



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2010-0033966  
(43) 공개일자 2010년03월31일

(51) Int. Cl.

G06F 15/16 (2006.01) G06F 17/30 (2006.01)  
H04L 7/00 (2006.01)

(21) 출원번호 10-2009-7026257

(22) 출원일자 2008년06월20일

심사청구일자 없음

(85) 번역문제출일자 2009년12월16일

(86) 국제출원번호 PCT/US2008/067636

(87) 국제공개번호 WO 2009/002831

국제공개일자 2008년12월31일

(30) 우선권주장

11/766,985 2007년06월22일 미국(US)

(71) 출원인

마이크로소프트 코포레이션

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원  
마이크로소프트 웨이

(72) 발명자

사가르, 아카쉬, 제이.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로  
소프트 웨이 마이크로소프트 코포레이션 국제특허  
부 내

모로미사토, 조지, 피.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로  
소프트 웨이 마이크로소프트 코포레이션 국제특허  
부 내

(뒷면에 계속)

(74) 대리인

양영준, 백만기

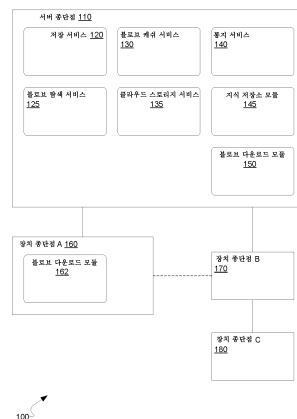
전체 청구항 수 : 총 20 항

(54) 서버 지원형 피어-투-피어 동기화

(57) 요약

중앙집중형 및 분산형 동기화 시스템 및 통신 토폴로지들의 구성요소들을 사용하여 종단점들 간에 데이터를 동기화시키는 시스템들 및 방법들이 개시되어 있다. 이러한 시스템들 및 방법들은 어떤 경우들에 데이터의 어떤 일부분을 중앙집중형 종단점과 동기화시킬 수 있는 반면, 데이터의 다른 일부분은 분산형 방식으로 다른 종단점들과 직접 동기화된다. 이러한 시스템들 및 방법들은 종단점들 간의 데이터 동기화를 돕는 각종의 협동적 기능을 포함할 수 있다.

대표도 - 도1



(72) 발명자

**청, 리차드, 이우-사이**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제특허부 내

**오지에, 레이몬드, 이.**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제특허부 내

**오지에, 잭, 이.**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제특허부 내

**리드, 데이비드, 리차드**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제특허부 내

**버날, 마이클, 스티븐**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제특허부 내

**페도로브, 블라디미르, 드미트리**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제특허부 내

**안나말라이, 무두카루판**

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제특허부 내

## 특허청구의 범위

### 청구항 1

서버 종단점으로부터 메타데이터를 획득하는 단계 - 상기 메타데이터는 블로브(blob)를 식별해주고 상기 블로브는 상기 메타데이터에 포함되어 있지 않음 -,

상기 메타데이터를 로컬 메타데이터와 병합시키는 단계,

상기 블로브와 연관된 적어도 하나의 위치 지정자를 포함하는 세트가 있는지 블로브 탐색 서비스에 질문을 하는 단계, 및

제1 종단점에서, 상기 적어도 하나의 위치 지정자의 세트 내의 특정의 위치 지정자에 의해 식별되는 제2 종단점으로부터 상기 블로브의 일부분을 검색하는 단계를 포함하는 방법.

### 청구항 2

제1항에 있어서, 상기 블로브의 상기 일부분이 처음에 검색될 수 없을 때 블로브 캐쉬 서비스에 상기 블로브의 적어도 제2 일부분을 저장하라고 요청하는 단계를 더 포함하는 방법.

### 청구항 3

제1항에 있어서, 상기 블로브의 일부분을 검색하는 상기 단계는,

상기 제2 종단점과 다른 제3 종단점으로부터 상기 블로브의 상기 일부분과 다른 상기 블로브의 제2 일부분을 검색하는 단계를 더 포함하는 방법.

### 청구항 4

제1항에 있어서, 상기 검색하는 단계는 제1 블로브 검색 메커니즘을 사용하여 상기 블로브의 상기 일부분을 검색하는 블로브 다운로드 모듈에 의해 구현되고,

상기 블로브 다운로드 모듈은 상기 제1 블로브 검색 메커니즘과 다른 제2 블로브 검색 메커니즘을 사용하여 상기 블로브의 제2 일부분 및 제2 블로브의 일부분 중 적어도 하나를 검색하는 기능을 갖는 방법.

### 청구항 5

제1항에 있어서, 상기 블로브의 상기 일부분이 검색된 후에 상기 블로브 탐색 서비스에 상기 블로브의 상기 일부분이 상기 제1 종단점으로부터 검색될 수 있다고 통지하는 단계를 더 포함하는 방법.

### 청구항 6

제1항에 있어서, 제3 종단점으로부터의 상기 블로브의 상기 일부분에 대한 요청에 응답하여 상기 블로브의 상기 일부분을 상기 제3 종단점에 제공하는 단계를 더 포함하는 방법.

### 청구항 7

제1항에 있어서, 상기 메타데이터가 상기 서버 종단점에서 업데이트되었다는 통지를 상기 제1 종단점에서 수신하는 단계를 더 포함하며,

상기 통지가 상기 획득하는 단계(operation), 상기 병합시키는 단계, 상기 질문을 하는 단계 및 상기 검색하는 단계를 개시하게 하는 방법.

### 청구항 8

제1항에 있어서, 상기 로컬 메타데이터와 연관된 데이터에 로컬 변경을 수행하는 단계,

상기 로컬 메타데이터를 업데이트하여 업데이트된 로컬 메타데이터를 형성하고 상기 업데이트된 로컬 메타데이터 및 상기 블로브의 제2 일부분이 상기 로컬 변경을 포함하도록 상기 블로브의 상기 제2 일부분을 상기 제1 종단점에서 업데이트시키는 단계, 및

상기 업데이트된 로컬 메타데이터를 상기 서버 종단점으로 전달하는 단계를 더 포함하는 방법.

#### 청구항 9

서버 종단점에서, 제1 종단점으로부터 메타데이터를 획득하는 단계 - 상기 메타데이터는 블로그를 식별해주고 상기 블로그는 상기 메타데이터에 포함되어 있지 않음 -,

상기 메타데이터를 상기 서버 종단점에 의해 유지되는 서버 메타데이터와 병합시키는 단계, 및

상기 제1 종단점을 식별해주는 위치로 블로그 탐색 서비스를 업데이트시키는 단계를 포함하는 방법.

#### 청구항 10

제9항에 있어서, 상기 서버 메타데이터를 상기 제1 종단점과 다른 제2 종단점으로 전달하는 단계를 더 포함하는 방법.

#### 청구항 11

제9항에 있어서, 상기 블로그의 일부분이 검색될 수 있는 곳을 식별해주는 위치 지정자에 대한 요청을 요청자로부터 수신하는 단계,

상기 블로그 탐색 서비스를 사용하여 상기 위치 지정자를 식별하는 단계, 및

상기 위치 지정자를 상기 요청자에게 전달하는 단계를 더 포함하는 방법.

#### 청구항 12

제11항에 있어서, 상기 위치 지정자가 상기 제1 종단점과 연관된 제1 지식 특성들을 사용하여 상기 블로그 탐색 서비스에 의해 식별되고, 제2 종단점과 연관되어 있고 상기 블로그의 제2 일부분이 검색될 수 있는 곳을 식별해주는 제2 위치 지정자가 식별되지 않는데, 그 이유는 상기 제2 종단점과 연관된 제2 지식 특성들이 상기 제1 지식 특성들보다 덜 바람직하기 때문인 방법.

#### 청구항 13

제9항에 있어서, 제2 종단점을 식별해주는 제2 위치를 받아들이는(accept) 단계 - 상기 제2 종단점은 상기 블로그의 제2 일부분을 저장하고 상기 블로그의 상기 제2 일부분은 상기 제2 종단점으로부터 검색될 수 있음 -,

상기 블로그가 검색될 수 있는 곳을 식별해주는 적어도 하나의 위치 지정자에 대한 요청을 수신하는 단계,

상기 위치와 연관된 제1 위치 지정자 및 상기 제2 위치와 연관된 제2 위치 지정자를 식별해주는 단계, 및

상기 요청에 응답하여 상기 제1 위치 지정자 및 상기 제2 위치 지정자를 전달하는 단계를 더 포함하는 방법.

#### 청구항 14

제9항에 있어서, 상기 블로그의 일부분을 캐싱하라는 요청을 받아들이는 단계,

상기 블로그의 상기 일부분을 검색하는 단계,

상기 블로그의 상기 일부분을 블로그 캐쉬에 저장하는 단계, 및

상기 블로그 캐쉬를 식별해주는 제2 위치로 상기 블로그 탐색 서비스를 업데이트시키는 단계를 더 포함하는 방법.

#### 청구항 15

제14항에 있어서, 제공측 종단점(providing endpoint)이 상기 제공측 종단점과 연관된 지식 특성들을 사용하여 식별되고,

상기 블로그의 상기 일부분이 상기 제공측 종단점으로부터 검색되는 방법.

#### 청구항 16

제9항에 있어서, 상기 블로그의 일부분을 비일시적인 클라우드 스토리지 데이터 저장소(non-transient cloud storage data store)에 저장하라는 요청을 받아들이는 단계,

상기 블로브의 상기 일부분을 검색하는 단계, 및

상기 블로브의 상기 일부분을 상기 클라우드 스토리지 데이터 저장소에 저장하는 단계를 더 포함하는 방법.

#### 청구항 17

제9항에 있어서, 상기 병합시키는 단계 이후에, 상기 서버 메타데이터가 수정되었다는 것을 가입측 종단점에 통지하는 단계, 및

상기 서버 메타데이터를 상기 가입측 종단점으로 전달하는 단계를 더 포함하는 방법.

#### 청구항 18

제14항에 있어서, 상기 검색하는 단계는 제1 블로브 검색 메커니즘을 사용하여 상기 블로브의 상기 일부분을 검색하는 블로브 다운로드 모듈에 의해 구현되고,

상기 블로브 다운로드 모듈은 상기 제1 블로브 검색 메커니즘과 다른 제2 블로브 검색 메커니즘을 사용하여 상기 블로브의 제2 일부분 및 제2 블로브의 일부분 중 적어도 하나를 검색하는 기능을 갖는 방법.

#### 청구항 19

제14항에 있어서, 상기 블로브의 상기 일부분을 상기 블로브 캐쉬로부터 상기 제1 종단점과 다른 제2 종단점으로 전달하는 단계를 더 포함하는 방법.

#### 청구항 20

시스템으로서,

블로브 탐색 서비스, 저장 서비스, 및 블로브 캐쉬 서비스를 포함하며,

상기 저장 서비스는,

제1 종단점으로부터 메타데이터를 획득하고 - 상기 메타데이터는 블로브를 식별해주고 상기 블로브는 상기 메타데이터에 포함되어 있지 않음 -,

상기 메타데이터를 상기 저장 서비스에 의해 유지되는 서버 메타데이터와 병합시키며,

상기 제1 종단점을 식별해주는 위치로 상기 블로브 탐색 서비스를 업데이트시키도록 구성되어 있고,

상기 블로브 캐쉬 서비스는,

상기 블로브의 일부분을 캐싱하라는 요청을 받아들이고,

상기 제1 종단점으로부터 상기 블로브의 상기 일부분을 검색하며,

상기 블로브 캐쉬 서비스와 연관된 스토리지를 사용하여 상기 블로브의 상기 일부분을 저장하고,

상기 블로브 캐쉬 서비스를 식별해주는 제2 위치로 상기 블로브 탐색 서비스를 업데이트시키도록 구성되어 있는 것인 시스템.

### 명세서

#### 배경 기술

[0001]

서로 다른 컴퓨팅 장치들과 같은 서로 다른 종단점들 간에 다양한 방식으로 또 각종의 연결 토폴로지들을 사용하여 데이터가 동기화될 수 있다. 예를 들어, 어떤 시스템들 또는 기법들은 컴퓨팅 서버들과 같은 하나의 (또는 다수의) 중앙집중형 종단점(centralized endpoints)과 동기화하기 위해, 종단점들에 의존할 수 있다. 다른 시스템들 또는 기법들에서, 종단점들은 하나 이상의 분산형 또는 피어-투-피어 토폴로지들(peer-to-peer topologies)을 비롯한 각종의 방식으로 서로 직접 통신을 할 수 있다.

[0002]

서로 다른 종단점 배열들 및 서로 다른 통신 토폴로지들 각각은 그 자신의 이점들 및 단점들을 가질 수 있다. 단지 한 예로서, 종단점들이, 어쩌면 중앙집중형, 즉 "서버" 종단점을 사용하지 않고서, 서로 직접 데이터를 동기화시키는 시스템은 각종의 이점들을 가질 수 있다. 예를 들어, 이러한 시스템에서, 종단점들은 동기화 중

인 중단점들보다 멀리 떨어져 위치할지도 모르는 중앙 서버와 먼저(또는 어쩌면 결코) 데이터를 동기화시킬 필요 없이 물리적으로 또는 논리적으로 "가까운" 다른 중단점들과 통신을 하여 데이터를 동기화시킬 수 있다. 다른 예로서, 중단점들이 각종의 다른 중단점들과 통신을 할 수 있는 경우, 하나의 고장점(point of failure)(중앙집중형 서버 중단점이 필요한 경우 존재할지도 모름)이 제거될 수 있다. 다른 예로서, 어떤 구현들에서, 적어도 피어-투-피어 동기화 시스템이 설정 또는 구성하기 더 쉬울 수 있으며, 예를 들어, 이러한 설정에서 다른 중단점들로부터의 서비스 요청들에 항상 응할 수 있어야 하는 것 또는 부가의 용량을 제공하기 위해 확장될 수 있는 것 등의 특성의 요구사항들을 갖는 하나 이상의 서버를 구성할 필요가 없는 경우에 그렇다.

[0003]

그렇지만, 분산형 시스템도 역시 어떤 경우들에 하나 이상의 중앙집중형 중단점들을 사용하는 토폴로지에서도 고정되거나 대비할 수 있는 적어도 몇몇 단점들을 비롯한 단점들을 가질 수 있다. 예를 들어, 피어-투-피어 중단점들이 ("항상 이용가능"하게 구성되어 있을지 모르는 적어도 일부의 "서버" 중단점들과 달리) 항상 이용가능하거나 데이터를 동기화시킬 수 있어야 하는 것은 아니며, 이용가능한 중단점이 없다는 것은 때때로, 특성의 중단점이 이용가능할 때까지 또는 그렇게 되지 않는 한, 데이터가 동기화될 수 없다는 것을 의미할 수 있다. 다른 잠재적인 단점은, 어쩌면 임의의 토폴로지로 연결되어 있는 일련의 중단점들 간에 데이터, 특히 대량의 데이터를 효율적으로 동기화시키는 방법을 결정하는 것에 관한 것일 수 있다. 적어도 몇몇 경우들에, 어느 중단점이 어느 다른 중단점과 동기화해야 하는지, 어떤 데이터가 동기화되어야 하는지, 기타 등등은 사소한 문제가 아니며, 해결 방안들이 찾아내는 데 많은 계산을 필요로 하거나, 차선책일 수 있거나(차레로 필요한 것보다 더 많은 데이터를 전달하는 등의 문제를 야기할 수 있음), 기타 등등일 수 있다. 분산형 시스템의 다른 잠재적인 단점들은, 어떤 중단점도 또는 일부 중단점들이 꼭 동기화된 데이터의 전부 또는 적어도 상당 부분을 동기화시켜야만 하거나 액세스해야만 하는 것은 아닌 경우, 중단점들의 시스템에 관한 정보 - 중단점들의 특성, 특성의 또는 모든 중단점들에 의해 전달되는 데이터의 전체적인 관점, 기타 등등 - 가 수집하기 더 어려울 수 있다는 것이다. 또 다른 잠재적인 단점은 임의의 특성의 피어 기계를 완전히 신뢰할 수는 없더라도 보안 정책 또는 기타 보안-관련 기능을 적용하는 것에 관한 것일 수 있으며, 이러한 보안 문제들 중 적어도 몇몇은 어떤 구현들에서 하나 이상의 중앙집중형, 어쩌면 신뢰된 중단점들의 존재에 의해 개선될 수 있다.

### 발명의 상세한 설명

[0004]

이하는 읽는 사람에게 기본적인 이해를 제공하기 위해 본 발명의 간략화된 요약을 제공한다. 이 요약은 본 발명의 전반적인 개요가 아니며, 본 발명의 주요 또는 중요 구성요소들을 확인하거나 본 발명의 범위를 결정하지 않는다. 이 요약의 유일한 목적은 나중에 제공되는 보다 상세한 설명에 대한 서문으로서 본 명세서에 개시된 몇몇 개념들을 간략화된 형태로 제공하는 데 있다.

[0005]

중앙집중형 및 분산형 동기화 시스템들 및 통신 토폴로지들의 구성요소들을 사용하여 중단점들 간에 데이터를 동기화시키는 것에 관한 다양한 기법들 및 기술들이 본 명세서에 개시되어 있다. 적어도 몇몇 구현들에서, 동기화된 데이터의 어떤 일부가 중앙집중형 중단점으로 전달될 수 있는 반면, 동기화된 데이터의 다른 일부는 다른 중단점들에 직접 분산 방식 또는 피어-투-피어 방식으로 전달될 수 있다. 그에 부가하여, 어떤 구현들에서, 중단점들 간의 데이터의 동기화를 돕기 위해 각종의 협동적, 어쩌면 유익한 기능이 특성의 중단점들(중앙집중형 중단점들을 포함함)에 구현될 수 있다.

### 실시예

[0016]

중앙집중형 및 분산형 동기화 시스템들 및 통신 토폴로지들의 구성요소들을 사용하여 중단점들 간에 데이터를 동기화시키는 것에 관한 다양한 기법들 및 기술들이 본 명세서에 개시되어 있다. 보다 상세하게는, 적어도 몇몇 구현들에서, 동기화된 데이터의 어떤 일부가 중앙집중형 중단점으로 전달될 수 있는 반면, 동기화된 데이터의 다른 일부는 다른 중단점들에 직접 분산 방식 또는 피어-투-피어 방식으로 전달될 수 있다. 그에 부가하여, 어떤 구현들에서, 중단점들 간의 데이터의 동기화를 돕기 위해 각종의 협동적, 어쩌면 유익한 기능이 특성의 중단점들(중앙집중형 중단점들을 포함함)에 구현될 수 있다.

[0017]

이제부터 도 1을 참조하면, 데이터가 중앙집중형 토폴로지 및 피어-투-피어 토폴로지 둘다에서 동기화되어 공유될 수 있는 예시적인 시스템(100)이 도시되어 있다. 예시적인 시스템(100)은 서버 중단점(110), 장치 중단점 A(160), 장치 중단점 B(170), 및 장치 중단점 C(180)을 포함하고 있다. 예시적인 서버 중단점(110)은 저장 서비스(120), 블로그 탐색 서비스(125), 블로그 캐쉬 서비스(130), 클라우드 스토리지 서비스(135), 통지 서비스(140), 지식 저장소 모듈(145), 및 블로그 다운로드 모듈(150)을 포함하는 것으로 도시되어 있다. 예시적인 장치 중단점 A(160)은 블로그 다운로드 모듈(162)을 포함하는 것으로 도시되어 있다. 도 1이 다른 도면들을 참조

하여 설명될 수 있다. 그렇지만, 도 1을 참조하여 기술되는 구성요소들이 다른 도면들을 참조하여 기술되는 구성요소들과 함께 사용되는 것으로 제한되는 것으로 의도하기 위한 것이 아님을 잘 알 것이다. 그에 부가하여, 도 1의 예시적인 다이어그램이 특정의 구성요소들을 나타내고 있지만, 어떤 구현들에서는 이들 구성요소 전부가 존재하는 것은 아닐 수 있으며, 어떤 구현들에서는 부가의 구성요소들이 존재할 수 있다.

[0018] 일반적으로 또 적어도 몇몇 구현들에서, 임의의 2개 또는 그 이상의 중단점들 간에 데이터가 동기화될 수 있다. 예를 들어, 도 1에 나타낸 바와 같이, 장치 중단점 A(160)와 서버 중단점(110) 간에 데이터가 동기화될 수 있지만, 그에 제한되지 않는다. 동일한 또는 다른 데이터가 그 다음에 서버 중단점과 장치 중단점 B(170) 간에 동기화될 수 있다. 어떤 경우들에서, 장치 중단점 B는 그 다음에 차례로 장치 중단점 C(180)와 데이터를 동기화시킬 수 있다. 어떤 구현들에서 특정의 중단점들이 특정의 다른 중단점들과만 동기화할 수 있는 반면, 다른 구현들에서 중단점들이 각종의 중단점들(어떤 구현들에서 모든 다른 중단점들을 포함함)과 동기화할 수 있다.

[0019] 각종의 동기화 기법들이 존재하고 또 사용될 수 있지만, 적어도 몇몇 구현들에서 적어도 어떤 데이터가 SSE(Simple Sharing Extensions) 등의 기법을 사용하여 중단점들 간에 동기화될 수 있다. SSE는 일반적으로 RSS("Really Simple Syndication" 또는 "Rich Site Summary") 또는 Atom 문서 등의 피드 문서(feed document)에 (비교적) 간단한 데이터를 추가하는 것을 사용하는, 중단점들 간에 데이터를 공유 및 동기화시키기 위한 데이터 포맷 및 프로세스를 포함하는 것으로 기술될 수 있다. 어떤 경우들에, SSE 정보를 포함하는 피드를 "SSE 피드"라고 할 수 있다.

[0020] 이러한 구현들에서, 또는 적어도 어떤 다른 구현들에서, 중단점들은 SSE 정보 및 동기화된 데이터의 적어도 일부분 둘다를 포함하는 피드 문서를 제공함으로써 동기화를 위해 다른 중단점들이 데이터를 이용할 수 있게 해줄 수 있다. 정보가 동기화되고 있는 중단점은 피드들로부터의 정보를 하나 이상의 데이터 저장소 내의 정보와 병합시키기 위해, 기법들 중에서도 특히, SSE 프로세스를 사용하여 피드 문서를 획득하고 그의 내용을 해석할 수 있다. 이 프로세스의 일부로서, 정보가 동기화되어 있는 중단점은 일반적으로 동기화된 데이터의 그 자신의 사본 또는 버전을 업데이트할 수 있으며, 그에 의해 제1 중단점과 제2 중단점 간의 데이터 동기화를 완료할 수 있다. 단지 한 구체적인 예로서, 장치 중단점 A(160)는, 장치 중단점 A에 의해 유지 또는 액세스되는 정보가 서버 중단점에 동기화될 수 있도록, SSE 정보를 포함하는 RSS 피드를 서버 중단점(110)이 이용할 수 있게 해줄 수 있다. 서버 중단점은 피드를 획득하고, 어쩌면 "SSE 병합(SSE merge)"을 사용하여, 피드의 내용을 서버 중단점에 의해 유지되는 데이터와 병합시킬 수 있다. 이러한 병합 후에, 장치 중단점 A에 의해 제공되는 데이터의 적어도 일부가 서버 중단점에 의해 유지되는 데이터에 포함되어 있을 수 있다. 게다가, 장치 중단점 A에서 행해지는 데이터에 대한 추가의 변경들도 역시 유사한 방식으로 서버 중단점에 동기화될 수 있고, 이 부가의 변경은 장치 중단점 A에서 행해질 수 있고, 장치 중단점 A는 그의 제공된 피드를 부가의 변경을 포함하도록 업데이트할 수 있으며, 그 다음에 서버 중단점은 업데이트된 피드를 처리하여 부가의 변경을 포함시킬 수 있다.

[0021] 이전의 설명은 데이터 및 변경이 한 방향으로, 즉 제1 중단점에서 제2 중단점으로 동기화될 수 있는 방법에 대해 기술하고 있다. SSE를 비롯한 각종의 동기화 기법들이 또한 변경들이 "다른 방향"으로, 즉 이 예에서 제2 중단점에서 제1 중단점으로 동기화될 수 있게 해준다. 예를 들어, 이전의 예에서와 같이 장치 중단점 A(160)와 서버 중단점(110)에서 SSE가 사용되는 경우, 서버 중단점은 또한 (처음에 적어도) 장치 중단점 A에 의해 제공된 동일 정보를 포함하고 있는 그 자신의 SSE 피드를 제공(make available)할 수 있다. 그 다음에, 피드에 포함된 정보의 일부가 서버 중단점에서 업데이트될 때, 서버 중단점은 차례로 변경 또는 업데이트된 데이터, 및 피드 내의 SSE 정보에 대한 대응하는 업데이트 둘다를 포함시키기 위해 그 자신의 피드를 업데이트할 수 있다. 장치 중단점 A를 포함한 서버 중단점으로부터의 이 정보를 동기화시키는 다른 중단점은 그 다음에 피드를 획득하여 변경들을 병합시킬 수 있다(장치 중단점 A로부터의 변경들을 포함시키기 위해 아마도 서버 중단점에 의해 실행되는 것과 동일한 또는 유사한 병합 처리를 수행함). 요약하면, 몇몇 동기화 기법들에서, 예를 들어, 어느 한 중단점이 데이터에 변경을 할 수 있도록, 각각이 그 데이터를 제공하고 또 상대방 중단점에 의해 제공된 데이터가 변할 때 각각이 그 자신의 로컬 데이터를 업데이트함으로써, 2개의 중단점이 동일한 데이터를 서로 동기화시킬 수 있다.

[0022] 게다가, SSE를 비롯한 적어도 몇몇 동기화 기법들에서 기술된 것과 같은 기법들을 사용하여, 다수의 중단점이 하나의 (또는 다수의) 중단점과 동기화할 수 있다. 예를 들어, 장치 중단점 B(170)는 장치 중단점 A(160)와 서버 중단점(110) 간에 동기화되어 있는 동일한 (또는 어떤 다른) 데이터를 동기화시킬 수 있다. SSE 피드가 사용될 때, 장치 중단점은, 서버 중단점에 의해 제공된 피드(어떤 경우들에, 장치 중단점 A에 의해 검색된 동일 피드)를 검색하고 피드가 나타내는 데이터 또는 변경을 포함시킴으로써, 동일한 데이터를 동기화시킬 수 있다. 서버 중단점이 장치 중단점 B에 의해 데이터에 행해진 변경들을 포함시킬 수 있도록, 장치 중단점 B도 역시 피



드를 서버 종단점에 제공할 수 있다.

- [0023] 어떤 경우들에 (예를 들어, 장치 종단점 A(160)에 의해 행해진 변경들이 서버 종단점(110)을 통해 장치 종단점 B(170)에 동기화되었을 수 있는 이전의 예시적인 설명에서와 같이) 다른 종단점을 통해 장치들 간에 데이터가 동기화될 수 있지만, 적어도 어떤 경우들에 또 적어도 특정의 동기화 기법들을 사용할 때 다른 동기화 토폴로지들도 역시 이용가능할 수 있다. 예를 들어, 어떤 경우들에, SSE를 비롯한 동기화 기법 또는 프로토콜은 중앙집중형 서버(아마도 서버 종단점(110) 등)를 통해 또 다른 종단점들과 직접 동기화할 수 있는 기능을 제공할 수 있다. 예를 들어, 어떤 구현들에서 장치 종단점 A는 서버 종단점(110)을 통해 동기화하는 것에 의해 또는 장치 종단점 B와 직접 동기화하는 것에 의해(도 1에서 장치 종단점 A와 장치 종단점 B 간의 점선으로 도시되어 있음) 장치 종단점 B에 대해 데이터를 동기화시킬 수 있다. 동일한 또는 다른 구현들에서, 장치 종단점 B는 장치 종단점 C(180)와 직접 데이터를 동기화시킬 수 있다. 예를 들어, 이러한 직접 동기화는 장치 종단점 A와 장치 종단점 B 간에 또는 장치 종단점 B와 장치 종단점 C 간에 SSE 피드 문서의 교환 및 교환된 피드에 포함된 데이터의 병합에 의해 구현될 수 있다. 게다가, 어떤 구현들에서 심지어 서버 종단점의 존재가 필요하지 않을 수도 있다. 예를 들어, 어떤 구현들에서 메타데이터 및 블로그 데이터가 어떤 경우들에, 하나 이상의 서버 종단점의 개입 또는 그에 의해 제공되는 도움 없이, 다른 장치 종단점들의 위치를 알고 있는 장치 종단점들 간에 직접 전달될 수 있다.
- [0024] 적어도 몇몇 구현들에서, 각종 유형의 데이터가 종단점들 간에 동기화될 수 있다. 예를 들어, 2개의 종단점이 각종의 정보를 포함하는 텍스트 정보를 동기화시킬 수 있거나, 오디오나 비디오 파일, 또는 컴파일된 실행가능 코드 등의 "바이너리(binary)" 정보를 동기화시킬 수 있거나, 임의의 다른 유형 또는 포맷의 데이터를 동기화시킬 수 있다.
- [0025] 동기화되는 데이터는 다양한 방식으로 표현될 수 있다. 어떤 구현들에서, 이러한 데이터가 앞서 기술한 단계들을 사용하여 동기화되는 정보를 비롯한, 종단점들 간에 동기화되는 정보에 포함되어 있을 수 있다. 예를 들어, SSE를 사용하는 구현들에서, 이러한 데이터는 종단점들 간에 교환되는 SSE 피드에 포함되어 있을 수 있다. 그렇지만, 동일한 또는 다른 구현들에서, 종단점들 간에 전달되는 정보가 종단점들 간에 궁극적으로 동기화 또는 전달될 수 있는 데이터 모두를 포함하는 것은 아닐 수도 있다. 이러한 경우들 중 몇몇에서, 종단점들 간에 전달되는 정보가 그 대신에 또는 그에 부가하여 동기화될 다른 데이터에 대한 참조를 포함할 수 있다. 이러한 참조는, 어쩌면 리소스를 참조하는 URL을 포함하는 RSS 또는 Atom 피드에서 "enclosure" 엘리먼트를 사용하는 것과 같은, 다양한 방식으로 제공될 수 있다. 참조에 의해 식별되는 데이터가 그 다음에 다양한 수단을 사용하여 액세스 또는 저장될 수 있다.
- [0026] 본 명세서에서 사용되는 바와 같이, 용어 "메타데이터"는 동기화 또는 전달되는 데이터를 식별해주는(어떤 구현들에서, 포함할 수 있는), 종단점들 간에 전달되는 정보를 말하는 것으로 해석될 수 있다. 예를 들어, 어떤 구현들에서 메타데이터는 각각의 동기화된 데이터의 "항목(item)"을 포함하는 문서 또는 피드에 의해 구현될 수 있다. 각각의 항목은, 차례로 또 어떤 예시적인 구현들에서, "동기 데이터(sync data)" 부분 및 "항목 데이터(item data)" 부분을 포함할 수 있다. 이러한 경우들에, 동기 데이터는 특정 항목의 동기화에 관한 정보와 연관될 수 있는 반면, 항목 데이터는 동기화된 정보와 연관될 수 있다. 예를 들어, 동기 데이터는 항목과 연관된 버전 번호 및 항목이 업데이트된 방법 또는 시기의 이력 등의 정보를 포함할 수 있는 반면, 항목 데이터는 항목 자체 또는 적어도 항목과 연관된 정보를 포함할 수 있다.
- [0027] 어떤 경우들에 항목 데이터 전부가 메타데이터에 포함될 수 있는 반면, 다른 경우들에 메타데이터가 항목 데이터 전부를 포함하지는 않을 수 있고 그 대신에 항목 데이터에 대한 하나 이상의 참조를 포함할 수 있다. 참조를 사용하면 메타데이터를 (비교적) 작게 유지할 수 있으며, 대량의 항목 데이터가 있는 경우들에 특히 그렇다. 예를 들어, 메타데이터 자체의 항목 데이터 부분에 큰 오디오 또는 비디오 파일 전부를 포함시키는 것보다는, 그 대신에 항목 데이터 요소가 큰 오디오 또는 비디오 파일을 참조하거나, 식별해주거나, 또는 그의 검색에 도움이 되는 어떤 유형의 참조 또는 식별자를 포함할 수 있다. 메타데이터에 의해 참조될 수 있는 것과 같은 이러한 정보 또는 데이터는 메타데이터와 별도로 저장 또는 액세스될 수 있고, 어떤 경우들에 데이터의 "블로브(blob)" 또는 간단히 "블로브" 또는 "인클로저(enclosure)"라고 할 수 있다. (어떤 경우들에, 동기 데이터의 적어도 일부분이 메타데이터에 포함되지 않을 수 있고 그 대신에 메타데이터로부터 참조될 수 있다.)
- [0028] SSE를 사용하는 구현들에서, 메타데이터는 어떤 경우들에 SSE 피드에 의해 구현될 수 있다. 예를 들어, SSE 정보를 포함하는 RSS 피드는 각각의 데이터에 대한 "item" 엘리먼트 등의 통상의 RSS XML 엘리먼트들을 포함하는 XML 문서로서 구현될 수 있다. 각각의 "item" 엘리먼트는 차례로 동기 데이터(적어도 어떤 구현들에서, "syn



c"라고 하는 XML 엘리먼트에 의해 구현될 수 있음) 및 항목 데이터(적어도 어떤 구현들에서, "item"이라고 하는 XML 엘리먼트에 의해 구현될 수 있으며 항목에 관한 정보 또는 항목 자체를 포함할 수 있음)를 포함할 수 있다.

[0029] 일반적으로, 항목 데이터 또는 항목 데이터 요소는 임의의 유형의 데이터를 포함할 수 있다. 간단한 경우에, RSS 문서에서 항목 데이터는 "title" 및 "description"과 같은 표준 RSS 엘리먼트를 포함할 수 있다. 동일한 또는 다른 경우들에서, 항목 데이터는, 예를 들어, 어쩌면 hCard와 같은 포맷을 사용하여 인코딩된 연락처 정보 또는 기타 데이터를 포함할 수 있다. 또다른 또는 동일한 구현들에서, 항목 데이터는 메타데이터 또는 RSS 피드에 포함되어 있지 않는 정보를 비롯한, 다른 곳에 위치한 정보에 대한 하나 이상의 참조를 포함할 수 있다. 메타데이터가 RSS를 사용할 때, 이러한 참조는 RSS "enclosure" 엘리먼트와 같은 엘리먼트를 사용하여 구현될 수 있다.

[0030] 유의할 점은 적어도 몇몇 경우들에서 메타데이터가 동기화되는 정보 모두를 포함할 수 있다는 것이다. 즉, 중단점들 간에 동기화된 메타데이터가 항상 다른 정보를 참조하거나 식별해주는 것은 아닐 수 있다. 예를 들어, 메타데이터가, 말하자면, 연락처와 연관된 정보 모두를 포함하는 경우, 연락처 정보를 포함하는 동기화된 데이터(피드 또는 문서를 포함함)를 여전히 일반적으로 메타데이터라고 할 수 있다.

[0031] 많은 동기화 기법들이 전달될 데이터에 관한 정보(즉, 메타데이터) 및 정보 자체(메타데이터에 의해 참조되는 데이터의 블로그를 포함함) 둘다를 동시에 함께 또는 동일한 중단점을 사용하여 전달하지만, 본 명세서에 기술된 기법들 중 적어도 몇몇에 의해 행해지는 바와 같이 메타데이터 및 블로그를 개별적으로 처리할 때 몇몇 이점들이 있을 수 있다. 적어도 몇몇 구현들에서, 예를 들어, 메타데이터의 양이 참조되는 데이터 또는 블로그 데이터의 양만큼 많지 않을 수 있고, 따라서 메타데이터를 특정의 중단점들 간에 한 방식으로 전달하는 반면 블로그 데이터를 아마도 다른 일련의 중단점들 간에 다른 방식으로 전달하는 것이 유익할 수 있다. 한 이러한 구성에서, 메타데이터는 서버 중단점(110)으로 또 서버 중단점(110)을 사용하여 전달될 수 있는 반면, 블로그 데이터는 적어도 몇몇 경우들에 장치 중단점들 간에 직접 전달될 수 있다. 게다가, 메타데이터의 일부 또는 그 전부가 데이터를 동기화시키는 모든 중단점들 간에 전달될 수 있는 반면, 블로그 데이터가 모든 중단점들 간에 꼭 전달되는 것은 아닐 수 있다. 예를 들어, 블로그가 예를 들어 특별히 요청되거나 요구되는 경우에, 그 블로그가 중단점들 간에만 전달될 수 있다.

[0032] 게다가, 어떤 구현들에서, 각종의 특성들에 따라 블로그 데이터의 하나 이상의 대안의 표현들이 생성되어 중단점들 간에 전달될 수 있다. 예를 들어, 정보를 비교적 느리게 전달하는 통신 메카니즘을 사용하는 하나 이상의 다른 중단점들에만 연결되어 있는 중단점은 어떤 경우들에 적어도 어떤 다른 중단점들에 의해 검색될 수 있는 블로그보다 작은 블로그를 원하거나 검색할 수 있다. 한 구체적인 예에서, 비교적 저대역폭 데이터 네트워크를 사용하여 연결되는 이동 전화는 이미지 또는 비디오가 (어쩌면 이미지의 크기를 감소시킴으로써, 보다 공격적인 압축 설정을 사용함으로써, 기타 등등에 의해) 보다 작은 양의 공간에 표현되도록 트랜스코딩(transcode)되거나 변환된 이미지 또는 비디오 파일을 획득할 수 있다. 적어도 몇몇 구현들에서 다른 종류의 대안의 표현들도 역시 가능하거나 지원될 수 있다. 예를 들어, 어떤 중단점들은 전체 오디오 또는 비디오 파일, 등 보다는, 말하자면, 오디오 또는 비디오 파일의 클립을 검색할 수 있다.

[0033] 이러한 구현들 중 적어도 몇몇 구현들에서, 하나 이상의 특정의 중단점들이 특정의 대안의 표현들을 발생 또는 제공할 수 있는 기능을 가지고 있을 수 있다. 예를 들어, 어떤 구현들에서 "대안의 표현 서비스(alternate representation service)" 또는 "트랜스코딩 서비스(transcoding service)"는 서버 중단점(110)에 존재할 수 있고(도 1에 도시되어 있지 않음), 서버 중단점에 또는 다른 중단점들에 저장된 블로그들의 적어도 어떤 대안의 표현들의 발생에 참여할 수 있다. 동일한 또는 다른 구현들에서, 하나 이상의 장치 중단점들과 같은 적어도 어떤 특정의 다른 중단점들은 그에 부가하여 또는 그 대신에 블로그들의 대안의 표현들을 제공할 수 있는 기능을 가지고 있을 수 있다. 대안의 표현들의 발생 및 제공과 관련한 부가적인 모듈들, 서비스들, 및 프로세스들도 역시 본 명세서의 다른 곳에 더 기술되어 있는 것과 같다.

[0034] 서버 중단점의 예시적인 구현은 예시적인 서버 중단점(110)의 일부인 것으로 도 1에 도시되어 있는 서비스들 및 모듈들 중 하나 이상을 비롯한 다양한 기능을 포함하고 있을 수 있다. 앞서 설명한 바와 같이, 서버 중단점들 모두가 도시된 모듈들 및 서비스들 전부를 포함하는 것은 아닐 수 있는 반면, 다른 서버 중단점들이 부가의 모듈들 및 서비스들을 포함할 수 있다. 게다가, 모듈들 및 서비스들 중 몇몇이 특정의 방식으로 서로 간에 통신을 하는 것으로 기술될 수 있지만, 일반적으로 많은 모듈들 및 서비스들이, 서버 중단점들은 물론 기타 중단점들에 위치한 모듈들 및 서비스들과 이러한 통신에 대해 구체적으로 기술되어 있지 않은 모듈들 및 서비스들을 비롯한, 다른 모듈들 또는 서비스들과 통신을 할 수 있다. 예를 들어, 지식 저장소 모듈(145)은, 말하자면, 하

나 이상의 특정의 방식으로 블로브 캐쉬 서비스(130)에서 사용되는 것으로 기술될 수 있지만, 이러한 설명은 지식 저장소 모듈을 블로브 캐쉬 서비스와 또는 통신이 구체적으로 기술되어 있는 기타 모듈들 또는 서비스들과 통신하는 것만으로 제한하지 않으며, 지식 저장소 모듈은 적어도 몇몇 구현들에서도 역시 각종의 다른 모듈들 또는 서비스들과 통신할 수 있다. 게다가, 도 1에 도시되어 있지는 않지만, 어떤 구현들에서 2개 이상의 서버 중단점들이 존재할 수 있다. 이러한 구현들 중 적어도 몇몇 구현들에서, 적어도 어떤 서버 중단점들은 다른 서버 중단점들과 통신을 할 수 있고 또 다른 중단점들 간의 데이터 동기화에 참여할 수 있다. 이러한 다수의 서버 중단점들 각각은 예시적인 서버 중단점(110)의 일부인 것으로 도 1에 도시되어 있는 예시적인 서비스들 및 모듈들 중 일부만 또는 어쩌면 그 전부를 포함할 수 있다.

[0035] 예시적인 저장 서비스(120)는 일반적으로 메타데이터를 저장하고 다른 중단점들이 서버 중단점(110)과 메타데이터를 동기화시킬 수 있게 해주는 기능을 제공할 수 있다. 예시적인 장치 중단점 A(160)는, 예를 들어, 로컬 변경을 할 수 있고 메타데이터가 그 변경을 포함하도록 그 장치 중단점이 유지하고 있는 메타데이터를 업데이트하며 업데이트된 메타데이터를 저장 서비스로 전달할 수 있다. 저장 서비스는 그 다음에 장치 중단점 A로부터 전달된 메타데이터의 변경을 저장 서비스에 의해 관리되는 동일 메타데이터의 어떤 표현과 병합시킬 수 있다. 저장 서비스는 그 다음에 그 자신의 병합된 또는 업데이트된 메타데이터를 다른 중단점들에 제공할 수 있다(이에 따라 차례로 다른 중단점들이 저장 서비스에 의해 제공된 메타데이터를 검색할 수 있고 그 다음에 원래 장치 중단점 A에 의해 행해진 변경을 포함시킬 수 있게 된다). 저장 서비스는 적어도 몇몇 구현들에서 다수의 SSE 피드와 같은 다수의 개별적인 메타데이터를 저장 및 제공할 수 있고, 각각의 메타데이터는 서로 다른 (또는 동일한) 데이터 세트를 식별해줄 수 있다. 예시적인 저장 서비스(120)에 의해 수행될 수 있는 동작들 중 적어도 몇몇 동작들이, 특히 도 5를 참조하여, 이하에서 더 상세히 설명될 것이다.

[0036] SSE를 사용하는 구현들을 비롯한 몇몇 구현들에서, 저장 서비스는 다른 중단점들에 의해 제공된 SSE 피드를 받아들이기(accept)(또는 검색 또는 어떤 방식으로 획득)할 수 있다. 저장 서비스는 이어서 획득된 피드들에서의 변경들을, 저장 서비스가 다양한 방식으로 - 파일 자체로, 데이터베이스에서의 레코드로, 기타 등등을 포함함 - 저장 또는 액세스할 수 있는 피드의 로컬 표현과 병합시킬 수 있다. 저장 서비스는 이어서 병합 동작의 결과인 업데이트된 피드를 다양한 방식으로 - 다른 중단점들에 의해 액세스가능한 파일로서(어쩌면 웹 또는 파일 서버를 사용함), 사전 예방적으로 업데이트된 피드를 다른 중단점들에 전달함으로써, 기타 등등을 포함함 - 적어도 몇몇 중단점들에 제공할 수 있다.

[0037] 몇몇 구현들에서, 예시적인 블로브 탐색 서비스(125)는 블로브를 식별해주는 어떤 데이터(블로브 식별자 등)를 수신하고 또 중단점이 그 다음에 블로브를 획득 또는 검색하기 위해 사용할지도 모르는 하나 이상의 "위치 지정자(locator)"를 제공할 수 있다. 예를 들어, 메타데이터 및 블로브가 서로 다르게 또는 개별적으로 전달 또는 동기화될지도 모르는 구현들에서, 하나 이상의 중단점들이 특정의 블로브가 검색될 수 있는 위치 또는 위치들을 포함한 정보를 블로브 탐색 서비스에 제공할 수 있다. 몇몇 구현들에서, 이러한 위치들은 장치 중단점들, 블로브 캐쉬 서비스(130), 또는 클라우드 스토리지 서비스(135)를 포함할 수 있다. 블로브 탐색 서비스는 이어서 나중에 언젠가 이들 위치 중 하나 이상을 블로브를 획득하고자 하는 다른 중단점에게 제공할지도 모른다. 예시적인 블로브 탐색 서비스에 의해 수행될 수 있는 적어도 몇몇 동작들이, 특히 도 4 및 도 6을 참조하여, 이하에서 더 상세히 설명될 것이다.

[0038] 블로브 식별자는 일반적으로 블로브를 식별해주는 어떤 데이터일 수 있다. 몇몇 구현들에서, 특히 블로브 자체가 전달될 필요없이 블로브를 식별하기 위해, 블로브 식별자가 각종의 장치 및 서버 기능에 의해 사용될 수 있다. 따라서, 예를 들어, 저장 서비스 및 기타 중단점들은 메타데이터 내의 블로브 식별자를 사용할 수 있고, 블로브 탐색 서비스는 블로브 식별자를 사용하여 위치들을 저장 및 제공할 수 있으며, 블로브 캐쉬 서비스 또는 클라우드 스토리지 서비스는 인덱싱된 블로브를 저장할 수 있거나 연관된 블로브 식별자를 사용하여 검색될 수 있고, 기타 등등이 행해질 수 있다. 어떤 구현들에서 블로브를 식별하기 위해 하나의 블로브 식별자가 각종의 모듈들 또는 기능에 의해 사용될 수 있는 반면, 동일한 또는 다른 구현들에서 특정의 단일 블로브를 식별하기 위해 다수의, 아마도 서로 다른 블로브 식별자들이 - 아마도 서로 다른 모듈들 또는 기능에 의해 - 사용될 수 있다.

[0039] 블로브 식별자가 다양한 방식들로 구현될 수 있고 또 다양한 형태를 취할 수 있다. 적어도 하나의 구현에서, 블로브 식별자는 "ABC123", "ABCDEFGH", "123456", 기타 등등과 같은 영숫자 또는 기타 문자들의 문자열(string) 또는 집합일 수 있다. 다른 구현에서, 블로브 식별자는, 예를 들어, 아마도 "http://www.els.live.com/ABC123", "http://www.els.live.com/ABCDEFGH", "http://www.els.live.com/123456", 기타 등등의 URL(uniform resource locator) 또는 URI(uniform resource

identifier)의 형태를 취할 수 있다. 이러한 예에서, 식별자의 "http://www.els.live.com/" 부분 다음에 오는 문자열이 실제로 블로브를 식별해줄 수 있다. 어떤 경우들에, 식별자의 "http://www.els.live.com/" 부분이 각 종의 다른 목적으로 - 아마도 블로브 식별자를 사용자에게 익숙할지도 모르는 식별자와 보다 비슷하게 보이도록 하기 위해 또는 심지어 적어도 어떤 특정의 경우들에(특히 도 9를 참조하여 이하에 기술되는 예시적인 구현을 포함함) 블로브를 실제로 찾아내기 위해 - 사용될 수 있다.

[0040] 예시적인 블로브 캐쉬 서비스(130)는 임시 블로브 또는 파일의 저장 장소를 제공할 수 있다. 이러한 장소는 어떤 경우들에 종단점들 간의 블로브 전달에 도움이 될 수 있다. 예를 들어, 장치 종단점 A(160)가 서버 종단점 (110)과 메타데이터를 동기화시키고 이러한 메타데이터가 장치 종단점 A로부터 이용가능한 블로브를 식별해주는 것으로 가정한다. 또한 블로브가 서버 종단점으로 전달되지 않고 그 대신에 장치 종단점 A에 남아 있는 것(검색가능함)으로 가정한다. 그 다음에 장치 종단점 A가 오프라인으로 되는 것으로, 즉 서버 종단점 및 장치 종단점 B(170) 중 하나 또는 둘다에 대해 이용가능하지 않게 되는 것으로 가정한다. 이제, 장치 종단점 B가 서버 종단점으로부터의 메타데이터를 동기화시키고 또 서버 종단점이 장치 종단점 A로부터 블로브를 검색하기를 원하는 것으로 판정하는 경우, 서버 종단점이 그렇게 할 수 없을지도 모르는데, 그 이유는 장치 종단점 A(블로브의 유일한 사본을 가지고 있을 수 있음)가 온라인이 아니거나 이용가능하지 않기 때문이다. 이러한 문제가 블로브 캐쉬 서비스의 사용을 비롯한 각종 수단들을 통해 해결될 수 있다. 한 이러한 구현에서, 장치 종단점 A가 이용가능하지 않게 되기 전에, 장치 종단점 A에 의해 제공된 메타데이터와 연관된 블로브가 블로브 캐쉬 서비스에 저장될 수 있고, 그 다음에 장치 종단점 B가 장치 종단점 A로부터가 아니라 블로브 캐쉬 서비스로부터 블로브를 검색할 수 있다.

[0041] 캐쉬처럼, 예시적인 블로브 캐쉬 서비스는 블로브를 임시 데이터로서 저장할 수 있다. 즉, 어쩌면 이하에 기술되는 클라우드 스토리지 서비스(135)와 달리, 블로브 캐쉬 서비스에 의해 저장된 블로브들은 제한된 기간 동안만 저장될 수 있고, 어떤 경우들에 블로브 캐쉬 서비스 자체에 의해 또는 어떤 다른 모듈 또는 서비스에 의해 제어되는 다양한 때에 제거될 수 있다. 예를 들어, 어떤 구현들에서, 블로브 캐쉬 서비스는 언제 블로브를 삭제할지 또는 현재 저장된 블로브를 새로운 블로브로 대체시킬지를 결정하기 위해 하나 이상의 캐싱 알고리즘을 사용할 수 있다. 예를 들어, 적어도 한 구현에서, 블로브 캐쉬 서비스는 먼저 그의 이용가능한 저장 공간을 채우거나 점유시킬 수 있고, 그 다음에 저장해야할 새로운 블로브가 있을 때, 최근에 사용되지 않은 하나 이상의 블로브를 찾아내어 삭제함으로써 새로운 블로브 데이터에 이용가능한 공간을 만들기 위해 "LRU(least recently used)" 알고리즘 등의 알고리즘을 사용할 수 있다. 어떤 경우들에, 적어도 블로브 캐쉬 서비스와 연관된 기능을 구현하는 것이, 어쩌면 이하에 기술되는 클라우드 스토리지 서비스(135)를 비롯한 영속적 저장소와 연관된 기능을 구현하는 것보다 더 적은 비용, 동작 또는 기타 자원을 필요로 할지도 모른다. 예를 들어, 블로브 캐쉬 서비스와 연관된 정보가 신뢰성있게 또는 무기한으로 저장된다고 보장될 수 없기 때문에, 블로브 캐쉬 서비스는 적어도 어떤 다른 비밀시적 데이터를 보유하도록 설계된 저장소에서 요구될 수 있는 것과 동일한 수준 또는 유형의 중복 데이터 저장 기능을 필요로 하지 않을 수 있다.

[0042] 예시적인 블로브 캐쉬 서비스(130)에 의해 수행될 수 있는 적어도 몇몇 동작들이, 특히 도 7을 참조하여, 이하에서 더 상세히 설명될 것이다.

[0043] 예시적인 클라우드 스토리지 서비스(135)는 일반적으로 비밀시적인 데이터에 대한 "클라우드"에의 저장을 제공할 수 있다. 즉, 여러가지 점에서 - 즉, 예를 들어, 블로브를 저장할 수 있다는 점에서 - 블로브 캐쉬 서비스처럼 동작할 수 있지만, 클라우드 스토리지 서비스가 사용자들 및 종단점들에 의해, 예를 들어, "항상 켜있는" 또는 "항상 액세스가능한" 그리고 "신뢰성있는" 저장을 제공하는 것으로 보여질 수 있다. 종단점들은 블로브 데이터를, 말하자면, 장치 종단점에 저장하는 것 이외의 어떤 경우들을 비롯한 다양한 경우들에 클라우드 스토리지 서비스를 사용하여 블로브 데이터를 저장할 수 있다. 예시적인 클라우드 스토리지 서비스에 의해 수행될 수 있는 적어도 몇몇 동작들이, 특히 도 8을 참조하여, 이하에서 더 상세히 설명될 것이다.

[0044] 몇몇 구현들에서, 블로브 데이터에 대한 저장을 제공하고 블로브 캐쉬 서비스(130) 또는 클라우드 스토리지 서비스(135)와 다르게 동작하는 하나 이상의 부가적인 서비스들 또는 모듈들이 예시적인 서버 종단점(110)의 일부일 수 있다. 예를 들어, 한 구현에서, 이러한 다른 블로브 데이터 저장 서비스가 블로브 캐쉬 서비스에 의해 제공되는 것보다는 비교적 더 영속적이지만 클라우드 스토리지 서비스에 의해 제공되는 것보다는 비교적 덜 영속적인 저장을 제공할 수 있다(적어도 이들 예시적인 서비스가 이상에서 소개되었음). 이러한 서비스는, 예를 들어, 블로브가 저장되어 있을지도 모르는 또는 블로브가 있을지도 모르는 다른 종단점들에 대한 지식을 사용하여 언제 블로브를 저장할지를 결정함으로써 그렇게 할 수 있다. 예를 들어, 어떤 지식이 블로브가 이용가능성이 비교적 높은 것으로 알려져 있는 몇개의 다른 종단점들에 이미 저장되어 있음을 나타내는 경우, 이러한 블로



브 데이터 저장 서비스는 블로그를 저장하지 않을 수 있다. 한 특정의 예에서, 그 다음에 블로그 데이터 저장 서비스에 의해 액세스가능한 지식이 특정의 블로그가, 예를 들어, 통상적으로 켜져 있고 대부분 또는 항상 네트워크에 연결되어 있는 4개의 다른 종단점들에 저장되어 있음을 나타내는 경우, 요청측 종단점이 아마도 4개의 다른 종단점들 중 적어도 하나로부터 블로그를 검색할 수 있을 것이기 때문에 예시적인 블로그 데이터 저장 서비스는 블로그를 저장하지 못할지도 모른다. 또한 유의할 점은, 적어도 몇몇 구현들에서, 이러한 다른 메카니즘들이 꼭 어떤 새로운 유형의 블로그 데이터 저장 서비스로 구현되어야만 하는 것은 아니며, 그 대신에 이러한 메카니즘들이 블로그 캐쉬 서비스 또는 클라우드 스토리지 서비스의 일부이거나 그 안에 포함되어 있을 수 있다는 것이다.

[0045] 특정의 메타데이터를 동기화시키는 종단점은, 예를 들어, 그 종단점이 그 자신의 메타데이터 사본을 업데이트시킬 수 있는 것은 물론 어쩌면 임의의 원하는 블로그 데이터를 검색할 수 있도록, 일반적으로 이러한 메타데이터가 하나 이상의 다른 종단점들에서 언제 업데이트되었는지를 알고 싶어할지도 모른다. 때때로 비효율적인 한 구현에서, 종단점은 하나 이상의 다른 이용가능한 메타데이터 - 어쩌면 서버 종단점(110)에 의해 제공된 것들을 포함함 - 를 "폴링", 즉 주기적으로 검사할 수 있고, 또 다른 메타데이터가 변경되었을 때 동기화시킬 변경들이 있는지를 판정할 수 있다. 다른 대안으로서, 종단점은 예시적인 통지 서비스(140)에 의해 제공되는 기능을 사용할 수 있다. 통지 서비스는 종단점들이 특정의 메타데이터에 대한 관심을 등록하거나 표명할 수 있는 기능을 제공할 수 있고, 그 다음에 특정의 메타데이터가 변경될 때 관심있는 또는 등록된 종단점에 통지를 제공할 수 있다. 그 결과, 종단점은 변경이 있는지 폴링하는 것을 회피할 수 있거나, 적어도 보다 적은 횟수로 폴링을 할 수 있고, 메타데이터가 변경될 때 여전히 어떤 조치를 취할 수 있다. 통지는 각종의 네트워킹 또는 기타 통신 메카니즘들에 의해 제공되는 포인트-투-포인트 또는 브로드캐스트 기능을 사용하는 것을 비롯하여 다양한 방식으로 구현될 수 있다. 동일한 또는 다른 구현들에서, 예시적인 통지 모듈은 또한 메타데이터에 대한 변경 이외의 다른 이벤트들에 관한 통지를 제공할 수 있다. 예를 들어, 어떤 구현들에서 통지 모듈은 업데이트된 또는 변경된 위치 지정자들(이러한 위치 지정자들은 예시적인 블로그 탐색 서비스에 의해 관리될 수 있음)에 관한 정보를 포함하는 통지를 종단점들에게 주기적으로 제공할 수 있다. 블로그를 검색할 때 업데이트된 위치 지정자들을 사용함으로써 종단점은 블로그를 검색하는 방법을 변경할 수 있고 따라서 어쩌면 블로그를 보다 효율적으로 검색할 수 있으며, 네트워크 또는 다른 조건들의 변경, 기타 등등에 적응할 수 있다.

[0046] 예시적인 지식 저장소 모듈(145)은 서버 종단점(110)에 의해 제공되는 상호작용 또는 기능을 통해 획득될 수 있는 "지식"에 대한 저장 및 액세스를 제공할 수 있다. 이러한 지식은 그 다음에 어떤 경우들에 데이터의 동기화를 제어 또는 최적화하는 것을 비롯한 다양한 목적을 위해 사용될 수 있다. 지식이 수집되어 사용될 수 있는 방법의 더 많은 예들이 본 명세서의 다른 곳에서 기술되어 있을 수 있지만, 어떤 경우들에 지식은 일반적으로 서버 종단점과 통신을 하는 또는 다른 종단점들과 통신을 하는 종단점들과 연관된 특성들 등의 데이터와 관련되어 있을 수 있다. 지식 저장소 모듈에 의해 저장되거나 사용되는 특성들을 일반적으로 "지식 특성(knowledge characteristic)"이라고 할 수 있다. 예를 들어, 어떤 예시적인 지식 특성들은 장치 종단점이 통상 서버 종단점에 연결되는 방식에 관한 정보(연결의 속도, 연결의 가용 대역폭, 연결과 연관된 비용, 기타 등등)를 포함할 수 있다. 예를 들어, 연결이 고속 데이터 네트워크 또는 비교적 느린 이동 전화 네트워크를 통할 수 있고, 연결이 하루 중 특정의 때에는 미사용 대역폭(free bandwidth)을 가질 수 있지만 하루 중 다른 때에는 사용 중일 수 있으며, 기타 등등일 수 있다. 지식 특성은 또한 종단점이 통상 온라인이거나 이용가능한 때에 관한 정보를 비롯한 종단점들에 관한 다른 정보도 포함할 수 있다 - 장치가 랩톱 또는 이동 전화일 수 있고 간헐적으로 연결될 수 있거나, "항상" 연결되어 있는 "항상 켜진" 데스크톱 컴퓨터일 수 있거나, 기타 등등일 수 있다 -. 다른 예로서, 지식 모듈은 종단점들과 서버 종단점 간의 상호작용들로부터 발생하는 정보(특정의 블로그가 요청되는 횟수, 블로그를 요청하는 종단점, 기타 등등)를 저장할 수 있다. 지식이 획득되는 방법 또는 장소에 상관없이, 지식이 서버 종단점에 의해 또는 어쩌면 다른 종단점들에 의해 다양한 방식으로(예를 들어, 데이터가 동기화되는 방법을 제어 또는 최적화하는 것의 일부로서) 제공 및 사용될 수 있다.

[0047] 또다른 예에서, 요청된 블로그의 하나 이상의 대안의 표현을 식별 또는 제공할지 여부를 판정하기 위해 지식(장치의 연결 특성 또는 기타 특성, 또는 기타 지식)이 사용될 수 있다. 예를 들어 앞서 소개한 바와 같이, 대안의 표현은, 저속 연결 메카니즘을 사용하여 연결된 종단점에 의해 이미지가 요청될 때 또는 비교적 보다 제한된 저장 공간을 갖는 종단점에 블로그가 저장될 때, 예를 들어, 그렇지 않았더라면 컷을지도 모르는 이미지의 보다 작은 트랜스코딩된 버전을 포함할 수 있다. 동일한 또는 다른 예에서, 지식은 특정의 종단점들이 특정의 블로그의 특정의 대안의 표현들을 가지고 있는지 및/또는 특정의 종단점들이 특정의 대안의 표현을 사용하여 표현된 블로그를 발생 또는 제공할 수 있는 기능을 가지고 있는지(아마도 특정의 종단점에서 대안의 표현을 발생하는

것, 어떤 다른 종단점으로부터 대안의 버전을 검색하는 것, 기타 등등)를 포함할 수 있다.

[0048] 마지막으로, 서버 종단점(110)의 일부인 블로브 다운로드 모듈(150) 또는 장치 종단점 A(160)의 일부인 블로브 다운로드 모듈(162) 등의 예시적인 블로브 다운로드 모듈은, 어떤 경우들에, 블로브를 다양한 다른 장소들(다른 종단점들을 포함함)로부터 다운로드 또는 검색할 수 있다. 일반적으로, 블로브 다운로드 모듈은 블로브의 어떤 일부분 또는 블로브 전체를 검색하기 위해 하나 이상의 블로브 검색 메카니즘들을 사용할 수 있으며, 이 경우 "블로브 검색 메카니즘"은, 예를 들어, 블로브를 찾아낼 수 있는 하나 이상의 방식들 및/또는 블로브를 검색할 수 있는 하나 이상의 방식들을 지정할 수 있다. 어쩌면 비교적 간단한, 한가지 블로브 다운로드 모듈은, 예를 들어, HTTP, FTP 또는 각종의 다른 파일 공유 프로토콜들 중 하나 등의 프로토콜을 사용하는 단일 블로브 검색 메카니즘을 사용하여 블로브를 검색하기만 할 수 있다. 다른 블로브 다운로드 모듈은 이러한 블로브 검색 메카니즘들은 물론 다른 블로브 검색 메카니즘들 중 임의의 메카니즘을 사용하여 파일을 검색할 수 있다. 예를 들어, 어떤 블로브 다운로드 모듈들은 "동시에" 각종의 종단점들로부터 동일한 블로브의 일부를 검색할 수 있는 메카니즘들(BitTorrent 프로토콜 등의 프로토콜에서 기술된 메카니즘 등)을 사용할 수 있다. 동일한 또는 다른 블로브 다운로드 모듈들은 또한 보다 효율적으로 블로브를 검색하기 위해 다른 기능(예를 들어, "차분 압축(differential compression)" 등)을 사용할 수 있으며, 차분 압축에서는 심지어 블로브의 작은 부분이 변경될 때마다 블로브 전체가 전달될 필요가 없도록 변경된 블로브의 일부만이 전달된다. 어떤 블로브 다운로드 모듈은 어떤 경우들에 블로브 탐색 서비스(어떤 구현들에서 앞서 소개한 예시적인 블로브 탐색 서비스(125)와 동일하거나 유사할 수 있음)를 사용하여 블로브를 찾아낼 수 있는 반면, 동일한 또는 다른 구현들에서 블로브 다운로드 모듈은, 블로브 탐색 기능을 사용하지 않거나 블로브 탐색 서비스를 사용하지 않고, 블로브를 찾아내거나 검색할 수 있다.

[0049] 게다가, 어떤 경우들에 블로브 다운로드 모듈은 블로브 다운로드 모듈에 의해 사용되는 블로브 검색 메카니즘(즉, 이와 관련해서는 "드라이버")을 구성하기 위해 "플러그블(pluggable)" 아키텍처를 사용할 수 있다. 예를 들어, 한 블로브 검색 메카니즘 또는 드라이버는 예시적인 블로브 캐쉬 서비스로부터 블로브를 검색하는 것을 가능하게 해줄 수 있고, 다른 드라이버는 블로브를 제공할 수 있는 종단점들의 세트를 검색하고 그 다음에 이러한 종단점들 중 하나 이상으로부터 블로브를 검색하기 위해 블로브 탐색 서비스를 사용할 수 있으며(2개 이상의 종단점을 갖는 어떤 경우들에 BitTorrent와 같은 메카니즘을 사용하는 것을 포함함), 또다른 드라이버는 블로브 탐색 서비스를 사용하지 않을 수 있고 각종의 다른 메카니즘들 중 하나 이상을 사용하여 블로브를 검색할 하나 이상의 종단점을 식별할 수 있고, 기타 등등이 있을 수 있다. 플러그블 아키텍처는 또한 블로브 다운로드 모듈 또는 블로브 다운로드 모듈을 포함하는 종단점의 전체적인 구성 또는 구현을 변경할 필요없이 부가의 블로브 검색 메카니즘들이 다양한 때에(블로브 다운로드 모듈이 설치되거나 이미 사용된 후를 포함함) 추가될 수 있게 해줄 수 있다. 이러한 기능은, 핵심 블로브 다운로드 모듈 또는 종단점 자체를 변경하지 않고, 나중에 새로운 블로브 검색 메카니즘을 추가하는 것을 가능하게 해줄 수 있다.

[0050] 유의할 점은 각각의 종단점이 이 예에서 데스크톱 컴퓨터, 서버 컴퓨터, 랩톱 컴퓨터, 워크스테이션 컴퓨터, 이동 전화 또는 셀룰러 전화, PDA(personal digital assistant), 기타 등등을 비롯한 임의의 개수의 범용 또는 전용 컴퓨터를 나타낼 수 있다는 것이다. 게다가, 특징의 예시적인 종단점이 "장치" 또는 "서버" 종단점인 것으로 기술될 수 있지만, 이러한 지정이 꼭 컴퓨팅 하드웨어의 성질 또는 종단점의 유형을 제한하는 것은 아니다. 서버 종단점 또는 장치 종단점은, 적어도 어떤 구현들에서, 데스크톱 컴퓨터, 서버 컴퓨터, 랩톱 컴퓨터, 워크스테이션 컴퓨터, 이동 전화 또는 셀룰러 전화, PDA, 기타 등등을 비롯한 임의의 유형의 컴퓨팅 하드웨어에 구현될 수 있다. 일반적으로, 종단점이 장치 종단점으로 간주되는지 서버 종단점으로 간주되는지는, 특성들 중에서도 특히, 예를 들어, 종단점이 구현되는 컴퓨팅 하드웨어의 성질에 의해서가 아니라 종단점에 의해 제공되는 기능에 의해 결정될 수 있다. 예를 들어, 종단점이, 예를 들어, 랩톱 컴퓨터 상에 구현되어 있더라도, 예시적인 서버 종단점(110)과 연관되어 있는 것으로 앞서 기술된 서비스들 및 모듈들 중 하나 이상을 사용하여 서버 기능을 제공하는 종단점이 서버 종단점인 것으로 간주될 수 있다. 또한 유의할 점은 종단점이 특징의 때에 서버 기능을 제공하는 반면(따라서 서버 종단점으로 간주될 수 있음) 동일한 또는 다른 때에 장치 종단점 또는 다른 유형의 종단점으로 동작할 수 있다는 것이다. 또한, 어떤 구현들에서, 특징의 또는 단일의 컴퓨팅 장치가 다수의 종단점들을 호스팅하거나 포함할 수 있다. 이들 또는 다른 구현들에서, 종단점들 간의 정보의 전달이, 적어도 어떤 경우들에, 특징의 단일 컴퓨팅 장치 상에서 실행되는 실행가능 코드 간의 통신만을 포함할 수 있다.

[0051] 종단점들 및 심지어 종단점 내의 모듈들 및 서비스들이 각종의 네트워킹 또는 다른 연결 수단을 사용하여 연결될 수 있다. 이러한 통신 수단은 임의의 유형의 네트워크(이더넷, Wi-Fi, 또는 이동 전화 또는 데이터 네트워

크 등)를 비롯한 데이터가 전송될 수 있는 임의의 수단과, 적어도 어떤 경우들에 CD(compact disc) 또는 플래쉬 메모리 드라이브 등의 물리 매체의 전달을 비롯한 임의의 다른 종류의 전달을 포함할 수 있다.

[0052] 이제 도 2를 참조하면, 동기화된 데이터를 변경하고 이러한 변경을 다른 종단점들이 이용할 수 있게 해줄 때, 장치 종단점을 비롯한 종단점에 의해 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름(200)이 도시되어 있다. 도 2에 대한 이하의 설명은 다른 도면들을 참조하여 행해질 수 있다. 그렇지만, 도 2을 참조하여 기술되는 동작 흐름이 이들 다른 도면들을 참조하여 기술되는 구성요소들과 함께 사용되는 것으로 제한되는 것으로 의도하기 위한 것이 아님을 잘 알 것이다. 그에 부가하여, 도 2의 예시적인 동작 흐름이 특정의 실행 순서를 나타내고 있지만, 하나 이상의 대안의 실시예들에서, 동작들이 다른 순서로 되어 있을 수 있다. 게다가, 예시적인 동작 흐름이 다수의 단계들을 포함하고 있지만, 어떤 구현들에서 이들 동작 중 적어도 몇몇이 결합, 즉 동시에 실행될 수 있고 또 동일한 또는 다른 구현들에서 몇몇 단계들이 실행되지 않을 수 있다는 것을 잘 알 것이다.

[0053] 동작(210)의 예시적인 구현에서, 어쩌면 도 1을 참조하여 앞서 소개한 장치 종단점 A(160) 등의 종단점에서 로컬 데이터에 대해 변경이 행해질 수 있다. 예를 들어, 사용자는 새로운 데이터를 생성하기 위해 또는 기존의 데이터를 수정 또는 삭제하기 위해 애플리케이션을 사용할 수 있다. 수정된 데이터는, 예를 들어, 연락처, 일정표 항목(calendar item), 오디오 클립, 비디오 클립, 워드 프로세싱 또는 스프레드시트 파일, 기타 파일, 기타 등등(이들로 제한되지 않음)의 아주 다양한 유형의 데이터 중 하나 이상을 포함할 수 있다.

[0054] 이러한 변경에 의해 그 다음에 종단점들 간에 동기화되어 있는 메타데이터 또는 블로그 데이터에 대한 하나 이상의 변경이 일어날 수 있다. 예를 들어, 사용자가, 예를 들어, 비디오 편집 애플리케이션을 사용하여 새로운 비디오 파일을 생성하고 또 이 새로운 파일이 적어도 하나의 다른 종단점과 동기화되는 데이터의 일부인 것으로 가정하자. 동작(210)의 일부로서, 또는 다른 동작의 일부로서, 변경이 행해진 종단점에 의해 유지되는 메타데이터가 새로운 비디오 파일과 연관된 항목을 포함하도록 업데이트될 수 있다. 이러한 새로운 항목은, 적어도 어떤 구현들에서, 항목의 동기화와 관련된 데이터(아마도 버전 정보 또는 항목이 생성된 또는 나중에 업데이트된 때에 관한 데이터 등)는 물론 항목 자체에 관한 정보(아마도 비디오의 이름, 비디오의 텍스트 설명, 및 실제 비디오 데이터 또는 콘텐츠를 식별해주는 참조 등)(비디오 데이터 또는 콘텐츠가 메타데이터 피드 자체에 포함되어 있지 않은 경우)를 포함할 수 있으며, 기존의 항목이 수정될 때, 메타데이터 및 어쩌면 블로그 데이터도 이와 유사하게 업데이트될 수 있다. 기존의 항목이 삭제되는 경우, 예를 들어, 그 항목이 삭제된 것으로 표시되도록 메타데이터가 수정될 수 있고, 임의의 연관된 블로그 데이터가 삭제될 수 있다(또는 적어도 어떤 구현들에서 이용가능하게 유지될 수 있다).

[0055] 동작(215)의 예시적인 구현에서, 동작(210)의 일부로서 업데이트된 메타데이터가 서버 종단점(어쩌면 적어도 어떤 측면들에서 도 1을 참조하여 앞서 기술한 예시적인 서버 종단점(110)과 같은 서버 종단점)으로 전달될 수 있다. 이러한 전달은, 예를 들어, 메타데이터가 표현되는 방법 및 변경을 한 종단점이 이용할 수 있는 또는 수신측 종단점이 이용할 수 있는 네트워킹 또는 기타 통신 기능에 따라 다양한 방식으로 구현될 수 있다. 예를 들어, 메타데이터가 XML 문서로 구현되는 구현에서, 메타데이터는, XML 문서를 포함하고 또 서버 종단점에서 운영되거나 서버 종단점과 연관된 HTTP 서버로 전송되는 HTTP POST 요청을 사용하여, 서버 종단점으로 전달될 수 있다. 다른 구현에서, 변경을 행한 종단점은 업데이트된 메타데이터를, 예를 들어, 네트워크 공유(network share)를 통해 또는 종단점 자체와 연관된 HTTP 서버를 통해 특정의 장소에 제공할 수 있고, 서버 종단점은 그 장소로부터 메타데이터를 검색할 수 있다.

[0056] 적어도 어떤 구현들에서, 메타데이터 변경과 연관된 블로그 데이터가 서버 종단점으로 전달되지 않을 수 있다. 그 대신에, 블로그 데이터가, 예를 들어, 변경이 행해진 종단점에 (적어도 어떤 기간 동안) 보유될 수 있지만, 그에 제한되지 않는다. 예를 들어, 블로그 데이터가 종단점에서 생성된 비디오 데이터를 포함할 때, 동작(215)의 구현이 완료된 후에 비디오 데이터는 종단점에만 존재할 수 있다. 어떤 경우들에, 이러한 블로그 데이터는 나중에 언젠가 다른 종단점에 의해 검색 또는 획득될 수 있다.

[0057] 어떤 구현들에서, 블로그 탐색 서비스가 메타데이터 변경과 연관된 블로그 데이터에 대한 장소 정보를 포함하도록 동작(215)의 구현의 일부로서 업데이트될 수 있다. 다른 종단점은 그 다음에 그 종단점으로부터 블로그 데이터를 검색하는 것의 일부로서 블로그 탐색 서비스를 사용할 수 있다. 동일한 또는 다른 구현들에서, 다른 종단점이 다른 종단점으로 전달되는 메타데이터를 병합 또는 처리할 때 블로그 탐색 서비스가 업데이트될 수 있다. 이러한 동작이, 적어도 어떤 예들에서, 도 5를 참조하여 이하에 기술되는 동작(525)과 같은 동작에 의해 구현될 수 있다.



- [0058] 마지막으로, 동작(215)의 어떤 구현들에서, 업데이트된 메타데이터가 서버 중단점으로 전달될 수 있는 반면, 동일한 또는 다른 구현들에서 업데이트된 메타데이터가 그 대신에 또는 그에 부가하여 다른 서버 중단점 또는 다른 장치 중단점(도 1을 참조하여 이전에 설명한 장치 중단점 B(170) 및/또는 장치 중단점 C(180) 등)을 비롯한 하나 이상의 다른 중단점들로 전달될 수 있다. 예를 들어, 메타데이터가 SSE 정보를 포함하거나 사용하고 따라서 메타데이터가 임의의 중단점들(때때로 서버 중단점이 아님) 간에 전달 및 동기화될 수 있는 구현에서, 동작(215)의 적어도 어떤 구현들은 업데이트된 메타데이터를 각종의 다른 중단점들로 전달할 수 있다.
- [0059] 동작(220)의 예시적인 구현에서, 변경이 행해진 중단점은 이전에 서버 중단점(또는 다른 중단점)으로 전달되었던 메타데이터에 대한 장래의 업데이트를 통지받기 위해 가입하거나 또는 등록할 수 있다. 이러한 가입 또는 등록 후에, 중단점은 그 다음에 적어도 어떤 경우들에 업데이트된 메타데이터가 이전에 전달되었던 서버 중단점 등의 어떤 다른 중단점에서 메타데이터가 변경될 때 통지(다양한 방식으로 구현 또는 실현됨)를 수신할 수 있다. 앞서 설명한 바와 같이, 이렇게 함으로써 중단점은 데이터에 대한 변경을 능동적으로 폴링, 즉 검사할 필요없이 자신과 연관된 데이터에 대한 다른 변경들을 알아챌 수 있다.
- [0060] 어떤 경우들에 동작(220)을 실행하는 것의 일부로서 가입 또는 등록이 행해질 수 있는 반면, 동일한 (또는 다른) 가입 또는 등록이 다른 때에 행해질 수 있거나 전혀 행해질 수 없다.
- [0061] 동작(225)의 예시적인 구현에서, 중단점은 메타데이터 변경과 연관된 블로그에 대한 요청에 대해, 예를 들어, 블로그를 요청자에게 제공하는 것으로 응답할 수 있다. 즉, 예를 들어, 블로그를 참조하는 또는 블로그와 연관된 메타데이터 변경이 하나 이상의 다른 중단점들에 의해 수신 또는 처리된 후의 어떤 시점에서, 이들 다른 중단점들 중 하나 이상은 블로그에 포함된 데이터를 필요로 하거나 원하는지를 결정할 수 있다. 블로그 데이터가 메타데이터의 일부로서 전달되지 않는 구현들을 비롯한 어떤 구현들에서, 이러한 다른 중단점들은 그 다음에 블로그가 존재하는 다른 중단점에 블로그를 요청함으로써 블로그를 검색할 수 있다. 블로그가 요청되는 방식은 물론 블로그가 요청자에게 제공되는 방식도, 예를 들어, 블로그 탐색 서비스 및 블로그 다운로드 모듈과 관련하여 앞서 소개한 방식들은 물론 이하에서 더 상세히 설명되는 방식들 중 몇몇을 포함한 다양한 방식으로 구현될 수 있다.
- [0062] 이제 도 3을 참조하면, 다른 중단점으로부터의 데이터를 동기화시킬 때, 장치 중단점을 비롯한 중단점에 의해 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름(300)이 도시되어 있다. 도 3에 대한 이하의 설명은 다른 도면들을 참조하여 행해질 수 있다. 그렇지만, 도 3을 참조하여 기술되는 동작 흐름이 이들 다른 도면들을 참조하여 기술되는 구성요소들과 함께 사용되는 것으로 제한되는 것으로 의도하기 위한 것이 아님을 잘 알 것이다. 그에 부가하여, 도 3의 예시적인 동작 흐름이 특정의 실행 순서를 나타내고 있지만, 하나 이상의 대안의 실시예들에서, 동작들이 다른 순서로 되어 있을 수 있다. 게다가, 예시적인 동작 흐름이 다수의 단계들을 포함하고 있지만, 어떤 구현들에서 이들 동작 중 적어도 몇몇이 결합, 즉 동시에 실행될 수 있고 또 동일한 또는 다른 구현들에서 몇몇 단계들이 실행되지 않을 수 있다는 것을 잘 알 것이다.
- [0063] 동작(310)의 예시적인 구현에서, 중단점은 중단점이 얼마간 관심을 갖고 있는 메타데이터가 변경되었는지를 판정할 수 있다. 예를 들어, 중단점은, 어쩌면 도 2를 참조하여 앞서 설명한 것과 같은 방식으로, 동일한 메타데이터를 이전에 업데이트했을 수 있고 또 장래의 변경들을 통지받기 위해 등록했을 수 있다. 이 경우에 또는 다른 경우에, 이러한 중단점은 메타데이터의 사본을 유지하는 서버 중단점을 비롯한 어떤 다른 중단점으로부터 통지를 수신할 수 있다. 다른 구현에서, 중단점은 어떤 장소를 주기적으로 폴링할 수 있고 또 폴링 동작을 통해 그 장소에 있는 메타데이터가 변경되었는지를 판정할 수 있다. 또 다른 구현들에서, 이 동작이 수행되지 않을 수 있으며, 이러한 구현에서 중단점은 메타데이터가 변경되었는지 여부에 상관없이, 예를 들어, 예시적인 동작 흐름(300)에서의 다른 동작들 중 적어도 일부(메타데이터를 획득하는 것 및 메타데이터를 병합하는 것 등)를 주기적으로 실행할 수 있다. (이 동작 흐름과 연관된 설명은, 어떤 경우들에, 문제의 메타데이터에 변경이 존재하는 것으로 가정할 수 있으며, 변경이 실제로는 존재하지 않는 경우에, 각종의 동작들이 다르게 실행될 수 있거나 어쩌면 전혀 실행되지 않을 수 있다.)
- [0064] 동작(315)의 적어도 어떤 구현들에서, 중단점은 각종의 수단들을 통해 변경을 포함할 수 있는 메타데이터를 획득할 수 있다. 예를 들어, 어쩌면 도 1을 참조하여 앞서 설명한 장치 중단점 A(160)와 같은 장치 중단점을 비롯한 중단점은 어쩌면 도 1을 참조하여 역시 앞서 설명한 서버 중단점(110) 등의 서버 중단점의 일부이거나 그와 연관되어 있는 웹 서버에 대해 HTTP GET 요청을 개시할 수 있고, 그에 응답하여 하나 이상의 변경을 포함하고 있는 메타데이터를 수신할 수 있다. 다른 구현들에서, 메타데이터가 사전 예방적으로 다른 중단점에 의해 전달될 수 있거나, 각종의 다른 방식으로 획득될 수 있다. 어떤 구현들에서, 획득된 메타데이터가 특정의 메타

데이터에 의해 표현될 수 있는 모든 항목들을 포함할 수 있는 반면, 동일한 또는 다른 구현들에서, 적어도 어떤 경우들에 메타데이터가 항목들 중 일부만(예를 들어, 변경된 항목들만)을 포함할 수 있다.

[0065] 동작(320)의 예시적인 구현에서, 획득된 메타데이터가 로컬적으로 유지되는 메타데이터의 사본과 병합될 수 있다. 이러한 병합 동작은 획득된 메타데이터에 실시된 변경들을, 동작 흐름(300)을 실행하는 중단점에 의해 유지되거나 수정될 수 있는 메타데이터의 다른 사본에 포함시킬 수 있다. 이러한 메타데이터의 로컬 사본 또는 로컬 메타데이터 저장소는 다양한 방식으로(예를 들어, 적어도 어떤 구현들에서 데이터베이스에서의 행 또는 레코드로서, 파일 시스템에서의 하나 이상의 파일에 존재할 수 있는 XML 피드 (또는 다른) 문서의 사본으로서, 기타 등등으로) 유지될 수 있다.

[0066] 메타데이터가 SSE 정보를 포함하는 구현들에서, 이러한 병합 동작은 SSE 병합을 사용하여 구현될 수 있다. SSE 병합을 구현하는 한 방법의 간략화된 설명에서, "외래(foreign)" 항목들(이 예에서, 획득된 메타데이터에 의해 구현되는 항목들)을 "로컬(local)" 항목들(메타데이터를 병합시키는 중단점에 로컬적으로 유지되거나 그와 연관되어 있는 항목들)과 비교하는, SSE에 의해 명시된 프로세스를 따를 수 있다. 로컬 메타데이터 저장소에 존재하지 않는 새로운 외래 항목들은 일반적으로 로컬 메타데이터 저장소에 포함됨으로써 새로운 로컬 항목을 생성할 수 있다. 외래 항목이 로컬 항목에 대응하는 경우(아마도 양쪽 항목이 동일한 SSE 식별자를 갖기 때문임), 병합 프로세스는 하나 이상의 수단들을 사용하여 "당선 항목(winning item)" 및 "탈락 항목(losing item)"을 선택할 수 있다(예를 들어, 보다 최근에 업데이트된 항목을 당선 항목으로 선택함). 마지막으로, 로컬 항목이 당선 항목이 아닐 때, 로컬 항목은 당선 항목에 의해 구현되는 데이터를 포함하도록 업데이트될 수 있다.

[0067] 동작(325)의 적어도 어떤 구현들에서, 정보를 동기화하는 중이고 또 메타데이터를 획득한 중단점이 또한 메타데이터에 의해 참조되거나 식별되는 하나 이상의 블로그를 요구하거나 원하는지가 판정될 수 있다. 예를 들어, 메타데이터가 메타데이터와 연관된 적어도 어떤 정보를 포함하지 않고 그 대신에, 예를 들어, 다른 곳에 저장된 데이터 블로그들에 대한 하나 이상의 참조를 포함하는 구현들에서, 이 동작은 그 블로그들 중 하나 이상을 원하는지 또는 요구하는지를 판정할 수 있다. 단지 한 예에서, 동작 흐름(300)의 목적의 적어도 일부가 메타데이터 및 연관된 블로그 데이터 둘다를 동작 흐름(300)을 실행하고 있는 중단점에 동기화시키는 것인 경우에 이러한 수 있다. 블로그가 필요한 경우, 예시적인 동작 흐름은 동작(330)으로 진행할 수 있다. 블로그가 필요하지 않은 경우, 예시적인 동작 흐름이 종료할 수 있다.

[0068] 동작(330)의 예시적인 구현에서, 획득된 메타데이터에 의해 식별되거나 참조되는 하나 이상의 블로그가 검색될 수 있다. 검색할 블로그 또는 블로그들이 다양한 방식으로 식별될 수 있다. 예를 들어, 어떤 경우들에, 변경된 메타데이터 항목들과 연관된 모든 블로그들이 검색될 수 있다. 다른 구현들에서, 변경된 블로그들 중 일부 분만이 검색될 수 있으며, 어쩌면 특정의 크기 아래의 블로그만이 검색될 수 있거나, 검색할 블로그 또는 블로그들을 필터링 또는 결정하기 위해 어떤 다른 기준이 사용될 수 있다. 어떤 구현들에서, 하나의 메타데이터 항목이 하나의 블로그를 식별할 수 있는 반면, 동일한 또는 다른 구현들에서, 하나의 메타데이터 항목이 다수의 블로그들을 식별할 수 있다. 각각의 식별된 블로그는 그 다음에 하나 이상의 다른 방식으로 검색될 수 있다. 어떤 구현들에서, 특정의 파일들(이미지, 비디오, 또는 오디오 파일 등)의 트랜스코딩된 버전, 기타 등등과 같은, 식별된 블로그(들)의 하나 이상의 대안의 표현들이 검색될 수 있다. 마지막으로, 어떤 구현들에서, 도 4를 참조하여 이하에서 설명되는 동작들과 동일하거나 유사한 동작들을 사용하여 블로그가 검색될 수 있다. 다른 구현들에서, 블로그가 하나 이상의 다른 방식으로 검색될 수 있다.

[0069] 이제 도 4를 참조하면, 블로그(blob)를 검색 또는 다운로드하기 위해 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름(400)이 도시되어 있다. 도 4에 대한 이하의 설명은 다른 도면들을 참조하여 행해질 수 있다. 그렇지만, 도 4를 참조하여 기술되는 동작 흐름이 이들 다른 도면들을 참조하여 기술되는 구성요소들과 함께 사용되는 것으로 제한되는 것으로 의도하기 위한 것이 아님을 잘 알 것이다. 그에 부가하여, 도 4의 예시적인 동작 흐름이 특정의 실행 순서를 나타내고 있지만, 하나 이상의 대안의 실시예들에서, 동작들이 다른 순서로 되어 있을 수 있다. 게다가, 예시적인 동작 흐름이 다수의 단계들을 포함하고 있지만, 어떤 구현들에서 이들 동작 중 적어도 몇몇이 결합, 즉 동시에 실행될 수 있고 또 동일한 또는 다른 구현들에서 몇몇 단계들이 실행되지 않을 수 있다는 것을 잘 알 것이다.

[0070] 동작(410)의 예시적인 구현에서, 블로그 탐색 서비스(blob lookup service, BLS)가 블로그를 검색하는 것의 일부로서 사용되는지가 판정될 수 있다. 예를 들어, 이 판정은 행해질 수 있는 이유는, 적어도 부분적으로 어떤 구현들에서 중단점이 먼저 블로그 탐색 서비스를 사용하지 않고(또는 블로그 탐색 서비스를 사용하여) 블로그를 검색하려고 시도할지도 모르기 때문이다. 이러한 구현에서, 블로그 탐색 서비스는, 예를 들어, 블로그 탐

색 서비스를 사용하지 않고 블로브를 검색하려는 최초의 시도가 행해진 후에만 사용될 수 있다. 이러한 최초의 시도가 실패하는 경우, 블로브 탐색 서비스가 사용될 수 있다. 다른 구현에서, 블로브 탐색 서비스가 이용 가능하지 않거나, 판정이 다양한 다른 방식으로 행해질 수 있다. 블로브 탐색 서비스가 사용되는 경우, 동작 흐름(400)은 동작(415)으로 진행할 수 있다. 블로브 탐색 서비스가 사용되지 않는 경우, 동작 흐름은 동작(435)으로 진행할 수 있다.

[0071] 블로브 탐색 서비스가 사용되는 경우, 동작 흐름(400)은 동작(415)으로 진행할 수 있고, 예시적인 구현에서 동작(415)에서 종단점은 원하는 블로브와 연관된 하나 이상의 위치 지정자들에 대해 블로브 탐색 서비스에 질문 또는 질의를 할 수 있다. 이러한 질의는 각종의 통신 메카니즘 또는 수단을 사용하여 블로브 탐색 서비스로 전달될 수 있다. 어떤 구현들에서, 부가적인 정보가 또한 요청에 포함될 수 있다. 예를 들어, 어떤 구현들에서, 종단점은 또한 특정의 블로브가 하나 이상의 대안의 표현들로 제공되도록 요청할 수 있다. 예를 들어, 이동 전화는 (어쩌면 저속인 네트워크 연결을 통해 전송하기 더 쉽도록, 제한된 저장 공간을 갖는 장치에 저장하기 더 쉽도록, 기타 등등을 위해) 비교적 보다 작은 공간을 차지하는 블로브의 표현을 요청할 수 있다.

[0072] 이러한 요청에 응답하여, 블로브 탐색 서비스는 하나 이상의 "위치 지정자"를 제공할 수 있으며, 이 경우 위치 지정자는 일반적으로 블로브의 적어도 일부분을 제공할 수 있는 종단점의 ID(identification)를 포함할 수 있다. 예를 들어, 한 구현에서 블로브 탐색 서비스는 하나 이상의 위치 지정자를 제공할 수 있으며, 각각의 위치 지정자는 차례로 블로브 탐색 서비스가 요청된 블로브의 적어도 일부분을 제공할 수 있는 것으로 판정한 하나 이상의 종단점을 식별해주는 도메인 이름 또는 IP 주소를 포함한다.

[0073] 이러한 구현들 중 적어도 몇몇 구현들(꼭 모든 구현들이어야 하는 것은 아님)은 물론 다른 구현들에서, 블로브 탐색 서비스에 의해 제공된 하나 이상의 위치 지정자들은 또한 다른 정보도 포함할 수 있다. 예를 들어, 어떤 경우들에, 위치 지정자는, 적어도 부분적으로, 위치 지정자에 의해 식별된 종단점과 연락하는 방식 또는 블로브가 검색되는 방식을 결정하는 데 사용될지도 모르는 정보를 포함할 수 있다. 예를 들어, 위치 지정자는 어떤 종류의 피어-투-피어 또는 분산 데이터 전송 메카니즘, HTTP, FTP, 기타 등등과 같은, 사용될지도 모르는 지원되는 프로토콜들의 어떤 표시를 포함할 수 있다. 동일한 또는 다른 경우들에, 하나 이상의 위치 지정자들이 특정의 위치 지정자 또는 위치 지정자들을 선호되는 것으로 식별해줄 수 있는 선호도 또는 순서의 어떤 표시를 포함할 수 있으며, 이러한 선호되는 위치 지정자들이 먼저 (예를 들어, 다른 위치 지정자들이 사용되기 전에) 사용될 수 있다.

[0074] 동일한 또는 다른 구현들에서, 특정의 블로브의 일부분 또는 일부만이 특정의 종단점으로부터 이용가능할 수 있다. 예를 들어, 특정의 블로브의 처음 50%가 특정의 종단점을 사용하여 액세스될 수 있는 반면, 동일한 블로브의 나머지 50%는 다른 종단점을 사용하여 액세스될 수 있다. 이러한 구현들에서, 위치 지정자는 어떤 경우들에 블로브의 어떤 부분들이 위치 지정자와 연관된 특정의 종단점에 의해 제공될 수 있는지를 나타내는 정보를 포함할 수 있다.

[0075] 동일한 또는 또다른 구현들에서, 위치 지정자는 도 1을 참조하여 앞서 설명한 바와 같은 예시적인 지식 저장소 모듈(145)에 의해 제공되는 지식 등의 어떤 유형의 지식에 의해 도출되거나 제공되는 정보를 포함할 수 있다. 이러한 지식은, 예를 들어, 하나 이상의 위치 지정자를 제공하는 블로브 탐색 서비스가 특정의 위치 지정자를 사용하는 선호도를 나타낼 수 있게 해주거나(어쩌면 위치 지정자들이 보다 빠른 연결을 갖는, 위치 지정자를 요청하는 종단점에 물리적으로 또는 논리적으로 더 가까운, 기타 등등의 종단점을 식별해주기 때문임), 블로브 탐색 서비스가 자신이 제공하는 위치 지정자들을 다양한 다른 방식으로 수정 또는 변경할 수 있게 해줄 수 있다.

[0076] 또한 유의할 점은 위치 지정자가 앞서 설명한 것들을 비롯한 부가의 정보를 포함할 수 있는 반면, 동일한 경우들 또는 구현들 중 몇몇에서, 제공된 위치 지정자가 어떤 부가의 정보도 포함하지 않을 수 있다는 것이다. 이러한 경우들에 또 다른 경우들에, 도시되지 않은 것들을 비롯한 다른 동작들이 다른 수단들을 통해(사용할 통신 또는 전송 프로토콜을 결정하기 위한 위치 지정자에서 식별된 종단점과의 협상 프로세스, 기타 등등에 의해) 블로브를 검색할 방법을 결정할 수 있다.

[0077] 동작(420)의 적어도 어떤 구현들에서, 제공된 위치 지정자(들)가 블로브를 검색하는 데 사용될 수 있는지 여부가 판정될 수 있다. 예를 들어, 위치 지정자에서 식별된 종단점에 질의를 할 수 있거나, 블로브를 검색하라는, 블로브 검색을 개시하라는, 블로브의 상태 또는 이용가능성을 검사하라는, 기타 등등의 요청이 전송될 수 있다. 블로브가 검색될 수 없다고 판정될 수 있는 경우 - 어쩌면 위치 지정자가, 예를 들어, 오프라인인 종단점을 식별하는 경우, 또는 블로브가 어떤 다른 이유로 검색될 수 없는 경우 -, 적어도 어떤 구현들에



서, 동작 흐름(400)은 동작(430)으로 진행할 수 있다. 위치 지정자들 중 하나 이상이 블로브를 검색하는 데 사용될 수 있는 경우, 동작 흐름은 동작(425)으로 진행할 수 있다.

[0078] 동작(425)의 예시적인 구현에서, 위치 지정자(들)가 블로브를 검색하는 데 사용될 수 있다. 이 동작은 위치 지정자(들)에서 제공되는 정보, 지원되는 또는 이용가능한 통신 또는 파일 전송 프로토콜, 기타 등등(이들로 제한되지 않음)을 비롯한 다양한 특성들에 따라 다양한 방식으로 구현될 수 있다. 어떤 구현들에서, 블로브를 검색하는 것이, 적어도 부분적으로, 도 1을 참조하여 앞서 설명한 블로브 다운로드 모듈(150) 및 블로브 다운로드 모듈(162)과 유사하거나 그와 동일할 수 있는 블로브 다운로드 모듈을 사용하여 구현될 수 있다.

[0079] 예를 들어, 예시적인 블로브 다운로드 모듈을 사용하는 구현에서, 중단점은 위치 지정자(들)를 블로브 다운로드 모듈에 제공할 수 있고, 블로브 다운로드 모듈은 그 다음에 위치 지정자(들)에서의 정보를 평가하고 그 정보를 사용하여 요청된 블로브를 검색 또는 다운로드하는 하나 이상의 방식들을 결정할 수 있다. 비교적 간단한 경우에, 중단점 또는 블로브 다운로드 모듈은 단순히 하나의 위치 지정자를 사용하여, 블로브를 갖는 다른 중단점을 찾아내고 식별된 다른 중단점으로 어떤 유형의 네트워크 통신 - 어쩌면 HTTP GET 또는 파일 시스템 파일 검색 요청 등 - 을 전송하며 그에 응답하여 요청된 블로브를 수신할 수 있다. 이러한 구현에서 또는 다른 구현들에서, 이러한 요청이 성공하지 않거나 어떤 이유로(예를 들어, 어쩌면 연결이 저속임) 덜 바람직한 것으로 판정되는 경우, 하나 이상의 다른 위치 지정자가 그에 부가하여 또는 그 대신에 사용될 수 있다.

[0080] 다른 구현에서, 중단점 또는 블로브 다운로드 서비스는 서로 다른 중단점들에 대한 다수의 요청(어쩌면 하나의 블로브의 서로 다른 (또는 동일한) 부분들에 대한 개별적인 요청을 포함함)을 개시하기 위해 다수의 위치 지정자들을 사용할 수 있다. 어떤 경우들에, 다수의 중단점들로부터 동일한 블로브의 서로 다른 부분들을 획득하는 것은 블로브의 검색이 보다 신속하게 또는 임의의 특징의 하나의 중단점에 영향을 덜 미치고 완료될 수 있게 해주거나, 다른 이점들을 가질 수 있다.

[0081] 적어도 어떤 블로브 검색 구현들 또는 동작들에서, 전송될 데이터량을 최소화하기 위해 또는 다른 방식으로 블로브의 검색을 더 최적화하기 위해 하나 이상의 기법들이 사용될 수 있다. 예를 들어, 어떤 구현들은 보다 적은 데이터를 검색하기 위해 데이터 압축(ZIP 또는 기타 압축 기법들 등)을 사용할 수 있다. 동일한 또는 다른 구현들에서, 요청측 중단점에 이미 존재해 있을지도 모르는 블로브의 사본으로부터 변경된 블로브의 부분만이 하나 이상의 다른 중단점들에 요청되거나 그로부터 전달될 수 있다. 이러한 "차분 압축"은 변경된 데이터가 존재하는 블로브가 비교적 큰 경우에도 적은 양의 데이터만이 전달될 수 있게 해줄 수 있다.

[0082] 어떤 구현들에서, 중단점은 블로브 검색 또는 다운로드의 상태에 관하여 다른 서비스 또는 중단점에 알려줄 수 있다. 예를 들어, 중단점은 특정의 블로브의 일부 또는 그 전부를 검색했다는 것을 블로브 탐색 서비스에 알려줄 수 있다. 이러한 정보를 사용하여, 블로브 탐색 서비스는 동일한 블로브에 대한 어떤 다른 요청을 서비스하는 것의 일부로서 중단점을 식별해주는 위치 지정자를 제공할 수 있다. 어떤 경우들에, 이러한 정보를 제공하는 중단점은 블로브가 완전히 검색되었을 때에만 그렇게 할 수 있는 반면, 다른 경우들 또는 구현들에서 중단점은, 예를 들어 블로브의 25%가 검색되었을 때, 블로브의 50%가 검색되었을 때, 기타 등등에 블로브 탐색 서비스가 업데이트되도록, 블로브가 검색됨에 따라 주기적으로 정보를 제공할 수 있다. 동일한 또는 다른 구현들에서, 다른 중단점으로 전달되는 정보는 블로브의 어느 부분이 검색되었는지(예를 들어, 블로브의 처음 50%가 검색되었는지, 블로브의 처음 10% 및 마지막 10%가 검색되었는지, 기타 등등)의 식별표시를 포함할 수 있고, 이러한 식별표시가 차례로 다른 중단점들이 서로 다른 중단점들로부터 동일한 블로브의 일부분들을 검색할 수 있게 해주는 데 유용할 수 있다.

[0083] 동작(425)의 실행 동안에 블로브의 검색이 실패하는 경우, 어떤 경우들에 동작 흐름이 종료될 수 있다. 다른 구현들에서, 검색 실패가 앞서 설명한 동작(420)의 실패와 유사하게 또는 그와 동일하게 해석될 수 있다. 이러한 구현에서, 동작(425) 동안에 블로브 검색이 실패하는 경우, 동작 흐름은 동작(430)으로 진행할 수 있다(이러한 경로가 도 4에서 점선을 사용하여 도시되어 있음).

[0084] 블로브가 다양한 이유들 중 하나 이상으로 인해 검색될 수 없는 경우, 동작 흐름(400)은 어떤 경우들에 동작(430)으로 진행할 수 있다. 동작(430)을 실행하는 이러한 이유들은, 블로브 탐색 서비스에 의해 식별된 중단점들이 어쩌면 온라인이 아니거나 이용가능하지 않기 때문에 블로브를 제공할 수 없을 때와 같은, 이상에서 설명한 것들을 포함할 수 있다. 동작(430)의 예시적인 구현에서, 중단점은 블로브 캐쉬 서비스에게 요청된 블로브의 사본을 검색 및 캐싱하도록 요청할 수 있다. 어떤 구현들에서, 이러한 블로브 캐쉬 서비스는 적어도 어떤 다른 중단점들보다 더 신뢰성있게 액세스될 수 있으며, 따라서 블로브를 검색하고자 하는 중단점이 어떤 다른 (어쩌면 이용가능하지 않은) 중단점 대신에 사용할 수 있는 유용한 중간 또는 임시 블로브 장소로서 역할할 수

있다.

- [0085] 이러한 구현들 중 몇몇에서, 블로그 캐쉬 서비스는 요청을 받고 가능한 경우 블로그 자체를 검색할 수 있다. 예를 들어, 블로그 캐쉬 서비스는 최초로 식별된 중단점이 이용가능할 때까지 기다릴 수 있고 그 다음에 최초의 중단점으로부터 블로그를 검색할 수 있다(또는 중단점에게 블로그를 업로드하라고 지시할 수 있다). 블로그 캐쉬 서비스에 의해 수행될 수 있는 어떤 예시적인 블로그 캐쉬 서비스 구현들 및 동작들은, 특히 도 1 및 도 7을 참조하여 본 명세서의 다른 곳에서 설명된다.
- [0086] 어떤 구현들에서, 블로그가 블로그 캐쉬 서비스에 의해 저장되도록 요청하는 중단점은 나중에 다시 동작(415)을 실행하고, 블로그를 검색하는 데 사용될 수 있는 위치 지정자들이 있는지 블로그 탐색 서비스에 질의를 할 수 있다. 예를 들어, 중단점은 블로그 캐쉬 서비스가 원하는 블로그를 제공할 수 있다는 통지를 (어쩌면 도 1을 참조하여 앞서 설명한 통지 서비스(140)와 같은 것으로부터) 수신할 수 있다. 블로그 캐쉬 서비스가 블로그를 저장한 경우, 블로그 캐쉬 서비스는 자신이 블로그를 가지고 있음을 블로그 탐색 서비스에 이미 통지했을 수 있고, 따라서 블로그 탐색 서비스는 블로그를 블로그 캐쉬 서비스를 사용하여 이용가능한 것으로 식별해주는 위치 지정자를 요청측 중단점에게 제공할 수 있다. 궁극적으로, 최초의 중단점은 블로그 캐쉬 서비스로부터 블로그를 검색할 수 있다.
- [0087] 다른 구현들에서, 다른 보다 신뢰성있거나 이용가능한 중단점들(어쩌면 도 1을 참조하여 앞서 설명한 것과 같은 클라우드 스토리지 서비스 등)을 비롯한 다른 중단점들이 블로그 캐쉬 서비스 대신에 중단점이 블로그를 검색할 수 있는 장소로서 사용될 수 있다.
- [0088] 블로그 탐색 서비스가, 예를 들어, 동작(410)의 일부로서 사용되지 않는 것으로 판정된 경우, 동작 흐름(400)은 동작(435)으로 진행할 수 있으며, 동작(435)에서 블로그 탐색 서비스를 사용하지 않을지도 모르는 다양한 다른 메카니즘들을 사용하여 블로그가 검색될 수 있다. 예를 들어, 어떤 구현들에서, 하나 이상의 중단점들은 블로그 탐색 서비스와 연관되어 있지 않거나 블로그 탐색 서비스가 모르고 있을 수 있는 어떤 저장소를 사용하여 블로그를 검색하려고 시도할 수 있다. 이러한 경우에, 또는 다른 경우들에, HTTP, FTP, 피어-투-피어 "BitTorrent"-방식 통신과 같은 앞서 설명한 메카니즘들 또는 기타 메카니즘들을 비롯한 다양한 메카니즘들 중 임의의 것을 사용하여 블로그가 검색될 수 있다. 그에 부가하여, 블로그 캐쉬 서비스에 의해 제공되는 기능 또는 기타 서버 기능과 같은 다른 곳에서 제공되는 기능도 역시 블로그 탐색 서비스를 사용하지 않는 적어도 어떤 구현들에서 사용될 수 있다.
- [0089] 유의할 점은 어떤 구현들에서 블로그를 검색하는 다른 방법들이 도 4를 참조하여 앞서 설명한 예시적인 동작 흐름 대신에 또는 그에 부가하여 사용될 수 있다는 것이다. 이들 다른 방법 중 적어도 어떤 방법들은 또한 도 4를 참조하여 설명한 예시적인 동작들 중 하나 이상을 실행할 수 있거나, 예를 들어, 도 1을 참조하여 설명한 모듈들 및 서비스들을 비롯한 하나 이상의 다른 모듈들 또는 서비스들을 사용할 수 있다. 이들 다른 방법들 중 적어도 어떤 방법들은 또한 다른 명령어들을 실행하거나 다른 모듈들 및 서비스들을 사용할 수 있다. 단지 일례로서, 중단점은 블로그 다운로드 모듈(어쩌면 도 1을 참조하여 앞서 설명한 블로그 다운로드 모듈(150) 또는 블로그 다운로드 모듈(162) 등)에게 특정의 블로그를 획득하도록 요청할 수 있다. 어떤 구현들에서, 블로그 다운로드 모듈들은 각각이 어쩌면 서로 다른 방식으로 요청된 블로그를 찾아낼 수 있는 및/또는 획득할 수 있는 하나 이상의 플러그인 "드라이버"를 가질 수 있다. 예를 들어, 한 드라이버는 블로그 탐색 서비스를 사용하여 위치 지정자들의 세트를 획득하고 이어서 하나의 위치 지정자를 사용하여 블로그를 검색할 수 있고, 다른 드라이버도 역시 블로그 탐색 서비스를 사용할 수 있고 또 다수의 위치 지정자들을 (어쩌면 병렬로, 또는 BitTorrent나 다른 유사한 유형의 메카니즘과 함께) 사용하여 블로그를 검색할 수 있으며, 또다른 드라이버는 블로그 탐색 서비스를 사용하지 않을 수 있고 또 각종의 다른 방식들로 블로그를 검색할 수 있으며, 기타 등등일 수 있다. 이러한 구현들 중 적어도 몇몇 구현에서, 블로그 다운로드 모듈은 다양한 방식으로 사용할 드라이버를 결정할 수 있다. 예를 들어, 블로그 다운로드 모듈은 드라이버가 블로그를 검색하는 데 얼마나 걸릴지를 추정하기 위해 각각의 드라이버에게 질의를 할 수 있고 각각의 드라이버는 이어서 추정된 시간 또는 어쩌면 드라이버가 블로그를 검색할 수 없음을 나타내는 어떤 값을 반환할 수 있다. 이러한 반환된 값을 사용하여, 블로그 다운로드 모듈은 하나 (또는 그 이상의) 드라이버를 선택하고 선택된 드라이버(들)에게 블로그를 실제로 검색하라고 지시할 수 있다.
- [0090] 이제 도 5를 참조하면, 메타데이터를 유지 및 제공할 때 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일련화된 동작 흐름(500)이 도시되어 있다. 도 5에 대한 이하의 설명은 다른 도면들을 참조하여 행해질 수 있다. 그렇지만, 도 5을 참조하여 기술되는 동작 흐름이 이들 다른 도면들을 참조하여 기술되는 구성요소들과 함께 사

용되는 것으로 제한되는 것으로 의도하기 위한 것이 아님을 잘 알 것이다. 그에 부가하여, 도 5의 예시적인 동작 흐름이 특정의 실행 순서를 나타내고 있지만, 하나 이상의 대안의 실시예들에서, 동작들이 다른 순서로 되어 있을 수 있다. 게다가, 예시적인 동작 흐름이 다수의 단계들을 포함하고 있지만, 어떤 구현들에서 이들 동작 중 적어도 몇몇이 결합, 즉 동시에 실행될 수 있고 또 동일한 또는 다른 구현들에서 몇몇 단계들이 실행되지 않을 수 있다는 것을 잘 알 것이다.

[0091] 동작(510)의 예시적인 구현에서, 도 1을 참조하여 앞서 설명한 서버 중단점(110) 등의 중단점은 다른 중단점으로부터 메타데이터를 획득할 수 있다. 예를 들어, 메타데이터를 획득하는 중단점은 어떤 구현들에서, 예를 들어, 도 2의 동작(215)을 참조하여 앞서 설명한 바와 같이, 다른 중단점들에 의해 전달된 변경들을 갖는 메타데이터를 수신하는 서버 중단점일 수 있다. 게다가, 어떤 구현들에서, 동작 흐름(400)을 참조하여 본 명세서에 설명한 동작들 중 적어도 몇몇 동작들이, 역시 도 1을 참조하여 앞서 설명한 바와 같은 예시적인 저장 서비스(120)에 의해 구현될 수 있다. (본 설명 중 일부에서 동작 흐름(400)에서 메타데이터를 수신하는 중단점을 "서버 중단점"이라고 지칭할 수 있지만, 메타데이터를 수신하는 중단점이 꼭 서버 중단점일 필요가 없고, 다른 곳에서 어쩌면 서버 중단점과 연관된 것으로 설명한 기능의 적어도 일부 또는 그 전부를 꼭 가지고 있을 필요가 없으며, 기타 등등일 수 있다는 것을 잘 알 것이다.)

[0092] 메타데이터를 전달하는 데 적합한 각종의 통신 메커니즘들 중 임의의 것을 사용하여 메타데이터가 획득될 수 있다. 예를 들어, 어떤 구현들에서, 수신측 또는 획득측 중단점과 연관된 HTTP 서버는 어떤 URL에 있는 메타데이터를 포함하는 HTTP POST 요청을 받을 수 있다. 동일한 또는 다른 구현들에서, 다른 중단점은 FTP 또는 하나 이상의 파일 공유 프로토콜, 이메일, 기타 등등과 같은 다른 프로토콜을 사용하여 메타데이터를 전달할 수 있다. 어떤 구현들에서, 수신측 중단점은 메타데이터를 전송하는 다른 중단점에 의존하기 보다는 사전 예방적으로 메타데이터를 검색할 수 있다. 예를 들어, 획득측 중단점은 어떤 다른 중단점에 대해 HTTP GET 요청을 개시하고 그에 응답하여 메타데이터를 공급받을 수 있다.

[0093] 동작(515)의 적어도 어떤 구현들에서, 획득된 메타데이터가 서버 중단점에 의해 유지되는 메타데이터의 사본과 병합될 수 있다. 이러한 병합 동작으로 인해, 획득된 메타데이터에 실시된 변경들이 동작 흐름(500)을 실행하는 중단점에 의해 유지되고 수정될 수 있는 메타데이터의 다른 사본에 포함될 수 있다. 이러한 메타데이터의 로컬 사본은 다양한 방식으로 - (적어도 어떤 구현들에서) 데이터베이스에서의 행 또는 레코드로서, XML 피드 (또는 다른) 문서의 사본으로서, 기타 등등을 포함함 - 유지될 수 있다. SSE 정보를 포함하는 메타데이터를 사용하는 구현들에서, 획득된 메타데이터가 SSE 병합 기법들을 사용하여 로컬 메타데이터와 병합될 수 있다. 어떤 구현들에서, 이러한 SSE 병합 기법들은, 특히 도 3의 동작(320)을 참조하여 앞서 설명한 SSE 병합 기법들과 동일하거나 그와 유사할 수 있다.

[0094] 동작(520)의 예시적인 구현에서, 문제의 메타데이터가 변할 때 통지를 받기 위해 이전에 등록 또는 가입을 한 하나 이상의 중단점들은 하나 이상의 통지 수단을 사용하여 통지를 받을 수 있다. (이러한 통지를 수신한 후에, 중단점은 어떤 구현들에서 서버 중단점으로부터 새로 업데이트된 메타데이터를 획득하고 새로 업데이트된 메타데이터에서의 변경들을 중단점 자신의 로컬 메타데이터 저장소와 병합하기 위한 동작들의 실행을 개시할 수 있다. 이러한 동작들은, 어쩌면 도 3을 참조하여 앞서 설명한 동작들 중 적어도 몇몇 동작들을 포함할 수 있다.) 통지를 제공하지 않는 구현들을 비롯한 적어도 어떤 다른 구현들에서, 어떤 가입자도 통지를 받지 않을 수 있다.

[0095] 동작(525)의 어떤 구현들에서, 메타데이터를 획득 또는 사용하는 중단점들이 메타데이터에 의해 참조되거나 메타데이터와 연관된 블로그들을 검색할 수 있도록, 블로그 탐색 서비스는 획득된 메타데이터에 의해 식별되는 블로그 또는 블로그들에 대한 위치 정보로 업데이트될 수 있다. 즉, 예를 들어, 획득된 메타데이터가 새로운 블로그를 참조하는 새로운 항목을 포함하는 것으로 가정하자. 동작 흐름(500)이 실행될 때, 참조된 블로그가 업데이트된 메타데이터를 발생하여 서버 중단점으로 전달했던 중단점에만 존재하는 경우가 있을 수 있다. 어느 중단점(서버 중단점 또는 다른 장치 중단점들을 포함함)이라도 블로그를 검색할 수 있기 위해, 블로그의 (초기) 위치가 본 명세서의 다른 곳에서 설명한 블로그 탐색 서비스들과 같은 블로그 탐색 서비스를 비롯한 블로그 탐색 서비스에 의해 저장될 수 있다.

[0096] 마지막으로, 동작(530)의 예시적인 구현에서, 병합 동작(515)의 일부로서 생성되는 업데이트된 메타데이터가 하나 이상의 중단점들로 전달될 수 있다. 예를 들어, 메타데이터에 의해 표현되는 데이터를 동기화시키는 다른 중단점은 업데이트된 메타데이터에 대한 요청을 서버 중단점으로 전송할 수 있다. 이러한 요청은 상대방 중단점이 메타데이터가 변경되었다는 통지를 수신한 후에, 상대방 중단점이 서버 중단점을 폴링하고 메타데이



터가 변경되었음을 발견할 때, 기타 등등에 전송될 수 있다. 서버 중단점은 본 명세서의 다른 곳에서 앞서 설명한 방식들을 비롯한 다양한 방식으로 (HTTP GET 또는 POST 요청 및 응답, 파일 공유 프로토콜, FTP, 이메일, 어떤 다른 통신 메카니즘 또는 메카니즘들, 기타 등등을 사용하여) 상대방 중단점에 메타데이터를 제공할 수 있다.

[0097] 이제 도 6을 참조하면, 메타데이터에 의해 참조되는 블로그와 연관된 위치 및 위치 지정자(locator)를 저장 및 제공할 때 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름(600)이 도시되어 있다. 도 6에 대한 이하의 설명은 다른 도면들을 참조하여 행해질 수 있다. 그렇지만, 도 6을 참조하여 기술되는 동작 흐름이 이들 다른 도면들을 참조하여 기술되는 구성요소들과 함께 사용되는 것으로 제한되는 것으로 의도하기 위한 것이 아님을 잘 알 것이다. 그에 부가하여, 도 6의 예시적인 동작 흐름이 특정의 실행 순서를 나타내고 있지만, 하나 이상의 대안의 실시예들에서, 동작들이 다른 순서로 되어 있을 수 있다. 게다가, 예시적인 동작 흐름이 다수의 단계들을 포함하고 있지만, 어떤 구현들에서 이들 동작 중 적어도 몇몇이 결합, 즉 동시에 실행될 수 있고 또 동일한 또는 다른 구현들에서 몇몇 단계들이 실행되지 않을 수 있다는 것을 잘 알 것이다.

[0098] 동작(610)의 예시적인 구현에서, 중단점은 특정의 블로그와 연관된 하나 이상의 위치들을 받을 수 있다. 일반적으로, 이들 위치는 특정의 블로그의 적어도 일부가 검색될 수 있는 중단점을 지정할 수 있다. 적어도 어떤 구현들에서, 블로그 또는 블로그들은 하나 이상의 블로그 식별자들을 사용하여 식별될 수 있다. 하나 이상의 위치를 받는 중단점은 도 1을 참조하여 앞서 설명한 예시적인 서버 중단점(110)과 같은 적어도 어떤 서버 중단점들을 포함할 수 있다. 게다가, 동작 흐름(600)과 연관된 동작들 중 적어도 몇몇 동작이, 적어도 어떤 구현들에서, 역시 도 1을 참조하여 앞서 설명한 예시적인 블로그 탐색 서비스(125)와 같은 블로그 탐색 서비스에 의해 구현될 수 있다. 동작 흐름(600)의 동작들 중 몇몇 동작이 본 명세서에서 예시적인 서버 중단점 또는 예시적인 블로그 탐색 서비스와 연관되어 있는 것으로 설명될 수 있지만, 이들 동작이 서버 중단점 또는 블로그 탐색 서비스에 의해 꼭 구현되거나 실행되어야 하는 것은 아니며 그 대신에 각종의 중단점 또는 그 중단점들에 포함된 모듈들 또는 서비스들 중 하나 이상에 의해 구현되거나 실행될 수 있다는 것을 잘 알 것이다.

[0099] 하나 이상의 블로그 위치들을 각종의 다른 중단점들 및 프로세스들로부터 받거나 검색할 수 있다. 예를 들어, 서버 중단점 또는 저장 서비스는 다른 중단점으로부터의 메타데이터를 처리할 때, 수신한 메타데이터에 의해 식별되는 하나 이상의 블로그들과 연관된 하나 이상의 위치들을 이 동작에 제공할 수 있다(이러한 동작은, 예를 들어, 도 5를 참조하여 앞서 설명한 동작(525)와 연관되어 있을 수 있음). 다른 예에서, 중단점이 도 4를 참조하여 앞서 설명한 동작들을 사용하는 것 등에 의해, 어쩌면 메타데이터 및 메타데이터와 연관된 블로그들을 동기화시키는 것의 일부로서, 하나 이상의 다른 중단점들로부터의 블로그의 일부 또는 그 전부를 획득할 때, 그 중단점은 또한, 다른 중단점들이 (예를 들어, 블로그가 최초로 생성된 중단점으로부터 블로그를 수신하는 것에 부가하여) 현재 역시 블로그를 획득한 중단점으로부터 블로그의 일부 또는 그 전부를 검색할 수 있도록, 자신의 위치로 블로그 탐색 서비스를 업데이트시킬 수 있다.

[0100] 현재 블로그의 적어도 일부를 갖는 중단점의 위치에 부가하여, 블로그 탐색 서비스를 업데이트시키는 중단점은 또한 다른 정보도 블로그 탐색 서비스에 제공할 수 있다. 이러한 다른 정보는, 예를 들어, 특정의 중단점에서 이용가능한 블로그의 일부(또는 일부들)의 식별표시(또는 블로그 전체가 이용가능하다는 표시), 특정의 중단점이 블로그의 하나 이상의 대안의 표현들을 제공할 수 있다는 어떤 표시, 기타 등등을 포함할 수 있다.

[0101] 동작(615)의 적어도 어떤 구현들에서, 수신한 위치(어쩌면 다른 수신된 데이터)가 각종의 수단들 또는 메카니즘들(데이터베이스에서의 행 또는 레코드, 기타 등등을 포함함) 중 하나 이상을 사용하여 블로그 탐색 서비스에 의해 저장될 수 있다.

[0102] 동작(620)의 예시적인 구현에서, 블로그 탐색 서비스는 하나 이상의 블로그들과 연관된 위치 지정자들에 대한 요청을 수신할 수 있다. 이러한 요청들은 각종의 중단점들, 모듈들 또는 서비스들로부터 전송 또는 전달될 수 있다. 예를 들어, 장치 중단점은 특정의 메타데이터에 의해 식별된 블로그를 동기화시킬 수 있도록 특정의 블로그의 위치 지정자에 대한 요청을 전송할 수 있다. (장치 중단점은 그 다음에 반환된 위치 지정자들을 사용하여 블로그를 실제로 검색할 수 있다.) 다른 예에서, 블로그 캐쉬 서비스 또는 클라우드 스토리지 서비스는 하나 이상의 블로그들을 검색할 수 있도록 위치 지정자들을 요청할 수 있다.

[0103] 동작(625)의 어떤 구현들에서, 블로그 탐색 서비스는 특정의 요청된 블로그와 연관된 하나 이상의 위치 지정자들을 식별할 수 있다. 어떤 구현들에서, 이 동작은 적어도 부분적으로 특정의 블로그를 탐색하고(어쩌면 블로그와 연관된 블로그 식별자를 사용함) 요청된 블로그를 제공할 수 있는 것으로 등록되어 있는 모든 중단점에 대한 위치 지정자를 식별하는 것에 의해 실행될 수 있다. 예를 들어, 3개의 중단점이 요청된 블로그를 갖는 것으

로 이미 등록되어 있는 경우, 이 동작은 3개의 중단점 각각에 대해 하나씩, 3개의 위치 지정자를 식별할 수 있다.

[0104] 어떤 구현들에서, 식별된 위치 지정자들은 또한, 블로브와 연관된 중단점을 식별하는 수단일 뿐만 아니라, 부가의 정보도 포함할 수 있다. 예를 들어, 어떤 위치 지정자들은 블로브의 어느 부분 또는 일부분이 특정의 중단점으로부터 검색될 수 있는지에 관한 정보를 포함할 수 있으며, 동일한 또는 다른 위치 지정자들은 블로브를 검색하기 위해 특정의 중단점에서 사용될 수 있는 하나 이상의 통신 프로토콜에 관한 정보를 포함할 수 있고, 기타 등등일 수 있다.

[0105] 어떤 구현들에서 블로브 탐색 서비스가 단순히, 예를 들어, 특정의 블로브를 제공할 수 있는 각각의 및 모든 중단점에 대한 위치 지정자를 식별할 수 있는 반면, 다른 구현들에서 위치 지정자들을 식별하는 프로세스는, 예를 들어, 부가의 정보 또는 지식을 사용할 수 있다. 예를 들어, 블로브 탐색 서비스는 메타데이터에 관한, 서버 중단점(들) 및 장치 중단점(들)에 관한, 기타 등등에 관한 각종의 지식을 사용하여 (어쩌면 가장 적합한) 위치 지정자들을 필터링, 우선순위 지정, 또는 다른 방식으로 식별할 수 있다. 어떤 구현들에서, 이러한 지식은 적어도 부분적으로 도 1을 참조하여 앞서 설명한 예시적인 지식 저장소 모듈(145) 등의 지식 저장소에 의해 제공될 수 있다.

[0106] 예를 들어, 다수의 중단점들이 특정의 블로브의 일부분들을 제공할 수 있는 것으로 이전에 등록된 것으로 가정하자. 게다가, 제공측 중단점들 중 하나가 고속, 고대역폭 네트워크 연결을 사용하여 네트워크에 연결되는 경우가 많은 워크스테이션급 데스크톱 컴퓨터에 의해 구현되었고, 제공측 중단점들 중 다른 하나가 다른 연결 특성들을 갖는 각종의 네트워크들을 사용하여 연결을 하는 랩톱 컴퓨터이며, 제3의 제공측 중단점이 종종 커져 네트워크에 연결되지만 데이터 전송이 단위별 과금(per-unit monetary charge)과 연관되어 있는 비교적 저속인 연결을 사용하는 이동 전화라는 지식을 지식 저장소가 갖고 있는 것으로 가정하자. 이러한 예에서, 동작(625)의 적어도 어떤 구현들은 이 지식은 물론 어쩌면 다른 지식도 사용하여 식별된 위치 지정자 또는 위치 지정자들을 필터링 또는 우선순위 지정할 수 있다. 예를 들어, 한 구현에서, 3개의 중단점 모두에 대한 위치 지정자들이 식별될 수 있지만, 이들 위치 지정자가 데스크톱 컴퓨터가 먼저 사용되어야 하고 랩톱 컴퓨터가 그 다음에 사용되어야 하며 이동 전화는 단지 최후의 수단으로 사용되어야 한다는 것을 나타내는 우선순위 정보를 포함하거나 그와 함께 전달될 수 있다. 다른 예에서, 식별된 위치 지정자들은 심지어 특정의 중단점(이 예에서, 이동 전화 등)을 포함하지 않을 수 있으며, 그 대신에 데스크톱 및 랩톱 컴퓨터에 대한 위치 지정자들만이 식별될 수 있다.

[0107] 다른 정보 또는 기준도 역시 하나 이상의 위치 지정자들을 식별하는 데 또 심지어 어쩌면 어떤 위치 지정자도 전혀 반환되지 않는지를 판정하는 데 사용될 수 있다. 예를 들어, 메타데이터가 위험하거나, 적절하지 않거나, 또는 어떤 이유로 바람직하지 않은 것으로 판정된 블로브를 참조하는 항목을 포함하는 것으로 가정하자(예를 들어, 블로브가 바이러스를 포함하는 실행가능 코드를 포함할 수 있거나, 불쾌한 콘텐츠를 갖는 비디오 데이터를 포함할 수 있거나, 기타 등등일 수 있음). 이러한 경우에, 동작(625)의 구현은 하나 이상의 중단점이 문제의 블로브를 제공할 수 있는 것으로 등록되어 있더라도 특정의 블로브에 대한 어떤 위치 지정자도 식별하지 못할지도 모른다. 이와 같이, 데이터를 동기화 또는 전달할 때 전송될 수 있는 콘텐츠를 필터링 또는 제어하기 위해 블로브 탐색 서비스가 사용될 수 있다. 이러한 제어는, 예를 들어, 중단점들이 어쩌면 중앙집중형 중단점의 개입 없이 데이터를 교환할 수 있는 분산형 또는 피어-투-피어 동기화 시스템에서 구현하기 어렵거나 불가능할 수 있다.

[0108] 마지막으로, 동작(630)의 예시적인 구현에서, 예를 들어, 동작(625)에서 식별되는 위치 지정자 또는 위치 지정자들은 동작(620)의 일부로서 수신된 요청을 했던 중단점을 비롯한 하나 이상의 중단점들로 전달 또는 제공될 수 있다.

[0109] 이제 도 7을 참조하면, 캐쉬에 블로브를 저장하고 캐쉬로부터 블로브를 제공할 때 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름(700)이 도시되어 있다. 도 7에 대한 이하의 설명은 다른 도면들을 참조하여 행해질 수 있다. 그렇지만, 도 7을 참조하여 기술되는 동작 흐름이 이들 다른 도면들을 참조하여 기술되는 구성요소들과 함께 사용되는 것으로 제한되는 것으로 의도하기 위한 것이 아님을 잘 알 것이다. 그에 부가하여, 도 7의 예시적인 동작 흐름이 특정의 실행 순서를 나타내고 있지만, 하나 이상의 대안의 실시예들에서, 동작들이 다른 순서로 되어 있을 수 있다. 게다가, 예시적인 동작 흐름이 다수의 단계들을 포함하고 있지만, 어떤 구현들에서 이들 동작 중 적어도 몇몇이 결합, 즉 동시에 실행될 수 있고 또 동일한 또는 다른 구현들에서 몇몇 단계들이 실행되지 않을 수 있다는 것을 잘 알 것이다.

- [0110] 동작(710)의 예시적인 구현에서, 중단점은 특정의 블로브를 저장 또는 캐싱하라는 요청을 받거나 수신할 수 있다. 이러한 요청을 수신하는 중단점은 도 1을 참조하여 앞서 설명한 예시적인 서버 중단점(110)과 같은 적어도 어떤 서버 중단점들을 포함할 수 있다. 게다가, 동작 흐름(700)과 연관된 동작들 중 적어도 몇몇 동작이, 적어도 어떤 구현들에서, 역시 도 1을 참조하여 앞서 설명한 예시적인 블로브 캐쉬 서비스(130)와 같은 블로브 캐쉬 서비스에 의해 구현될 수 있다. 동작 흐름(700)의 동작들 중 몇몇 동작이 본 명세서에서 예시적인 서버 중단점 또는 예시적인 블로브 캐쉬 서비스와 연관되어 있는 것으로 설명될 수 있지만, 이들 동작이 서버 중단점 또는 블로브 캐쉬 서비스에 의해 꼭 구현되거나 실행되어야 하는 것은 아니며 그 대신에 각종의 중단점 또는 그 중단점들에 포함된 모듈들 또는 서비스들 중 하나 이상에 의해 구현되거나 실행될 수 있다는 것을 잘 알 것이다.
- [0111] 특정의 블로브를 캐싱하라는 요청이 서버 중단점들 및 장치 중단점들 둘다를 비롯한 다양한 중단점들로부터 수신될 수 있다. 예를 들어, 장치 중단점은, 일례에서, 그 블로브 자체를 다른 중단점으로부터 획득할 수 없을 때(예를 들어, 앞서 설명한 바와 같은 경우처럼, 2개의 장치 중단점이 동시에 온라인되어 있지 않을 때) 블로브를 캐싱하라는 요청을 전송할 수 있다. 다른 예에서, 저장 서비스 또는 기타 서비스 또는 모듈은 다양한 이유들 중 하나 이상의 이유로 특정의 블로브를 캐싱하라는 요청을 전송할 수 있다. 예를 들어, 어쩌면 도 1을 참조하여 앞서 설명한 지식 저장소 모듈(145)와 같은 지식 저장소에 의해 유지되는 지식을 사용하여, 저장 서비스는 특정의 중단점이 종종 이용가능하지 않다는 지식을 가지고 있을 수 있다. 이러한 경우에 또 어쩌면 다른 경우들에, 이러한 중단점이 새로운 또는 변경된 블로브들과 연관된 메타데이터를 제공할 때, 저장 서비스는 사전 예방적으로 (심지어 다른 중단점이 블로브를 요청하기 전에도) 특정의 블로브 또는 블로브들을 캐싱하라고 요청할 수 있다. 이러한 구현에서는 이러한 블로브를 검색하고자 하는 장치 중단점이 블로브 자체를 캐싱하라는 요청을 하는 것이 필요없게 될 수 있다.
- [0112] 동작(715)의 어떤 구현들에서, 블로브 캐쉬 서비스는 요청된 블로브를 검색할 수 있다. 블로브를 검색하는 것은 다양한 방식으로 구현될 수 있다. 어떤 구현들에서, 블로브 캐쉬 서비스는 도 4를 참조하여 앞서 설명한 동작들과 동일하거나 유사한 하나 이상의 동작들을 실행할 수 있다. 이러한 동작들은, 예를 들어, 블로브 탐색 서비스를 사용하여 블로브가 이용가능한 하나 이상의 장소들을 식별할 수 있고 이러한 장소들로부터 블로브를 검색할 수 있다.
- [0113] 동일한 또는 다른 구현들에서, 중단점들은 주기적으로 블로브 캐쉬 서비스와 통신을 할 수 있거나, 블로브 캐쉬 서비스와의 통신이 필요하다는 통지를 수신할 수 있고, 또 블로브 캐쉬 서비스가 요청하는 블로브들을 제공 또는 업로드할 수 있다. 예를 들어, 장치 중단점은 블로브 캐쉬 서비스가 장치 중단점에서 이용가능한 특정의 블로브를 원한다는 통신 또는 통지를 수신할 수 있고, 따라서 동작(715)의 일부로서 블로브가 블로브 캐쉬 서비스에 의해 검색될 수 있게 해줄 수 있거나, 동작(715)의 다른 구현의 일부로서 사전 예방적으로 블로브를 블로브 캐쉬 서비스에 업로드 또는 제공할 수 있다.
- [0114] 블로브 캐쉬 서비스는 어떤 구현들에서 동기화 시스템, 메타데이터, 중단점들, 기타 등등에 관한 지식(도 1을 참조하여 앞서 설명한 예시적인 지식 저장소(145) 등의 지식 저장소에 의해 유지되는 지식을 포함함)을 사용하여 캐싱되어야 하는 블로브들을 보다 효율적으로 검색할 수 있다. 단지 일례에서, 블로브 캐쉬 서비스는 어떤 다른 중단점에 의해 요청된 블로브를 캐싱하려고 즉각 시도하지 않을지도 모른다. 그 대신에, 블로브 캐쉬 서비스는, 예를 들어, 블로브를 제공할 수 있는 하나 이상의 중단점들이 이용할 수 있는 미사용 대역폭(free bandwidth)이 있을 때까지, 비교적 저렴한 대역폭이 이용가능할 때까지, 또는 요청을 서비스하기에 충분한 컴퓨팅 능력을 갖는 중단점들이 이용가능할 때까지, 기타 등등 때까지 기다릴지도 모른다.
- [0115] 동작(720)의 한 구현에서, 검색된 블로브는 각종의 저장 메커니즘들 또는 수단들 중 임의의 것을 사용하여 저장될 수 있다. 어떤 구현들에서, 블로브들은 데이터베이스에 저장될 수 있거나, 파일 시스템에 파일로서 저장될 수 있거나, 어떤 다른 방식으로 저장될 수 있다. 적어도 어떤 구현들에서, 블로브 캐쉬 서비스는 각종의 캐싱 기법들을 사용하여, 예를 들어, 특정의 블로브가 얼마 동안 저장되어야 하는지를 결정할 수 있다. 예를 들어, 블로브 캐쉬 서비스의 한 구현은 그의 이용가능한 저장 공간이 채워질 때까지 블로브를 저장할 수 있고, 그 다음에 하나 이상의 알고리즘(LRU 알고리즘을 포함함)을 사용하여, 부가의 블로브들에 대한 공간이 이용가능하도록 어느 블로브를 삭제할지를 결정할 수 있다. 어떤 구현들에서, 최근에 검색 또는 액세스된 블로브들이 유지될 수 있는 반면 최근에 검색되지 않은 블로브들이 삭제될 수 있다. 동일한 또는 다른 구현들에서, 블로브들은 몇개의 중단점들이 블로브를 검색하고자 할 수 있는지를 정의하는 참조 횟수와 연관될 수 있고, 어떤 경우들에 지정된 수의 중단점들이 블로브를 검색했을 때 블로브가 삭제될 수 있다. 동일한 또는 또다른 구현들에서, 각종의 다른 기법들 및 알고리즘들이 블로브 캐쉬 서비스와 연관된 스토리지를 관리하는 데 사용될 수 있다.



- [0116] 적어도 어떤 구현들에서, 블로브 캐쉬 서비스가 특정의 블로브의 적어도 일부분을 저장했을 때 블로브 캐쉬 서비스는 블로브 캐쉬 서비스와 연관된 장소로 예시적인 블로브 탐색 서비스를 업데이트시킬 수 있고, 블로브의 저장된 일부분을 다른 중단점들에 제공할 수 있다. 이러한 장소를 사용하여, 블로브 탐색 서비스는 블로브 캐쉬 서비스와 연관된 위치 지정자를 중단점에 제공할 수 있고, 이러한 중단점은 그에 따라 블로브 캐쉬 서비스로부터 블로브를 검색할 수 있다. 어떤 구현들에서, 중단점들은 또한 블로브 탐색 서비스를 사용하지 않고 블로브 캐쉬 서비스로부터 블로브들을 검색할 수 있다.
- [0117] 동작(725)의 예시적인 구현에서, 특정의 블로브 또는 블로브들이 캐싱된 후 어떤 시점에서, 블로브 캐쉬 서비스는 캐싱된 블로브의 전부 또는 그 일부분에 대한 요청을 하나 이상의 중단점들로부터 수신할 수 있고, 동작(730)의 예시적인 구현에서, 요청된 블로브(또는 블로브의 일부분)를 요청자에게 제공할 수 있다. 캐싱된 블로브들과 연관된 요청들 및 응답들이 HTTP 전송, FTP 전송, BitTorrent-방식 전송, 기타 등등의 본 명세서의 다른 곳에서 설명되는 메카니즘들을 비롯한 다양한 통신 메카니즘들, 기타 등등을 사용하여 다양한 방식으로 구현될 수 있다.
- [0118] 이제 도 8을 참조하면, 클라우드 스토리지(cloud storage)에 블로브를 저장하고 클라우드 스토리지로부터 블로브를 제공할 때 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름(800)이 도시되어 있다. 도 8에 대한 이하의 설명은 다른 도면들을 참조하여 행해질 수 있다. 그렇지만, 도 8을 참조하여 기술되는 동작 흐름이 이들 다른 도면들을 참조하여 기술되는 구성요소들과 함께 사용되는 것으로 제한되는 것으로 의도하기 위한 것이 아님을 잘 알 것이다. 그에 부가하여, 도 8의 예시적인 동작 흐름이 특정의 실행 순서를 나타내고 있지만, 하나 이상의 대안의 실시예들에서, 동작들이 다른 순서로 되어 있을 수 있다. 게다가, 예시적인 동작 흐름이 다수의 단계들을 포함하고 있지만, 어떤 구현들에서 이들 동작 중 적어도 몇몇이 결합, 즉 동시에 실행될 수 있고 또 동일한 또는 다른 구현들에서 몇몇 단계들이 실행되지 않을 수 있다는 것을 잘 알 것이다.
- [0119] 동작(810)의 예시적인 구현에서, 중단점은 특정의 블로브의 일부 또는 전부를 "클라우드에", 즉 클라우드 스토리지에 저장하라는 요청을 받거나 수신할 수 있다. 이러한 요청을 받는 중단점 또는 모듈 또는 서비스는 어떤 경우들에 어쩌면 서버 중단점(110)과 같은 서버 중단점에 의해 구현될 수 있고, 동작 흐름(800)과 연관된 동작들 중 적어도 몇몇 동작에서, 예시적인 클라우드 스토리지 모듈(135)에 의해 구현되거나 제공될 수 있으며, 이들에 대해서는 도 1을 참조하여 앞서 설명하였다. 동작 흐름(800)의 동작들 중 몇몇 동작이 본 명세서에서 예시적인 서버 중단점 또는 예시적인 클라우드 스토리지 서비스와 연관되어 있는 것으로 설명될 수 있지만, 이들 동작이 서버 중단점 또는 클라우드 스토리지 서비스에 의해 꼭 구현되거나 실행되어야 하는 것은 아니며 실제로는 각종의 중단점 또는 그 중단점들에 포함된 모듈들 또는 서비스들 중 하나 이상에 의해 구현되거나 실행될 수 있다는 것을 잘 알 것이다.
- [0120] 동작 흐름(800)에서의 동작들 중 다수가 블로브 캐쉬 서비스 동작 흐름(700)에서 설명된 동작들과 유사한 것처럼 보일 수 있지만, 클라우드 스토리지 서비스는 블로브 캐쉬 서비스와 다르게 사용될 수 있다. 예를 들어, 블로브 캐쉬 서비스에 의해 제공되는 일시적 저장과는 달리, 클라우드 스토리지 서비스는, 예를 들어, 사용자에 의해 특별히 지시되지 않는 한 저장된 블로브들을 삭제하지 않음으로써, 저장된 데이터의 사본을 어쩌면 별도의 디스크 드라이브들 또는 다른 저장 메카니즘들에 여러개 유지함으로써, 기타 등등에 의해 블로브들(어쩌면 기타 데이터)에 대한 비일시적, 즉 "신뢰성있는" 저장을 제공할 수 있다.
- [0121] 동작(815)의 적어도 어떤 구현들에서, 클라우드 스토리지 서비스는 블로브 데이터(어떤 경우들에 동작(810)에서 수신되었을 수 있는 요청과 연관된 블로브의 일부분 또는 그 전체를 포함함)를 검색할 수 있다, 즉 제공받을 수 있다. 클라우드 스토리지 서비스는 다양한 방식으로(어떤 경우들에, 블로브 탐색 서비스, 블로브 다운로드 모듈, 또는, 예를 들어, 도 4를 참조하여 앞서 설명한 동작들 중 몇몇 동작들을 사용하는 것을 포함함) 블로브를 검색할 수 있다. 다른 구현들에서, 클라우드 스토리지 서비스는 각종의 다른 통신 또는 파일 전송 메카니즘들 중 임의의 것을 사용하여 블로브를 검색할 수 있다, 즉 제공받을 수 있다. 적어도 몇몇 구현들에서, 통신 시스템, 메타데이터, 블로브들, 기타 등등에 대한 지식(어쩌면 도 1을 참조하여 앞서 설명한 지식 저장소 모듈(145)과 같은 예시적인 지식 저장소와 연관된 지식을 포함함)이, 어쩌면 블로브 캐쉬 서비스와 관련하여 이미 설명한 방식들과 유사한 방식으로 또는 다른 방식들로, 요청된 블로브를 보다 효율적으로 획득하는 데 사용될 수 있다.
- [0122] 동작(820)의 예시적인 구현에서, 검색된 또는 제공된 블로브는 다양한 방식으로 클라우드 스토리지 서비스에 의해 저장될 수 있다. 예를 들어, 앞서 소개한 바와 같이, 클라우드 스토리지 서비스는 다수의 중복 저장소들을 사용하여 블로브들을 비롯한 데이터를 저장할 수 있다. 그에 부가하여, 클라우드 스토리지 서비스는 어떤

경우들에 블로그를 저장했을 때 정보를 업데이트시키거나 예시적인 블로그 탐색 서비스에 제공할 수 있고, 그에 따라 블로그 탐색 서비스는 어떤 구현들에서 클라우드 스토리지 서비스를 참조하거나 식별해주는 위치 지정자들을 다른 종단점들에게 제공할 수 있으며 그 후에 다른 종단점들은 어떤 경우들에 클라우드 스토리지 서비스로부터 블로그들을 검색할 수 있다.

[0123] 어떤 시점에서, 클라우드 스토리지 서비스는, 동작(825)의 적어도 어떤 구현들에서, 저장된 블로그 또는 저장된 블로그의 일부분을 제공하라는 요청을 수신할 수 있다. 이러한 요청에 응답하여, 클라우드 스토리지 서비스는 이어서, 동작(830)의 예시적인 구현에서, 어쩌면 본 명세서의 다른 곳에서 설명한 메카니즘들을 비롯한 다양한 통신 또는 전송 메카니즘들을 사용하여, 요청된 블로그 데이터를 제공할 수 있다.

[0124] 이제부터 도 9를 참조하면, 전체 메타데이터 또는 예시적인 서버 종단점에 의해 제공되는 기능의 적어도 일부를 사용하지 않고 블로그를 검색할 때 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름(900)이 도시되어 있다. 도 9에 대한 이하의 설명은 다른 도면들을 참조하여 행해질 수 있다. 그렇지만, 도 9를 참조하여 기술되는 동작 흐름이 이들 다른 도면들을 참조하여 기술되는 구성요소들과 함께 사용되는 것으로 제한되는 것으로 의도하기 위한 것이 아님을 잘 알 것이다. 그에 부가하여, 도 9의 예시적인 동작 흐름이 특정의 실행 순서를 나타내고 있지만, 하나 이상의 대안의 실시예들에서, 동작들이 다른 순서로 되어 있을 수 있다. 게다가, 예시적인 동작 흐름이 다수의 단계들을 포함하고 있지만, 어떤 구현들에서 이들 동작 중 적어도 몇몇이 결합, 즉 동시에 실행될 수 있고 또 동일한 또는 다른 구현들에서 몇몇 단계들이 실행되지 않을 수 있다는 것을 잘 알 것이다.

[0125] 본 명세서에서의 설명의 대부분이 메타데이터를 사용하여 블로그 데이터를 검색하는 것에 대해 기술하고 있지만, 적어도 어떤 구현들에서 블로그들을 검색할 때 메타데이터 전체를 사용하는 것이 항상 필요한 것은 아닐 수 있다. 그 대신에, 어떤 경우들에 다양한 종단점 기능들(예를 들어, 블로그 탐색 서비스 및 어쩌면 블로그의 전부 또는 일부를 제공할 수 있는 다양한 종단점들을 포함함)을 이용하기 위해 최소 분량의 데이터만(예를 들어, 블로그 식별자만)이 필요할 수 있다. 어떤 환경들에서, 이러한 기능을 사용함으로써 종단점은 보다 신속하게, 보다 저렴하게, 또는 어떤 점에서 보다 효율적으로 블로그를 검색할 수 있게 된다.

[0126] 동작(910)의 예시적인 구현에서, 종단점은 어떤 수단들을 통해 블로그 식별자를 획득할 수 있다. 앞서 설명한 구현들 중 몇몇 구현들에서, 블로그 식별자들은 서로 다른 종단점들 간에 동기화될 수 있는 메타데이터에 포함되어 있다. 이러한 동기화된 메타데이터는 종단점이 블로그 식별자를 획득할 수 있는 적어도 하나의 메카니즘을 제공한다. 그렇지만, 메타데이터를 동기화시키지 않는 구현들, 기타 등등을 비롯한 다양한 동일한 또는 다른 구현들에서, 블로그 식별자는 다양한 다른 방식으로 제공될 수 있다. 예를 들어, 사용자는 블로그 식별자를 포함하는 이메일 또는 인스턴트 메시지를 다른 사용자에게 전송할 수 있으며, 이 블로그 식별자는 차례로 어쩌면 오디오 또는 비디오 클립, 워드 프로세싱 문서 또는 스프레드시트, 어떤 다른 파일, 기타 등등의 블로그를 식별해준다.

[0127] 동작(915)의 적어도 어떤 구현들에서, 블로그 탐색 서비스가 블로그를 검색하는 데 사용되는지 여부가 판정될 수 있다. 예를 들어, 어떤 종단점들은 블로그 탐색 서비스에 액세스하지 않거나 심지어 그에 관해 모르고 있을 수도 있다. 이러한 경우에 또 다른 경우들에(블로그 탐색 서비스가 이용가능하지만 사용되지 않을 때를 포함함), 블로그 탐색 서비스를 사용하지 않기로 결정할 수 있고, 동작 흐름(900)은 동작(930)으로 진행할 수 있다. 블로그가 블로그 탐색 서비스를 사용하여 검색되는 경우, 동작 흐름(900)은 동작(920)으로 진행할 수 있다.

[0128] 블로그 탐색 서비스가 사용되거나 사용되지 않을지도 모르는 환경의 일례로서, 웹 브라우저의 예를 생각해보자. 많은 웹 브라우저들이 HTTP 등의 프로토콜들을 사용하여 리소스들의 다운로드 또는 검색을 가능하게 해주는 실행가능 코드를 포함하고 있다. 이러한 웹 브라우저에서, 단일 서버(또는 적어도 하나의 URL)에 대해 HTTP 요청이 행해질 수 있고, 단일 서버로부터 응답이 수신될 수 있다. 그렇지만, 어떤 경우들에(예를 들어, 클지도 모르는 블로그를 검색할 때), BitTorrent 또는 BitTorrent와 유사한 전송 구현들을 비롯한 메카니즘들을 비롯한 하나 이상의 다른 메카니즘들을 사용하여 블로그를 다운로드하는 것이 보다 효율적이고 보다 빠를 수 있으며, 이 경우 특정의 블로그의 서로 다른 부분들이 다수의 서로 다른 종단점들로부터 검색되거나 그들에 의해 제공될 수 있다.

[0129] 그렇지만, 파일들 또는 블로그들을 다운로드하는 이러한 메카니즘들이 바람직할 수 있는 반면, 웹 브라우저가 이러한 또는 다른 유형들의 다운로드를 가능하게 해주는 기능을 본래 포함하고 있지 않을 수 있다. 어떤 경우들에, 예를 들어, 웹 브라우저 또는 애플리케이션이 블로그 탐색 서비스(또한 이하에서 더 상세히 설명하는 바

와 같이, 어쩌면 블로그 다운로드 모듈)의 사용을 지원할 수 있게 해줌으로써 이러한 능력을 추가하는 기능을 웹 브라우저 또는 어떤 다른 애플리케이션에 추가하는 것이 가능할 수 있다. 예를 들어, 웹 브라우저는 사용자들에 의해 추가되는 "플러그인" 또는 기타 코드를 지원할 수 있다. 한가지 이러한 플러그인은 블로그 탐색 서비스에 대한 지원을 구현하며, 설치될 때, 웹 브라우저가 블로그 탐색 서비스를 사용할 수 있게 해줄 수 있다 (따라서 동작(930)으로 진행하지 않고 동작(920)으로 진행함).

[0130] 동작(920)의 예시적인 구현에서, 블로그를 검색하는 종단점은, 어쩌면 원하는 블로그를 식별해주는 블로그 식별자와 같은 어떤 데이터를 블로그 탐색 서비스에 제공함으로써, 원하는 블로그와 연관된 하나 이상의 위치 지정자들이 있는지 블로그 탐색 서비스에 질의 또는 질문을 할 수 있다. 이러한 동작은 적어도 어떤 구현들에서, 예를 들어, 도 4를 참조하여 앞서 설명한 동작(415)이 구현되었던 방식과 유사하거나 동일한 방식으로 구현될 수 있다.

[0131] 그 다음에, 동작(925)의 예시적인 구현에서, 획득된 위치 지정자들 중 하나 이상 및 어떤 경우들에 블로그 다운로드 모듈이 블로그를 실제로 검색하는 데 사용될 수 있다. 이러한 동작은 적어도 어떤 구현들에서, 예를 들어, 도 4를 참조하여 앞서 설명한 동작(425)이 구현되었던 방식과 유사하거나 동일한 방식으로 구현될 수 있다.

[0132] 어떤 구현들에서, 종단점은 블로그가 다운로드될 수 없거나 블로그의 검색이 실패하는 경우 부가의 조치들을 취할 수 있다. 예를 들어, (네트워크 조건들로 인해, 위치 지정자들에 의해 식별된 종단점들이 이용가능하지 않기 때문에, 기타 등등의 이유로) 블로그 탐색 서비스가 위치 지정자들을 제공하지 않거나 블로그를 실제로 검색하는 데 사용될 수 없는 위치 지정자들만 제공하는 경우, 블로그를 검색하는 종단점은, 예를 들어, 블로그 캐쉬 서비스에 블로그를 저장하라고 요청할 수 있고 그 다음에 나중에 블로그 캐쉬 서비스로부터 블로그를 검색하려고 시도할 수 있다. 예를 들어, 이러한 메카니즘이 도 4를 참조하여 앞서 설명되었으며, 도 9에는 도시되어 있지 않다.

[0133] 종단점이 성공적으로 블로그를 검색하는 경우, 그 종단점은 다른 종단점들이 동일한 블로그를 요청할 때 어떤 경우들에 블로그 탐색 서비스가 그 종단점을 블로그의 소스로서 식별할 수 있도록 부가의 정보로 블로그 탐색 서비스를 업데이트시킬 수 있다. 이러한 구현에서, 예를 들어, 사무실에 있는 한 사용자가, 예를 들어, 큰 파일 또는 블로그를 다운로드 또는 검색하는 것이 가능할 수 있고, 또 예를 들어 동일한 사무실에 있는 다른 사용자들이 하나 이상의 다른 네트워크들 상의 다른 곳에 위치할 수 있는 어떤 다른 종단점으로부터보다는 제1 사용자로부터 블로그를 검색하게 하는 것이 가능할 수 있다. 이러한 예에서, 제1 사용자가 블로그를 검색한 후에, 제1 사용자와 연관된 종단점이 종단점을 식별해주는 정보를 블로그 탐색 서비스에 제공했을 수 있다. 제2 사용자 또는 나중의 사용자가 블로그 탐색 서비스에 동일한 블로그를 요청할 때, 블로그 탐색 서비스는 제1 사용자의 종단점을 식별해주는 위치 지정자를 제공할 수 있다. 어떤 구현들에서, 블로그 탐색 서비스는 제1 사용자의 종단점이 어쩌면 최초의 종단점보다 지리적으로 더 가깝고, 최초의 종단점보다 고속인 네트워크 연결을 사용하여 제2 종단점에 연결되어 있으며, 기타 등등이라는 지식(지식 저장소에 의해 유지되고 제공되는 지식을 포함함)을 사용함으로써 이러한 위치 지정자를 제공할 수 있다.

[0134] 블로그 탐색 서비스가 사용되지 않는 경우, 동작 흐름(900)은 동작(930)으로 진행할 수 있다. 동작(930)의 예시적인 구현에서, 블로그 탐색 서비스 또는 블로그 다운로드 모듈을 직접 사용하지 않는(그렇지만 어떤 구현들에서 간접적으로는 사용할 수 있는) 하나 이상의 파일 전송 또는 블로그 검색 메카니즘들을 사용하여 블로그가 검색될 수 있다. 이러한 메카니즘들은 HTTP, FTP, 파일 공유 또는 전송 프로토콜, 기타 등등을 포함할 수 있다. 예를 들어, 블로그 식별자가 "http://www.els.live.com/ABC123"와 같은 경우, 종단점은 단순히 블로그 식별자에 의해 구현된 URL에 대해 HTTP 요청을 개시할 수 있다. 어떤 구현들에서, 블로그에 대한 요청에 응답하는 서버 종단점 또는 HTTP 서버는 그 다음에, 식별자("ABC123" 등)를 추출하고 이 식별자를 사용하여 요청된 블로그를 찾아낸 다음에 마지막으로 HTTP 응답으로 블로그를 반환함으로써, 이 특정 형태의 URL을 사용하도록 구성될 수 있다. (어떤 경우들에, 서버 종단점 자체는 도 4를 참조하여 앞서 설명한, 예를 들어, 블로그 검색 동작들과 동일하거나 유사한 동작들을 사용함으로써 블로그를 찾아낼 수 있다. 동일한 또는 다른 구현들에서, 서버 종단점은 블로그 캐쉬 서비스 또는 클라우드 스토리지 서비스 등의 특징의 저장 장소에 존재하는 블로그를 반환하기만 할 수 있다.)

[0135] 예시적인 컴퓨팅 환경

[0136] 이제 도 10을 참조하면, 동 도면 및 관련 설명은 본 명세서에 설명된 다양한 기술들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 간략한 전반적인 설명을 제공하기 위한 것이다. 꼭 그럴 필요는 없지만, 이들 기술은 본



명세서에 일반적으로 적어도 부분적으로 컨트롤러, 프로세서, 퍼스널 컴퓨터, 또는 도 10에 도시된 컴퓨팅 장치(1000) 등의 컴퓨팅 장치에 의해 실행되는 프로그램 모듈 등의 컴퓨터 실행가능 명령어들과 관련하여 설명되어 있다.

- [0137] 일반적으로, 프로그램 모듈은 특정의 태스크를 수행하거나, 특정의 정보를 디스플레이하거나, 특정의 추상 데이터 유형을 구현하는 루틴, 프로그램, 객체, 컴포넌트, 사용자 인터페이스, 데이터 구조, 기타 등등을 포함한다. 프로그램 모듈들에 의해 수행되는 동작들이 하나 이상의 블록도 및 동작 플로우차트를 사용하여 앞서 설명되었다.
- [0138] 당업자라면 이들 설명, 블록도 및 동작 흐름을, 하나 이상의 형태의 컴퓨터 판독가능 매체에 구현될 수 있는 컴퓨터 실행가능 명령어의 형태로 구현할 수 있다. 본 명세서에서 사용되는 바와 같이, 컴퓨터 판독가능 매체는 컴퓨터에 의해 액세스되고 이해될 수 있는 형태로 인코딩된 정보를 저장 또는 구현할 수 있는 매체라면 어느 것이라도 될 수 있다. 통상적인 형태의 컴퓨터 판독가능 매체로는 휘발성 및 비휘발성 메모리 둘다, 이동식 및/또는 비이동식 매체를 포함한 데이터 저장 장치, 그리고 통신 매체가 있지만, 이들로 제한되지 않는다.
- [0139] 통신 매체는 컴퓨터 판독가능 정보를 반송파 또는 기타 전송 메카니즘 등의 피변조 데이터 신호에 구현하고 모든 정보 전달 매체를 포함한다. "피변조 데이터 신호"라는 용어는 신호의 특성들 중 하나 이상이 정보를 그 신호에 인코딩하는 방식으로 설정 또는 변경된 신호를 의미한다. 제한이 아닌 예로서, 통신 매체는 유선 네트워크 또는 직접 배선 접속(direct-wired connection) 등의 유선 매체와, 음향, RF, 적외선 및 기타 무선 매체 등의 무선 매체를 포함한다.
- [0140] 도 10에 도시된 컴퓨팅 장치(1000)는, 가장 기본적인 구성에서, 적어도 하나의 처리 장치(1002) 및 메모리(1004)를 포함한다. 어떤 구현들에서, 컴퓨팅 장치(1000)는, 예를 들어, 도 1을 참조하여 앞서 설명한 종단점들(서버 종단점(110), 장치 종단점 A(160), 장치 종단점 B(170), 기타 등등) 중 하나의 적어도 일부를 구현할 수 있다. 어떤 구현들에서, 처리 장치(1002)는, 예를 들어, 데스크톱 및 랩톱 컴퓨터를 비롯한 각종의 컴퓨터에 존재하는 것과 같은 범용 CPU(central processing unit)일 수 있다. 컴퓨팅 장치의 정확한 구성 및 유형에 따라, 메모리(1004)는 휘발성(RAM 등), 비휘발성(ROM, 플래쉬 메모리, 기타 등등), 또는 이 둘의 어떤 조합일 수 있다. 이러한 가장 기본적인 구성이 도 10에서 점선(1006)으로 도시되어 있다. 그에 부가하여, 컴퓨팅 장치(1000)는 또한 부가의 특징들 및 기능들을 가질 수 있다. 예를 들어, 컴퓨팅 장치(1000)는 또한 자기 또는 광학 디스크 또는 테이프(이들로 제한되지 않음)를 비롯한 부가의 저장 장치(이동식 및/또는 비이동식)를 포함할 수 있다. 이러한 부가의 저장 장치가 도 10에서 이동식 저장 장치(1008) 및 비이동식 저장 장치(1010)로 도시되어 있다.
- [0141] 컴퓨팅 장치(1000)는 또한 컴퓨팅 장치(1000)가 다른 장치들 및 서비스들과 통신을 할 수 있게 해주는 하나 이상의 통신 연결(들)(1012)을 포함할 수 있다. 예를 들어, 컴퓨팅 장치는, 예를 들어, 도 1을 참조하여 앞서 설명한 것과 같은 종단점들에 대한 연결 또는 이들 종단점 간의 연결을 비롯한, 각종의 통신 수단 또는 컴퓨팅 장치들에 대한 하나 이상의 연결을 가질 수 있다. 컴퓨팅 장치(1000)는 또한 카메라 또는 스캐너 등의 이미지 입력 장치, 키보드, 마우스, 펜, 마이크 어레이를 비롯한 음성 입력 장치, 터치 입력 장치, 기타 등등의 하나 이상의 입력 장치(들)(1014)를 가질 수 있다. 디스플레이, 스피커, 프린터, 기타 등등의 하나 이상의 출력 장치(들)(1016)도 또한 컴퓨팅 장치(1000)에 포함되어 있을 수 있다.
- [0142] 당업자라면 본 명세서에 설명된 기술들이 도 10에 도시된 컴퓨팅 장치(1000)와 다른 컴퓨팅 장치들에 의해 실시될 수 있다는 것을 잘 알 것이다. 예를 들어, 본 명세서에 설명된 기술들이 마찬가지로 이동 전화 및 PDA를 비롯한 핸드헬드 장치, 멀티프로세서 시스템, 마이크로프로세서-기반 또는 프로그램가능 가전 제품, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 기타 등등에서 실시될 수 있지만, 이들로 제한되지 않는다. 이들 컴퓨팅 장치 각각이 도 10의 시스템에 의해 어떤 수준의 상세로 설명될 수 있거나 다른 방식으로 설명될 수 있다.
- [0143] 본 명세서에 설명된 기술들은 또한 동작들이 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 수행되는 분산 컴퓨팅 환경에서도 실시될 수 있다. 분산 컴퓨팅 환경에서는, 프로그램 모듈들이 로컬 및 원격 장치 둘다에 위치할 수 있다.
- [0144] 소프트웨어로 구현되는 것으로 본 명세서에 설명되어 있지만, 또한 본 명세서에 설명된 기술들이 다른 대안으로서 전체적으로 또는 부분적으로 하드웨어, 펌웨어, 또는 소프트웨어, 하드웨어 및/또는 펌웨어의 다양한 조합으로서 구현될 수 있다는 것도 잘 알 것이다.
- [0145] 방법들 및 시스템들의 어떤 특정의 구현들이 첨부 도면들에 도시되고 이상에서 설명되어 있지만, 도시되고 설명

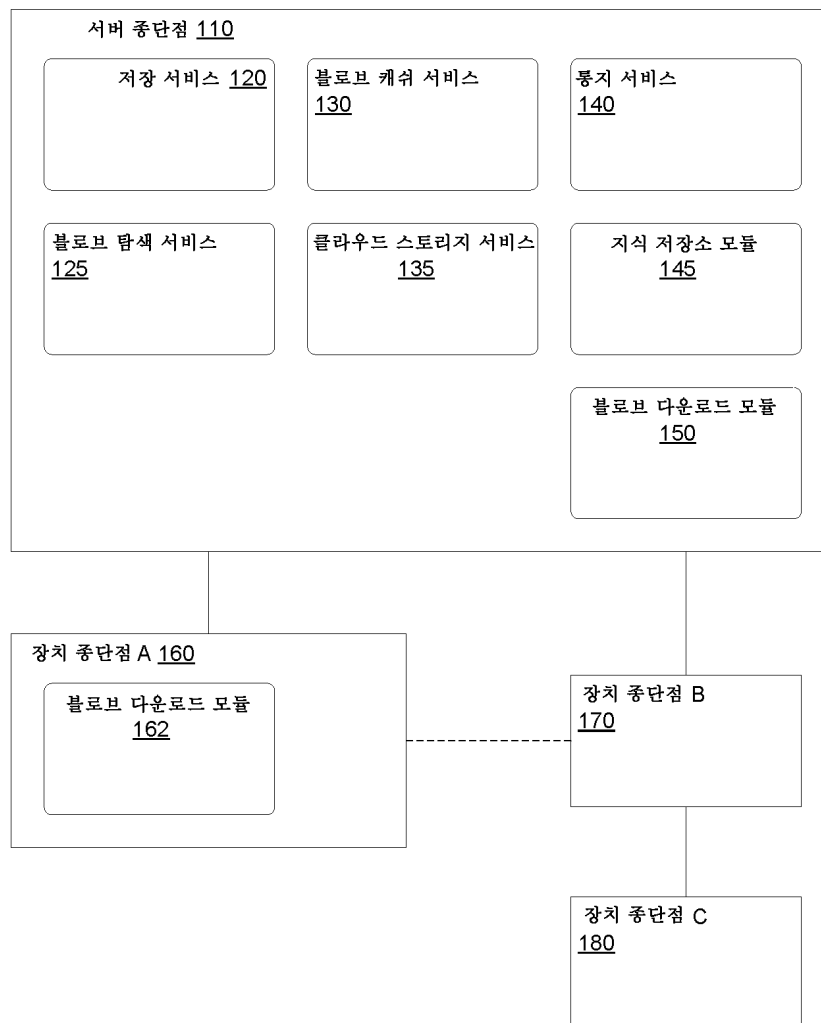
된 방법들 및 시스템들이 기술된 특정의 구현들로 제한되지 않으며 이하의 청구항들에 기술되고 정의된 발명의 정신을 벗어나지 않고 여러가지로 재배열, 수정 및 대체될 수 있다는 것을 잘 알 것이다.

### 도면의 간단한 설명

- [0006] 도 1은 데이터가 중앙집중형 토폴로지 및 피어-투-피어 토폴로지 둘다에서 동기화되어 공유될 수 있는 예시적인 시스템을 나타낸 도면.
- [0007] 도 2는 동기화된 데이터를 변경하고 이러한 변경을 다른 종단점들이 이용할 수 있게 해줄 때, 장치 종단점을 비롯한 종단점에 의해 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름을 나타낸 도면.
- [0008] 도 3는 다른 종단점으로부터의 데이터를 동기화시킬 때, 장치 종단점을 비롯한 종단점에 의해 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름을 나타낸 도면.
- [0009] 도 4는 블로브(blob)를 검색 또는 다운로드하기 위해 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름을 나타낸 도면.
- [0010] 도 5는 메타데이터를 유지 및 제공할 때 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름을 나타낸 도면.
- [0011] 도 6는 메타데이터에 의해 참조되는 블로브와 연관된 위치 및 위치 지정자(locator)를 저장 및 제공할 때 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름을 나타낸 도면.
- [0012] 도 7는 캐쉬에 블로브를 저장하고 캐쉬로부터 블로브를 제공할 때 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름을 나타낸 도면.
- [0013] 도 8는 클라우드 스토리지(cloud storage)에 블로브를 저장하고 클라우드 스토리지로부터 블로브를 제공할 때 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름을 나타낸 도면.
- [0014] 도 9는 전체 메타데이터 또는 예시적인 서버 종단점에 의해 제공되는 기능의 적어도 일부를 사용하지 않고 블로브를 검색할 때 수행될 수 있는 다양한 동작들을 포함하는 예시적인 일반화된 동작 흐름을 나타낸 도면.
- [0015] 도 10은 본 명세서에 기술된 다양한 기술들이 구현될 수 있는 예시적인 컴퓨팅 환경을 나타낸 도면.

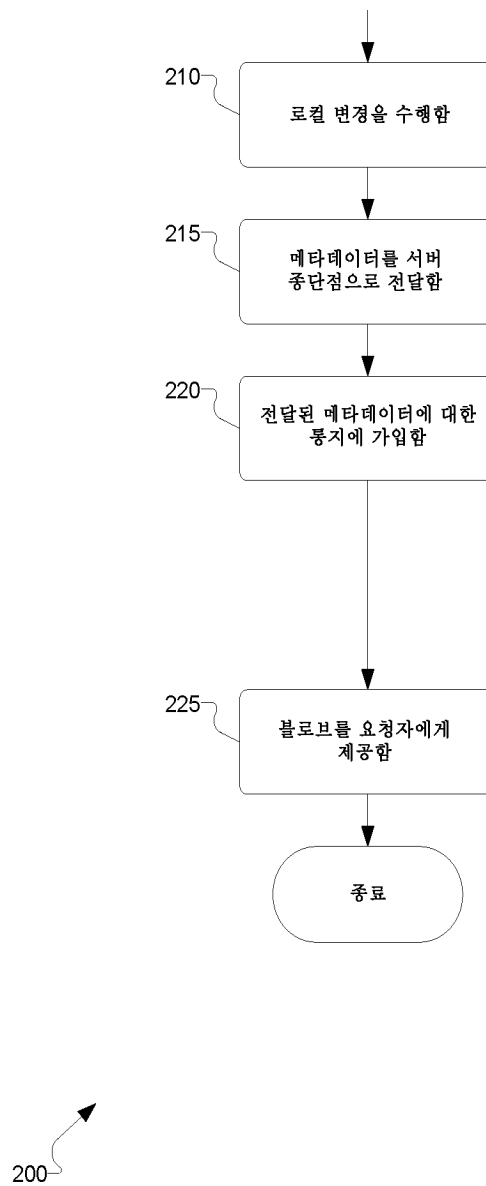
도면

도면1

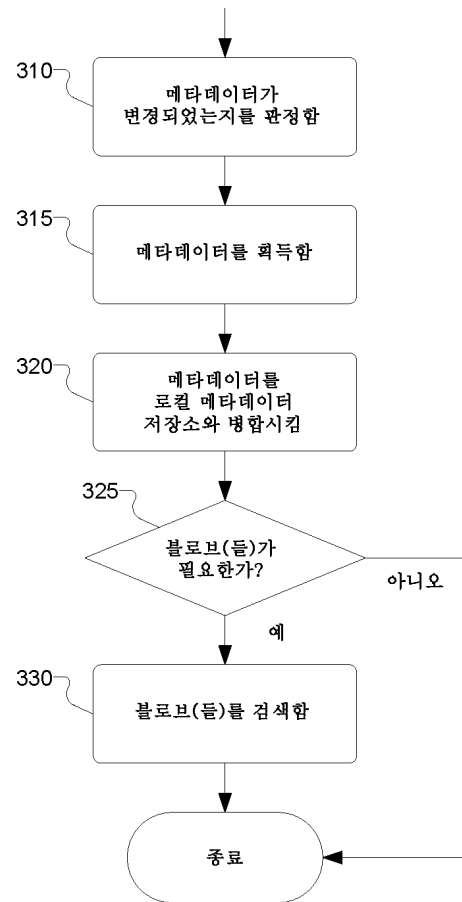


100

도면2

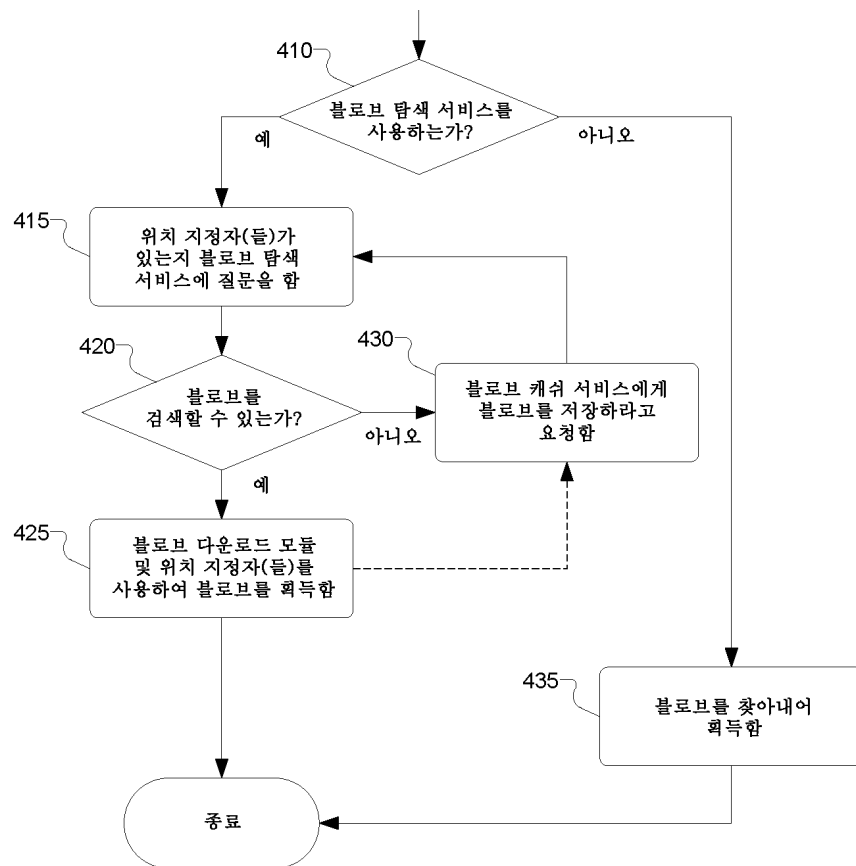


도면3



300 ↗

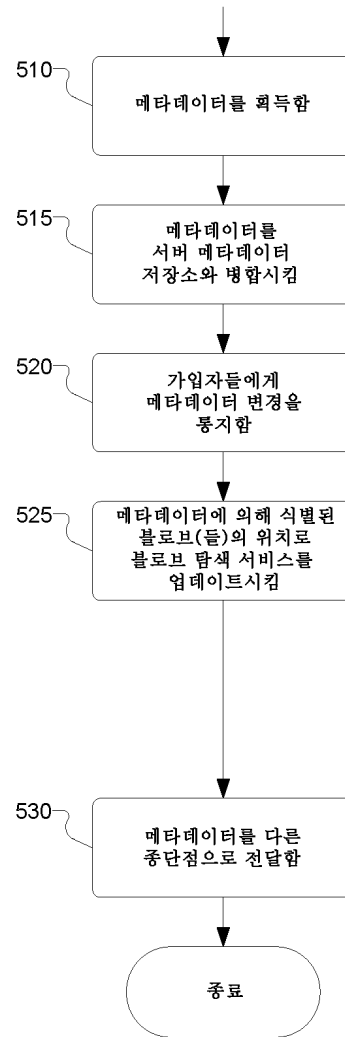
도면4



400

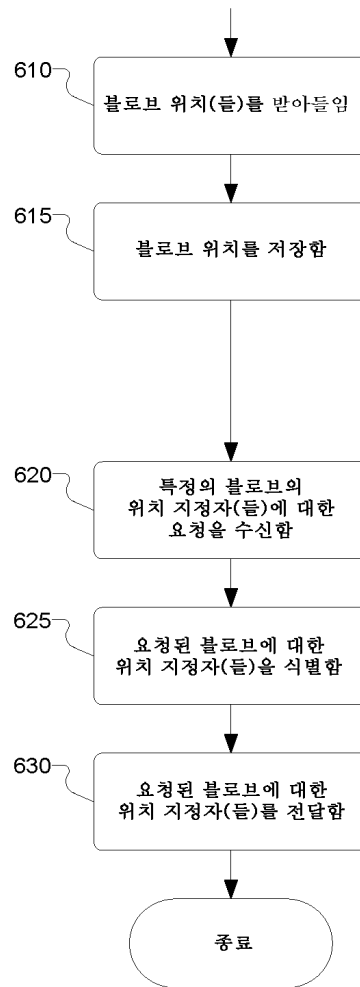


도면5



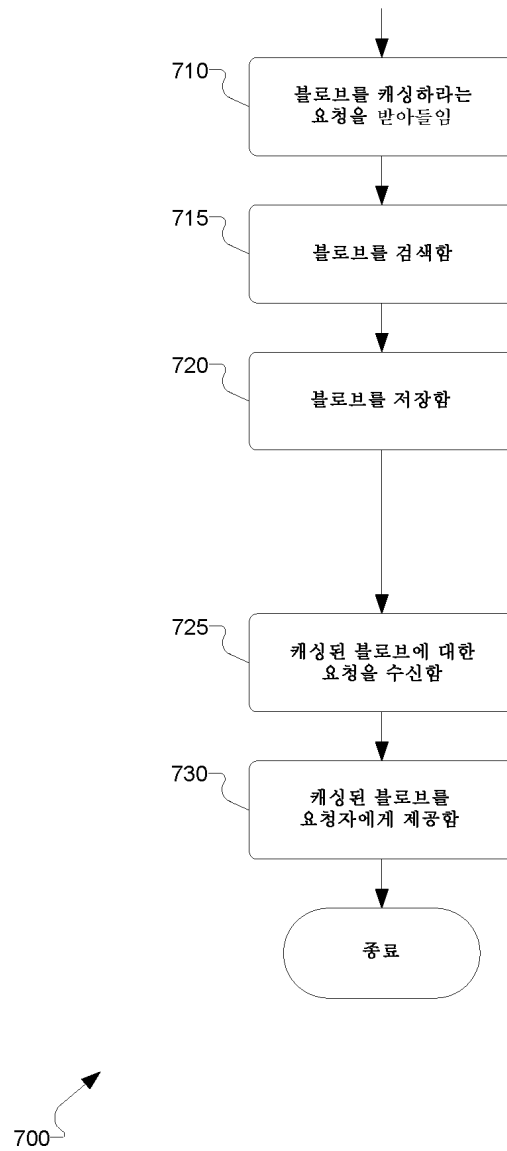
500 ↗

도면6

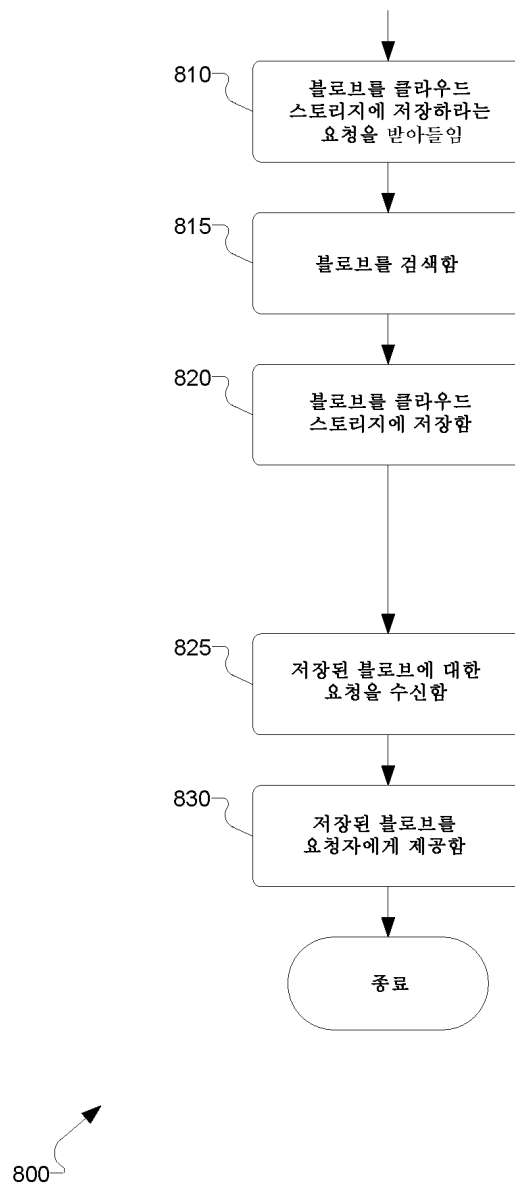


600 ↗

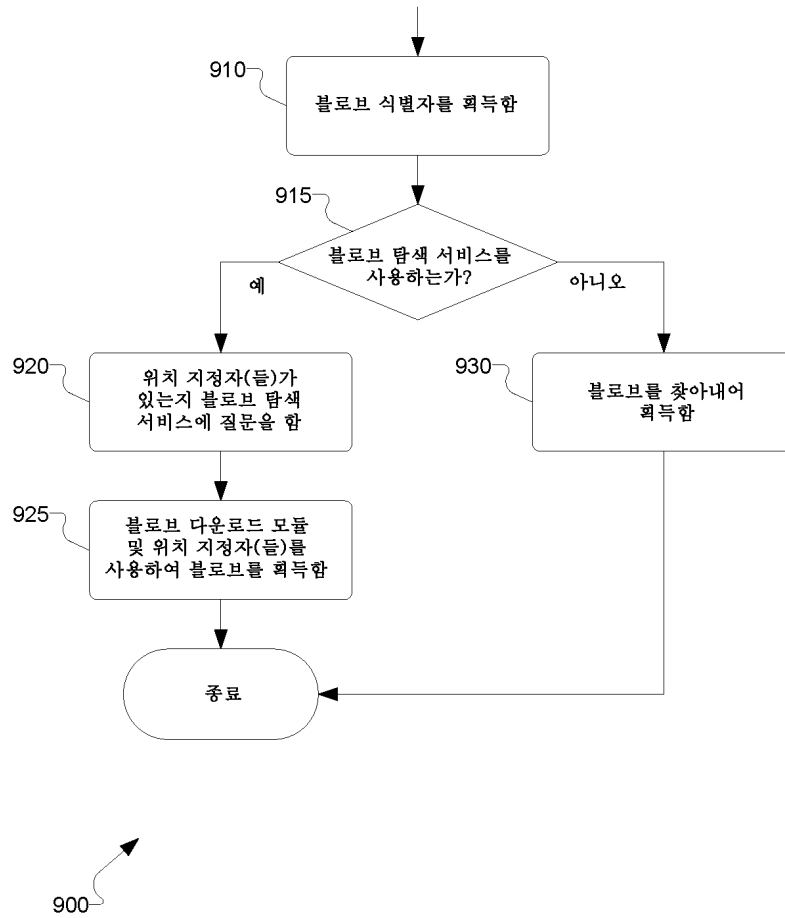
도면7



도면8



도면9



도면10

