(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0051293
(43) 공개일자 2014년04월30일

(51) 국제특허분류(Int. Cl.)
G06F 15/16 (2006.01)
(21) 출원번호 10-2014-7003286
(22) 출원일자(국제) 2012년07월19일
심사청구일자 없음
(85) 번역문제출일자 2014년02월07일
(86) 국제출원번호 PCT/US2012/047261
(87) 국제공개번호 WO 2013/022582
국제공개일자 2013년02월14일
(30) 우선권주장
13/207,014 2011년08월10일 미국(US)

(71) 출원인
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
크리스티안센 닐 알
미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션
그린 더스틴 엘
미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션
(뒷면에 계속)
(74) 대리인
제일특허법인

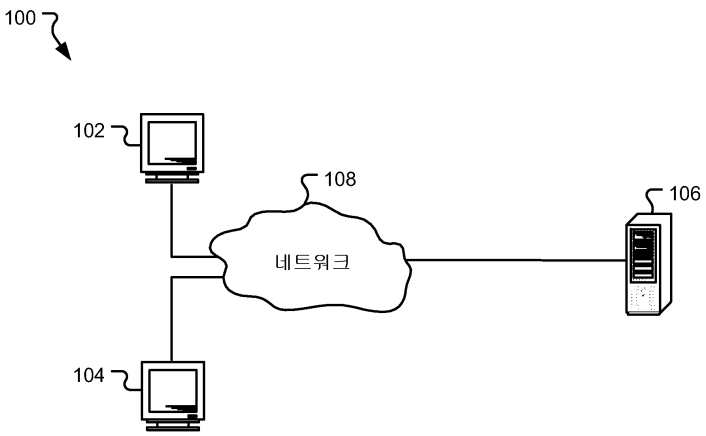
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **토큰 기반 파일 작업 기법**

(57) 요 약

토큰 기반 파일 작업들을 허가하는 실시예들이 설명된다. 클라이언트는 파일 액세스 프로토콜에 따라 포맷팅되는 특수한 오프로드 파일 작업을 요청할 수 있다. 파일 작업은 오프로드 판독 작업 또는 오프로드 기록 작업일 수 있다. 오프로드 판독 작업에서, 클라이언트는 저장된 파일 또는 그 일부로부터 데이터가 논리적으로 판독될 것으로 요청한다. 이에 응답하여, 파일 서버는 논리적으로 판독된 데이터를 나타내는 토큰을 포함하는 응답을 제공한다. 일부 실시예들에서, 파일 서버는 소정의 이유로 데이터 전부를 나타내는 토큰을 제공할 수 없는 경우에는 요청된 데이터의 전부보다 적은 데이터를 나타내는 토큰을 갖는 응답을 반환할 수 있다. 이어서, 클라이언트는 후속 오프로드 기록 작업에서 토큰을 사용할 수 있다. 실시예들에서, 토큰들은 서버들 및 클라이언트들 전반에서 안전하고 확실하게 사용될 수 있는 불변 데이터를 나타낸다.

대 표 도 - 도1



(72) 발명자

핑커톤 제임스 티

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

나가 라지브

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

매튜 브라이언 스티븐

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

아이살 자이비르 케이

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

특허청구의 범위

청구항 1

토큰 기반 파일 작업을 제공하는 컴퓨터 구현 방법으로서,

파일 서버에서, 파일 시스템 내의 정보에 액세스하기 위해 상기 파일 서버에 접속하기 위한 제1 요청을 수신하는 단계와,

상기 파일 서버로부터 제1 응답을 전송하는 단계 - 상기 응답은 상기 파일 시스템 내의 상기 정보에 대한 액세스를 허가하기 위해 클라이언트와의 세션을 설정함 - 와,

상기 파일 서버에서, 상기 파일 시스템 내의 파일로부터 파일 정보에 액세스하기 위해 상기 파일을 개방하기 위한 제2 요청을 수신하는 단계와,

상기 제2 요청의 수신에 응답하여, 상기 파일 서버가 상기 파일에 대한 액세스를 허가하는 제2 응답을 상기 클라이언트로 전송하는 단계와,

상기 파일 서버에서, 상기 파일의 일부로부터의 제1 데이터의 오프로드 판독을 위한 제3 요청을 수신하는 단계 - 상기 제3 요청은 파일 액세스 프로토콜에 따라 포맷팅됨 - 와,

상기 제3 요청의 수신에 응답하여, 상기 파일 서버가 상기 제1 데이터를 표현하는 토큰을 갖는 제3 응답을 전송하는 단계 - 상기 제1 데이터는 상기 파일의 상기 일부로부터 논리적으로 판독되고, 상기 제3 응답은 상기 파일 액세스 프로토콜에 따라 포맷팅됨 -

를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 제3 요청은 상기 파일로부터 판독될 제1 데이터 및 상기 파일로부터 판독될 제2 데이터를 나타내고, 상기 제3 응답은 상기 제1 데이터를 표현하는 상기 토큰 및 상기 제2 데이터를 표현하는 제2 토큰을 포함하는 방법.

청구항 3

제1항에 있어서,

상기 제3 요청은 제1 파일의 제1 부분 및 제2 파일의 제2 부분을 나타내고, 상기 제3 응답은 상기 제1 파일의 상기 제1 부분으로부터 논리적으로 판독된 제1 데이터를 표현하는 상기 토큰 및 상기 제2 파일의 상기 제2 부분으로부터 논리적으로 판독된 제2 데이터를 표현하는 제2 토큰을 포함하는 방법.

청구항 4

제1항에 있어서,

상기 파일 서버에서, 상기 제1 데이터의 요청된 부분의 제2 파일로의 오프로드 기록을 위한 제4 요청을 수신하는 단계 - 상기 제4 요청은 상기 토큰을 포함하고, 상기 파일 액세스 프로토콜에 따라 포맷팅됨 - 와,

상기 제4 요청의 수신에 응답하여, 상기 파일 서버가

상기 제1 데이터의 상기 요청된 부분을 상기 제2 파일에 기록하는 단계와,

상기 제1 데이터의 상기 요청된 부분이 상기 제2 파일에 기록되었음을 나타내는 제4 응답을 전송하는 단계 - 상기 제4 응답은 상기 파일 액세스 프로토콜에 따라 포맷팅됨 -

를 더 포함하는 방법.

청구항 5

제1항에 있어서,

상기 파일 서버에서, 상기 제1 데이터의 요청된 부분의 제2 파일로의 오프로드 기록을 위한 제4 요청을 수신하는 단계 - 상기 제4 요청은 상기 토큰을 포함하고, 상기 파일 액세스 프로토콜에 따라 포맷팅됨 - 와,

상기 제4 요청의 수신에 응답하여, 상기 파일 서버가

상기 제1 데이터의 상기 요청된 부분의 제1 부분을 상기 제2 파일에 기록하는 단계 - 상기 요청된 부분의 상기 제1 부분은 상기 제1 데이터의 상기 요청된 부분의 전부보다 적음 - 와,

상기 제1 데이터의 상기 요청된 부분의 상기 제1 부분이 상기 제2 파일에 기록되었음을 나타내는 제4 응답을 전송하는 단계 - 상기 제4 응답은 상기 파일 액세스 프로토콜에 따라 포맷팅됨 -

를 더 포함하는 방법.

청구항 6

프로세서에 의해 실행될 때 토큰 기반 파일 작업을 요청하는 방법을 수행하는 컴퓨터 실행 가능 명령어를 포함하는 컴퓨터 판독 가능 저장 매체로서,

상기 방법은

클라이언트에 의해, 파일 시스템 내의 정보에 액세스하기 위해 파일 서버에 접속하기 위한 제1 요청을 전송하는 단계와,

제1 응답을 수신하는 단계 - 상기 응답은 상기 파일 정보에 대한 액세스를 허가하기 위해 상기 클라이언트와의 세션을 설정함 - 와,

상기 파일 시스템 내의 파일을 개방하기 위한 제2 요청을 전송하는 단계와,

상기 파일에 대한 액세스를 허가하는 제2 응답을 수신하는 단계와,

토큰에 의해 표현되는 데이터의 제1 부분의 파일로의 오프로드 기록을 위한 제3 요청을 전송하는 단계 - 상기 제3 요청은 서버 메시지 블록(SMB) 프로토콜의 버전에 따라 포맷팅되고, 상기 데이터를 표현하는 상기 토큰을 포함함 - 와,

응답을 수신하는 단계

를 포함하는 컴퓨터 판독 가능 저장 매체.

청구항 7

제6항에 있어서,

상기 응답은 상기 데이터의 제2 부분이 상기 파일에 성공적으로 기록되었음을 나타내고, 상기 제2 부분은 상기 제1 부분 내의 상기 데이터의 전부보다 적은 데이터를 나타내는 컴퓨터 판독 가능 저장 매체.

청구항 8

제6항에 있어서,

제2 파일의 일부로부터의 제2 데이터의 오프로드 판독을 위한 제4 요청을 전송하는 단계 - 상기 제4 요청은 상기 SMB 프로토콜의 상기 버전에 따라 포맷팅됨 - 와,

상기 제2 데이터를 표현하는 토큰을 갖는 제4 응답을 수신하는 단계 - 상기 제4 응답은 상기 SMB 프로토콜의 상기 버전에 따라 포맷팅됨 -

를 더 포함하는 컴퓨터 판독 가능 저장 매체.

청구항 9

토큰 기반 파일 작업을 허가하기 위한 시스템으로서,

적어도 하나의 서버를 포함하고,

상기 적어도 하나의 서버는

컴퓨터 실행 가능 명령어를 실행하도록 구성된 적어도 하나의 프로세서,

상기 컴퓨터 실행 가능 명령어를 저장하는 적어도 하나의 컴퓨터 판독 가능 저장 매체

를 포함하고,

상기 컴퓨터 실행 가능 명령어는 상기 적어도 하나의 프로세서에 의해 실행될 때,

파일의 일부로부터의 데이터의 오프로드 판독을 위한 요청을 수신하고 - 상기 요청은 서버 메시지 블록(SMB) 프로토콜의 버전에 따라 포맷팅됨 -,

상기 요청의 수신에 응답하여, 상기 데이터를 표현하는 토큰을 갖는 응답을 전송

하도록 구성되는 파일 서버를 제공하며, 상기 응답은 상기 SMB 프로토콜의 상기 버전에 따라 포맷팅되는 시스템.

청구항 10

제9항에 있어서,

적어도 하나의 클라이언트를 더 포함하고,

상기 적어도 하나의 클라이언트는

컴퓨터 실행 가능 명령어를 실행하도록 구성되는 적어도 하나의 프로세서와,

상기 적어도 하나의 프로세서에 의해 실행될 때, 상기 파일의 상기 일부로부터의 데이터의 상기 오프로드 판독을 위한 상기 요청을 전송하고, 상기 데이터를 표현하는 상기 토큰을 갖는 상기 응답을 수신하는 상기 컴퓨터 실행 가능 명령어를 저장하는 적어도 하나의 컴퓨터 판독 가능 저장 매체

를 포함하는 시스템.

명 세 서

배 경 기 술

[0001] 많은 양의 데이터를 복사하는 전통적인 방식들에서는 데이터가 소스 파일로부터 로컬 RAM 내로 판독된 후에 동일 바이트들이 RAM으로부터 다시 목적지 파일에 기록된다. 이러한 프로세스는 복사의 완료가 로컬 RAM 내에 데이터가 항상 있을 것을 본질적으로 요구하지 않는 경우에도 데이터가 로컬 RAM을 포함하는 라우트를 통과할 것을 요구한다. 데이터가 최종 소스와 최종 목적지 사이에서 취할 수 있는 신뢰되는 더 빠른 라우트가 존재할 때, 로컬 RAM을 통한 우회는 불필요하다. 이러한 문제는 신뢰되는 더 빠른 라우트와 로컬 RAM을 경유하는 라우트 사이에 속도 차이가 클 때 더 심각하다. 일반적으로, 서버 메시지 블록(SMB) 파일 서버들과 같은 일부 파일 서버들은 클라이언트가 소스 파일의 범위들을 목적지 파일의 범위들에 복사하는 것을 가능하게 한다. 그러나, 동일 파일 서버 상에서 열린 파일들 사이에서의 데이터 복사에 대한 제한들과 같은 다수의 제한이 존재한다. 또한, SMB 파일 서버들은 클라이언트가 소스 및 목적지 범위들 둘 다를 지정하는 단일 명령을 발하는 것을 허가할 뿐이다. 종종, 클라이언트 코드 구조는 복사를 달성하기 위해 판독 및 기록을 따로따로 사용하도록 셋업될

것이며, 이는 SMB 파일 서버들이 데이터의 범위들을 복사하기 위해 제공하는 통상의 방식에 적합하지 않다.

[0002] 이들 및 다른 사항들에 관하여 실시예들이 이루어졌다. 또한, 비교적 특정한 문제들이 설명되었지만, 실시예들은 배경 기술에서 식별된 특정한 문제들을 해결하는 것으로 한정되지 않아야 한다는 것을 이해해야 한다.

발명의 내용

[0003] 이 요약은 아래의 상세한 설명 부분에서 더 설명되는 개념들의 발췌를 간단한 형태로 소개하기 위해 제공된다. 이 요약은 청구 발명 대상의 중요한 특징들 또는 본질적인 특징들을 식별하는 것을 의도하지 않으며, 청구 발명 대상의 범위를 결정하는 데 있어서의 보조물로서 사용되는 것도 의도하지 않는다.

[0004] 토큰 기반 파일 작업들을 허가하는 실시예들이 설명된다. 실시예들은 파일 서버와의 세션을 설정하는 클라이언트를 제공한다. 일례로 서버 메시지 블록(SMB) 프로토콜을 포함하는 임의의 파일 액세스 프로토콜을 이용하여 세션이 설정될 수 있다. 세션이 설정된 후, 클라이언트는 파일 액세스 프로토콜에 따라 포맷팅되는 특수한 오프로드 파일 작업을 요청할 수 있다. 파일 작업은 판독 작업 또는 기록 작업일 수 있다. 오프로드 판독 작업에서, 클라이언트는 파일 서버가 액세스할 수 있는 파일 저장 시스템 내에 저장된 파일로부터 파일 데이터가 판독될 것으로 요청한다. 이어서, 파일 서버는 파일 데이터를 나타내는 토큰을 포함하는 응답을 제공할 것이다. 일부 실시예들에서, 파일 서버는 소정의 이유로 파일 데이터 전부를 나타내는 토큰을 제공할 수 없는 경우에는 전부보다 적은 파일 데이터를 나타내는 토큰을 갖는 응답을 반환할 수 있다. 이어서, 클라이언트는 후속 오프로드 기록 작업 또는 다른 관련된 작업들(예를 들어, 토큰에 의해 표현되는 데이터를 필요할 경우에 후속적으로 획득하는 작업)에서 토큰을 사용할 수 있다. 실시예들에서, 토큰들은 서버들 및 클라이언트들 전반에서 안전하고 확실하게 사용될 수 있는 불변 데이터를 나타낸다.

[0005] 실시예들은 컴퓨터 프로세스로서, 컴퓨팅 시스템으로서, 또는 컴퓨터 프로그램 제품 또는 컴퓨터 판독 가능 매체와 같은 제조물로서 구현될 수 있다. 컴퓨터 프로그램 제품은, 컴퓨터 시스템에 의해 판독될 수 있고, 컴퓨터 프로세스를 실행하기 위한 명령어들의 컴퓨터 프로그램을 인코딩하는 컴퓨터 저장 매체일 수 있다. 컴퓨터 프로그램 제품은, 컴퓨팅 시스템에 의해 판독될 수 있고, 컴퓨터 프로세스를 실행하기 위한 명령어들의 컴퓨터 프로그램을 인코딩하는 반송파 상의 전파 신호일 수도 있다.

도면의 간단한 설명

[0006] 비한정적이고 비포괄적인 실시예들이 아래의 도면들을 참조하여 설명된다.

도 1은 실시예들을 구현하는 데 사용될 수 있는 시스템을 도시한다.

도 2는 일부 실시예들에 따른 파일 액세스 프로토콜을 이용하여 토큰 기반 파일 작업들을 행하는 클라이언트들 및 서버들의 블록도를 도시한다.

도 3은 일부 실시예들에 따른 오프로드 파일 작업들을 처리하기 위한 동작 흐름을 도시한다.

도 4는 일부 실시예들에 따른 오프로드 판독 요청을 처리하기 위한 동작 흐름을 도시한다.

도 5는 일부 실시예들에 따른 오프로드 기록 요청을 처리하기 위한 동작 흐름을 도시한다.

도 6은 일부 실시예들에 따른 오프로드 파일 작업들을 요청하기 위한 동작 흐름을 도시한다.

도 7은 실시예들을 구현하는 데 적합한 컴퓨팅 환경의 블록도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0007] 이하, 다양한 실시예들이 첨부 도면들을 참조하여 더 충분히 설명되고, 첨부 도면들은 그 일부를 형성하고 특정 실시예들을 도시한다. 그러나, 실시예들은 많은 상이한 형태로 구현될 수 있고, 본 명세서에서 설명되는 실시예들로 한정되는 것으로 해석되지 않아야 하며, 오히려 이러한 실시예들은 본 명세서가 면밀하고 완전하며 이 분야의 기술자들에게 실시예들의 범위를 충분히 전달하도록 제공된다. 실시예들은 방법들, 시스템들 또는 장치들로서 실시될 수 있다. 따라서, 실시예들은 하드웨어 구현, 오로지 소프트웨어 구현 또는 소프트웨어 및 하드웨어 양태들을 결합하는 구현의 형태를 취할 수 있다. 따라서, 아래의 상세한 설명은 한정적인 것으로 간주되

지 않아야 한다.

[0008] 도 1은 일부 실시예들을 구현하는 데 사용될 수 있는 시스템(100)을 도시한다. 시스템(100)은 클라이언트들(102, 104) 및 서버(106)를 포함한다. 클라이언트들(102, 104)은 네트워크(108)를 통해 서버(106)와 통신한다. 서버(106)는 클라이언트들(102, 104) 상의 애플리케이션들에 의해 액세스되는 정보를 저장한다. 클라이언트들(102, 104)은 서버(106) 상의 정보에 액세스하기 위해 서버(106)와의 세션들을 설정한다. 도 1에는 클라이언트들(102, 104)만이 서버(106)와 통신하는 것으로 도시되지만, 다른 실시예들에서는 서버(106)로부터의 정보에 액세스하는 둘보다 많은 클라이언트가 존재할 수 있다.

[0009] 실시예들에서, 클라이언트들(102, 104) 상의 애플리케이션들은 애플리케이션에 투명한, 파일 시스템으로부터의 파일 정보를 요청한다. 파일 정보는 서버(106) 상의 파일 시스템으로부터 검색된다. 일 실시예에서, 서버(106) 상의 그러한 파일 시스템은 원격 파일 시스템이다. 다른 실시예에서, 서버(106) 상의 파일 시스템은 분산 파일 시스템이다. 본 발명의 사상 및 범위로부터 벗어나지 않고서 본 명세서에서 설명되는 실시예들에 따라 다수의 타입의 파일 시스템들이 사용될 수 있다. 게다가, 도시되지는 않았지만, 일부 실시예들에서는 단일 서버(106) 대신에 서버는 예를 들어 서버 클러스터의 일부인 다수의 서버 중 하나일 수 있다. 다른 실시예들에서, 서버는 서버 클러스터의 일부가 아닌 다수의 서버 중 하나일 수 있다. 그러한 실시예들에서 복수의 파일 서버는 정보, 예를 들어 파일 정보의 중복성 및 높은 가용성을 클라이언트들(102, 104)에 제공한다.

[0010] 일 실시예에서, 클라이언트들(102, 104)은 서버(106) 상의 원격 파일 시스템에 저장된 파일들에 대해 수행될 다수의 파일 작업을 전송할 수 있다. 클라이언트들(102, 104)은 파일 액세스 프로토콜을 이용하여, 파일들에 대해 수행될 파일 작업들에 대한 요청들을 포맷팅한다. 파일 액세스 프로토콜은 네트워크 파일 시스템(NFS) 또는 서버 메시지 블록(SMB) 프로토콜의 일 버전과 같은 임의의 적절한 프로토콜일 수 있다. 일부 실시예들에 따르면, 클라이언트들은 정규 판독 및 기록 파일 작업들을 전송하는 것에 더하여 토큰 기반 작업들인 오프로드 판독 및 오프로드 기록 작업들을 요청할 수 있다. 아래에 더 상세히 설명되는 바와 같이, 오프로드 파일 작업들은 실제 데이터를 네트워크를 통해 클라이언트들(102, 104) 중 하나의 클라이언트 상의 로컬 RAM으로 전송할 필요 없이 많은 양의 데이터가 클라이언트들(102, 104)에 의해 이동되는 것을 가능하게 한다.

[0011] 예를 들어, 일 실시예에서, 클라이언트(102)는 서버(106)와의 세션을 설정하기 위한 요청을 전송할 수 있다. 예를 들어, 클라이언트(102)는 서버 메시지 블록(SMB) 프로토콜의 일 버전을 이용하여 서버(106) 상에 저장된 파일 시스템에 액세스하기 위해 서버(106)와의 세션을 설정할 수 있다. 세션의 설정은 클라이언트(102)와 서버(106) 사이에 전송되는 다수의 협상 요청 및 응답의 교환을 포함할 수 있다. SMB 프로토콜의 버전들에서는, 세션 동안 사용될 프로토콜의 정확한 버전을 협상하는 것은 물론, 클라이언트, 예를 들어 102 및 서버, 예를 들어 106 양자의 능력들을 서로에게 광고하는 데 사용되는 구체적으로 정의된 협상 패킷들이 존재한다. 일 실시예에서, 협상 패킷들은 서버(106)가 토큰 기반 파일 작업들, 즉 오프로드 판독 및 오프로드 기록 명령들을 처리할 수 있다는 지시를 포함할 수 있다. 이것은 클라이언트로 하여금 클라이언트가 원할 경우에 서버로부터 오프로드 파일 작업들을 요청할 수 있다는 것을 알게 해준다.

[0012] 위의 예를 계속하면, 세션이 설정된 후, 클라이언트(102)는 SMB 프로토콜에 따라 포맷팅된 메시지를 서버(106)로 전송하여, 서버(106) 상의 파일 시스템 내의 파일을 열 수 있다. 서버는 열린 파일에 대한 핸들로 응답할 수 있다. 이어서, 클라이언트(102)는 SMB 프로토콜에 따라 포맷팅된 오프로드 판독 동작을 요청하여, 파일로부터의 파일 데이터를 요청할 수 있다. 일 실시예에서, 클라이언트는 오프로드 판독 작업에서 파일의 일부로부터의 데이터를 요청한다. 오프로드 판독 작업 요청은 토큰 기반 판독 작업이다.

[0013] 클라이언트(102)로부터의 요청에 응답하여, 서버(106)는 SMB 프로토콜에 따라 포맷팅된 응답을 클라이언트(102)에 의해 요청된 파일 데이터를 나타내는 토큰과 함께 전송한다. 일부 실시예들에서, 서버(106)는 토큰을 생성하고, 토큰이 동일 파일 데이터를 요청할 수 있는 클라이언트(104)와 같은 다른 클라이언트들로부터의 임의의 요청들에 걸쳐 파일 데이터를 일관성 있게 표현하는 것을 보증하는 것을 담당할 수 있다. 다른 실시예들에서, 파일 서버는 임의의 요청들을 클라이언트들로부터 기본 파일 저장 시스템으로 전달할 수 있다. 이러한 실시예들에서, 기본 파일 저장 시스템은 클라이언트(102)에 의해 요청된 파일 데이터를 나타내는 토큰을 생성하는 것을 담당한다. 어느 하나의 실시예에서, 서버(106)는 토큰과 함께 응답을 클라이언트(102)로 전송할 것이다. 토큰들을 생성함에 있어서, 실시예들은 예를 들어 토큰을 생성하는 데 사용되는 소스 파일의 범위들이 연속적이 아닌 경우에도 토큰이 생성될 수 있도록 규정한다. 그러한 실시예들에서, 그러한 소스 범위들로부터의 데이터는 토큰에 의해 표현되는 데이터의 단일 논리 범위 내로 논리적으로 연결된다. 일부 실시예들에서, 구현들은 토큰을 특정 소스 범위들과 내부적으로 연관시킬 수 있으며, 그러한 소스 범위들은 서로 연속적이 아닐 수

있다.

[0014] 토큰은 불변 데이터, 즉 클라이언트(102)에 의해 요청된 파일 데이터를 나타낸다. 따라서, 클라이언트(102)는 서버(106)에 의해 반환되는 토큰을 이용하여 다른 파일 작업들을 수행할 수 있다. 예를 들어, 후속 시점에, 클라이언트(102)는 토큰을 이용하여 데이터를 다른 파일 내에 기록할 수 있다. 이 예에서, 클라이언트(102)는 SMB 프로토콜에 따라 포맷팅된 오프로드 기록 작업을 요청할 수 있으며, 이 오프로드 기록 작업도 토큰 기반 파일 작업이다. 오프로드 기록 작업은 이전에 클라이언트(102)에 제공된 토큰을 포함할 수 있다. 오프로드 기록 작업은 토큰에 의해 표현되는 파일 데이터가 서버(106) 상의 다른 파일 내에 기록될 것을 요청할 수 있다. 요청의 수신에 응답하여, 서버(106)는 일 실시예에서 토큰에 의해 표현되는 파일 데이터를 서버(106) 상의 다른 파일 내에 기록함으로써 요청을 처리할 것이다. 다른 실시예에서, 요청의 수신에 응답하여, 서버(106)는 먼저 수신된 토큰을 검증하고, 토큰이 유효한 경우에 토큰에 의해 표현되는 파일 데이터를 서버(106) 상의 다른 파일 내에 기록할 것이다. 전술한 바와 같이, 토큰들이 기본 파일 저장 시스템 또는 하위 계층(들)에 의해 생성되는 실시예들에서, 서버(106)는 단지 오프로드 기록 작업을 기본 파일 저장 시스템으로 전달할 것이며, 이어서 기본 파일 저장 시스템은 요청을 처리하고 파일 데이터를 다른 파일 내에 기록할 것이다. 이어서, 서버(106)는 오프로드 기록이 성공적이었는지를 지시하는 응답을 클라이언트(102)로 전송할 것이다. "파일" 데이터가 참조되지만, 다른 실시예들은 임의의 타입의 데이터를 표현하는 토큰을 제공한다. 예를 들어, 실시예들은 파일, 볼륨, 디스크, 볼륨 스냅샷, 디스크 스냅샷, 블로브 스토어 등과 같은 임의의 저장 컨테이너로부터 획득된 토큰을 제공한다. 용어 "파일 데이터"는 본 명세서에서 설명의 목적을 위해 사용되며, 한정을 의도하지 않는다. 게다가, 전술한 실시예는 토큰에 의해 표현되는 파일 데이터가 서버(106) 상의 다른 파일 내에 기록될 것을 요청하는 오프로드 기록 작업을 제공하지만, 다른 실시예는 토큰에 의해 표현되는 파일 데이터의 일부가 서버(106) 상의 다른 파일 내에 기록될 것을 요청하는 오프로드 기록 작업을 제공한다.

[0015] 게다가, 전술한 실시예들은 클라이언트(102)에 이전에 제공된 토큰을 포함하는 오프로드 기록 작업을 제공하지만, 다른 실시예들은 공지 토큰을 오프로드 기록 작업을 갖는 토큰으로서 사용하는 클라이언트(102)를 제공한다. 예를 들어, 어떠한 이전의 대응하는 오프로드 판독 없이도 제로 토큰과 같은 공지 토큰을 이용하여 (제로들과 같은) 데이터를 기록할 수 있다.

[0016] 일부 실시예들에서, 파일 서버(106)는 클라이언트에 의해 요청된 파일 데이터 전부에 대한 토큰을 생성하지 못할 수 있다. 이것은 예를 들어 클라이언트(104)와 같은 다른 클라이언트가 클라이언트(102)에 의해 요청된 파일 데이터의 범위의 소정 부분에 대한 잠금을 갖는 경우에 발생할 수 있다. 이러한 실시예들에서, 서버(106)는 줄여진 응답을 전송할 수 있다. 즉, 응답은 클라이언트(102)에 의해 요청된 파일 데이터의 일부만을 나타내는 토큰을 포함한다. 따라서, 실시예들에서, 응답은 토큰이 오프로드 판독 요청에서 요청된 제1 데이터의 전부보다 적은 데이터를 나타낸다는 것을 지시한다. 일부 실시예들에서, 이러한 토큰은 불연속적인 데이터 범위를 나타낼 수 있다. 그러나, 다른 실시예들에서, 토큰은 데이터의 연속 범위를 나타낼 수 있지만, 오프로드 판독 요청에서 클라이언트(102)에 의해 요청된 파일 데이터보다 적을 수 있다. 이러한 실시예들에서, 서버 응답은 줄여진 응답 내에서 전송된 토큰에 의해 표현되는 파일 데이터의 부분의 지시를 포함할 것이다.

[0017] 일부 실시예들은 줄여진 오프로드 판독 응답을 제공하지만, 본 발명의 실시예들은 줄여진 오프로드 기록도 제공한다. 전술한 바와 같이, 오프로드 기록 작업은 관련 토큰에 의해 표현되는 데이터가 서버, 예를 들어 서버(106) 상의 다른 파일 내에 기록될 것을 요청할 수 있다. 요청의 수신에 응답하여, 서버(106)는 토큰에 의해 표현되는 파일 데이터를 서버(106) 상의 다른 파일 내에 기록함으로써 요청을 처리할 것이다. 실시예들에서, 서버(106)는 토큰에 의해 표현되는 파일 데이터의 전부를 다른 파일 내에 기록하지 못할 수 있다. 결과적으로, 오프로드 기록은 실시예들에서 줄여질 수 있으며, 예를 들어 "기록된 길이"는 요청된 기록 길이보다 작다. 예를 들어, 크기 제한들은 토큰에 의해 표현되는 데이터 전부를 다른 파일 내에 기록하는 능력을 제한할 수 있다. 다른 실시예에서, 기록될 파일의 일부에 대한 잠금은 파일의 그 부분에 대한 기록을 막을 수 있다. 처리 에러들 또는 다른 타입의 에러들도 토큰에 의해 표현되는 데이터의 부분들이 다른 파일 내에 기록되지 못하게 할 수 있다. 게다가, 토큰은 부분적으로 손상되거나 부분적으로 무효할 수 있으며, 이 경우에 서버(106)는 데이터의 손상된/무효 부분을 다른 파일 내에 성공적으로 기록하지 못한다. 본 발명의 사상 및 범위로부터 벗어나지 않고서 본 명세서에서 개시되는 실시예들에 따르면, 다른 이유들도 토큰에 의해 표현되는 데이터 전부가 다른 파일 내에 기록되지 못하게 할 수 있다. 줄여진 오프로드 기록을 포함하는 실시예들에서는, 줄여진 오프로드 기록 응답이 서버(106)로부터 클라이언트(102)로 전송되어, 예를 들어 데이터의 일부가 다른 파일 내에 기록되지 못했다는 것을 지시할 수 있다. 그러한 지시는 실시예들에서 예를 들어 플래그 또는 다른 지시자의 사용을 통해 이루어질 수 있다. 게다가, 실시예들에서, 줄여진 오프로드 기록 응답은 얼마나 많은 데이터가 실제로 기록

되었는지를 지시한다.

[0018] 위의 설명은 도 1에 도시된 실시예가 어떻게 동작할 수 있는지에 대한 일례일 뿐이다. 아래에 더 상세히 설명되는 바와 같이, 실시예들은 상이한 단계들 또는 동작들을 포함할 수 있다. 이들은 임의의 적절한 소프트웨어 또는 하드웨어 컴포넌트들을 이용하여 구현될 수 있다.

[0019] 이제, 도 2를 참조하면, 도 2는 클라이언트(202), 클라이언트(204), 서버(206) 및 서버(208)를 포함하는 소프트웨어 환경(200)의 블록도를 나타낸다. 파일 정보가 저장되는 파일 저장소(210)도 도시된다.

[0020] 도 2에 도시된 바와 같이, 클라이언트(202) 및 클라이언트(204) 각각은 파일 정보를 요청할 수 있는 애플리케이션을 포함한다. 애플리케이션은 예를 들어 워드 프로세싱 애플리케이션, 스프레드시트 애플리케이션, 브라우저 애플리케이션 또는 파일들에 대한 액세스를 요청하는 임의의 다른 애플리케이션일 수 있다. 도 2에 도시된 실시예에서, 파일들은 파일 저장소(210) 내에 저장된 파일 시스템 내에 위치한다. 도 2는 본 명세서에서 개시되는 일 실시예에 따라 서버들(206, 208)에 대한 공유 저장 능력들을 제공하는 파일 저장소(210)를 도시하지만, 다른 실시예들은 다른 저장 수단을 갖는다. 예를 들어, 서버(206) 및 서버(208) 각각은 실시예들에 따라 분리되거나 부착되는지에 관계없이 그들 자신의 저장 수단을 가질 수 있다. 또 다른 실시예들에서, 서버(206) 및 서버(208)는 각각 그들 자신의 저장 수단을 가질 수 있고, 저장소(210)의 사용을 통해 공유 저장 능력들을 가질 수 있다. 본 발명의 사상 및 범위로부터 벗어나지 않고서 본 명세서에서 개시되는 실시예들에 따라 다양한 타입의 저장소가 사용될 수 있다. 클라이언트(202) 및 클라이언트(204) 각각은 애플리케이션들로부터의 파일들에 대한 요청을 원격 파일 시스템에 대한 액세스를 제공하는 파일 서버로 재지향시키는 재지향기를 더 포함할 수 있다. 재지향기들은 파일 액세스 프로토콜을 이용하여 파일 서버들과 통신한다. 일부 실시예들에서, 파일 액세스 프로토콜은 NFS 또는 SMB 프로토콜의 일 버전일 수 있다. 설명의 목적을 위해, 도 2는 클라이언트(202) 및 클라이언트(204) 내의 재지향기들이 SMB 2와 같은 SMB 프로토콜의 일 버전을 이용하여 파일 서버들과 통신하는 것으로 가정하여 설명될 것이다. 그러나, 실시예들은 SMB 프로토콜의 사용으로 한정되지 않는다.

[0021] 서버들(206, 208)은 도 2에서 각자가 파일 서버를 포함하는 것으로 도시된다. 전술한 바와 같이, 파일 서버들은 SMB 프로토콜의 일 버전을 이용하여 클라이언트(202) 및 클라이언트(204) 상의 재지향기들과 통신할 수 있다. 서버들(206, 208) 각각은 파일 데이터를 나타내는 토큰들을 생성하는 토큰 생성기 모듈도 포함한다. 게다가, 파일 저장소(210)도 파일 데이터를 나타내는 토큰들을 생성하기 위한 토큰 생성기 모듈을 포함한다.

[0022] SMB 프로토콜을 이용하여 클라이언트와 서버 사이에 세션을 설정하는 것은 클라이언트(202) 상의 재지향기와 같은 재지향기가 협상 요청을 서버(206)와 같은 파일 서버로 전송하는 것으로부터 시작된다. 재지향기 및 파일 서버는 세션을 위해 사용될 SMB의 버전을 협상하기 위해 협상 패킷들을 교환한다. 게다가, 협상 동안, 능력들도 교환될 수 있다. 일 실시예에서, 서버(206)는 파일 서버로부터 클라이언트로 전송되는 협상 응답 패킷 내에 파일 서버가 오프로드 파일 작업들의 사용을 지원한다는 것을 클라이언트에게 지시하는 능력 플래그를 포함시킬 수 있다. 다른 실시예들에서, 클라이언트(202) 및 서버(206)는 SMB 프로토콜의 버전만을 협상할 수 있으며, 따라서 그 버전이 오프로드 파일 작업들의 사용에 대한 지원을 포함한다는 것을 이해할 수 있다. 또 다른 실시예들에서는, 오프로드 파일 작업이 시도될 때, 프로토콜의 일 버전이 오프로드 파일 작업들을 지원한다는 결정이 이루어진다. 예를 들어, 클라이언트(202)는 오프로드 판독(또는 오프로드 기록) 작업을 요청할 수 있다. 서버(206)가 오프로드 파일 작업들을 지원하는 경우, 서버(206)는 요청의 처리를 진행할 것이다. 서버(206)가 오프로드 파일 작업을 지원하지 않는 경우, 서버(206)는 요청된 오프로드 파일 작업이 수행되지 못한다는 것을 지시하는 응답을 클라이언트(202)로 전송할 것이다. 예를 들어, 일 실시예에서, 서버(206)가 오프로드 파일 작업을 지원하지 않는 경우, 서버(206)는 에러 메시지 및/또는 에러를 지시하는 플래그를 이용하여 클라이언트(202)에 응답한다.

[0023] 협상이 완료되면, 클라이언트(202) 상의 재지향기 및 파일 서버(206)는 세션을 설정한다. 이어서, 클라이언트 재지향기는 파일 액세스 요청들을 파일 서버로 전송할 수 있다. 일 실시예에서, 클라이언트(202) 상의 재지향기는 파일에 대한 개방을 요청한다. 서버(206)는 개방을 위한 핸들을 포함하는 응답을 제공한다. 이어서, 클라이언트(202)는 핸들을 이용하여 오프로드 판독 작업을 요청할 수 있다. 실시예들에서, 오프로드 판독 작업은 SMB 프로토콜에 따라 포맷팅된다. 일부 실시예들에서, 오프로드 판독 및 오프로드 기록 작업들은 다른 파일 시스템 제어 명령들(FSCTL들)과 동일한 방식으로 SMB2 입출력 제어(IOCTL) 요청 내에 명령들을 캡슐화함으로써 SMB 프로토콜 접속을 이용하여 전송된다. 아래는 일부 실시예들에서 오프로드 판독 작업을 요청하는 데 사용될 수 있는 구조의 일례이다.

[0024] typedef struct_FSCTL_OFFLOAD_READ_INPUT {

[0025] ULONG Size;

[0026] ULONG Flags;

[0027] ULONG TokenTimeToLive; // 밀리초 단위

[0028] ULONG Reserved;

[0029] ULONGLONG FileOffset;

[0030] ULONGLONG CopyLength;

[0031] } FSCTL_OFFLOAD_READ_INPUT, *PFSCTL_OFFLOAD_READ_INPUT;

[0032] 전술한 바와 같이, 오프로드 판독 작업을 요청하기 위해 클라이언트에 의해 사용되는 구조는 다수의 필드를 포함할 수 있다. 실시예들에서, 이것은 서버에 대한 생존 시간(time to live) 제안을 포함할 수 있다. 즉, 필드는 서버가 전송할 토큰에 대한 제안된 수명을 지시할 수 있다. 이것은 클라이언트에 의해 요청된 파일 데이터의 파일 오프셋 및 복사 길이도 포함한다.

[0033] 요청에 응답하여, 서버(206)는 응답을 반환할 것이다. 아래는 오프로드 판독 작업에 응답하는 데 사용될 수 있는 구조의 일례이다.

[0034] typedef struct_FSCTL_OFFLOAD_READ_OUTPUT {

[0035] ULONG Size;

[0036] ULONG Flags;

[0037] ULONGLONG TransferLength;

[0038] UCHAR Token[512];

[0039] }FSCTL_OFFLOAD_READ_OUTPUT, *PFSCTL_OFFLOAD_READ_OUTPUT;

[0040] 전술한 바와 같이, 실시예들에서 응답은 클라이언트(202)에 의해 요청된 파일 데이터를 표현하는 토큰을 포함할 것이다. 토큰은 실시예들에서 토큰에 의해 표현되는 파일 데이터의 길이도 포함할 것이다.

[0041] 일부 실시예들에서, 서버(206)는 실패 또는 판독 요청이 적게 처리되었다는 지시를 이용하여 오프로드 판독 요청에 응답할 수 있다. 아래는 클라이언트에 대한 추가적인 정보를 지시하기 위해 서버에 의해 오프로드 판독 응답 내에 설정될 수 있는 3개의 플래그이다.

[0042] #define OFFLOAD_READ_FLAG_ALL_ZERO_BEYOND_CURRENT_RANGE

[0043] (1)

[0044] #define OFFLOAD_READ_FLAG_FILE_TOO_SMALL (2)

[0045] #define OFFLOAD_READ_FLAG_CANNOT_OFFLOAD_BEYOND_CURRENT

[0046] RANGE (4)

[0047] 위에 열거된 제2 플래그는 파일이 너무 작기 때문에 요청이 실패하였다는 것을 지시한다. 작은 파일에 대한 파일 데이터 페이로드가 개별 클러스터들 내에 저장되는 것이 아니라 파일 레코드 내에 직접 저장되는 상황들이 존재할 수 있다. 그러한 파일들의 경우, 상응하게 작은 오프로드 판독으로부터는 거의 효율이 얻어지지 않으며, 따라서 그러한 파일에 대한 작은 오프로드 판독을 처리하는 것이 아니라, 소스 파일 시스템은 오프로드 판독에 실패할 수 있고, 응답은 파일 데이터가 너무 작기 때문에 요청이 실패했다는 것을 지시하는 위에서 정의된 제2 플래그를 포함할 수 있다. 위에서 정의된 제1 플래그는 오프로드 판독 응답에 의해 지시되는 전송 길이를 넘는 나머지 데이터가 모든 제로들을 포함한다는 것을 지시한다. 마지막으로, 제3 플래그는 서버가 오프로드 판독 응답에 의해 지시되는 전송 길이를 넘는 오프로드 판독 요청들이 성공하지 못할 것으로 결정하는 상황들에서 사용될 수 있다. 제3 플래그는 서버가 현재의 오프로드 판독 응답에 의해 지시되는 범위를 초과하는 클라이언트에 의해 요청된 데이터에 대한 토큰을 제공할 수 없을 것이라는 것을 지시한다. 도시되지 않았지만, 오프로드 판독 응답은 반환된 하위 부분이 (오프셋 0으로부터의 연속적인 바이트 카운트만이 아니라) 비연속적인 데이터를 나타낸다는 지시와 같은 정보도 제공할 수 있다. 오프로드 판독 응답은 실시예들에서 오프로드 판

독이 성공할 수 있는 다음의 후속 소스 오프셋에 관한 힌트도 포함할 수 있다. 위에 제공된 3개의 플래그는 플래그들에 대한 수치 값들을 포함하지만, 이러한 수치 값들은 설명의 목적을 위해 제공된다. 일반적으로 다른 수치 값들 또는 값들이 본 발명의 사상 및 범위로부터 벗어나지 않고 실시예들에 따라 사용될 수 있다.

[0048] 클라이언트(202)가 이전의 오프로드 판독 요청에 의해 또는 소정의 다른 수단에 의해 토큰을 수신하면, 클라이언트(202)는 오프로드 기록 요청을 서버(206)로 전송할 수 있다. 아래는 실시예들에서 클라이언트가 오프로드 기록 요청을 전송하기 위해 사용할 수 있는 구조의 일례이다.

[0049] typedef struct_FSCTL_OFFLOAD_WRITE_INPUT {

[0050] ULONG Size;

[0051] ULONG Flags;

[0052] ULONGLONG FileOffset;

[0053] ULONGLONG CopyLength;

[0054] ULONGLONG TransferOffset;

[0055] UCHAR Token[512];

[0056] }FSCTL_OFFLOAD_WRITE_INPUT, *PFSCTL_OFFLOAD_WRITE INPUT;

[0057] 위의 예에서 지시되는 바와 같이, 오프로드 기록의 구조는 복사할 목적지 파일의 오프셋, 복사할 데이터의 길이는 물론, 복사할 곳인 토큰에 의해 표현되는 데이터 내의 오프셋도 포함한다. 서버로부터 또는 다른 수단에 의해 수신되었을 수 있는 토큰도 포함된다.

[0058] 오프로드 기록 요청에 응답하여, 서버(206)는 오프로드 기록 응답을 제공한다. 오프로드 기록 응답에서 서버(206)에 의해 사용하기 위한 구조의 일례가 아래에 제공된다.

[0059] typedef struct_FSCTL_OFFLOAD_WRITE_OUTPUT {

[0060] ULONG Size;

[0061] ULONG Flags;

[0062] ULONGLONG LengthWritten;

[0063] }FSCTL_OFFLOAD_WRITE_OUTPUT,

[0064] *PFSCTL_OFFLOAD_WRITE_OUTPUT;

[0065] 일부 실시예들에서, 서버(206)는 실패로서 오프로드 기록 요청에 응답할 수 있다. 예를 들어, 오프로드 기록 작업은 요청된 파일이 예를 들어 정의된 크기 임계치 아래의 작은 파일일 경우에 실패할 수 있다. 아래는 파일 정보가 너무 작기 때문에 요청이 실패했다는 것을 지시하기 위해 서버에 의해 오프로드 기록 응답 내에 설정될 수 있는 플래그의 일례이다. 전술한 바와 같이, 작은 파일들은 그들의 데이터를 큰 파일들과 다르게 저장할 수 있으며, 실제 파일 데이터 대신에 토큰을 이용하는 것으로부터는 매우 적은 효율이 얻어지므로 그러한 작은 파일에 대해 발해진 오프로드 요청에 실패하는 것이 더 적절할 수 있다. 다른 실시예에서, 오프로드 기록 작업은 오프로드 기록의 수신자와 데이터 소스 간의 링크가 느릴 경우에 실패할 수 있다. 그러한 실시예에서, 파일 시스템은 오프로드 기록 작업에 응답할지를 결정함에 있어서 링크 상태 체크를 수행할 수 있다. 그러한 예들에서, 서버(206)는 오프로드 기록 요청에 실패할 수 있다. 또 다른 실시예들에서, 서버(206)는 무효 토큰으로 인해 요청에 실패할 수 있다. 예를 들어, 토큰은 만료된 경우에 무효할 수 있다. 서버가 무효/만료 토큰으로 인해 요청에 실패하는 경우, 서버(206)는 실패로서 오프로드 기록 요청에 응답할 수 있으며, 예를 들어 토큰이 무효이거나 만료되었기 때문에 요청이 실패했다는 것을 지시하는 (예시적인 플래그로서 아래에 설명되는 바와 같은) 플래그가 서버에 의해 오프로드 기록 응답 내에 설정될 수 있다. 이러한 타입의 플래그는, 기록 작업의 단순한 재시도가 효과가 없을 것이고, 새로운 토큰을 생성하기 위해 재판독해야 한다(즉, 실패는 링크의 일시적인 느림으로 인한 것이 아니라, 예를 들어 토큰의 무효로 인한 것이었다)는 클라이언트에 대한 힌트로서 작용할 수 있다. 대안 실시예에서, 주어진 토큰이 더 이상 유효하지 않다는 것을 지시하기 위해 특정 상태가 반환될 수 있다. 따라서, 실시예들은 토큰이 더 이상 유효하지 않다는 것을 지시하는 데 사용되는 반환 상태를 제공하며, 다른 실시예들은 그러한 지시를 제공하는 데 사용되는 플래그를 제공한다. 또 다른 실시예들은 그러

한 지시를 제공하는 데 사용되는 플래그 및 반환 상태 둘 다를 제공한다. 예를 들어, 에러 코드 대신에, 예를 들어 토큰 만료로 인한 줄임 시에, 작업은 동일 토큰(들)을 이용하여 오프로드 기록의 나머지를 재시도하는 것이 무익하다는 것을 지시하기 위해 플래그, 예를 들어 OFFLOAD_WRITE_FLAG_TOKEN_INVALID와 함께, 줄여진 값을 갖는 성공을 FSCTL의 호출자에게 반환할 수 있다. 본 발명의 사상 및 범위로부터 벗어나지 않고서 본 명세서에서 설명되는 실시예들에 따라 다수의 다른 타입의 조건들이 오프로드 기록 실패를 유발할 수 있다.

[0066] #define OFFLOAD_WRITE_FLAG_FILE_TOO_SMALL (1)

[0067] #define OFFLOAD_WRITE_FLAG_TOKEN_INVALID (2)

[0068] 위에 제공된 플래그는 플래그들에 대한 수치 값들을 포함하지만, 이러한 수치 값들은 설명의 목적을 위해 제공된다. 일반적으로 다른 수치 값들 또는 값들이 본 발명의 사상 및 범위로부터 벗어나지 않고서 실시예들에 따라 사용될 수 있다.

[0069] 게다가, 일부 실시예들은 전술한 바와 같이 줄여지는 오프로드 기록을 제공하며, 예를 들어 기록된 길이는 요청된 기록 길이보다 작다. 그러한 실시예들에서, 서버는 기록되도록 요청된 데이터의 일부만이 실제로 기록되었다는 것을 지시하는 줄여진 오프로드 기록 응답을 이용하여 오프로드 기록 요청에 응답한다.

[0070] 실시예들에서, 오프로드 판독 및 기록 작업들에서 사용되는 토큰들은 표준에 따라 포맷팅된다. 예를 들어, 소형 컴퓨터 시스템 인터페이스(SCSI) 표준은 일 실시예에 따라 사용될 수 있는 토큰 포맷들의 소정 정의를 제공할 수 있다. 특히 고속 컴퓨터 인터페이스 사양들 및/또는 표준들을 포함하는 다양한 타입의 표준들이 본 발명의 사상 및 범위로부터 벗어나지 않고서 본 명세서에서 설명되는 실시예들에 따라 사용될 수 있다. SCSI 표준은 예시적으로 제공된다. 표준 포맷의 사용은 상이한 데이터 액세스 프로토콜을 이용하여 토큰들이 다른 서버들과 연동되게 할 수 있다.

[0071] 서버(206)는 실시예들에서(서버로부터의 또는 임의의 소스로부터의) 이전의 오프로드 판독이 존재하지 않을 때에도 오프로드 기록 명령에서 지정되는 공지 토큰 값들을 인식하고 이용할 수 있다. 예를 들어, 서버(206)는 제로들을 포함하는 범위를 나타내는 공지 토큰을 반환할 수 있다. 클라이언트(202)는 이것을, 소스 파일의 기본 범위들이 제로이고, 토큰과 관련된 데이터가 모드 제로들이라는 것을 지시하는 것으로 해석할 것이다. 다른 실시예에서, 예를 들어, 파일 시스템이 정상적으로 판독되는 경우에 제로들을 반환할 경우, 오프로드 판독 요청에 응답하여 제로 토큰이 반환될 수 있다. 그러한 상황은 예를 들어 파일의 희박한 범위가 판독되는 경우에 발생할 수 있다. 추가 실시예들에서, 서버(206)는 할당 해제된 토큰과 같은 다른 공지 토큰들도 수용할 수 있다.

[0072] 서버(206) 상의 토큰 생성기가 토큰들을 생성하는 데 사용되는 실시예들에서, 오프로드 판독 요청으로부터 반환되는 토큰은 서버(206)로부터 정보를 요청하는 임의의 클라이언트에 의해 사용될 수 있다. 따라서, 클라이언트(204)는 클라이언트(202)로부터 토큰을 수신할 수 있으며, 서버(206)로부터 오프로드 기록 작업들을 요청하기 위해 그 토큰을 사용할 수 있다. 토큰들이 클라이언트들 사이에서 전송되는 실시예들에서, 클라이언트들은 그러한 토큰들을 그들이 선택한 임의의 프로토콜 또는 전송 수단을 통해 전송할 수 있다. 클라이언트들 사이에서 토큰들을 전송하는 모드는 토큰들 자체와 무관하다. 따라서, 토큰들은 상이한 클라이언트들에 의해 설정될 수 있는 서버(206)와의 상이한 접속들 전반에서 사용 가능하다. 이러한 방식으로, 서버(206)는 파일 저장소(210)가 오프로드 판독 및 기록 작업들을 지원하는지의 여부에 관계없이 오프로드 판독 요청들 및 일부 오프로드 기록 요청들을 서비스할 수 있다. 오프로드 기록에서 클라이언트에 의해 서버(206)로 제공되는 토큰은 클라이언트가 현재 개방된 핸들을 갖는 파일로부터 획득되었을 필요가 없으며, 클라이언트가 액세스를 갖는 파일로부터 획득되었을 필요가 없다. 클라이언트는 적어도 토큰이 획득된 시간에 토큰이 획득된 파일(들)에 대한 액세스를 가졌던 다른 클라이언트를 통해 간접적으로 토큰을 획득할 수 있다. 접속을 통해 하나의 공유, 예를 들어 파일 공유로부터 오프로드 판독을 통해 클라이언트에 의해 획득된 토큰은 상이한 공유 또는 상이한 접속에 대해 클라이언트에 의해 발해진 오프로드 기록에서 성공적으로 사용될 수 있다.

[0073] 실시예들에서, 토큰들은 다수의 서버에 걸쳐 이용 가능할 수 있다. 즉, 토큰이 원래 서버로부터 오지 않은 경우에, 토큰은 그 서버 상에서 사용될 수 있다. 즉, 토큰이 서버(206), 서버(208) 또는 파일 저장소(210)에서 생성되는 경우, 그러한 토큰은 토큰을 승낙하기로 결정하는 임의의 서버 상에서 사용될 수 있다. 예를 들어, 위의 실시예에서, 클라이언트(202)에 의해 전송되는 오프로드 판독 요청은 파일 저장소(210)로 전송될 수 있으며, 이 파일 저장소는 오프로드 판독 요청에 대한 토큰을 생성한다. 나중에 클라이언트(202)가 서버(208)에 접속하는 경우, 클라이언트는 서버(206)에 대한 그의 접속에 의해 이전에 제공된 토큰을 이용하여 오프로드 기록 작업과 같은 다른 작업들을 수행할 수 있다. 이 예에서, 서버(208)는 토큰을 처음 생성한 파일 저장소(210)로

임의의 오프로드 기록 작업을 전달할 것이다. 이러한 방식으로, 토큰들이 다수의 서버들에 걸쳐 사용될 수 있다. 이 실시예는 예를 들어 공유 저장소를 사용하는 서버 클러스터가 클라이언트들에게 파일 서비스들을 제공하는 데 사용되는 상황들에서 사용될 수 있다.

[0074] 알 수 있듯이, 환경(200)에 대한 위의 설명은 본 명세서에서 설명되는 실시예들을 한정하는 것을 의도하지 않는다. 도 2 및 그의 설명은 일부 실시예들의 구현을 설명하는 것을 의도할 뿐이다. 다른 실시예들에서, 오프로드 작업들은 하나 이상의 파일 및 하나 이상의 토큰을 포함할 수 있다. 따라서, 실시예들은 도 2에 도시되고 설명된 것들로 한정되지 않는다. 예를 들어, 오프로드 판독은 단일 파일의 다수의 세그먼트 또는 다수의 파일의 판독을 제공할 수 있으며, 오프로드 판독 응답은 단일 토큰 또는 다수의 토큰을 포함할 수 있다. 유사하게, 일부 실시예들에서, 오프로드 기록 작업들은 하나 이상의 파일과 관련된 하나 이상의 토큰을 식별할 수 있다.

[0075] 도 3, 4, 5 및 6은 실시예들에 따른 동작 흐름들(300, 400)을 도시한다. 동작 흐름들(300, 400)은 임의의 적절한 컴퓨팅 환경에서 수행될 수 있다. 예를 들어, 동작 흐름들은 도 1 및 2에 도시된 바와 같은 시스템들 및 환경들에 의해 실행될 수 있다. 따라서, 동작 흐름들(300, 400)의 설명은 도 1 및 2의 컴포넌트들 중 적어도 하나를 참조할 수 있다. 그러나, 도 1 및 2의 컴포넌트들에 대한 임의의 그러한 참조는 설명의 목적을 위한 것일 뿐이며, 도 1 및 2의 구현들은 동작 흐름들(300, 400)에 대한 비한정적인 환경들이라는 것을 이해해야 한다.

[0076] 더구나, 동작 흐름들(300, 400)은 특정 순서로 순차적으로 도시되고 설명되지만, 다른 실시예들에서는 동작들이 상이한 순서, 상이한 시간 및/또는 병렬로 수행될 수 있다. 게다가, 일부 실시예들에서는 하나 이상의 동작이 생략되거나 결합될 수 있다.

[0077] 실시예들에서, 도 3에 도시된 흐름은 서버, 예를 들어 서버(206)(도 2) 상에서 실행되는 파일 서버에 의해 적어도 부분적으로 수행될 수 있다. 흐름(300)은 파일 서버에 접속하기 위한 요청이 수신되는 동작 302에서 시작된다. 동작 302에서 수신되는 요청은 파일 서버를 통해 액세스 가능한 원격 파일 시스템에 저장된 파일 정보에 액세스하기 위해 파일 서버와의 세션을 설정하기 위한 요청이다. 요청은 클라이언트, 예를 들어 클라이언트들(202, 204)(도 2)에 의해 전송될 수 있다. 동작 302에 이어서, 흐름(300)은 세션이 설정되었다는 것을 지시하는 응답이 전송되는 동작 304로 진행한다. 일부 실시예들에서, 동작 302 및 304에서 전송되는 요청 및 응답은 세션을 협상하기 위해 클라이언트와 서버 사이에서 교환되는 다수의 메시지 중 일부일 수 있다. 메시지들의 교환은 오프로드 파일 작업들을 서비스하기 위한 파일 서버의 능력을 포함하는 능력들의 교환을 포함할 수 있다.

[0078] 동작 흐름(300)은 동작 304로부터 동작 306으로 진행하여, 파일을 열기 위한 제2 요청이 수신된다. 요청은 파일 내의 정보에 액세스하기 위해 클라이언트에 의해 전송된다. 흐름은 동작 306으로부터 동작 308로 진행하여, 파일에 대한 액세스를 허가하는 응답이 클라이언트로 전송된다. 응답은 응답 내에서 파일 서버에 의해 제공되는 파일 식별자를 포함할 수 있다.

[0079] 이어서, 흐름(300)은 동작 310으로 진행하여, 오프로드 작업을 위한 요청이 수신된다. 오프로드 작업은 토큰에 의해 표현되는 파일 데이터를 요청하는 오프로드 판독 작업 또는 목적지 파일에 기록될 파일 데이터를 표현하는 토큰을 포함하는 오프로드 기록 작업일 수 있다. 작업이 오프로드 판독 작업인 경우, 흐름은 도 4에서 계속되는 A로 진행한다.

[0080] 도 4에 도시된 바와 같이, 흐름(300)은 결정 312로 진행하여, 오프로드 판독 작업에서 요청된 데이터 전부가 토큰에 의해 표현될 수 있는지를 결정한다. 이러한 결정(312)은 실시예들에서 다수의 상이한 결정을 포함할 수 있다. 예를 들어, 오프로드 판독 작업에서 요청되는 데이터의 임의 부분이 다른 클라이언트에 의한 독점 사용을 위해 잠겨 있는지에 대한 결정이 행해질 수 있다. 이러한 상황들에서, 서버는 요청된 데이터의 전부를 표현하는 토큰을 제공하지 못할 수 있다. 결정 312에서, 요청된 데이터의 전부가 토큰에 의해 표현되는 것이 가능하지 않은 것으로 결정되는 경우, 흐름은 동작 314로 향해 NO로 진행하여, 토큰을 갖는 줄여진 응답이 전송된다. 줄여진 응답은 응답 내에서 제공되고 있는 토큰이 오프로드 판독 요청에서 요청된 데이터 전부를 나타내지는 않는다는 것을 지시한다. 응답은 또한 어떤 범위의 데이터가 응답 내의 토큰에 의해 표현되는지를 지시할 수 있다. 동작 314에 이어서, 흐름은 316에 종료된다.

[0081] 결정 312에서, 요청된 데이터의 전부가 토큰에 의해 표현될 수 있는 것으로 결정되는 경우, 동작 318에서 오프로드 판독 요청에서 요청된 파일 데이터의 전부를 표현하는 토큰을 포함하는 응답이 전송된다. 이어서, 흐름(300)은 316에서 종료된다.

[0082] 일부 실시예들에서, 서버는 토큰들의 제공자가 아닐 수 있다. 이러한 실시예들에서는, 결정 312, 동작 314 및/또는 동작 318 대신에 점선들 내에 도시된 대안 동작들이 수행될 수 있다. 점선들 내의 동작들은 토큰들의 생

성이 하위 계층, 예를 들어 기초 파일 저장 레벨(들)에서 발생하는 실시예들에서 수행된다. 이러한 실시예들에서, 흐름(300)은 결정 312 대신에 조회 320으로 진행할 것이다. 조회 320에서, 오프로드 판독 작업이 줄여질 것인지를 결정한다. 오프로드 판독이 줄여져서, 예를 들어 요청된 데이터의 길이에 대한 조정이 수행될 경우, 프로세스(300)는 조정 또는 줄임 322를 향해 YES로 진행한다. 서버가 어떠한 조정도 행하지 않을 경우, 프로세스(300)는 NO로 진행하여, 요청을 변경하지 않는다(321). 이어서, 프로세스(300)는 동작 323으로 진행하여, 오프로드 판독 요청이 서버로부터 파일 저장 컴포넌트 또는 토큰들의 생성을 담당하는 다른 모듈과 같은 하위 계층으로 전달된다.

[0083] 동작 323에 이어서, 흐름(300)은 조회 324로 진행하여, 파일 저장소(또는 토큰들의 생성을 담당하는 다른 컴포넌트)가 줄여진 판독 동작을 수행할 것이다. 줄임이 발생하지 않는 경우, 프로세스(300)는 NO로 진행하여, 하위 계층에 의해 요청을 처리한다(325). 이어서, 토큰을 갖는 응답이 서버에 의해 수신되며(326), 동작 326에서 수신되는 응답은 오프로드 판독 작업에서 요청된 파일 데이터의 적어도 일부를 나타내는 토큰(들)을 포함한다. 토큰들은 파일 저장소에 의해 사용되는 임의의 적절한 포맷에 따라 포맷팅될 수 있다. 일 실시예에서, 토큰들은 사전 정의된 SCSI 포맷에 따라 포맷팅된다. 이어서, 토큰(들)을 포함하는 응답은 일 실시예에 따라 서버에 의해 클라이언트로 전송된다(329). 다른 실시예에서, 토큰(들)을 포함하는 응답은 예를 들어 파일 저장소 또는 다른 하위 계층으로부터 클라이언트로 직접 전송된다.

[0084] 조회 324로 돌아가서, 전술한 바와 같이, 토큰들의 생성을 담당하는 파일 저장소 또는 다른 컴포넌트가 요청된 데이터의 전부를 포함하는 응답을 제공할지를 결정한다. 예를 들어, 일 실시예에서, 파일 저장소는 오프로드 판독 요청에서 요청된 데이터의 전부를 제공하지 못할 수 있다. 전술한 바와 같이, 완전한 판독을 막는 저장 컨테이너에 대한 잠금 등을 포함하는 다양한 이유들이 줄여진 오프로드 판독 응답을 유발할 수 있다. 파일 저장소가 요청된 데이터의 일부만을 제공하는 경우, 프로세스(300)는 동작 327로 향해 YES로 진행하여, 하위 계층, 예를 들어 파일 저장소에 의해 요청이 처리되고, 줄여진 응답이 제공된다. 이어서, 일 실시예에서, 토큰을 포함하는 줄여진 응답이 서버에서 수신되어(328), 클라이언트로 전송된다(329). 다른 실시예들에서, 토큰을 포함하는 응답은 토큰들의 생성을 담당하는 파일 저장소 또는 다른 컴포넌트로부터 클라이언트로 직접 전송된다. 실시예들에서, 수신된 줄여진 응답(328)은 요청이 하나 이상의 계층에 의해 줄여졌다는 것을 지시한다. 예를 들어, 응답은 토큰이 오프로드 판독 요청에서 요청된 데이터의 전부보다 적은 데이터를 나타낸다는 것을 지시한다. 다른 실시예들에서, 줄여진 응답(328)은 요청이 줄여졌다는 지시를 제공하지 않는다. 이어서, 흐름(300)은 동작 316에서 종료된다. 다른 실시예(도시되지 않음)에서, 서버는 파일 저장소로부터 데이터를 수신하는 것에 응답하여 그리고 데이터가 이미 파일 저장소에 의해 줄여진 것으로 결정하는 경우에도 데이터를 더 줄이기로 결정할 수 있다. 그러한 실시예에서, 그러한 줄임은 동작 326 및 328 후에 그리고 토큰을 포함하는 줄여진 응답을 클라이언트로 전송하는 동작 329 전에 발생할 수 있다. 알 수 있듯이, 동작 320-329는 서버가 하위 계층 토큰 제공자에 대한 창구로서만 작용할 때 수행된다. 도 4에 도시된 실시예에서, 토큰 제공자는 파일 저장 시스템이다. 일부 실시예들에서, 파일 저장 시스템은 또한 오프로드 요청을 더 하위 레벨의 토큰 제공자에게 전달할 수 있다. 실시예들에서, 임의의 계층이 요청을 하위 계층으로 전송하기 전에 또는 후에 줄일 수 있다. 예를 들어, 실시예들은 요청을 하위 계층, 예를 파일 저장소로 전달하고, 하위 계층으로부터 응답을 수신하고, 이어서 서버가 토큰(들)을 갖는 응답을 클라이언트로 전송하기 전에 추가적인 줄임을 수행할지를 결정하는 서버를 제공한다. 그러나, 일부 실시예들에서는 요청을 하위 계층(들)으로 전송하기 전에 줄이는 것이 더 효율적일 수 있다. 전술한 바와 같이, 흐름(300)은 실시예들에 따라 수행될 수 있는 동작 흐름의 일례일 뿐이다. 실시예들은 도 3-5와 관련하여 제공되는 특정 설명으로 한정되지 않으며, 추가적인 동작들을 포함할 수 있다. 예를 들어, 도시된 동작 단계들은 다른 단계들로 결합되고/되거나 재배열될 수 있다. 게다가, 예를 들어 더 적거나 추가적인 단계들이 사용될 수 있다.

[0085] 도 3을 다시 참조하면, 동작 310에서 작업이 오프로드 기록 작업인 경우, 흐름은 도 5에서 계속되는 B로 진행한다. 알 수 있듯이, 오프로드 기록 작업은 데이터를 표현하는 토큰을 포함할 것이다. 토큰은 본 발명의 사상 및 범위로부터 벗어나지 않고서 본 명세서에서 설명되는 실시예들에 따라 다양한 타입의 저장 컨테이너들로부터 획득될 수 있다. 예를 들어, 토큰은 파일, 볼륨, 디스크, 볼륨 스냅샷, 디스크 스냅샷, 블로브 스토어 등으로부터 획득될 수 있다. 일 실시예에서, 토큰은 소스 파일로부터의 데이터를 토큰 내로 복사함으로써 생성된다. 추가 실시예에서, 토큰은 소스 파일로부터의 데이터를 토큰과 관련된 유지 영역 내에 복사함으로써 생성된다. 따라서, 생성된 토큰은 소스 파일과 무관하며, 논리적으로는 데이터의 그 자신의 판독 전용 컨테이너이다. 도 5에 도시된 바와 같이, 흐름은 동작 310으로부터 동작 330으로 진행하여, 오프로드 기록 작업에서 토큰과 관련된 데이터가 식별된다.

[0086] 동작 330에서 데이터가 식별된 후, 흐름(300)은 조회 331로 진행하여, 모든 요청된 데이터가 예를 들어 서버에 의해 목적지 파일에 기록될 수 있는지를 결정한다. 모든 요청된 데이터가 기록될 수 있는 경우, 프로세스(300)는 YES로 진행하여, 토큰에 의해 표현되는 데이터를 목적지 파일에 기록한다(332). 즉, 토큰에 의해 표현되는 데이터의 요청된 부분이 오프로드 기록 작업에서 요청된 특정 위치에 기록된다. 성공 또는 실패 및 일부 실시예들에서 목적지 파일에 기록되는 데이터의 양을 지시하는 응답이 동작 334에서 클라이언트로 전송된다. 이어서, 흐름(300)은 316에서 종료된다. 한편, 요청된 데이터의 전부가 예를 들어 목적지 파일에 기록되지 못하는 경우, 프로세스(300)는 NO로 진행하여, 줄여진 데이터를 기록하여(동작 336), 요청된 데이터의 일부를 기록한다. 이어서, 요청된 데이터의 일부가 기록되었음을 지시하는 줄여진 기록 응답이 동작 338에서 클라이언트로 전송된다. 이어서, 흐름(300)은 316에서 종료된다.

[0087] 전술한 바와 같이, 일부 실시예들에서, 서버는 토큰들의 제공자가 아닐 수 있다. 이러한 실시예들에서는, 도 5에 점선들 내에 도시된 대안 동작들이 동작들 330-338 대신에 수행될 수 있다. 점선들 내의 동작들은 토큰들의 생성이 하위 계층, 예를 들어 파일 저장 레벨 또는 그 아래에서 발생하는 실시예들에서 수행된다. 이러한 실시예들에서, 흐름(300)은 동작 330 대신에 조회 340으로 진행할 것이다. 조회 340에서, 오프로드 기록이 하위 계층, 예를 들어 기초 파일 저장소로 전달되기 전에 서버에서 줄여질지를 결정한다. 오프로드 기록이 줄여질 경우, 프로세스(300)는 YEW로 진행하여 조정 또는 줄인다(344). 서버가 어떠한 조정도 행하지 않을 경우, 흐름(300)은 NO로 진행하여, 오프로드 기록 요청을 변경하지 않는다(342). 이어서, 프로세스(300)는 동작 346으로 진행하여, (토큰을 갖는) 오프로드 기록 요청이 서버로부터 하위 계층, 예를 들어 오프로드 기록 요청의 처리를 담당하는 파일 저장 컴포넌트 또는 다른 모듈로 전달된다.

[0088] 오프로드 기록 요청을 하위 계층(들)으로 전달(346)한 후, 흐름(300)은 조회 348로 진행하여, 하위 계층, 예를 들어 파일 저장소가 줄여진 기록 작업을 수행할지를 결정한다. 줄임이 발생하지 않는 경우, 프로세스(300)는 동작 350을 향해 NO로 진행하여, 하위 계층이 기록 요청을 처리하며, 이는 예를 들어 요청 내의 토큰과 관련된 데이터를 식별하고 토큰에 의해 표현되는 데이터를 목적지 파일에 기록하는 것을 포함한다. 하위 계층이 줄여진 기록을 수행하는 경우, 프로세스(300)는 YES로 진행하여, 하위 계층이 줄이고 처리하며(동작 352), 요청된 데이터의 일부가 예를 들어 목적지 파일에 기록된다. 하위 계층(들)에 의한 오프로드 기록 요청의 처리에 이어서, 실시예들에서 토큰에 의해 표현되는 데이터가 목적지 파일에 성공적으로 기록되었는지는 물론, 얼마나 많은 데이터가 기록되었는지를 지시하는 오프로드 기록 응답이 하위 계층으로부터 서버에서 수신된다(354). 다른 실시예들에서, 오프로드 기록 응답은 오프로드 기록 요청의 처리를 담당하는 파일 저장소 또는 다른 컴포넌트로부터 클라이언트로 직접 전달된다.

[0089] 도 5로 돌아가서, 오프로드 기록 응답이 본 발명의 실시예들에 따라 서버에서 수신되는 경우(354), 이어서 조회 356은 데이터의 전부가 목적지에 기록되었는지 또는 기록이 줄여졌는지를 결정한다. 기록이 줄여져서, 요청된 데이터의 일부가 목적지에 기록된 경우, 프로세스(300)는 YES로 진행하여, 줄여진 기록 응답(358)을 클라이언트로 전송한다. 이어서, 흐름은 316에서 종료된다. 조회 356으로 돌아가서, 조회 356에서, 요청된 데이터의 전부가 목적지에 기록된 것으로 결정되는 경우, 프로세스(300)는 동작 360을 향해 NO로 진행하여, 기초 파일 저장 시스템으로부터의 응답이 클라이언트로 전송된다. 이어서, 흐름(300)은 316에서 종료된다.

[0090] 따라서, 실시예들에서, 요청이 하위 계층, 예를 들어 파일 저장소로 전달되는 경우에도 서버 자체에서 기록을 줄일 수 있다. 예를 들어, 서버는 실시예들에 따르면 기록 요청을 처리하기 위한 시간이 사전 결정된 임계치를 초과하는 경우에 기록을 줄일 수 있다. 실시예들에서, 서버는 요청을 하위 계층으로 전달하기 전에 기록을 줄인다. 예를 들어, 줄임은 오프로드 기록 요청을 파일 저장소로 전송하기 전에 발생할 수 있다. 다른 실시예들에서, 서버는 하위 계층, 예를 들어 파일 저장소에 의한 처리 후에 기록을 줄인다. 설명된 바와 같이, 흐름(300)은 실시예들에 따라 수행될 수 있는 동작 흐름의 일례일 뿐이다. 실시예들은 도 3-5와 관련하여 위에서 제공된 특정 설명으로 한정되지 않으며, 추가적인 동작들을 포함할 수 있다. 예를 들어, 도시된 동작 단계들은 다른 단계들로 결합되고/되거나 재배열될 수 있다. 게다가, 예를 들어 더 적거나 추가적인 단계들이 이용될 수 있다.

[0091] 도 6을 참조하면, 동작 흐름(400)은 오프로드 파일 작업들을 요청하기 위한 단계들을 나타낸다. 실시예들에서, 흐름(400)은 파일 시스템 내의 파일들에 액세스하기 위해 파일 서버와 통신하고 있는 클라이언트들(202, 204)(도 2)과 같은 클라이언트들 상의 재지향기들에 의해 수행될 수 있다. 실시예들에서, 클라이언트는 SMB 프로토콜의 일 버전 또는 NFS의 일 버전과 같은 파일 액세스 프로토콜을 이용하여 파일 서버와 통신한다.

[0092] 흐름(400)은 파일 서버에 접속하기 위한 요청이 전송되는 동작 402에서 시작된다. 동작 402에서 전송되는 요청

은 파일 서버를 통해 액세스 가능한 파일 시스템 상에 저장된 파일 정보에 액세스하기 위해 파일 서버와의 세션을 설정하기 위한 요청이다. 요청은 서버, 예를 들어 서버(206)(도 2) 상의 파일 서버로 전송될 수 있다. 요청은 SMB 또는 NFS의 일 버전과 같은 파일 액세스 프로토콜에 따라 포맷팅된다.

[0093] 동작 402에 이어서, 흐름(400)은 동작 404로 진행하여, 세션이 설정되었음을 지시하는 응답이 수신된다. 일부 실시예들에서, 동작 402 및 404는 세션을 협상하기 위해 클라이언트와 서버 사이에서 교환되는 다수의 메시지 중 일부일 수 있다. 메시지들의 교환은 오프로드 작업들을 서비스하기 위한 파일 서버의 능력을 포함하는 능력들의 교환을 포함할 수 있다.

[0094] 동작 흐름은 동작 404로부터 동작 406으로 진행하여, 파일을 개방하기 위한 요청이 전송된다. 흐름(400)은 동작 406으로부터 동작 408로 진행하여, 파일에 대한 액세스를 허가하는 응답이 수신된다. 흐름은 동작 408로부터 동작 410으로 진행하여, 클라이언트가 오프로드 판독 요청을 전송할 것이다. 오프로드 판독 요청은 요청되는 파일 데이터의 일부를 지시한다. 오프로드 판독 요청은 또한 본질적으로, 데이터가 오프로드 판독 요청에 응답하여 전송되는 토큰에 의해 표현될 것을 요청한다. 동작 412에서, 토큰을 갖는 오프로드 판독 요청이 수신된다.

[0095] 실시예들에서, 동작 410에서 오프로드 판독 요청을 전송하는 클라이언트는 그가 판독 요청에서 어떤 데이터를 요청할 수 있는지에 관하여 일부 제한들을 가질 수 있다. 예를 들어, 실시예들에서, 클라이언트가 일부 데이터를 국지적으로 캐싱(caching)한 경우, 클라이언트는 그의 판독 요청을 전송하기 전에 임의의 캐싱된 "오염된" 데이터를 방출할 것이다. 오프로드 판독을 전송하기 전에 캐싱된 오염된 데이터를 서버로 방출하지 못하는 것은 오프로드 판독이 손상된 데이터를 표현하는 토큰을 제공하게 할 수 있다. 일부 실시예들에서, 클라이언트는 오프로드 판독 자체를 줄일 수 있다. 즉, 클라이언트는 파일 데이터의 전체 범위를 요청하지 않을 수 있으며, 따라서 캐싱된 오염된 데이터를 배제할 수 있다.

[0096] 동작 412에 이어서, 클라이언트는 동작 414에서 오프로드 기록 요청을 전송할 수 있다. 흐름(400)은 동작 414가 동작 412에 바로 이어지는 것으로 도시하지만, 이것은 설명의 목적을 위한 것일 뿐이라는 것을 이해할 수 있다. 다른 실시예들에서, 흐름(400)을 수행하는 클라이언트가 소정의 다른 수단으로부터 데이터를 표현하는 토큰을 수신하는 경우, 동작 414에서 전송되는 오프로드 기록 요청은 동작 410에서 전송되는 요청과 같은 임의의 오프로드 판독 요청들 전에 수행될 수 있다.

[0097] 실시예들에서, 동작 414에서 오프로드 기록 요청을 전송하는 클라이언트는 또한 그가 오프로드 기록 요청에서 어떤 데이터를 전송할 수 있는지에 관하여 일부 제한들을 가질 수 있다. 오프로드 기록이 오염된 데이터를 캐싱한 목적지 오프셋에 토큰 데이터를 기록하도록 허가되는 경우, 캐싱된 오염된 데이터는 나중에 캐싱된 오염된 데이터가 저장소에 다시 기록될 때 오프로드 기록에 의해 기록된 데이터를 잘못 덮어쓰기할 것이다. 오프로드 기록을 전송하는 클라이언트는 클라이언트가 캐싱된 오염된 데이터를 유지하고 있는 임의의 오프셋에 대한 기록 요청의 전송을 피할 수 있다. 실시예들에 따르면, 클라이언트는 오프로드 기록 자체를 줄일 수 있거나, 클라이언트는 오프로드 기록 목적지 오프셋들과 오버랩되는 캐싱된 오염된 데이터를 폐기할 수 있거나, 클라이언트는 오프로드 기록에 실패할 수 있다.

[0098] 오프로드 기록 요청이 동작 414에서 전송된 후, 흐름(400)은 동작 416으로 진행하여, 오프로드 기록 요청에 대한 응답이 수신된다. 응답은 오프로드 기록 요청에서 전송된 토큰과 관련된 데이터가 목적지 파일에 성공적으로 기록되었는지를 지시할 수 있다. 데이터가 부분적으로 기록되었을 수 있는 실시예들에서, 동작 416에서 수신되는 응답은 성공적으로 기록된 데이터의 부분을 지시하고, 데이터의 전부가 목적지 파일 내에 기록되지는 않았다는 것을 지시할 것이다.

[0099] 일부 실시예들에서, 점선들 내에 도시된 동작 418은 흐름(400)의 실행 동안의 임의의 시간에 수행된다. 도 6에 도시된 실시예에서, 동작 418은 동작 416 뒤에 도시되지만, 실시예들은 이러한 순서로 반드시 한정되지는 않는다. 동작 418은 흐름(400)을 수행하는 클라이언트가 토큰과 관련된 실제 데이터를 요구하는 실시예들에서 전송된다. 즉, 클라이언트는 오프로드 판독 요청을 전송함으로써 또는 소정의 다른 수단에 의해 수신되는 토큰을 소유할 수 있다. 그러나, 클라이언트는 예를 들어 실제 데이터를 요청하는 애플리케이션에 데이터를 제공하기 위하여 토큰과 관련된 실제 데이터를 필요로 할 수 있다. 이러한 실시예들에서, 동작 418은 토큰과 관련된 데이터의 일부를 검색하기 위한 요청을 전송할 수 있다. 클라이언트는 데이터를 검색하기 위한 요청을 실제 데이터를 제공할 수 있는 서버로 전송할 것이다. 이에 응답하여, 클라이언트는 실제 데이터를 수신할 것이고, 그 데이터를 실제 데이터를 요청하는 애플리케이션에 제공할 수 있다. 이것은 일부 실시예들에서 일부 클라이언트들에 의해 수행될 수 있는 하나의 추가적인 동작일 뿐이다. 흐름(400)은 420에서 종료된다.

[0100] 전술한 바와 같이, 흐름들(300, 400)은 실시예들에 따라 수행될 수 있는 동작 흐름들의 일부 예들일 뿐이다. 실시예들은 도 3-6과 관련하여 위에서 제공된 특정 설명으로 한정되지 않으며, 추가적인 동작들을 포함할 수 있다. 게다가, 도시된 동작 단계들은 다른 단계들로 결합되고/되거나 재배열될 수 있다. 게다가, 예를 들어 더 적거나 추가적인 단계들이 이용될 수 있다.

[0101] 도 7은 본 명세서에서 설명되는 실시예들을 구현하는 데 사용될 수 있는 범용 컴퓨터 시스템(700)을 도시한다. 컴퓨터 시스템(700)은 컴퓨팅 환경의 일례일 뿐이며, 컴퓨터 및 네트워크 아키텍처들의 이용 또는 기능의 범위에 관하여 어떠한 한정도 시사하지 않는 것을 의도한다. 컴퓨터 시스템(700)은 예시적인 컴퓨터 시스템(700) 내에 도시된 컴포넌트들 중 어느 하나 또는 조합에 관하여 어떠한 의존성 또는 요구도 갖지 않는 것으로 해석되어야 한다. 실시예들에서, 시스템(700)은 도 1과 관련하여 전술한 클라이언트 및/또는 서버로 사용될 수 있다.

[0102] 시스템(700)은 그의 가장 기본적인 구성에서 통상적으로 적어도 하나의 처리 유닛(702) 및 메모리(704)를 포함한다. 컴퓨팅 장치의 정확한 구성 및 타입에 따라, 메모리(704)는 휘발성(RAM 등), 비휘발성(ROM, 플래시 메모리 등) 또는 소정 조합일 수 있다. 이러한 가장 기본적인 구성은 도 7에 점선(706) 내에 도시된다. 시스템 메모리(704)는 저장소(708)와 같은 저장소를 갖는 파일 저장 시스템 내에 저장될 수 있는 데이터(729)를 표현하는 토큰들(723)과 같은 데이터를 저장한다.

[0103] 본 명세서에서 사용되는 바와 같은 컴퓨터 판독 가능 매체라는 용어는 컴퓨터 저장 매체를 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독 가능 명령어들, 데이터 구조들, 프로그램 모듈들 및 다른 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술에서 구현되는 휘발성 및 비휘발성, 이동식 및 비이동식 매체를 포함할 수 있다. 시스템 메모리(704), 이동식 저장 장치 및 비이동식 저장 장치(708)는 모두 컴퓨터 저장 매체의 예들(즉, 메모리 저장 장치)이다. 컴퓨터 저장 매체는 RAM, ROM, 전기적으로 소거 가능한 판독 전용 메모리(EEPROM), 플래시 메모리 또는 다른 메모리 기술, CD-ROM, 디지털 다기능 디스크(DVD) 또는 다른 광학 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 다른 자기 저장 장치, 또는 정보를 저장하는 데 사용될 수 있고 컴퓨팅 장치(700)에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있지만 이에 한정되지 않는다. 임의의 그러한 컴퓨터 저장 매체는 장치(700)의 일부일 수 있다. 컴퓨팅 장치(700)는 또한 키보드, 마우스, 펜, 사운드 입력 장치, 터치 입력 장치 등과 같은 입력 장치(들)(714)를 가질 수 있다. 디스플레이, 스피커, 프린터 등과 같은 출력 장치(들)(716)도 포함될 수 있다. 전술한 장치들은 예들이고, 다른 것들도 사용될 수 있다.

[0104] 본 명세서에서 사용되는 바와 같은 컴퓨터 판독 가능 매체라는 용어는 통신 매체도 포함할 수 있다. 통신 매체는 반송파 또는 다른 전송 메커니즘과 같은 피변조 데이터 신호 내의 컴퓨터 판독 가능 명령어들, 데이터 구조들, 프로그램 모듈들 또는 다른 데이터에 의해 구현될 수 있으며, 임의의 정보 전달 매체를 포함한다. 용어 "피변조 데이터 신호"는 신호 내에 정보를 인코딩하는 방식으로 하나 이상의 특성이 설정 또는 변경된 신호를 설명할 수 있다. 한정이 아니라 예로서, 통신 매체는 유선 네트워크 또는 직접 유선 접속과 같은 유선 매체, 및 음향, 무선 주파수(RF), 적외선 및 다른 무선 매체와 같은 무선 매체를 포함할 수 있다.

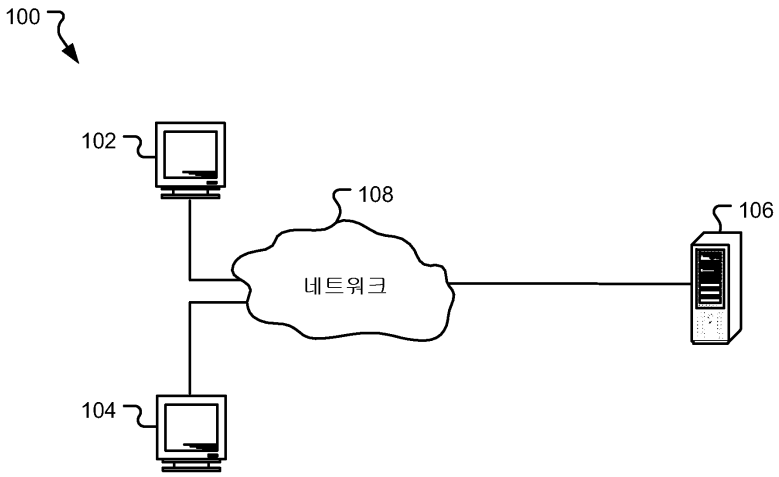
[0105] 본 명세서 전반에서 "하나의 실시예" 또는 "일 실시예"에 대한 참조가 이루어졌으며, 이는 설명되는 특정 특징, 구조 또는 특성이 적어도 하나의 실시예 내에 포함된다는 것을 의미한다. 따라서, 그러한 표현들의 사용은 하나보다 많은 실시예를 지칭할 수 있다. 더구나, 설명되는 특징들, 구조들 또는 특성들은 하나 이상의 실시예에서 임의의 적절한 방식으로 결합될 수 있다.

[0106] 그러나, 관련 분야의 기술자는 실시예들이 특정 상세들 중 하나 이상의 상세 없이도 또는 다른 방법, 자원, 재료 등을 이용하여 실시될 수 있다는 것을 인식할 수 있다. 다른 예들에서는, 단지 실시예들의 양태들을 불명확하게 하지 않기 위해, 공지 구조들, 자원들 또는 동작들은 상세히 도시되거나 설명되지 않았다.

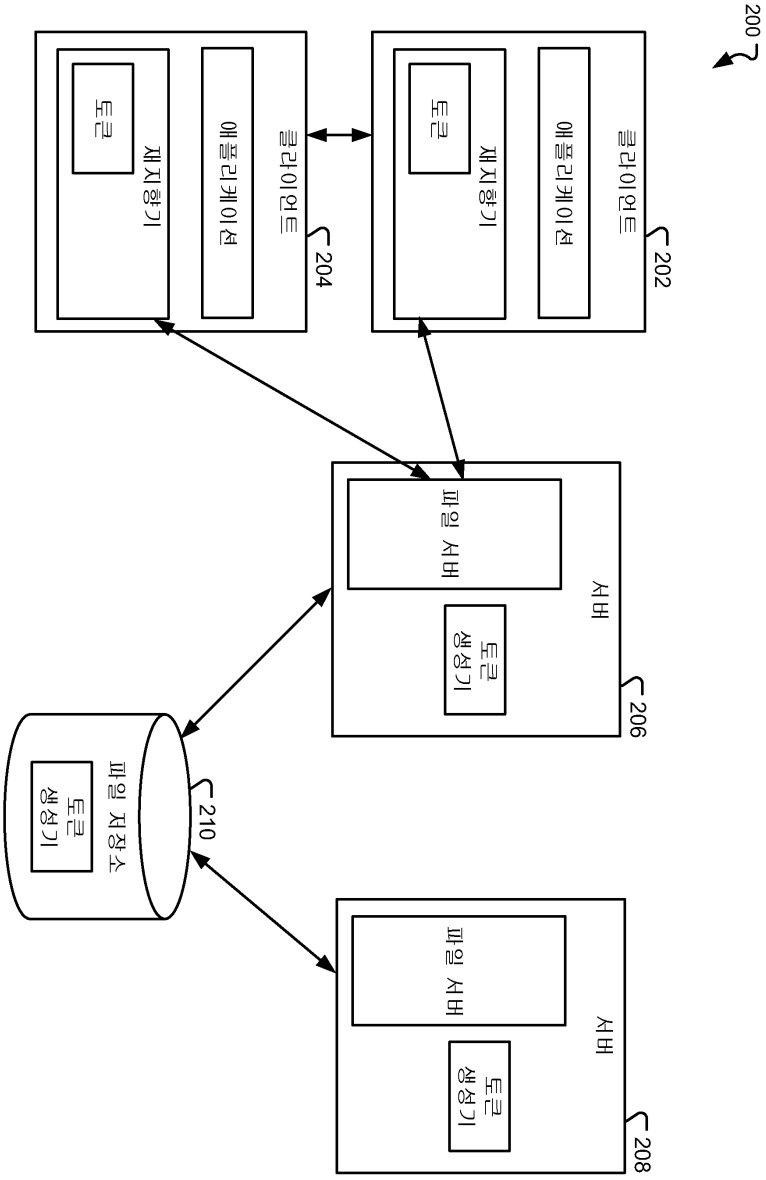
[0107] 예시적인 실시예들 및 응용들이 도시되고 설명되었지만, 실시예들은 전술한 바로 그 구성 및 자원들로 한정되지 않는다는 것을 이해해야 한다. 청구되는 실시예들의 범위로부터 벗어나지 않고서 본 명세서에서 개시되는 방법들 및 시스템들의 배열, 동작 및 상세들에 있어서 이 분야의 기술자에게 명백한 다양한 수정들, 변경들 및 변형들이 이루어질 수 있다.

도면

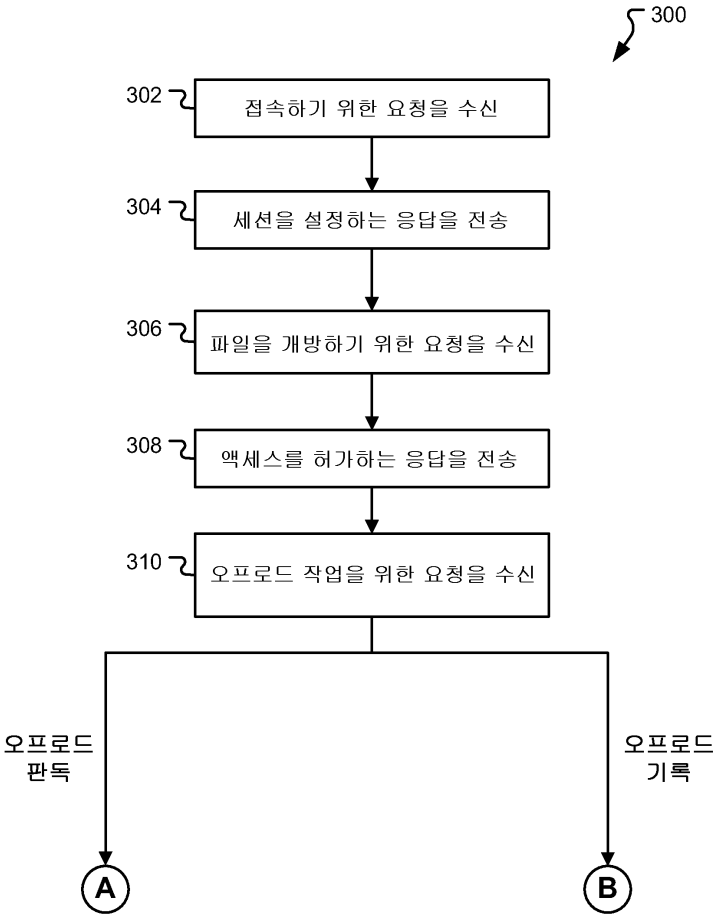
도면1



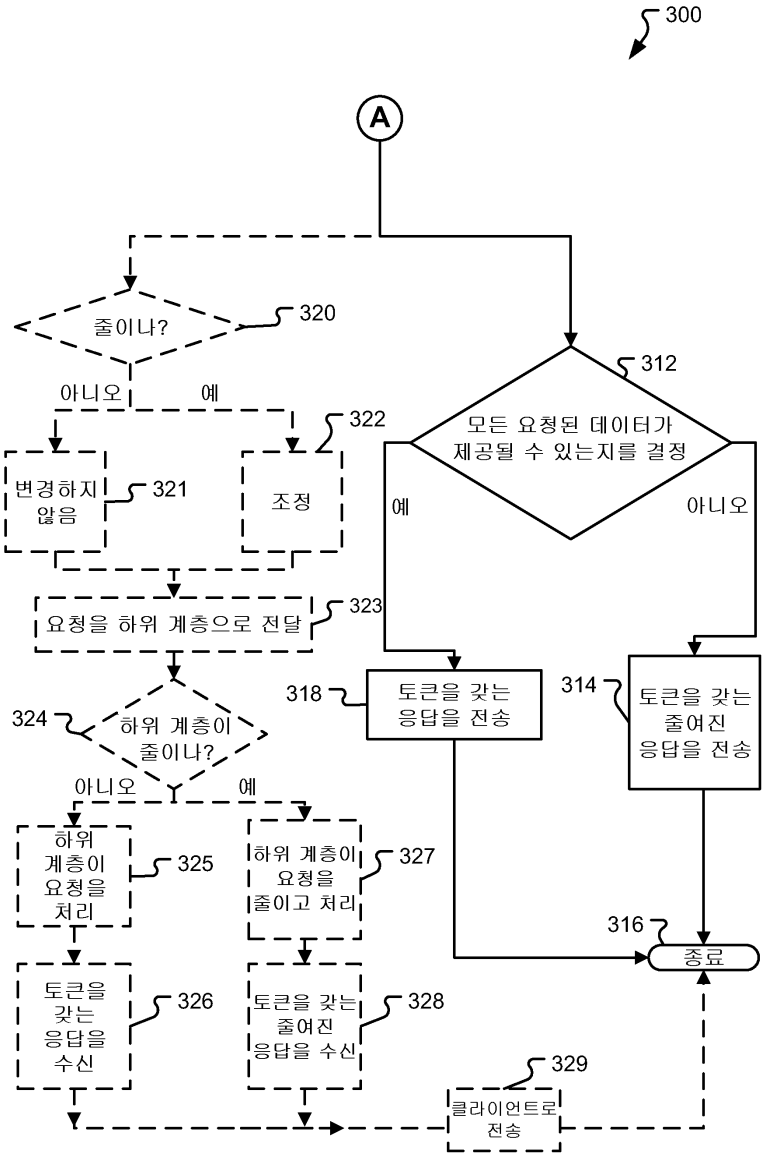
도면2


200
클라이언트
202
애플리케이션
재지향기
토큰
클라이언트
204
애플리케이션
재지향기
토큰
서버
206
파일 서버
토큰 생성기
파일 저장소
210
토큰 생성기
서버
208
파일 서버
토큰 생성기

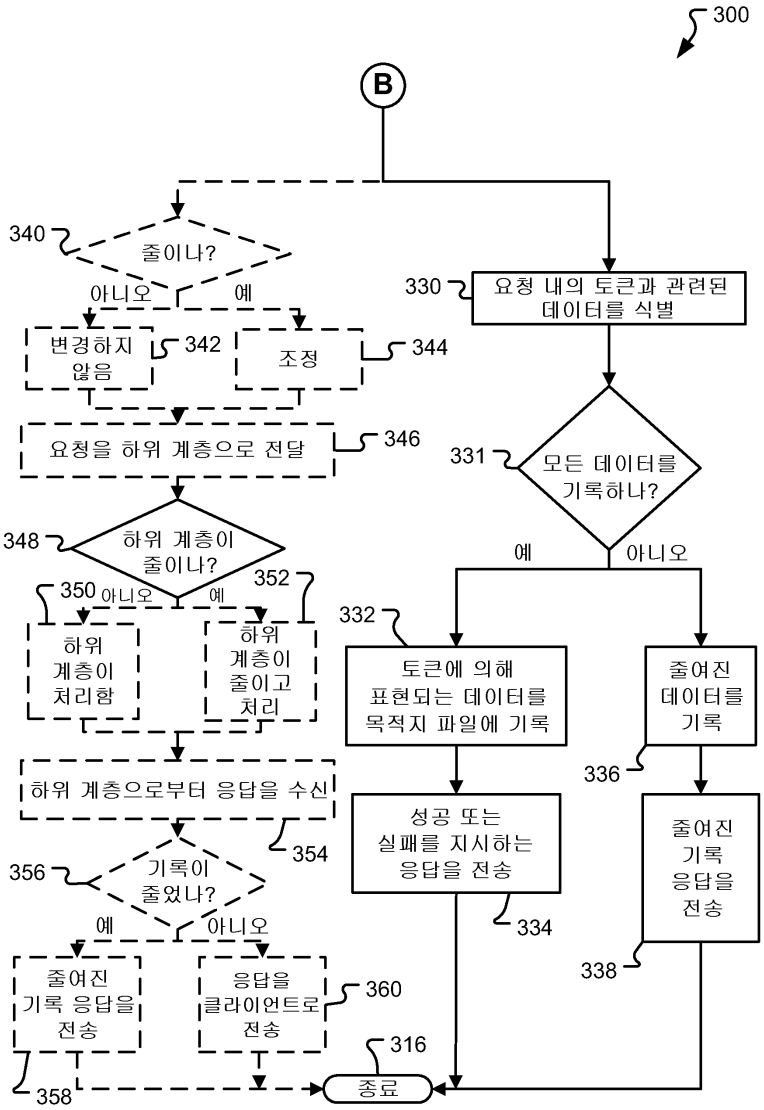
도면3


300
302
접속하기 위한 요청을 수신
304
세션을 설정하는 응답을 전송
306
파일을 개방하기 위한 요청을 수신
308
액세스를 허가하는 응답을 전송
310
오프로드 작업을 위한 요청을 수신
오프로드
판독
오프로드
기록
A
B

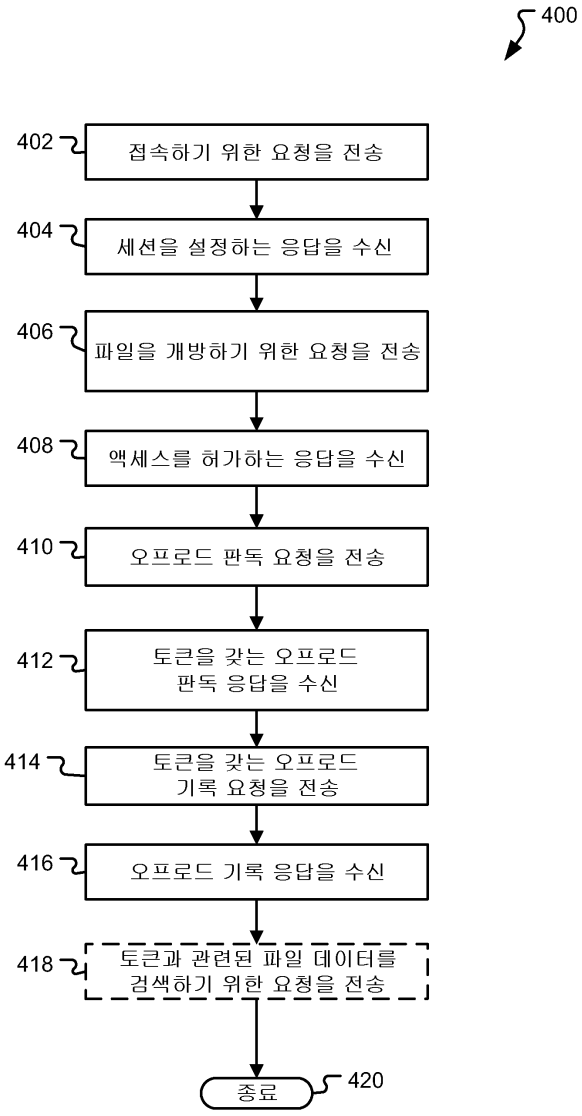
도면4


300
A
줄이나?
320
아니오
예
322
변경하지 않음
321
조정
요청을 하위 계층으로 전달
323
324
하위 계층이 줄이나?
아니오
예
하위 계층이 요청을 처리
325
토큰을 갖는 응답을 수신
326
하위 계층이 요청을 줄이고 처리
327
토큰을 갖는 줄여진 응답을 수신
328
312
모든 요청된 데이터가 제공될 수 있는지를 결정
예
아니오
318
토큰을 갖는 응답을 전송
314
토큰을 갖는 줄여진 응답을 전송
316
종료
329
클라이언트로 전송

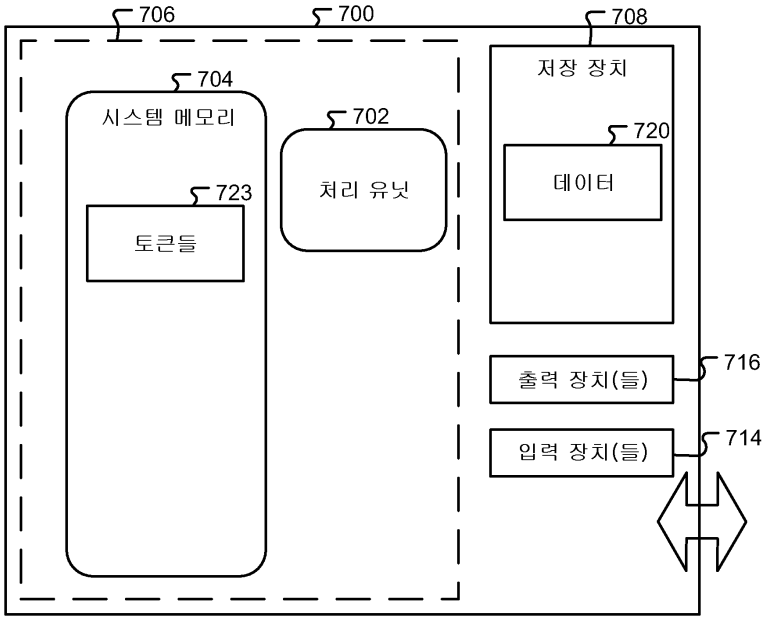
도면5


300
B
340
줄이나?
아니오
예
330
요청 내의 토큰과 관련된 데이터를 식별
변경하지 않음
342
조정
344
요청을 하위 계층으로 전달
346
331
모든 데이터를 기록하나?
348
하위 계층이 줄이나?
예
아니오
350
아니오
예
352
332
하위 계층이 처리함
하위 계층이 줄이고 처리
토큰에 의해 표현되는 데이터를 목적지 파일에 기록
줄여진 데이터를 기록
336
하위 계층으로부터 응답을 수신
354
성공 또는 실패를 지시하는 응답을 전송
줄여진 기록 응답을 전송
356
기록이 줄었나?
예
아니오
334
338
줄여진 기록 응답을 전송
응답을 클라이언트로 전송
360
316
358
종료

도면6


400
402
접속하기 위한 요청을 전송
404
세션을 설정하는 응답을 수신
406
파일을 개방하기 위한 요청을 전송
408
액세스를 허가하는 응답을 수신
410
오프로드 판독 요청을 전송
412
토큰을 갖는 오프로드
판독 응답을 수신
414
토큰을 갖는 오프로드
기록 요청을 전송
416
오프로드 기록 응답을 수신
418
토큰과 관련된 파일 데이터를
검색하기 위한 요청을 전송
종료
420

도면7


706
700
708
저장 장치
704
시스템 메모리
702
720
데이터
723
처리 유닛
토큰들
716
출력 장치(들)
714
입력 장치(들)