 걸쳐 스트라이프 될 수 있으며, KV-SSD 별로 저장되는 데이터의 양은 블록 크기와 같은 임의의 밸류가 아닌, 스트라이프 계수(striping factor)에 의해 결정될 수 있다.
- [0042] 분산 스토리지 시스템(예 : 100, 100a, 100b, 300 및/또는 400)이 키-밸류 튜플만 저장하는 경우, 즉, 블록-기반 파일이 없는 경우, 파일들 및 관련 디렉토리들을 관리하기 위해 기존 블록-기반 파일 시스템 메타데이터 시스템을 사용할 필요성이 줄어들거나 필요하지 않을 수 있다. 대신에, 본 명세서에 개시된 실시예들은 키-밸류 메타데이터 시스템 (예를 들어, 100, 100a, 100b, 300 및/또는 400)을 사용하여 클라이언트 키-밸류 튜플의 위치를 관리하기 위한 시스템 및 방법을 제공하며, 이는 또한 네임노드(예를 들어, 110) 상에 네이티브 KV-SSD(예를 들어, 115 및/또는 120)로 저장될 수 있다. 특정 클라이언트 키(예를 들어, KEY 1)가 주어지면, 이 키-밸류 메타데이터 시스템은 클라이언트의 특정 키(예를 들어, 키 1)에 대한 실제 밸류를 저장하는 스토리지 노드 또는 노드(예를 들어, 130 및/또는 135)를 찾는 데 사용될 수 있다.
- [0043] 따라서, 키-밸류 튜플은 JVM 기반 분산 시스템을 통해 제로-카피 데이터 전송을 사용하는 KV-SSD(예를 들어, 130 및/또는 135)를 사용하여 클라이언트 (예를 들어, 105)로부터 스토리지 노드(예를 들어, 130 및/또는 135)로 원격으로 전송될 수 있다. NVMe-oF 프로토콜을 사용하는 기존의 데이터 IO 접근 방식은 단지 대상 장치(예 : SSD)에서 고정된 크기 및 미리 결정된 어드레스를 가진 블록에서 데이터 전송만을 지원한다. 키-밸류 쌍을 원격으로 전송하기 위해, NVMe-oF 프로토콜 (또는 유사한 프로토콜)은 고정된 블록 어드레스 및 길이에 추가하여 임의의 크기의 키 및 밸류를 포함하도록 본 명세서에 개시된 바와 같이 확장된다. 클라이언트 키-밸류 튜플(예를 들어, KEY 1 : FOO, BAR)은 네임노드에서 생성된 메타데이터(예를 들어, 305 및/또는 310)와 함께, 클라이언트의 키-밸류 튜플(예를 들어, KEY 1 : FOO, BAR)의 전부 또는 일부를 저장하는 하나 이상의 특정 데이터노드(예를 들어, 115 및/또는 120)와 함께 JVM 애플리케이션에 의해 생성될 수 있다. KV-SSD(예를 들어, 130 및/또는

135)를 사용하여, 키-밸류 튜플의 블록으로의 매핑을 유지할 필요는 없다. 대신, 클라이언트 키-밸류 튜플을 추가 메타데이터 없이 스토리지 노드의 KV-SSD에 직접 저장할 수 있다.

[0044] 본 명세서에 개시된 실시예들의 하나의 장점은 클라이언트로부터 스토리지 장치까지의 키-밸류 튜플이 고유하게 통합되어, 간접 매핑 계층의 수를 감소시킨다는 것이다. 네임노드의 개별 블록 관리의 오버 헤드가 최소화된다. KV-SSD를 사용함으로써, 네임노드를 개별 블록 정보에 저장할 필요가 없다. 네임노드는 각 튜플이 있는 위치(즉, 데이터노드)의 저장만을 담당하며, 이는 큰 분산 파일 시스템을 유지하기 위해 더 적은 메타데이터를 요구할 수 있다. 또한, KV-SSD는 데이터노드에 대한 키-밸류 튜플을 지속적으로 저장하기 때문에, 데이터노드에 로컬 파일 시스템이 필요하지 않을 수 있다. 또한, 블록 크기가 제거될 수 있으며, 이는 블록을 갖는 아티팩트(artifact) 및 분산 스토리지 시스템에서 각 블록을 관리해야 하는 네임노드의 오버헤드일 수 있다. 네이티브 키-밸류 저장소를 사용하면, 시스템이 위에서 설명한 하이브리드 시스템이 아닌 한, 블록이 필요하지 않을 수 있다. 다른 장점은 개시된 시스템에서 더 적은 수의 원격 프로시저 호출(Remote Procedure Calls: RPC)이 필요하다는 것이다. 스토리지 시스템에는 어떠한 블록도 필요하지 않기 때문에, 그리고 KV-SSD는 애플리케이션에 대한 전체 키-밸류 튜플을 저장할 수 있으므로, 필요한 RPC가 더 적다. 본 명세서에 개시된 실시예들은 클라이언트 키-밸류 튜플에 대한 네임노드 상의 메타데이터를 관리하기 위해 KV-SSD를 사용한다. 이는 전통적인 블록-기반 스토리지가 파일 시스템 네임스페이스를 관리하는 데 사용될 필요가 없다는 장점을 갖는다. 즉, 클라이언트 키의 밸류 콘텐츠는 메타데이터를 나타내며, 그리고 그 자체는 네임노드에서 KV-SSD에 키-밸류 튜플로서 저장된다. 이는 블록-기반 아키텍처에 대한 종속성이 제거할 수 있다.

[0045] 본 명세서에 개시된 일부 실시예들은 네이티브 키-밸류 분산 저장 시스템을 포함한다. 시스템은 KV-SSD를 포함하는 네임노드를 포함할 수 있다. 시스템은 각각 하나 이상의 KV-SSD를 포함하는 하나 이상의 데이터노드를 포함할 수 있다. 상기 시스템은 상기 네임노드 및 상기 하나 이상의 데이터노드에 통신 가능하게 연결된 클라이언트 장치를 포함할 수 있다. 상기 클라이언트 장치는 네이티브 키-밸류 스토리지 및 네트워크 스택을 포함할 수 있다.

[0046] 일부 실시예들에서, 클라이언트 장치는 네이티브 키-밸류 스토리지 및 네트워크 스택에 통신 가능하게 결합될 수 있는 애플리케이션 계층을 포함한다. 일부 실시예들에서, 클라이언트 장치는 오퍼레이션에 대한 호출에 응답하여 키-밸류 튜플이 생성되도록 구성될 수 있다. 일부 실시예들에서, 동작에 대한 호출은 클라이언트의 애플리케이션 계층 내에서 이루어질 수 있다. 일부 실시예들에서, 키-밸류 튜플은 하나 이상의 데이터노드의 하나 이상의 KV-SSD에 저장될 수 있다.

[0047] 일부 실시예들에서, 키-밸류 튜플은 하나 이상의 KV-SSD 중 제1 KV-SSD 및 하나 이상의 KV-SSD 중 제2 KV-SSD에 걸쳐 스트라이프될 수 있다. 일부 실시예들에서, 스트라이프 되는 키-밸류 튜플은 제1 KV-SSD에 저장된 제1 내부 키-밸류 튜플 및 제2 KV-SSD에 저장된 제2 내부 키-밸류 튜플을 포함할 수 있다. 일부 실시예들에서, 오퍼레이션에 대한 호출은 제1 오퍼레이션에 대한 제1 호출일 수 있다. 일부 실시예에서, 클라이언트 장치는 제2 동작에 대한 제2 호출에 응답하여 키-밸류 튜플에 액세스되도록 구성될 수 있다. 일부 실시예들에서, 제2 동작에 대한 제2 호출은 애플리케이션 계층 내에서 이루어질 수 있다.

[0048] 일부 실시예들에서, 제2 동작은 네이티브 키-밸류 get() 동작 또는 네이티브 키-밸류 put() 동작 중 적어도 하나일 수 있다. 일부 실시예에서, 클라이언트 장치는 제3 동작에 대한 제3 호출에 응답하여 키-밸류 튜플이 삭제되도록 구성된다. 제3 동작에 대한 제3 호출은 애플리케이션 계층 내에서 이루어질 수 있다. 일부 실시예들에서, 네임노드는 데이터노드의 순서화된 목록을 포함하는 키-밸류 매핑 섹션을 포함한다. 일부 실시예들에서, 데이터노드의 순서화된 목록은 하나 이상의 데이터노드를 포함할 수 있다. 일부 실시예들에서, 순서화된 목록은 제1 순서화된 목록일 수 있다. 일부 실시예들에서, 제1 순서화된 리스트는 제1 키-밸류 튜플과 연관될 수 있다. 일부 실시예들에서, 키-밸류 매핑 섹션은 데이터노드의 제2 순서화된 목록을 포함할 수 있다. 일부 실시예들에서, 제2 순서화된 목록은 제2 키-밸류 튜플과 연관될 수 있다.

[0049] 일부 실시예들에서, 네임노드는 하나 이상의 데이터노드의 각각의 총 용량을 저장하거나 추적하도록 구성된 공간 관리 섹션을 포함한다. 일부 실시예들에서, 공간 관리 섹션은 하나 이상의 데이터노드 각각의 사용된 공간을 추적하도록 구성될 수 있다.

[0050] 일부 실시예들에서, 클라이언트 장치는 네임노드와 협력하여 하나 이상의 데이터노드 상에 저장된 복수의 키-밸류 튜플과 연관된 생성 동작, 액세스 동작 또는 삭제 동작 중 적어도 하나를 기본적으로 수행하도록 구성될 수 있다. 일부 실시예들에서, 네임노드의 KV-SSD는 하나 이상의 데이터노드의 하나 이상의 KV-SSD와 동일할 수 있다. 일부 실시예에서, 클라이언트 장치는 원격 직접 메모리 액세스(Remote Direct Memory Access: RDMA) 엔진

을 포함할 수 있다. 일부 실시예들에서, 하나 이상의 데이터노드는 각각 원격 직접 메모리 액세스(RDMA) 엔진을 포함할 수 있다. 일부 실시예에서, 클라이언트 장치는, 네이티브 키-밸류 스토리지 및 네트워킹 스택을 통한 키-밸류 튜플과 연관된 데이터를 복사하지 않고, 클라이언트 장치의 RDMA 엔진 및 하나 이상의 데이터노드의 RDMA 엔진을 사용하는 키 밸류 튜플과 연관된 데이터의 다이렉트 데이터 전송을 야기하도록 구성될 수 있다.

[0051] 본 명세서에 개시된 일부 실시예들은 하이브리드 블록-기반 네이티브 키-밸류 분산 저장 시스템을 포함할 수 있다. 상기 하이브리드 시스템은 KV-SSD를 포함하는 네임노드, 각각 하나 이상의 KV-SSD를 포함하는 하나 이상의 제1 데이터노드 및 각각 하나 이상의 블록-기반 SSD를 포함하는 하나 이상의 제2 데이터노드를 포함할 수 있다. 하이브리드 시스템은 네임노드, 하나 이상의 제1 데이터노드 및 하나 이상의 제2 데이터노드에 통신 가능하게 연결된 클라이언트 장치를 포함할 수 있다. 상기 클라이언트 장치는 하이브리드 블록-기반 네이티브 키-밸류 스토리지 및 네트워킹 스택을 포함할 수 있다.

[0052] 일부 실시예들에서, 클라이언트 장치는 하이브리드 블록-기반 네이티브 키-밸류 스토리지 및 네트워킹 스택에 통신 가능하게 연결된 애플리케이션 계층을 포함할 수 있다. 일부 실시예들에서, 클라이언트 장치는 제1 동작에 대한 호출에 응답하여 키-밸류 튜플이 생성되도록 구성된다. 일부 실시예들에서, 제1 동작에 대한 호출은 애플리케이션 계층 내에서 이루어진다. 일부 실시예들에서, 키-밸류 튜플은 하나 이상의 제1 데이터노드의 하나 이상의 KV-SSD에 저장될 수 있다. 일부 실시예들에서, 클라이언트 장치는 제2 동작에 대한 제2 호출에 응답하여 블록-기반 파일을 생성하도록 구성될 수 있다. 일부 실시예들에서, 제2 동작에 대한 제2 호출은 애플리케이션 계층 내에서 이루어진다. 일부 실시예에서, 블록-기반 파일은 하나 이상의 제2 데이터노드의 하나 이상의 SSD에 저장된다.

[0053] 일부 실시예들에서, 클라이언트 장치는 제3 동작에 대한 제3 호출에 응답하여 키-밸류 튜플에 액세스되도록 구성된다. 일부 실시예들에서, 제3 동작에 대한 제3 호출은 애플리케이션 계층 내에서 이루어질 수 있다. 일부 실시예들에서, 클라이언트 장치는 제4 동작에 대한 제4 호출에 응답하여 블록-기반 파일에 액세스하도록 구성될 수 있다. 일부 실시예들에서, 제4 동작에 대한 제4 호출은 애플리케이션 계층 내에서 이루어질 수 있다.

[0054] 일부 실시예들에서, 클라이언트 장치는 제5 동작에 대한 제5 호출에 응답하여 키-밸류 튜플이 삭제되도록 구성될 수 있다. 일부 실시예들에서, 제5 동작에 대한 제5 호출은 애플리케이션 계층 내에서 이루어질 수 있다. 일부 실시예에서, 클라이언트 장치는 제6 동작에 대한 제6 호출에 응답하여 블록-기반 파일을 삭제하도록 구성될 수 있다. 일부 실시예들에서, 제6 동작에 대한 제6 호출은 애플리케이션 계층 내에서 이루어질 수 있다.

[0055] 진술한 방법의 다양한 동작은 다양한 하드웨어 및/또는 소프트웨어 구성 요소(들), 회로 및/또는 모듈(들)과 같은 동작을 수행할 수 있는 임의의 적절한 수단에 의해 수행될 수 있다.

[0056] 본 명세서에 개시된 실시예와 관련하여 설명된 방법 또는 알고리즘 및 기능의 블록 또는 단계는 하드웨어, 프로세서에 의해 실행되는 소프트웨어 모듈, 또는 이 둘의 조합으로 직접 구현될 수 있다. 소프트웨어에서 구현되는 경우, 기능들은 유형의 비-일시적 컴퓨터-판독가능 매체 상에 하나 이상의 명령어 또는 코드로서 저장되거나 이를 통해 전송될 수 있다. 소프트웨어 모듈은 RAM(Random Access Memory), 플래시 메모리, ROM(Read Only Memory), EPROM(Electrically Programmable ROM), EEPROM(Electrically Erasable Programmable ROM), 레지스터, 하드 디스크, 이동식 디스크, CD ROM 또는 당업계에 공지된 임의의 다른 형태의 저장 매체에 있을 수 있다.

[0057] 다음의 논의는 본 발명의 개념의 특정 측면이 구현될 수 있는 적합한 기계에 대한 간략하고 일반적인 설명을 제공하기 위한 것이다. 일반적으로, 머신은 프로세서, 메모리, 예를 들어 RAM, ROM 또는 기타 상태 보존 매체, 스토리지 장치, 비디오 인터페이스 및 입/출력 인터페이스 포트가 부착된 시스템 버스를 포함한다. 기계 또는 기계들은 적어도 부분적으로 키보드, 마우스 등과 같은 기존의 입력 장치로부터 입력에 의해 제어될 수 있고, 또한, 다른 기계로부터 수신한 지시문, 가상 현실(VR) 환경과의 상호작용, 생체 피드백 또는 기타 입력 신호를 통해 제어될 수 있다. 본 명세서에서 사용되는 용어 "기계"는 단일 기계, 가상 기계, 또는 통신적으로 결합된 기계, 가상 기계 또는 함께 작동하는 장치의 시스템을 광범위하게 포함하는 것으로 의도된다. 예시적인 기계는 개인용 컴퓨터, 워크 스테이션, 서버, 휴대용 컴퓨터, 핸드 헬드 장치, 전화, 태블릿 등과 같은 컴퓨팅 장치뿐만 아니라 개인 또는 대중 교통과 같은 운송 장치, 예를 들어 자동차, 기차, 택시 등을 포함한다.

[0058] 기계 또는 기계들은 프로그래밍 가능하거나 프로그래밍 불가능한 로직 장치 또는 어레이, ASIC(Application Specific Integrated Circuits), 임베디드 컴퓨터, 스마트 카드 등과 같은 임베디드 컨트롤러를 포함할 수 있다. 기계 또는 기계들은 네트워크 인터페이스, 모뎀 또는 다른 통신 결합을 통한 것과 같은 하나 이상의 원격 머신에 대한 하나 이상의 연결을 이용할 수 있다. 기계는 인트라넷, 인터넷, 근거리 통신망, 광역 통신망 등과

같은 물리적 및/또는 논리적 네트워크를 통해 상호 연결될 수 있다. 당업자는 네트워크 통신이 무선 주파수(RF), 위성, 마이크로웨이브, IEEE(Institute of Electrical and Electronics Engineers) 545.11, Bluetooth®, 광학, 적외선, 케이블, 레이저 등을 포함한 다양한 유선 및/또는 무선 단거리 또는 장거리 캐리어 및 프로토콜을 사용할 수 있는 것을 이해할 수 있다.

[0059] 본 개시의 실시예들은, 기계가 작업을 수행하거나 또는 추상적인 데이터 유형 또는 로우-레벨 하드웨어 컨텍스트를 정의하는 것을 유발하는 기계에 의해 액세스될 때, 기능, 절차, 데이터 구조, 애플리케이션 프로그램 등을 포함하는 관련된 데이터와 함께 또는 이를 참조하여 설명될 수 있다. 관련 데이터는 예를 들어, 휘발성 및/또는 비휘발성 메모리, 예를 들어, RAM, ROM 등, 또는 다른 스토리지 장치 및 하드-드라이브, 플로피-디스크, 광학 스토리지, 테이프, 플래시 메모리, 메모리 스틱, 디지털 비디오 디스크, 생물학적 스토리지 등을 포함하는 관련 저장 매체 내에 저장될 수 있다. 연관된 데이터는 패킷, 시리얼 데이터, 병렬 데이터, 전파되는 신호 등의 형태로, 물리적 및/또는 논리적 네트워크를 포함한 전송 환경을 통해 전달될 수 있고, 압축되거나 암호화된 형식으로 사용될 수 있다. 연관된 데이터는 분산 환경에서 사용될 수 있으며, 기계 액세스를 위해 로컬 및/또는 원격으로 저장될 수 있다.

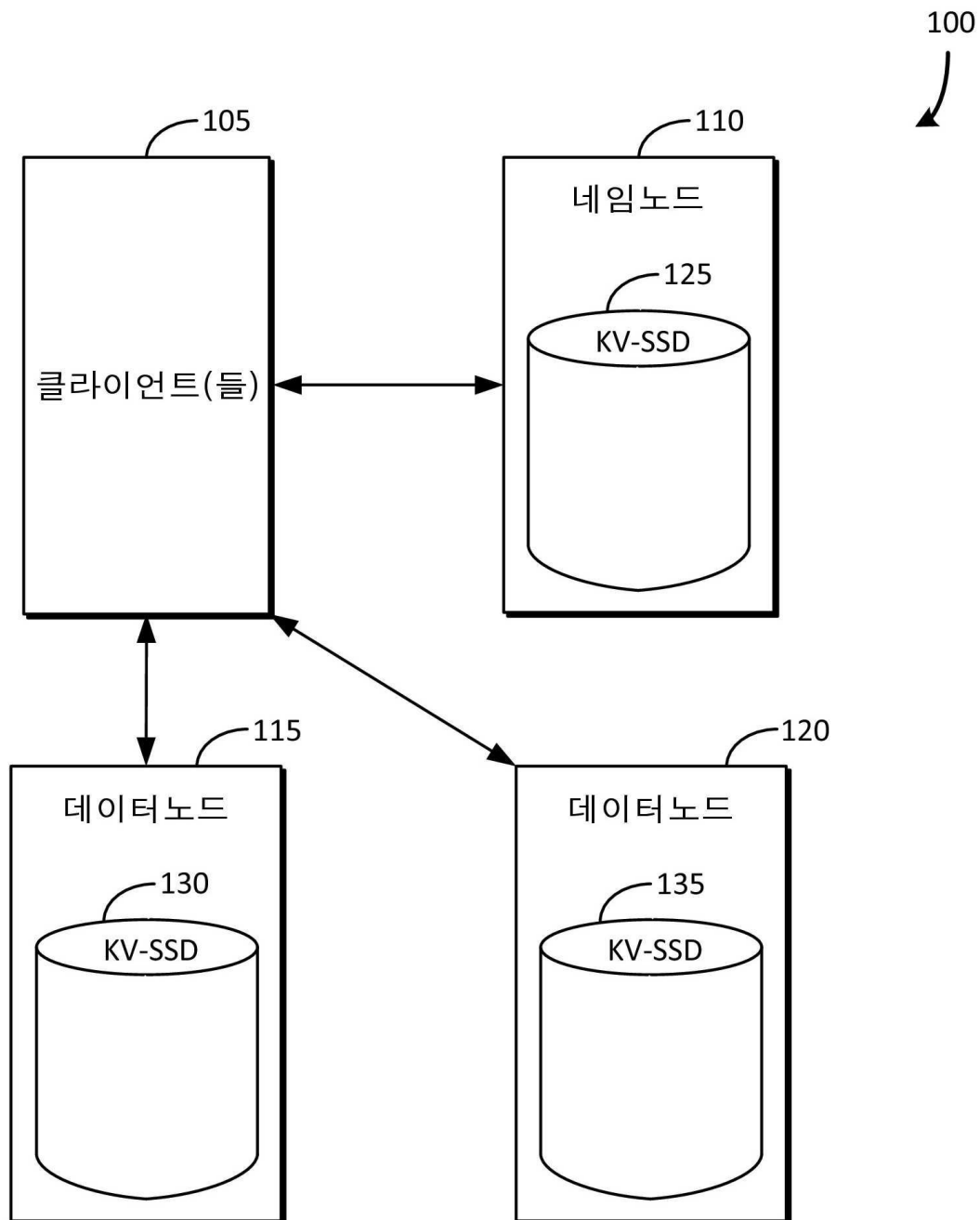
[0060] 예시된 실시예들을 참조하여 본 개시의 원리를 설명하고 예시하였지만, 예시된 실시예들은 그러한 원리를 벗어나지 않고 정연하고 상세하게 수정될 수 있으며, 임의의 원하는 방식으로 조합될 수 있음이 인식될 수 있다. 그리고 전술한 논의는 특정 실시예들에 초점을 맞추었지만, 다른 구성이 고려된다. 특히, "본 발명의 개념의 실시예들에 따른"등과 같은 표현이 본 명세서에서 사용 되더라도, 이들 문구는 일반적으로 실시예의 가능성을 참조하기 위한 것이며, 본 발명의 개념을 특정한 실시예의 구성으로 제한하려는 것은 아니다. 본 명세서에서 사용된 바와 같이, 이들 용어는 다른 실시예에 결합 가능한 동일하거나 상이한 실시예를 지칭할 수 있다.

[0061] 본 개시의 실시예들은 하나 이상의 프로세서에 의해 실행 가능한 명령어를 포함하는 비-일시적 머신 판독가능 매체를 포함할 수 있으며, 명령어는 여기에 설명된 바와 같은 본 발명의 개념의 요소를 수행하기 위한 명령어를 포함한다.

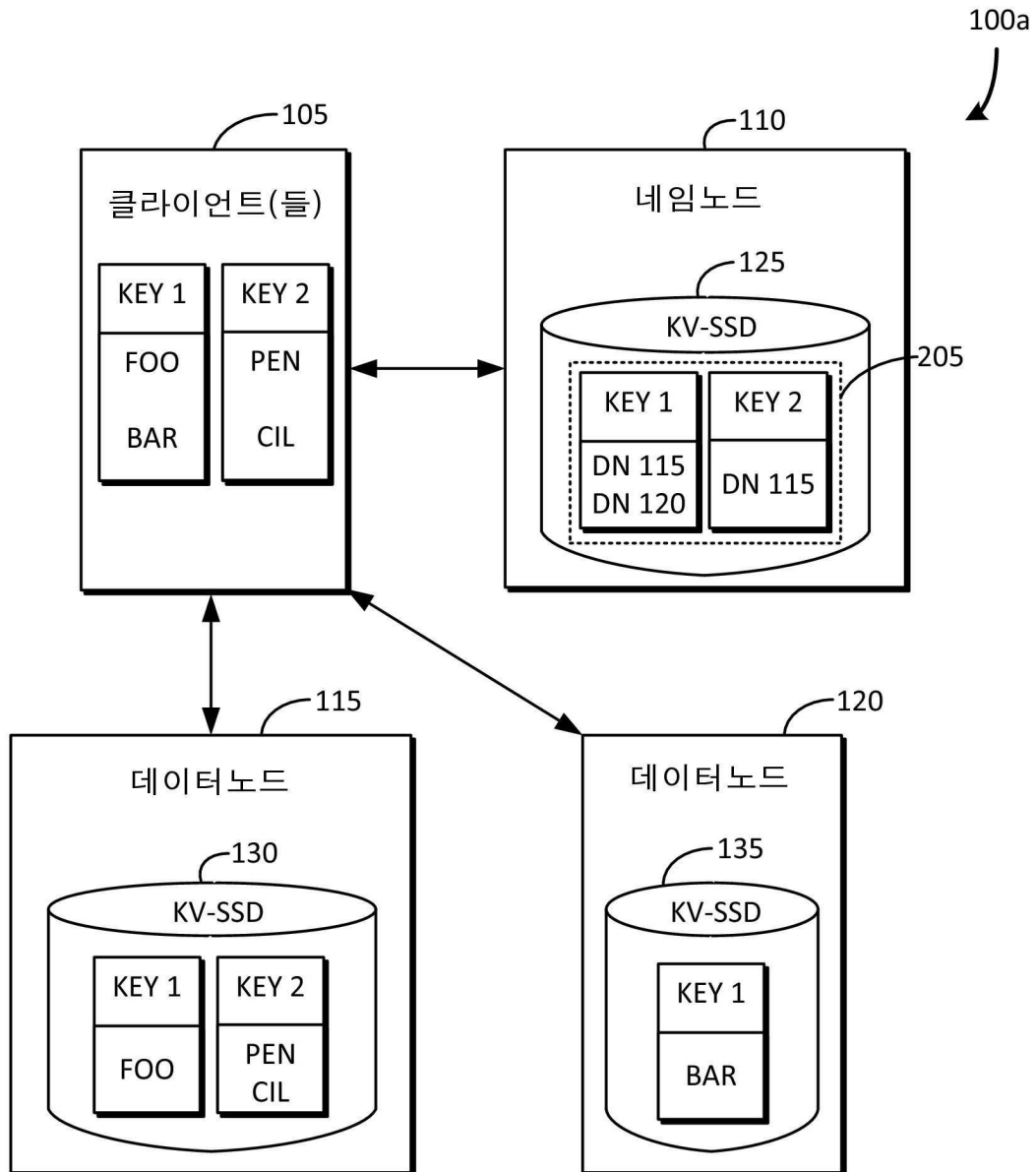
[0062] 전술한 예시적인 실시예는 본 발명의 개념을 제한하는 것으로 해석되지 않아야 한다. 비록 몇몇 실시예들이 설명되었지만, 당업자는 본 발명의 신규한 교시 및 장점을 실질적으로 벗어나지 않으면서 이들 실시예들에 대한 많은 수정이 가능하다는 것을 쉽게 이해할 것이다. 따라서, 그러한 모든 수정은 청구 범위에 정의된 바와 같이 본 발명의 범위 내에 포함되도록 의도된다.

도면

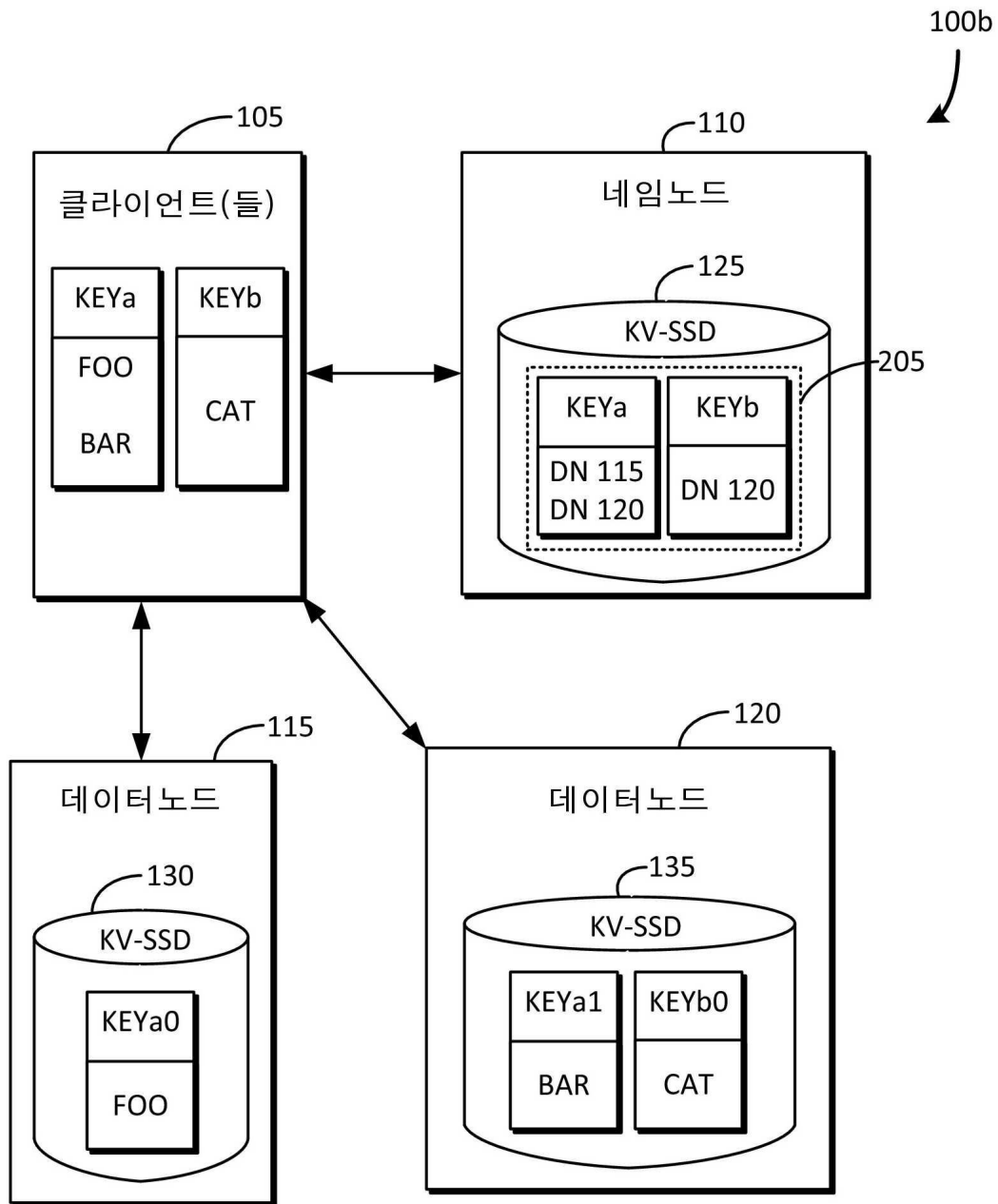
도면1



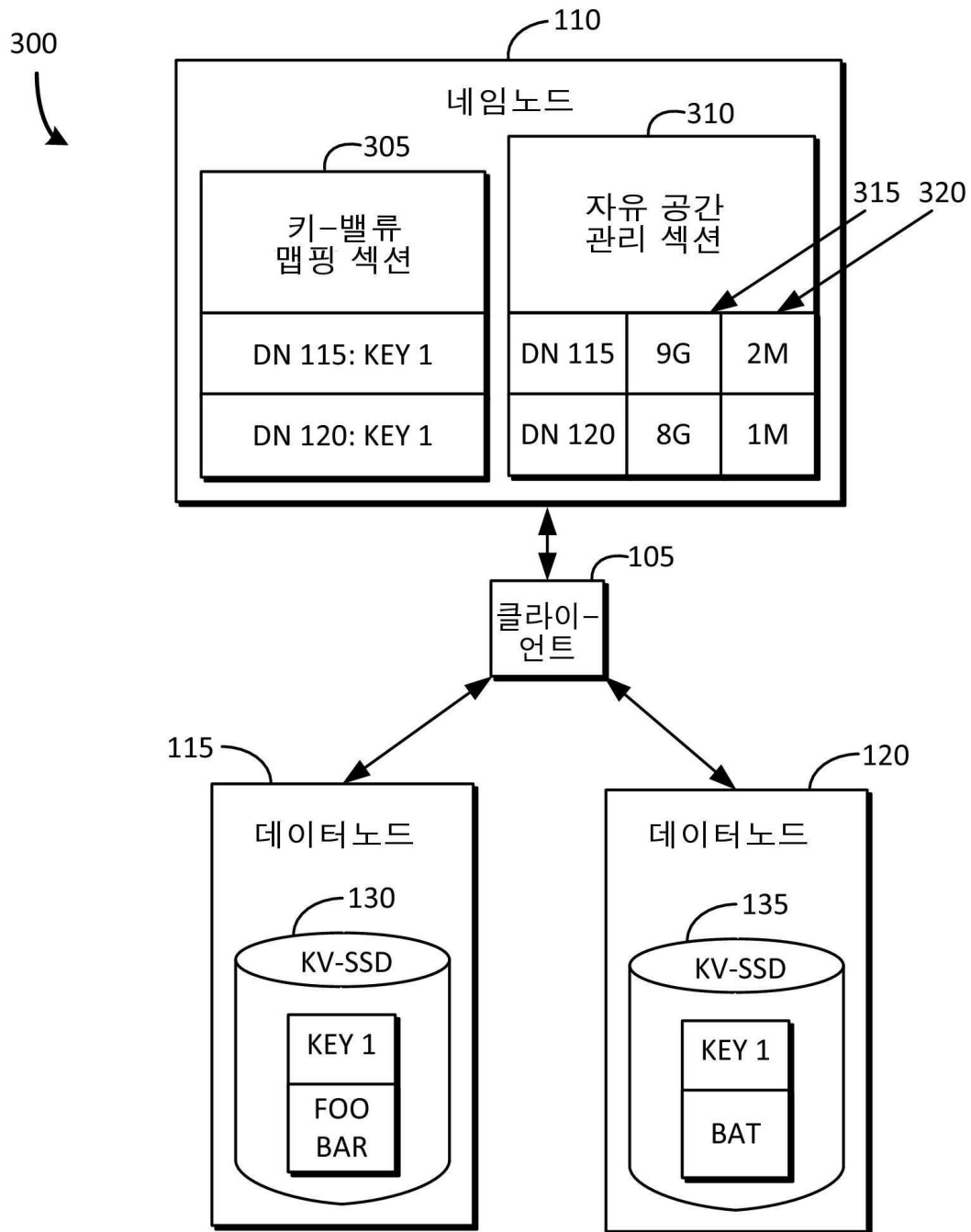
도면2a



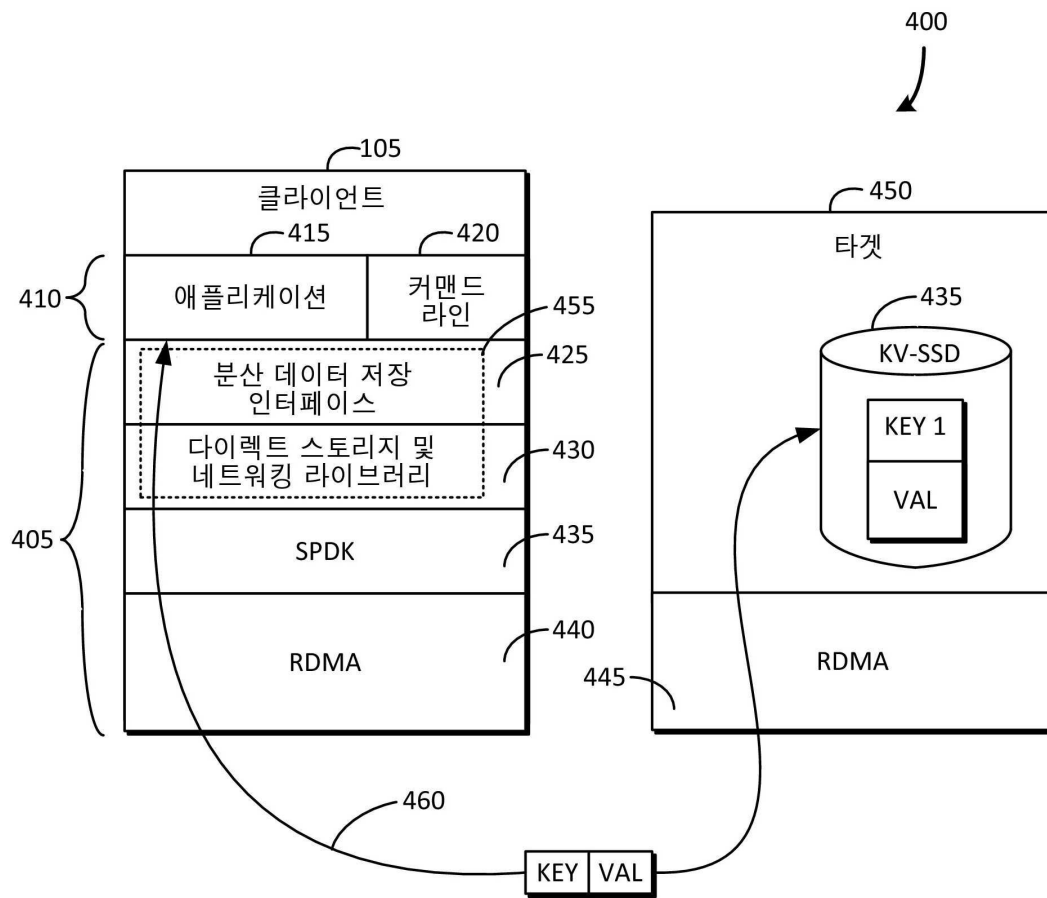
도면2b



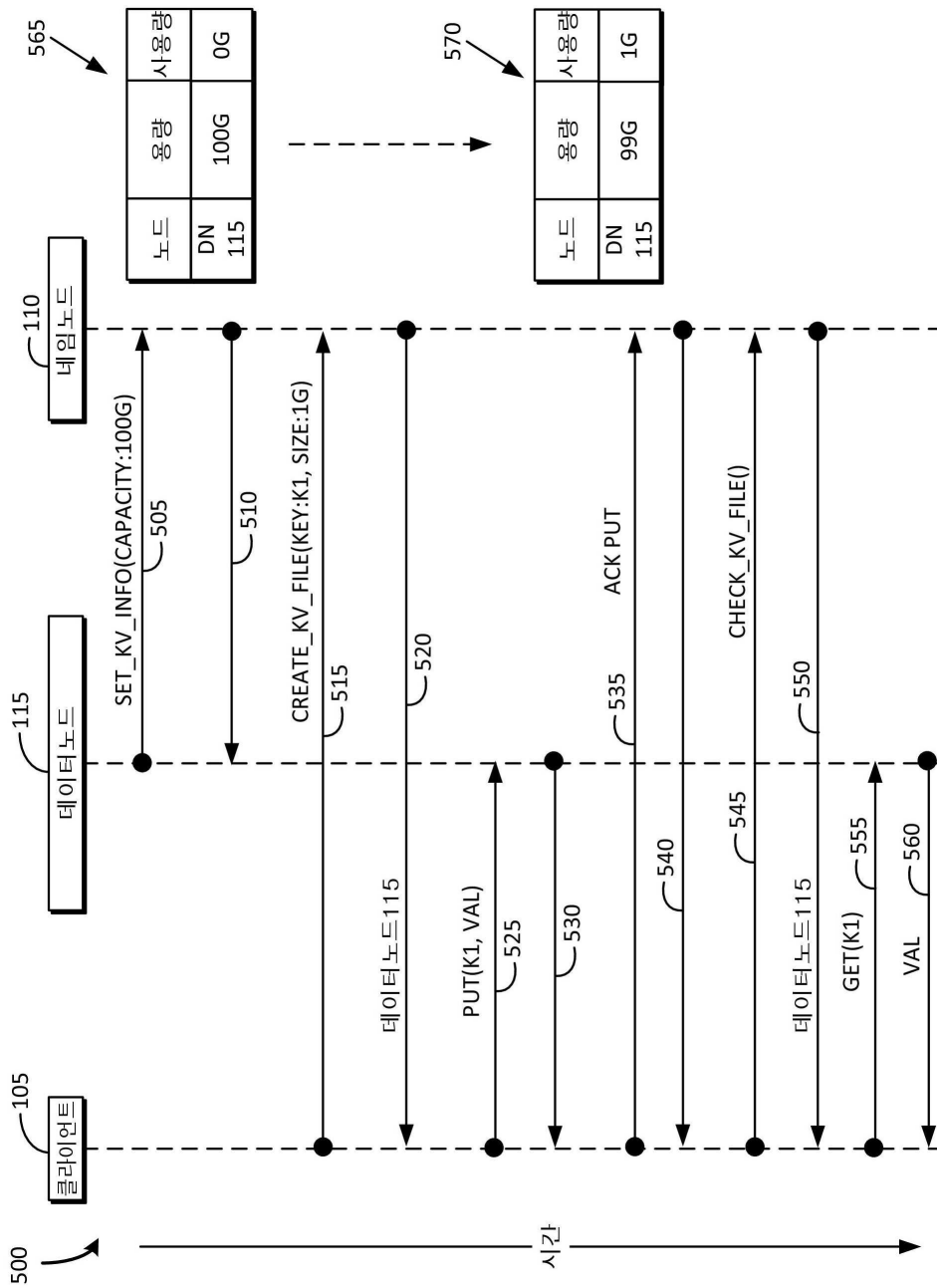
도면3



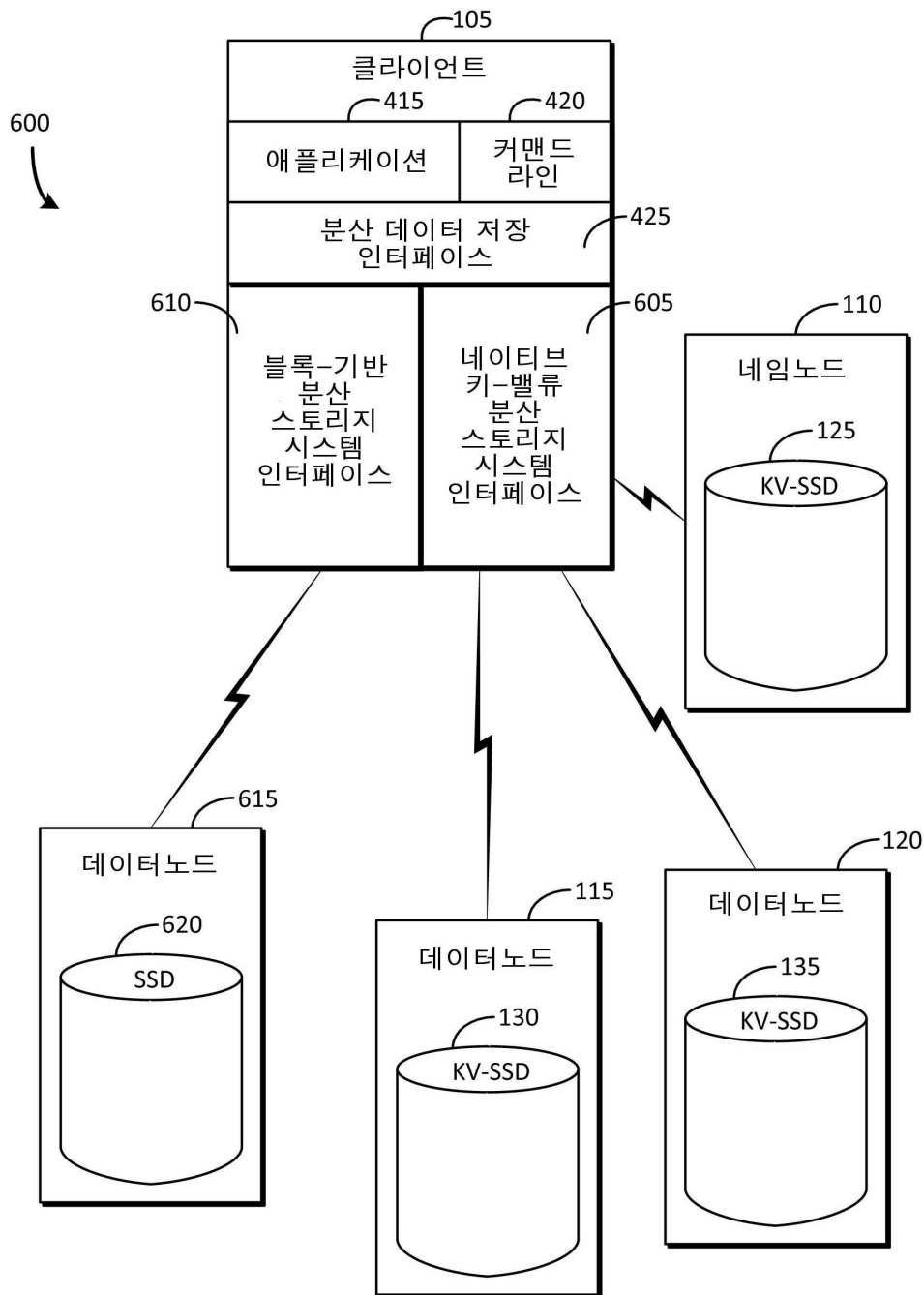
도면4



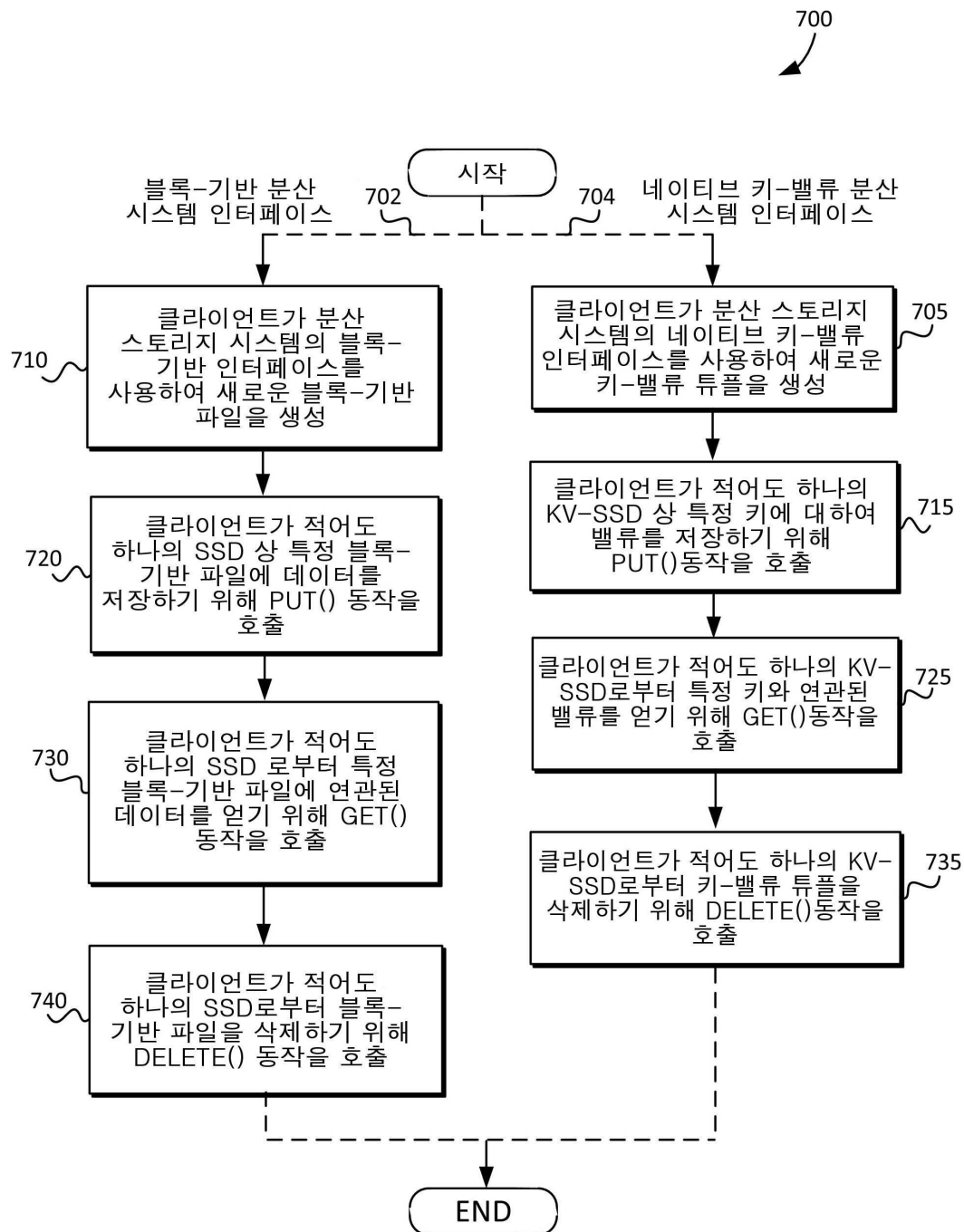
도면5



도면6



도면7



도면8

