



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2016년09월07일  
(11) 등록번호 10-1655497  
(24) 등록일자 2016년09월01일

(51) 국제특허분류(Int. Cl.)  
G06F 17/30 (2006.01) G06F 15/16 (2006.01)  
G06F 17/24 (2006.01)  
(21) 출원번호 10-2011-7028994  
(22) 출원일자(국제) 2010년06월01일  
심사청구일자 2015년05월12일  
(85) 번역문제출일자 2011년12월02일  
(65) 공개번호 10-2012-0017442  
(43) 공개일자 2012년02월28일  
(86) 국제출원번호 PCT/US2010/036968  
(87) 국제공개번호 WO 2010/141517  
국제공개일자 2010년12월09일  
(30) 우선권주장  
12/479,189 2009년06월05일 미국(US)  
(56) 선행기술조사문헌  
US20050223047 A1  
US2006/0206533 A1  
US20080098042 A1  
US20090006936 A1

(73) 특허권자  
마이크로소프트 테크놀로지 라이선싱, 엘엘씨  
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원  
마이크로소프트 웨이  
(72) 발명자  
클라크 사이먼 피터  
미국 워싱턴주 98052-6399 레드몬드 원 마이크로  
소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마  
이크로소프트 코포레이션  
보스 미코 아나브 에스  
미국 워싱턴주 98052-6399 레드몬드 원 마이크로  
소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마  
이크로소프트 코포레이션  
선 슈레이  
미국 워싱턴주 98052-6399 레드몬드 원 마이크로  
소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마  
이크로소프트 코포레이션  
(74) 대리인  
제일특허법인

전체 청구항 수 : 총 14 항

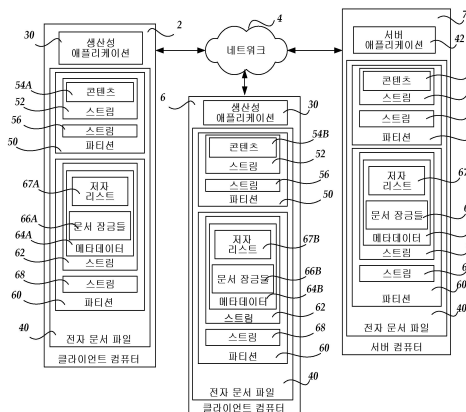
심사관 : 윤혜숙

(54) 발명의 명칭 서버 저장 모델을 이용하는 파일 파티션들의 동기화

(57) 요약

클라이언트-서버 컴퓨터 네트워크에서 서버 저장 모델을 이용하여 파일 파티션들을 동기화하기 위한 실시예들이 제공된다. 전자 문서의 콘텐츠에 대한 편집들이 클라이언트 컴퓨터에서 수신될 수 있다. 콘텐츠는 서버 컴퓨터 상에 저장된 파일의 제1 파티션 내에 포함될 수 있다. 제1 파티션은 전자 문서 콘텐츠를 포함하는 제1 스트림을 포함할 수 있다. 파일은 다수의 파티션을 포함할 수 있으며, 각각의 파티션은 하나 이상의 스트림을 포함할 수 있다. 클라이언트 컴퓨터는 파일의 제2 파티션 내에 메타데이터를 생성할 수 있다. 메타데이터는 제1 파티션 내의 전자 문서 콘텐츠에 대한 편집들과 연관될 수 있다. 이어서, 메타데이터를 저장하기 위해 제2 파티션이 서버 컴퓨터와 개별적으로 동기화될 수 있다. 이어서, 전자 문서 콘텐츠에 대해 행해진 편집들을 저장하기 위해 제1 파티션이 서버 컴퓨터와 개별적으로 동기화될 수 있다.

대표도



## 명세서

### 청구범위

#### 청구항 1

클라이언트-서버 컴퓨터 네트워크에서 서버 저장 모델을 이용하여 파일 파티션을 동기화하는 방법으로서,

클라이언트 컴퓨터에서 전자 문서의 콘텐츠에 대한 편집을 수신하는 단계 - 상기 콘텐츠는 파일 내의 제1 파티션의 제1 스트림을 포함하고, 상기 파일은 상기 제1 파티션 및 제2 파티션을 포함하며, 상기 제1 파티션 및 상기 제2 파티션은 각각 적어도 하나의 스트림을 포함함 - 와,

상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내에서 상기 제1 파티션 내의 상기 전자 문서 콘텐츠에 대한 편집과 관련된 메타데이터를 생성하는 단계와,

상기 서버 컴퓨터 상에 상기 파일의 상기 제2 파티션 내의 상기 메타데이터를 저장하기 위해 상기 클라이언트 컴퓨터 상의 상기 제2 파티션을 상기 서버 컴퓨터와 동기화시키는 단계와,

상기 서버 컴퓨터 상에 상기 파일의 제1 파티션 내의 상기 전자 문서 콘텐츠에 대해 행해진 편집을 저장하기 위해 상기 클라이언트 컴퓨터 상의 상기 제1 파티션을 상기 서버 컴퓨터와 동기화시키는 단계

를 포함하되,

상기 제2 파티션을 동기화시키는 단계는

상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내에서 생성된 메타데이터와 상기 서버 컴퓨터 상에 저장된 상기 파일의 상기 제2 파티션 내에 저장된 메타데이터를 비교하는 단계와,

상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내에서 생성된 메타데이터가 상기 서버 컴퓨터 상의 상기 파일의 상기 제2 파티션 내에 저장된 메타데이터와 다를 경우 충돌(conflict)을 판정하는 단계와,

상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내의 메타데이터가 상기 서버 컴퓨터 상에 저장된 상기 파일의 상기 제2 파티션 내에 저장된 메타데이터보다 더 최근 것일 경우, 상기 충돌을 해결하기 위해 상기 클라이언트 컴퓨터 상의 상기 제2 파티션을 상기 서버 컴퓨터와 자동으로 동기화시키는 단계와,

상기 서버 컴퓨터 상에 저장된 상기 파일의 상기 제2 파티션 내에 저장된 메타데이터가 상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내에 저장된 메타데이터보다 더 최근 것일 경우, 상기 충돌을 해결하기 위해 상기 클라이언트 컴퓨터 상의 상기 제2 파티션과 상기 서버 컴퓨터의 동기화가 자동으로 실패하게 하는 단계

를 포함하는 파일 파티션 동기화 방법.

#### 청구항 2

제1항에 있어서,

상기 서버 컴퓨터 상에 상기 파일의 제1 파티션 내의 상기 전자 문서 콘텐츠에 대해 행해진 상기 편집을 저장하기 위해 상기 클라이언트 컴퓨터 상의 상기 제1 파티션을 상기 서버 컴퓨터와 동기화시키는 단계는

상기 클라이언트 컴퓨터 상의 상기 제1 파티션 내의 상기 편집된 전자 문서 콘텐츠와 상기 서버 컴퓨터 상에 저장된 상기 파일의 상기 제1 파티션 내에 저장된 전자 문서 콘텐츠를 비교하는 단계와,

상기 서버 컴퓨터 상에 저장된 상기 파일의 상기 제1 파티션 내에 저장된 상기 전자 문서 콘텐츠가 상기 클라이언트 컴퓨터 상의 상기 제1 파티션 내의 상기 편집된 전자 문서 콘텐츠 내에 존재하지 않는 편집을 포함할 경우 충돌을 판정하는 단계

를 포함하는 파일 파티션 동기화 방법.

### 청구항 3

제2항에 있어서,

상기 클라이언트 컴퓨터와 상기 서버 컴퓨터 사이의 상기 제1 파티션의 동기화가 실패하게 하는 단계와,

상기 클라이언트 컴퓨터 상에서 충돌 에러 메시지를 생성하는 단계

를 더 포함하는 파일 파티션 동기화 방법.

### 청구항 4

제1항에 있어서,

상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내에서 상기 제1 파티션 내의 상기 전자 문서 콘텐츠에 대한 편집과 관련된 메타데이터를 생성하는 단계는 상기 제2 파티션의 스트림 내에 상기 전자 문서 콘텐츠에 대한 문서 저자의 리스트를 포함하는 메타데이터를 생성하는 단계를 포함하는

파일 파티션 동기화 방법.

### 청구항 5

클라이언트 컴퓨터 상에서 실행될 때 상기 클라이언트 컴퓨터로 하여금 서버 저장 모델을 이용하여 파일 파티션을 동기화하는 방법을 수행하게 하는 컴퓨터 실행 가능 명령어를 포함하는 컴퓨터 판독 가능 저장 매체로서,

상기 방법은

상기 클라이언트 컴퓨터에서 전자 문서의 콘텐츠에 대한 편집을 수신하는 단계 - 상기 콘텐츠는 파일 내의 제1 파티션의 제1 스트림을 포함하고, 상기 파일은 적어도 상기 제1 파티션 및 제2 파티션을 포함하며, 상기 제1 파티션 및 제2 파티션의 각각은 적어도 하나의 스트림을 포함함 - 와,

상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내에서 상기 제1 파티션 내의 상기 전자 문서 콘텐츠에 대한 편집과 관련된 메타데이터를 생성하는 단계와,

서버 컴퓨터 상에 상기 파일의 상기 제2 파티션 내의 상기 메타데이터를 저장하기 위해 상기 클라이언트 컴퓨터 상의 상기 제2 파티션을 상기 서버 컴퓨터와 동기화시키는 단계와,

상기 서버 컴퓨터 상에 상기 파일의 제1 파티션 내의 상기 전자 문서 콘텐츠에 대해 행해진 편집을 저장하기 위해 상기 클라이언트 컴퓨터 상의 상기 제1 파티션을 상기 서버 컴퓨터와 동기화시키는 단계

를 포함하되,

상기 제2 파티션을 동기화시키는 단계는

상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내에서 생성된 메타데이터와 상기 서버 컴퓨터 상에 저장된 상기 파일의 상기 제2 파티션 내에 저장된 메타데이터를 비교하는 단계와,

상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내에서 생성된 메타데이터가 상기 서버 컴퓨터 상의 상기 파일의 상기 제2 파티션 내에 저장된 메타데이터와 다를 경우 충돌을 판정하는 단계와,

상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내의 메타데이터가 상기 서버 컴퓨터 상에 저장된 상기 파일의 상기 제2 파티션 내에 저장된 메타데이터보다 더 최근 것일 경우, 상기 충돌을 해결하기 위하여 상기 클라이언트 컴퓨터 상의 상기 제2 파티션을 상기 서버 컴퓨터와 자동으로 동기화시키는 단계와,

상기 서버 컴퓨터 상에 저장된 상기 파일의 상기 제2 파티션 내에 저장된 메타데이터가 상기 클라이언트 컴퓨터 상의 상기 제2 파티션 내에 저장된 메타데이터보다 더 최근 것일 경우, 상기 충돌을 해결하기 위하여 상기 클라이언트 컴퓨터 상의 상기 제2 파티션과 상기 서버 컴퓨터의 동기화가 자동으로 실패하게 하는 단계

를 포함하는 컴퓨터 판독 가능 저장 매체.

## 청구항 6

제5항에 있어서,

상기 서버 컴퓨터 상에 상기 파일의 제1 파티션 내의 상기 전자 문서 콘텐츠에 대해 행해진 상기 편집을 저장하기 위해 상기 클라이언트 컴퓨터 상의 상기 제1 파티션을 상기 서버 컴퓨터와 동기화시키는 단계는

상기 클라이언트 컴퓨터 상의 상기 제1 파티션 내의 상기 편집된 전자 문서 콘텐츠와 상기 서버 컴퓨터 상에 저장된 상기 파일의 상기 제1 파티션 내에 저장된 편집된 전자 문서 콘텐츠를 비교하는 단계와,

상기 서버 컴퓨터 상에 저장된 상기 파일의 상기 제1 파티션 내에 저장된 상기 전자 문서 콘텐츠가 상기 클라이언트 컴퓨터 상의 상기 제1 파티션 내의 상기 편집된 전자 문서 콘텐츠 내에 존재하지 않는 편집을 포함할 경우 충돌을 판정하는 단계

를 포함하는 컴퓨터 판독 가능 저장 매체.

## 청구항 7

제5항에 있어서,

상기 클라이언트 컴퓨터 상의 상기 제2 파티션의 스트림 내에서 상기 제1 파티션 내의 상기 전자 문서 콘텐츠에 대한 편집과 관련된 메타데이터를 생성하는 단계는 상기 전자 문서 콘텐츠에 대한 문서 저자의 리스트를 포함하는 메타데이터를 생성하는 단계를 포함하는

컴퓨터 판독 가능 저장 매체.

## 청구항 8

클라이언트-서버 컴퓨터 네트워크에서 서버 저장 모델을 이용하여 파일 파티션을 동기화하는 방법으로서,

제1 클라이언트 컴퓨터에서, 전자 문서의 콘텐츠에 대한 편집을 수신하는 단계 - 상기 콘텐츠는 파일 내의 제1 파티션의 제1 스트림을 포함하고, 상기 파일은 적어도 상기 제1 파티션 및 제2 파티션을 포함하며, 상기 제1 파티션 및 상기 제2 파티션은 각각 적어도 하나의 스트림을 포함함 - 와,

상기 제1 클라이언트 컴퓨터 상의 상기 제2 파티션의 스트림 내에, 상기 제1 클라이언트 컴퓨터에서의 상기 제1 파티션 내의 상기 전자 문서 콘텐츠의 편집자를 식별하는 데이터를 생성하는 단계와,

서버 컴퓨터 상에 상기 데이터를 저장하기 위해 상기 제1 클라이언트 컴퓨터 상의 상기 제2 파티션을 상기 서버 컴퓨터와 동기화시키는 단계와,

제2 클라이언트 컴퓨터에서, 적어도 상기 제1 파티션 및 상기 제2 파티션을 포함하는 상기 서버 컴퓨터 상에 저장된 상기 파일을 수신하는 단계 - 상기 제2 파티션은, 상기 제1 클라이언트 컴퓨터에서의 상기 전자 문서 콘텐츠의 상기 편집자를 식별하고 상기 제1 파티션의 상기 전자 문서 콘텐츠가 편집을 위해 상기 제1 클라이언트 컴퓨터에 현재 다운로드됨을 식별하는 데이터를 포함함 - 와,

상기 제2 클라이언트 컴퓨터에서, 상기 제1 파티션 내의 상기 전자 문서 콘텐츠에 대한 편집을 수신하는 단계 - 상기 제2 클라이언트 컴퓨터에서 수신된 상기 전자 문서 콘텐츠에 대한 상기 편집은 상기 제1 클라이언트 컴퓨터에서 수신된 상기 전자 문서 콘텐츠에 대한 상기 편집을 포함하지 않음 - 와,

상기 서버 컴퓨터 상에 상기 전자 문서 콘텐츠에 대해 행해진 상기 편집을 저장하기 위해 상기 제1 클라이언트 컴퓨터 상의 상기 제1 파티션을 상기 서버 컴퓨터와 동기화시키는 단계

를 포함하는 파일 파티션 동기화 방법.

## 청구항 9

제8항에 있어서,

상기 제1 클라이언트 컴퓨터 상의 상기 제2 파티션의 스트림 내에 상기 제1 클라이언트 컴퓨터에서의 상기 제1 파티션 내의 상기 전자 문서 콘텐츠의 상기 편집자를 식별하는 데이터를 생성하는 단계는 상기 제1 클라이언트 컴퓨터 상의 제2 파티션의 스트림 내에 상기 제1 클라이언트 컴퓨터에서의 상기 제1 파티션 내의 상기 전자 문서 콘텐츠의 상기 편집자를 식별하는 메타데이터를 생성하는 단계를 포함하는

파일 파티션 동기화 방법.

## 청구항 10

제9항에 있어서,

상기 서버 컴퓨터 상에 상기 데이터를 저장하기 위해 상기 제1 클라이언트 컴퓨터 상의 상기 제2 파티션을 상기 서버 컴퓨터와 동기화시키는 단계는 상기 서버 컴퓨터 상에 상기 메타데이터를 저장하기 위해 상기 제1 클라이언트 컴퓨터 상의 상기 제2 파티션을 상기 서버 컴퓨터와 동기화시키는 단계를 포함하는

파일 파티션 동기화 방법.

## 청구항 11

제10항에 있어서,

상기 제1 클라이언트 컴퓨터에서의 상기 전자 문서 콘텐츠의 상기 편집자를 식별하는 상기 메타데이터를 상기 제2 클라이언트 컴퓨터에서의 상기 전자 문서 콘텐츠의 상기 편집자의 식별로 갱신하기 위해 상기 제2 클라이언트 컴퓨터 상의 상기 제2 파티션을 상기 서버 컴퓨터와 동기화시키는 단계를 더 포함하는

파일 파티션 동기화 방법.

## 청구항 12

제8항에 있어서,

상기 서버 컴퓨터 상의 상기 제1 파티션에 대해, 상기 제2 클라이언트 컴퓨터에서의 상기 전자 문서 콘텐츠에 대해 행해진 상기 편집을 상기 제1 클라이언트 컴퓨터에서의 상기 전자 문서에 대해 행해진 상기 편집에 추가하기 위해 상기 제2 클라이언트 컴퓨터 상의 상기 제1 파티션을 상기 서버 컴퓨터와 동기화시키는 단계를 더 포함하는

파일 파티션 동기화 방법.

## 청구항 13

제12항에 있어서,

상기 서버 컴퓨터 상의 상기 제1 파티션에 대해, 상기 제2 클라이언트 컴퓨터에서의 상기 전자 문서 콘텐츠에 대해 행해진 상기 편집을 상기 제1 클라이언트 컴퓨터에서의 상기 전자 문서에 대해 행해진 상기 편집에 추가하기 위해 상기 제2 클라이언트 컴퓨터 상의 상기 제1 파티션을 상기 서버 컴퓨터와 동기화시키는 단계는

상기 제1 파티션에 대한 업데이트에 대해 상기 서버 컴퓨터를 검사하는 단계와,

상기 서버 컴퓨터로부터 상기 제1 파티션이 업데이트되었다는 확인을 수신하는 단계

를 포함하는 파일 파티션 동기화 방법.

#### 청구항 14

제12항에 있어서,

상기 서버 컴퓨터 상의 상기 제1 파티션에 대해, 상기 제2 클라이언트 컴퓨터에서의 상기 전자 문서 콘텐츠에 대해 행해진 상기 편집을 상기 제1 클라이언트 컴퓨터에서의 상기 전자 문서에 대해 행해진 상기 편집에 추가하기 위해 상기 제2 클라이언트 컴퓨터 상의 상기 제1 파티션을 상기 서버 컴퓨터와 동기화시키는 단계는

상기 서버 컴퓨터로부터, 상기 제1 클라이언트 컴퓨터에서의 상기 전자 문서 콘텐츠에 대해 행해진 상기 편집을 수신하는 단계와,

상기 서버 컴퓨터 상의 상기 제1 파티션을 상기 제2 클라이언트 컴퓨터 및 상기 제1 클라이언트 컴퓨터에서의 상기 전자 문서 콘텐츠에 대해 행해진 편집으로 갱신하는 단계

를 더 포함하는 파일 파티션 동기화 방법.

#### 청구항 15

삭제

### 발명의 설명

#### 배경 기술

[0001] 클라이언트-서버 컴퓨터 네트워크들에서, 전자 문서 파일은 상이한 클라이언트 컴퓨터들에서 다수의 사용자에게 의해 편집될 수 있다. 전자 문서 파일은 주요 문서 파일 콘텐츠와 다르며 독립적인 관련 편집 세션 데이터(예를 들어, 메타데이터)를 포함할 수 있다. 전자 문서 파일은 네트워크 내의 다른 클라이언트 컴퓨터들의 사용자들이 다음에 볼 수 있는 전자 문서 파일에 대해 행해진 임의의 편집들을 저장하기 위하여 클라이언트 컴퓨터와 중앙 서버 사이에서 동기화된다. 기존의 동기화 방법들에서는, 임의의 관련 메타데이터가 전자 문서 파일에 대해 행해진 편집들과 함께 동기화되어야 한다. 특히, 마크업 언어 또는 객체 연결 및 삽입("OLE") 복합 문서들의 경우에는, 동기화를 위해 메타데이터가 확장형 파일 포맷 내에 넣어질 수 있다. 동기화의 결과로서, 전자 문서 파일을 동기화하는 사용자가 다른 사용자에게 의해 메타데이터가 보여지거나 (전자 문서 파일 내에) 다운로드되는 것을 원하지 않는 경우에도, 전자 문서 파일과 관련된 메타데이터가 서버 상에 자동으로 노출된다. 본 발명의 실시예들이 창출된 것은 이러한 고려 사항 및 다른 고려 사항들에 관련된 것이다.

### 발명의 내용

#### 과제의 해결 수단

[0002] 본 요약은 아래에 발명을 실시하기 위한 구체적인 내용에서 더 설명되는 개념들의 발체를 간단한 형태로 소개하기 위해 제공된다. 본 요약은 청구 발명의 중요한 특징들 또는 필수적인 특징들을 식별하도록 의도되지 않으며, 청구 발명의 범위를 결정하는 데 있어서의 보조로서도 의도되지 않는다.

[0003] 클라이언트-서버 컴퓨터 네트워크에서 서버 저장 모델을 이용하여 파일 파티션들을 동기화하기 위한 실시예들이 제공된다. 전자 문서의 콘텐츠에 대한 편집들이 클라이언트 컴퓨터에서 수신될 수 있다. 콘텐츠는 서버 컴퓨터 상에 저장된 파일의 제1 파티션 내에 포함될 수 있다. 제1 파티션은 전자 문서 콘텐츠를 포함하는 제1 스트림을 포함할 수 있다. 파일은 다수의 파티션을 포함할 수 있으며, 각각의 파티션은 하나 이상의 스트림을 포함할 수 있다. 클라이언트 컴퓨터는 파일의 제2 파티션 내에 메타데이터를 생성할 수 있다. 메타데이터는 제1 파티션 내의 전자 문서 콘텐츠에 대한 편집들과 연관될 수 있다. 이어서, 메타데이터를 저장하기 위해 제2 파티션이 서버 컴퓨터와 개별적으로 동기화될 수 있다. 이어서, 전자 문서 콘텐츠에 대해 행해진 편집들을 저장하기 위해 제1 파티션이 서버 컴퓨터와 개별적으로 동기화될 수 있다.

[0004] 이들 및 다른 특징들 및 이점들은 아래의 상세한 설명을 읽고 관련 도면들을 검토할 때 명확할 것이다. 위의

일반 설명 및 아래의 상세한 설명 모두는 예시적인 것, 청구되는 바와 같은 본 발명을 한정하지 않는다.

### 도면의 간단한 설명

- [0005] 도 1은 다양한 실시예들에 따른, 파일 파티션들을 동기화하기 위해 서버 저장 모델을 이용할 수 있는 클라이언트-서버 네트워크 아키텍처를 나타내는 블록도이다.
- 도 2는 다양한 실시예들에 따른, 파일 파티션들을 동기화하기 위해 서버 저장 모델을 이용할 수 있는 클라이언트 컴퓨팅 환경을 나타내는 블록도이다.
- 도 3은 일 실시예에 따른, 서버 저장 모델을 이용하여 파일 파티션들을 동기화하기 위한 루틴을 나타내는 흐름도이다.
- 도 4는 도 3에 도시된 실시예에 따른, 서버 저장 모델을 이용하여 단일 파일 파티션을 동기화하기 위한 루틴을 나타내는 흐름도이다.
- 도 5는 도 3에 도시된 실시예에 따른, 서버 저장 모델을 이용하여 다른 파일 파티션을 동기화하기 위한 루틴을 나타내는 흐름도이다.
- 도 6은 다른 실시예에 따른, 서버 저장 모델을 이용하여 파일 파티션들을 동기화하기 위한 루틴을 나타내는 흐름도이다.
- 도 7은 도 6에 도시된 실시예에 따른, 서버 저장 모델을 이용하여 제2 클라이언트 컴퓨터 상의 파일 파티션을 동기화하기 위한 루틴을 나타내는 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

- [0006] 클라이언트-서버 컴퓨터 네트워크에서 서버 저장 모델을 이용하여 파일 파티션들을 동기화하기 위한 실시예들이 제공된다. 전자 문서의 콘텐츠에 대한 편집들이 클라이언트 컴퓨터에서 수신될 수 있다. 콘텐츠는 서버 컴퓨터 상에 저장된 파일의 제1 파티션 내에 포함될 수 있다. 제1 파티션은 전자 문서 콘텐츠를 포함하는 제1 스트림을 포함할 수 있다. 파일은 다수의 파티션을 포함할 수 있고, 각각의 파티션은 하나 이상의 스트림을 포함할 수 있다. 클라이언트 컴퓨터는 파일의 제2 파티션 내에 메타데이터를 생성할 수 있다. 메타데이터는 제1 파티션 내의 전자 문서 콘텐츠에 대한 편집들과 연관될 수 있다. 이어서, 메타데이터를 저장하기 위해 제2 파티션이 서버 컴퓨터와 개별적으로 동기화될 수 있다. 이어서, 전자 문서 콘텐츠에 대해 행해진 편집들을 저장하기 위해 제1 파티션이 서버 컴퓨터와 개별적으로 동기화될 수 있다.
- [0007] 아래의 상세한 설명에서는, 그 일부를 형성하고, 특정 실시예들 또는 사례들을 예시적으로 도시하는 첨부 도면들을 참조한다. 이들 실시예는 결합될 수 있고, 다른 실시예들이 이용될 수 있으며, 본 발명의 사상 또는 범위를 벗어나지 않고 구조적인 변경들이 이루어질 수 있다. 따라서, 아래의 상세한 설명은 제한적인 의미로 간주되지 않아야 하며, 본 발명의 범위는 첨부된 청구항들 및 그들의 균등물들에 의해 정해진다.
- [0008] 이제, 여러 도면들을 통해 동일한 번호들이 동일한 요소들을 나타내는 도면들을 참조하여 본 발명의 다양한 양태들을 설명한다. 도 1은 다양한 실시예들에 따른, 파일 파티션들을 동기화하기 위해 서버 저장 모델을 이용할 수 있는 클라이언트-서버 네트워크 아키텍처를 나타내는 블록도이다. 네트워크 아키텍처는 클라이언트 컴퓨터(2) 및 클라이언트 컴퓨터(6)를 포함하며, 이들 양자는 네트워크(4)를 통해 서버(70)와 통신한다. 네트워크(4)는 근거리 네트워크 또는 광역 네트워크(예를 들어, 인터넷)를 포함할 수 있다.
- [0009] 클라이언트 컴퓨터(2)는 (또한 클라이언트 컴퓨터(2) 상에 저장된) 전자 문서 파일(40)을 편집하는 데 사용될 수 있는 생산성 애플리케이션(30)을 저장할 수 있다. 다양한 실시예들에 따르면, 생산성 애플리케이션(30)은 워싱턴 레드몬드의 마이크로소프트사로부터의 WORD 워드 프로세싱 소프트웨어, POWERPOINT 프레젠테이션 그래픽 프로그램 및 GROOVE 소프트웨어를 포함할 수 있다. 그러나, 다른 제조자들로부터의 다른 생산성 또는 애플리케이션 프로그램들도 본 명세서에서 설명되는 다양한 실시예들에 따라 이용될 수 있다는 것을 알아야 한다.
- [0010] 전자 문서 파일(40)은 서버 컴퓨터(70)로부터의 네트워크(4)를 통한 다운로드 동작을 통해 클라이언트 컴퓨터(2)에서 수신되거나, 클라이언트 컴퓨터(2) 상의 생산성 애플리케이션(30)에 의해 생성될 수 있다. 전자 문서 파일(40)은 파티션(50)을 포함할 수 있다. 파티션(50)은 스트림들(52, 56)을 포함할 수 있다. 스트림(52)은 콘텐츠(즉, 전자 문서 콘텐츠)(54A)를 더 포함할 수 있다. 전자 문서 파일(40)은 파티션(60)도 포함할 수

있다. 파티션(60)은 스트림들(62, 68)을 포함할 수 있다. 스트림(62)은 메타데이터(64A)를 더 포함할 수 있다. 일 실시예에 따르면, 메타데이터(64A)는 파티션(50) 내에 저장된 콘텐츠(54A)와 연관될 수 있다. 특히, 메타데이터(64A)는 콘텐츠(54A)의 하나 이상의 저자를 기술하는 저자 리스트(67A) 및 전자 문서 파일(40)의 현재 사용자(예를 들어, 콘텐츠(54A)를 현재 편집하고 있는 사용자)를 기술하는 문서 잠금들(66A)을 포함할 수 있다.

[0011] 다양한 실시예들에 따르면, "파티션"은 (전자 문서 파일(40)과 같은) 파일 내의 "파일"로서 정의되며, 하나 이상의 스트림을 포함할 수 있다. 파티션 내에 포함되는 "스트림"은 (콘텐츠(54A) 또는 메타데이터(64A)와 같은) 전자 문서 파일과 관련된 데이터를 포함한다. 파일 내의 파티션들 각각은 서로 독립적으로 그리고 상이한 동기화 레이트들로 동기화될 수 있다는 것을 이해해야 한다(즉, 단일 파일 내의 파티션들은 상이한 시각들에 동기화될 수 있다). 더구나, 동일 파티션 내의 다수의 스트림은 서로 관련되지만, 상이한 파티션들 내의 스트림들 사이에는 관계가 존재하지 않을 수 있다. 예컨대, 스트림(56)은 전자 문서 콘텐츠와 연관된(따라서, 스트림(52)과 관련된) 데이터를 포함하지만, 전자 문서 파일(40)과 연관된 메타데이터를 포함하지 않을 수 있다. 게다가, 주어진 파티션 내의 다수의 스트림은 동시에 동기화되며, 따라서 동일 파티션 내의 스트림들 사이의 일관성이 보증된다.

[0012] 클라이언트 컴퓨터(6)는 (또한 클라이언트 컴퓨터(6)에 저장된) 전자 문서 파일(40)을 편집하는 데에도 사용될 수 있는 생산성 애플리케이션(30)을 저장할 수 있다. 전자 문서 파일(40)은 서버 컴퓨터(70)로부터의 네트워크(4)를 통한 다운로드 동작을 통해 클라이언트 컴퓨터(6)에서 수신되거나, 클라이언트 컴퓨터(6) 상의 생산성 애플리케이션(30)에 의해 생성될 수 있다. 전자 문서 파일(40)은 파티션(50)을 포함할 수 있다. 파티션(50)은 스트림들(52, 56)을 포함할 수 있다. 스트림(52)은 콘텐츠(즉, 전자 문서 콘텐츠)(54B)를 더 포함할 수 있다. 전자 문서 파일(40)은 파티션(60)도 포함할 수 있다. 파티션(60)은 스트림들(62, 68)을 포함할 수 있다. 스트림(62)은 메타데이터(64B)를 더 포함할 수 있다. 다양한 실시예에 따르면, "메타데이터"는 파티션 내에 스트림과 함께 포함된 콘텐츠를 나타낸다는 것을 이해해야 한다. 일 실시예에서, 메타데이터(64B)는 파티션(50) 내에 저장된 콘텐츠(54B)와 연관될 수 있다. 특히, 메타데이터(64B)는 콘텐츠(54B)의 하나 이상의 저자를 기술하는 저자 리스트(67B) 및 전자 문서 파일(40)의 현재 사용자(예를 들어, 콘텐츠(54B)를 현재 편집하고 있는 사용자)를 기술하는 문서 잠금들(66B)을 포함할 수 있다.

[0013] 서버 컴퓨터(70)는 서버 애플리케이션(42)을 저장할 수 있다. 다양한 실시예에 따르면, 서버 컴퓨터(70)는 클라이언트 컴퓨터들(2, 6) 상에서 생성된 전자 파일들에 대한 문서 저장 시스템으로서 기능할 수 있다. 서버 애플리케이션(42)은 마이크로소프트사로부터의 SHAREPOINT SERVER 서비스 애플리케이션과 같은 협력 서비스 애플리케이션 프로그램을 포함할 수 있다. 이 분야의 기술자들에게 공지된 바와 같이, SHAREPOINT 서비스 기술은 사용자들로 하여금 정보를 공유하기 위한 협력 환경을 생성하고, 유지하고, 제공할 수 있게 해준다. 이 기술을 이용하여, 사용자 또는 조직은 하나 이상의 웹사이트를 생성하여, 웹사이트들과 관련된 다른 사용자들을 위해 정보(예컨대, 웹 서버 또는 웹 폴더 상의 문서 등)를 제공하고 공유할 수 있다. 본 명세서에서 설명되는 실시예들은 SHAREPOINT 서비스 기술로 한정되는 것으로 해석되지 않아야 하며, 다른 개발자들 및/또는 제조자들로부터의 다른 협력 서비스 기술도 이용될 수 있다는 것을 이해해야 한다.

[0014] 서버 컴퓨터(70)는 전자 문서 파일(40)도 저장할 수 있다. 전자 문서 파일(40)은 파티션(50)을 포함할 수 있다. 파티션(50)은 스트림들(52, 56)을 포함할 수 있다. 스트림(52)은 콘텐츠(즉, 전자 문서 콘텐츠)(54)를 더 포함할 수 있다. 전자 문서 파일(40)은 파티션(60)도 포함할 수 있다. 파티션(60)은 스트림들(62, 68)을 포함할 수 있다. 스트림(62)은 메타데이터(64)를 더 포함할 수 있다. 일 실시예에 따르면, 메타데이터(64)는 파티션(50)에 저장된 콘텐츠(54)와 연관될 수 있다. 특히, 메타데이터(64)는 콘텐츠(54)의 하나 이상의 저자를 기술하는 저자 리스트(67) 및 전자 문서 파일(40)의 현재 사용자(예를 들어, 콘텐츠(54)를 현재 편집하고 있는 사용자)를 기술하는 문서 잠금들(66)을 포함할 수 있다. 일 실시예에 따르면, 전자 문서 파일(40)은 편집을 위해 서버 컴퓨터(70)로부터 클라이언트 컴퓨터들(2, 6)로 다운로드될 수 있으며, 따라서 서버 컴퓨터(70)에 다시 동기화되기 전에 변경들이 행해질 때 전자 문서 파일(40) 내의 콘텐츠(54) 및 메타데이터(64)의 상이한 버전들이 잠재적으로 생성될 수 있다는 것을 이해해야 한다. 따라서, 본 상세한 설명에서, 클라이언트 컴퓨터들(2, 6) 상의 콘텐츠(54) 및 메타데이터(64)의 버전들은 이러한 파일 컴포넌트들에 대한 참조 번호들 뒤에 부가된 문자 "A" 및 "B"에 의해 식별된다.

[0015] 다양한 실시예에 따르면 그리고 도 3-7과 관련하여 아래에 더 상세히 설명되는 바와 같이, 서버 애플리케이션(42) 및 생산성 애플리케이션(30)은 클라이언트 컴퓨터들(2, 6) 상의 파티션들(50, 60)과 서버 컴퓨터(70)의 동기화를 용이하게 하도록 구성될 수 있다. 일 실시예에 따르면, 서버 애플리케이션(42)은 (무상태(stateless))

프로토콜을 포함하지만 이에 한정되지 않는) 프로토콜을 이용하여 증대 문서 동기화를 용이하게 하고, 서버 파일들에 대한 지원을 추가하여 다수의 독립적인 또는 종속적인 데이터 스트림을 노출시키도록 구성될 수 있다. 따라서, 전자 파일 내의 파티션들의 각각은 서로 독립적으로 그리고 상이한 동기화 레이트들로 동기화될 수 있다(즉, 단일 파일 내의 파티션들은 상이한 시각들에 동기화될 수 있다). 예를 들어, 서버 애플리케이션(42)은 생산성 애플리케이션(30)과 연계하여 (콘텐츠(54A)를 포함하는) 파티션(50) 전에 클라이언트 컴퓨터(2) 상의 (메타데이터(64)를 포함하는) 파티션(60)과 서버 컴퓨터(70)를 동기화하도록 구성될 수 있다. 상이한 레이트들로 파티션들을 동기화함으로써, 네트워크(4)를 통해 서버 컴퓨터(70)로 통신되는 데이터의 양이 감소하며, 따라서 서버 확장성이 개선된다는 것을 알아야 한다. 다양한 실시예에 따라 이용될 수 있는 예시적인 무상태 프로토콜이 2008년 5월 2일자로 출원된 "Document Synchronization over Stateless Protocols"라는 제목의 관련 미국 특허 출원 제12/113,975호에 설명되어 있으며, 이 출원의 명세서에는 그 전체가 본 명세서에 참고로 포함된다.

[0016] 클라이언트 컴퓨터들(2, 6) 및 서버 컴퓨터(70)와 관련하여 설명되는 다양한 파티션들 및 스트림들은 예시적인 것일 뿐, 다양한 실시예에 따라 전자 문서 파일 내에 포함될 수 있는 파티션들 또는 스트림들의 수로 한정되지 않는다는 것을 이해해야 한다. 예컨대, 일 실시예에 따르면, 전자 문서 파일은 단일 파티션 및 단일 스트림만을 포함할 수 있는 반면, 다른 실시예들에 따르면, 전자 문서 파일은 다수의 파티션 및 다수의 스트림을 포함할 수 있다. 다양한 실시예에 따르면, 클라이언트 컴퓨터들(2, 6) 및 서버 컴퓨터(70) 상의 전자 문서 파일 내에 포함된 스트림들의 콘텐츠는 메타데이터로 한정되는 것이 아니라, 일반 데이터(즉, 전자 문서 파일 파티션에 저장된 콘텐츠와 관련되지 않은 데이터)도 포함할 수 있다는 것을 더 이해해야 한다. 본 명세서에서 설명되는 실시예들은 전술한 소프트웨어 애플리케이션들로 한정되는 것으로 해석되지 않아야 하며, 다른 개발자들 및/또는 제조자들로부터의 다른 소프트웨어 애플리케이션들도 이용될 수 있다는 것을 더 이해해야 한다. 도 1의 네트워크 아키텍처 및 그의 컴포넌트들은 다른 컴퓨팅 장치들, 통신 장치들 및/또는 다른 시스템들과 통신하기 위한 기능을 포함하며, 본 명세서에서 설명되는 실시예들 및 사례들로 한정되는 것을 의도하지 않는다는 것을 더 이해해야 한다. 따라서, 예컨대, 도 1의 네트워크 아키텍처는 다양한 실시예에 따라 서버 컴퓨터(70)와 통신하는 추가적인 클라이언트 컴퓨터들을 포함할 수 있다.

[0017] 예시적인 운영 환경

[0018] 이제, 도 2를 참조하면, 아래의 설명은 다양한 예시적인 실시예들이 구현될 수 있는 적절한 컴퓨팅 환경의 간단한 일반 설명을 제공하는 것을 의도한다. 개인용 컴퓨터 상의 운영 체제 상에서 동작하는 프로그램 모듈들과 관련하여 실행되는 프로그램 모듈들과 일반적으로 관련하여 다양한 실시예들이 설명되지만, 이 분야의 기술자들은 다양한 실시예들이 다른 타입의 컴퓨터 시스템들 및 프로그램 모듈들과 관련하여서도 구현될 수 있다는 것을 인식할 것이다.

[0019] 일반적으로, 프로그램 모듈들은 특정 작업들을 수행하거나 특정 추상 데이터 타입들을 구현하는 루틴들, 프로그램들, 컴포넌트들, 데이터 구조들 및 다른 타입의 구조들을 포함한다. 더욱이, 이 분야의 기술자들은 다양한 실시예들이 핸드헬드 장치들, 멀티프로세서 시스템들, 마이크로프로세서 기반 또는 프로그래머블 소비자 전자 장치들, 미니컴퓨터들, 메인프레임 컴퓨터들 등을 포함하는 다른 컴퓨터 시스템 구성들을 이용하여 실시될 수 있다는 것을 알 것이다. 다양한 실시예들은 통신 네트워크를 통해 연결된 원격 처리 장치들에 의해 작업들이 수행되는 분산 컴퓨팅 환경들에서도 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈들은 로컬 및 원격 메모리 저장 장치들 모두에 위치할 수 있다.

[0020] 도 2는 하나 이상의 애플리케이션 프로그램을 실행할 수 있는 범용 데스크탑, 랩탑, 핸드헬드, 태블릿 또는 다른 타입의 컴퓨터를 포함할 수 있는 클라이언트 컴퓨터(2)를 나타낸다. 클라이언트 컴퓨터(2)는 적어도 하나의 중앙 처리 유닛(8)("CPU"), 랜덤 액세스 메모리(18)("RAM") 및 판독 전용 메모리("ROM")(20)를 포함하는 시스템 메모리(12), 및 메모리를 CPU(8)에 결합하는 시스템 버스(10)를 포함한다. 이를테면 시동 중에 컴퓨터 내의 요소들 간의 정보 전달을 돕는 기본 루틴들을 포함하는 기본 입출력 시스템이 ROM(20)에 저장된다. 클라이언트 컴퓨터(2)는 운영 체제(32), 생산성 애플리케이션(30), 및 도 1과 관련하여 전술한 파티션(50), 스트림들(52, 56, 62, 68), 콘텐츠(54A), 메타데이터(64A), 저자 리스트(67A) 및 문서 잠금들(66A)을 포함하는 전자 문서 파일(40)을 저장하기 위한 대용량 저장 장치(14)를 더 포함한다.

[0021] 다양한 실시예에 따르면, 운영 체제(32)는 워싱턴 레드몬드의 마이크로소프트사로부터의 WINDOWS 운영 체제들과 같이 네트워킹된 개인용 컴퓨터의 동작을 제어하는 데 적합할 수 있다. 대용량 저장 장치(14)는 버스(10)에 접속된 대용량 저장 장치 제어기(도시되지 않음)를 통해 CPU(8)에 접속된다. 대용량 저장 장치(14) 및 그와 관련된 컴퓨터 판독 가능 매체들은 클라이언트 컴퓨터(2)를 위한 비휘발성 저장을 제공한다. 본 명세서에 포함된

컴퓨터 판독 가능 매체들에 대한 설명은 하드 디스크 또는 CD-ROM 드라이브와 같은 대용량 저장 장치를 참조하지만, 이 분야의 기술자들은 컴퓨터 판독 가능 매체들이 클라이언트 컴퓨터(2)에 의해 액세스되거나 사용될 수 있는 임의의 이용 가능 매체들일 수 있다는 것을 알아야 한다. 한정이나 예외를 들어, 컴퓨터 판독 가능 매체들은 컴퓨터 저장 매체들 및 통신 매체들을 포함할 수 있다.

[0022] 컴퓨터 저장 매체들은 컴퓨터 판독 가능 명령어들, 데이터 구조들, 프로그램 모듈들 또는 다른 데이터와 같은 정보의 저장을 위한 임의의 물리적 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 비이동식 하드웨어 저장 매체들을 포함한다. 컴퓨터 저장 매체들은 원하는 정보를 저장하는 데 사용될 수 있고 클라이언트 컴퓨터(2)에 의해 액세스될 수 있는 RAM, ROM, EPROM, EEPROM, 플래시 메모리 또는 다른 반도체 메모리 기술, CD-ROM, 디지털 다기능 디스크("DVD") 또는 다른 광학 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 다른 자기 저장 장치를 포함하지만, 이에 한정되지 않는다. 통신 매체들은 통상적으로 컴퓨터 판독 가능 명령어들, 데이터 구조들, 프로그램 모듈들 또는 다른 데이터를 반송파 또는 다른 전송 매커니즘과 같은 피변조 데이터 신호 내에 구현하며, 임의의 정보 전달 매체들을 포함한다. "피변조 데이터 신호"라는 용어는 신호 내에 정보를 인코딩하는 방식으로 신호의 특성들 중 하나 이상이 설정 또는 변경된 신호를 의미한다. 한정이나 예외를 들어, 통신 매체들은 유선 네트워크 또는 직접 유선 접속과 같은 유선 매체들, 및 음향, RF, 적외선 및 다른 무선 매체들과 같은 무선 매체들을 포함한다. 위의 임의의 것들의 조합들도 컴퓨터 판독 가능 매체들의 범위 내에 포함되어야 한다. 컴퓨터 판독 가능 매체들은 컴퓨터 프로그램 제품으로도 지칭될 수 있다.

[0023] 다양한 실시예들에 따르면, 클라이언트 컴퓨터(2)는 예를 들어 근거리 네트워크 또는 광역 네트워크(예를 들어, 인터넷)를 포함할 수 있는 네트워크(4)를 통한 원격 컴퓨터들에 대한 논리적 접속들을 이용하여 네트워킹된 환경에서 동작할 수 있다. 클라이언트 컴퓨터(2)는 버스(10)에 접속된 네트워크 인터페이스 유닛(16)을 통해 네트워크(4)에 접속할 수 있다. 네트워크 인터페이스 유닛(16)은 다른 타입의 네트워크들 및 원격 컴퓨팅 시스템들에 접속하는 데에도 사용될 수 있다는 것을 알아야 한다. 클라이언트 컴퓨터(2)는 키보드, 마우스, 펜, 스타일러스, 손가락 및/또는 다른 수단들을 포함하는 다양한 입력 타입들로부터 입력을 수신하고 처리하기 위한 입출력 제어기(22)도 포함할 수 있다. 유사하게, 입출력 제어기(22)는 디스플레이 장치(82), 프린터 또는 다른 타입의 출력 장치에 출력을 제공할 수 있다. 게다가, 터치 스크린이 입력 및 출력 메커니즘으로서 이용될 수 있다. 도 1에 도시된 클라이언트 컴퓨터(6) 및 서버 컴퓨터(70)는 도 2의 클라이언트 컴퓨터(2)와 관련하여 도시된 통상의 컴포넌트들 중 다수를 포함할 수 있다는 것을 알아야 한다.

[0024] 도 3은 일 실시예에 따른, 서버 저장 모델을 이용하여 파일 파티션들을 동기화하기 위한 루틴(300)을 나타내는 흐름도이다. 본 명세서에서 제공되는 루틴들에 대한 설명을 읽을 때, 본 발명의 다양한 실시예들의 논리적 동작들은 (1) 컴퓨팅 시스템 상에서 실행되는 컴퓨터 구현 액트(act)들 또는 프로그램 모듈들의 시퀀스로서 그리고/또는 (2) 컴퓨팅 시스템 내의 상호접속된 기계 논리 회로들 또는 회로 모듈들로서 구현된다는 것을 알아야 한다. 구현은 본 발명을 구현하는 컴퓨팅 시스템의 성능 요구들에 의존하는 선택의 문제이다. 따라서, 도 3-7에 도시되고 본 명세서에서 설명되는 다양한 실시예들을 구성하는 논리적 동작들은 동작들, 구조적 장치들, 액트들 또는 모듈들로서 다양하게 지칭된다. 이 분야의 기술자는 이러한 동작들, 구조적 장치들, 액트들 및 모듈들이 본 명세서에서 설명되는 청구항들 내에 기재된 바와 같은 본 발명의 사상 및 범위로 벗어남 없이, 소프트웨어로, 펌웨어로, 특수 목적 디지털 논리로 그리고 이들의 임의의 조합으로 구현될 수 있다는 것을 인식할 것이다.

[0025] 루틴(300)은 동작 305에서 시작되며, 여기서 클라이언트 컴퓨터(2) 상에서 실행되는 생산성 애플리케이션(30)이 전자 문서 파일(40)의 파티션(50) 내의 (스트림(52) 내에 포함된) 전자 문서 콘텐츠에 대한 편집들을 수신한다. 일 실시예에 따르면, 전자 문서 파일(40)은 파티션들 및 그 안에 포함된 스트림들 모두를 포함하는 서버 컴퓨터(70)로부터 다운로드될 수 있다. 이어서, 파티션(50)을 개방하여, 그 안에 포함된 콘텐츠(54)를 편집함으로써, 콘텐츠(54A)를 생성할 수 있다.

[0026] 동작 305로부터 루틴(300)은 동작 310으로 계속되어, 생산성 애플리케이션(30)이 전자 문서 파일(40)의 파티션(60) 내에 메타데이터(64A)를 생성한다. 특히, 생산성 애플리케이션(30)은 메타데이터를 생성하여, 전자 문서 파일(40)의 파티션(60) 내의 저자 리스트(67)를 갱신하여, 클라이언트 컴퓨터(2)의 사용자를 저자로서 추가할 수 있다(따라서, 저자 리스트(67A)를 생성할 수 있다). 또한, 생산성 애플리케이션(30)은 메타데이터를 생성하여 전자 문서 파일(40)의 파티션(60)에 문서 잠금(66A)을 추가하여, 전자 문서 파일(40)이 현재 클라이언트 컴퓨터(2)의 사용자에 의해 편집되고 있다는 것을 지시할 수 있다.

[0027] 동작 310으로부터 루틴(300)은 동작 315로 계속되어, 생산성 애플리케이션(30)은 메타데이터(64)에 대해 행해진

변경들(즉, 저자 리스트(67A) 및 문서 잠금(66A)을 포함하는 메타데이터(64A))을 서버 컴퓨터(70)에 저장하기 위해 서버 애플리케이션(42)에 파티션(60)을 서버 컴퓨터(70)와 동기화하는 요청을 전송한다. 파티션(60)은 파티션(50)과 독립적으로 동기화될 수 있으며, 따라서 변경된 메타데이터(64A)만이 서버 컴퓨터(70)에 저장된다는 것을 이해해야 한다. 파티션(60)을 서버 컴퓨터와 동기화하기 위한 예시적인 루틴이 도 4와 관련하여 아래에 더 상세히 설명된다.

[0028] 동작 315로부터 루틴(300)은 동작 320으로 계속되어, 생산성 애플리케이션(30)은 편집된 콘텐츠(54A)를 서버 컴퓨터(70)에 저장하기 위해 서버 애플리케이션(42)에 파티션(50)을 서버 컴퓨터(70)와 동기화하는 요청을 전송한다. 파티션(50)은 파티션(60)과 독립적으로 동기화될 수 있으며, 따라서 편집된 콘텐츠(54A)만이 서버 컴퓨터(70)에 저장된다는 것을 이해해야 한다. 파티션들(60, 50)의 동기화의 순서는 역전될 수 있으며, 따라서 파티션(50)이 파티션(60)에 앞서 서버 컴퓨터(70)와 독립적으로 동기화될 수 있다는 것을 더 이해해야 한다. 파티션(50)을 서버 컴퓨터(70)와 동기화하기 위한 예시적인 루틴이 도 5와 관련하여 아래에 더 상세히 설명된다. 동작 320으로부터 루틴(300)은 동작 325로 계속되어, 루틴이 종료된다. 전송한 동작들(310-320)은 일 실시예에 따르면 클라이언트 컴퓨터(6) 상에서 실행되는 생산성 애플리케이션(30)에 의해서도 수행될 수 있다는 것을 이해해야 한다.

[0029] 도 4는 도 3에 도시된 실시예에 따른, 서버 저장 모델을 이용하여 단일 파일 파티션을 동기화하기 위한 루틴(400)을 나타내는 흐름도이다. 루틴(400)은 도 3의 동작 315로부터 시작되고, 동작 405로 계속되어, 서버 애플리케이션(42)은 클라이언트 컴퓨터(2) 상의 파티션(60)과 서버 컴퓨터(70)의 동기화에서 충돌이 존재하는지를 판정한다. 구체적으로 그리고 일 실시예에 따르면, 서버 애플리케이션(42)은 클라이언트 컴퓨터(2) 상의 파티션(60) 내에 저장된 메타데이터(64A)와 서버 컴퓨터(70) 상에 저장된 메타데이터(64)를 비교하여, 임의의 차이들을 판정할 수 있다. 메타데이터(64A)와 메타데이터(64)가 다른 경우, 서버 애플리케이션(42)은 충돌이 존재하는 것으로 판정하며, 루틴(400)은 동작 410으로 계속된다. 메타데이터(64A)와 메타데이터(64) 사이에 차이가 존재하지 않는 경우, 루틴(400)은 도 3의 동작 320으로 복귀한다.

[0030] 동작 410에서, 서버 애플리케이션(42)은 클라이언트 컴퓨터(2) 상의 파티션(60) 내의 메타데이터(64A)가 서버 컴퓨터(70) 상에 저장된 메타데이터(64)보다 새로운 것인지의 여부에 기초하여 충돌을 어떻게 해결할 것인지를 결정한다. 구체적으로, 파티션(60)을 동기화하기 위한 생산성 애플리케이션(30)으로부터의 요청은 메타데이터(64A)가 클라이언트 컴퓨터(2) 상의 파티션(60) 내에 언제 저장되었는지를 지시하는 타임스탬프를 포함할 수 있다. 서버 애플리케이션(42)은 이 타임스탬프와, 메타데이터(64)가 서버 컴퓨터(70) 상의 파티션(60) 내에 언제 저장되었는지를 지시하는 타임스탬프를 비교할 수 있다. 메타데이터(64A)가 메타데이터(64)보다 새로운 것으로 결정되는 경우, 루틴(400)은 동작 415로 계속된다. 메타데이터(64)가 메타데이터(64A)보다 새로운 것으로 결정되는 경우, 루틴(400)은 동작 420으로 분기된다. 이 분야의 기술자들은 다른 실시예에 따르면 파티션(60)을 동기화하기 위한 생산성 애플리케이션(30)으로부터의 요청이 충돌들을 식별하기 위한(그리고 해결하기 위한) 하이퍼텍스트 마크업 언어("HTTP") 엔티티 태그("ETAG")를 포함할 수 있다는 것을 이해해야 한다.

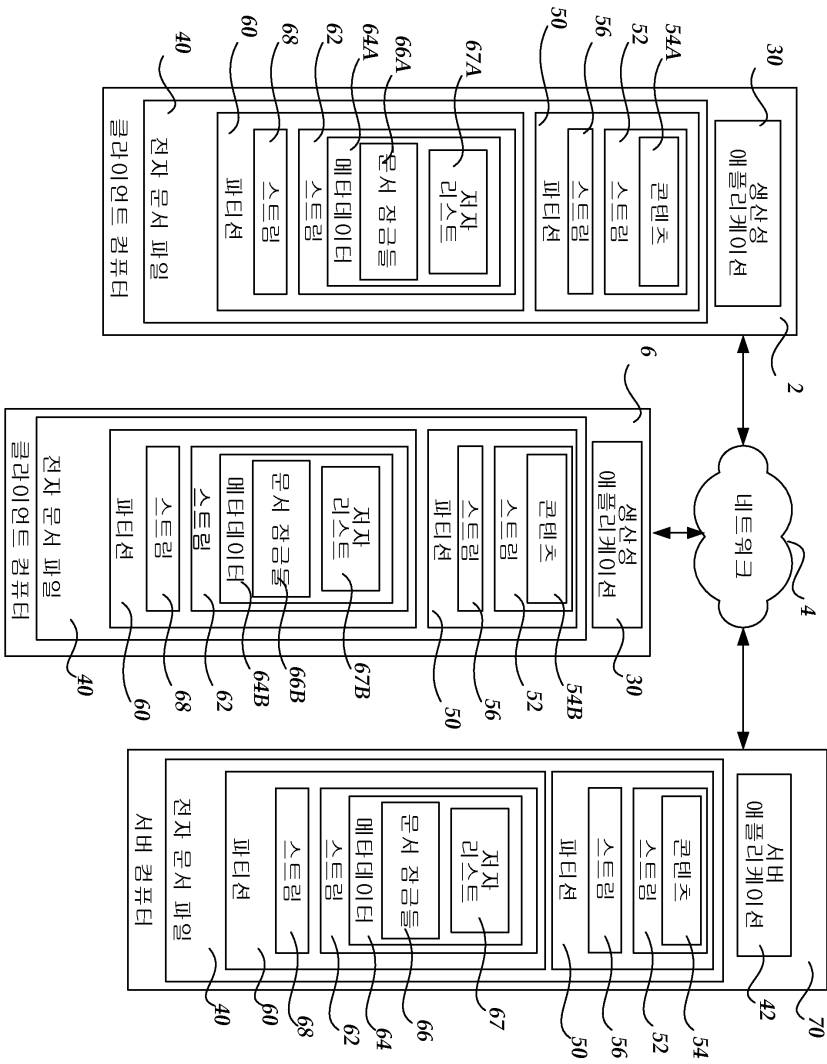
[0031] 동작 415에서, 서버 애플리케이션(42)은 클라이언트 컴퓨터(2) 상의 파티션(60)과 서버 컴퓨터(70)를 자동으로 동기화하여, 클라이언트 컴퓨터(2) 상에 존재하는 가장 최신의 메타데이터(64A)의 편에서 충돌을 해결한다. 따라서, 서버 애플리케이션(42)은 서버 컴퓨터(70) 상의 파티션(60) 내의 메타데이터(64)를 메타데이터(64A)로 갱신할 수 있다. 동작 415로부터 루틴(400)은 도 3의 동작 320으로 복귀한다.

[0032] 동작 420에서, 서버 애플리케이션(42)은 클라이언트 컴퓨터(2) 상의 파티션(60)과 서버 컴퓨터(70)의 동기화가 자동으로 실패하게 하여, 메타데이터(64A)보다 새로운 메타데이터(64)의 편에서 충돌을 해결한다. 따라서, 서버 애플리케이션(42)은 서버 컴퓨터(70) 상의 파티션(60) 내의 메타데이터(64)를 유지할 수 있다. 동작 420으로부터 루틴(400)은 도 3의 동작 320으로 복귀한다.

[0033] 도 5는 도 3에 도시된 실시예에 따른, 서버 저장 모델을 이용하여 다른 파일 파티션을 동기화하기 위한 루틴(500)을 나타내는 흐름도이다. 루틴(500)은 도 3의 동작 320으로부터 시작되고, 동작 505로 계속되어, 서버 애플리케이션(42)은 클라이언트 컴퓨터(2)의 파티션(50)과 서버 컴퓨터(70)의 동기화에서 충돌이 존재하는지를 판정한다. 구체적으로 그리고 일 실시예에 따르면, 서버 애플리케이션(42)은 클라이언트 컴퓨터(2) 상의 파티션(50)에 저장된 콘텐츠(54A)와 서버 컴퓨터(70) 상의 파티션(50) 내에 저장된 콘텐츠(54)를 비교할 수 있다. 콘텐츠(54)가 클라이언트 컴퓨터(2) 상의 파티션(50) 내의 편집된 콘텐츠(54A) 내에 존재하지 않는 편집들을 포함하는 경우, 서버 컴퓨터(70)는 충돌이 존재하는 것으로 판정하며, 루틴(500)은 동작 510으로 계속된다. 콘텐츠(54)가 어떠한 편집도 포함하지 않는 경우, 루틴(500)은 동작 515로 계속된다.

- [0034] 동작 510에서, 서버 애플리케이션(42)은 클라이언트 컴퓨터(2)의 사용자에게 대한 충돌 에러 메시지를 생성한다. 예컨대, 클라이언트 컴퓨터(2)에 대해 서버 애플리케이션(42)에 의해 생성되는 충돌 에러 메시지는 이전에 서버(70)에 업로드된 (예를 들어, 클라이언트 컴퓨터(6)의 사용자에게 의해 행해진) 콘텐츠(54)에 대한 보이지 않는 편집들을 포함할 수 있다. 충돌 에러 메시지에 응답하여, 클라이언트 컴퓨터(2)의 사용자는 편집된 콘텐츠(54A)와 콘텐츠(54)에 대한 편집들을 합체하거나, 대안으로서 콘텐츠(54)에 대한 이전의 편집들을 서버(70) 상에서 변경되지 않게 유지함으로써 수동으로 충돌을 해결할 수 있다. 동작 510으로부터 루틴(500)은 도 3의 동작 325로 복귀한다.
- [0035] 동작 515에서, 서버 애플리케이션(42)은 파티션(50) 내의 편집된 콘텐츠(54A)와 서버(70)를 동기화하여, 서버(70) 상의 콘텐츠(54)를 편집된 콘텐츠(54A)로 갱신한다. 동작 515로부터 루틴(500)은 도 3의 동작(325)으로 복귀한다.
- [0036] 도 6은 다른 실시예에 따른, 서버 저장 모델을 이용하여 파일 파티션들을 동기화하기 위한 루틴(600)을 나타내는 흐름도이다. 루틴(600)은 동작 605에서 시작되어, 클라이언트 컴퓨터(2) 상에서 실행되는 생산성 애플리케이션(30)은 전자 문서 파일(40)의 파티션(50) 내의 (스트림(52) 내에 포함된) 전자 문서 콘텐츠에 대한 편집들을 수신한다. 일 실시예에 따르면, 전자 문서 파일(40)은 파티션들 및 그 안에 포함된 스트림들 모두를 포함하는 서버 컴퓨터(70)로부터 다운로드될 수 있다. 이어서, 파티션(50)을 개방하여, 그 안에 포함된 콘텐츠(54)를 편집함으로써, 콘텐츠(54A)를 생성할 수 있다.
- [0037] 동작 605로부터 루틴(600)은 동작 610으로 계속되어, 생산성 애플리케이션(30)은 전자 문서 파일(40)의 파티션(60) 내에 메타데이터(64A)를 생성하여, 파티션(50) 내의 콘텐츠(54A)의 편집자를 식별한다. 구체적으로, 생산성 애플리케이션(30)은 메타데이터를 생성하여, 전자 문서 파일(40)의 파티션(60) 내의 저자 리스트(67)에 콘텐츠(54A)의 편집자의 이름을 추가할 수 있다(따라서, 저자 리스트(67A)를 생성할 수 있다).
- [0038] 동작 610으로부터 루틴(600)은 동작 615로 계속되어, 생산성 애플리케이션(30)은 메타데이터(64)에 대해 행해진 변경들(즉, 저자 리스트(67A)를 포함하는 메타데이터(64A))를 서버 컴퓨터(70)에 저장하기 위해 파티션(60)과 서버 컴퓨터(70)를 동기화하는 서버 애플리케이션(42)에 전송한다. 파티션(60)은 파티션(50)과 독립적으로 동기화될 수 있으며, 따라서 변경된 메타데이터(64A)만이 서버 컴퓨터(70)에 저장된다는 것을 이해해야 한다.
- [0039] 동작 615로부터 루틴(600)은 동작 620으로 계속되어, 클라이언트 컴퓨터(6) 상에서 실행되는 생산성 애플리케이션(30)은 전자 문서 파일(40)을 다운로드함으로써 서버 컴퓨터(70)로부터 (그 안에 포함된 스트림들(42, 56, 62, 68)을 포함하는) 파일 파티션들(50, 60)을 수신한다. 클라이언트 컴퓨터(6)는 파일 파티션(60)의 수신시에 동작 615에서 클라이언트 컴퓨터(2) 상의 파티션(60)과 서버 컴퓨터(70)의 동기화의 결과로서 클라이언트 컴퓨터(2)로부터 콘텐츠(54A)의 편집자를 식별하는 메타데이터(64A)도 수신한다는 것을 이해해야 한다. 따라서, 클라이언트 컴퓨터(6)의 사용자는 클라이언트 컴퓨터(2)의 사용자도 편집을 위해 서버 컴퓨터(70)로부터 콘텐츠(54)를 다운로드한 것으로 결정할 수 있다.
- [0040] 동작 620으로부터 루틴(600)은 동작 625로 계속되어, 클라이언트 컴퓨터(6) 상에서 실행되는 생산성 애플리케이션(30)은 전자 문서 파일(40)의 파티션(50) 내의 (스트림(52) 내에 포함된) 전자 문서 콘텐츠에 대한 편집들을 수신한다. 이어서, 파티션(50)을 개방하여, 그 안에 포함된 콘텐츠(54)를 편집하여, 콘텐츠(54B)를 생성할 수 있다. 편집된 콘텐츠(54B)는 클라이언트 컴퓨터(2) 상에서 행해진 편집들(즉, 편집된 콘텐츠(54A))을 포함하지 않는다는 것을 이해해야 한다.
- [0041] 동작 625로부터 루틴(600)은 동작 630으로 계속되어, 클라이언트 컴퓨터(6) 상에서 실행되는 생산성 애플리케이션(30)은 요청을 서버 애플리케이션(42)에 전송하여, 파티션(60)과 서버 컴퓨터(70)를 동기화하여, 메타데이터(64)에 대해 행해진 변경들(즉, 저자 리스트(67B)를 포함하는 메타데이터(64B))를 서버 컴퓨터(70)에 저장한다. 파티션(60)은 파티션(50)과 독립적으로 동기화될 수 있으며, 따라서 변경된 메타데이터(64B)만이 서버 컴퓨터(70)에 저장된다는 것을 이해해야 한다.
- [0042] 동작 630으로부터 루틴(600)은 동작 635로 계속되어, 클라이언트 컴퓨터(2)는 동작 630에서 파티션(60)과 서버 컴퓨터(70)의 동기화의 결과로서 클라이언트 컴퓨터(6)로부터 콘텐츠(54B)의 편집자를 식별하는 메타데이터(64B)를 수신한다. 구체적으로, 클라이언트 컴퓨터(2) 상에서 실행되는 생산성 애플리케이션(30)은 클라이언트 컴퓨터(6)로부터의 메타데이터(64B)로 이전에 갱신된 파티션(60)을 서버 컴퓨터(70)로부터 다운로드할 수 있다. 따라서, 클라이언트 컴퓨터(2)의 사용자는 클라이언트 컴퓨터(6)의 사용자도 편집을 위해 서버 컴퓨터(70)로부터 콘텐츠(54)를 다운로드한 것으로 결정할 수 있다.

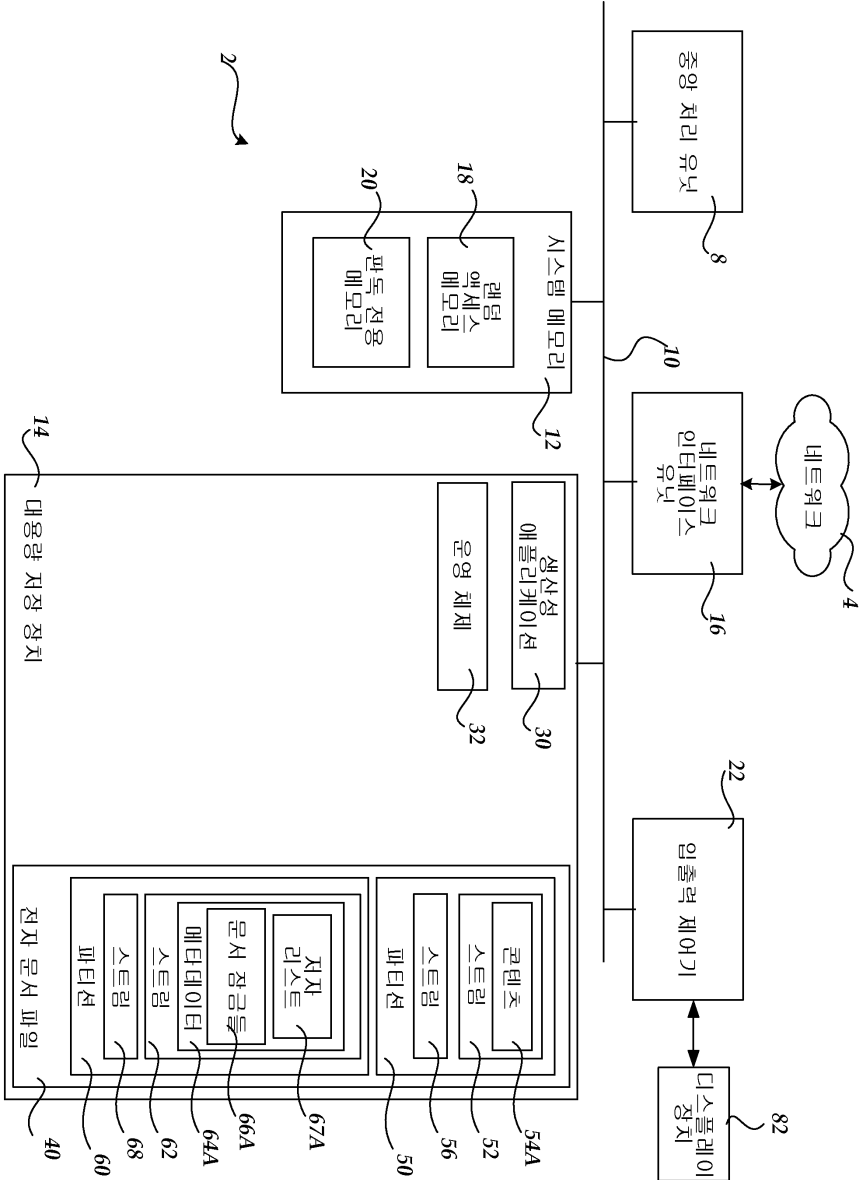
- [0043] 동작 635로부터 루틴(600)은 동작 640으로 계속되어, 클라이언트 컴퓨터(2) 상에서 실행되는 생산성 애플리케이션(30)은 요청을 서버 애플리케이션(42)으로 전송하여, 파티션(50)과 서버 컴퓨터(70)를 동기화하여, 편집된 콘텐츠(54A)를 서버 컴퓨터(70)에 저장한다. 파티션(50)은 파티션(60)과 독립적으로 동기화될 수 있으며, 따라서 편집된 콘텐츠(54A)만이 서버 컴퓨터(70)에 저장된다는 것을 이해해야 한다. 파티션들(60, 50)의 동기화의 순서는 역전될 수 있으며, 따라서 파티션(50)이 파티션(60)에 앞서 서버 컴퓨터(70)와 독립적으로 동기화될 수 있다는 것을 더 이해해야 한다.
- [0044] 동작 640으로부터 루틴(600)은 동작 645로 계속되어, 클라이언트 컴퓨터(6) 상에서 실행되는 생산성 애플리케이션(30)은 요청을 서버 애플리케이션(42)으로 전송하여, 파티션(50)과 서버 컴퓨터(70)를 동기화하여, 편집된 콘텐츠(54B)를 서버 컴퓨터(70)에 저장한다. 파티션(50)은 파티션(60)과 독립적으로 동기화될 수 있으며, 따라서 편집된 콘텐츠(54B)만이 서버 컴퓨터(70)에 저장될 수 있다는 것을 이해해야 한다. 파티션들(60, 50)의 동기화의 순서는 역전될 수 있고, 따라서 파티션(50)은 파티션(60)에 앞서 서버 컴퓨터(70)와 독립적으로 동기화될 수 있다는 것을 더 이해해야 한다. 파티션(50)과 서버 컴퓨터(70)를 동기화하기 위한 예시적인 루틴이 도 7과 관련하여 아래에 더 상세히 설명된다. 동작 645로부터 루틴(600)은 동작 600으로 계속되어, 루틴이 종료된다.
- [0045] 도 7은 도 6에 도시된 실시예에 따른, 서버 저장 모델을 이용하여 제2 클라이언트 컴퓨터 상의 파일 파티션을 동기화하기 위한 루틴(700)을 나타내는 흐름도이다. 루틴(700)은 도 6의 동작 645로부터 시작되고, 동작 605로 계속되어, 클라이언트 컴퓨터(6) 상에서 실행되는 생산성 애플리케이션(30)은 파티션(50)에 대해 행해진 임의의 갱신들에 대해 서버 컴퓨터(70)를 검사한다. 구체적으로, 생산성 애플리케이션(30)은 서버 컴퓨터(70) 상의 파티션(60) 내의 콘텐츠(54)가 클라이언트 컴퓨터(2)에 의해 갱신되었는지에 관한 질의를 서버 애플리케이션(42)으로 전송할 수 있다. 도 6의 동작 620에서 전송한 바와 같이, 클라이언트 컴퓨터(6)의 사용자는 서버 컴퓨터(70)로부터 다운로드된 전자 파일(40) 내의 파일 파티션들(50, 60)의 수신시에 서버 컴퓨터(70)의 파티션(50) 내의 콘텐츠(54)가 클라이언트 컴퓨터(2)에서 다른 사용자에게 의해 편집되고 있다는 것을 알 수 있다.
- [0046] 동작 705로부터 루틴(700)은 동작 710으로 계속되어, 클라이언트 컴퓨터(6)는 서버 컴퓨터(70) 상의 파티션(50)이 (즉, 클라이언트 컴퓨터(2)에 의해) 갱신되었고, 따라서 클라이언트 컴퓨터(2)로부터의 편집된 콘텐츠(54A)가 클라이언트 컴퓨터(6)의 파티션(50) 내의 편집된 콘텐츠(54B)와 합체되도록 이용될 수 있다는 확인을 서버 애플리케이션(42)으로부터 수신한다. 동작 710으로부터 루틴(700)은 동작 715로 계속되어, 클라이언트 컴퓨터(6)는 서버 컴퓨터(70)로부터 편집된 콘텐츠(54A)를 수신한다.
- [0047] 동작 715로부터 루틴(700)은 동작 720으로 계속되어, 클라이언트 컴퓨터(6) 상에서 실행되는 생산성 애플리케이션(30)은 (클라이언트 컴퓨터(2)로부터의) 편집된 콘텐츠(54A) 및 (클라이언트 컴퓨터(6)로부터의) 편집된 콘텐츠(54B)로 파티션(50)을 갱신하도록 서버 컴퓨터(70) 상의 서버 애플리케이션(42)에 요청한다. 동작 720으로부터 루틴(700)은 도 6의 동작 650으로 복귀한다.
- [0048] 본 발명은 다양한 예시적인 실시예들과 관련하여 설명되었지만, 이 분야의 기술자들은 이어지는 청구항들의 범위 내에서 그에 대한 많은 개량이 이루어질 수 있다는 것을 이해할 것이다. 따라서, 본 발명의 범위는 위의 설명에 의해 한정되는 것을 의도하지 않으며, 그 대신에, 이어지는 청구항들을 참조하여서만 결정되어야 한다.



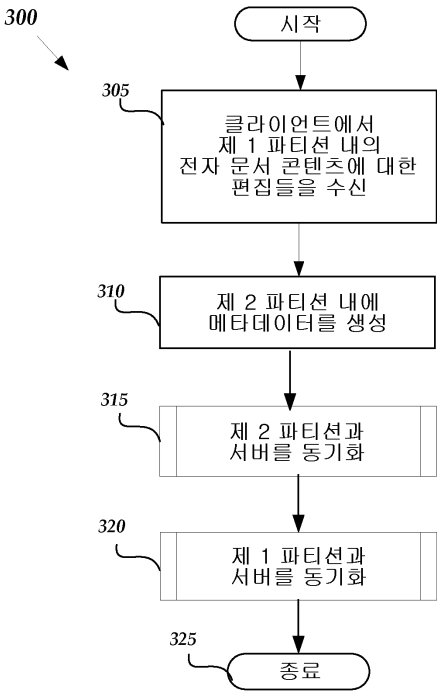
도면

도면1

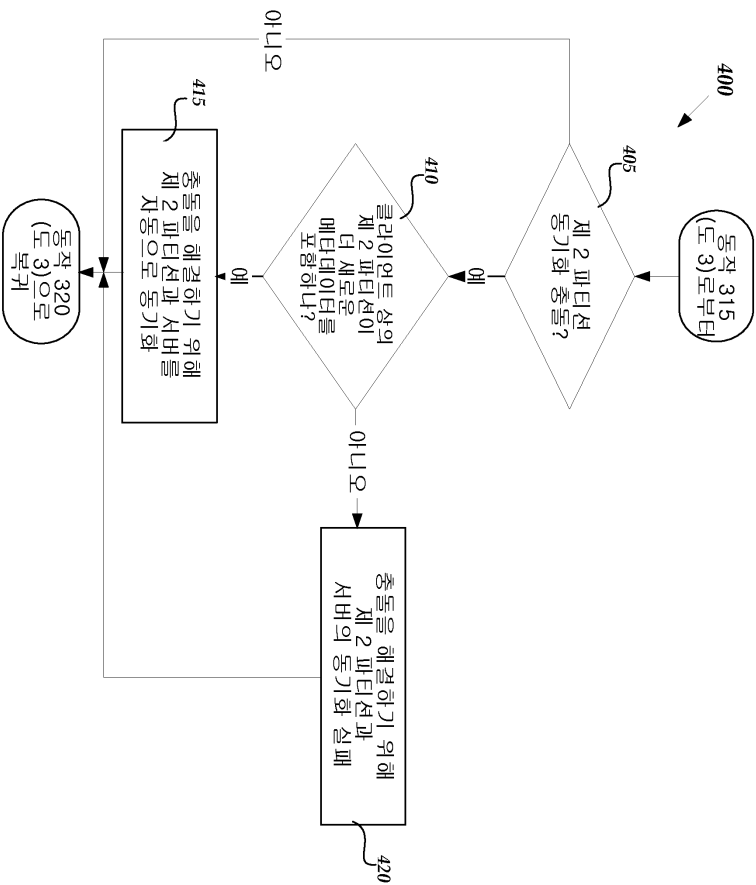
도면2



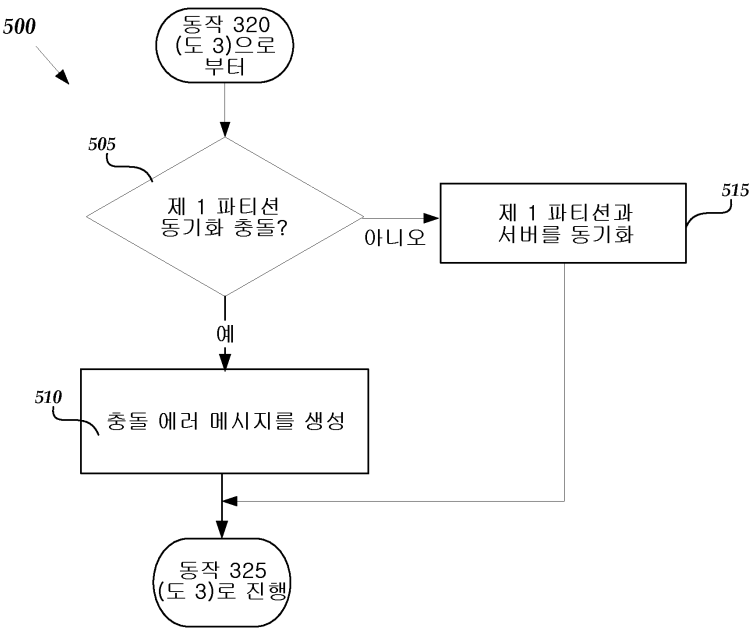
도면3



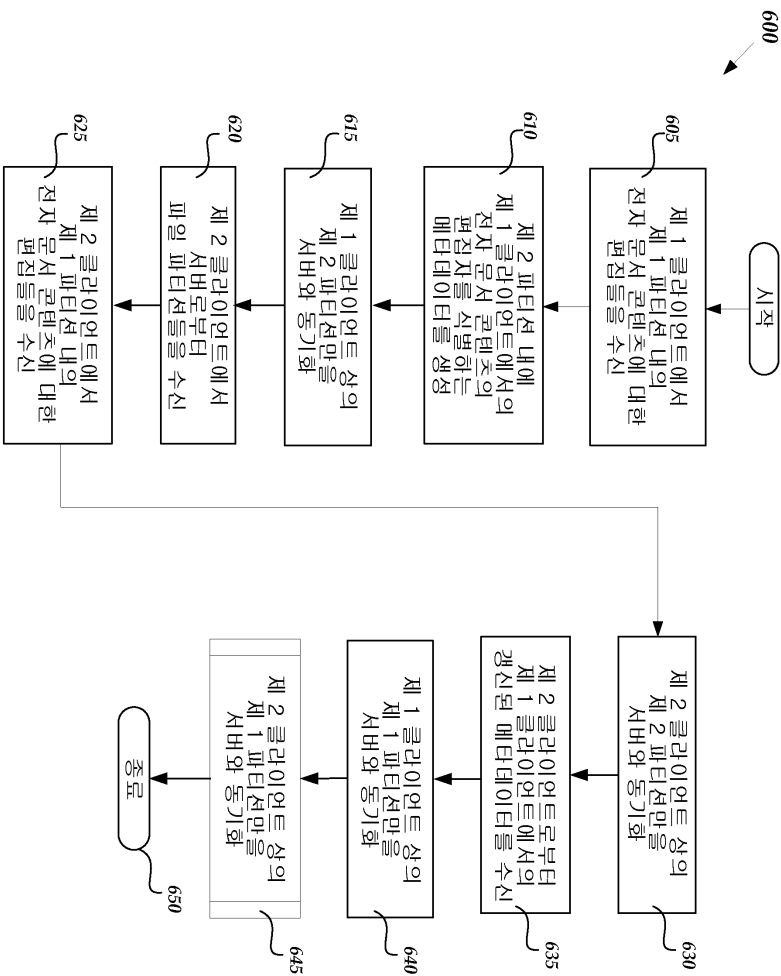
도면4



도면5



도면6



도면7

