



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년10월08일

(11) 등록번호 10-1559046

(24) 등록일자 2015년10월02일

(51) 국제특허분류(Int. Cl.)

G06F 15/16 (2006.01)

(21) 출원번호 10-2011-7011731

(22) 출원일자(국제) 2009년10월21일

심사청구일자 2014년10월21일

(85) 번역출제출일자 2011년05월23일

(65) 공개번호 10-2011-0081867

(43) 공개일자 2011년07월14일

(86) 국제출원번호 PCT/US2009/061530

(87) 국제공개번호 WO 2010/048324

국제공개일자 2010년04월29일

(30) 우선권주장

61/107,312 2008년10월21일 미국(US)

(56) 선행기술조사문헌

Synchronization operations for disconnected IMAP4 clients;  
draft-melnikov-imap-disc-02.txt", IETF  
standard-working-draft, Internet engineering  
task force, IETF, CH, no. 2, 1 Oct. 2002,  
page 13 line 20

US7860825 B1

JP2002-014860 A

WO2007/021454 A1

(73) 특허권자

구글 인코퍼레이티드

미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이  
1600 (우:94043)

(72) 발명자

팔래이, 앤드류, 제이.

미국 94040 캘리포니아 마운틴 뷰 우드스톡 레인  
3445

화이트, 아아론

미국 94122 캘리포니아 샌프란시스코 28 애브뉴  
1463

(74) 대리인

특허법인 남앤드남

전체 청구항 수 : 총 14 항

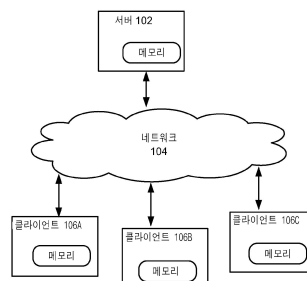
심사관 : 윤혜숙

(54) 발명의 명칭 **항상 준비된 클라이언트/서버 데이터 동기화 방법**

### (57) 요약

데이터 동기화를 위한 시스템들과 방법들이 설명된다. 몇몇 실시예들에서, 방법은 클라이언트가 서버와의 정보 동기화를 위하여 창안된다. 이 클라이언트는 서버와 클라이언트 사이에 정보를 선택적으로 동기화하기 위하여 서버로부터 통신을 수신하고, 상기 정보는 클라이언트와 동기화되지 않은 데이터와 마지막 데이터 동기화에서 동기화되지 않은 변화 동작들을 포함하며, 상기 데이터는 높은 우선순위에서 낮은 우선순위의 순서로 배열되고 수신된다. 클라이언트는 수신된 순서에 따라 클라이언트의 메모리에 적어도 데이터의 일부분과 적어도 변화 동작들의 일부분을 저장하고, 클라이언트 디바이스의 사용자로 하여금 정보 동기의 종료 시점에 클라이언트에 저장된 통신에서의 변화 동작들의 일부분과 데이터의 일부분에 즉각적인 액세스가 가능하도록 한다.

**대표도** - 도1



## 명세서

### 청구범위

#### 청구항 1

프로세서에 의한 실행을 위해 프로그램들을 저장하는 메모리 및 상기 프로세서를 포함하는 클라이언트에서:

서버와 상기 클라이언트 사이의 정보를 선택적으로 동기화하기 위하여 상기 서버로부터 통신을 수신하는 단계;

상기 서버로부터 정보를 수신하는 단계 - 상기 정보는 제1 데이터 세트 및 제2 데이터 세트를 포함하고,

상기 제1 데이터 세트 및 상기 제2 데이터 세트가 상기 서버 상에서 유지되는 과거 동기화들의 기록에 근거하여 상기 서버에 의해 정의되며;

상기 제1 데이터 세트는 상기 제2 데이터 세트와 별개이고, 상기 클라이언트와 전혀 동기화되지 않았고 데이터의 우선순위에 따라서 선택된 데이터 항목들로 구성되며;

상기 제2 데이터 세트는 상기 제1 데이터 세트와 별개이고, 변화 동작들의 시간 순서에 따라서 선택된 변화 동작들로 구성되며, 변화 동작들은 이전에 동기화된 데이터 항목들로의 데이터 변화들이 진행되는 것이고, 상기 변화는 과거 동기화들의 기록의 이전 동기화 동작 이후로 발생함 -;

상기 데이터의 우선순위에 따라서 상기 제1 데이터 세트로부터 상기 데이터의 적어도 일부분을 상기 클라이언트의 메모리에 저장하는 단계;

상기 변화 동작들의 시간 순서에 따라서 상기 제2 데이터 세트로부터 상기 변화 동작들의 적어도 일부분을 수행하는 단계; 및

상기 클라이언트의 사용자로 하여금 하나 이상의 상기 변화 동작들이 수행된 데이터 및 데이터의 일부분에, 상기 클라이언트에서 액세스(access)를 갖게하는 단계

를 포함하는,

방법.

#### 청구항 2

제 1 항에 있어서,

상기 클라이언트와의 데이터 동기화를 개시하기 위하여 상기 클라이언트로부터 상기 서버에 요청을 전송하는 단계를 더 포함하는,

방법.

#### 청구항 3

제 2 항에 있어서,

상기 요청은 상기 서버에게 상기 클라이언트와 동기화될 다음 데이터 범위에 관하여 통지하기 위하여 가장 최근의 동기화에서 성공적으로 동기화된 데이터에 상응하는 데이터 범위에 관한 정보를 포함하는,

방법.

#### 청구항 4

제 1 항 또는 제 2 항에 있어서,

상기 데이터의 우선순위는 상기 서버에서의 우선순위 결정에 근거하고, 더 높은 우선순위 데이터는 더 낮은 우선순위 데이터 이전에 수신되며; 그리고

상기 변화 동작들의 시간 순서는 변화 동작들이 상기 서버 상에서 발생하였을 때에 근거하고, 오래된 변화들은 새로운 변화들 이전에 수신되는,

방법.

#### 청구항 5

제 1 항에 있어서,

상기 데이터와 상기 변화 동작들은 동시에 동기화되고, 데이터에 상응하는 중첩하는 변화 동작들의 동기화 및 그 역은 반복되지 않는,

방법.

#### 청구항 6

클라이언트 시스템으로서,

하나 이상의 프로세서들;

상기 하나 이상의 프로세서들에 커플링되는 메모리; 및

상기 메모리에 저장되고, 상기 하나 이상의 프로세서들에 의한 실행을 위해 구성되는 하나 이상의 프로그램들을 포함하고,

상기 하나 이상의 프로그램들은, 제 1 항 내지 제 3 항 중 어느 한 항의 방법들을 수행하기 위한 명령들을 포함하는,

클라이언트 시스템.

#### 청구항 7

컴퓨터 판독가능 저장 매체로서,

상기 컴퓨터 판독가능 저장 매체에 내장된 하나 이상의 컴퓨터 프로그램들은 명령들을 포함하며, 상기 명령들은 컴퓨터 시스템에 의해 실행될 때 상기 컴퓨터 시스템으로 하여금 제 1 항 내지 제 3 항 중 어느 한 항의 방법들을 수행하게 하는,

컴퓨터 판독가능 저장 매체.

#### 청구항 8

프로세서에 의한 실행을 위해 프로그램들을 저장하는 메모리 및 상기 프로세서를 포함하는 서버에서:

클라이언트와 전혀 동기화되지 않은 데이터 및 상기 클라이언트와 동기화되지 않은 변화 동작들을, 상기 서버 상에서 유지되는 과거 동기화들의 기록에 근거하여 식별하는 단계;

상기 서버와 상기 클라이언트 사이의 정보를 선택적으로 동기화하기 위하여 통신을 상기 클라이언트에게 전송하는 단계;

상기 클라이언트에게 정보를 전송하는 단계 - 상기 정보는 식별된 데이터의 우선순위에 따라서 상기 식별된 데이터로부터 선택된 제1 데이터 세트 및 변화 동작들의 시간 순서에 따라서 식별된 변화 동작들로부터 선택된 제2 데이터 세트를 포함하고, 상기 제1 데이터 세트는 상기 제2 데이터 세트와 별개이고 변화 동작들이 아니며, 상기 제2 데이터 세트는 상기 제1 데이터 세트와 별개이며, 변화 동작들은 이전에 동기화된 데이터 항목들로의 데이터 변화들이 진행되는 것이고, 상기 변화는 과거 동기화들의 기록의 이전 동기화 동작 이후로 발생함 -; 및

동기화 종료시 상기 클라이언트와 성공적으로 동기화된 통신에서의 변화 동작들 및 데이터의 일부분들을 포함하기 위해 상기 기록을 업데이트하는 단계

를 포함하는,

방법.

#### 청구항 9

제 8 항에 있어서,

상기 서버와의 데이터 동기화를 개시하기 위하여 상기 클라이언트로부터 요청을 수신하는 단계를 더 포함하는, 방법.

#### 청구항 10

제 9 항에 있어서,

상기 요청은 상기 서버에게 상기 클라이언트와 동기화될 다음 데이터 범위에 관하여 통지하기 위하여 가장 최근의 동기화에서 성공적으로 동기화된 데이터에 상응하는 데이터 범위에 관한 정보를 포함하는,

방법.

#### 청구항 11

제 8 항 또는 제 9 항에 있어서,

상기 데이터의 우선순위는 상기 서버에서의 우선순위 결정에 근거하고, 더 높은 우선순위 데이터는 더 낮은 우선순위 데이터 이전에 전송되며;

상기 변화 동작들의 시간 순서는 변화 동작들이 상기 서버 상에서 발생하였을 때에 근거하고, 오래된 변화들은 새로운 변화들 이전에 전송되는,

방법.

#### 청구항 12

제 8 항에 있어서,

상기 데이터와 상기 변화 동작들은 동시에 동기화되고, 데이터에 상응하는 중첩하는 변화 동작들의 동기화 및 그 역은 반복되지 않는,

방법.

#### 청구항 13

서버 시스템으로서,

하나 이상의 프로세서들;

상기 하나 이상의 프로세서들에 커플링되는 메모리; 및

상기 메모리에 저장되고, 상기 하나 이상의 프로세서들에 의한 실행을 위해 구성되는 하나 이상의 프로그램들을 포함하고,

상기 하나 이상의 프로그램들은, 제 8 항 내지 제 10 항 중 어느 한 항의 방법을 수행하기 위한 명령들을 포함하는,

서버 시스템.

#### 청구항 14

컴퓨터 판독가능 저장 매체로서,

상기 컴퓨터 판독가능 저장 매체에 내장된 하나 이상의 컴퓨터 프로그램들은 명령들을 포함하며, 상기 명령들은 컴퓨터 시스템에 의해 실행될 때 상기 컴퓨터 시스템으로 하여금 제 8 항 내지 제 10 항 중 어느 한 항의 방법을 수행하게 하는,

컴퓨터 판독가능 저장 매체.

#### 청구항 15



삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

## 발명의 설명

## 기술 분야

[0001]

개시된 실시예들은 일반적으로 클라이언트-서버 상호작용 분야에 관련된 것으로, 더욱 상세하게는, 클라이언트 디바이스와 서버 사이에 아이템들의 동기화에 관련된 것이다.

## 배경 기술

[0002]

많은 소프트웨어 애플리케이션들(예컨대, 이-메일)은 클라이언트 디바이스와 서버 시스템 모두에서 운영하거나, 서버 시스템으로 지지를 받는 클라이언트 디바이스 상에서 운영하도록 설계된다. 이 같은 애플리케이션들은 사용자가 클라이언트 디바이스 상 또는 서버 상에 운영하고 있는지 여부와 관계없이 사용자에게 데이터를 자유롭게 창조, 수신, 전송 및 수정을 가능하게 한다. 클라이언트와 서버 사이에 데이터 동기화는 클라이언트와 서버 상에 잔류하는 애플리케이션 데이터 사이에 일관성이 유지될 수 있도록 허용한다. 동기화는, 효율성을 위해, 클라이언트 디바이스가 일반적으로 로컬-저장된 데이터로 운영함으로써 중요하다. 예를 들면, 이-메일 클라이언트가 떨어진 이-메일 서버와 통신하는 휴대용 디바이스(랩탑 또는 모바일 텔레폰 같은) 상에서 실행하는 클라

이언트-서버 이메일 애플리케이션의 경우에, 이메일 애플리케이션 데이터의 동기화는 이메일 서버에 수신되는 이메일 메시지들이 이메일 클라이언트에 다운로드(이것은 다운힐 동기화라고 부름)되고 클라이언트에 의해 로컬-저장된 이메일 데이터 상에서 수행되는 변화(operation)들은 연속적으로 이메일 서버에 업로드(이것은 업힐 동기화라고 부름)되는 것이 요구된다.

[0003] 데이터 동기화는 서버와 클라이언트가 통신을 유지하고 있는 한 거의 연속적으로 클라이언트와 서버 사이에서 운영할 수 있다. 그러나, 통신 중단이거나 클라이언트가 오프라인 또는 "비행" 모드에서 운영할 필요성 때문에 서버와 클라이언트가 잠시 동안 통신이 단절되었을 때, 클라이언트 디바이스가 메모리에 저장된 로컬 애플리케이션 데이터가 거의 또는 전혀 없는 경우에 동기화는 더 필요하게 된다. 그러한 상태에서, 통신 중단 동안 서버에 수신되는 데이터의 많은 양은 클라이언트가 서버에 재연결될 때까지 클라이언트와 동기화되어 질 수 없을 것이고, 그리고 오프라인 클라이언트 변화들은 서버에 동기화되어 질 수 없을 것이다. 애플리케이션 데이터의 많은 양이 클라이언트에 다운로드 될 필요가 있는, 연속적인 동기화 단계는 때때로 클라이언트 디바이스의 "캐쉬 프라이밍(priming the cache)"로서 지칭된다.

[0004] 클라이언트 디바이스의 캐쉬 프라이밍에 대해 이전 접근법들은 불편 및/또는 느렸다. 이는 특히 클라이언트가 다운로드된 애플리케이션 데이터의 운영을 시작할 수 있기 전에 프라이밍(priming)이 완료될 것을 요구하는 이전 구현들에서 이다. 이 같은 방법에서 캐쉬 프라이밍은 사용자로 하여금 로컬-저장된 데이터를 사용하기 전에 상당한 시간을 기다릴 것을 요구하는 긴 프로세스일 수 있다.

[0005] 예를 들면, 이메일 애플리케이션 데이터를 가지는 클라이언트 디바이스의 캐쉬 프라이밍을 위한 하나의 이전 구현에서, 서버에 저장된 이메일 데이터의 상당한 부분이 하나 이상의 CD들 상에 처음 쓰여지고 나서, 사용자는 CD들로부터 클라이언트 디바이스(랩탑 같은) 상에 데이터를 물리적으로 복사한다. 다른 실시예들에서, 클라이언트에 동기화될 데이터의 세트는 네트워크 연결을 통해 다운로드되고, 이것은 통신 중단이 긴 다운힐 동기화 운영을 방해할 수 있고, 클라이언트 디바이스가 다운로드된 데이터의 어떤 것도 사용할 수 없게 된다는 것을 의미한다.

[0006] 다운힐 동기화의 다른 구현들에서, 클라이언트에 동기화될 데이터가, 가장 오래된 것부터 가장 최근의 데이터로, 시간순으로 서버로부터 다운로드되고, 이것은 클라이언트가 동기화 완료 이전에 오프라인으로 진입해야( 그리고 불완전한 데이터를 사용할 수 있음)한다면 가장 오래된 데이터만 클라이언트 디바이스 상에서 단지 이용될 수 있다는 것을 의미하고, 특히 사용자가 더 이상 서버에 연결할 수 없는 경우에 사용자를 좌절시킬 수 있다.

[0007] 클라이언트 디바이스들은 일반적으로 서버들보다 애플리케이션 데이터를 저장할 수 있는 메모리를 더 적게 가지고 있으므로, 클라이언트 사용자가 각각 동기화 이후에 가장 적절한 데이터에 즉각적인 액세스(access)할 수 있도록 하기 위해, 서버(종종 서버에 저장된 모든 데이터의 부분집합)에서 클라이언트로 더 만족할 만한 동기화 방법이 요구된다.

## 발명의 내용

[0008] 몇몇 실시예들에서 서버가 클라이언트와 정보를 동기화하기 위한 방법은, 서버에서, 서버상에서 유지되는 과거 동기화들의 기록에 기초하여, 지난 동기화에서 클라이언트와 동기화되지 않은 적어도 하나의 데이터와 동기화되지 않은 변화 동작 (change operation)들을 식별하는 단계; 서버와 클라이언트 사이에 정보를 선택적으로 동기화하기 위하여 통신을 전송하는 단계 - 상기 정보는 식별된 데이터와 식별된 변화 동작들을 포함하고, 상기 데이터는 높은 우선순위에서 낮은 우선순위의 순서로 배열되고 전송됨 -; 및 정보 동기화의 어느 종로 시점에서 클라이언트와 성공적으로 동기화된 통신에서 변화 동작들과 데이터의 일부분을 포함하도록 기록을 업데이트하는 단계를 포함한다.

[0009] 다른 실시예들에서, 서버가 클라이언트와 정보를 동기화하기 위한 방법은 서버와 데이터 동기화를 개시하기 위하여 클라이언트로부터 요청(request)을 수신하는 단계를 더 포함한다.

[0010] 다른 실시예들에서, 서버가 클라이언트와 정보를 동기화하기 위한 방법은 변화 동작들이 서버상에서 발생할 때에 따라 연대기 순서(chronological order)로 배열되는 변화 동작들을 포함하고, 오래된 변화들은 새로운 변화들 이전에 전송된다.

[0011] 몇몇 실시예들에서, 클라이언트가 서버와 정보를 동기화하기 위한 방법은, 클라이언트 디바이스에서, 다음 동기

화를 시작하기 위해 데이터와 변화 동작들의 다음 범위를 서버에게 통지하도록 요청(request)을 전송하는 단계; 상기 요청에 대한 응답에서, 클라이언트에 서버로부터 정보를 선택적으로 동기화하기 위해 서버로부터 통신을 수신하는 단계는 제1 동기화 메카니즘(mechanism)에서, 클라이언트와 이전에 동기화되지 않은 서버상에 데이터를 포함하는 제1 데이터 세트(set) - 제1 데이터 세트는 높은 우선순위에서 낮은 우선순위의 순서로 수신됨 -, 그리고 제2 동기화 메카니즘에서, 클라이언트와 지난 동기화에서 동기화되지 않은 변화 동작들을 포함하는 제2 데이터 세트 - 제2 데이터 세트는 서버상에서 각 변화 동작의 발생에 기초하여 연대기 순으로 배열되고 오래된 변화 동작들은 새로운 변화 동작들 이전에 수신됨 - 를 포함함; 클라이언트의 메모리에 적어도 상기 정보의 일부분을 저장하는 단계; 클라이언트 디바이스 사용자로 하여금 액세스(access)하고 정보 동기화의 어느 종료 시점에서 메모리에 성공적으로 저장된 적어도 상기 정보의 일부분을 사용가능한 단계를 포함한다.

[0012] 몇몇 실시예들에서, 이 클라이언트가 서버와 정보를 동기화하기 위한 방법은 데이터의 최근성(recency)와 같은 클라이언트의 사용자에 대한 데이터의 관련성(relevance)에 따라 배열된 우선순위를 가지며, 이는 더 관련 있는 데이터는 더 높은 우선순위가 주어지고 낮은 우선순위의 덜 관련 있는 데이터보다 이전에 클라이언트에 전송된다.

[0013] 몇몇 실시예들에서, 클라이언트가 서버와 정보를 동기화하기 위한 방법은, 제1 데이터 동기화 메카니즘과 제2 데이터 동기화 메카니즘은 서로 독립적이나 제1 데이터 세트와 제2 데이터 세트의 동기화가 동시에 동기화될 수 있도록 동시에 발생한다.

[0014] 몇몇 실시예들에서, 클라이언트가 서버와 정보를 동기화하기 위한 방법은 클라이언트와 동기화를 위한 서버상에 다음 데이터 범위(range)를 식별하도록 서버에 전송될 동기화를 위한 다음 요청(request)에 포함되도록 서버와 성공적으로 동기화된 마지막 데이터 범위의 기록을 클라이언트 상에서 유지하는 단계를 포함한다.

### 도면의 간단한 설명

[0015] 발명의 본질과 실시예들의 더 나은 이해를 위해, 같은 참조부호들이 도면들 내내 상응하는 부분들을 지칭하는 이하의 도면들과 함께, 레퍼런스(reference)는 이하 실시예들의 설명에서 만들어진다

도 1은 본원 발명의 몇 가지 실시예들에 따른 클라이언트-서버 기반의 네트워크 시스템을 설명하는 다이어그램이다.

도 2는 본원 발명의 몇 가지 실시예들에 따른 클라이언트-서버 기반의 네트워크 시스템에서 예시적인 서버의 블록 다이어그램이다.

도 3은 본원 발명의 몇 가지 실시예들에 따른 서버와 상호작용하는 예시적인 클라이언트의 블록 다이어그램이다.

도 4는 본원 발명의 몇 가지 실시예들에 따른 서버 상의 글로벌 히스토리 테이블에 포함된 정보를 설명하는 다이어그램이다.

도 5는 본원 발명의 몇 가지 실시예들에 따른 클라이언트의 로컬 히스토리 테이블에 포함된 정보를 설명하는 다이어그램이다.

도 6은 본원 발명의 몇 가지 실시예들에 따른 서버와 클라이언트 사이에 예시적인 동기화 상호작용 방식을 설명하는 다이어그램이다.

도 7A와 7B는 본원 발명의 몇 가지 실시예들에 따른 서버와 클라이언트 사이에 스타트 동기화 상호작용들에 포함된 정보를 설명하는 다이어그램들이다.

도 8A와 8B는 본원 발명의 몇 가지 실시예들에 따른 서버와 클라이언트 사이에 메인 동기화 상호작용들에 포함된 정보를 설명하는 다이어그램이다.

도 9A는 본원 발명의 몇 가지 실시예들에 따른 서버 상의 글로벌 히스토리 테이블과 서버 아이템들 사이의 관계를 설명하는 다이어그램이다.

도 9B는 본원 발명의 몇 가지 실시예들에 따른 클라이언트 사에 로컬/클라이언트 히스토리 테이블과 클라이언트 아이템들 사이의 관계를 설명하는 다이어그램이다.

도 9C는 본원 발명의 몇 가지 실시예들에 따른 서버와 클라이언트 사이에 동기화된 데이터의 예시적인 형태를 설명하는 다이어그램이다.

도 10A-10B, 10W 및 10X 들은 본원 발명의 몇 가지 실시예들에 따른 메인 동기화를 이끄는 스타트 동기화의 개념을 설명하는 다이어그램들이다.

도 10C-10D, 10Y 들은 본원 발명의 몇 가지 실시예들에 따른 일반적인 운영 상태 아래 메인 동기화의 개념을 설명하는 다이어그램들이다.

도 10E, -10F, 10Z 들은 본원 발명의 몇 가지 실시예들에 따른 단절의 긴 주기 후에 메인 동기화의 개념을 설명하는 다이어그램들이다.

도 11A은 본원 발명의 몇 가지 실시예들에 따른 순방향 동기화 동안 서버와 클라이언트 사이에 예시적인 상호작용 방식을 설명하는 다이어그램들이다.

도 11B는 본원 발명의 몇 가지 실시예들에 따른 클라이언트에 순방향 동기화의 예시적인 방법을 설명하는 흐름도이다.

도 11C는 본원 발명의 몇 가지 실시예들에 따른 서버에 순방향 동기화의 예시적인 방법을 설명하는 흐름도이다.

도 12A는 본원 발명의 몇 가지 실시예들에 따른 역방향 동기화 동안 서버와 클라이언트 사이에 예시적인 상호작용 방식을 설명하는 다이어그램들이다.

도 12B는 본원 발명의 몇 가지 실시예들에 따른 클라이언트에 역방향 동기화의 예시적인 방법을 설명하는 흐름도이다.

도 12C는 본원 발명의 몇 가지 실시예들에 따른 서버에 역방향 동기화의 예시적인 방법을 설명하는 흐름도이다.

도 13A-D 및 13X-Z 들은 본원 발명의 몇 가지 실시예들에 따른 클라이언트와 서버 사이에 업힐 동기화의 개념을 설명하는 다이어그램들이다.

도 14A는 본원 발명의 몇 가지 실시예들에 따른 업힐 동기화 동안 서버와 클라이언트 사이에 예시적인 상호작용 방식을 설명하는 다이어그램들이다.

도 14B는 본원 발명의 몇 가지 실시예들에 따른 클라이언트에 업힐 동기화의 예시적인 방법을 설명하는 흐름도이다.

도 14C는 본원 발명의 몇 가지 실시예들에 따른 서버에 업힐 동기화의 예시적인 방법을 설명하는 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

[0016] 본원 발명은 클라이언트-서버 시스템에 대한 것이고, 클라이언트와 서버 사이에 아이템들 동기화 시스템들과 상응하는 방법들에 대한 것이다. 이하의 개시물은 동기화된 데이터가 액세스하기 전에 로컬 캐쉬의 프라이밍을 완료할 위한 필요성을 극복하는 동시에, 클라이언트와 서버 사이의 데이터 교환과 클라이언트 캐쉬의 프라이밍이 일어나도록 허용하는 동기화 시스템과 방법을 설명한다. 설명된 방법들과 시스템들은 데이터 동기화와 관련된 어떤 클라이언트-서버 애플리케이션들에도 적용될 수 있다.

[0017] 도 1은 예시적인 클라이언트-서버 기반의 네트워크 시스템을 도식적으로 나타낸다. 클라이언트-서버 기반의 네트워크 시스템은 서버(102), 통신을 위한 네트워크(104) 및 복수의 클라이언트 디바이스들(106A-C)을 포함한다. 이 시스템의 보통의 구현이 적어도 하나의 서버, 적어도 하나의 통신 네트워크 및 복수의 클라이언트 디바이스들을 가지는 것으로 이해되어야 한다. 다른 구성에서, 시스템이 복수의 서버들, 복수의 네트워크들 및 복수의 클라이언트 디바이스들을 포함할 수 있다.

[0018] 몇몇 실시예들에서, 서버(102)와 클라이언트들(106)을 연결하는 네트워크(104)는 사설 또는 공공 네트워크 일 수 있고 데이터 교환이 가능하도록 서버(104)와 클라이언트들(106) 사이에 통신들을 할 수 있게 유선 또는 무선 연결될 수 있다. 몇몇 실시예들에서, 네트워크는 인터넷일 수 있고; 다른 것들에서, 통신 네트워크는 사설 또는 사용자들 그룹의 액세스가 제한되는 보안성을 가질 수 있다. 후자에서, 네트워크는 LAN 일 수 있다. 네트워크(104)의 다른 실시예들은 또한 WiFi 네트워크 및/또는 WiMax 네트워크를 포함한다. 네트워크(104)의 또 다른 실시예는 셀룰러 텔레폰들과 스마트 폰들과 같은 모바일 디바이스들에 의해 데이터의 전송과 수신을 가능하게 하는 셀룰러 네트워크일 수 있다. 그러한 셀룰러 네트워크는 단지 지불된 가입자들만 거기에 접속될 수 있는 사설

망으로 생각될 수 있다.

- [0019] 클라이언트 디바이스들은(106) 다양한 형태들을 가질 수 있다. 몇몇 실시예들에서, 클라이언트 디바이스는 컴퓨터 서버와 통신하는 로컬 컴퓨터일 수 있다. 다른 실시예들에서, 클라이언트 디바이스는 PDA(Personal Digital Assistant), 스마트 폰 또는 셀룰러 폰 중 어느 하나일 수 있다. 클라이언트 디바이스는 서버 상에도 운영할 수 있는 소프트웨어 애플리케이션을 운영할 수 있고, 또한, 서버 상에 운영하는 애플리케이션 뒤 끝과 함께 협력하는 애플리케이션 앞 끝을 운영할 수 있다. 서버(102)는 관리, 데이터 저장 요구들, 조직 및/또는 애플리케이션의 실행과 관련된 운영들을 수행할 수 있다. 개시물의 목적을 위하여, 클라이언트(106)와 서버(102) 모두는 동기화된 애플리케이션 데이터를 저장할 수 있는 메모리를 포함한다. 서버와 클라이언트의 더 상세한 설명은 각도 2와 3에서 이하 설명된다.
- [0020] 도 2는 도 1을 참고하여 설명된 클라이언트-서버 기반의 네트워크 시스템에서 예시적인 서버의 블록 다이어그램이다. 서버(200)는 전형적으로 하나 이상의 프로세싱 유닛들(CPU 들)(202), 하나 이상의 네트워크 또는 다른 통신 인터페이스들(206), 메모리(205) 및 이 같은 요소들을 상호 연결하도록 하나 이상의 통신 버스들(204)을 포함한다. 메모리(205)는 DRAM, SRAM, DDR RAM 또는 다른 램덤 액세스 반도체 이용한 메모리 디바이스들과 같은 고속 램덤 액세스 메모리; 그리고 자기 디스크 저장 디바이스들, 광 디스크 저장 디바이스들, 플래쉬 메모리 디바이스들 또는 다른 비-휘발성 반도체 이용한 저장 디바이스들과 같은 비-휘발성 메모리를 포함한다. 메모리(205)는 선택적으로 CPU(들)(202)로부터 따로 위치된 하나 이상의 저장 디바이스들을 포함할 수 있다.
- [0021] 몇몇 실시예들에서, 메모리(205)는 운영 시스템(207), 통신 모듈(208), 서버 애플리케이션 모듈(209), 서버 동기화 모듈(210) 및 데이터 구조들(270) 을 포함하는 프로그램들, 모듈들 및 데이터 구조들, 또는 앞선 것들의 부분집합을 저장한다. 이 컴포넌트들의 도시된 구조는 예시적인 것이며 컴포넌트들에 기여한 기능의 다른 배치나 구조를 불가능하게 하는 것이 아님을 주의하라. 다른 실시예들은 이러한 컴포넌트들의 부분집합 또는 확대집합을 포함하여, 어떤 조합에서 이러한 컴포넌트들에 기여한 기능들을 갖출 수 있다. 어떤 서버나 클라이언트 디바이스를 참고하여 여기에서 설명된 모든 그리고 어떤 소프트웨어 컴포넌트들에서도 같다.
- [0022] 운영 시스템(207)은 다양한 기본 시스템 서비스들을 다루고 하드웨어 중속된 일들을 수행하기 위한 절차들을 포함한다.
- [0023] 통신 모듈(208)은 서버 애플리케이션(209)와 다른 서버 또는 클라이언트 디바이스들과 인터페이싱을 위하여 사용된다. 통신 모듈(208)을 사용하는 인터페이싱은 하나 이상의 통신 네트워크 인터페이스(206)(유선 또는 무선) 그리고 인터넷, 다른 와이드 영역 네트워크들, 로컬 영역 네트워크들, 대도시 영역 네트워크들, 기타 등등과 같은 하나 이상의 통신 네트워크들을 통하여 성취된다.
- [0024] 애플리케이션 모듈(209)은 다른 클라이언트-주도의 또는 서버-주도의 애플리케이션들을 포함한다. 애플리케이션이 서버상에서 활동할 때, 애플리케이션이 클라이언트-주도의 애플리케이션인 경우, 애플리케이션 그 자체는 클라이언트 디바이스 에 상응되는 애플리케이션으로부터 유래하는 명령들에 의해 운전된다. 애플리케이션이 서버상에서 활동할 때, 애플리케이션이 서버-주도의 애플리케이션인 경우, 서버상의 애플리케이션은 클라이언트 디바이스에 상응하는 애플리케이션을 운전한다. 몇몇 실시예들에서, 셀룰러 텔레폰과 같은 클라이언트 디바이스들에 운영하기 위해 구성되고 조정된 애플리케이션들은 클라이언트와 서버에 의해 같이 운전될 수 있고, 서버 동기화 모듈(210)을 포함할 수 있다.
- [0025] 서버 동기화 모듈(210)은 클라이언트와 서버에 운영하는 또는 클라이언트와 서버 사이에 함께 운영하는(이메일 클라이언트와 서버 애플리케이션들의 경우와 같이) 하나 이상의 애플리케이션들에 의해 이용될 수 있는 데이터를 동기화하기 위해 사용된다. 동기화 모듈(210)은 하나의 애플리케이션에 특정되거나 하나 이상의 애플리케이션에 보편화된 동기화 구조들을 포함할 수 있다. 일 실시예에서, 동기화 모듈(210)은 다운힐 동기화 모듈(220), 업힐 동기화 모듈(280) 및 동기화 제어(290)를 포함한다. 이 같은 모듈들의 기능들은 클라이언트 상의 그 상태들에 대응하고, 이는 도 3을 참고하여 설명된다.
- [0026] 동기화 제어(290)는 데이터 동기화가 발생할 때 통신들 네트워크에 연결, 비동기화된 데이터 검출 또는 현재 다른 상태들에 따른 데이터 동기화 운영들을 제어하는 소프트웨어 모듈이다. 일 실시예에서, 비록 그 반대 또한 가능할지라도, 서버는 클라이언트로부터 동기화 요청에 응답하는 것에 의해 동기화를 개시하도록 구성된다.
- [0027] 다운힐 동기화 모듈(220)은 서버로부터 클라이언트로, 서버 상에 발생하였던 데이터 변화들과 데이터를 동기화하는 책임을 맡는다. 다시 말해서, 이 모듈은 서버 상에 데이터 변화들이 클라이언트 상에 잔류하는 데이터에 나타나도록 보증하기 위한 그리고 클라이언트 캐쉬를 프라이밍하기 위한 책임을 맡는다. 일 실시예에서, 다운힐



동기화는 순방향 동기화와 역방향 동기화 모듈들, 및 다른 애플리케이션들에 특정된 어떤 다른 동기화 모듈에 의해 구현된다.

[0028] 몇몇 실시예들에서, 순방향 동기화 모듈(230)은 서버상 데이터 변화들을 모니터하고 이 같은 변화들이 순차적으로 발생하고 그것들을 발생 순서로 클라이언트에 보고한다. / 즉, 순방향 동기화 모듈은 순방향 발생 순서-더 오래된 것에서 더 최근의 업데이트들로-로 클라이언트 디바이스에 데이터 업데이트들을 다운로드한다.

[0029] 몇몇 실시예들에서, 역방향 동기화 모듈(240)은 사용자와 관련성 정도 또는 우선순위의 순서로 클라이언트에 존재하는 데이터 아이템들을 다운로드한다. 몇몇 실시예들에서, 우선순위는 하나 이상의 기준에 근거하여 데이터의 물리적 조직으로 정의될 수 있고, 관련성은 동기화될 데이터에 사용자의 흥미 레벨로 정의될 수 있다. 그러므로, 우선순위는 사용자에게 데이터의 관련성 또는 흥미 정보를 정의하는 기준에 근거하여 설정될 수 있다. 예를 들면, 특정 데이터 업데이트의 우선순위 또는 관련성은 동기화되는 데이터 타입의 기능, 데이터 변화들이 발생하는 시간, 그리고 데이터 또는 데이터 변화들의 사용자 관심에 특징을 부여하는 데이터 또는 데이터 변화들과 관련된 다른 요소들로서 결정될 수 있다. 몇몇 실시예에서 높은 우선순위 데이터 아이템들이 낮은 우선순위 데이터들 전에 클라이언트에 다운로드되고, 그리고 클라이언트들은 오프라인 모드에서 부분적으로 동기화된 데이터로 운영할 수 있다는 것은 본원 발명의 장점이다. 선행 기술과 대비해서, 역방향 동기화 운영이 완료 이전에 방해받는 경우(예컨대, 통신 중단 또는 사용자가 클라이언트 디바이스를 오프라인에 두는 경우), 이것은 사용자가 오프라인 모드에서 하나 이상의 애플리케이션들을 위해 가장 높은 우선순위 로컬 데이터를 여전히 사용할 수 있도록 보장한다. 몇몇 실시예에서, 통신들이 클라이언트와 서버 사이에 재건될 때, 역방향 동기화는 중단된 곳에서 역방향 동기화가 완료될 때까지 계속해서 낮은 우선순위/관련성 데이터를 서버 다운로드를 계속한다.

[0030] 몇몇 실시예들에서, 역방향 동기화 운영을 위한 데이터 우선순위는 업데이트를 위한 더 높은 우선순위를 가지는 더 신생의 아이템들과 함께, 서버상에 저장된 데이터의 일부 또는 운영의 상대적인 시기에 의해 결정된다. 예를 들면, 이메일 애플리케이션의 환경에서, 이메일 클라이언트를 가지는 휴대용 디바이스의 사용자는 서버에 수신된 최신의 이메일 메시지들을 디바이스의 오프라인 사용 가능한 것을 선호할 수 있다. 이 예에서, 가장 새로운 메시지들은 그들이 클라이언트 디바이스에 처음 동기화되는 것을 보증하도록 가장 높은 우선순위를 가질 있다. 다른 실시예들에서, 가장 높은 우선순위는 특정 발송자(매니저 또는 배우자 같은) 또는 특정 주요 라인(경고 같은)과 관련된 이메일 데이터에 할당될 수 있다. 이 같은 방법에서, 본원 발명의 실시예들은 휴대용 전자 디바이스들의 사용자로 하여금 무슨 서버 데이터가 그들의 관심에 가장 관련 또는 가장 우선순위가 있는지 오프라인 사용이 가능토록 한다. 이 우선순위 개념은 사진들, 일정 또는 컨택 정보와 같은 디지털 매체 파일들, 또는 서류들, 스프레드시트와 같은 애플리케이션 데이터를, 포함하는 이메일 외 애플리케이션들을 위한 데이터에 적용되고 이에 한정되지 않는다.

[0031] 그러므로, 몇몇 실시예들에서, 서버상에 발생하는 데이터 변화들은 발생순(순방향 동기화의 경우에)과 우선순위/관련성 순(역방향 동기화를 위해) 모두에 클라이언트 디바이스와 동기화된다. 다른 동기화 모듈들(250)은 이 같은 애플리케이션에서 데이터를 위해 주문된 구성을 포함하는 다른 애플리케이션이 특별히 존재할 수 있다.

[0032] 순방향 동기화와 역방향 동기화를 사용하는 데이터 동기화의 같은 매커니즘(mechanism)은 이미지들, 별개이며 떨어진 이메일 메시지들을 포함하는 이메일 대화들, 컨택 정보 및 일정 정보 등을 포함하지만 이제 한정하지 않는 데이터의 다른 타입들에 유용될 수 있다. 몇몇 실시예에서, 관련성과 우선순위는 여기에 한정하지 않으나 에디팅(editing) 시간, 누가 데이터 또는 이미지들을 에디팅하는지, 메시지가 언제 도착하는지, 누가 메시지를 발송했는지와 같은 하나 이상의 기준들에 근거하여 설정될 수 있다.

[0033] 업힐 동기화 모듈(280)은 서버에서 클라이언트 상에 발생하였던 데이터 변화들의 동기화를 위한 책임이 있다. 서버상에서, 이 같은 업힐 동기화 모듈(280)은 클라이언트 상에 발생하였던 데이터 변화들을 얻고 서버상에 그 데이터가 나타나도록 (도 3과 관련하여 설명된)클라이언트의 업힐 동기화 모듈과 상호작용한다.

[0034] 데이터 모듈(270)은 클라이언트 디바이스들과 동기화를 위해 사용된 데이터뿐만 아니라 서버 애플리케이션 모듈(209)에 애플리케이션들에 의해 사용된 데이터 파일들을 포함한다. 몇몇 실시예들에서 데이터 모듈들(270)은 적어도 하나의 글로벌 히스토리 테이블(271), 클라이언트 특정 데이터를 포함하는 클라이언트 데이터(273) 및 애플리케이션 데이터(277)를 포함할 수 있다.

[0035] 글로벌 히스토리 테이블(271)은 서버상 데이터 변화들에 관한 정보를 포함한다. 특히, 글로벌 히스토리 테이블에 정보는 클라이언트와 순방향 동기화를 위해 사용될 수 있다. 몇몇 실시예들에서, 글로벌 히스토리 테이블에 정보는 발생순으로 저장된다.

- [0036] 클라이언트 데이터(271)은 서버와 상호작용하는 각각의 클라이언트에 특정된 데이터를 포함한다:( 예를 들면, 개개의 클라이언트는 클라이언트 'm'(272-B)에서 클라이언트 1 (272-A)를 포함한다). 각각의 클라이언트를 위해 저장된 특정 데이터는 최후에 처리된 서버 히스토리 운영 ID(identification)(서버 히스토리 운영 ID)(278), 역방향 동기화 영역들(275), 및 서버(274)에 의해 승인된 최후 클라이언트 히스토리 운영 ID(클라이언트 히스토리 운영 ID)를 포함한다. 캐쉬된 클라이언트 데이터는 다른 클라이언트-특정 데이터(279)가 동기화를 위해 사용된 다른 데이터(예컨대, 동기화 구성)를 포함하는 동안 특정 클라이언트와 관련된 서버상의 데이터 변화들과 데이터를 포함할 수 있다.
- [0037] 서버상 각 데이터 아이템은 데이터가 식별될 수 있고 역방향 동기화를 위한 동기화 메카니즘에 의해 우선적으로 처리될 수 있도록 고유 식별 또는 역방향 동기화 토큰(BST: Backward synchronization Token)를 가진다. 고유 식별은 올바른 데이터가 동기화될 수 있도록 보장한다. 데이터 아이템들의 우선순위가 역방향 동기화를 위해 설정될 수 있도록 BST가 잘 순서화되어 있다는 것을 인식하는 것은 중요하다. 몇몇 실시예들에서, 고유 식별은 특별한 데이터 아이템(예컨대, 메일 메시지)이 만들어지거나 수신되는 타임 스탬프에 의해 정의될 수 있고, 그 결과 모든 데이터 아이템들은 시간순으로 정렬될 수 있다. 여전히 다른 실시예들에서, 각 데이터 아이템에 대한 BST는 우선순위 또는 관련성을 결정하기 위해 사용되는 다른 파라미터들에 따라 적어도 하나의 타임 스탬프 요소를 가질 수 있다. 예를 들면, 몇몇 실시예들은 데이터와 크기에 근거하여 우선순위를 매기고, 여기서 데이터는 주요 기준으로, 크기는 부수적인 파라미터로써 사용될 수 있다. 몇몇 실시예들에서, 라벨들, 크기, 발송인 등과 같은 다른 정보와 타임 스탬프를 포함하는 다수의 파라미터를 결합하거나 가중치를 위한 알고리즘들은 역방향 동기화를 위한 데이터의 순서를 결정하는데 사용될 수 있다.
- [0038] 역방향 동기화 영역들(275)은 역방향 동기화 동안 클라이언트와 동기화되어 지는 데이터 영역들을 나타낸다. 각 역방향 동기화 영역은 두개의 BST를 포함한다. 함께, 이 같은 두 개의 BST들은 클라이언트와 동기화되어지는 서버상 데이터의 범위에서 시작점과 끝점을 나타낸다. 그러므로, 하나의 BST는 동기화되는 범위에서 첫 번째 데이터 아이템을 나타내고 두 번째 BST는 동기화되는 범위에서 마지막 데이터 아이템을 나타낸다. 역방향 동기화 영역들에 데이터는 연속되는 역방향 동기화들에서 생략될 것이다.
- [0039] 마지막으로 처리된 서버 히스토리 운영(SHO) ID(278)는 순방향 동기화에서 클라이언트와 동기화되는 마지막 SHO ID이다. 마지막으로 처리된 SHO ID는 클라이언트와 마지막으로 동기화되는 데이터 변화를 나타낸다. 이것은 서버로 하역금 글로벌 히스토리 테이블(271)에 데이터 변화들이 클라이언트와 동기화되고 더 이상 글로벌 히스토리 테이블에 유지될 필요가 없으므로 제거될 수 있다는 것을 결정하도록 한다. 그러므로 마지막으로 처리된 SHO ID(278)는 순방향 동기화에서 글로벌 히스토리 테이블의 크기를 조직하고 유지하기 위해 사용된다.
- [0040] 서버에 의해 승인되는 마지막 클라이언트 히스토리 운영(CHO) ID는 서버와 동기화되는 클라이언트 상의 마지막 CHO ID를 나타낸다. 이 CHO ID는 마지막 순방향 동기화에서 서버와 동기화되는 클라이언트 상의 마지막 데이터 변화를 나타낸다. 이는 또한 서버가 클라이언트와 다음에 연결할 때 다음 순방향 동기화를 위한 시작점으로 제공된다. SHO ID들과 유사하게 CHO ID들은 특정하게 클라이언트와 관련된다. 예를 들면, 그들은 클라이언트 상의 이메일의 제작, 발송 또는 수정과 관련된 타임 스탬프에 의해 나타날 수 있다.
- [0041] 도 3은 서버와 상호 작용하는 예시적인 클라이언트(300)의 블록 다이어그램이다. 클라이언트(300)는 일반적으로 하나 이상의 프로세싱 유닛들(CPU 들)(302), 하나 이상의 네트워크 또는 다른 통신 인터페이스들(306), 메모리(305), 그리고 이 같은 요소들을 상호연결시키기 위한 하나 이상의 통신 버스들(304)을 포함한다. 통신 버스들(304)은 시스템 컴포넌트들 사이 통신을 상호연결하고 제어하는 회로(때로 칩셋으로 칭함)를 포함할 수 있다.
- [0042] 클라이언트 디바이스(300)는 디스플레이(392) 같은 출력 디바이스와 사용자 입력 디바이스(394)를 가지는 사용자 인터페이스(390)를 포함할 수 있다. 디스플레이(392)는 능동 매트릭스 디스플레이 또는 터치 스크린 디스플레이 등 일 수 있고, 사용자 입력 디바이스(394)는 예를 들면, 수 입력 키 패드, 소프트 키들, 터치 패드, 글자와 숫자 입력 키 패드 또는 터치 스크린 등의 어느 조합을 포함할 수 있다. 메모리(305)는 고속 램덤 액세스 메모리(Random Access Memory)를 포함할 수 있고, 또한 하나 이상의 자기 디스크 저장 디바이스들과 같은 비-휘발성 메모리와 플래시 메모리같은 중앙 처리 장치들(302)로부터 떨어진 휴대용 저장 디바이스들을 포함한다. 몇몇 실시예들에서, 메모리(305)는 운영 시스템(301), 통신 모듈(303), 그래픽 모듈(307), 메모리 캐쉬(308), 클라이언트 애플리케이션 모듈(310), 클라이언트 동기화 모듈(320) 및 데이터(380)를 포함하는 프로그램들, 모듈들 및 데이터 구조들 또는 그것의 부분집합을 포함할 수 있다.
- [0043] 운영 시스템(301)은 다양한 기본적인 시스템 서비스들을 처리하고 하드웨어에 종속된 일들을 수행하기 위한 절



차들을 포함한다.

- [0044] 통신 모듈(303)은 하나 이상의 통신 네트워크 인터페이스들(306)(유선 또는 무선) 그리고 인터넷, 다른 와이드 범위 네트워크들, 로컬 범위 네트워크들, 도시형 네트워크들, 셀룰러 네트워크들 등과 같은 하나 이상의 통신 네트워크를 통하여 서버에 애플리케이션들(209)을 연결한다.
- [0045] 메모리 캐쉬(308)은 활성화된 애플리케이션에 빠른 접속을 위해 일시적인 정보를 일시적으로 저장한다. 저장된 정보의 예들은 메타데이터와 애플리케이션에 특정한 다른 정보를 포함한다. 일반적으로, 애플리케이션이 비활성화될 때, 정보는 삭제된다.
- [0046] 클라이언트 애플리케이션 모듈(310)은 클라이언트 디바이스(300)상에 실행될 수 있는 하나 이상의 애플리케이션을 포함한다. 이같은 애플리케이션 중 몇몇은 통신들, 사용자 인터페이스 관리, 주문형 애플리케이션, 그리고 이메일, 일정, 텍스트 메시징, 매체 플레이어 또는 서류 에디팅이나 소프트웨어 보는 것 등을 포함하는 특정한 클라이언트 애플리케이션들의 관리에 관련된 일들을 수행하도록 서버(200)와 상호작용하도록 구성된다. 클라이언트 애플리케이션은 운영중일 때 클라이언트 디바이스에 의해 운전된다.
- [0047] 클라이언트 동기화 모듈(320)은 클라이언트 디바이스 또는 서버상 독립적으로 운영될 있는 애플리케이션들을 위한 데이터의 동기화를 위하여 사용된다. 클라이언트 동기화 모듈(320)은 하나의 애플리케이션에 특정한 또는 하나 이상의 애플리케이션에 보편적인 동기화 방식들을 포함한다. 일 실시예에서, 동기화 모듈(210)은 다운힐 동기화 모듈(325), 업힐 동기화 모듈(326) 및 동기화 제어(375)를 포함한다. 이 모듈들의 기능은 서버상 그들의 상대들에 상응한다.
- [0048] 동기화 제어(375)는 통신 네트워크에 연결, 비동기화된 데이터의 검출, 또는 데이터 동기화가 발생할 때 존재하는 다른 상태들에 따라 데이터 동기화의 운영을 제어한다. 일 실시예에서, 비록 그 반대로 가능할지라도, 클라이언트가 서버에 동기화 요청을 개시하도록 구성된다.
- [0049] 다운힐 동기화 모듈(325)은 서버상 발생하는 클라이언트 데이터 변화들에 동기화하기 위한 책임이 있다. 클라이언트 상에 다운힐 동기화 모듈(325)은 서버상 데이터 변화들이 클라이언트 상 데이터와 동기화되는 것을 보장하도록 (도 2를 참고하여 설명된)서버의 카운터파트와 함께 작업한다. 일 실시예에서, 서버 카운터파트와 유사하게, 다운힐 동기화는 순방향 동기화와 역방향 동기화를 포함한다. 도 2를 참고하여 설명된 바와 같이, 순방향 동기화 모듈(330)은 이 같은 변화들이 시간순으로 발생하고 그것들을 클라이언트에 전송하는 서버상 데이터 변화들을 감시한다. 클라이언트 역방향 동기화 모듈(340)은 우선순위의 순서로, 또는 데이터 변화들의 관련성에 근거하여 서버상 데이터 변화들을 감시한다. 유사하게, 우선순위의 순서와 관련성의 정도는 서버에서 사용된 유사한 미리-결정된 기준에 근거한다. 그러므로 서버상 데이터 변화들은 발생순서와 우선순위의 순서 양쪽으로 클라이언트와 동기화될 수 있다. 다른 동기화 모듈들(350)은 애플리케이션에 데이터를 위해 맞춤형 구성을 포함하는 다른 애플리케이션들을 위하여 특별히 존재한다.
- [0050] 서버 카운터파트와 유사하게, 업힐 동기화 모듈(365)은 서버에 클라이언트 상 발생하는 데이터 변화들을 동기화하기 위한 책임이 있다. 클라이언트 상에서, 이 업힐 동기화 모듈(365)은 클라이언트 상에서 발생하는 데이터 변화들을 전송하고 서버에 의해 처리된 마지막 CHO ID를 클라이언트에 다시 전송할 때 같은 변화들이 서버상에 반영되는 것이 보장되도록 서버와 상호작용한다.
- [0051] 데이터 모듈(380)은 서버와 동기화를 위해 사용된 데이터뿐만 아니라 애플리케이션 모듈(310)에 애플리케이션들에 의해 사용된 데이터 파일들을 포함한다. 몇몇 실시예들에서, 데이터 모듈(380)들은 서버와 데이터 동기화의 목적을 위한 동기화 데이터(370), 캐쉬된 데이터(382), 및 구성 사양(384)을 포함할 수 있다.
- [0052] 동기화 데이터(370)는 로컬/클라이언트 히스토리 테이블(374), 다음 역방향 동기화 토큰(372), 가장 높은 동기화된 서버 히스토리 운영 식별(SHO ID)(378), 및 몇몇 애플리케이션들에 특별할 수 있는 다른 동기화 데이터와 같은 특정한 데이터를 포함한다. 이 같은 데이터는 서버와 클라이언트 디바이스의 동기화를 위하여 특별히 사용된다.
- [0053] 로컬/클라이언트 히스토리 테이블(374)은 클라이언트 상에서 데이터 변화들에 관한 정보를 포함한다. 이 테이블에서 각 목록은 CHO ID를 가지고 특별한 데이터 운영을 나타낸다. 특히, 로컬/클라이언트 히스토리 테이블에 정보는 클라이언트와 데이터의 업힐 동기화를 위하여 사용된다.
- [0054] 다음 역방향 동기화 토큰(BST)(372)은 다음 역방향 동기화를 위한 시작점을 서버에게 통지하도록 클라이언트로부터 전송된 다음 동기화 토큰이다. 가장 높은 동기화된 SHO ID(378)은 다음 순방향 동기화를 위한 시작점을 서

버에게 통지하도록 클라이언트로부터 전송된다. 다시 말해, 가장 높은 동기화된 SHO는 클라이언트에 되돌리개 되는 다음 변화들의 세트(예컨대, SHO ID가 가진 변화들이 가장 높은 SHO ID(378)보다 더 큼)에 시작점으로 나타난다.

[0055] 다른 클라이언트에 특정한 데이터(376)는 보편적인 애플리케이션들에서 동기화 또는 특정한 애플리케이션들을 위한 동기화를 위해 사용될 수 있다. 예를 들면, 다른 애플리케이션들은 그들의 각 데이터 동기화들을 위해 다른 기준을 가질 수 있다. 이 같은 다른 기준은 다른 애플리케이션에서 동기화를 위해 사용된 다른 클라이언트에 특정한 데이터로서 저장될 수 있다.

[0056] 캐쉬된 데이터(382)는 클라이언트 상에 저장된 데이터를 나타낸다. 이것은 다양한 애플리케이션들을 위해 사용된 데이터와 서버와 동기화되는 데이터를 포함한다.

[0057] 구성 사양(384)는 클라이언트 디바이스에 관한 특정한 구성 정보를 포함한다. 이 정보는 동기화, 상호작용들, 및 애플리케이션들의 비활동과 운영을 위한 다른 디바이스들에서 또는 서버에서 클라이언트를 확인하기 위해 사용된다. 예를 들면, 구성 사양에 포함된 정보는 클라이언트 식별, 클라이언트의 디바이스 사양, 동기화되는 애플리케이션의 식별 및 동기화를 위한 기준 등을 포함하며, 그러나 이에 한정되지는 않는다.

[0058] 몇몇 실시예에서, 데이터 동기화는 하나 이상의 독립된 동기화 메카니즘들을 포함한다. 다운힐 동기화는 서버가 클라이언트에 서버상 비동기화된 변화들을 전송하는 곳이고, 업힐 동기화는 클라이언트가 클라이언트로부터 서버에 비동기화된 데이터 변화들을 전송하는 곳이다. 몇몇 실시예에서, 다운힐 동기화는 역방향 동기화와 순방향 동기화를 포함한다. 역방향 동기화는 클라이언트와 동기화되지 않은 데이터를 동기화하고(즉, 클라이언트 상에 반영되어지지 않은 서버상에 데이터 아이템들), 반면에 순방향 동기화는 변화 동작들을 동기화 한다(즉, 이전 동기화 운영 이후에 발생하는 진행중인 데이터 변화들). 이 같은 동기화 메카니즘들의 각각은 별도이고 서로 독립적이며, 그들은 적절히 연속적 또는 동시적으로 운영될 수 있다. 다시, 데이터 동기화의 이 개념은 이메일 대화들 그리고 각 메일 대화들 및 관련된 데이터 세트(예컨대, 이미지들, 서류들)가 포함되는 개개의 메시지들에 관한 동기화에 동일하게 적용될 수 있다.

[0059] 도 4는 서버상에서 글로벌 히스토리 테이블(401)에 포함된 정보를 나타내는 다이어그램이다. 글로벌 히스토리 테이블(401)은 서버상에 데이터 변화들을 지칭하는 목록을 포함하고 순방향 동기화를 위해 사용된다. 몇몇 실시예들에서, 서버(402)에 데이터의 모든 변화는 서버 히스토리 운영 식별(SHO ID)로서 알려진 고유 식별을 가진다. 일 실시예에서, (첫 번째에서 n번째 서버 데이터 변화에 상응하는)서버 데이터 변화들(402A-D)은 테이블의 꼭대기에 가장 최신의 변화와 바닥에 가장 오래된 변화를 가지는 발생순으로 순서화될 수 있다. 이 실시예에서, 순서화는 변화 동작의 시간에 맞춘 발생에 의해 결정된다. 우선순위 또는 변화 동작의 관련성, 변화 동작의 카테고리나 타입 등과 같은 다른 방법들로 순서화도 또한 가능하다. 순방향 동기화 동안, 서버상에서 변화 동작들은 클라이언트 히스토리 테이블에 순서에 따라 서버에 전송된다.

[0060] 도 5는 클라이언트의 로컬 히스토리 테이블에 포함된 정보를 나타내는 다이어그램이다. 로컬/클라이언트 히스토리 테이블(501)은 클라이언트 상에서 데이터 변화들을 포함하고 업힐 동기화를 위해서 사용된다. 각 클라이언트 데이터 변화(502)는 클라이언트 히스토리 운영 식별(CHO ID)로서 알려진 고유 식별을 가진다. 일 실시예에서, (첫 번째에서 n번째 클라이언트 데이터 변화에 상응하는)클라이언트 데이터 변화들(502A-D)은 테이블의 꼭대기에 최신의 변화와 바닥에 가장 오래된 변화를 가지는 발생순으로 순서화될 수 있다. 서버상에서 그것의 카운트파트 글로벌 히스토리 테이블과 유사하게, 클라이언트 히스토리 테이블에서 아이템들의 순서화는 관련성, 우선순위 및 변화 동작의 카테고리 등에 근거하는 것과 같이 다르게 배열될 수 있다. 업힐 동기화 동안, 클라이언트는 운영들이 배열되는 순서로 클라이언트에 새로운 변화들을 전송한다.

[0061] 도 6은 서버(602)와 클라이언트(604) 사이에 예시적인 동기화 상호작용 방식을 나타내는 다이어그램이다. 동기화 상호작용 방식의 몇몇 실시예들에서, 스타트 동기화 핸드셰이크(621)는 여기서 메인 동기화 핸드셰이크(631)로 표현된 실제 데이터 동기화 프로세스를 개시하기 위해 서버(602)와 클라이언트(604) 사이에 발생한다. 스타트 동기화 핸드셰이크(621)는 처음 데이터 동기화 이전에 발생할 수 있고, 클라이언트가 서버(602)와 통신하지 않는 동안 연장된 기간 이후(예컨대, 통신 네트워크 범위 밖, 클라이언트 디바이스 전력 끊김, 클라이언트 상에서 애플리케이션 종료, 또는 디바이스 또는 서버가 통신 네트워크로부터 접속 해제될 때)에 발생할 수 있다.

[0062] 스타트 동기화 핸드셰이크(621)는 데이터 동기화의 시작점을 설정하도록 서브한다. 각 스타트 동기화 핸드셰이크는 요청(611)과 응답(612)을 가진다. 몇몇 실시예들에서, 요청(611)은 클라이언트로부터 비롯되고, 요청을 수

신하에, 서버는 응답(612)을 전송한다. 스타트 동기화는 이 개시물에서 제시된 바와 같이 메인 동기화의 일부 분으로써 포함되거나 완전히 제거될 수 있다. 게다가, 요청은 클라이언트 대신에 서버로부터 비롯될 수 있다.

[0063]

메인 동기화 핸드셰이크(631)는 동기화를 위한 데이터 범위들이 서버상에서 설정된 후에 발생한다. 스타트 동기화 핸드셰이크와 유사하게, 각 메인 동기화는 클라이언트로부터 전송된 요청(613)과 함께 시작한다. 요청(613)의 수신 이후, 서버는 응답(615)을 전송한다. 몇몇 실시예에서, 메인 동기화 요청(613)은 서버와 동기화되는 클라이언트로부터의 데이터 변화들을 포함하고, 그리고 메인 동기화 응답(615)은 클라이언트와 동기화되는 서버로부터 서버상에 데이터와 변화 동작들을 포함한다. 메인 동기화는 긴 접속 해제가 있거나 프로그램이 끝날 때(617) 종료한다.

[0064]

도 7A와 7B는 서버와 클라이언트 사이에 스타트 동기화 핸드셰이크들에 포함되는 정보를 나타내는 다이어그램들이다. 설명된 정보는 다운힐 동기화(순방향 그리고 역방향 동기화)와 업힐 동기화를 위해 포함된 것들을 포함한다. 도 7A는 스타트 동기화 요청(710)의 예이다. 스타트 동기화 요청(710)에 포함된 정보는 클라이언트 ID(711), 가장 마지막 역방향 동기화 영역(712), 및 다른 애플리케이션에 특정한 데이터(713)를 포함할 수 있다.

[0065]

클라이언트 ID(711)는 서버에서 클라이언트를 확인해서 서버는 매칭 클라이언트에 해당하는 정보를 회수한다. 마지막 역방향 동기화 영역(712)은 역방향 동기화에서 클라이언트와 마지막으로 동기화된 서버로부터의 데이터의 최종 범위를 포함한다. 역방향 동기화 영역(712)은 데이터 범위의 시작과 끝을 정의하는 두 개의 역방향 동기화 토큰들(BST)을 포함한다. 각 BST는 서버상에서 고유의 데이터를 유일하게 확인한다. 어떤 동기화도 발생하지 않았다면(예컨대, 처음 동기화 시도) 클라이언트는 어떤 범위도 가지지 않을 것이다. 두 개의 BST들은 다른 경우들(예컨대, 긴 기간의 접속 해제) 나타낼 것이다. 다른 애플리케이션에 특정한 데이터(713)는 동기화를 위한 특정한 구성 정보를 포함할 수 있다. 예를 들면, 처음 동기화하는 아이템들, 특정 데이터, 객체, 또는 아 아이템들을 동기화하기 위한 필터들, 및 동기화하기 위한 어떤 다른 기준에 관한 우선순위 정보가 포함될 수 있다. 구성 정보는 애플리케이션들에 특정될 수 있다.

[0066]

도 7B는 스타트 동기화 요청(710)을 인식하는 클라이언트로 서버에 의해 전송된 스타트 동기화 응답(760)의 다이어그램이다. 스타트 동기화 응답(760)에 포함된 정보는 서버(761)에 의해 인식된 마지막 클라이언트 히스토리 운영 식별, 가장 높은 서버 히스토리 운영 식별(762), 가장 높은 역방향 동기화 토큰(763) 및 다른 애플리케이션에 특정한 데이터(764)를 포함한다.

[0067]

서버에 의해 인식되는 마지막 클라이언트 히스토리 운영 식별(CHO ID)은 마지막 업힐 동기화에서 서버와 동기화되는 또는 서버에 의해 인식되는 클라이언트로부터 마지막 데이터 변화를 확인한다. 이 CHO ID의 목적은 다음 업힐 동기화를 위한 새로운 범위를 결정할 때 클라이언트를 돕는 것이다.

[0068]

가장 높은 서버 히스토리 운영 식별(SHO ID)(762)은 가장 최근 서버 데이터 변화 동작을 나타내고 순방향 동기화를 위한 다음 SHO ID로써 쓰일 것이다. 가장 높은 역방향 동기화 토큰(BST)(763)은 가장 높은 우선순위 데이터, 객체, 아이템 또는 클라이언트와 동기화되지 않은 서버상에서 데이터 변화를 나타낸다. 이것은 또한 다음 역방향 동기화에서 동기화를 위한 첫 번째 아이템이다. 몇몇 실시예들에서, 동기화를 위한 이벤트/데이터 변화의 관련성 또는 우선순위는 데이터 변화의 새로움(즉 발생의 시간), 데이터 변화와 관련된 아이템의 상태(예컨대, 이메일 애플리케이션에서 새로운 메시지는 컨택 업데이트보다 더 높은 상태/우선순위일 수 있음), 그리고 데이터 변화와 관련된 아이템과 관련된 라벨들 또는 태그들(예컨대, 몇몇 실시예들에서, 아이템은 "중요" 또는 "긴급" 또는 "항상 다운로드"에 관련된 라벨 집합을 가질 수 있음), 또는 이것들 중 어떤 것 또는 유사한 요인들의 조합들에 근거할 수 있다. 다른 특정한 데이터(764)는 특정한 애플리케이션들과 관련된 데이터의 동기화를 위한 구성 정보를 포함할 수 있다.

[0069]

도 8A와 8B는 서버와 클라이언트 사이에 메인 동기화 핸드셰이크들에 포함되는 정보를 나타내는 다이어그램이다. 도 8A는 서버에서 클라이언트에 의해 전송된 메인 동기화 요청(810)의 다이어그램이다. 몇몇 실시예들에서 메인 동기화 핸드셰이크들은 정보의 범위들이 스타트 동기화 핸드셰이크 후에 데이터 동기화 동안 설정된 이후에 즉시 발생한다. 메인 동기화 핸드셰이크들은 무기한으로 반복될 수 있어서 클라이언트와 서버가 연결되어 있는 한, 스타트 동기화 핸드셰이크들은 불필요하다. 일 실시예에서, 각 메인 동기화 요청(810)은 클라이언트 ID(811), 클라이언트(812)에 의해 처리된 마지막 서버 히스토리 운영(SHO) ID, 서버(813)와 동기화하는 클라이언트 데이터 변화들, 다음 역방향 동기화 토큰(BST)(814), 및 다른 애플리케이션에 특정한 데이터를 포함한다.

- [0070] 클라이언트 ID(811)은 데이터 동기화를 위한 서버상에서 해당하는 데이터 세트를 확인하기 위하여 사용된다. 클라이언트(812)에 의해 처리된 마지막 SHO ID는 클라이언트와 동기화된 마지막 데이터 변화를 확인하고, 클라이언트와 순방향 동기화를 위한 글로벌 히스토리 테이블 상에서 다음 서버 데이터 변화들의 세트를 결정하는 것을 돕는다.
- [0071] 서버(813)와 동기화되는 클라이언트 데이터 변화들은 클라이언트로부터 서버로 메인 동기화 요청에서 또한 전송된다. 몇몇 실시예들에서, 최대 수는 동기화를 시간을 감소시키기 위해 업힐 동기화에서 서버에 전송되는 클라이언트 데이터 변화들의 수에 제한한다.
- [0072] 다음 BST(814)는 다음 역방향 동기화가 어디서 시작해야하는 지를 나타낸다. 긴 기간의 접속해제가 있지 않다면, 역방향 동기화는 한 지점에서 시작하고 모든 관련된 데이터가 동기화될 때까지 높은 우선순위에서 낮은 우선순위로 한 방향으로 움직인다. 다시 말해, 역방향 동기화는 시간에서 특정 시점에 시작하는 서버상에 존재하는 데이터 아이템들의 세트를 동기화하는 것에 집중한다. 그러므로 다음 BST(814)는 전형적으로 동기화된 사전에 송부된 다음 BST(814)보다 우선순위에서 더 낮게 된다. 그러나 이 다음 BST(814)가 동기화가 시작하는 특정한 고유 시점에 가장 높은 우선순위 데이터 아이템을 나타내므로, 다음 BST(814)가 재연결의 시점에 가장 높은 우선순위 데이터일 수 있는 곳에서, 동기화가 서버로부터 긴 기간의 접속해제 이후에 재개되는 경우는 없다. 그러므로, 메인 동기화 핸드셰이크들은 방해없이 반복되는 한, 역방향 동기화는 한 방향으로 움직이고 새로운 시작점을 나타내는 각각의 다음 BST(814)는 사전에 송부된 모든 다음 BST(814)보다 우선순위에서 더 낮게 된다. 역방향 동기화가 종료될 때, 클라이언트와 서버가 단절되고 다시 연결되지 않는다면 다시 개시하지 않는다. 다른 애플리케이션에 특정한 데이터는 하나 이상의 애플리케이션들에 특정한 동기화를 위하여 구성 정보를 포함한다.
- [0073] 도 8B는 메인 동기화 요청(810)에 응답하여 서버로부터 클라이언트에 전송된 메인 동기화 응답(860)의 다이어그램이다. 메인 동기화 응답(860)에 포함된 정보는 새로운 다음 BST(861), 역방향 동기화를 위한 아이템들의 리스트(862), 다음 가장 높은 SHO ID(863), 비동기화된 변화들이 더 있는지 여부에 관한 식별자(866) 및 다른 애플리케이션에 특정한 데이터(867)를 포함한다.
- [0074] 역방향 동기화를 위한 아이템들의 리스트(862)는 현재 역방향 동기화에서 동기화를 위한 클라이언트에 전송된 비동기화된 서버 데이터 아이템들의 리스트를 포함한다. 아이템들의 리스트는 높은 우선순위에서 낮은 우선순위로 순서화하고 데이터 동기화를 위한 시간을 최소화하기 위해 최대 수를 제한할 수 있다. 몇몇 실시예들에서, 아이템들은 서버상에서 정렬되고 순서화되며, 일 실시예에서 아이템의 순서화는 아이템들이 큐 되어질 때 실시간으로 결정되고 클라이언트에 전송된다.
- [0075] 몇몇 실시예들에서, 새로운 다음 BST(861)는 다음 가장 높은 우선순위의 비동기화된 데이터, 객체, 아이템, 또는 역방향 동기화(862)를 위한 아이템들의 리스트에서 마지막 아이템 이후 변화를 나타낸다. 이 새로운 다음 BST(861)은 리스트에 모든 데이터(862)가 서버로부터 클라이언트로 성공적으로 동기화되었다고 가정할 때 역방향 동기화의 시작을 표시한다. 이 새로운 다음 BST(861)은 다음 역방향 동기화를 위해 다음 비동기화된 데이터 범위의 시작을 확인하기 위하여 클라이언트가 서버에 다시 전송하는 추천된 BST이다. 긴 접속 해제에 경우에는, 클라이언트는 현재 역방향 동기화에서 모든 데이터가 성공적으로 동기화되었다고 가정하면 재접속될 때 이 새로운 다음 BST(861)을 전송할 수 있다. 그러나, 현재 역방향 동기화가 방해받고 모든 데이터가 완전히 동기화되지 않는 경우, 클라이언트는 서버가 적절한 시작점을 결정하는 대신 서버에서 클라이언트에 의해 성공적으로 처리된 또는 서버와 성공적으로 동기화된) 마지막 BST를 서버에 전송할 옵션을 가질 것이다.
- [0076] 서버 히스토리 운영들(SHO) 또는 순방향 동기화(868)의 리스트는 마지막 동기화 이후에 서버 데이터에서 수행된 변화들의 리스트이다. 이 리스트는 데이터에 수행된 운영들과 변화들을 나타낸다. 몇몇 실시예들에서, 이 같은 변화들과 운영들은 이전에 클라이언트와 동기화된 데이터에 변화들과 운영들, 새로운 데이터를 수신하거나 만드는 것 또는 클라이언트와 동기화되지 않았던 데이터에 변화들과 운영들 등을 지칭한다. 일 실시예에서, 이 같은 변화 동작들은 별개로 데이터 그 자체로부터 확인될 수 있어서 변화들과 운영들은 데이터와 독립적으로 클라이언트와 동기화될 수 있다. 또 다른 실시예에서, 변화와 운영을 포함하는 데이터는 동기화를 위하여 선택될 수 있다. 몇몇 실시예에서, SHO들의 이 리스트는 서버상에서 정렬되고 순서화되며, 다른 실시예들에서, 아이템들의 순서화는 아이템들이 큐 되어질 때 실시간으로 결정되고 클라이언트에 보내진다.
- [0077] 순방향 동기화를 위한 새로운 가장 높은 동기화된 SHO ID(863)은 클라이언트가 서버와 순방향 동기화가 완료되었고, 클라이언트와 동기화되는 데이터 변화들을 나타내는, SHO ID의 리스트가 서버로부터 클라이언트에 성공적으로 순방향-동기화되었다고 가정하면 글로벌 히스토리 테이블에서 가장 높은 SHO ID를 나타낸다. 몇몇 실시예



에서, 이 새로운 가장 높은 동기화된 SHO ID(863)은 클라이언트와 정확하게 동기화된 글로벌 히스토리 테이블 상에서 데이터 변화일 수 있다. 다른 실시예들에서, 이 새로운 가장 높은 동기화된 SHO ID(863)은 사실 클라이언트와 동기화되는 데이터 변화들의 범위의 끝에 표시하는 글로벌 히스토리 상에서 데이터 변화일 수 있으나, 사실 클라이언트와 순방향 동기화되지 않을 것이다.

[0078] 이 순방향 동기화가 성공적인 완료될 때, 이 새로운 가장 높은 동기화된 SHD ID(863)은 클라이언트에 의해 처리된 가장 높은 SHD ID로서 메인 동기화 요청에서 클라이언트로부터 서버로 다시 전송될 것이다. 순방향 동기화가 방해되고 완료되지 못한다면, 클라이언트는 다음 순방향 동기화를 위해 적절한 시작점의 지정을 위하여 성공적으로 동기화되는 SHO ID의 리스트(868)에서 가장 높은 SHD ID를 다시 전송할 것이다. 이 매카니즘의 근거는 클라이언트로 하여금 이미 클라이언트와 무관하다고 결정된 운영들의 요청을 막는 것이다.

[0079] 긴 접속 해제에 경우에, 현재 순방향 동기화에서 모든 변화 동작들이 성공적으로 동기화된다고 가정하면, 클라이언트가 재접속될 때 클라이언트는 이 새로운 SHO ID를 전송할 수 있다. 순방향 동기화는 동기화 처음 개시된 이후에 변화한 것을 동기화하는 것을 인식해야 한다. 순방향 동기화는 역방향 동기화처럼 끝나지 않고 순방향 동기화는 동기화 개시 이후에 만들어지거나 수신된 새로운 데이터 또는 데이터 변화들을 동기화한다.

[0080] 현재의 발명의 고유한 양상 하나는 역방향 동기화와 순방향 동기화를 동시에 구현하는 능력에 있다. 이 구현은 동기화 과정들의 속도 향상뿐만 아니라 모든 데이터 아이템들과 데이터 아이템들의 우선순위 변화들이 적절한 방법으로 동기화되는 것을 보증하기 위하여 데이터 아이템들과 데이터 아이템들의 변화들이 제 시간에 역방향로(역방향 동기화), 제 시간에 순방향로(순방향 동기화) 가게 하여 동시에 동기화되는 것을 허용한다.

[0081] 서버에 의해 처리된 마지막 CHO ID(865)는 서버상에 동기화된 마지막 클라이언트 데이터 변화의 CHO ID이다. 수신될 때, 이것은 클라이언트에게 다음 업힐 동기화를 시작하는 클라이언트 히스토리 테이블 상에서 다음 가장 높은 CHO ID에서 시작하는 것을 알린다. 다른 실시예에서, 이미 설명한 바와 같이, 이 다음 가장 높은 CHO ID는 클라이언트에서 서버로 전송될 수 있고, 클라이언트로부터의 모든 데이터 변화들이 서버와 동기화된다면, 대신에 이 다음 가장 높은 CHO ID가 직접적으로 전송될 수 있다. 유사하게, 클라이언트 데이터 변화들의 최대 수는 데이터 동기화에 대한 시간을 최소화하기 위해 클라이언트에서 서버로 전송된 아이템들의 수를 제한하여 사용될 수 있다.

[0082] 식별자(866)는 동기화되지 않고 현재 동기화에 포함되지 않은 서버상에서 데이터 변화들이 있는지 여부를 표시하여 클라이언트에 전송될 것이다. 몇몇 실시예들에서, 비동기화된 아이템들이 남아있다면, 메인 동기화 과정은 즉시 클라이언트로부터 메인 동기화 요청을 시작하는 것을 다시 반복한다. 만일 동기화를 위한 남아있는 아이템이 없다면, 클라이언트와 서버는 다음 메인 동기화가 발생할 때까지 대기 모드로 진입할 것이다.

[0083] 다른 애플리케이션 특유의 데이터(867)는 특정 애플리케이션들에 동기화하기 위해 사용되는 데이터를 포함할 수 있다.

[0084] 도 9A와 9B는 데이터 아이템들의 다른 리스트들 사이의 관계와 서버와 클라이언트에서 각각 해당하는 히스토리 테이블을 나타내는 다이어그램이다. 도 9A는 서버 데이터의 리스트(910)와 상응하는 글로벌 히스토리 테이블(920)의 다이어그램이다. 서버 데이터(910)는 적어도 일부분은 특정 클라이언트 상에 저장된 데이터에 해당하는 데이터 리스트를 나타낸다.

[0085] 서버 데이터 리스트에서 각 데이터는 고유 식별(예컨대, 역방향 동기화 토큰(BST))과 관련되며, 어느 두 개의 데이터도 같은 BSD ID를 가지지 않는다. 일 실시예에서, BST는 데이터/타임 스탬프와 같을 수 있다; 다른 실시예에서, 다르게 정의될 수 있다. 도면에서, 각 BST는 알파벳(예, item a, item b 등)에 의해 식별되고, 각각은 하나의 데이터(datum)를 나타낸다. 데이터는 텍스트, 매체, 그래픽을 포함하나 여기에 제한하지 않는 어떤 애플리케이션과 관련될 수 있다. 몇몇 실시예에서, 이메일 메시지들, 연락처 정보, 일정 정보 등과 같은 데이터의 다른 타입들을 포함하는 하나의 데이터 리스트일 수 있다. 다른 실시예들에서, 다른 타입의 데이터를 나타내는 다른 리스트들 일 수 있다. 하나의 실시예에서, 모든 데이터에 대한 모든 데이터 변화들을 기록하는 하나의 글로벌 히스토리 테이블이 있다. 다른 실시예들에서, 서로 논리적으로 파티션된 다른 데이터에 대한 다른 글로벌 히스토리 테이블들이 있을 수 있다. 다시 말해, 하나의 파티션 내의 데이터는 같은 글로벌 히스토리 테이블을 가질 것이며, 두 번째 파티션 내의 데이터는 다른 글로벌 히스토리 테이블을 가질 것이다. 몇몇 실시예에서, 데이터는 데이터 타입들에 기반하여 다르게 파티션될 수 있다.

[0086] 서버 데이터(910)는 우선순위, 참신성, 관련성 또는 카테고리 등과 같은 어떤 순서에 의해 배열될 수 있다. 예를 들면, 우선순위에 의한 배열은 데이터 아이템들이 하나 이상의 기준에 기반하여 순서대로 순위를 매기도록

제안한다. 하나의 실시예에서, 우선순위에 의한 배열은 참신성에 의한 배열과 유사할 수 있으며, 여기서 더 최근의 타임-스탬프들을 가진 데이터는 더 오래된 타임 스탬프들을 가진 데이터보다 더 높은 순위가 매겨진다. 예를 들면, 더 최근에 수신되거나 만들어진 이메일 메시지들은 오래된 메시지들보다 더 높은 우선순위를 가질 것이다. 관련성은 클라이언트 디바이스의 사용자에게 의해 더 의도된 아이템들을 제안할 수 있다. 예를 들면, 이메일 문맥에서, 읽지 않은 메시지들, 일주일 내 수신된 메시지들, 또는 특정 주제에 관련된 메시지들을 다른 것보다 사용자에게 더 관련될 수 있다. 카테고리는 이메일 정보, 일정 정보, 컨택 정보 등과 같은 카테고리를 포함하나 이에 한정하지 않는 값인지 다른 애플리케이션에 대한 다른 데이터 그룹들이 동기화되기 전에 순위를 매겨지는 것을 제안한다.

[0087] 글로벌 히스토리 테이블(920)는 서버 데이터(910) 상에 수행되는 서버 히스토리 운영들(SHO)의 역사적인 기록들에 상응하는 입력들을 포함한다. 글로벌 히스토리 테이블(920)에 각 SHO는 고유 식별을 가진다. SHO ID는 운영이 발생할 때를 기반으로 또는 특정한 규칙들의 집합에 따라 할당될 수 있다. 일반적으로 데이터 변화들은 연속적인 같은 데이터에 대한 변화가 적절히 동일하게 될 수 있도록 시간에 대응한다. 순방향 동기화에서 데이터 변화를 처리하는 다른 방법들이 있다. 몇몇 실시예에서, 시간에서 순방향으로 움직이는 데이터 변화들에는 개별적으로 처리된다. 이 같은 구현은 모든 관련된 변화들이 그들이 발생할 때 연대순으로 처리되는 것이 요구된다. 다른 실시예에서, 변화를 가지는 모든 데이터 아이템들의 리스트는 유지되고 이 같은 데이터 아이템들의 변화들은 그들의 변화들과 함께 아이템들의 리스트를 서버로부터 송부하는 것에 의해 완전히 대체하거나 아이템들의 리스트에 대한 변화들의 리스트를 송부하는 것에 의해 완전히 대체될 수 있다. 만일 아이템들의 리스트에 대한 변화들의 리스트가 송부된다면, 서로 관련된 변화들은 그들이 발생할 때 연대순으로 또한 처리되어야 한다.

[0088] 실시예의 목적들을 위해, 서버 데이터(910)와 글로벌 히스토리 테이블(920)에 운영들 사이의 관계는 이메일 애플리케이션의 맥락에서 설명될 수 있다. 예를 들면, SHO(921)는 라벨이 서버상, 이메일 메시지, 아이템 d에 추가되는 것을 나타낸다. 유사하게, SHO(922)와 SHO(928) 또한 라벨이 각각 서버상 이메일 메시지, 아이템 f와 아이템 g에 추가되어 진 것을 나타낸다. SHO(923)과 SHO(927)는 스타(star)가 각각 아이템들 j와 k에 의해 표현되는 서버상 이메일 메시지에 추가되어 지는 것을 나타낸다. 계속하여, SHO(925, 926)에 새로운 메시지들의 수신과 SHO(929)에 서버 상의 초대 수신은 각각 서버상 새로운 아이템들(o, p, 및 q)을 나타낸다. 아이템 m과 같이 서버상 메시지가 편집될 때, 메시지에 대한 변화는 SHO(924)에 기록된다. 운영들은 동기화된 또는 비동기화된 모든 데이터에 적용될 수 있다. 운영들은 또한 데이터에 만들어진 사용자 변화들(예, 라벨을 추가, 스타를 추가 등) 또는 내재하는 시스템 변화들(예, 새롭게 수신된 이메일, 기록 보관된 이메일 메시지, 필터링하는 규칙들 등)를 나타낸다.

[0089] 도 9B는 클라이언트 데이터 리스트(930)와 로컬/클라이언트 히스토리 테이블(940)의 다이어그램이다. 클라이언트 데이터 리스트(930) 상의 하나의 데이터는 고유 식별(예, 토큰 ID)을 가진다. 실시예의 목적을 위해, 클라이언트 데이터 리스트(930) 상의 각 아이템은 숫자로 라벨(예, 아이템 2020, 아이템 2021 등)된다. 그러나, 실제 구현에서, 클라이언트 데이터(930)는 서버 데이터(910)에 적어도 하나의 데이터 부분집합에 상응하고, 여기서 상응하는 데이터는 동일하며(예, 같은 BST ID) 같은 정보를 포함한다. 동기화된 클라이언트 데이터(930)은 일반적으로 모든 운영들을 포함하는 서버상 그것들의 동일한 중복이다. 클라이언트 메모리 능력이 종종 서방 메모리 능력에 비해 풍부하지 않기 때문에, 클라이언트 데이터 리스트(930)은 적어도 성공적인 동기화 후 서버 데이터 일부분(910)과 동일하다.

[0090] 일 실시예에서, 클라이언트 디바이스의 클라이언트 데이터의 배치는 임의적이고 순서대로 일 수 있다. 이 예는 클라이언트 데이터가 서버와 직접적으로 동기화되지 않고 대신에 데이터 변화들 또는 변화 동작들이 서버와 동기화되는 것이다. 변화 동작들은 데이터 변화들이 데이터 만들기, 데이터 삭제, 데이터 수정 등에 한정하지 않고 포함하기 때문에 데이터보다 더 포괄적이다. 운영들은 데이터에 적용된다. 그러므로, 클라이언트 데이터(930)의 배치는 데이터 동기화의 목적을 위해 불필요하고 우선순위, 참신성, 라벨들, 관련성 또는 어떤 다른 설정 가능한 카테고리에 기반하여 배열될 수 있다.

[0091] 클라이언트 히스토리 테이블(940)은 클라이언트 데이터(930)의 적어도 일부분들 상 수행되는 클라이언트 히스토리 운영들(CHO)에 관한 기록을 유지한다. 클라이언트 상 CHO는 서버상 SHO와 독립적이며; 그들은 독립적인 운영들을 나타내고 각각 서로 직접적으로 동기화되지 않는다. 그러나, 데이터 상 CHO와 SHO의 효과들은 다운힐과 업힐 동기화들에 의해 동기화된다.

[0092] 실시예의 목적을 위해, 클라이언트 히스토리 테이블(940)과 클라이언트 데이터(930) 운영들 사이의 관계는 이메일 애플리케이션의 맥락에서 설명된다. 예를 들면, 클라이언트 히스토리 테이블 상 CHO(942)는 클라이언트 상

삭제된 아이템 2007 을 나타내고, CHO(943)은 아이템 2011에 상응하는 클라이언트 상 새로운 초대를 수신하는 것을 나타내며, CHO(944)는 사용자에게 의해 별 표시된 클라이언트 상 아이템 2018 또는 메시지를 나타내고, 라벨들은 각각 클라이언트 상 아이템들 2017과 2016에 CHO 945와 948을 통하여 부가되고, CHO 946과 947은 각각 클라이언트 상 아이템들 2020과 2021 로써 새로운 메시지를 수신한 것을 나타낸다.

[0093] 서버와 클라이언트 양쪽에 데이터의 고유 식별과 데이터 변화들 또는 운영들의 정렬은 동기화의 목적을 위해 중요하다. 식별과 정렬은 데이터 동기화가 어디서 중단되고 어디서 다시 시작하는지의 특정한 범위를 구별하기 위해 중요하다. 예를 들면, 서버 데이터(910)는 더 높은 우선순위 데이터(예, 아이템 s, r, q)가 더 낮은 우선순위 데이터(예, 아이템 n, o, p) 전에 리스트 되는 우선순위 순으로 배열된다. 유사하게, 글로벌 히스토리 테이블(920)의 SHO들과 클라이언트 히스토리 테이블(940)의 CHO들은 각 운영이 수행될 때 후 운영들이 선 운영들 상에 발생 되도록 예를 들면 연대순으로 배열된다. 식별과 정렬은 데이터 동기화 중단과 재시작할 때만 동기화를 위하여 다음 비동기화된 정보 범위를 확인하는 데 중요하다.

[0094] 도 9C는 서버와 클라이언트 사이에 동기화될 수 있는 예시적인 데이터를 나타낸다. 하나의 데이터(951)는 메타데이터(960)와 콘텐츠(964)를 가지면서 도 9C에 보여진다. 메타데이터(960)은 고유 식별자(961), 데이터 타입 정보(962) 및 식별을 위한 다른 형태들(963)에 제한하지 않고 포함한다. 실시예의 목적을 위해, 하나의 데이터(951)는 이메일 메시지, 이메일 대화, 컨택, 일정 입력, 또는 어떤 다른 데이터 객체 또는 JPG 파일, TIFF 파일, 텍스트 파일 등과 같은 데이터 파일일 수 있다. 일 실시예에서, 고유 식별자(961)는 하나의 데이터가 만 들어지거나 수정되어 질 때 타임 스탬프 및 도 9A와 9B에 설명된 BST ID와 같을 수도 있다.

[0095] 하나의 데이터는 데이터 타입 식별에 기반하여 카테고리될 수 있고 데이터 동기화 과정에서 배열되거나 우선순위를 매길 수 있다. 예를 들면, 하나의 데이터는 컨택 입력, 메시지, 일정 입력, 문서, 텍스트 파일, 이미지 파일 등으로 분류될 수 있다. 유사하게, 라벨, 속성, 플러그 등과 같은 다른 식별(963)이 데이터 동기화 동안 배열 또는 우선순위를 매기기 위해 추가 조건으로 또한 사용될 수도 있다. 예시적인 실시예의 목적을 위해, 이메일의 맥락에서, 라벨은 시스템 또는 사용자에게 의해 할당될 수 있고 모든 읽지 않은 메시지, 쓰레기, 보내진 메시지들, 또는 작업, 여과, 스포츠와 같은 다른 사용자 정의에 의한 카테고리를 포함하나 이에 제한하지 않는다.

[0096] 콘텐츠(964)는 남아있는 하나의 데이터를 형성하고, 이메일의 경우에는 메시지의 텍스트가 될 수 있고; 컨택의 경우에는, 이름, 전화 번호, 주소, 이메일 주소 등과 같은 다른 필드들을 포함할 수 있고; 일정 입력의 경우에는 약속 시간, 약속 주제 및 약속과 관련된 정보일 수 있다.

[0097] 몇몇 실시예들에서, 앞서 데이터 동기화에 대해 설명한 바와 같이, 세 가지 다른 프로세스로써 설명될 수 있다. 각 프로세스는 별개이고 나머지와는 독자적으로 운영한다. 선호되는 실시예에서, 그것들은 데이터 동기화의 속도를 증가시키고자 동시에 운영한다. 다른 실시예들에서, 각각은 한번에 하나, 한번에 두 개, 필요에 따라 임의적인 순서, 우선순위 순서 또는 적절한 어느 설정에 따라 별개로 활성화 될 수 있다. 이 같은 프로세스들 각각은 별개로 아래에서 설명되나, 세 가지 모두는 도 7과 8에 스타트 동기화와 메인 동기화 핸드셰이크들의 맥락에서 설명된 바와 같이 함께 적용되고, 결합될 수 있는 것으로 인식해야 한다.

[0098] 순방향과 역방향 동기화 양쪽을 포함하는 다운힐 동기화는 도 10A 내지 10F에서 설명된다. 역방향 동기화의 각 프로세스는 도 11A 내지 11C에서 설명되고, 반면에 순방향 동기화의 각 프로세스는 도 12A 내지 12C에서 설명된다. 업힐 동기화는 도 13과 14에서 설명된다. 예시적인 실시예의 목적을 위하여, 동기화 프로세스는 이메일 애플리케이션의 맥락에서 설명될 것이며, 다른 애플리케이션들에 똑같이 적용될 수 있음을 인식해야 한다.

[0099] 도 10A-10B, 10W 및 10X는 서버와 클라이언트 사이에 다운힐 동기화의 개시를 설명한다. 도 10A는 데이터 동기화가 긴 기간의 접속 해제 후 재-개시되기 전의 상태나 데이터 동기화가 서버와 클라이언트 사이에 발생하지 않을 때의 상태에서 글로벌 히스토리 테이블(1010B)와 서버 데이터 리스트(1010A)의 다이어그램을 보여준다. 도트라인은 스타트 동기화의 개시 또는 마지막 메인 동기화의 발생 후의 상태를 나타낸다. 이 도면에서, 메인 동기화는 역방향 동기화와 포워드 동기화 양쪽이 발생하는 곳과 데이터가 서버로부터 클라이언트로 동기화되는 곳이다. 다른 실시예들에서, 메인 동기화는 또한 공존하는 업힐 동기화의 운영을 포함한다.

[0100] 서버 데이터의 리스트(1010A)에 각 아이템은 고유 역방향 동기화 토큰(BST)을 가진다. 일 실시예에서, 서버 데이터의 리스트(1010A)는 꼭대기(예, BST n)에 최상의 우선순위와 바닥(예, BST a)에 최하의 우선순위를 가지고, 우선순위의 순서대로 배열된다. 데이터를 우선순위로 분류하는 것은 데이터 동기화와 동시에 또는 개시 이전에 수행될 수 있다. 데이터는 높은 우선순위에서 낮은 우선순위 순서로 동기화되고 전송된다. 일 실시예에서, 우선순위 순으로 데이터의 역방향 동기화는 개시 시점에 모든 비동기화된 데이터가 성공적으로 동기화될 때 까지 하

나의 방향으로 진행한다. 이 실시예에서 역방향 동기화는 애플리케이션이 종료되지 않는 한 또는 긴 접속 해제 이 있다면 동기화의 시점 후에 부가된 데이터를 동기화하기 위해 돌릴 수 없다. 이 실시예에서, 역방향 동기화는 데이터 상에 수행된 변화 동작들을 동기화할 수 없다. 또한 역방향 동기화 우선순위 순서는 관련성, 참신함 또는 다른 기준에 근거하여 배열될 수 있다.

[0101] 글로벌 히스토리 테이블(1010B)에 각 데이터 변화 동작은 고유 식별(SHO ID를 가진다. 몇몇 실시예들에서, 글로벌 히스토리 테이블(1010B)에 SHO들은 각 SHO가 발생한 때에 따라 연대순으로 배열되고, 그래서 오래된 운영들은 바닥에 나타나고 새로운 운영들은 꼭대기에 나타난다. 도면에서, 최신의 운영은 SHO 1이고 최후의 것은 SHO 18이다. 일 실시예에서, SHO들에 의해 표현되는 이 같은 변화 동작들은 글로벌 히스토리 테이블(1010B)에 배열된 같은 순서로 순방향 동기화된다.

[0102] 도면 예의 목적을 위해, 일 실시예에서, 데이터 동기화(예, 메인 동기화) 동안 역방향 동기화와 순방향 동기화 양쪽은 동시에 일어난다. 만일 데이터 동기화가 일어나지 않는다면, 역방향 동기화는 가장 높은 우선순위 BST(예, BST n)에서 시작하여 가장 낮은 우선순위(예, BST a)로 움직이고, 순방향 동기화는 연대순으로 SHO 1, SHO 2에서 SHO 18로 시작한다. 몇몇 실시예에서, 데이터와 SHO들의 최대 수가 매번 송부된다.

[0103] 도 10W는 데이터 동기화를 개시하기 위한 스타트 동기화 핸드셰이크 동안 서버와 클라이언트 사이에 핸드셰이크를 도시한다. 도 7A, 7B에 설명된 것과 유사하게 업힐 동기화들에 대한 정보의 일부분들보다 더 적다. 처음 스타트 동기화 핸드셰이크들에서, 클라이언트는 단지 도 10W의 요청(1047)에 클라이언트 ID를 송부할 것이다. 서버는 글로벌 히스토리 테이블에 가장 높은 SHO(예, SHO 18)와 가장 높은 BST(예, BST n)를 포함하는 응답(1048)으로 답신한다. 도 10X에 의해 도시된 바와 같이 연속되는 메인 동기화 핸드셰이크에서, 클라이언트 요청(1051)은 클라이언트 ID, 스타트 동기화 응답(1046)에서 수신된 가장 높은 SHO(예, SHO 18)와 서버 상 가장 높은 BST(예, BST n)를 포함할 것이다. 서버는 역방향 동기화를 위한 아이템 BST n, m, l, k를 포함하고 순방향 동기화를 위한 어떤 아이템들도 포함하지 않은 응답(1052)으로 답신한다. 양쪽 동기화에 되돌아온 아이템들의 수는 또한 같은 또는 다른 미리 정의된 최대 수에 의해 제한될 것이다. 응답(1052)는 또한 다음 메인 동기화를 위한 가장 높은 처리된 SHO ID(예, SHO 18)와 새로운 다음 BST(예, BST J)를 포함한다.

[0104] 도 10B는 처음 메인 동기화 후 서버 상 글로벌 히스토리 테이블과 데이터의 상태를 보여준다. 서버 BST n, m, l, k와 변화 동작들 SHO 18과 SHO 18전 다른 변화 동작들에 음영을 넣고, 이것들은 클라이언트와 더 이상 동기화될 필요가 없는 것을 나타낸다. 몇몇 실시예들에서, 서버는 모든 데이터 동기화의 기록을 유지한다. 기록은 서버 데이터 리스트와 글로벌 히스토리 테이블의 일부분으로써 유지될 수 있고, 또한 그것은 별개로 저장될 수 있다. 클라이언트와 성공적으로 동기화되었던 서버 상 변화 동작들의 범위와 서버 상 데이터의 범위를 추적하는 것에 의해, 이것들과 같은 데이터 범위들과 변화 동작들 범위들은 미래 데이터 동기화들에서 마주치게 될 때는 생략될 수도 있다. 모든 데이터는 그들이 사용자에 의해 삭제될 때까지 저장되는 동안, 글로벌 히스토리 테이블에 아이템은 모든 존재하는 클라이언트들에 동기화되어진 후 삭제될 수 있다. 이 같은 동기화된 SHO들의 운영들은 클라이언트들의 로컬 데이터에 포함될 것이고, 그러므로 미래 동기화에 필요하지 않을 것이다.

[0105] 도 10C 내지 10D, 10Y는 클라이언트와 서버 사이에 긴 접속 해제가 없는 일반적인 운영 상태 아래 메인 동기화의 개념을 설명하는 다이어그램이다. 몇몇 실시예들에서, 애플리케이션 또는 클라이언트와 서버 사이에 연결이 활성화된 이상, 메인 동기화는 규칙적 및 무기한으로 계속된다.

[0106] 도 10C는 첫 메인 동기화(도트 라인에 의해 나타난 바와 같이) 후에 더해진 새로운 변화 동작들과 새로운 데이터와 함께 도 10B와 같이 서버 상 데이터 변화 동작들과 데이터의 같은 상태를 보여준다. 예를 들면, BST o와 BST p에 상응하는 새로운 메시지들은 BST f와 관련된 메시지가 SHO 20에 의해 수정되어 지는 동안 서버에 도착한다. 새로운 메시지들(BST o와 BST p)의 도착은 글로벌 히스토리 테이블에서 각각 SHO 19와 SHO 21로써 나타난다. 동기화된 아니든 무관하게 데이터의 추가, 삭제 및 데이터의 수정하는 어떤 방법을 나타내는 변화 동작들을 포괄하는 것임을 인식해야 한다. 다시 말해, 글로벌 히스토리 테이블에 데이터 운영들은 서버 데이터 리스트에 모든 데이터의 모든 양상에 상응한다.

[0107] 도 10Y에 나타난 바와 같이 두 번째 메인 동기화 프로세스에서, 클라이언트는 다음 BST(예, BST j)와 클라이언트에 의해 처리된 가장 높은 동기화된 SHO(예, SHO 18)에 대한 서버에 요청(1053)을 송부한다. 클라이언트 요청(1053)에 응답하여, 서버는 각각 역방향과 순방향 동기화 프로세스에서 SHO(예, SHO 19, 20, 21)와 서버 데이터 다음 세트(예, BST j, i, h)를 포함하여 클라이언트에게 통신(1054)을 송부할 것이다. 추가로, 통신은 서버 상 변화 동작들과 비동기화된 데이터가 있는지 여부의 표시를 포함하고, 있다면 이 경우에(예, BST g와 SHO 21)와 같이, 그것들이 같이 클라이언트에 송부될 것이다.



- [0108] 도 10D는 두 번째 메인 동기화 후 서버 상 데이터(1010A)와 글로벌 히스토리 테이블(1010B)의 상태에 관한 다이어그램이다. 모든 동기화된 데이터(예, BST h 에서 h)와 동기화된 SHO(예, SHO 18-21)는 음영처리된다. 이전에 설명한 바와 같이, 몇 실시예들에서, 동기화된 SHO는 데이터 동기화 후 즉시 또는 미리-정의된 시간에 글로벌 히스토리 테이블에서 삭제될 것이다.
- [0109] 도 10E-10F, 10Z는 서버와 클라이언트 사이에 긴 접속 해제 후 데이터 동기화를 설명하는 다이어그램이다. 도 10E는 도트 라인에 의해 도시된 바와 같이, 마지막 (예, 두 번째) 메인 동기화 이후 서버에 더해진 새로운 운영들(예, SHO 22-29)과 새로운 아이템들(예, BST q, r)을 보여준다. 새로운 SHO들은 오래됐으나 비동기화된 메시지(예, BST c 상의 운영을 위한 SHO 22) 상에 수행되는 데이터 변화들, 서버 리스트에 새로운 아이템들의 추가(예, BST q, r의 추가에 상응하는 SHO 23, 24), 이전에 동기화된 데이터의 수정(예, BST q 상의 운영을 위한 SHO 25) 및 비동기화된 데이터의 수정(예, BST u 상의 운영을 위한 SHO 29)를 포함한다.
- [0110] 데이터 동기화를 개시하기 위해 서버에 클라이언트에 의한 스타트 동기화의 발생은 도면에서 생략했다. 몇몇 실시예들에서, 스타트 동기화는 영역의 시작점(예, BST p)와 영역의 끝점(예, BST h)에 의해 표현된 마지막 역방향 동기화된 영역(예, BST h에서 q)과 클라이언트 ID를 포함하는 서버와 클라이언트를 위한 선택적인 프로세스일 수 있다. 응답에서, 서버는 서버로부터 가장 높은 BST(예, BST v)와 가장 높은 SHO(예, SHO 29)를 포함하는 통신으로 회신한다.
- [0111] 그래서, 스타트 동기화에 응답하여, 메인 동기화는 도 10Z에 도시된 메인 동기화 핸드셰이크들과 같이 발생한다. 스타트 동기화 후, 클라이언트는 메인 동기화 요청(1055)에 서버에 송신할 정보를 선택한다. 메인 동기화는 다음 최고 높은 BST(예, BST r)과 최고 높은 동기화된 SHO(예, SHO 21)를 포함한다. 메인 동기화 요청(1055)에 응답하여, 서버는 마지막 동기화된 영역(BST h에서 p)은 생략하면서 높은 우선순위에서 낮은 우선순위로 역방향 동기화에 아이템들(BST r, q, g, f)을 다시 송부하고, 메인 동기화 응답(1056)을 거쳐 순방향 동기화에 SHO 22, 23, 24에 상응하는 데이터의 발생 순서대로 된 변화들을 송부한다.
- [0112] 몇몇 실시예들에서, 역방향과 순방향 동기화 프로세스들은 효율성을 유지하기 위해 같은 데이터 및/또는 변화 동작을 두 번 동기화하지 않기 위하여 각 다른 프로세스에 관한 충분한 정보를 공유한다. 일 실시예에서, 이 정보는 데이터 상에서 수행되는 변화들과 그 데이터가 별도로 구별될 수 있기 때문에 가능하게 된다. 예를 들면, 아이템(예컨대, 이메일 메시지)은 서버상에서 데이터 리스트에 더해지고, 대응하는 추가 변화는 글로벌 히스토리 테이블에 저장된다. 유사하게, 같은 이메일 메시지가 수정되는 경우, 라벨을 부가시키는 것과 같이, 라벨 부가는 글로벌 히스토리 테이블 상에서 별도의 데이터 변화로써 등록된다.
- [0113] 데이터와 변화 동작들의 별도의 동기화 개념은 적어도 세 가지 다른 시나리오들에 의해 설명될 수 있다. 첫 번째, 이메일 메시지가 사용자가 라벨을 부가하기 전에 처음 역방향 동기화에 의해 이전에 동기화된 경우, 순방향 동기화는 메시지 부가 운영을 생략하고 라벨 부가 운영을 동기화할 것이다. 두 번째, 이메일 메시지가 이전에 동기화되지 않은 경우, 역방향 동기화가 수정되고 비동기화된 이메일 메시지에 이르면, 수정과 메시지 양쪽은 역방향 동기화에 의해 동기화되고, 순방향 동기화는 메시지 부가와 라벨 부가 운영 양쪽을 생략할 것이다. 세 번째, 이메일 메시지가 동기화되지 않고 메시지 부가 운영이 데이터의 역방향 동기화 전에 순방향 동기화에 이르게 되는 경우, 비동기화되고 비수정된 메시지는 메시지 부가 운영에 의해 처음 동기화되고 데이터의 역방향 동기화는 생략될 것이다. 라벨 부가 운영은 그것이 발생할 때 메시지 부가 운영이 즉시 있는지 여부 또는 메시지 부가 운영 후 얼마나 많은 시간이 지나는 지에 따라 부가될 것이다.
- [0114] 순방향과 역방향 동기화의 동시 운영은 빠르고 효율적인 데이터 동기화를 보장한다. 순방향 동기화는 기록된 서버 데이터 상에서 수행된 각각 및 모든 변화 동작을 허용한다; 서버상에서 데이터에 수행된 데이터 운영들의 모든 양상을 완전히 다루는 것에 집중된다. 순방향 동기화는 클라이언트에서 더 높은 우선순위 데이터가 더 낮은 우선순위 데이터 이전에 동기화될 수 있도록 우선순위의 순서에 따라 서버상에서 데이터를 동기화하는 것을 중시한다. 역방향 동기화가 클라이언트 디바이스의 사용자에게 가장 중요한 가장 높은 우선순위 정보를 동기화하는 것이 목표인 반면, 순방향 동기화는 데이터의 어떤 운영도 동기화에서 빠지지 않도록 보장한다. 이 두 프로세스들이 동시에 발생하고 다른 동기화 프로세스에 관한 정보를 공유함에 따라, 각 프로세스가 전체의 데이터 동기화 프로세스를 더 빠르고 효율적으로 만들도록 서로 돕는 것을 허용함으로써, 하나의 프로세스에 의해 동기화된 데이터는 또 다른 것에 의해 동기화되지 않을 것이다. 게다가, 겹치는 데이터와 변화 동작들은 하나의 운영에 의해 어드레스 될 것이고 다른 것에 의해 생략될 것이다. 역방향 동기화가 완료된 이후, 짧은 시간에, 순방향 동기화는 클라이언트와 서버 사이에 접속단절이 되지 않는 한, 발생순서로 서버 데이터에 남아있는 변화 동작들 모두를 동기화하기 위해 남게 될 것이다. 간 접속 단절이 있을 때, 역방향 동기화는 일어나고 동기화되

지 않은 높은 우선순위 데이터를 동기화하도록 도울 것이며, 순방향 동기화가 그 프로세스를 따라잡도록 도울 것이다. 이 방법에서 역방향 동기화를 경유하여 클라이언트에 높은 우선순위 서버 데이터 변화들을 동기화하는 것에 의해, 클라이언트가 역방향 동기화 완료 이전에 오프라인으로 진입한다면, 클라이언트 사용자는 가장 높은 우선순위 데이터를 오프라인용으로 이용할 것이지만, 우선순위는 다른 실시예들에 따라서 정의된다.

[0115] 순방향과 역방향 동기화 방식의 더 나은 이해를 위하여, 각 동기화 프로세스는 각각 설명될 것이다. 도 11A는 클라이언트와 서버 사이에 예시적인 순방향 동기화 상호작용 방식을 나타낸다. 몇몇 실시예들에서, 스타트 동기화 핸드셰이크는 애플리케이션이 시작될 때 또는 애플리케이션이 특정 몇분 또는 몇시간으로 미리 정의된 긴 기간의 접속단절 없이 활동할 때 동기화를 개시한다.

[0116] 스타트 동기화는 동기화를 위한 개시로써 제공될 수 있다. 동기화 프로세스를 시작하는 것이 단지 요구되나, 우리의 현재 실시예에서, 그것은 긴 접속 단절 이후에 또한 사용된다. 이전에 설명한 바와 같이, 클라이언트는 서버에 클라이언트를 식별하기 위한 클라이언트 ID를 포함하는 스타트 동기화 요청(1101)을 서버에 전송하고, 이전에 동기화된 범위에서는 어떤 아이템도 순방향 동기화를 위해 불필요하다. 서버는 클라이언트에 다음 가장 높은 히스토리 운영 식별(SHO ID)을 포함하는 스타트 동기화 응답(1102)으로 답신한다. 이 SHO ID는 서버에서 데이터에 수행된 가장 최근 운영을 나타낸다. 클라이언트는 수신된 것 또는 스타트 동기화 이전에 클라이언트에 있는 다른 것에서, 어떤 SHO ID를 서버에 다시 전송할 것인지를 결정한다. 긴 접속 단절이 있는 경우에, 클라이언트는 이전에 순방향 동기화로부터 기인한 클라이언트에 있는 것을 전송한다.

[0117] 메인 동기화 핸드셰이크는 스타트 동기화 핸드셰이크의 뒤를 따른다. 몇몇 실시예들에서, 메인 동기화는 클라이언트에서 서버로 전송된 메인 동기화 요청에 의해 개시된다. 다른 실시예들에서, 메인 동기화는 서버로부터의 요청에 의해 개시될 수 있고 또는 전혀 요청이 없을 수도 있다. 도 11A에서, 클라이언트는 클라이언트 ID와 서버에서 클라이언트로 'x' 아이템들의 최대 수를 동기화하기 위하여 서버 글로벌 히스토리에서 시작점을 식별하도록 가장 높은 동기화된 SHO ID를 포함하는 메인 동기화 요청 1(1103)을 전송한다. 서버는 메인 동기화 요청 1(1103)에서 SHO ID에서 시작하는 글로벌 히스토리 테이블로부터 'x' 운영들의 최대, 새로운 가장 높은 동기화된 SHO ID 및 서버 변화들이 더 남아 있는지에 관한 표시자를 포함하는 메인 동기화 응답 1(1104)로 회신한다. 다른 실시예에서, 서버는 클라이언트에 의해 처리된 것으로 이미 인식되어 있는 어떤 변화들이 전송되지 않도록 확실히 하기 위하여 충분한 정보를 유지할 수 있다.

[0118] 클라이언트와 서버 사이에 긴 접속단절이 없는 일반적인 운영 조건들 아래, 메인 동기화는 비 동기화된 데이터 또는 변화 동작들이 있는 한 즉시 반복한다. 이것은 클라이언트에서 서버로 이전 메인 동기화 응답(예컨대, 메인 동기화 응답 1(1104)에서 서버로부터 수신된 다음 가장 높은 동기화된 SHO ID와 클라이언트 ID 및 서버로부터 송신을 위한 라인들 'x'의 최대 수를 포함하는 메인 동기화 요청 2(1105)를 전송하는 것을 포함한다. 서버는 서버로부터 'x' 이하의 운영들, 서버상에 비동기화된 운영들이 더 남아있는지에 관한 표시자, 및 존재한다면 글로벌 히스토리 테이블에서 다음 비동기화된 운영의 SHO ID를 포함하는 메인 동기화 응답 2(1106)로 회신할 것이다.

[0119] 메인 동기화 요청과 응답 핸드셰이크는 서버상에 비동기화된 운영들이 있거나 클라이언트와 서버 사이에 활성 연결이 있는 한 무기한으로 계속될 것이다. 서버에서 비동기화된 변화 동작들이 더 이상 없을 때 메인 동기화 핸드셰이크의 빈도는 감소할 수 있다. 활성 동기화 모드가 아닐 때 클라이언트와 서버는 대기 모드(1107)로 진입할 것이다. 어떤 접속 방해도 미리 정의된 기간 이상이 되지 않는 한, 메인 동기화 핸드셰이크는 서버상에서 고정된 기간 또는 활동성 등 (2208)을 포함하나 이에 한정하지 않는 수많은 미리-결정된 기준 중 하나에 근거하여 반복될 것이다.

[0120] 도 11B는 클라이언트에서 순방향 동기화의 예시적인 방법을 설명하는 흐름도이다. 몇몇 실시예에서, 클라이언트는 클라이언트 ID와 함께 서버에 스타트 동기화 요청을 전송하는 것에 의해 데이터 동기화를 개시한다. 응답에서, 클라이언트는 서버상에 가장 높은 SHO ID를 포함하는 서버로부터 스타트 동기화 응답을 수신한다(1121). 클라이언트는 순방향 동기화를 개시하기 위해 이전 메인 동기화로부터 클라이언트 상에 있는 것, 또는 서버로부터 수신된 것, 중 서버에 전송할 SHO ID에 관한 결정을 한다. 클라이언트는 클라이언트 ID, 운영들을 동기화하기 위한 글로벌 히스토리 테이블에서 시작점을 설정하기 위한 서버에서 선택된 SHO ID, 및 서버로부터 전송된 'x' 운영들의 한계를 포함하는 메인 동기화 요청(1122)을 전송한다. 클라이언트는 클라이언트와 이전에 비동기화된 'x' 운영들의 최대, 비동기화된 아이템들이 남아있는 지에 관한 표시자, 및 다음 가장 높은 비동기화된 SHO ID를 포함하는 서버로부터 메인 동기화 응답(1123)을 수신한다. 메인 동기화 응답에 근거하여, 클라이언트는 블록 (1124)에서 남아있는 비동기화된 아이템들이 있는지 결정한다. 만일 비동기화된 운영들이 있다면, 클라이언트는

블록(1122)에서 보여준 바와 같이 다른 메인 동기화 요청을 개시하고; 비동기화된 운영이 없다면, 클라이언트는 블록(1125)에서 미리-정의된 기준에 의해 정의된 대로 다음 메인 동기화가 활성화 될 때까지 유휴(idle) 상태를 유지한다. 만일 블록(1126)에서 보여준 대로 마지막 메인 동기화 이후에 긴 접속 해제가 있다면 클라이언트는 블록(1120)에서와 같이 다시 스타트 동기화를 개시할 것이고, 만일 긴 접속 해제가 없다면, 클라이언트는 미리 정의된 기준에 따라 새로운 메인 동기화 요청을 개시한다.

[0121]

도 11C는 서버에서 순방향 동기화의 예시적인 방법을 나타내는 흐름도이다. 서버는 블록(1131)에서 클라이언트 ID와 함께 클라이언트로부터 스타트 동기화 요청을 수신한다. 그 다음, 서버는 블록(1131)에서, 서버상 글로벌 히스토리 테이블에서 가장 최근의 SHO ID를 포함하는 클라이언트에 스타트 동기화 응답을 전송한다. 메인 동기화의 개시와 블록(1133)에서, 서버는 순방향 동기화를 위한 시작점을 설정하도록 SHO ID와 서버로부터 전송된 'x' 운영들의 최대와 함께 클라이언트로부터 메인 동기화 요청을 수신한다. 블록(1134)에서 서버는 마지막 동기화 이후에 'x' 비동기화된 운영들의 최대, 남아있는 비동기화된 아이템들의 표시자, 및 만일 그렇다면, 글로벌 히스토리 테이블에서 다음 비동기화된 SHO ID를 포함하는 클라이언트에 메인 동기화 응답을 전송한다. 블록(1135)에서, 서버는 클라이언트로부터 다음 메인 동기화 요청까지 기다리고 유휴 상태를 유지한다. 블록(1136)에서, 클라이언트로부터 메인 동기화 요청이 미리정의된 기간 이내와 같은, 미리정의된 기준에 맞는다면, 서버는 블록(1133)에서와 같이, 클라이언트로부터 메인 동기화 요청을 수신할 것이고, 만일 아니라면, 서버는 블록(1131)에서와 같이, 클라이언트로부터 스타트 동기화 요청을 수신할 것이다.

[0122]

도 12A는 역방향 동기화 동안 서버와 클라이언트 사이에 예시적인 상호작용 방식을 설명하는 다이어그램이다. 몇몇 실시예들에서, 이전에 설명된 바와 같이, 스타트 동기화는 서버 데이터 동기화 프로세스를 개시하고, 클라이언트가 긴 기간 동안 서버로부터 접속 해제된 이후에 서버 데이터 동기화를 재개한다. 스타트 동기화 요청(1201)은 클라이언트에서 서버로 전송된 클라이언트에 의해 처리된 마지막 역방향 동기화된 데이터 범위 및 클라이언트 ID를 포함한다. 서버는 가장 높은 역방향 동기화 토큰(BST)을 포함하는 스타트 동기화 응답으로 회신한다(1202). 그 다음, 클라이언트는 클라이언트 ID, 스타트 동기화 응답에서 수신된 가장 높은 BST, 및 동기화를 위한 'y' 데이터 아이템들의 최대 수를 포함하는 서버 메인 동기화 요청 1(1203)을 전송한다. 응답에서, 클라이언트는 높은 우선순위에서 낮은 우선순위 순으로 'y' 서버 데이터 아이템들의 최대, 남아있는 비동기화된 데이터의 표시자, 및 존재한다면, 새로운 다음 BST를 포함하는 서버로부터 메인 동기화 응답 1(1204)을 수신한다.

[0123]

서버와 클라이언트 사이에 네트워크 연결이 활성화 상태를 유지한다면, 메인 동기화 핸드셰이크는 반복한다. 메인 동기화 요청 2에서, 클라이언트는 비동기화된 서버 데이터의 다음 범위를 확인하기 위하여 메인 동기화 응답 1(1203)로부터 수신된 다음 가장 높은 BST, 클라이언트 ID 및 'y' 비동기화된 서버 데이터의 최대 수를 전송한다. 그 다음, 클라이언트는 우선순위 순으로 배역된 'y' 비동기화된 서버 데이터의 최대와 남아있는 비동기화된 데이터, 그리고 존재한다면, 다음 가장 높은 비동기화된 BST를 포함하는 서버로부터 메인 동기화 응답 2(1206)을 수신한다. 메인 동기화 요청과 응답은 역방향 동기화를 위해 남아있는 데이터가 더 이상 없고 클라이언트가 대기 모드로 진입할 때까지 계속한다(1207). 메인 동기화 프로세스는 고정된 기간과 같은, 미리 정의된 조건이 만족할 때까지 반복한다.

[0124]

도 12B는 클라이언트에서 역방향 동기화의 예시적인 방법을 나타내는 흐름도이다. 블록(1221)에서, 클라이언트는 클라이언트 ID와 처음 BST와 마지막 BST를 비동기화된 영역의 처음과 끝에 전달하도록 하는 클라이언트에 의해 처리된 마지막 동기화된 영역을 포함하는 서버에 스타트 동기화 요청을 전송한다. 마지막 동기화된 영역은 처음 스타트 동기화에서 비어 있을 것이다. 블록(1222)에서, 클라이언트는 가장 높은 우선순위 비동기화된 서버 데이터에 해당하는 가장 높은 BST를 포함하는 서버로부터 스타트 동기화 응답을 수신한다. 블록(1223)에서, 클라이언트는 클라이언트 ID, 스타트 동기화 응답에서 수신된 가장높은 BST 및 역방향 동기화를 위해 'y' 비동기화된 서버 데이터의 최대 수를 포함하는 메인 동기화 요청을 전송한다. 블록(1224)에서, 클라이언트는 우선순위 순의 'y' 비동기화된 서버 데이터의 최대 수, 남아있는 비동기화된 데이터 아이템들의 표시자 및 존재한다면, 다음 가장 높은 비동기화된 BST를 포함하여 서버로부터 메인 동기화 응답을 수신한다. 그 다음, 클라이언트는 블록(1225)에서 남아있는 비동기화된 서버 데이터가 있는 지를 결정하고, 있다면, 클라이언트는 미리-정의된 조건에 의해 결정된 대로 다음 메인 동기화 발생할 때까지 유휴를 유지한다(1226). 블록(1227)에서, 클라이언트는 마지막 메인 동기화 이후에 긴 접속해제가 있는지를 결정한다. 만일 존재한다면, 클라이언트는 블록(1221)에서와 같이 데이터 동기화를 재-개시하기 위하여 스타트 동기화로 되돌아 갈 것이다. 만일 존재하지 않는다면, 클라이언트는 블록(1223)에서와 같이 서버와 메인 동기화를 반복한다.

[0125]

도 12C는 서버에서 역방향 동기화의 예시적인 방법을 설명하는 흐름도이다. 블록(1251)에서, 서버는 클라이언

트 ID, 데이터 영역에서 처음 BST와 마지막 BST에 의해 나타내는 클라이언트에 의해 처리된 마지막 동기화된 데이터 영역을 포함하는 클라이언트로부터 스타트 동기화 요청을 수신한다. 동기화된 데이터 영역은 클라이언트에 대한 처음 스타트 동기화 요청이라면 비어있을 것이다. 블록(1252)에서, 스타트 동기화 응답이 서버상에서 가장 높은 BST를 포함하여 클라이언트에 전송된다. 블록(1253)에서, 서버는 클라이언트 ID, 역방향 동기화를 설정하기 위한 가장 높은 BST 및 서버로부터 역방향 동기화를 위한 'y' 아이템들의 최대 수를 포함하는 클라이언트로부터 메인 동기화 요청을 수신한다. 블록(1254)에서, 서버는 우선순위 순으로 'y' 비동기화된 데이터 아이템들의 최대, 남아있는 비동기화된 데이터의 표시자 및 존재한다면, 다음 비동기화된 데이터의 BST를 포함하는 클라이언트에 메인 동기화 응답을 전송한다. 블록(1255)에서, 서버는 미리-정의된 조건들에 의해 결정된 대로 다음 메인 동기화까지 휴식을 유지한다. 블록(1256)에서, 미리-정의된 기간 내에 클라이언트로부터 메인 동기화 요청이 있다면, 서버는 블록(1253)에서와 같이 클라이언트로부터 메인 동기화 요청을 수신할 것이다. 만일 긴 기간의 접속단절이 있다면, 서버는 블록(1251)에서와 같이 클라이언트로부터 스타트 동기화 요청을 수신할 것이다.

[0126]

도 13A-D 와 13X-Z는 클라이언트와 서버 사이에 업힐 동기화의 개념을 설명하는 다이어그램이다. 도 13A는 클라이언트 데이터의 리스트(1310A)와 클라이언트 히스토리 테이블(1310B)을 보여준다. 클라이언트 히스토리에서 각 목록은 클라이언트에 데이터에 수행된 운영들을 나타내는 클라이언트 히스토리 운영(CHO)이다. 몇몇 실시예들에서, 각 CHO는 클라이언트에 데이터에 적용됨으로써 만들기, 수정 또는 삭제 운영들을 포함하나, 서버로부터 새로운 데이터의 추가를 포함하지 않는다. 일 실시예에서, 클라이언트 데이터는 서버에서 모든 데이터의 부분집합으로 구성된다. 그러므로, 서버는 모든 데이터의 완전한 기록을 유지하고 클라이언트 데이터는 그 완전한 데이터 세트의 일부분을 포함한다. 다른 실시예들에서, 클라이언트에서 데이터 세트는 서버에서 세트와 같다.

[0127]

클라이언트가 늘어난 기간 동안 서버로부터 접속해제되거나 서버와 동기화되지 않을 때 또는 데이터를 사용하는 애플리케이션이 종료되고 나서 다시 작동될 때, 스타트 동기화 핸드셰이크는 전체의 동기화 프로세스를 개시하기 위해 이용된다. 만일 클라이언트가 서버와 동기화되지 않는다면, 서버와 동기화할 CHO는 없고, 그래서 어떤 데이터 변화도 개시 이후에 클라이언트에서 서버로 전송되지 않을 것이다. 클라이언트가 늘어난 기간동안 서버로부터 접속해제된다면, 데이터 변화들은 마지막 동기화 이후에 클라이언트 상에서 발생할 것이다. 또한, 마지막 동기화가 방해받거나 방해 이전에 모든 데이터가 마지막 동기화에서 동기화되지 않는다면 비동기화된 데이터 변화들은 클라이언트 상에서 유지될 수 있다.

[0128]

도 13A는 일련의 데이터 번호 운영들(CHO 25-29)이 서버와 동기화되지 않은 클라이언트와 서버 사이에 긴 접속해제가 있는 예를 보여준다. 도트 라인은 개시 때 데이터와 클라이언트 히스토리 테이블의 상태를 나타낸다. 다이어그램은 CHO 25가 클라이언트 아이템 gg에서 데이터 변화를 나타내고, CHO 26은 아이템 kk 에서 데이터 변화를 나타내고, CHO 27은 아이템 ii에서 데이터 변화를 나타내고, CHO 28은 아이템 oo에서 데이터 변화를 나타내고, CHO 29은 아이템 pp에서 데이터 변화를 나타내는 것을 보여준다. 예시적인 도해의 목적을 위하여, 각 클라이언트 아인템이 이메일 애플리케이션에서 메시지라고 한다면, 각 데이터 변화는 클라이언트 상에서 라벨을 부가하는 것, 메시지에 별표를 표시하는 것, 메시지를 삭제하는 것, 메시지를 작성하는 것 등에 한정하지 않고 포함하는 메시지에서의 운영들을 나타낸다.

[0129]

도 13X는 동기화 프로세스를 개시하기 위한 클라이언트와 서버 사이에 스타트 동기화 핸드셰이크를 설명하는 다이어그램이다. 클라이언트는 서버에서 클라이언트를 식별하도록 클라이언트 ID를 포함하는 스타트 동기화 요청(1351)을 전송한다. 서버는 데이터 동기화를 위해 시작점으로서 클라이언트에 (마지막 데이터 동기화에서)서버에 의해 처리된 마지막 CHO ID(예컨대, CHO 24)를포함하는 스타트 동기화 응답(1352)으로 회신한다. 몇몇 실시예들에서, 데이터 변화들은, 이전에 설명한 순방향 동기화처럼, 발생순서로 정렬된다. 클라이언트 히스토리 테이블의 꼭대기에 새로운 아이템들은 더 높은 CHO ID(예컨대, CHO 29)를 가지고, 오래된 아이템들은 더 낮은 CHO ID(예컨대 26)를 가진 바닥에 위치한다. 일 실시예에서, CHO ID는 데이터 운영이 발생하는 시간과 관련된다.

[0130]

도 13Y는 데이터 동기화의 프로세스에서 클라이언트와 서버 사이에 메인 동기화 핸드셰이크를 설명하는 다이어그램이다. 스타트 동기화 응답(1352)에서 서버에 의해 제공된 CHO ID와 함께, 클라이언트는 CHO 25-28에 의해 표현되는 아이템들 gg, kk, oo에 해당하는 서버 데이터 변화들을 포함하는 메인 동기화 요청(1352)을 전송한다. 몇몇 실시예들에서 데이터 변화들은 데이터에서 분리된다. 이것은 데이터 변화들이 전체의 데이터보다 오히려 데이터에 변화 동작들으로써 전송된다는 것을 의미한다. 예를 들면, 별표가 메시지에 부가된다면, 또는 연락처를 위해 폰 번호가 변화한다면, 또는 두 개의 새로운 라벨들이 메시지에 부가된다면, 별표를 부가하는 것, 연락처를 위한 새로운 폰 번호를 변화하는 것, 및 두 개의 라벨들을 부가하는 것의 운영들은 데이터와 독립적으로 전



송된다. 다른 실시예들에서, 수정들을 포함하는 데이터(예컨대 아이템들 gg, kk, oo)는 서버에서 상응하는 수정되지 않은 아이템들로 대체하기 위하여 서버에 전송될 수 있다. 클라이언트의 요청에 전송된 모든 CHO가 동기화된다면, 서버는 다음 동기화를 위한 시작점으로서 다음 메인 동기화 응답에서 클라이언트에게 다시, 클라이언트에 의해 제안된 대로 다음 CHO(예컨대, CHO 28)를 포함하는 메인 동기화 응답을 전송한다. 클라이언트로부터 모든 CHO들이 서버와 동기화된다면, 서버는 메인 동기화 응답에서 클라이언트에 마지막 동기화된 CHO를 되돌리거나 또는 CHO ID를 전혀 되돌리지 않을 수 있다.

[0131]

도 13B는 처음 메인 동기화 이후에 클라이언트 데이터의 리스트와 클라이언트 히스토리 테이블의 다이어그램을 보여준다. 클라이언트 상에 그늘진 아이템들(예컨대, 아이템들 gg, ii, kk, oo)는 서버와 동기화되어진 변화들을 가진 영향 받은 데이터를 나타낸다. 로컬 히스토리 테이블에서, CHO 29는 비동기화된 채로 남아있다. 몇몇 실시예들에서, 성 공적으로 동기화된 CHO들은 메모리 공간을 절약하기 위해 삭제되고 제거된다. 다른 실시예들에서, 동기화된 CHO는 짧은 기간 동안 유지되고 나서 특정 듀레이션 이후 또는 메모리가 필요될 때 제거된다.

[0132]

도 13Z는 클라이언트와 서버 사이에 두 번째 메인 동기화 핸드셰이크를 보여준다. 클라이언트와 서버 사이에 연결이 활성화되고 긴 접속해제가 없는 보통 운영 아래서, 몇몇 실시예들에서, 이전 메인 동기화가 종료될 때, 클라이언트 상에 남아있는 비동기화된 데이터 변화들이 있다면 새로운 메인 동기화가 즉시 발생한다. 메인 동기화 핸드셰이크는 클라이언트 상에 더 이상 비동기화된 아이템들이 남아있지 않을 때까지 반복된다. 이 예에서, 클라이언트는 메인 동기화 응답(1345)에 의해 전송된 다음 가장 높은 CHO ID(예컨대, CHO 29)에 따라, 아이템들 pp, dd, qq, 및 mm에서 변화들을 표현하는, CHO 29-32 및 도 13C에 도시된 바와 같이 클라이언트 히스토리 테이블 상에서 다음 가장 높은 비동기화된 CHO(예컨대, 남아있지 않음)를 포함하는 서버에 메인 동기화 요청(1355)을 전송한다. 순방향 동기화와 역방향 동기화와 유사하게, 최대 수는 동기화 시간들을 최소화하기 위하여 데이터 동기화들에서 데이터 변화들의 수에 제한되어 사용될 수 있다. 이 메인 동기화는 클라이언트 상에 모든 CHO가 서버와 동기화되어 질 때까지 계속한다. 예를 들면, CHO 32 이후에 더 이상 데이터 변화들은 없으므로, 다음 가장 높은 CHO는 클라이언트에서 서버로 메인 동기화 반응(1356)에서 전송될 것이다. 도 13D는 데이터 변화들(CHO 29-32)에 상응하는 아이템들 pp, dd, qq, 및 mm이 음영 처리를 한 것을 보여주고, 이 변화들이 서버와 동기화되는 것을 보여준다. 어떤 비동기화된 CHO도 동기화된 CHO들이 삭제됨으로써 클라이언트 히스토리 테이블에 남지 않는다.

[0133]

몇몇 실시예들에서, 데이터 동기화는 클라이언트에 의하는 것 대신에 서버에 의해 개시될 수 있음을 이해해야 한다. 예를 들면, 서버는 데이터 번호들이 클라이언트 상에서 더 발생하는 지를 결정하기 위하여 클라이언트와 간격을 두고 통신할 수 있다. 다운힐 또는 업힐 동기화 양쪽에서, 데이터 동기화를 위하여 어떤 디바이스를 사용하는 것은 중요하지 않다. 대신에, 하나의 디바이스에서 다른 것에 전송될 수 있는 통신과 정보는 데이터 변화들이 동기화되었는지 여부를 구별짓기 위하여 그리고 시작점을 결정하기 위하여 중대하다.

[0134]

도 14A는 업힐 동기화 동안 서버와 클라이언트 사이에 예시적인 상호작용 방식을 설명하는 다이어그램이다. 일 실시예에서, 클라이언트는 클라이언트에 의해 처리된 마지막 CHO ID와 클라이언트 ID를 포함하는 서버에 스타트 동기화 요청(1401)을 전송한다. 서버는 그것이 처리되거나 인식되거나 또는 동기화되는 가장 높은 CHO ID를 포함하는 스타트 동기화 응답(1402)을 가지고 회신한다. 데이터 동기화가 스타트 동기화 응답(1402)에 정보에서 근거하여 시작해야한다는 것을 알면서, 클라이언트는 클라이언트 ID와 서버에 'z' 비동기화된 CHO의 최대 수를 포함하는 메인 동기화 요청 1(1403)을 전송한다. 서버는 화살표에서 서버에 의해 처리되는 또는 인식되는 가장 높은 CHO ID를 포함하는 메인 응답 1(1404)으로 회신한다. 동기화가 중단되지 않는다고 가정하면서, 다음 가장 높은 CHO ID는 서버가 메인 동기화 요청 1(1403)에서 클라이언트로부터 수신되는 가장 높은 비동기화된 CHO ID와 같을 것이다. 그러나, 중단이 발생한다면, 서버는 클라이언트에 의해 처리된 마지막 CHO ID를 전송할 수 있고, 서버는 다음 동기화가 어디서 시작해야하는 지를 결정할 수 있다.

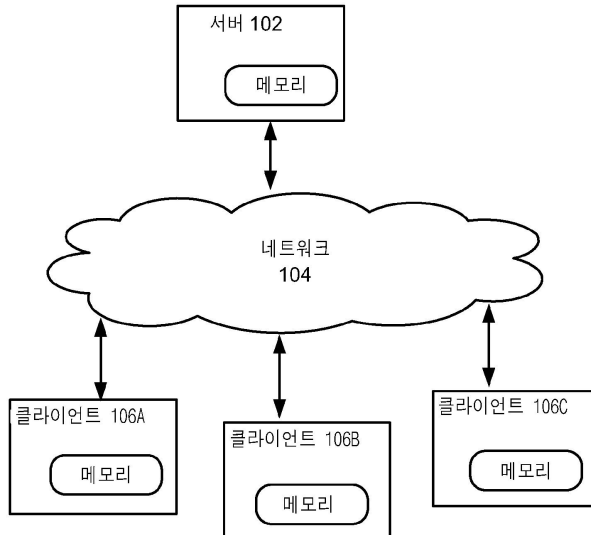
[0135]

메인 동기화 2 에서, 클라이언트에 남아있는 비동기화된 데이터의 동기화는 클라이언트 ID와 'z' 데이터 변화들의 최대 수의 제한을 포함하는 메인 동기화 요청 2(1405)를 전송하는 클라이언트에서 계속한다. 서버는 메인 동기화 요청 2에 의해 제안된 바와 같이 다음 가장 높은 CHO ID 또는 서버에 의해 처리된/동기화된 가장 높은 CHO를 포함하는 메인 동기화 응답 2(1406)으로 회신한다. 메인 동기화 핸드셰이크는 클라이언트에 모든 데이터 변화들이 서버와 동기화되고 클라이언트와 서버가 화살표(1407)에서처럼 대기 모드로 스위치될 때까지 반복된다. 클라이언트와 서버 사이에 긴 접속해제의 기간들이 없다고 가정하면, 클라이언트 데이터 변화들을 동기화하기 위한 메인 동기화 핸드셰이크는 화살표(1408)에서처럼 미리정의된 조건 또는 기준에 따라 다시 개시한다. 스타트 동기화는 긴 접속해제가 있지 않거나 애플리케이션이 종료하지 않는다면 재시작하지 않는다.

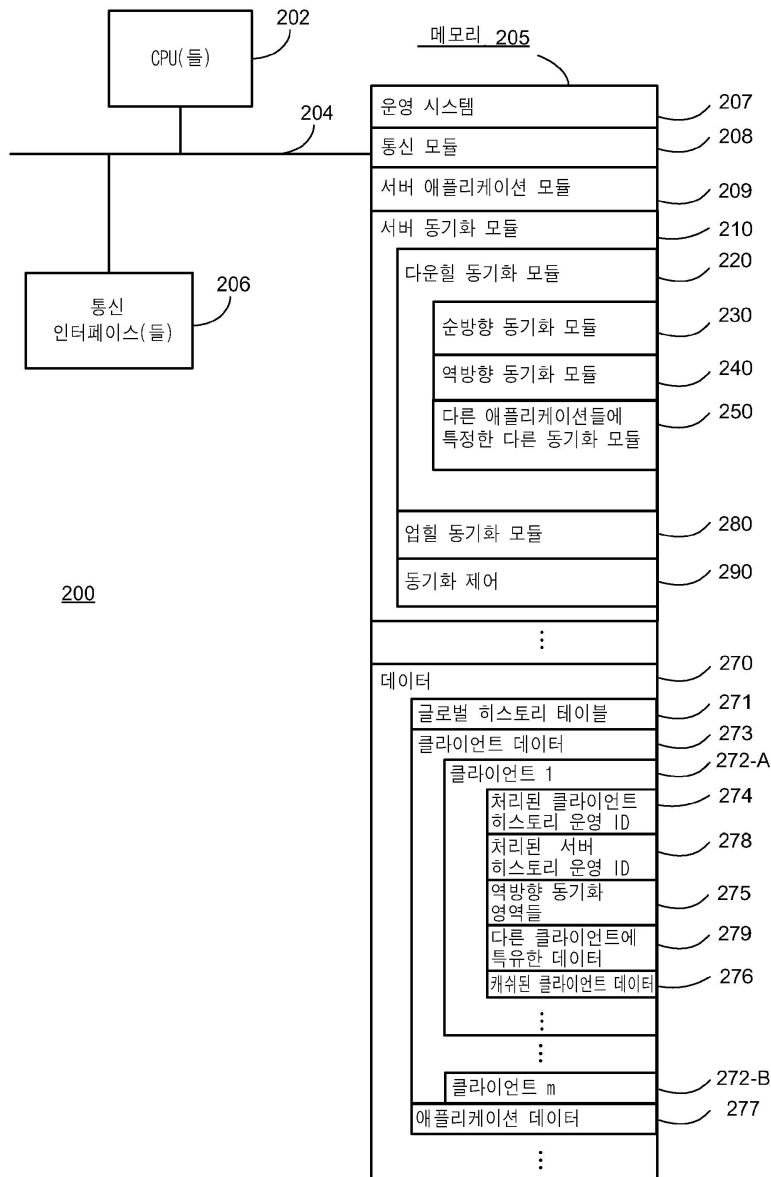
- [0136] 도 14B는 클라이언트에서 업힐 동기화의 예시적인 방법을 나타내는 흐름도이다.
- [0137] 개시에서 또는 긴 접속해제의 기간 이후에서, 클라이언트는 클라이언트 ID와 블록(1420)에서 클라이언트에 의해 인식되는 마지막 CHO ID와 함께 서버에 스타트 동기화 요청을 전송한다. 블록(1421)에서, 그 다음 클라이언트는 클라이언트에게 마지막 동기화가 서버에 대해 어디에서 끝났는지를 통지하기 위한 서버에 의해 처리된 마지막 CHO ID를 포함하는 서버로부터 스타트 동기화 응답을 수신한다. 블록(1422)에서, 메인 동기화 또는 데이터 동기화의 프로세스 동안에, 클라이언트는 클라이언트 ID와 'z' 비동기화된 데이터 변화들의 최대를 전송한다. 블록(1423)에서, 클라이언트는 클라이언트에 의해 더 일찍 전송된 다음 가장 높은 CHO ID 또는 서버에 의해 인식된 마지막 CHO ID를 포함하는 서버로부터 메인 동기화 응답을 수신할 것이다. 블록(1424)에서, 그 다음 클라이언트는 어떤 남아있는 비동기화된 데이터 변화들이 있는지를 결정한다. 만일 남아있는 변화들이 있다면, 클라이언트는 블록(1422)에서 나타난 바와 같이 클라이언트에 이 같은 비동기화된 데이터 변화들을 전송하는 것을 계속한다. 만일 더 이상 변화들이 없다면, 클라이언트는 블록(1425)에 도시된 바와 같이 미리정의된 기준에 따라 다음 메인 동기화의 발생까지 유효를 유지할 것이다. 만일 긴 접속해제가 있다면, 그 다음 클라이언트는 블록(1420)에 도시된 바와 같이 스타트 동기화 핸드셰이크와 함께 전체의 프로세스를 다시 개시하고, 그렇지 않다면, 블록(1422)에 도시된 바와 같이 메인 동기화를 재개한다.
- [0138] 도 14C는 서버에서 업힐 동기화의 예시적인 방법을 설명하는 다이어그램이다. 개시 또는 긴 접속해제의 기간 이후에, 서버는 블록(1431)에서 클라이언트 ID와 클라이언트에 의해 처리된 마지막 CHO ID를 포함하는 클라이언트로부터 스타트 동기화 요청을 수신한다. 그 다음 서버는 블록(1432)에 도시된 바와 같이 클라이언트가 다음 비동기화된 데이터 영역을 어디서 시작해야할 지를 결정하는 것을 돕기 위하여 클라이언트에서 서버상에 동기화 되거나 인식된 마지막 CHO ID를 포함하는 클라이언트에 스타트 동기화 응답을 전송한다. 블록(1433)에서, 서버는 클라이언트 ID와 'z' 데이터 변화의 최대 허용가능한 수와 함께 클라이언트로부터 메인 동기화 요청을 수신한다. 블록(1434)에서, 모든 데이터 변화들이 성공적으로 동기화되어 다면, 서버는 블록(1433)에서 도시된 바와 같이 클라이언트에 의해 제안된 대로 다음 가장 높은 비동기화된 CHO ID를 포함하는 클라이언트에 메인 동기화 응답을 전송할 것이고; 만일 데이터 동기화가 중단된다면, 서버에 의해 처리된 마지막 CHO ID가 대신에 전송될 것이다. 블록(1435)에서, 서버는 미리정의된 조건들 아래 클라이언트로부터 다음 메인 동기화 요청까지 유효를 유지한다. 만일 미리정의된 조건들에 만족하면서 수신된 메인 동기화 요청이 있다면, 메인 동기화는 블록(1433)에서 도시된 바와 같이 재개하고; 만일 없다면, 서버는 데이터 동기화 프로세스를 재개하기 위하여 블록(1431)에 도시된 바와 같이 클라이언트로부터 스타트 동기화 요청을 수신할 것이다.
- [0139] 설명을 목적으로, 앞서 말한 설명은 특정한 실시예들에 관련하여 설명되었다. 그러나, 상술한 실시예 설명들은 개시된 정확한 형태들에 발명을 제한하거나 완전하게 할 의도는 아니다. 많은 수정들과 다양성이 상술한 가르침들의 관점에서 가능하다. 실시예들은 발명의 원리들과 그것의 실제 적용들에 대한 최고의 설명을 위하여 선택되고 설명되며, 그렇게 함으로써 당업자에 의해 발명과 고려될 수 있는 특정한 용도에 적합하도록 다양한 수정을 가진 다양한 실시예들을 최고로 이용할 수 있다.
- [0140]
- [0141]
- [0142]
- [0143]

도면

도면1

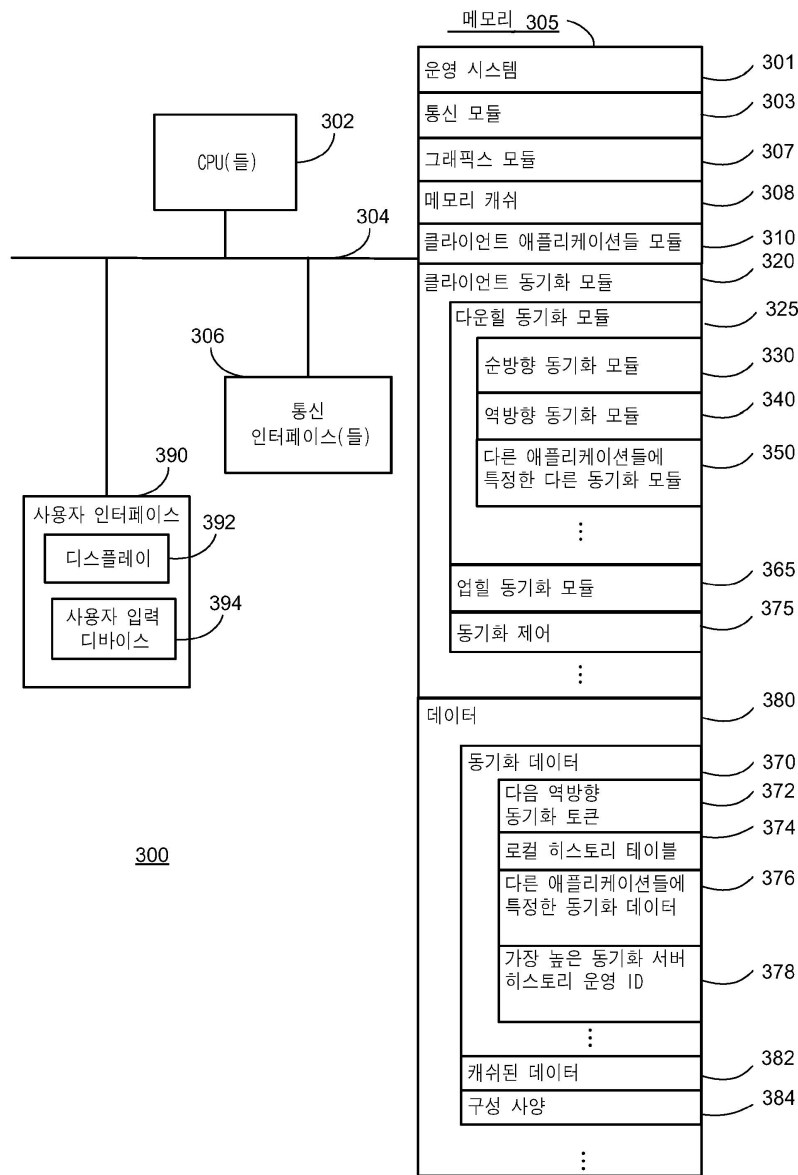


도면2

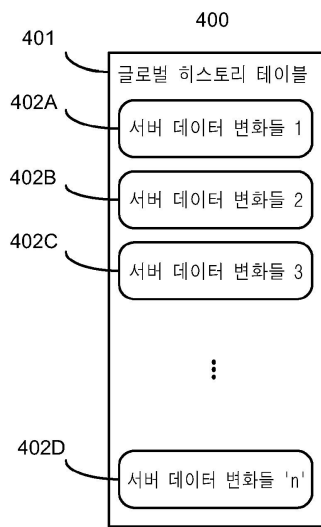




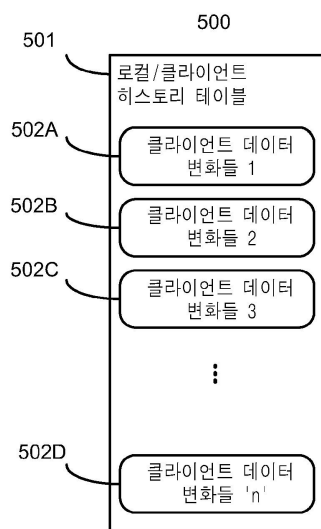
도면3



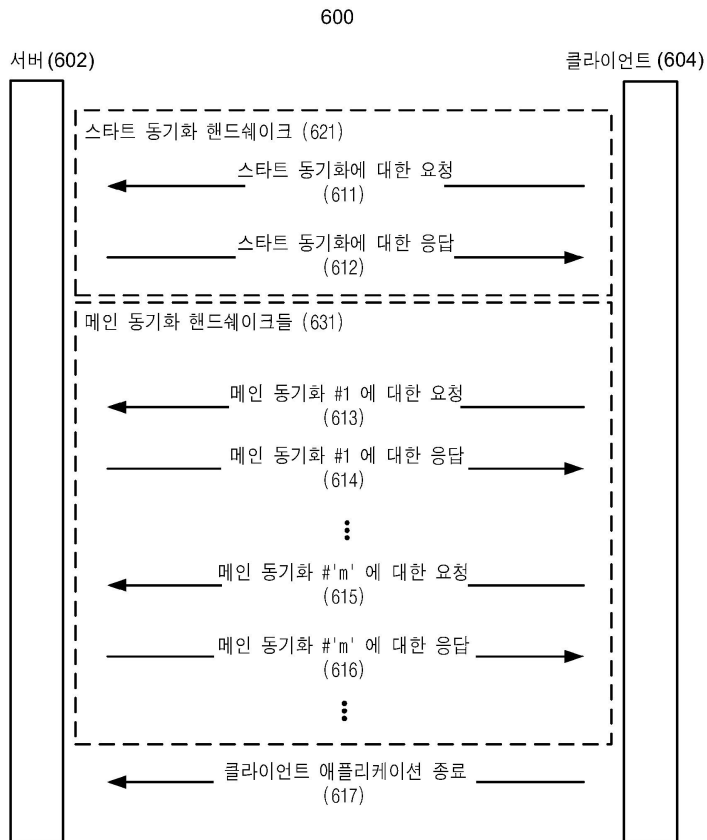
도면4



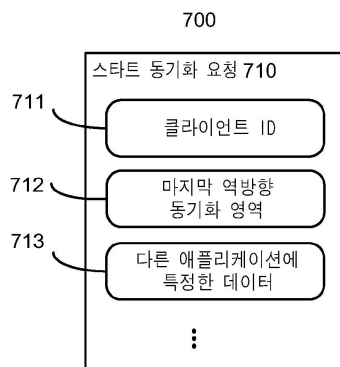
도면5



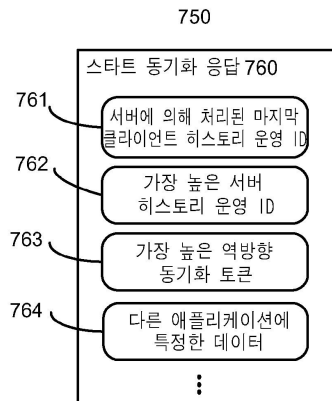
도면6



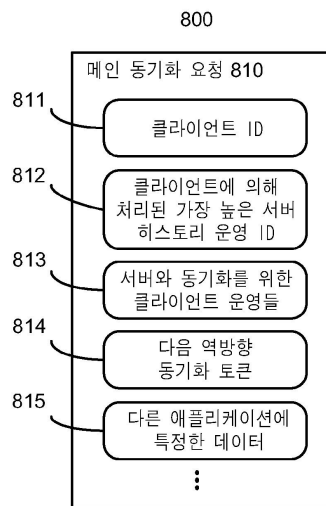
도면7a



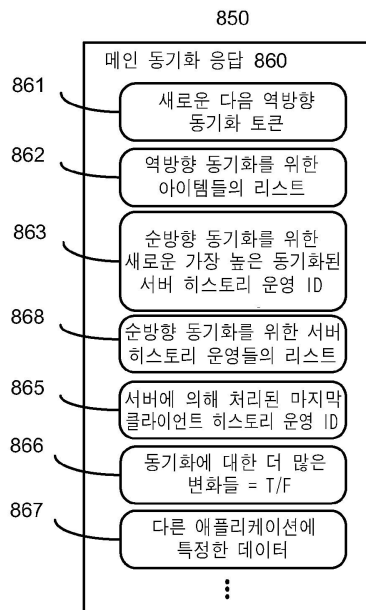
도면7b



도면8a

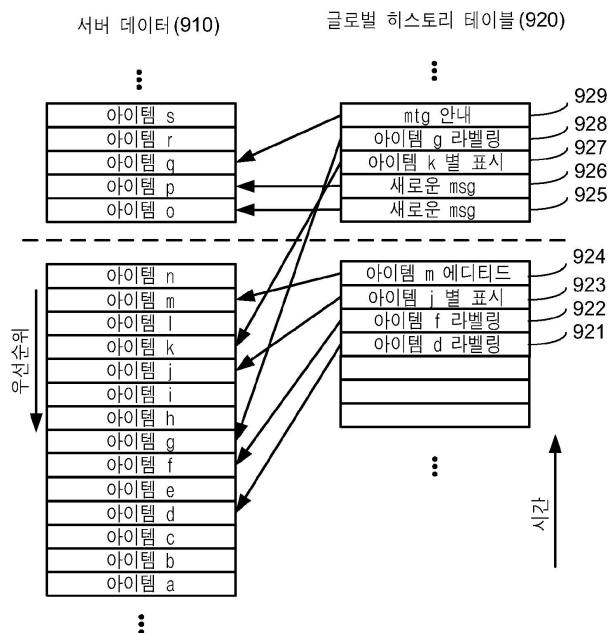


도면8b

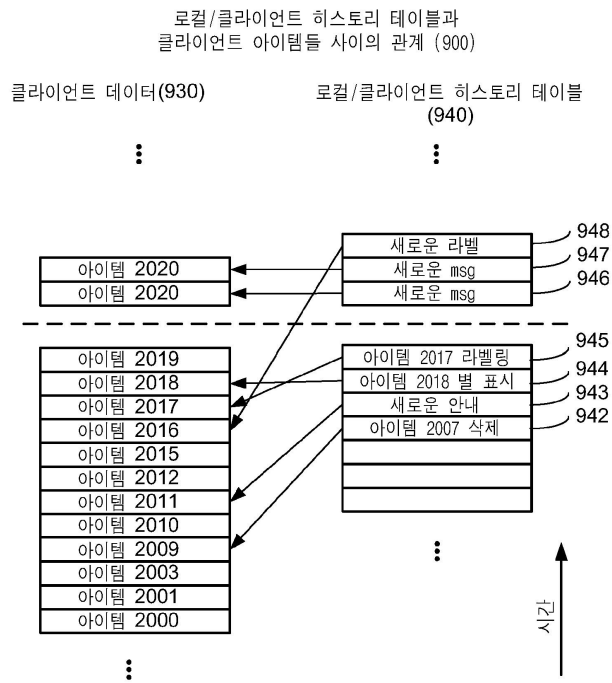


도면9a

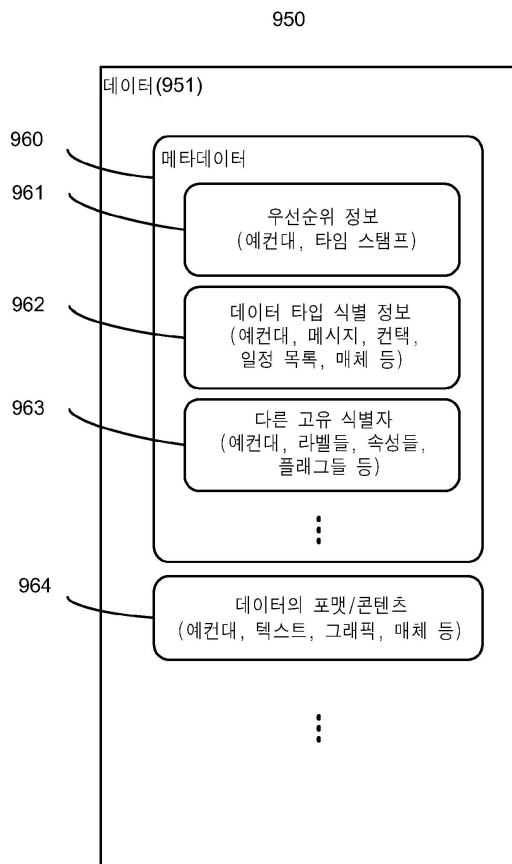
글로벌 히스토리 테이블과 서버  
아이템들 사이의 관계 (900)



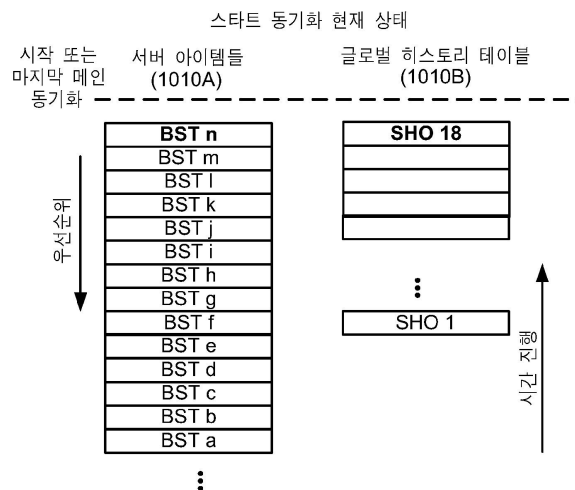
도면9b



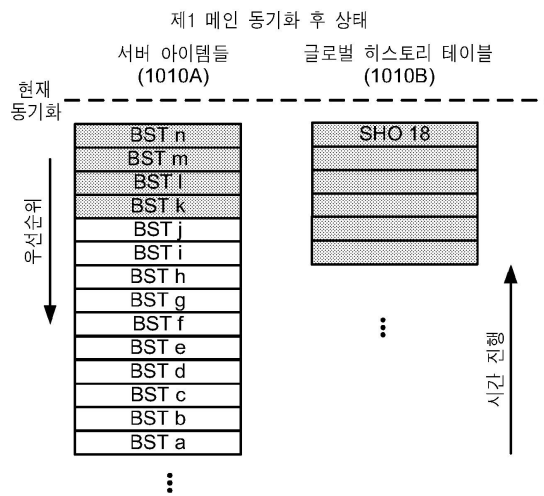
도면9c



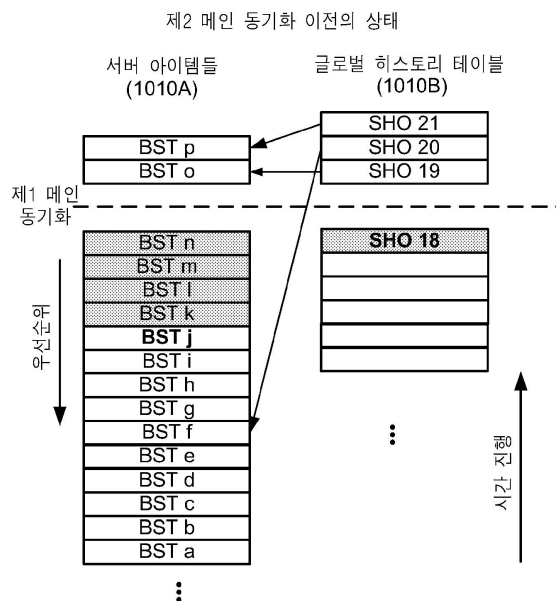
도면10a



도면10b

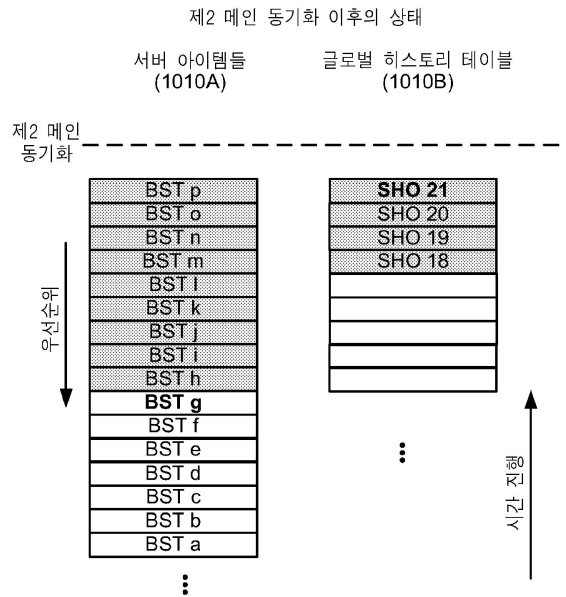


도면10c

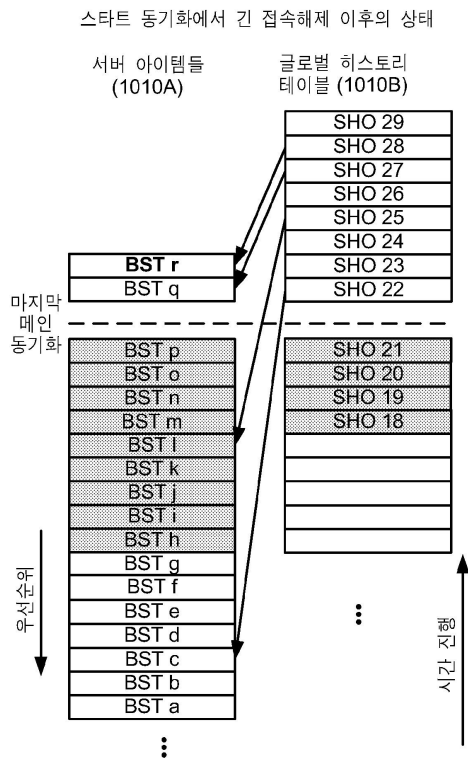




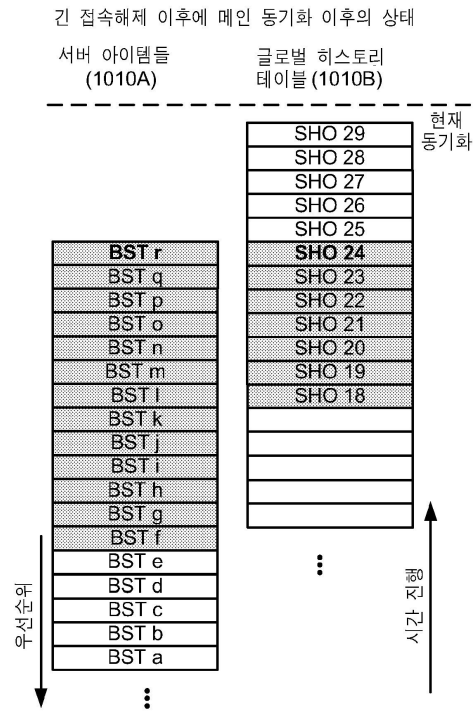
도면10d



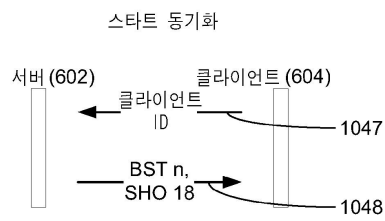
도면10e



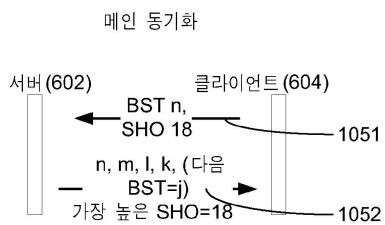
도면10f



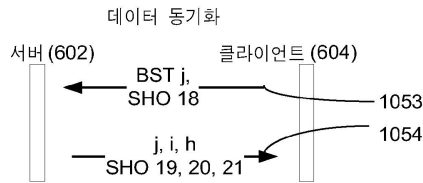
도면10w



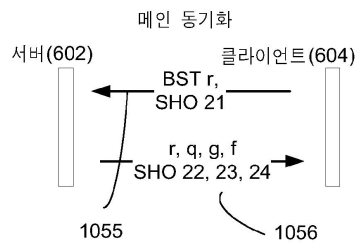
도면10x



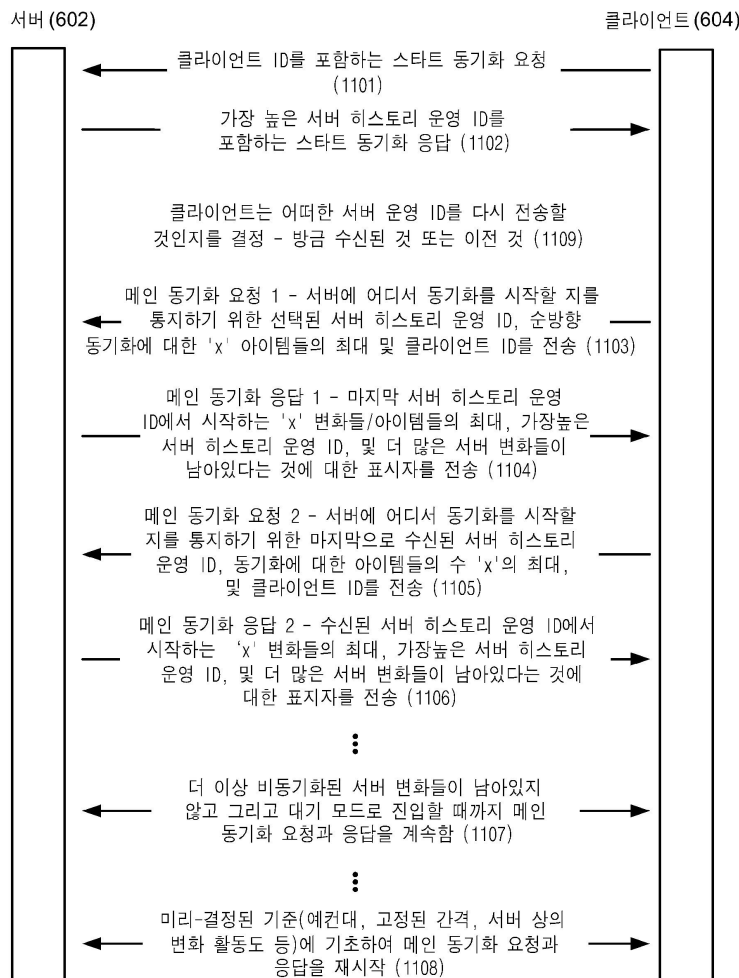
도면10y



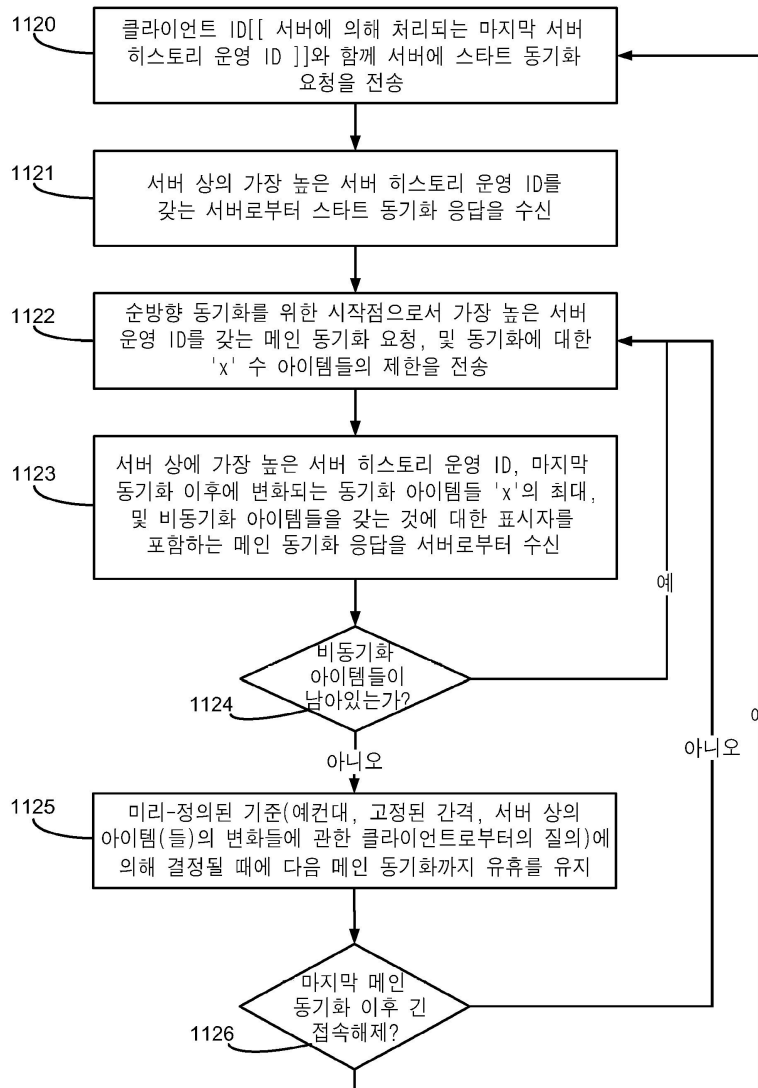
도면10z



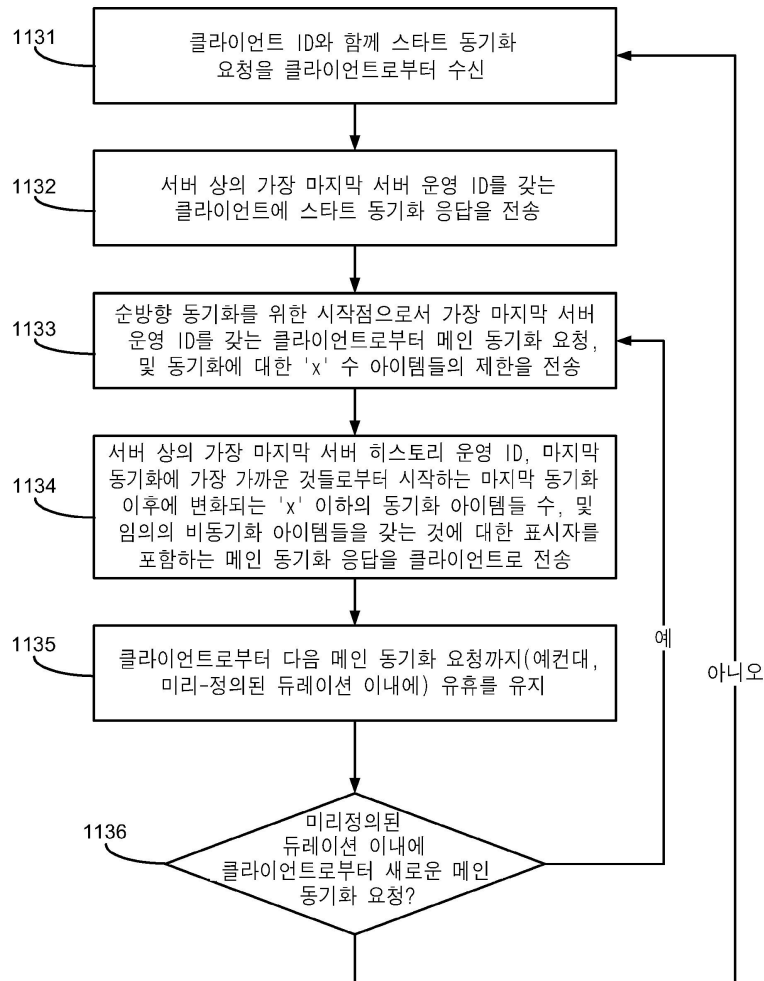
도면11a



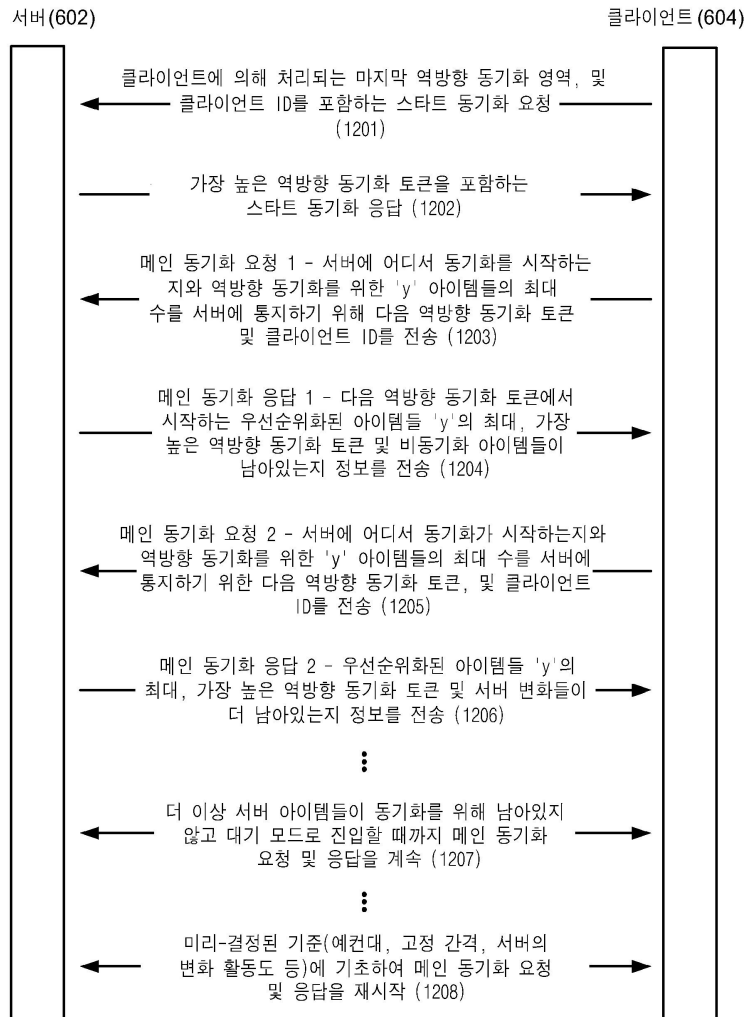
도면11b



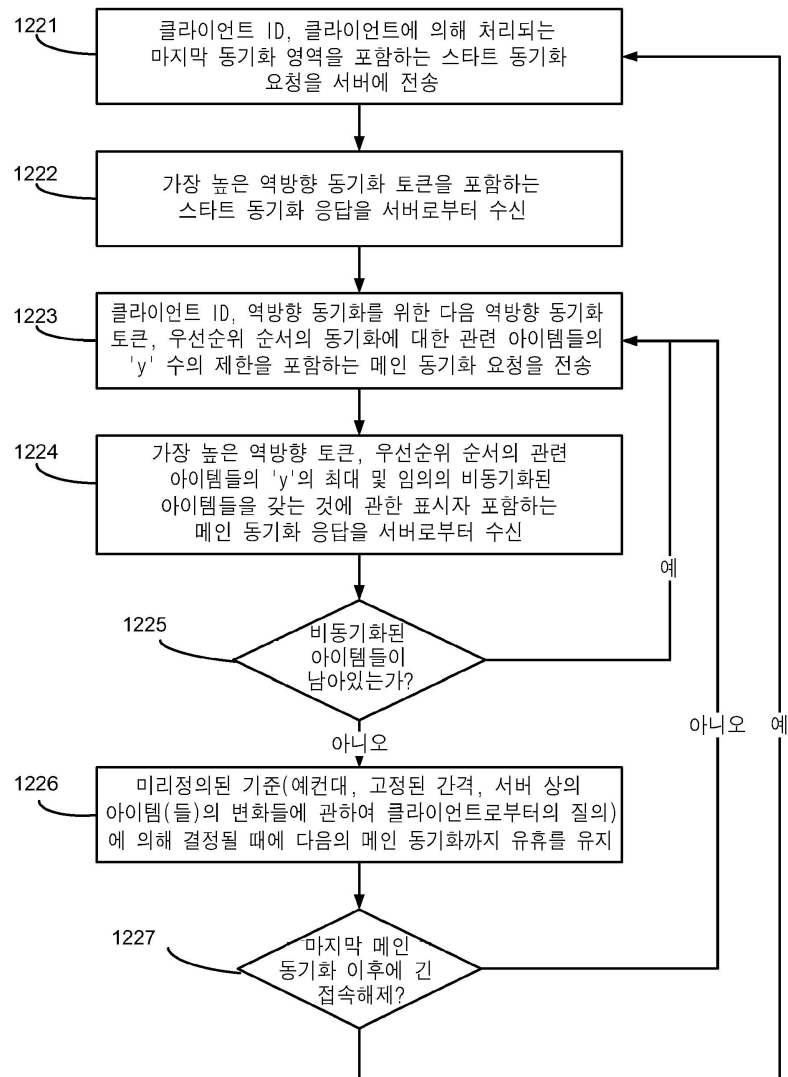
도면11c



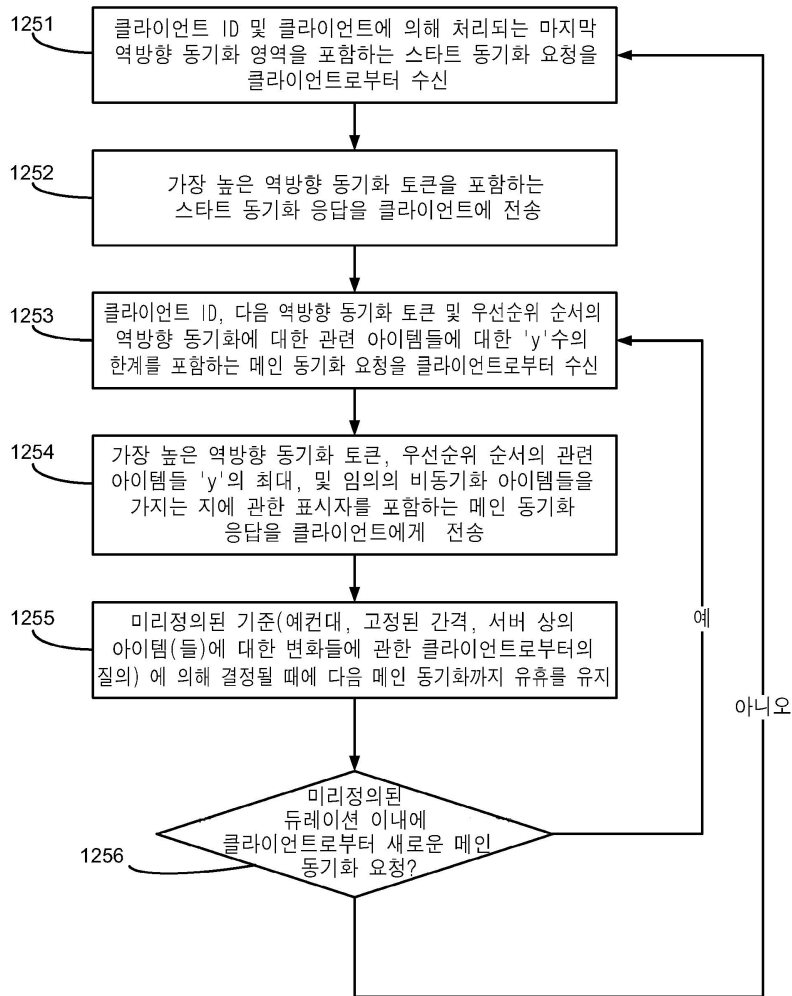
도면12a



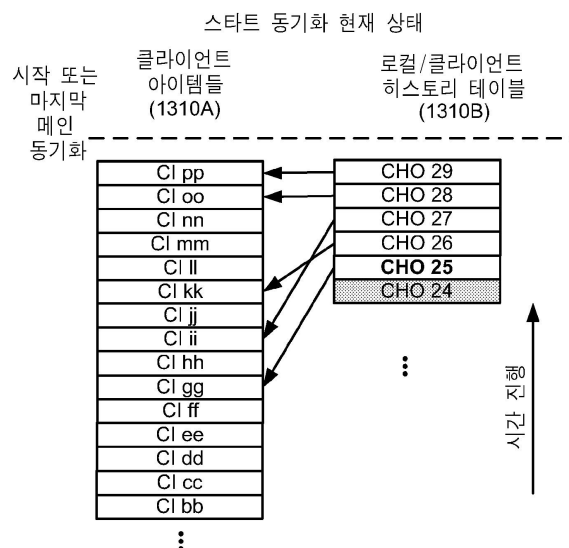
도면12b



도면12c

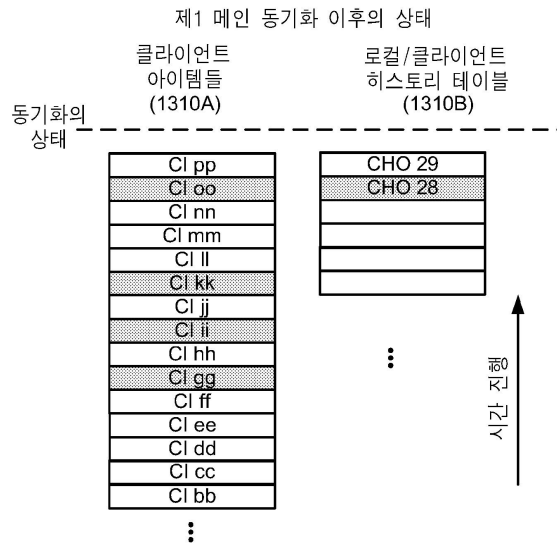


도면13a

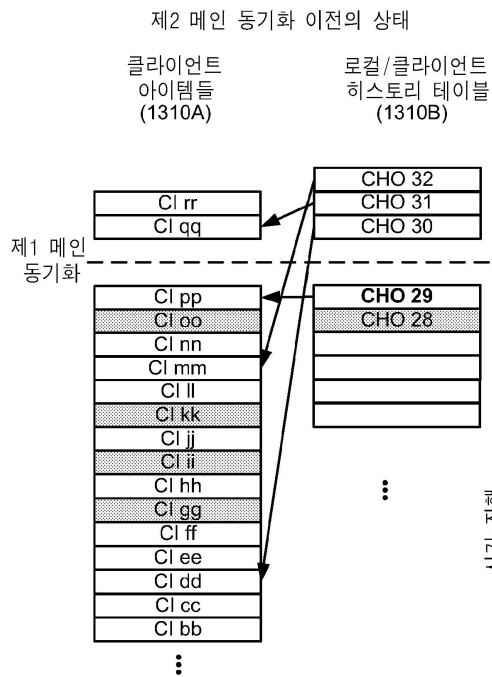




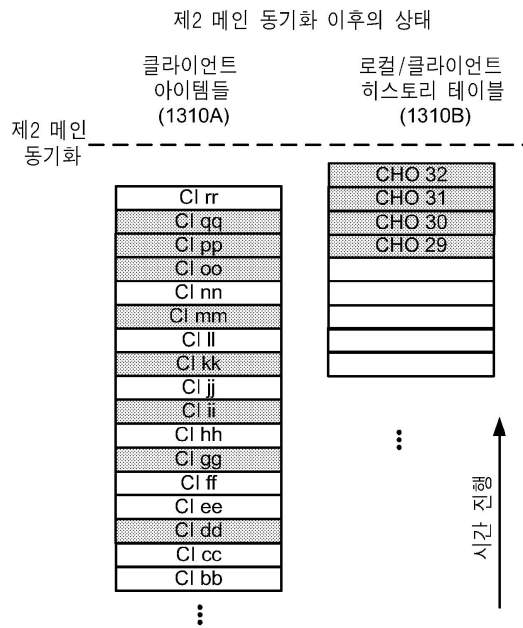
도면13b



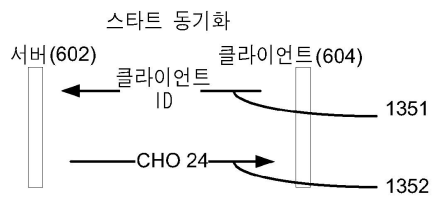
도면13c



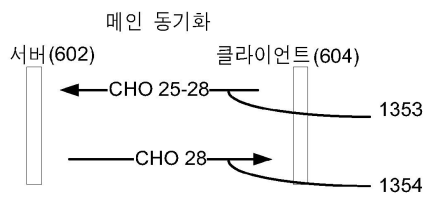
도면13d



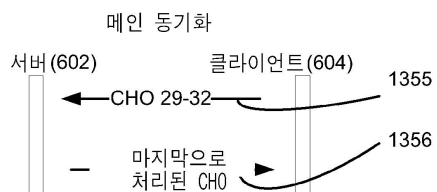
도면13x



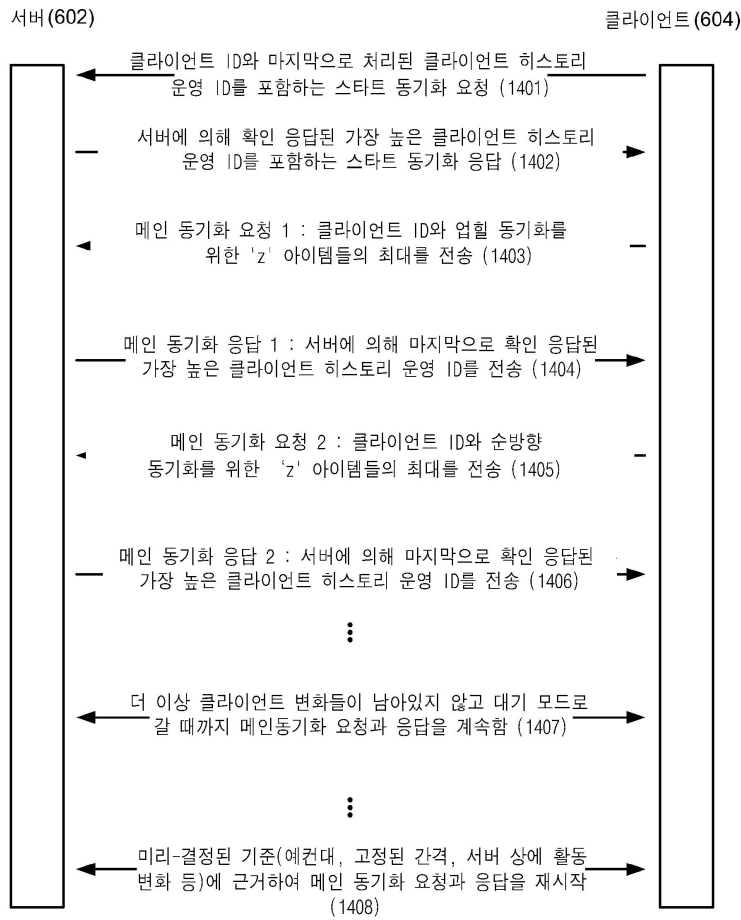
도면13y



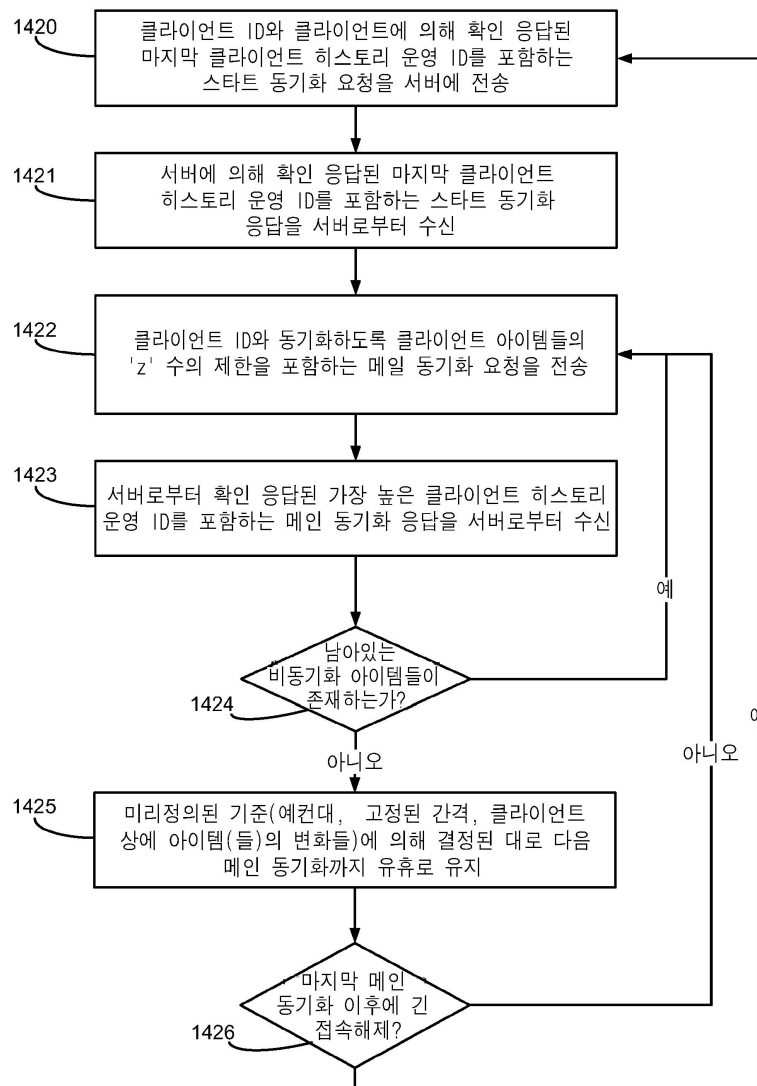
도면13z



도면14a



도면14b



도면14c

