

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7579302号  
(P7579302)

(45)発行日 令和6年11月7日(2024.11.7)

(24)登録日 令和6年10月29日(2024.10.29)

(51)国際特許分類 F I  
G 0 6 F 16/182 (2019.01) G 0 6 F 16/182  
G 0 6 F 16/178 (2019.01) G 0 6 F 16/178

請求項の数 27 外国語出願 (全55頁)

(21)出願番号	特願2022-102834(P2022-102834)	(73)特許権者	509117964
(22)出願日	令和4年6月27日(2022.6.27)		ドロップボックス, インコーポレイテッド
(62)分割の表示	特願2020-521435(P2020-521435)の分割		アメリカ合衆国, カリフォルニア州 94158, サンフランシスコ, オーウェンズ ストリート 1800, スイート200
原出願日	平成30年12月17日(2018.12.17)	(74)代理人	110003281
(65)公開番号	特開2022-141669(P2022-141669A)		弁理士法人大塚国際特許事務所
(43)公開日	令和4年9月29日(2022.9.29)	(72)発明者	ゴールドベルグ, アイザック
審査請求日	令和4年7月25日(2022.7.25)		アメリカ合衆国, カリフォルニア州 94107, サンフランシスコ, ブラナン ストリート 333, ドロップボックス, インコーポレイテッド 内
(31)優先権主張番号	15/868,518	(72)発明者	ジャヤカル, スジャイ
(32)優先日	平成30年1月11日(2018.1.11)		
(33)優先権主張国・地域又は機関	米国(US)		
(31)優先権主張番号	15/868,489		
(32)優先日	平成30年1月11日(2018.1.11)		
	最終頁に続く		最終頁に続く

(54)【発明の名称】 コンテンツ管理クライアント同期サービス

(57)【特許請求の範囲】

【請求項1】

クライアントデバイスであって、  
少なくとも1つのプロセッサと、  
命令およびツリーデータ構造を格納する少なくとも1つのコンピュータ可読媒体と、を  
備え、  
前記ツリーデータ構造は、  
現在の期間中のコンテンツ管理システム上のユーザーアカウントに関連付けられたコン  
テンツアイテムのサーバ状態を表すリモートツリーと、  
前記現在の期間中の前記クライアントデバイス上の前記ユーザーアカウントに関連付け  
られたコンテンツアイテムのファイルシステム状態を表すローカルツリーと、  
同期ツリー、前記ローカルツリー、および前記リモートツリーが同一であった以前の期  
間における既知の同期状態を表す前記同期ツリーであって、前記以前の期間において前記  
ローカルツリーおよび前記リモートツリーの両方に含まれていたコンテンツアイテムの表  
現を含む、前記同期ツリーと、  
を含み、  
前記命令は、前記プロセッサによって実行されると、前記プロセッサに、  
前記クライアントデバイスに格納されている前記リモートツリー、前記ローカルツリー  
、および前記同期ツリーの少なくとも1つを更新することと、  
前記同期ツリーと前記リモートツリーとを比較することと、

10

20

前記比較に基づいて、前記同期ツリーと前記リモートツリーとの間の少なくとも1つの差異を特定することと、

前記少なくとも1つの差異に基づいて操作セットを生成することと、ここで、前記操作セットは、前記サーバ状態と前記ファイルシステム状態とを収束するように構成され、

前記操作セットの実行を管理することと、を行わせる、クライアントデバイス。

【請求項2】

前記操作セットの動作の実行を管理するための命令が、前記少なくとも1つのプロセッサに、さらに、

コンテンツ管理システムによって格納されたコンテンツアイテムのサーバ修正データを受信することと、

前記同期ツリーと前記リモートツリーを比較する前に、前記サーバ修正データに基づいて、前記クライアントデバイスに格納されている前記リモートツリーを更新することと、を行わせる、請求項1に記載のクライアントデバイス。

【請求項3】

前記命令は、前記少なくとも1つのプロセッサに、さらに、

前記クライアントデバイスによって格納されたコンテンツアイテムのクライアント修正データを受信することと、

前記クライアント修正データに基づいて、前記ローカルツリーを更新することと、を行わせる、請求項1または2に記載のクライアントデバイス。

【請求項4】

前記命令は、前記少なくとも1つのプロセッサに、さらに、

前記同期ツリーを前記ローカルツリーと比較することと、

前記比較に基づいて、前記同期ツリーと前記ローカルツリーとの間の少なくとも1つの差異を特定することと、

前記少なくとも1つの差異に基づいて、サーバ操作セットを生成することと、

前記サーバ操作セットをコンテンツ管理システムに送信することと、を行わせる、請求項3に記載のクライアントデバイス。

【請求項5】

前記リモートツリー、前記ローカルツリー、および前記同期ツリーが、前記クライアントデバイスの前記コンピュータ可読媒体上に格納され、更新される、請求項1乃至4のいずれか一項に記載のクライアントデバイス。

【請求項6】

前記サーバ状態と前記ファイルシステム状態とが同期していないことは、サーバに格納された前記コンテンツアイテムに対して実行されたアクションが、前記クライアントデバイスに格納された前記コンテンツアイテムに反映されていないことを示す、請求項3に記載のクライアントデバイス。

【請求項7】

前記サーバ状態と前記ファイルシステム状態とが同期していないことは、前記クライアントデバイスに格納された前記コンテンツアイテムに対して実行されたアクションが、コンテンツ管理サービスによって格納された前記コンテンツアイテムに反映されていないことを示す、請求項2に記載のクライアントデバイス。

【請求項8】

プログラムであって、クライアントデバイスによって実行されると、前記クライアントデバイスに、

現在の期間中のコンテンツ管理システム上のユーザーアカウントに関連付けられたコンテンツアイテムのサーバ状態を表すリモートツリーと、

前記現在の期間中の前記クライアントデバイス上の前記ユーザーアカウントに関連付けられたコンテンツアイテムのファイルシステム状態を表すローカルツリーと、

同期ツリー、前記ローカルツリー、および前記リモートツリーが同一であった以前の期間における既知の同期状態を表す前記同期ツリーであって、前記以前の期間において前記

10

20

30

40

50

ローカルツリーおよび前記リモートツリーの両方に含まれていたコンテンツアイテムの表現を含む、前記同期ツリーと、

を含む、ツリーデータ構造を前記クライアントデバイスに格納することと、

前記クライアントデバイスに格納されている前記リモートツリー、前記ローカルツリー、および前記同期ツリーの少なくとも1つを更新することと、

前記同期ツリーと前記リモートツリーとの間の差異を特定するために、前記リモートツリーと前記同期ツリーとを比較することと、

前記リモートツリーと前記同期ツリーとの間の前記差異に基づいて、前記サーバ状態と前記ファイルシステム状態が同期していないことを判断することと、

前記リモートツリーと前記同期ツリーとの間の前記差異に基づいて、前記クライアントデバイスに格納されているコンテンツアイテムのためのクライアント操作セットを生成することと、ここで、前記クライアント操作セットは、前記クライアントデバイスに格納されている前記コンテンツアイテムを操作して前記サーバ状態と前記ファイルシステム状態とを収束するように構成されている、を行わせる、プログラム。

【請求項9】

プログラムであって、クライアントデバイスによって実行されると、前記クライアントデバイスに、

現在の期間中のコンテンツ管理システム上のユーザーアカウントに関連付けられたコンテンツアイテムのサーバ状態を表すリモートツリーと、

前記現在の期間中の前記クライアントデバイス上の前記ユーザーアカウントに関連付けられたコンテンツアイテムのファイルシステム状態を表すローカルツリーと、

同期ツリー、前記ローカルツリー、および前記リモートツリーが同一であった以前の期間における既知の同期状態を表す前記同期ツリーであって、前記以前の期間において前記ローカルツリーおよび前記リモートツリーの両方に含まれていたコンテンツアイテムの表現を含む、前記同期ツリーと、

を含む、ツリーデータ構造を格納することと、

前記リモートツリーと、前記ローカルツリーと、前記同期ツリーと、の少なくとも1つを更新することと、ここで、前記同期ツリーは、前記以前の期間の前記ローカルツリーおよび前記リモートツリーと一致し、前記既知の同期状態において、前記コンテンツ管理システム上の前記ユーザーアカウントに関連付けられた前記コンテンツアイテムは、前記クライアントデバイス上の前記ユーザーアカウントに関連付けられた前記コンテンツアイテムと同期し、

前記コンテンツ管理システムによって格納されたコンテンツアイテムのサーバ修正データを受信することと、

前記サーバ修正データに基づいて、前記リモートツリーを更新することと、

前記リモートツリーと前記同期ツリーとを比較することにより、前記コンテンツ管理システムの前記ユーザーアカウントに関連付けられているコンテンツアイテムの前記サーバ状態が、前記クライアントデバイスに同期されていない少なくとも1つの修正を含むことを判断することと、を行わせる、プログラム。

【請求項10】

命令は、さらに、前記クライアントデバイスに、前記同期ツリーと前記ローカルツリーとを比較することによって、前記クライアントデバイスの前記ファイルシステム状態が、前記コンテンツ管理システムに同期されていない少なくとも1つの修正を含むことを判断させる、請求項9に記載のプログラム。

【請求項11】

命令は、前記クライアントデバイスに、さらに、前記コンテンツ管理システムによって格納された前記コンテンツアイテムを操作して前記サーバ状態および前記ファイルシステム状態を収束させるように構成された、操作セットを生成させる、請求項10に記載のプログラム。

【請求項12】

10

20

30

40

50

ツリーデータ構造のセットが前記クライアントデバイスに格納されている、請求項 9 乃至 11 のいずれか一項に記載のプログラム。

【請求項 13】

コンテンツ管理システム上のユーザーアカウントへの修正を、前記ユーザーアカウントへのアクセスを許可されたクライアントデバイスに同期させるための、コンピュータ実装方法であって、

現在の期間中のコンテンツ管理システム上のユーザーアカウントに関連付けられたコンテンツアイテムのサーバ状態を表すリモートツリーと、

前記現在の期間中の前記クライアントデバイス上の前記ユーザーアカウントに関連付けられたコンテンツアイテムのファイルシステム状態を表すローカルツリーと、

同期ツリー、前記ローカルツリー、および前記リモートツリーが同一であった以前の期間における既知の同期状態を表す前記同期ツリーであって、前記以前の期間において前記ローカルツリーおよび前記リモートツリーの両方に含まれていたコンテンツアイテムの表現を含む、前記同期ツリーと、

を含む、ツリーデータ構造を格納することと、

前記クライアントデバイスに格納されている前記リモートツリー、前記ローカルツリー、および前記同期ツリーの少なくとも1つを更新することと、ここで、前記リモートツリーは第1の期間のサーバ状態を表し、前記ローカルツリーは、前記第1の期間のファイルシステム状態を表し、前記同期ツリーは、以前の期間の前記サーバ状態とファイルシステム状態の間の同期状態を表し、

前記コンテンツ管理システムの前記ユーザーアカウントに関連付けられたコンテンツアイテムの前記サーバ状態を表す前記リモートツリーのツリーデータ構造と、前記コンテンツ管理システムの前記サーバ状態と前記クライアントデバイスのユーザーアカウントに関連付けられたコンテンツアイテムの前記ファイルシステム状態との間の既知の同期状態を表す前記同期ツリーのツリーデータ構造と、の間の差異を計算することと、

前記差異に基づいて操作セットを生成することと、前記操作セットは、前記クライアントデバイスで実行されたときに、前記クライアントデバイスに格納されている前記コンテンツアイテムを更新して前記ファイルシステム状態と前記サーバ状態とを収束させる、を備えるコンピュータ実装方法。

【請求項 14】

前記操作セットを実行することをさらに含む、請求項 13 に記載のコンピュータ実装方法。

【請求項 15】

前記リモートツリーの前記ツリーデータ構造および前記同期ツリーの前記ツリーデータ構造が、前記クライアントデバイスに格納される、請求項 13 または 14 に記載のコンピュータ実装方法。

【請求項 16】

前記クライアントデバイスから、前記クライアントデバイスに格納されているコンテンツアイテムのクライアント修正データを受信することと、

前記クライアント修正データに基づいて、前記ローカルツリーを更新することと、ここで、前記ローカルツリーは、前記クライアントデバイス上の前記ファイルシステム状態を表し、をさらに含む、請求項 13 乃至 15 のいずれか一項に記載のコンピュータ実装方法。

【請求項 17】

前記クライアント修正データに基づいて、前記同期ツリーを更新することをさらに備える、請求項 16 に記載のコンピュータ実装方法。

【請求項 18】

前記クライアントデバイス上の前記ユーザーアカウントに関連付けられたコンテンツアイテムの前記ファイルシステム状態を表す前記ローカルツリーのツリーデータ構造と、前記コンテンツ管理システムと前記クライアントデバイスとの間の前記既知の同期状態を表す前記同期ツリーの前記ツリーデータ構造と、の間の第2の差異を計算することと、

10

20

30

40

50

前記差異に基づいて、実行されると、前記コンテンツ管理システムに格納された前記コンテンツアイテムを更新して、前記コンテンツ管理システム上の前記ユーザーアカウントに関連付けられたコンテンツアイテムの前記サーバ状態と前記ファイルシステム状態とを収束させる第2の操作セットを生成することと、

前記第2の操作セットの実行を管理することと、をさらに含む、請求項13乃至17のいずれか一項に記載のコンピュータ実装方法。

【請求項19】

前記ローカルツリーの前記ツリーデータ構造が前記クライアントデバイスに格納される、請求項18に記載のコンピュータ実装方法。

【請求項20】

サーバ修正データに基づいて、前記同期ツリーを更新することをさらに含む、請求項18または19に記載のコンピュータ実装方法。

【請求項21】

クライアントデバイス上のコンテンツアイテムに対する修正を同期するためのコンピュータ実装方法であって、

現在の期間中のコンテンツ管理システム上のユーザーアカウントに関連付けられたコンテンツアイテムのサーバ状態を表すリモートツリーと、

前記現在の期間中の前記クライアントデバイス上の前記ユーザーアカウントに関連付けられたコンテンツアイテムのファイルシステム状態を表すローカルツリーと、

同期ツリー、前記ローカルツリー、および前記リモートツリーが同一であった以前の期間における既知の同期状態を表す前記同期ツリーであって、前記以前の期間において前記ローカルツリーおよび前記リモートツリーの両方に含まれていたコンテンツアイテムの表現を含む、前記同期ツリーと、

を含む、ツリーデータ構造を前記クライアントデバイスに格納することと、

前記クライアントデバイスに格納されている前記リモートツリー、前記ローカルツリー、および前記同期ツリーの少なくとも1つを更新することと、

前記クライアントデバイス上のユーザーアカウントに関連付けられたコンテンツアイテムの前記ファイルシステム状態を表す前記ローカルツリーのツリーデータ構造と、コンテンツ管理システムと前記クライアントデバイスとの間の既知の同期状態を表す前記同期ツリーのツリーデータ構造と、の間の第1の差異を計算することと、

前記コンテンツ管理システム上の前記ユーザーアカウントに関連付けられたコンテンツアイテムの前記サーバ状態を表す前記リモートツリーのツリーデータ構造と、前記同期ツリーの前記ツリーデータ構造と、の間の第2の差異を計算することと、

前記第1の差異と前記第2の差異とに基づいて操作セットを生成することと、前記操作セットは、実行されると、前記コンテンツ管理システムに格納されている前記コンテンツアイテムを更新して、前記コンテンツ管理システムの前記ユーザーアカウントに関連付けられているコンテンツアイテムの前記サーバ状態と前記ファイルシステム状態とを収束する、を備えるコンピュータ実装方法。

【請求項22】

前記操作セットを実行することをさらに含む、請求項21に記載のコンピュータ実装方法。

【請求項23】

前記ローカルツリーの前記ツリーデータ構造および前記同期ツリーの前記ツリーデータ構造が前記クライアントデバイスに格納される、請求項22に記載のコンピュータ実装方法。

【請求項24】

前記クライアントデバイスから、前記クライアントデバイスに格納されているコンテンツアイテムのクライアント修正データを受信することと、

前記クライアント修正データに基づいて、前記ローカルツリーを更新することと、をさらに備える、請求項22または23のいずれか一項に記載のコンピュータ実装方法。

10

20

30

40

50

**【請求項 25】**

前記クライアント修正データに基づいて、前記同期ツリーを更新することをさらに備える、請求項 24 に記載のコンピュータ実装方法。

**【請求項 26】**

サーバ修正データに基づいて、前記同期ツリーを更新することをさらに含む、請求項 22 乃至 25 のいずれか一項に記載のコンピュータ実装方法。

**【請求項 27】**

前記リモートツリーの前記ツリーデータ構造は、前記クライアントデバイスに格納される、請求項 22 乃至 26 のいずれか一項に記載のコンピュータ実装方法。

**【発明の詳細な説明】****【技術分野】****【0001】**

## 関連出願の相互参照

本出願は、2018年1月11日に出願された米国非仮出願第15/868,489号、2018年1月11日に出願された米国非仮出願第15/868,518号、2018年1月11日に出願された米国非仮出願第15/868,511号、2018年1月11日に出願された米国非仮出願第15/868,505号、及び2017年12月28日に出願された米国仮出願第62/611,473号の優先権を主張し、これらは全て、その全体が本明細書に引用により明示的に組み込まれている。

**【0002】**

## [背景技術]

コンテンツ管理システムによって、ユーザはネットワークを使用して複数のデバイスでコンテンツアイテムにアクセスし、これらを管理することが可能になる。一部のコンテンツ管理システムは、ユーザがコンテンツアイテムを共有することを可能にし、ユーザがコンテンツアイテムを使用してコラボレーションするのを支援する追加機能を提供し得る。コンテンツ管理システムは通常、コンテンツアイテムをサーバに記憶し、ユーザがネットワーク経由でコンテンツアイテムにアクセスすることを可能にし得る。また、一部のコンテンツ管理システムは、ローカルコピーをクライアントデバイスに記憶して、より自然なインターフェースで（たとえば、ネイティブアプリケーションで、またはクライアントデバイスのファイルシステム内で）コンテンツアイテムへのより高速なアクセスをユーザに提供することを可能にする。加えて、これにより、ユーザはオフラインのときにコンテンツアイテムにアクセスすることが可能になる。コンテンツ管理システムは、いくつかのクライアントデバイス及びサーバ間でコンテンツアイテムのコピーを同期させて、各コピーが同一になるようにすることを試みる。しかしながら、コンテンツアイテムの同期は難しく、多くの技術的障害が伴う。

**【0003】**

本技術の上記及び他の利点及び特徴は、添付の図面に示す特定の実施態様を参照することにより明らかになる。当業者であれば、これらの図面が本技術の一部の例を示すにすぎず、本技術の範囲をこれらの例に限定するものではないことを理解するであろう。さらに、当業者であれば、添付の図面の使用を通じて、さらに具体的かつ詳細に記載及び説明する本技術の原理を理解するであろう。

**【図面の簡単な説明】****【0004】**

【図1】一部の実施形態による、コンテンツ管理システム及びクライアントデバイスの一例を示す図である。

【図2】一部の実施形態による、クライアント同期サービスの一例を示す図である。

【図3】様々な実施形態による、ツリーデータ構造の一例を示す図である。

【図4】様々な実施形態による、ツリーデータ構造の一例を示す図である。

【図5】本主題技術の様々な実施形態による、ツリーデータ構造を使用してサーバ状態とファイルシステム状態とを同期させるための例示的な方法を示す図である。

10

20

30

40

50

【図 6】本主題技術の様々な実施形態による、ツリーデータ構造を使用してサーバ状態とファイルシステム状態とを同期させる場合の競合を解決するための例示的な方法を示す図である。

【図 7】様々な実施形態による、追加操作に関するルールの違反を示すツリーデータ構造の一例の図である。

【図 8】本主題技術の様々な実施形態による、サーバ状態とファイルシステム状態とを段階的に収束させるための例示的な方法を示す図である。

【図 9】様々な実施形態による、ツリーデータ構造の一例を示す図である。

【図 10】例示的なシナリオを示す図である。

【図 11】本主題技術の様々な実施形態による、2つの操作プランの例示的なベン図表現である。

10

【図 12】本主題技術の様々な実施形態による、操作プランの変更を管理するための例示的な方法の図である。

【図 13】本主題技術の様々な実施形態による、ファイル名配列及びハッシュインデックス配列の実例の図である。

【図 14】本主題技術の様々な実施形態による、ファイル名を記憶するための例示的な方法を示す図である。

【図 15】本主題技術の様々な実施形態による、ファイル名が与えられた場合にファイル名の場所を取得するための例示的な方法を示す図である。

【図 16】図 16 A 及び図 16 B は、様々な実施形態による、ツリーデータ構造の例を示す図である。

20

【図 17】様々な実施形態による、ツリーデータ構造の一例を示す図である。

【図 18】本主題技術の様々な実施形態による、ファイル名が与えられた場合にファイル名の場所を取得するための例示的な方法を示す図である。

【図 19】本技術の特定の態様を実装するためのシステムの一例を示す図である。

【発明を実施するための形態】

【0005】

以下、本技術の様々な例を詳細に説明する。特定の実施態様を説明するが、これは説明のみを目的として行っていることを理解されたい。関連技術の当業者であれば、本技術の精神及び範囲から逸脱することなく、他のコンポーネント及び構成が使用され得ることを認識するであろう。

30

【0006】

コンピューティング技術及びネットワーク技術の様々な進歩により、コンテンツ管理システムは、ユーザに複数のデバイスによるコンテンツアイテムへのアクセスを提供できるようになった。コンテンツアイテムには、限定はしないが、ファイル、ドキュメント、メッセージ（たとえば、電子メールメッセージもしくはテキストメッセージ）、メディアファイル（たとえば、写真、ビデオ、及び音声ファイル）、他のコンテンツアイテムを含むフォルダ、または他の任意の単位のコンテンツが含まれ得る。コンテンツアイテムは、複数のユーザと共有、編集、削除、追加、名前変更、または移動され得る。しかしながら、これらのコンテンツアイテムを複数のコンピューティングデバイス（たとえば、サーバ及びクライアントデバイス）ならびにいくつかのユーザアカウントにわたって同期させることは不完全なままであり、技術的障害が多い。

40

【0007】

技術的障害の一部を説明すると、第1のマシン（たとえば、クライアントデバイスまたはサーバ）は、ユーザがコンテンツ管理システムによって管理されるコンテンツアイテムをどのように修正したかに関する情報を提供する通信を第2のマシンに送信し得る。これらの通信を第2のマシンが使用して、第2のマシン上のコンテンツアイテムを同期させることによって、第1のマシン上のコンテンツアイテムに対して実行されたアクションが第2のマシン上のコンテンツアイテムに反映され、第1のマシン上のコンテンツアイテムが第2のマシン上のコンテンツアイテムと実質的に同一になるようにし得る。

50

## 【 0 0 0 8 】

しかしながら、いくつかの通信が送信され得、それらの通信は、通信の送信に使用される1つまたは複数のネットワークにより使用される様々なネットワークルーティングプロトコル、第1または第2のマシンの技術的動作、または他の理由の結果として、順序が狂って受信され得る。さらに、ユーザは、多数のコンテンツアイテムに多数の修正を実行していたり、以前の修正を短時間で元に戻したり、以前に修正したコンテンツアイテムまたはコンテンツアイテムのセットに対して追加の修正をすぐに実行したりし得る。これにより、これらの通信が順序が狂って受信されたり、特定の通信が最新でなくなったり、第2のマシンが最新ではないコンテンツアイテムへの操作を実行したりする可能性が高まる。その結果、操作の多くがコンテンツアイテムの現在の状態に適合しない場合がある。実際には、一部の操作が他の操作またはコンテンツアイテムの現在の状態と競合しているか否かを検出することさえ困難な場合がある。

10

## 【 0 0 0 9 】

加えて、同期アクションに関する固有の遅延がある。たとえば、第1のマシン上で取られたアクションが最初に第1のマシンによって検出され、通信が生成され、その後ネットワークを介して送信される。この通信は第2のマシンによって受信され、第2のマシンは、以前の通信をまだ処理しており、通信に詳述されたアクションを実行している場合がある。この例示的なシナリオでは、第1のマシン、第2のマシン、及び/またはネットワークの限られたコンピューティングリソース（たとえば、帯域幅、メモリ、処理時間、処理サイクルなど）によって遅延が導入されるいくつかのポイントがある。遅延が増加するにつれて、通信が何らかの理由でコンテンツアイテムの現在の状態と競合する可能性が高くなる。さらに、これらの競合する通信を処理して競合を解決すると、たとえば、処理時間、メモリ、エネルギー、または帯域幅などの不要なコンピューティングリソースが消費され、遅延がさらに増加する。

20

## 【 0 0 1 0 】

事態をさらに複雑にすることに、第2のマシン及び/またはコンテンツアイテムにアクセス可能なさらなるマシンの同じまたは異なるユーザが、コンテンツアイテムへの修正を実行している場合もある。その結果、上記の問題は何倍にもなり得、ローカルアクションがリモートアクションと競合するか否か、及び/またはローカルアクションが最新のコンテンツアイテムに作用しているか否かに関するさらなる技術的問題が生じる。

30

## 【 0 0 1 1 】

開示する技術は、上記及び他の技術的問題に対する技術的解決方法を提供するコンテンツ管理システムのためのクライアント同期サービスに対する当技術分野における必要性に対処する。クライアント同期サービスは、クライアントデバイス上で動作し、コンテンツ管理システムのサーバ上のコンテンツアイテムと、クライアントデバイス上の対応するコンテンツアイテムとの間の同期の不一致を特定するように構成され得る。同期の不一致ごとに、クライアント同期サービスは、コンテンツアイテムの同期に必要な操作を特定し、それらの操作を開始し得る。

## 【 0 0 1 2 】

クライアント同期サービスは、ツリーデータ構造（「ツリー」）のセットを使用して、サーバ上のコンテンツアイテムのステータス、クライアントデバイス上のコンテンツアイテムのステータス、及びそれらの同期状態を追跡し得る。一部の実施形態によれば、3つのツリーのセットが使用され得る。3つのツリーは、サーバ状態を表すリモートツリーと、クライアントデバイス上のファイルシステム状態を表すローカルツリーと、ローカルツリー及びリモートツリーのマージベースを表す同期ツリーとを含み得る。マージベースは、ローカルツリー及びリモートツリーの共通の祖先、またはローカルツリー及びリモートツリーの間最後の既知の同期した状態と考えられ得る。したがって、クライアント同期サービスは、3つのツリー（たとえば、リモートツリー、同期ツリー、ローカルツリー）が全て同一である場合に、サーバ状態とクライアントデバイス状態とが同期していると判定し得る。

40

50

## 【 0 0 1 3 】

コンテンツアイテムのサーバ状態、またはコンテンツアイテムのクライアントデバイスファイルシステム状態（「ファイルシステム状態」）への修正が検出された場合、クライアント同期サービスは適切なツリーを更新し、ツリーの三つ組に基づいてサーバ状態とファイルシステム状態とが同期しているか否かを判定する。ツリーの1つへの更新に基づいて、サーバ状態及びファイルシステム状態は同期され、非同期化され、またはさらに非同期化され得る。サーバ状態とファイルシステム状態とが同期していない場合、クライアント同期サービスは、サーバ状態とファイルシステム状態とを収束させ、サーバ状態とファイルシステム状態とを同期した状態に近づけるために必要な初期操作セットを少なくともも特定し得る。

10

## 【 0 0 1 4 】

ツリーデータ構造のセットを頼りにサーバ状態とファイルシステム状態を監視することにより、様々な技術的問題に対するコンピューティング技術に根ざした代替手段及び/または解決方法を提供する。たとえば、クライアント同期サービスは、サーバ状態及びファイル状態を追跡し、2つの状態のマージベースの表現を記憶することができる。その結果、本主題技術の様々な実施形態は、ユーザがリモートでコンテンツアイテムを修正している方法を示すいくつかの通信を受信し、これらの修正がローカルで実行されるべき順序を決定し、修正が他の修正と競合するまたは最新でないか否か、及びリモートの修正がユーザによりローカルで実行されるローカルの修正と競合するか否かを判定することに関連する技術的問題を回避する。これらの問題の多くは、他の解決方法が、関連する様々な当事者（たとえば、サーバ及びクライアントデバイス）の状態を追跡することができず、状態が同期しているか否かを迅速に判定することができないことから生じる。代わりに、これらの他の解決方法は、サーバ状態とファイルシステム状態とが同期しているか否かのコンテキストなしで、コンテンツアイテムをローカルで修正する方法に関する命令を受信することに依存している。

20

## 【 0 0 1 5 】

さらに、サーバ状態及びファイルシステム状態は継続的に監視されるので、それらが同期しているか否かを判定することは、手順の複雑さと計算時間及びリソースとの観点ではるかに効率的である。以下でさらに詳細に説明するように、クライアント同期サービスにより、より決定的な方法でのサーバ状態とファイルシステム状態との段階的かつ順序立った同期が可能になる。その結果、コンテンツ管理システムの機能のスケールアップ及びテストもより効率的になる。

30

## 【 0 0 1 6 】

## コンテンツ管理システム

一部の実施形態では、開示する技術は、とりわけコンテンツアイテム同期機能及びコラボレーション機能を有するコンテンツ管理システムのコンテキストで配備される。例示的なシステム構成100を図1Aに示しており、図1Aは、クライアントデバイス150とインタラクションを行うコンテンツ管理システム110を示している。

## 【 0 0 1 7 】

## アカウント

40

## 【 0 0 1 8 】

コンテンツ管理システム110は、コンテンツアイテムをアカウントに関連付けて記憶すると共に、コンテンツアイテム（複数可）の取り出し、修正、閲覧、及び/または共有などの多様なコンテンツアイテム管理タスクを実行することができる。さらに、コンテンツ管理システム110は、アカウントが複数のクライアントデバイスからコンテンツアイテム（複数可）にアクセス可能にすることができる。

## 【 0 0 1 9 】

コンテンツ管理システム110は、複数のアカウントをサポートする。エンティティ（ユーザ、ユーザのグループ、チーム、会社など）は、コンテンツ管理システムによってアカウントを作成することができ、アカウントの詳細をアカウントデータベース140に記

50

憶することができる。アカウントデータベース140は、登録されたエンティティのプロフィール情報を記憶することができる。一部のケースでは、登録されたエンティティのプロフィール情報には、ユーザ名及び/または電子メールアドレスが含まれる。アカウントデータベース140はアカウント管理情報、たとえば、アカウントタイプ(たとえば、様々な階層の無料または有料アカウント)、割り当てられたストレージスペース、使用されたストレージスペース、登録コンテンツ管理クライアントアプリケーション152がその上に常駐するクライアントデバイス150、セキュリティ設定、個人構成設定などを含むことができる。

#### 【0020】

アカウントデータベース140は、エンティティに関連付けられたアカウントのグループを記憶することができる。グループは、グループポリシー及び/またはアクセス制御リストに基づいた権限を有することができる、グループのメンバーは権限を継承することができる。たとえば、マーケティンググループは、あるコンテンツアイテムのセットにアクセスすることができ、エンジニアリンググループは、他のコンテンツアイテムのセットにアクセスすることができる。管理者グループは、グループの修正、ユーザアカウントの修正などを行うことができる。

10

#### 【0021】

コンテンツアイテムの記憶

#### 【0022】

コンテンツ管理システム110の1つの機能はコンテンツアイテムの記憶であり、コンテンツアイテムはコンテンツストレージ142に記憶することができる。コンテンツアイテムは任意のデジタルデータ、たとえば、ドキュメント、コラボレーションコンテンツアイテム、テキストファイル、音声ファイル、画像ファイル、ビデオファイル、ウェブページ、実行可能ファイル、バイナリファイルなどとすることができる。また、コンテンツアイテムは、コレクションを含むことができ、またはコンテンツアイテム同士を異なる挙動と共にグループ化するための他のメカニズム、たとえば、フォルダ、zipファイル、プレイリスト、アルバムなどを含むことができる。コレクションは、フォルダ、または共通の属性によって関連付けまたはグループ化された複数のコンテンツアイテムを指す場合がある。一部の実施形態では、コンテンツストレージ142は、特定の機能に対処するために他のタイプのストレージまたはデータベースと組み合わせられる。コンテンツストレージ142はコンテンツアイテムを記憶することができ、コンテンツアイテムに関するメタデータはメタデータベース146に記憶することができる。同様に、コンテンツアイテムがコンテンツストレージ142のどこに記憶されているかに関するデータは、コンテンツディレクトリ144に記憶することができる。加えて、変更、アクセスなどに関するデータは、サーバファイルジャーナル148に記憶することができる。コンテンツストレージ142、コンテンツディレクトリ144、サーバファイルジャーナル148、及びメタデータベース146などの様々なストレージ/データベースのそれぞれは、複数のそのようなストレージまたはデータベースから構成することができ、多数のデバイス及び場所に分散させることができる。他の構成も可能である。たとえば、コンテンツストレージ142、コンテンツディレクトリ144、サーバファイルジャーナル148、及び/またはメタデータベース146からのデータは、1つまたは複数のコンテンツストレージまたはデータベースに結合され得、または追加のコンテンツストレージまたはデータベースにさらにセグメント化され得る。このように、コンテンツ管理システム110は、図1に示すよりも多いまたは少ないストレージ及び/またはデータベースを含み得る。

20

30

40

#### 【0023】

一部の実施形態では、コンテンツストレージ142は、少なくとも1つのコンテンツストレージサービス116に関連付けられ、これは、記憶すべきコンテンツアイテムの受信、記憶すべきコンテンツアイテムの準備、コンテンツアイテムの記憶場所の選択、ストレージからのコンテンツアイテムの取り出しなどを含むがこれらに限定されない、コンテンツアイテムの記憶を管理するためのソフトウェアまたは他のプロセッサ実行可能命令を含

50

む。一部の実施形態において、コンテンツストレージサービス 1 1 6 は、コンテンツアイテムをより小さいチャンクに分割してコンテンツストレージ 1 4 2 に記憶することができる。コンテンツアイテムを構成する各チャンクの場所は、コンテンツディレクトリ 1 4 4 に記録することができる。コンテンツディレクトリ 1 4 4 は、コンテンツストレージ 1 4 2 に記憶されたコンテンツアイテムごとにコンテンツエントリを含むことができる。コンテンツエントリは、コンテンツアイテムを識別する一意の ID に関連付けることができる。

【 0 0 2 4 】

一部の実施形態では、コンテンツディレクトリ 1 4 4 内のコンテンツアイテムを識別する一意の ID は、決定論的ハッシュ関数から導出することができる。コンテンツアイテムの一意の ID を導出する方法により、コンテンツアイテムの複製がそのようなものとして認識されるようにすることができ、その理由は、決定論的ハッシュ関数が、同じコンテンツアイテムの全てのコピーに対して同じ識別子を出力するが、異なるコンテンツアイテムには異なる識別子を出力するためである。この方法論を使用して、コンテンツストレージサービス 1 1 6 は、コンテンツアイテムごとに一意の ID を出力することができる。

【 0 0 2 5 】

コンテンツストレージサービス 1 1 6 は、メタデータデータベース 1 4 6 内のコンテンツアイテムのコンテンツパスを指定または記録することもできる。コンテンツパスは、コンテンツアイテムの名前及び/またはコンテンツアイテムに関連付けられたフォルダ階層を含むことができる。たとえば、コンテンツパスは、クライアントデバイス上のローカルファイルシステムにおいてコンテンツアイテムが記憶されたフォルダまたはフォルダのパスを含むことができる。コンテンツアイテムはコンテンツストレージ 1 4 2 にブロックで記憶され、ツリー状のディレクトリ構造の下に記憶されなくてもよいが、そのようなディレクトリ構造は、ユーザにとって快適な移動構造である。コンテンツストレージサービス 1 1 6 は、コンテンツアイテムのコンテンツパスを定義または記録ことができ、ディレクトリ構造の「ルート」ノードは、アカウントごとの名前空間とすることができる。名前空間内には、アカウントのユーザ及び/またはコンテンツストレージサービス 1 1 6 によって定義されるディレクトリ構造が存在することができる。メタデータデータベース 1 4 6 は、コンテンツアイテムごとのコンテンツパスをコンテンツエントリの一部として記憶することができる。

【 0 0 2 6 】

一部の実施形態では、名前空間は、ディレクトリ構造内にネストされた追加の名前空間を、あたかもそれらがルートノード内に記憶されているかのように含むことができる。これは、アカウントが共有されたコレクションにアクセス可能な場合に生じ得る。共有されたコレクションには、コンテンツ管理システム 1 1 0 内で独自の名前空間を割り当てることができる。いくつかの共有されたコレクションは、実際には共有されたコレクションのルートノードであるが、ディレクトリ構造内のアカウントの名前空間の下位に配置され、アカウントのフォルダ内のフォルダとして表示することができる。上記のように、ディレクトリ構造は、単にユーザにとって快適な移動構造であるが、コンテンツストレージ 1 4 2 内のコンテンツアイテムの記憶場所と相互関係がない。

【 0 0 2 7 】

アカウントがコンテンツアイテムを見るディレクトリ構造は、コンテンツ管理システム 1 1 0 の記憶場所と相互関係がないが、ディレクトリ構造は、クライアントデバイス 1 5 0 によって使用されるファイルシステムによっては、クライアントデバイス 1 5 0 上の記憶場所と相互関係があり得る。

【 0 0 2 8 】

上記のように、コンテンツディレクトリ 1 4 4 内のコンテンツエントリは、コンテンツアイテムを構成する各チャンクの場所を含むこともできる。より具体的には、コンテンツエントリは、コンテンツアイテムを構成するチャンクのコンテンツストレージ 1 4 2 内の場所を特定するコンテンツポインタを含むことができる。

【 0 0 2 9 】

10

20

30

40

50

コンテンツパス及びコンテンツポインタに加えて、コンテンツディレクトリ 1 4 4 内のコンテンツエントリは、コンテンツアイテムにアクセス可能なユーザアカウントを識別するユーザアカウント識別子、及び/またはコンテンツエントリにアクセス可能なグループを識別するグループ識別子、及び/またはコンテンツエントリが属する名前空間も含むことができる。

**【 0 0 3 0 】**

コンテンツストレージサービス 1 1 6 は、コンテンツアイテムまたはコンテンツアイテムのバージョンを構成する重複コンテンツアイテムまたは重複ブロックを特定することにより、必要なストレージスペースの量を減らすことができる。複数のコピーを記憶する代わりに、コンテンツストレージ 1 4 2 は、コンテンツアイテムまたはコンテンツアイテムのブロックの単一のコピーを記憶することができ、コンテンツディレクトリ 1 4 4 は、複製を単一のコピーにリンクさせるポインタまたは他のメカニズムを含むことができる。

10

**【 0 0 3 1 】**

コンテンツストレージサービス 1 1 6 は、コンテンツアイテム、コンテンツアイテムタイプ、フォルダ、ファイルパス、及び/またはコンテンツアイテムと様々なアカウント、コレクション、もしくはグループとの関係を記述するメタデータをメタデータベース 1 4 6 に、コンテンツアイテムの一意的 ID に関連付けて記憶することもできる。

**【 0 0 3 2 】**

コンテンツストレージサービス 1 1 6 は、変更、アクセスなどに関するデータのログをサーバファイルジャーナル 1 4 8 に記憶することもできる。サーバファイルジャーナル 1 4 8 は、コンテンツアイテムの一意的 ID と、変更またはアクセスアクションの説明とを、タイムスタンプまたはバージョン番号及び他の任意の関連データと共に含むことができる。サーバファイルジャーナル 1 4 8 は、変更またはコンテンツアイテムアクセスによって影響を受けるブロックへのポインタも含むことができる。コンテンツストレージサービスは、コンテンツアイテムへの変更、異なるバージョンのコンテンツアイテム（分岐したバージョンツリーを含む）、サーバファイルジャーナル 1 4 8 から取得できる変更履歴を追跡するコンテンツアイテムバージョン管理を使用して、操作を元に戻す機能を提供することができる。

20

**【 0 0 3 3 】**

コンテンツアイテムの同期

30

**【 0 0 3 4 】**

コンテンツ管理システム 1 1 0 の他の機能は、コンテンツアイテムと少なくとも 1 つのクライアントデバイス 1 5 0 との同期である。クライアントデバイス（複数可）は、様々な形式を取り、様々な機能を有することができる。たとえば、クライアントデバイス 1 5 0 1 は、その上に常駐する複数のアプリケーションによってアクセス可能なローカルファイルシステムを有するコンピューティングデバイスである。クライアントデバイス 1 5 0 2 は、コンテンツアイテムが、特定のアプリケーションのみ、または特定のアプリケーションによって与えられた権限によってのみアクセス可能であり、コンテンツアイテムが通常、アプリケーション固有のスペースまたはクラウドのいずれかに記憶されるコンピューティングデバイスである。クライアントデバイス 1 5 0 3 は、ウェブブラウザを介してコンテンツ管理システム 1 1 0 にアクセスし、ウェブインターフェースを介してコンテンツアイテムにアクセスする任意のクライアントデバイスである。例示的なクライアントデバイス 1 5 0 1、1 5 0 2、及び 1 5 0 3 は、ラップトップ、モバイルデバイス、またはウェブブラウザなどのフォームファクタで図示しているが、それらの説明はこれらの例示的なフォームファクタのデバイスに限定されないことを理解されたい。たとえば、クライアント 1 5 0 2 などのモバイルデバイスは、その上に常駐する複数のアプリケーションによってアクセス可能なローカルファイルシステムを有し得、またはクライアント 1 5 0 2 は、ウェブブラウザを介してコンテンツ管理システム 1 1 0 にアクセスし得る。したがって、フォームファクタは、クライアント 1 5 0 の機能を検討する場合の限定と見なされるべきではない。クライアントデバイス 1 5 0 に関して本明細書に記載の 1 つまたは複数の機

40

50

能は、デバイスの特定の機能に応じて、全てのクライアントデバイスで利用可能であってもなくてもよく、ファイルアクセスモデルはそのような機能の1つである。

【0035】

多くの実施形態では、クライアントデバイスは、コンテンツ管理システム110のアカウントに関連付けられるが、一部の実施形態では、クライアントデバイスは、共有されたリンクを使用してコンテンツにアクセスすることができ、アカウントを必要としない。

【0036】

上記のように、一部のクライアントデバイスは、ウェブブラウザを使用してコンテンツ管理システム110にアクセスすることができる。しかしながら、クライアントデバイスは、クライアントデバイス150に記憶され、その上で動作するクライアントアプリケーション152を使用してコンテンツ管理システム110にアクセスすることもできる。クライアントアプリケーション152は、クライアント同期サービス156を含むことができる。

10

【0037】

クライアント同期サービス156は、サーバ同期サービス112と通信して、クライアントデバイス150とコンテンツ管理システム110との間でコンテンツアイテムへの変更を同期させることができる。

【0038】

クライアントデバイス150は、クライアント同期サービス156を介してコンテンツをコンテンツ管理システム110と同期させることができる。同期はプラットフォーム非依存にすることができる。すなわち、様々なタイプ、機能、オペレーティングシステムなどの複数のクライアントデバイスにわたってコンテンツを同期させることができる。クライアント同期サービス156は、クライアントデバイス150のファイルシステムの指定された場所にあるコンテンツアイテムの任意の変更（新規の、削除された、修正された、コピーされた、または移動されたコンテンツアイテム）を同期させることができる。

20

【0039】

コンテンツアイテムは、クライアントデバイス150からコンテンツ管理システム110に、及びその逆に同期させることができる。同期がクライアントデバイス150からコンテンツ管理システム110へ行われる実施形態では、ユーザはクライアントデバイス150のファイルシステムから直接コンテンツアイテムを操作することができ、クライアント同期サービス156は、監視しているフォルダ内のファイルへの変更に関してクライアントデバイス150上のディレクトリを監視することができる。

30

【0040】

クライアント同期サービス156が、監視しているディレクトリ内のコンテンツの書き込み、移動、コピー、または削除を検出した場合、クライアント同期サービス156は、変更をコンテンツ管理システムサービス116へ同期させることができる。一部の実施形態では、クライアント同期サービス156は、たとえば、コンテンツアイテムをブロックに分割する、コンテンツアイテムをハッシュ化して一意の識別子を生成するなど、上記の機能を含む、コンテンツ管理システムサービス116の一部の機能を実行することができる。クライアント同期サービス156は、クライアントストレージインデックス164内のコンテンツのインデックスを作成し、その結果をストレージインデックス164に保存することができる。インデックス作成は、コンテンツアイテムごとに、パスに加え、一意のサーバ識別子及び一意のクライアント識別子を記憶することを含むことができる。一部の実施形態では、クライアント同期サービス156は、サーバ同期サービス112から一意のサーバ識別子を知り、クライアントデバイス150のオペレーティングシステムから一意のクライアント識別子を知る。

40

【0041】

クライアント同期サービス156は、ストレージインデックス164を使用して、クライアントストレージ内のコンテンツの少なくとも一部と、コンテンツ管理システム110上のユーザアカウントに関連付けられたコンテンツとの同期を容易にすることができる。

50

たとえば、クライアント同期サービス 156 は、ストレージインデックス 164 をコンテンツ管理システム 110 と比較し、クライアントストレージ上のコンテンツと、コンテンツ管理システム 110 上のユーザアカウントに関連付けられたコンテンツとの差異を検出することができる。次いで、クライアント同期サービス 156 は、クライアントストレージ上のコンテンツを適宜アップロード、ダウンロード、修正、及び削除することにより、差異の調整を試みることができる。コンテンツストレージサービス 116 は、コンテンツアイテムの変更されたまたは新たなブロックを記憶し、サーバファイルジャーナル 148、メタデータデータベース 146、コンテンツディレクトリ 144、コンテンツストレージ 142、アカウントデータベース 140などを適宜更新することができる。

#### 【0042】

コンテンツ管理システム 110 からクライアントデバイス 150 へ同期させる場合、サーバファイルジャーナル 148 に記録されたコンテンツアイテムのマウント、修正、追加、削除、移動は、通知サービス 117 を使用してクライアントデバイス 150 に送信される通知をトリガーすることができる。クライアントデバイス 150 に変更が通知された場合、クライアントデバイスに知られている最後の同期点以後のサーバファイルジャーナル 148 にリストされているリクエストが変更される。クライアントデバイス 150 がコンテンツ管理システム 110 と同期していないと判定した場合、クライアント同期サービス 156 は、変更を含むコンテンツアイテムブロックを要求し、変更されたコンテンツアイテムのローカルコピーを更新する。

#### 【0043】

一部の実施形態では、ストレージインデックス 164 は、あるツリーがサーバ同期サービス 112 によるディレクトリの最新の表現を反映し、他のツリーがクライアント同期サービス 156 によるディレクトリの最新の表現を反映する、ツリーデータ構造を記憶する。クライアント同期サービスは、サーバ同期サービス 112 にデータを要求するか、またはクライアントデバイス 150 上の変更をコンテンツ管理システム 110 に対してコミットすることにより、ツリー構造が一致するように機能することができる。

#### 【0044】

クライアントデバイス 150 は、利用可能なネットワーク接続を有さなくてもよいこともある。このシナリオでは、クライアント同期サービス 156 は、コンテンツアイテムの変更についてリンクされたコレクションを監視し、それらの変更をキューに入れて、後でネットワーク接続が利用可能なときにコンテンツ管理システム 110 へ同期させるようにすることができる。同様に、ユーザはコンテンツ管理システム 110 との同期を手動で開始、停止、一時停止、または再開することができる。

#### 【0045】

クライアント同期サービス 156 は、コンテンツ管理システム 110 上の特定のユーザアカウントに関連付けられた全てのコンテンツを同期させることができる。代替的には、クライアント同期サービス 156 は、コンテンツ管理システム 110 上の特定のユーザアカウントに関連付けられた全コンテンツのうちの一部のコンテンツの一部を選択的に同期させることができる。コンテンツの一部のみを選択的に同期させることにより、クライアントデバイス 150 上のスペースを保存し、帯域幅を節約することができる。

#### 【0046】

一部の実施形態では、クライアント同期サービス 156 は、特定のユーザアカウントに関連付けられたコンテンツの一部を選択的に記憶し、コンテンツの残りの部分についてプレースホルダーコンテンツアイテムをクライアントストレージに記憶する。たとえば、クライアント同期サービス 156 は、コンテンツ管理システム 110 上のそれぞれの完全なコンテンツアイテムと同じファイル名、パス、拡張子、メタデータを有するが、完全なコンテンツアイテムのデータを欠いているプレースホルダーコンテンツアイテムを記憶することができる。プレースホルダーコンテンツアイテムは、サイズが数バイト以下にすることができるが、それぞれの完全なコンテンツアイテムは非常に大きい場合がある。クライアントデバイス 150 がコンテンツアイテムへのアクセスを試みた後、クライアント同期

10

20

30

40

50

サービス156は、コンテンツ管理システム110からコンテンツアイテムのデータを取り出し、完全なコンテンツアイテムをアクセスしているクライアントデバイス150に提供することができる。この手法により、コンテンツ管理システム110上のユーザのコンテンツへのフルアクセスをなおも提供しながら、大幅なスペース及び帯域幅の節約を提供することができる。

【0047】

コラボレーション機能

【0048】

コンテンツ管理システム110の他の機能は、ユーザ間のコラボレーションを容易にすることである。コラボレーション機能には、コンテンツアイテムの共有、コンテンツアイテムへのコメント、コンテンツアイテムへの共同作業、インスタントメッセージング、コンテンツアイテムに関するプレゼンス及び閲覧状態情報の提供などが含まれる。

10

【0049】

共有

【0050】

コンテンツ管理システム110は、共有サービス128を介したコンテンツの共有を管理することができる。コンテンツへのリンクを提供することによるコンテンツの共有は、コンテンツ管理システム110とネットワーク通信する任意のコンピューティングデバイスからコンテンツアイテムにアクセス可能にすることを含むことができる。しかしながら、一部の実施形態では、リンクは、コンテンツ管理システム110及びアクセス制御リスト145によって実施されるアクセス制限に関連付けることができる。コンテンツの共有は、共有サービス128を使用してコンテンツをリンクさせることによって、コンテンツ管理システム110内のコンテンツを（コンテンツアイテムに関連付けられた元のユーザアカウントに加えて）少なくとも1つの追加のユーザアカウントと共有して、各ユーザアカウントがコンテンツアイテムにアクセス可能になるようにすることも含むことができる。追加のユーザアカウントは、コンテンツを承認することによってコンテンツにアクセスすることができ、コンテンツはその後、ウェブインターフェースサービス124を介して、またはクライアントデバイス150上のそれらのアカウントに関連付けられたディレクトリ構造内から直接アクセス可能になる。共有は、プラットフォームに依存しない方法で実行することができる。すなわち、様々なタイプ、機能、オペレーティングシステムなどの複数のクライアントデバイス150間でコンテンツを共有することができる。また、様々なタイプのユーザアカウント間でコンテンツを共有することもできる。

20

30

【0051】

コンテンツ管理システム110内でコンテンツアイテムを共有するために、共有サービス128は、コンテンツアイテムに関連付けられたアクセス制御リストデータベース145内のコンテンツエントリにユーザアカウント識別子または複数のユーザアカウント識別子を追加して、追加されたユーザアカウントにコンテンツアイテムへのアクセスを付与することができる。また、共有サービス128は、コンテンツエントリからユーザアカウント識別子を除去して、ユーザアカウントによるコンテンツアイテムへのアクセスを制限することもできる。共有サービス128は、コンテンツアイテム識別子、コンテンツアイテムへのアクセスが許可されたユーザアカウント識別子、及びアクセス制御リストデータベース145内のアクセスレベルを記録することができる。たとえば、一部の実施形態では、単一のコンテンツエントリに関連付けられたユーザアカウント識別子は、関連付けられたコンテンツアイテムに関してそれぞれのユーザアカウント識別子に異なる権限を指定することができる。

40

【0052】

コンテンツ管理システム110の外部でコンテンツアイテムを共有するために、共有サービス128は、任意のウェブブラウザが認証なしでコンテンツ管理システム110のコンテンツアイテムまたはコレクションにアクセスすることを可能にする、ユニフォームリソースロケータ（URL）などのカスタムネットワークアドレスを生成することができる

50

。これを実現するために、共有サービス128は、生成されたURLにコンテンツ識別データを含めることができ、これを後で使用して、要求されたコンテンツアイテムを適切に識別し、返すことができる。たとえば、共有サービス128は、アカウント識別子及びコンテンツパスまたはコンテンツアイテム識別コードを生成されたURLに含めることができる。URLが選択されると、URLに含まれるコンテンツ識別データをコンテンツ管理システム110に送信することができ、コンテンツ管理システム110は、受信したコンテンツ識別データを使用して適切なコンテンツアイテムを識別し、コンテンツアイテムを返すことができる。

#### 【0053】

URLの生成に加えて、共有サービス128は、コンテンツアイテムへのURLが作成されたことをアクセス制御リストデータベース145に記録するように構成することもできる。一部の実施形態では、コンテンツアイテムに関連付けられたコンテンツエントリは、コンテンツアイテムへのURLが作成されたか否かを示すURLフラグを含むことができる。たとえば、URLフラグは、コンテンツアイテムへのURLが作成されていないことを示す0すなわち偽に初期設定されたブール値とすることができる。共有サービス128は、コンテンツアイテムへのURLを生成した後に、フラグの値を1すなわち真に変更することができる。

10

#### 【0054】

一部の実施形態では、共有サービス128は、権限のセットをコンテンツアイテムのURLに関連付けることができる。たとえば、ユーザがURLを介してコンテンツアイテムにアクセスしようとする場合、共有サービス128は、限られた権限のセットをコンテンツアイテムに提供することができる。制限された権限の例には、ユーザがコンテンツアイテムをダウンロードできないこと、コンテンツアイテムを保存できないこと、コンテンツアイテムをコピーできないこと、コンテンツアイテムを修正できないことなどの制限が含まれる。一部の実施形態では、制限された権限には、指定されたドメインから、すなわち企業ネットワークドメイン内から、または指定されたドメインに関連付けられたアカウント、たとえば、企業アカウント（たとえば、@acme.com）に関連付けられたアカウントによってのみコンテンツアイテムにアクセスすることを許可する制限が含まれる。

20

#### 【0055】

一部の実施形態では、共有サービス128は、生成されたURLを非アクティブ化するように構成することもできる。たとえば、各コンテンツエントリは、生成されたURLからのリクエストに応じてコンテンツが返されるべきか否かを示すURLアクティブフラグを含むこともできる。たとえば、共有サービス128は、URLアクティブフラグが1すなわち真に設定されている場合にのみ、生成されたリンクによって要求されたコンテンツアイテムを返すことができる。このように、URLアクティブフラグの値を変更することにより、URLが生成されたコンテンツアイテムへのアクセスを簡単に制限することができる。これにより、ユーザは、コンテンツアイテムを移動したり、生成されたURLを削除したりする必要なく、共有されたコンテンツアイテムへのアクセスを制限することが可能になる。同様に、共有サービス128は、URLアクティブフラグの値を再度1すなわち真に変更することにより、URLを再アクティブ化することができる。このようにして、ユーザは、新たなURLを生成する必要なく、コンテンツアイテムへのアクセスを簡単に復元することができる。

30

40

#### 【0056】

一部の実施形態では、コンテンツ管理システム110は、コンテンツアイテムをアップロードするためのURLを指定することができる。たとえば、ユーザアカウントを有する第1のユーザは、そのようなURLを要求して、そのURLを投稿ユーザに提供し、投稿ユーザはそのURLを使用してコンテンツアイテムを第1のユーザのユーザアカウントにアップロードすることができる。

#### 【0057】

チームサービス

50

## 【 0 0 5 8 】

一部の実施形態では、コンテンツ管理システム 1 1 0 はチームサービス 1 3 0 を含む。チームサービス 1 3 0 は、定義されたユーザアカウントのチームを作成及び管理するための機能を提供することができる。会社に対してチームを、サブチーム（たとえば、ビジネスユニット、またはプロジェクトチームなど）と共に作成し、ユーザアカウントをチーム及びサブチームに割り当てることができる。または任意の定義されたユーザアカウントのグループに対してチームを作成することができる。チームサービス 1 3 0 は、チーム用の共通の共有スペース、プライベートユーザアカウントフォルダ、及びアクセス制限された共有フォルダを提供することができる。チームサービスは、管理者がチーム内のコレクション及びコンテンツアイテムを管理するための管理インターフェースを提供することもでき、チームに関連付けられたユーザアカウントを管理することができる。

10

## 【 0 0 5 9 】

認可サービス

## 【 0 0 6 0 】

一部の実施形態では、コンテンツ管理システム 1 1 0 は認可サービス 1 3 2 を含む。認可サービス 1 3 2 は、名前空間にアクセスしようとするユーザアカウントが名前空間にアクセスするための適切な権限を有することを確認する。認可サービス 1 3 2 は、名前空間にアクセスするリクエストに続くトークンをクライアントアプリケーション 1 5 2 から受信ことができ、ユーザアカウントに許可された機能を返すことができる。複数レベルのアクセスが可能なユーザアカウント（たとえば、ユーザ権限及び管理者権限を有するユーザアカウント）の場合、認可サービス 1 3 2 は、管理者による意図しないアクションを回避するために、明示的な特権昇格を要求することもできる。

20

## 【 0 0 6 1 】

プレゼンス及び閲覧状態

## 【 0 0 6 2 】

一部の実施形態では、コンテンツ管理システムは、コンテンツアイテムが共有されているユーザが、どのようにそのコンテンツアイテムとインタラクションを行っているか、または行ったかに関する情報を提供することができる。一部の実施形態では、コンテンツ管理システム 1 1 0 は、コンテンツアイテムが共有されているユーザが現在コンテンツアイテムを見ていることを報告することができる。たとえば、クライアントコラボレーションサービス 1 6 0 は、クライアントデバイス 1 5 0 がコンテンツアイテムにアクセスしているときに通知サービス 1 1 7 に通知することができる。次いで、通知サービス 1 1 7 は、同じコンテンツアイテムにアクセス可能な他のユーザの全てのクライアントデバイスに、コンテンツアイテムに関するクライアントデバイス 1 5 0 のユーザのプレゼンスを通知することができる。

30

## 【 0 0 6 3 】

一部の実施形態では、コンテンツ管理システム 1 1 0 は、共有されたコンテンツアイテムとのユーザインタラクションの履歴を報告することができる。コラボレーションサービス 1 2 6 は、メタデータデータベース 1 4 6 及びサーバファイルジャーナル 1 4 8 などのデータソースに問い合わせして、ユーザがコンテンツアイテムを保存したこと、ユーザがコンテンツアイテムをまだ見ていないことなどを特定し、通知サービス 1 1 7 を使用してこのステータス情報を他のユーザに周知することによって、誰がコンテンツアイテムを現在見ているか、もしくは見ていたか、または修正したかが分かるようにすることができる。

40

## 【 0 0 6 4 】

コラボレーションサービス 1 2 6 は、コンテンツアイテムがコメント機能をネイティブにサポートしていない場合でも、コンテンツに関連するコメントを容易にすることができる。そのようなコメントは、メタデータデータベース 1 4 6 に記憶することができる。

## 【 0 0 6 5 】

コラボレーションサービス 1 2 6 は、ユーザへの通知を発信及び送信することができる。たとえば、ユーザはコメントで他のユーザに言及することができ、コラボレーションサ

50

ービス 1 2 6 は、そのユーザがコメントで言及されたという通知をそのユーザに送信することができる。他の様々なコンテンツアイテムイベントは、コンテンツアイテムの削除、コンテンツアイテムの共有などを含む通知をトリガーすることができる。

【 0 0 6 6 】

コラボレーションサービス 1 2 6 は、ユーザがインスタントメッセージ、音声通話、電子メールなどを送受信できるメッセージングプラットフォームを提供することができる。

【 0 0 6 7 】

コラボレーションコンテンツアイテム

【 0 0 6 8 】

一部の実施形態では、コンテンツ管理サービスはコラボレーション型ドキュメントサービス 1 3 4 も含むことができ、これはインタラクティブコンテンツアイテムコラボレーションプラットフォームを提供することができ、それによってユーザは、コラボレーションコンテンツアイテムを同時に作成し、コラボレーションコンテンツアイテムにコメントし、コラボレーションコンテンツアイテム内のタスクを管理することができる。コラボレーションコンテンツアイテムは、ユーザがコラボレーションコンテンツアイテムエディタを使用して作成及び編集できるファイルとすることができ、コラボレーションコンテンツアイテム要素を含むことができる。コラボレーションコンテンツアイテム要素には、コラボレーションコンテンツアイテム識別子、1つまたは複数の作成者識別子、コラボレーションコンテンツアイテムテキスト、コラボレーションコンテンツアイテム属性、インタラクシオン情報、コメント、共有ユーザなどが含まれ得る。コラボレーションコンテンツアイテム要素は、データベースエンティティとして記憶することができ、これにより、コラボレーションコンテンツアイテムの検索及び取り出しが可能になる。複数のユーザが、同時にまたは異なる時間に、コラボレーションコンテンツアイテムにアクセスし、これを見て、編集し、及びコラボレーションし得る。一部の実施形態では、これは、2人のユーザにウェブインターフェースを介してコンテンツアイテムにアクセスするように要求することによって管理することができ、そこで彼らはコンテンツアイテムの同じコピーに対して同時に作業することができる。

【 0 0 6 9 】

コラボレーションコンパニオンインターフェース

【 0 0 7 0 】

一部の実施形態では、クライアントコラボレーションサービス 1 6 0 は、クライアントデバイス 1 5 0 上に提示されているコンテンツアイテムに関連する情報を表示する目的の、ネイティブアプリケーションコンパニオンインターフェースを提供することができる。クライアントデバイス 1 5 0 に記憶され、その上で実行されるネイティブアプリケーションによってコンテンツアイテムがアクセスされる実施形態では、コンテンツアイテムがコンテンツアプリケーション 1 5 2 によって管理されるように、コンテンツアイテムがクライアントデバイス 1 5 0 のファイルシステムの指定された場所にある場合、ネイティブアプリケーションは、上記のコラボレーションデータを表示するネイティブな方法を提供しなくてもよい。そのような実施形態では、クライアントコラボレーションサービス 1 6 0 は、ユーザがコンテンツアイテムを開いたことを検出することができ、コラボレーションデータなどのコンテンツアイテムの追加情報を有するオーバーレイを提供することができる。たとえば、追加情報は、コンテンツアイテムへのコメント、コンテンツアイテムのステータス、コンテンツアイテムを以前見ていたまたは現在見ている他のユーザのアクティビティを含むことができる。そのようなオーバーレイは、他のユーザが現在コンテンツアイテムを編集しているので、変更が失われ得ることをユーザに警告することができる。

【 0 0 7 1 】

一部の実施形態では、パブリックまたはプライベートのアプリケーションプログラミングインターフェースを使用して、上述のサービスまたはストレージ/データベースのうちの1つまたは複数にアクセスすることができる。

【 0 0 7 2 】

10

20

30

40

50

特定のソフトウェアアプリケーションは、ユーザに代わってAPIを介してコンテンツストレージ142にアクセスすることができる。たとえば、クライアントデバイス150上で動作するアプリケーションなどのソフトウェアパッケージは、ユーザが認証資格情報を提供してコンテンツの読み出し、書き込み、作成、削除、共有、または他の操作を行う場合に、プログラムでコンテンツ管理システム110に直接API呼び出しを行うことができる。

**【0073】**

ユーザは、ウェブインターフェースサービス124によって生成及びサービス提供されるウェブインターフェースを介して、ユーザアカウントに記憶されているコンテンツを見るまたは操作することができる。たとえば、ユーザは、ウェブブラウザ内でコンテンツ管理システム110によって提供されるウェブアドレスに移動することができる。ウェブインターフェースを介して行われたコンテンツストレージ142内のコンテンツへの変更または更新、たとえば、コンテンツアイテムの新たなバージョンのアップロードなどは、ユーザのアカウントに関連付けられた他のクライアントデバイスに戻って伝播させることができる。たとえば、それぞれが独自のクライアントソフトウェアを有する複数のクライアントデバイスを単一のアカウントに関連付けることができ、そのアカウント内のコンテンツアイテムを複数のクライアントデバイスのそれぞれの間で同期させることができる。

10

**【0074】**

クライアントデバイス150は、ユーザに代わってコンテンツ管理システム110に接続することができる。たとえば、クライアントデバイス150がデスクトップまたはラップトップコンピュータ、電話、テレビ、モノのインターネットデバイスなどである場合、ユーザはクライアントデバイス150と直接インタラクションを行うことができる。代替的または追加的には、たとえばクライアントデバイス150がサーバである場合、ユーザがクライアントデバイス150に物理的にアクセスすることなく、クライアントデバイス150がユーザに代わって行動することができる。

20

**【0075】**

クライアントデバイス150の一部の機能は、クライアントデバイス150にインストールされたアプリケーションによって有効化される。一部の実施形態では、アプリケーションは、コンテンツ管理システム特有のコンポーネントを含むことができる。たとえば、コンテンツ管理システム特有のコンポーネントは、スタンドアロンアプリケーション152、1つまたは複数のアプリケーションプラグイン、及び/またはブラウザ拡張機能とすることができる。しかしながら、ユーザは、クライアントデバイス150に常駐し、コンテンツ管理システム110と通信するように構成される、ウェブブラウザなどのサードパーティアプリケーションを介してコンテンツ管理システム110とインタラクションを行うこともできる。様々な実施形態において、クライアント側アプリケーション152は、ユーザがコンテンツ管理システム110とインタラクションを行うためのユーザインターフェース(UI)を提示することができる。たとえば、ユーザは、ファイルシステムと統合されたファイルシステムエクスプローラを介して、またはウェブブラウザアプリケーションを使用して表示されるウェブページを介して、コンテンツ管理システム110とインタラクションを行うことができる。

30

40

**【0076】**

一部の実施形態では、クライアントアプリケーション152は、コンテンツ管理システム110の複数のアカウントのコンテンツを管理し、同期させるように構成することができる。そのような実施形態では、クライアントアプリケーション152は、複数のアカウントにログインされたままにし、複数のアカウントに通常のサービスを提供することができる。一部の実施形態では、各アカウントはファイルシステム内でフォルダとして表示することができる、そのフォルダ内の全てのコンテンツアイテムをコンテンツ管理システム110と同期させることができる。一部の実施形態では、クライアントアプリケーション152は、プライマリアカウントまたはデフォルトアカウントとなる複数のアカウントのうちの1つを選択するためのセレクタを含むことができる。

50

## 【 0 0 7 7 】

コンテンツ管理システム 1 1 0 は特定のコンポーネントと共に提示しているが、システム 1 0 0 のアーキテクチャ構成は 1 つの可能な構成にすぎず、より多いまたは少ないコンポーネントを有する他の構成が可能であることを当業者には理解されたい。さらに、サービスはより多いまたは少ない機能を有することができ、他のサービスと共に記述されている機能であっても含むことができる。また、一実施形態に関して本明細書に記載している特徴は、他の実施形態に関して記載している特徴と組み合わせることができる。

## 【 0 0 7 8 】

システム 1 0 0 は特定のコンポーネントと共に提示しているが、システム 1 0 0 のアーキテクチャ構成は 1 つの可能な構成にすぎず、より多いまたは少ないコンポーネントを有する他の構成が可能であることを当業者には理解されたい。

10

## 【 0 0 7 9 】

## クライアント同期サービス

図 2 に、一部の実施形態による、クライアント同期サービス 1 5 6 の一例を示す。一部の実施形態によれば、クライアント同期サービス 1 5 6 は、図 1 のクライアントデバイスに実装され得る。しかしながら、他の実施形態では、クライアント同期サービス 1 5 6 は、他のコンピューティングデバイス上に実装され得る。クライアント同期サービス 1 5 6 は、コンテンツ管理システムと、クライアント同期サービス 1 5 6 が動作するクライアントデバイスとの間でコンテンツアイテムへの変更を同期させるように構成される。

## 【 0 0 8 0 】

クライアント同期サービス 1 5 6 は、ファイルシステムインターフェース 2 0 5、サーバインターフェース 2 1 0、ツリーストレージ 2 2 0、プランナ 2 2 5、及びスケジューラ 2 3 0 を含み得る。追加または代替のコンポーネントも含まれ得る。以下、図 2 に関して、クライアント同期サービス 1 5 6 及びそのコンポーネントの高レベルの説明を行う。しかしながら、クライアント同期サービス 1 5 6 及びそのコンポーネントのさらなる詳細及び実施形態については、全体にわたって説明している。

20

## 【 0 0 8 1 】

ファイルシステムインターフェース 2 0 5 は、クライアントデバイスのローカルファイルシステム上のコンテンツアイテムへの変更を処理し、ローカルツリーを更新するように構成される。たとえば、ファイルシステムインターフェース 2 0 5 は、図 1 のクライアント同期サービス 1 5 6 と通信して、クライアントデバイスのローカルファイルシステム上のコンテンツアイテムへの変更を検出することができる。変更は、図 1 のクライアントアプリケーション 1 5 2 を介しても行われ、検出され得る。ファイルシステムインターフェース 2 0 5 は、ローカルツリーへの更新を行い得る。ローカルツリーの更新は、クライアントデバイス上のコンテンツアイテムへの変更（新規の、削除された、修正された、コピーされた、名前変更された、または移動されたコンテンツアイテム）に基づいて行われ得る。

30

## 【 0 0 8 2 】

サーバインターフェース 2 1 0 は、コンテンツ管理システムのリモートストレージにおけるコンテンツアイテムへのリモートの変更の処理と、リモートツリーの更新とを支援するように構成される。たとえば、サーバインターフェース 2 1 0 は、図 1 のサーバ同期サービス 1 1 2 と通信して、クライアントデバイス 1 5 0 とコンテンツ管理システム 1 1 0 との間でコンテンツアイテムへの変更を同期させることができる。コンテンツ管理システム 1 1 0 におけるコンテンツアイテムへの変更（新規の、削除された、修正された、コピーされた、名前変更された、または移動されたコンテンツアイテム）が検出され得、コンテンツ管理システム 1 1 0 における変更を反映するようにリモートツリーに更新が行われ得る。

40

## 【 0 0 8 3 】

ツリーストレージ 2 2 0 は、クライアント同期サービス 1 5 6 によって使用されるツリーデータ構造を記憶及び維持するように構成される。たとえば、ツリーストレージ 2 2 0

50

は、ローカルツリー、同期ツリー、及びリモートツリーを記憶し得る。一部の実施形態によれば、ツリーストレージ 220 は、永続メモリ（たとえば、ハードディスクまたは他の二次記憶デバイス）だけでなく、遅延及び応答時間を短縮するために、メインメモリ（たとえば、RAM または他の一次記憶デバイス）にもツリーデータ構造を記憶し得る。たとえば、クライアントデバイスまたはクライアント同期サービス 156 の起動時に、ツリーデータ構造が永続メモリから取り出され、メインメモリにロードされ得る。ツリーストレージ 220 はメインメモリ上のツリーデータ構造にアクセスし、これを更新し得、クライアントデバイスまたはクライアント同期サービス 156 がシャットダウンされる前に、ツリーストレージ 220 は更新されたツリーデータ構造を永続メモリに記憶し得る。メインメモリはコストが高く、ほとんどのクライアントデバイスでサイズが限られていることが多いので、メインメモリ上のツリーデータ構造のフットプリントを縮小するために、追加の技術的改善が実装される。これらの技術的解決方法については、以下でさらに説明する。

#### 【0084】

プランナ 225 は、ツリーデータ構造の状態に基づいて、コンテンツ管理システムに関連するサーバ状態と、クライアントデバイスに関連するファイルシステム状態との差異を検出するように構成される。たとえば、プランナ 225 は、リモートツリーと同期ツリーとの間に差異があるか否かを判定し得る。リモートツリーと同期ツリーとの差異は、コンテンツ管理システムに記憶された 1 つまたは複数のコンテンツアイテムに対してリモートで実行されたアクションによって、サーバ状態とファイルシステム状態とが同期しなくなったことを示す。同様に、プランナ 225 はまた、ローカルツリーと同期ツリーとの間に差異があるか否かを判定し得る。ローカルツリーと同期ツリーとの差異は、クライアントデバイスに記憶された 1 つまたは複数のコンテンツアイテムに対してローカルに実行されたアクションによって、サーバ状態とファイルシステム状態とが同期しなくなったことを示す。差異が検出された場合、プランナ 225 は、ツリーデータ構造を同期させる操作セットを生成する。

#### 【0085】

一部のシナリオでは、リモートツリーと同期ツリーとの差異に基づいて生成される操作セットと、ローカルツリーと同期ツリーとの差異に基づいて生成される操作セットとは、競合し得る。プランナ 225 は、2 つの操作セットを単一のマージされた操作プランにマージするようにも構成され得る。

#### 【0086】

スケジューラ 230 は、生成された操作プランを取り、それらの操作の実行を管理するように構成される。一部の実施形態によれば、スケジューラ 230 は、操作プラン内の各操作を、その操作を行うために実行される必要がある一連の 1 つまたは複数のタスクに変換する。一部のシナリオでは、一部のタスクが古くなったり、もはや関連しなくなったりし得る。スケジューラ 230 は、これらのタスクを特定し、キャンセルするように構成される。

#### 【0087】

##### ツリーデータ構造

図 3 に、様々な実施形態による、ツリーデータ構造の一例を示す。ツリーデータ構造は、クライアントデバイスに記憶され、図 2 のクライアント同期サービス 156 などのクライアント同期サービスによって管理され得る。図 3 には、リモートツリー 310、同期ツリー 330、及びローカルツリー 350 を含むツリーデータ構造を示している。

#### 【0088】

リモートツリー 310 は、サーバ状態、すなわち、クライアントデバイスからリモートに（たとえば、コンテンツ管理システムのサーバ上に）記憶されたコンテンツアイテムの状態を表す。ローカルツリー 350 は、ファイルシステム状態、すなわち、クライアントデバイスにローカルに記憶された対応するコンテンツアイテムの状態を表す。同期ツリー 330 は、ローカルツリー及びリモートツリーのマージベースを表す。マージベースは、ローカルツリー及びリモートツリーの共通の祖先、またはローカルツリー及びリモートツ

10

20

30

40

50

リーの間の最後の既知の同期した状態と考えられ得る。

#### 【0089】

各ツリーデータ構造（たとえば、リモートツリー310、同期ツリー330、またはローカルツリー350）は、1つまたは複数のノードを含み得る。各ノードは1つまたは複数の子ノードを有し得、親子関係はエッジで表される。たとえば、リモートツリー310はノード312及び314を含む。ノード312はノード314の親であり、ノード314はノード312の子である。この親子関係はエッジ316で表される。ルートノード312などのルートノードは、親ノードを有さない。ノード314などのリーフノードは、子ノードを有さない。

#### 【0090】

ツリーデータ構造の各ノードは、コンテンツアイテム（たとえば、ファイル、ドキュメント、フォルダなど）を表し得る。たとえば、ルートノード312は、コンテンツ管理システムに関連付けられたルートフォルダを表し得、ノード314は、そのルートフォルダに配置されたファイル（たとえば、「foo.txt」という名前のテキストファイル）を表し得る。ツリーデータ構造の各ノードは、たとえば、コンテンツアイテムの親ノードのファイル識別子を示すディレクトリファイル識別子（「DirFileID」）、コンテンツアイテムのファイル名、コンテンツアイテムのファイル識別子、及びコンテンツアイテムのメタデータなどのデータを含み得る。

#### 【0091】

上述のように、クライアント同期サービスは、3つのツリー（たとえば、リモートツリー310、同期ツリー330、及びローカルツリー350）が全て同一である場合に、サーバ状態とクライアントデバイスのファイルシステム状態とが同期していると判定し得る。換言すれば、それらのツリー構造と、それらが表す関係とが同一であって、それらのノードに含まれるデータも同一である場合に、ツリーは同期している。逆に、3つのツリーが同一でない場合、ツリーは同期していない。図3に示す例示的なシナリオでは、リモートツリー310、同期ツリー330、及びローカルツリー350が同一であって同期しているように図示しており、その結果、サーバ状態とファイルシステム状態とが同期している。

#### 【0092】

ツリーデータ構造を使用した変更の追跡

図4に、様々な実施形態による、ツリーデータ構造の一例を示す。図3に示すツリーデータ構造と同様に、図4に示すツリーデータ構造（リモートツリー410、同期ツリー430、及びローカルツリー450を含む）は、クライアントデバイスに記憶され、図2のクライアント同期サービス156などのクライアント同期サービスによって管理され得る。図4には、ツリーデータ構造を示している。

#### 【0093】

図4には、図3に示すシナリオなどの、以前に同期していた状態の後のシナリオを示しており、コンテンツアイテムを修正するための追加のアクションが、ツリーで表されたコンテンツアイテムに対して実行され、ツリーがもはや同期していない。同期ツリー430は、以前の既知の同期した状態の表現を保持し、クライアント同期サービスによって使用されて、サーバ状態とファイルシステム状態との差異が特定されると共に、サーバ状態とファイルシステム状態とが同期するように収束させるためにコンテンツ管理システム及び/またはクライアントデバイスが実行する操作が生成され得る。

#### 【0094】

たとえば、ユーザ（クライアントデバイスに関連付けられたユーザと同じユーザ、またはコンテンツアイテムにアクセス可能な別のユーザ）は、コンテンツ管理システムによって記憶された「foo.txt」コンテンツアイテムへの修正を行い得る。このコンテンツアイテムは、リモートツリー410のノード414によって表される。リモートツリー410に示す修正は、foo.txtコンテンツアイテムの除去（たとえば、コンテンツ管理システムによって管理されるスペースからのコンテンツアイテムの除去）または削除

10

20

30

40

50

である。これらの修正は、たとえば、他のクライアントデバイスで実行され、その後コンテンツ管理システムへ同期され、またはコンテンツ管理システムに接続されたウェブブラウザを介して実行され得る。

**【 0 0 9 5 】**

コンテンツ管理システム上で変更が行われた場合、コンテンツ管理システムは、行われた変更を示す修正データを生成し、修正データをクライアントデバイス上のクライアント同期サービスに送信する。たとえば、コンテンツ管理システムがクライアントデバイスに一方的に変更を送信、すなわち「プッシュ」し得るプッシュモデルを使用する。他の実施態様では、サーバがクライアントデバイスによるリクエストに回答して変更を送信するプルモデルである。加えて、クライアントデバイスがリクエストを開始するが、接続を一定期間開放したままにして、コンテンツ管理システムが、接続がライブである間に追加の変更を必要に応じてプッシュできるようにする、ロングプルを含むハイブリッドモデルである。クライアント同期サービスは、コンテンツ管理システムによって記憶されたコンテンツアイテムのサーバ状態を表すリモートツリーを修正データに基づいて更新する。たとえば、リモートツリー 4 1 0 では、foo . text コンテンツアイテムを表すノード 4 1 4 が削除されるように図示している。

10

**【 0 0 9 6 】**

クライアント同期サービスは、リモートツリー 4 1 0 と同期ツリー 4 3 0 との差異を特定し、その結果、コンテンツ管理システムにおけるコンテンツアイテムの修正によって、サーバ状態とファイルシステム状態とが同期しなくなったと判定し得る。クライアント同期サービスはさらに、サーバ状態とファイルシステム状態とが同期するようにそれらを収束させるよう構成される、クライアントデバイスに記憶されたコンテンツアイテムに対する操作のセットまたはシーケンスを生成及び実行し得る。

20

**【 0 0 9 7 】**

追加的または代替的には、ユーザ（コンテンツ管理システムにおける修正に関連するユーザと同じユーザ、またはコンテンツアイテムにアクセス可能な別のユーザ）は、コンテンツ管理システムに関連付けられた、クライアントデバイスにローカルに記憶されたコンテンツアイテムに修正を行い得る。たとえば、ユーザは「/ bar」フォルダを「/ root」フォルダに追加し、「Hi . doc」ドキュメントを「/ bar」フォルダに追加し得る。

30

**【 0 0 9 8 】**

クライアントデバイスで変更が行われた場合、クライアントデバイス（たとえば、図 1 のクライアント同期サービス 1 5 6 またはクライアントアプリケーション 1 5 2）は、行われた変更を示す修正データを生成する。クライアント同期サービスは、修正データに基づいて、クライアントデバイスに記憶されたコンテンツアイテムのファイルシステム状態を表すローカルツリーを更新する。たとえば、ローカルツリー 4 5 0 において、ノード 4 5 2 及びノード 4 5 4 が追加されるように図示している。ノード 4 5 2 及びノード 4 5 4 は、それぞれ「/ bar」フォルダ及び「Hi . doc」ドキュメントを表す。

**【 0 0 9 9 】**

クライアント同期サービスは、ローカルツリー 4 5 0 と同期ツリー 4 3 0 との差異を特定し、その結果、クライアントデバイスにおけるコンテンツアイテムの修正によって、サーバ状態とファイルシステム状態とが同期しなくなったと判定し得る。クライアント同期サービスはさらに、サーバ状態とファイルシステム状態とが同期するようにそれらを収束させるよう構成される、コンテンツ管理システムによって記憶されたコンテンツアイテムに対する操作のセットまたはシーケンスを生成し得る。これらの操作は、コンテンツ管理システムに送信されて実行され得る。

40

**【 0 1 0 0 】**

図 4 に見られるように、クライアントデバイスに記憶されたコンテンツアイテム及びコンテンツ管理システムによって記憶されたコンテンツアイテムへの修正は、実質的に同時にまたは特定の期間内に行われ得る。これらの修正はツリーデータ構造に反映され、クラ

50

クライアント同期サービスが使用して、クライアントデバイス及びコンテンツ管理システムに対する操作を並行して生成することができる。しかしながら、他のシナリオでは、修正は必ずしも同じ期間内に行われなくてもよく、操作は必要に応じて生成され得る。さらに、図4にはコンテンツアイテムの追加及びコンテンツアイテムの削除のシナリオを示しているが、コンテンツアイテムの編集、名前変更、コピー、または移動などの他のタイプの修正もサポートされる。

#### 【0101】

様々な実施形態によれば、2つのツリーデータ構造間の差異を特定し、操作を生成することは、両方のツリーデータ構造内の各ノードをチェックし、そのノードでアクションが実行されたか否かを判定することを含み得る。アクションには、たとえば、ノードの追加、ノードの削除、ノードの編集、またはノードの移動が含まれ得る。そして、これらのアクションを使用して、サーバ状態とファイルシステム状態とを収束させるように構成される操作を生成し得る。

10

#### 【0102】

たとえば、2つのツリーデータ構造が同期ツリー及びリモートツリーである場合、クライアント同期サービスは、たとえば同期ツリー内の全てのノードのファイル識別子を要求することにより、同期ツリー内の各ノードを識別し得る。同期ツリー内のノードのノードまたはファイル識別子ごとに、クライアント同期サービスは、ノードまたはファイル識別子がリモートツリー内にも存在するか否かを判定し得る。リモートツリーで見つからない同期ツリー内のノードまたはファイル識別子は、リモートツリーで表されるサーバ状態からノードが削除されたことを示し得る。したがって、クライアント同期サービスは、リモートツリー上で削除アクションが行われたと判定し得る。ノードのノードまたはファイル識別子がリモートツリーで見つかった場合、クライアント同期サービスは、リモートツリーのノードが編集または移動されたか否かをチェックし得る。

20

#### 【0103】

リモートツリーのノードが同期ツリーのノードに対して編集されているか否かを判定するために、クライアント同期サービスは、同期ツリーのノードのメタデータを、リモートツリーの対応するノード(たとえば、同じファイル識別子を有するノード)のメタデータと比較し得る。メタデータは、ノードによって表されるコンテンツアイテムが編集されたか否かを判定するために使用され得る情報を含み得る。たとえば、メタデータは、コンテンツアイテムまたはその一部のデータに基づいて生成される1つまたは複数のハッシュ値を含み得る。メタデータには、追加的または代替的には、コンテンツアイテムのサイズ値、最終修正値、または他の値が含まれ得る。同期ツリーのノードのメタデータは、リモートツリーのノードのメタデータと比較され得る。メタデータが一致しない場合、リモートツリーで表されるサーバ状態においてコンテンツアイテムの編集が行われている場合がある。したがって、クライアント同期サービスは、リモートツリー上のノードに対して編集アクションが行われたと判定し得る。メタデータが一致する場合、編集が行われていない場合がある。

30

#### 【0104】

リモートツリー内のノードが移動されたか否かを判定するために、クライアント同期サービスは、同期ツリー内のノードの場所をリモートツリー内の対応するノード(たとえば、同じファイル識別子を有するノード)の場所と比較し得る。場所には、たとえば、ノードが配置されているパス、ファイル名、及び/またはノードの親のファイル識別子を示すディレクトリファイル識別子(「DirFileID」)が含まれ得る。場所が一致する場合、移動が行われていない場合がある。一方、場所が一致しない場合、リモートツリーで表されるサーバ状態においてコンテンツアイテムの移動が行われている場合がある。したがって、クライアント同期サービスは、リモートツリー上のノードに対して移動アクションが行われたと判定し得る。

40

#### 【0105】

ノードがリモートツリーに追加されたか否かを判定するために、クライアント同期サー

50

ピスは、同期ツリーで見つからないリモートツリー内の任意のノードまたはファイル識別子を識別し得る。ノードまたはファイル識別子がリモートツリーで見つかり、同期ツリーで見つからない場合、クライアント同期サービスは、サーバ状態を表すリモートツリーでこのノードの追加アクションが行われたと判定し得る。

**【0106】**

上記の例は同期ツリー及びリモートツリーに関して説明しているが、他の実施形態では、同期ツリーとローカルツリーとの差異を特定し、ファイルシステム状態を表すローカルツリーで発生したアクションを特定するために、同期ツリーとローカルツリーとを用いて同様の処理が行われ得る。

**【0107】**

ツリーデータ構造を使用した同期

図5に、本主題技術の様々な実施形態による、ツリーデータ構造を使用してサーバ状態とファイルシステム状態とを同期させるための例示的な方法を示す。本明細書に記載の方法及び処理は、特定の順序の特定のステップ及び動作と共に示す場合があるが、同様または代替の順序でまたは並行して実行される追加の、より少ない、または代替のステップ及び動作は、特に明記しない限り、様々な実施形態の範囲内にある。方法500は、たとえば、クライアントデバイス上で動作する図2のクライアント同期サービス156などのシステムによって実装され得る。

**【0108】**

システムは、コンテンツ管理システムによって記憶されたコンテンツアイテムのサーバ状態を表すリモートツリーと、クライアントデバイスに記憶された対応するコンテンツアイテムのファイルシステム状態を表すローカルツリーと、サーバ状態及びファイルシステム状態の間の既知の同期した状態を表す同期ツリーとの間の差異を特定するように構成される。これらの差異に基づいて操作セットが生成され得、操作セットは、実行された場合に、3つのツリーデータ構造が同一になる同期した状態へ向けてサーバ状態及びファイルシステム状態を収束させるように構成される。

**【0109】**

たとえば、動作505において、システムは、コンテンツ管理システムによってまたはクライアントデバイス上に記憶されたコンテンツアイテムの修正データを受信し得る。動作510において、修正データを使用してリモートツリーまたはローカルツリーを更新し得る。

**【0110】**

修正データは、コンテンツ管理サービスに関連付けられた1つまたは複数のコンテンツアイテムにどのような変更が行われたかを示す。したがって、修正データは、コンテンツ管理システムまたはクライアントデバイスから（たとえば、図1のクライアントデバイス150上で動作するクライアントアプリケーション152から）受信され得る。コンテンツ管理システムから受信される修正データは、サーバ修正データと呼ばれ得る。サーバ修正データは、コンテンツ管理システムによって1つまたは複数のコンテンツアイテムにどのような変更が行われたかを示し、動作510においてリモートツリーを更新するために使用され得る。クライアントデバイスから受信される修正データは、クライアント修正データと呼ばれ得る。クライアント修正データは、クライアントデバイス上で1つまたは複数のコンテンツアイテムにどのような変更が行われたかを示し、動作510においてローカルツリーを更新するために使用され得る。

**【0111】**

動作515において、システムは、コンテンツ管理システムによって記憶されたコンテンツアイテムのサーバ状態と、クライアントデバイスに記憶されたコンテンツアイテムのファイルシステム状態とが同期しているか否かを判定し得る。ローカルツリー及びリモートツリーはファイルシステム状態及びサーバ状態を表し、コンテンツ管理システム及びクライアントデバイスで行われた変更を追跡するために継続的に更新されるので、サーバ状態とファイルシステム状態とが同期しているか否かを判定することは、ローカルツリー及

10

20

30

40

50

び/またはリモートツリーを同期ツリーと比較して、ツリー間の差異を見つけることによって行われ得る。ツリー間の差異を見つけるこの処理は、ツリーの「差分をとる」と呼ばれることもある。

【0112】

一部の実施形態及びシナリオによれば、サーバ状態とファイルシステム状態とが同期しているか否かを判定することは、リモートツリーと同期ツリーとの差異を特定すること、及び/またはローカルツリーと同期ツリーとの差異を特定することのうちの1つまたは複数を含み得る。リモートツリーと同期ツリーとの差異は、クライアントデバイスに反映されていない場合がある、コンテンツ管理システムによって記憶されたコンテンツアイテムへの変更の発生を示し得る。同様に、ローカルツリーと同期ツリーとの差異は、コンテンツ管理システムに反映されていない場合がある、クライアントデバイスに記憶されたコンテンツアイテムへの変更の発生を示し得る。

10

【0113】

ツリー間に差異がない場合、サーバ状態及びファイルシステム状態は同期しており、同期アクションは必要とされない。したがって、この方法は動作505に戻り、新たな修正データを待機し得る。一方、差異が検出された場合、システムは、動作520においてサーバ状態とファイルシステム状態とを収束させるように構成される操作セットを生成し得る。

【0114】

生成される操作セットは、検出された1つまたは複数の差異に依存する。たとえば、2つのツリー間の差異が追加されたコンテンツアイテムである場合、生成される操作セットは、追加されたコンテンツアイテムの取り出し及び追加を含み得る。2つのツリー間の差異がコンテンツアイテムの削除である場合、生成される操作セットは、コンテンツアイテムの削除を含み得る。一部の実施形態によれば、操作セットは、ツリー制約が維持されるようにするためのいくつかのチェックも含み得る。以下でさらに説明するように、操作セットは、サーバ状態、ファイルシステム状態の現在の状態、または実行を保留している他の操作と競合し得る。したがって、システムはまた、先に進む前にこれらの競合を解決し得る。

20

【0115】

上記のように、リモートツリーと同期ツリーとの間に差異がある場合、クライアントデバイスに反映されていない場合がある、コンテンツ管理システムによって記憶されたコンテンツアイテムへの変更が行われている場合がある。したがって、このシナリオでは、システムは、サーバ状態とファイルシステム状態とを収束させるようにクライアントデバイスに記憶されたコンテンツアイテムを操作するよう構成されるクライアント操作セットを生成し得、このクライアント操作セットは、動作525においてクライアントデバイスに提供されて実行され得る。

30

【0116】

同様に、ローカルツリーと同期ツリーとの間に差異がある場合、コンテンツ管理システムに反映されていない場合がある、クライアントデバイスに記憶されたコンテンツアイテムへの変更が行われている場合がある。したがって、このシナリオでは、システムは、サーバ状態とファイルシステム状態とを収束させるようにコンテンツ管理システムによって記憶されたコンテンツアイテムを操作するよう構成されるサーバ操作セットを生成し得、動作525において、このサーバ操作セットはコンテンツ管理システムに提供されて実行され得る。場合によっては、両方のケースが当てはまることもあり、クライアント操作セット及びサーバ操作セットが生成され、動作525においてそれらの目的の受信者に提供され得る。

40

【0117】

操作セット(複数可)が目的の受信者(複数可)に提供されると、この方法は動作505に戻り、新たな修正データを待機し得る。操作セット(複数可)は、サーバ状態及びファイルシステム状態の収束を目的とする1つまたは複数のステップを提供し、またはサー

50

バ状態とファイルシステム状態とを同期させるために必要な全てのステップを提供し得る。たとえば、コンテンツ管理システムは、サーバ操作セットを受信し、コンテンツ管理システムによって記憶されたコンテンツアイテムにサーバ操作セットを実行し得る。このサーバ操作セットの実行により、コンテンツ管理システムによって記憶されたコンテンツアイテムへの変更が生成され、変更はサーバ修正データ内で検出及び指定され、サーバ修正データはシステムに返送される。システムは次いでリモートツリーを更新し、サーバ状態とファイルシステム状態とが同期しているか否かを判定し得る。

**【0118】**

クライアントデバイスは、クライアント操作セットを受信し、クライアントデバイスに記憶されたコンテンツアイテムにクライアント操作セットを実行し得る。このクライアント操作セットの実行により、クライアントデバイスに記憶されたコンテンツアイテムへの変更が生成され、変更はクライアント修正データ内で検出及び指定され、クライアント修正データはシステムに渡される。システムは次いでローカルツリーを更新し、サーバ状態とファイルシステム状態とが同期しているか否かを判定し得る。方法500のこれらの動作は、サーバ状態とファイルシステム状態とが同期するまで継続し得る。

10

**【0119】**

方法500の動作は、クライアント側及びサーバ側（たとえば、ローカルツリー及びリモートツリー、ファイルシステム状態及びサーバ状態、クライアント操作セット及びサーバ操作セット、クライアント修正データ及びサーバ修正データ）に関して説明している。様々な実施形態において、2つの側に関連する動作は、並行して、順番に、他方と分離して、または組み合わせて行われ得る。

20

**【0120】**

さらに詳細に説明するように、一部の実施形態によれば、操作が提供されて実行される前に、システムは操作をチェックして、それらがルールまたは不変条件のセットに従うか否かを判定し得る。操作がルールに違反する場合、システムはルールの違反に関連付けられた解決処理を実行する。

**【0121】**

加えて、一部の実施形態によれば、システム（たとえば、図2のクライアント同期サービス156のスケジューラ230）は、操作セットの実行を管理し得る。たとえば、操作セットの各操作は、タスク、実行スレッド、一連のステップ、または命令に関連付けられ得る。システムは、タスク、スレッド、ステップ、または命令を実行し、クライアントデバイス及び/またはコンテンツ管理システムとインターフェースして、操作セットを実行し、サーバ状態とファイルシステム状態とを収束させるように構成され得る。

30

**【0122】****競合処理**

図5に関して上述したように、同期ツリーとリモートツリーとの差異が特定され、使用されて、サーバ状態とファイルシステム状態とを収束させるように構成されるクライアント操作セットが生成される。しかしながら、一部のケースでは、クライアント操作セットは、ローカルツリーの現在の状態と競合し得る。同様に、同期ツリーとローカルツリーとの差異が特定され、使用されて、サーバ状態とファイルシステム状態とを収束させるように構成されるサーバ操作セットが生成される。しかしながら、サーバ操作セットは、リモートツリーの現在の状態と競合し得る。追加的または代替的には、クライアント操作セット及びサーバ操作セットは互いに競合し得、またはシステムによって維持される他のルールもしくは不変条件に違反し得る。したがって、本主題技術の様々な実施形態は、これらの競合を解決することにより追加の技術的改善を提供する。

40

**【0123】**

たとえば、図2のクライアント同期サービス156のプランナ225は、ルールと競合する操作セット（たとえば、クライアント操作セットまたはサーバ操作セット）内の操作を特定し得る。また、競合を特定するために使用される各ルールは、競合の解決策に関連付けられ得る。クライアント同期サービスは、競合の解決策に基づいて操作セットを更新

50

し得、または操作セットを提供して実行させる前に競合の解決策に関連付けられた操作を実行することにより、競合を解決し得る。

【 0 1 2 4 】

図 6 に、本主題技術の様々な実施形態による、ツリーデータ構造を使用してサーバ状態とファイルシステム状態とを同期させる場合の競合を解決するための例示的な方法 6 0 0 を示す。本明細書に記載の方法及び処理は、特定の順序の特定のステップ及び動作と共に示す場合があるが、同様または代替の順序でまたは並行して実行される追加の、より少ない、または代替のステップ及び動作は、特に明記しない限り、様々な実施形態の範囲内にある。方法 6 0 0 は、たとえば、クライアントデバイス上で動作する図 2 のクライアント同期サービス 1 5 6 などのシステムによって実装され得る。

10

【 0 1 2 5 】

システムは、動作 6 2 0 においてサーバ状態とファイルシステム状態とを収束させるように構成される操作セットを受信し得る。操作セットは、たとえば、クライアント操作セット、サーバ操作セット、または図 5 の方法 5 0 0 に関して生成及び説明される組み合わせられた操作セットであり得る。

【 0 1 2 6 】

動作 6 5 0 において、システムは、ルールのセットに基づいて、操作セットにおける 1 つまたは複数の違反を特定する。ルールのセットは、図 2 のクライアント同期サービス 1 5 6 によって記憶され、いくつかの制約、不変条件、または解決すべき操作の競合を規定し得る。ルールのセットはツリーデータ構造に適用され、同期動作の制御に役立ち得る。また、ルールのセット内の各ルールは、そのルールの違反の解決策に関連付けられるか、あるいは別の方法でリンクされ得る。たとえば、解決策には、操作セット内の 1 つまたは複数の操作の変更、1 つまたは複数の操作の除去、1 つまたは複数の操作の追加、サーバ状態またはファイルシステム状態への 1 つまたは複数の追加のアクション、またはアクションの組み合わせが含まれ得る。

20

【 0 1 2 7 】

操作セット内の操作ごとに、システムは、ルールのセット内の任意のルールに違反しているか否かを判定し得る。ルールに違反している場合、システムは、違反の解決策を特定し、動作 6 5 5 において解決策を実行する。解決策には、操作セット内の 1 つまたは複数の操作の修正、1 つまたは複数の操作の除去または追加、あるいはサーバ状態またはファイル状態に対する追加のアクションなどのアクションが含まれ得る。

30

【 0 1 2 8 】

解決アクションが実行されると、システムは、動作 6 6 0 において、解決策及び操作セットに基づいて解決されたまたはリベースされた操作セットを生成し、動作 6 6 5 において、解決された操作セットを適切なエンティティに提供して実行させ得る。たとえば、解決された操作セットは、管理された実行のために、図 2 のクライアント同期サービス 1 5 6 のスケジューラ 2 3 0 に提供され得る。代替的には、操作セットがクライアント操作セットである場合、解決された操作セットはクライアントデバイスに提供され得る。操作セットがサーバ操作セットである場合、解決された操作セットはコンテンツ管理サービスに提供され得る。加えて、図 6 の方法 6 0 0 は、クライアント操作セット及びサーバ操作セットに関して、順番に、並行して、または様々な異なる順序で実行され得る。

40

【 0 1 2 9 】

一部の実施形態によれば、各タイプの操作は、同じまたは異なるルールのセットに関連付けられ得る。たとえば、操作タイプには、たとえば、コンテンツアイテムの追加、コンテンツアイテムの削除、コンテンツアイテムの編集、コンテンツアイテムの移動、コンテンツアイテムの名前変更などが含まれ得る。操作セットは、それぞれが上記の操作タイプの 1 つに属する操作で構成され得る。各操作タイプは、特定のルールのセットに関連付けられ得る。

【 0 1 3 0 】

説明の目的で、「追加」操作タイプのルールのセットには、たとえば、コンテンツアイ

50

テムのファイル識別子はツリー内で一意でなければならない(たとえば、ツリー内の2つのノードは同じファイル識別子を有することはできない)というルールと、コンテンツアイテムの親ノードのファイル識別子を示すディレクトリファイル識別子(「DirFileID」)が反対側のツリーデータ構造に存在しなければならないというルールと、コンテンツアイテムのDirFileIDとファイル名との組み合わせは反対側のツリーで使用されないというルールと、が含まれ得る。

#### 【0131】

反対側のツリーとは、本明細書で使用する場合、反対側のエンティティの状態を表すツリーデータ構造を指す。たとえば、クライアントデバイス进行操作するように構成されるクライアント操作セットと、その結果のクライアントデバイス上のファイルシステムへの変更は、ローカルツリーに反映される。したがって、クライアント操作セットの反対側のツリーはリモートツリーである。同様に、サーバ操作セットは、コンテンツ管理システムに送信されて実行されるように構成され、その結果のサーバ状態への変更は、リモートツリーに反映される。したがって、サーバ操作セットの反対側のツリーはローカルツリーである。

10

#### 【0132】

図7に、様々な実施形態による、追加操作のルール違反を示すツリーデータ構造の一例を示す。ツリーデータ構造は、リモートツリー710、同期ツリー750、及びローカルツリー770を含む。ローカルツリー770を参照する場合、リモートツリー710は反対側のツリーと見なされ得る。一方、リモートツリー710を参照する場合、ローカルツリー770は反対側のツリーと見なされ得る。図7に、リモートツリー710においてノード712により表されるコンテンツアイテムを追加する操作セットを示す。たとえば、クライアント同期サービスは、リモートツリー710を同期ツリー750と比較し、差異を特定し、ノード712の追加を含む操作セットを生成し得る。ノード712は、FileID4と、DirFileID3(これはノード712の親である親ノード714を参照する)と、ファイル名「Hi」とに関連付けられている。親ノード714は、FileID3と、DirFileID1(これはノード714の親であるルートノード716を参照する)と、ファイル名「Foo」とに関連付けられている。

20

#### 【0133】

クライアント同期サービスは、図6の方法600を実行し、ノード712の追加操作が、「追加」操作タイプに関する「コンテンツアイテムのディレクトリファイル識別子(「DirFileID」)が反対側のツリーデータ構造に存在しなければならない」というルールに違反していると判定し得る。これは図7において、ローカルツリー770が、ノード712の親ノード714を参照するFileID3のノードを有さないことによって示されている。これが起こり得るのは、たとえば、リモートツリー710と同期ツリー750との差異が特定され、操作セットが生成された後に、ノード714に対応する「Foo」ノードが反対側のツリーから除去された場合である。

30

#### 【0134】

このルールに関連付けられた解決策は、ローカルツリー770から欠落しているノードを同期ツリー750から削除して、同期ツリー750とローカルツリー770とを同期させることと、リモートツリー710と同期ツリー750との差分を再度とること(たとえば、差分を見つけること)を含み得る。図7に示すシナリオでは、同期ツリー750のノード754が除去758され、差分取得操作が開始されて、リモートツリー710と同期ツリー750との差異が特定される。これにより、ノード714の追加操作に加え、ノード712の追加操作が操作セットに含まれることになる。

40

#### 【0135】

同様に、「追加」操作タイプに関する「コンテンツアイテムのファイル識別子はツリー内で一意でなければならない」というルールの違反は、追加されるノードの新たなFileIDをコンテンツ管理システムに要求し、ノードを追加するとき新たなFileIDを使用することを含む操作によって解決され得る。「追加」操作タイプに関する「コンテ

50

コンテンツアイテムのDirFileIDとファイル名との組み合わせは反対側のツリーで使用されない」というルール違反は、2つのノードに関連付けられたメタデータを介してコンテンツアイテムが同じであるか否かをチェックすることを含む操作によって解決され得る。コンテンツアイテムが同じである場合、追加されるコンテンツアイテムが他のアクションで既に追加されている可能性がある。コンテンツアイテムが同じでない場合、追加されるコンテンツアイテムのファイル名を変更することができる。たとえば、追加されるコンテンツアイテムのファイル名に「(競合するバージョン)」というテキストを付加することができる。

#### 【0136】

##### 段階的なプランナ

図3、図4、及び図7に示す様々なツリーデータ構造は、比較的少数のノードを含み、構造が比較的簡単であるが、システムによってサポートされるツリーデータ構造は、複数のレベルと、各レベルの多数になり得るノードとによって、はるかに大きく複雑であり得る。したがって、動作中にツリーデータ構造を記憶するために必要なメモリ使用量は非常に大きくなり得、ツリーデータ構造を操作するのに必要な計算時間及びリソースは非常に大きくなり得る。たとえば、リモートツリーと同期ツリー及び/またはローカルツリーと同期ツリーの差異を見つけ、リモートツリーと同期ツリー及び/またはローカルツリーと同期ツリーを収束させるために必要な操作を生成するには、大量のメモリ、時間、及び他のコンピューティングリソースが必要になり得る。

#### 【0137】

残念なことに、これらのコンピューティングリソースは限られている。たとえば、クライアントデバイスは、利用可能なメモリの量が限られている場合があり、ツリーの差分を抽出して操作を生成するのに必要な時間の長さは、クライアントデバイス、クライアントアプリケーション、またはコンテンツ管理システムによって提供されるコンテンツ管理サービスの使いやすさの妨げになり得る。さらに、サーバ状態とファイルシステム状態とを収束させるのに必要な時間が長いほど、いずれかの状態に介入する変更によって、計算または実行されている操作セット及び/または対象の同期した状態が最新でない可能性が高くなる。したがって、本主題技術の様々な実施形態では、サーバ状態及びファイルシステム状態を、それらを表すツリーデータ構造と共に段階的に収束させることにより追加の技術的改善を提供する。

#### 【0138】

図8に、本主題技術の様々な実施形態による、サーバ状態とファイルシステム状態とを段階的に収束させるための例示的な方法800を示す。本明細書に記載の方法及び処理は、特定の順序の特定のステップ及び動作と共に示す場合があるが、同様または代替の順序でまたは並行して実行される追加の、より少ない、または代替のステップ及び動作は、特に明記しない限り、様々な実施形態の範囲内にある。方法800は、たとえば、クライアントデバイス上で動作する図2のクライアント同期サービス156などのシステムによって実装され得る。

#### 【0139】

動作805において、システムは、リモートツリーまたはローカルツリーのいずれかを更新するために使用され得る修正データを受信し得る。たとえば、コンテンツ管理システムによって記憶された1つまたは複数のコンテンツアイテムに関連付けられた修正または他のアクション(たとえば、編集、追加、削除、移動、または名前変更)を特定するサーバ修正データが、コンテンツ管理システムから受信され得る。サーバ修正データは、コンテンツ管理システムによって記憶されたコンテンツアイテムのサーバ状態を表すリモートツリーを更新するために使用され得る。同様に、クライアント修正データは、クライアントデバイス(たとえば、クライアントアプリケーション)から受信され、クライアントデバイスに記憶された1つまたは複数のコンテンツアイテムに関連付けられた修正または他のアクションを示し得る。クライアント修正データは、クライアントデバイスに記憶されたコンテンツアイテムのファイルシステム状態を表すローカルツリーを更新するために使

10

20

30

40

50

用され得る。

【0140】

コンテンツアイテムに関連付けられた修正を示す受信した修正データに基づいて、システムは、動作810において、修正されたコンテンツアイテムに対応するノードを識別し、ノードを修正済みコンテンツアイテムのリストに追加し得る（たとえば、ノードに関連付けられたファイル識別子を修正されたコンテンツアイテムのリストに追加する）。動作805及び810は、システムが方法800の次の段階に進む前に、しばらくの間継続的に行われ得る。たとえば、追加の修正データを受信し、使用して、システムによって管理されるツリーを更新し、ノードを修正済みコンテンツアイテムのリストに追加し得る。

【0141】

サーバ状態とファイルシステム状態とを段階的に収束させるために、システムは動作815において、修正済みコンテンツアイテムのリスト内の各ノードを取得し、そのノードがどのように修正されたか（たとえば、どのアクションがノードに関連付けられているか）を判定する。一部の実施形態では、修正データは、ノードへの修正を示し得る。しかしながら、他の実施形態では、システムは、リモートツリーと同期ツリーとの比較及び/またはローカルツリーと同期ツリーとの比較に基づいて、ノードへの修正を決定し得る。たとえば、修正には、ノードの追加、ノードの削除、ノードの編集、またはノードの移動が含まれ得る。

【0142】

修正済みコンテンツアイテムのリスト内のノードのノードまたはファイル識別子ごとに、システムは一連のチェックを実行して、修正がノードで実行された場合に、どのような修正であったかを特定し得る。たとえば、システムは、ファイル識別子が同期ツリー内にあるがリモートツリー内にないか否かを判定し得る。リモートツリーで見つからない同期ツリー内のファイル識別子は、リモートツリーで表されるサーバ状態からノードが削除されたことを示し得る。したがって、クライアント同期サービスは、リモートツリーでノードの削除修正が行われたと判定し得る。同様に、システムは、ファイル識別子が同期ツリー内にあるがローカルツリー内にないか否かも判定する。ローカルツリーで見つからない同期ツリー内のファイル識別子は、ローカルツリーで表されるファイルシステム状態からノードが削除されたことを示し得る。したがって、クライアント同期サービスは、ローカルツリーでノードの削除修正が行われたと判定し得る。

【0143】

ノードで編集修正が実行されたか否かを判定するために、システムは、同期ツリーのノードのメタデータを、リモートツリー及び/またはローカルツリーの対応するノード（たとえば、同じファイル識別子を有するノード）のメタデータと比較し得る。メタデータは、ノードによって表されるコンテンツアイテムが編集されているか否かを判定するために使用され得る情報を含み得る。たとえば、メタデータは、コンテンツアイテムまたはその一部のデータに基づいて生成される1つまたは複数のハッシュ値を含み得る。メタデータには、追加的または代替的には、コンテンツアイテムのサイズ値、最終修正値、または他の値が含まれ得る。メタデータが一致しない場合、リモートツリーで表されるサーバ状態及び/またはローカルツリーで表されるファイルシステム状態でコンテンツアイテムの編集が行われている場合がある。したがって、システムは、リモートツリー及び/またはローカルツリー上のノードに対して編集アクションが行われたと判定し得る。

【0144】

リモートツリー内のノードが移動されたか否かを判定するために、システムは同期ツリー内のノードの場所をリモートツリー及び/またはローカルツリー内の対応するノード（たとえば、同じファイル識別子を有するノード）の場所と比較し得る。場所には、たとえば、ノードが配置されているパス、ファイル名、及び/またはノードの親のファイル識別子を示すディレクトリファイル識別子（「DirFileID」）が含まれ得る。場所が一致する場合、移動が行われていない場合がある。一方、場所が一致しない場合、リモートツリーまたはローカルツリーにおいてコンテンツアイテムの移動が行われている場合が

10

20

30

40

50

ある。したがって、クライアント同期サービスは、リモートツリー及び/またはローカルツリー上のノードに対して移動アクションが行われたと判定し得る。

【0145】

ノードがリモートツリーに追加されたか否かを判定するために、システムは、修正済みコンテンツアイテムのリスト内のファイル識別子がリモートツリーまたはローカルツリー内にあるが、同期ツリー内にはないか否かを判定し得る。ファイル識別子がリモートツリーまたはローカルツリーで見つかり、同期ツリーで見つからない場合、システムはこのノードの追加修正が行われたと判定し得る。

【0146】

修正済みコンテンツアイテムのリスト内のノードへの1つまたは複数の修正が決定されると、システムは動作820において、それらの修正のいずれかに依存関係があるか否かを判定し得る。図9に関してさらに説明するように、たとえば、他の修正が先に行われな

10

い限り修正を実行できない場合に、ノードの修正には依存関係がある。

【0147】

修正に依存関係がない場合、システムは動作825においてブロックされないアクションのリストに修正を追加する。修正に依存関係がある場合、修正は動作830においてしばらくの間ブロックされ、他の修正が先に処理されない限り実行することができない。したがって、処理は動作805に戻って、さらなる修正を待機する。各修正が処理された後、システムは修正済みコンテンツアイテムのリストから修正に関連付けられたファイル識別子をクリアし得る。

20

【0148】

図9に、様々な実施形態による、ツリーデータ構造の一例を示す。図9に示すツリーデータ構造は、クライアントデバイスに記憶され、図2のクライアント同期サービス156などのシステムによって管理され得る。説明の目的で、リモートツリー910及び同期ツリー950のみを図9に示し、説明する。同様の操作及び説明がローカルツリーにも当てはまり得る。

【0149】

リモートツリー910は、ファイル識別子1を有するルートノード912と、ファイル識別子5及びファイル名「Foo」を有するノード914と、ファイル識別子6及びファイル名「Bar」を有するノード916と、ファイル識別子7及びファイル名「Bye」を有するノード918とを含む。同期ツリーは、ファイル識別子1を有するルートノード952を含む。

30

【0150】

図9に示すツリーデータ構造に基づいて、システムは、図9の参照符号980によって示すように、動作810においてファイル識別子5、6、及び7を有するノードが修正されており、それらのノードを修正済みコンテンツアイテムのリストに追加していることを特定している場合がある。動作815において、システムは、修正済みコンテンツアイテムのリスト内のノードへの修正のリストを決定する。リモートツリー910と同期ツリー950との比較から分かるように、ノード914、916、及び918がリモートツリー910に追加されている。より具体的には、図9の参照符号982によって示すように、ファイル識別子6及び名前「Bar」を有するノード916が、ファイル識別子5を有するノード914に子として追加されている。これは、参照符号982の「Add(6, 5, Bar)」エントリで表される。ファイル識別子7及び名前「Bye」を有するノード918が、ファイル識別子5を有するノード914に子として追加されている。これは、参照符号982の「Add(7, 5, Bye)」エントリで表される。ファイル識別子5及び名前「Foo」を有するノード914が、ファイル識別子1を有するルートノード912に子として追加されている。これは、参照符号982の「Add(5, /root, Foo)」エントリで表される。

40

【0151】

動作820において、システムは、ノード914の追加修正が依存関係を有さず、その

50

結果、ブロックされないと判定する。したがって、システムは、動作 8 2 5 において、ノード 9 1 4 に関連する修正（たとえば、参照符号 9 8 2 の「Add ( 5 , / r o o t , F o o ) 」エントリによって表される修正）をブロックされないアクションのリストに追加する。これは図 9 の参照符号 9 8 4 に見られる。一方、参照符号 9 8 2 の「Add ( 6 , 5 , B a r ) 」エントリ及び「Add ( 7 , 5 , B y e ) 」エントリで表されるノード 9 1 6 及び 9 1 8 の修正は、最初に行われる「Add ( 5 , / r o o t , F o o ) 」によって表される修正に依存する。換言すれば、ノード 9 1 4 が追加されるまで、ノード 9 1 6 及びノード 9 1 8 は追加することができない。したがって、これらの修正は、図 9 の参照符号 9 8 6 に示すブロックされたアクションのリストに含まれる。

#### 【 0 1 5 2 】

図 8 の方法 8 0 0 に戻ると、動作 8 3 5 において、システムは、ブロックされないアクションのリストから修正セットを選択し、選択された修正セットに基づいて操作セットを生成し得る。操作セットは、サーバ状態とファイルシステム状態とを収束させるように構成される。生成される操作セットは、ブロックされないリストから選択された修正セットに依存する。たとえば、選択された修正セットが、図 9 のノード 9 1 4 に関連する追加修正（たとえば、参照符号 9 8 4 の「Add ( 5 , / r o o t , F o o ) 」エントリで表される修正）を含む場合、生成される操作セットは、追加されるコンテンツアイテムをコンテンツ管理システムから取り出し、これをクライアントデバイスのローカルファイルシステムに追加することを含み得る。

#### 【 0 1 5 3 】

一部の実施形態によれば、システムは、ブロックされないアクションのリストから全ての修正を選択して、1 つまたは複数の操作セットを生成し得る。しかしながら、一部のシナリオでは、ブロックされないリストの修正の数は非常に多い場合があり、全ての修正を処理するために必要なコンピューティングリソース（たとえば、メモリ及び処理時間）はかなりのものである。これらの技術的な負担を軽減するために、システムは、ブロックされないアクションのリスト内のより少ない修正のセットを選択して、段階的に処理するようにし得る。たとえば、システムは、最初もしくは上位 X 個またはパーセントの修正を選択して、操作を生成し得る。処理のさらなる反復において、ブロックされないリストの残りの修正が処理され得る。

#### 【 0 1 5 4 】

一部の実施形態では、ブロックされないリストの修正は、ランク付けされて処理され得る。これらの修正は、たとえば、修正タイプ（たとえば、削除修正が追加修正よりも優先される）、修正に関連付けられたメタデータ（たとえば、より小さいサイズのコンテンツアイテムの追加修正が、より大きいサイズのコンテンツアイテムの追加修正よりも優先される、より大きいサイズのコンテンツアイテムの削除修正が、より小さいサイズのコンテンツアイテムの削除修正よりも優先される、など）に基づいてランク付けされ得る。

#### 【 0 1 5 5 】

これらのランクルールはシステムによって記憶され得、コンテンツ同期の様々なパフォーマンス目標を達成するように設計され得る。たとえば、新たなコンテンツアイテムが追加され得る前に、ユーザに対して制限され得るストレージスペースをできるだけ多く解放するために、削除修正が追加修正よりも優先され得る。追加されるコンテンツアイテムの数に関してできるだけ多くの進捗をできるだけ早く提供するために、より小さいコンテンツアイテムの追加が、より大きいコンテンツアイテムよりも優先され得る。

#### 【 0 1 5 6 】

動作 8 3 5 において、システムは、操作セットをコンテンツ管理システム及びノードまたはクライアントデバイスに提供し得る。上記のように、コンテンツ管理システムによって実行されたアクションに関連する修正は、クライアントデバイスに反映されていない場合がある。したがって、このシナリオでは、システムは、サーバ状態とファイルシステム状態とを収束させるようにクライアントデバイスに記憶されたコンテンツアイテムを操作するよう構成されるクライアント操作セットを生成し得、動作 8 3 5 において、このクライア

10

20

30

40

50

ント操作セットはクライアントデバイスに提供されて実行され得る。

【0157】

一方、クライアントデバイスによって実行されたアクションに関連する修正は、コンテンツ管理システムに反映されていない場合がある。したがって、このシナリオでは、システムは、サーバ状態とファイルシステム状態とを収束させるようにコンテンツ管理システムによって記憶されたコンテンツアイテムを操作するよう構成されるサーバ操作セットを生成し得、動作835において、このサーバ操作セットはコンテンツ管理システムに提供されて実行され得る。

【0158】

場合によっては、両方のケースが当てはまることもあり、クライアント操作セット及びサーバ操作セットが生成され、動作835においてそれらの目的の受信者に提供され得る。操作セットは、ツリー制約が維持されるようにするためのいくつかのチェックも含み得る。たとえば、図6に関して説明したように、操作セットによって様々な競合または制約が解決され得る。

【0159】

操作セット（複数可）が目的の受信者（複数可）に提供されると、この方法は動作805に戻り、新たな修正データを待機し得る。たとえば、図9に示すシナリオに関して、操作セットは、ノード914に関連付けられたコンテンツアイテムをコンテンツ管理システムから取り出し、これをクライアントデバイスのローカルファイルシステムに追加することを含み得る。これにより、ノード914に対応するノードがローカルツリー（図9には図示せず）及び同期ツリー950に追加されることになる。図8の処理800の次の反復で、参照符号982の「Add(6, 5, Bar)」エントリ及び「Add(7, 5, Bye)」エントリによって表されるノード916及びノード918の追加修正は、それらの親であるノード914が既に同期ツリーに追加されているので、ブロックされなくなる。したがって、参照符号982の「Add(6, 5, Bar)」エントリ及び「Add(7, 5, Bye)」エントリによって表されるノード916及びノード918の追加修正は、ブロックされないアクションのリストに追加され、サーバ状態とファイルシステム状態とを収束させるように構成される1つまたは複数の操作セットを生成するために使用され得る。

【0160】

操作セット（複数可）は、サーバ状態及びファイルシステム状態を段階的に収束させるための1つまたは複数のステップを提供し得る。段階的な処理の実装は、時にはより複雑になり得るが、段階的な処理は処理時間の短縮と必要なメモリの削減とを実現し得る。これら及び他の最初の技術的改善は、当然ながら追加の技術的改善につながる。たとえば、処理時間が短縮されるので、特定の修正が使用されないものになったり、最新でなくなったりする、クライアントデバイスまたはコンテンツ管理システムからのさらなる変更の可能性も低くなる。

【0161】

図9に関して、説明の目的で、コンテンツアイテム、修正、アクション、またはファイル識別子の様々なグループ化を、リストとして説明している。他のタイプのデータ構造も互換性がある。たとえば、ブロックされないアクションのリストをBツリーデータ構造として実装して、データをソートされたままにし、対数時間での検索、順次アクセス、挿入、及び削除を可能にし得る。

【0162】

スケジューラ

一部の実施形態では、クライアント同期サービスは、サーバ状態とファイルシステム状態とを収束させるように構成される操作のセットまたはシーケンスを生成し、操作をコンテンツ管理システムまたはクライアントデバイスに提供して実行させ得る。しかしながら、一部のシナリオでは、クライアントデバイスのファイルシステムまたはコンテンツ管理システム上の変更によって、生成された操作セットの実行中に、その操作セットが最新で

10

20

30

40

50

なくなったり、使用されないものになったりし得る。様々な実施形態は、これら及び他の技術的問題に対する技術的解決方法を提供することに向けられている。たとえば、クライアント同期サービスは、クライアントデバイスのファイルシステムまたはコンテンツ管理システム上の変更を監視し、クライアントデバイス及び/またはコンテンツ管理システムを必要に応じて更新するように構成され得る。さらに、クライアント同期サービスは、操作の同時実行を許可にすることにより、パフォーマンスを改善し、処理時間を短縮するように構成され得る。

#### 【0163】

一部の実施形態によれば、図2に示すクライアント同期サービス156のプランナ225は、順序なしの操作セットからなるプランまたは操作プランを生成し得る。プラン内の全ての操作は依存関係を有さず、その結果、別々のスレッドでまたは任意の順序で同時に実行することができる。プラン内の操作は、一部の実施形態によれば、状態及びツリーデータ構造を収束させるために、コンテンツ管理システム及び/またはクライアントデバイスによって取られ得る抽象的な命令である。命令の例には、コンテンツアイテムのリモートもしくはローカルの追加、コンテンツアイテムのリモートもしくはローカルの削除、コンテンツアイテムのリモートもしくはローカルの編集、またはコンテンツアイテムのリモートもしくはローカルの移動が含まれ得る。

#### 【0164】

図2に示すクライアント同期サービス156のスケジューラ230は、プランナ225から操作プランを受信し、プラン内の操作の実行を管理し、プランが更新または変更されたか否かを判定し、更新または変更されたプランの実行を管理するように構成され得る。たとえば、スケジューラ230は、ファイルシステムインターフェース205及びサーバインターフェース210と連携して、プラン内の操作を実施するのに必要なタスク及びステップを実行し得る。これは、ファイルシステムもしくはコンテンツ管理システムからの確認の受信、またはネットワーク接続がない場合、もしくはコンテンツアイテムが他のアプリケーションによってロックされている場合の再試行の処理などのエラー処理アクティビティを含み得る。

#### 【0165】

各操作は、タスクと呼ばれるスクリプトまたはスレッドによって実装され得る。タスクは、関連する操作の適用を調整し、操作の実装に必要な1つまたは複数のステップを含み得る。たとえば、「ローカル追加操作」は、コンテンツアイテムがクライアントデバイスのローカルファイルシステムに追加されており、その結果、サーバ状態とファイルシステム状態とを同期させるために、コンテンツ管理システムにコンテンツアイテムが追加される必要があることを示し得る。したがって、ローカル追加操作は、ローカル追加操作を実装するために必要な1つまたは複数のステップを含む「ローカル追加タスク」に関連付けられ得る。これらのステップは、コンテンツ管理システムに新たなコンテンツアイテムを通知することと、コンテンツアイテムを1つまたは複数のデータブロックでコンテンツ管理システムにアップロードすることと、全てのデータブロックがコンテンツ管理システムによって受信されたことを確認することと、コンテンツアイテムが破損していないことを確認することと、コンテンツアイテムのメタデータをコンテンツ管理システムにアップロードすることと、コンテンツ管理システムの適切な場所へのコンテンツアイテムの追加をコミットすることと、のうちの1つまたは複数を含み得る。

#### 【0166】

タスクは実行を開始し、他のイベントの完了を待機しながら明確に定義されたポイントで一時停止し、イベントが発生したときに再開し、最終的に終了し得る。一部の実施形態によれば、スケジューラ230は、タスクをキャンセル、再生成、または置換するように構成される。たとえば、サーバ状態またはファイルシステム状態への変更に基づいて、タスクは実行される前に古くなり得、スケジューラ230は古くなったタスクを実行前にキャンセルし得る。

#### 【0167】

10

20

30

40

50

上述のように、プランナ 2 2 5 は、ツリーデータ構造のセット（たとえば、リモートツリー、同期ツリー、及びローカルツリー）に基づいて操作プランを生成し得る。時間と共に、プランナ 2 2 5 はツリーデータ構造のステータスに基づいて操作プランを生成し続ける。ツリーデータ構造がサーバ状態及びファイルシステム状態の状態を反映するように変化した場合、プランナ 2 2 5 は、前のプランと異なる新たな更新されたプランも生成し得る。スケジューラ 2 3 0 は、プランナ 2 2 5 によって生成された各操作プランを実行する。  
【 0 1 6 8 】

一部のシナリオでは、後のプランの操作の変更により、実行中の前のプランの操作との意図しない同期動作の競合が発生し得る。たとえば、第 1 のプランの操作が実行されているときに、操作のうちの 1 つまたは複数第 2 のプランにおいてキャンセルされる（または存在しない）。例として、図 1 0 に例示的なシナリオを示しており、時刻  $t_1$  において、全て一致するリモートツリー、同期ツリー、及びローカルツリーによって示すように、リモートツリーで表されるサーバ状態とローカルツリーで表されるファイルシステム状態とが同期している。この同期した状態に基づいて、プランナ 2 2 5 は、 $t_1$  において操作を有さないプラン（たとえば、空のプラン）を生成し得、または操作プランを生成しなくてもよい。

【 0 1 6 9 】

クライアントデバイスのユーザは、ローカルファイルシステムからコンテンツアイテム A を削除するか、またはクライアント同期サービス 1 5 6 によって管理されるフォルダの外へコンテンツアイテム A を移動し得、これは時刻  $t_2$  におけるローカルツリーからのノード A の除去によって反映される。プランナ 2 2 5 は、時刻  $t_2$  におけるツリーデータ構造の状態に基づいて操作 `Local Delete (A)` を含むプランを生成し得る。スケジューラ 2 3 0 は、`Local Delete (A)` 操作を実施するのに必要なタスクまたはステップを開始し得る。これらのステップは、コンテンツアイテム A を削除する命令をコンテンツ管理システムに送信することを含み得る。

【 0 1 7 0 】

コンテンツアイテム A を削除する命令がコンテンツ管理システムに送信された後、クライアントデバイス上のユーザは、コンテンツアイテム A の削除を元に戻すか、またはコンテンツアイテム A を以前の場所に戻し得る。ローカルツリーは、時刻  $t_3$  のこの新たなアクションに基づいて更新され、プランナは、操作を有さない空の新たなプランを生成し得る。この場合も、ツリーデータ構造は一致し、システムは時刻  $t_3$  に同期した状態になる。

【 0 1 7 1 】

しかしながら、コンテンツアイテム A を削除する命令がコンテンツ管理システムに送信されたので、コンテンツ管理システムはサーバ状態からコンテンツアイテム A を削除する。スケジューラ 2 3 0 はコンテンツアイテム A の削除のキャンセルを試み得るが、命令は既に送信され、コンテンツ管理システムによって完了している場合がある。サーバのこの変更は、クライアント同期サーバ 1 5 6 に通信され、クライアント同期サーバ 1 5 6 は、時間  $t_4$  にノード A を削除することによりリモートツリーを更新する。プランナ 2 2 5 は、リモートツリー内の変更と、リモートツリー及び同期ツリーの間の変更とを通知し、サーバ状態においてコンテンツアイテム A が除去されたと判定することができる。したがって、プランナ 2 2 5 は、時刻  $t_4$  において `Remote Delete (A)` 操作を有するプランを作成する。サーバ状態とファイルシステム状態とを同期させるために、コンテンツアイテム A は最終的にクライアントデバイス及びローカルツリーから削除される。

【 0 1 7 2 】

問題なことに、サーバ状態からのコンテンツアイテム A の除去、`Remote Delete (A)` 操作の生成、及びファイルシステム状態からのコンテンツアイテム A の最終的な除去は、全て意図したものではなく、ユーザにとって将来さらなる問題を引き起こし得る。さらに、一部のケースでは、アプリケーションまたは処理もコンテンツアイテムにアクセスし得、意図しない同期動作が追加の技術的問題の連鎖を引き起こし得る。様々な実施形態は、サーバ状態とファイルシステム状態との間のコンテンツアイテムの同期におけ

10

20

30

40

50

る意図しない結果を防ぐことに向けられている。

【0173】

一部の実施形態によれば、もはや操作プランにない古くなった操作のタスクをキャンセルする場合、スケジューラ230は、他のタスクの実行の開始に進む前にキャンセルが完了するのを待機し得る。たとえば、スケジューラ230は、他のタスクを進める前に、クライアントデバイスまたはコンテンツ管理システムからキャンセルの確認を受信するのを待機し得る。スケジューラ230は、タスクが開始されたか否かを判定し得、タスクが開始されていない場合、スケジューラはタスクをキャンセルし、タスクが実行を待機しなくなったことを確認し得る。タスクが開始された場合、確認がクライアントデバイスまたはコンテンツ管理システムから到来し、キャンセルされたタスクに関連する全てのステップが元に戻されたことをスケジューラに通知し得る。一部の実施形態によれば、スケジューラ230はタスクのキャンセルを、開始後には許可しない。これは、全てのタスク、またはタスクもしくはタスクタイプの特定のサブセット（たとえば、ファイルシステム状態の更新をコンテンツ管理システムに送信して、サーバ状態と同期させるコミットタスク）に当てはまり得る。

10

【0174】

パフォーマンスを改善し、タスクの同時実行だけでなくタスクのキャンセルも許可にするために、スケジューラ230は、第1の操作プランと更新された第2の操作プランとの差異に基づいてタスクの実行及びキャンセルを管理するようにさらに構成され得る。図11に、本主題技術の様々な実施形態による、2つの操作プランの例示的なベン図1100表現を示す。プランナ225は、第1の操作セットを有するプラン1110を生成し、ツリーデータ構造への更新を受信し、第2の操作セットを有する更新されたプラン21120を生成し得る。

20

【0175】

プラン1110及びプラン21120は、いくつかの共通の操作を共有し得、これはベン図1100の部分1130によって表される。また、プラン1110とプラン21120は、共通していないいくつかの操作を共有し得る。たとえば、プラン21120にないプラン1110の操作は、プランナ225によって検出されたツリー構造の更新によって古くなり、最新ではなくなる。プラン1110のこれらの古くなった操作は、ベン図1100の部分1140によって表される。プラン1110にないプラン21120の新たな操作は、部分1150によって表される。プラン1110とプラン21120との差異及び共通点を表す部分1130、1140、及び1150のそれぞれは、ツリーデータ構造に反映されるサーバ状態及びファイルシステム状態の更新に応じて、操作を含まない場合があり、または多数の操作を含み得る。

30

【0176】

部分1140の操作はもはや最新のプランにないので、スケジューラ230はこれらの操作に関連するタスクをキャンセルし得る。意図しない同期動作を防ぐために、プラン1にない（たとえば、部分1150内の）プラン2の操作に関連するタスクは、部分1140の操作に関連するタスクのキャンセルが完了するまで延期される。しかしながら、各プランの操作は同時に実行できるように構成されるので、部分1130で表されるプラン1とプラン2との共通部分の操作に関連するタスクは、部分1140の操作に関連するタスクのキャンセルと同時に、それらの完了を待機する必要なく実行され得る。部分1140に関連するタスクのキャンセルと、部分1130に関連するタスクの実行とを同時に行うことを許可にすることにより、利用可能なコンピューティングリソースのより効率的な使用に加え、処理時間の短縮が達成され得る。

40

【0177】

図12に、本主題技術の様々な実施形態による、操作プランの変更を管理するための例示的な方法を示す。本明細書に記載の方法及び処理は、特定の順序の特定のステップ及び動作と共に示す場合があるが、同様または代替の順序でまたは並行して実行される追加の、より少ない、または代替のステップ及び動作は、特に明記しない限り、様々な実施形態

50

の範囲内にある。方法 1 2 0 0 は、たとえば、クライアントデバイス上で動作する図 2 のクライアント同期サービス 1 5 6 などのシステムによって実装され得る。

【 0 1 7 8 】

システムは、コンテンツ管理サービスに関連付けられたコンテンツアイテムに関して、コンテンツ管理システム及び/またはクライアントデバイスから更新を受信するように構成され得る。たとえば、システムは、コンテンツ管理サービスによって記憶されたコンテンツアイテムのサーバ修正データを受信し、サーバ修正データに基づいてリモートツリーを更新し得る。リモートツリーは、コンテンツ管理システムによって記憶されたコンテンツアイテムのサーバ状態を表す。システムはまた、クライアントデバイスに記憶されたコンテンツアイテムのクライアント修正データを受信し、クライアント修正データに基づいてローカルツリーを更新し得る。ローカルツリーは、クライアントデバイスに記憶されたコンテンツアイテムのファイルシステム状態を表す。

10

【 0 1 7 9 】

動作 1 2 0 5 において、システムは、コンテンツ管理システムに関連するサーバ状態と、クライアントデバイスに関連するファイルシステム状態とを収束させるように構成される第 1 の操作セットを受信し得る。たとえば、システムは、同期ツリーとリモートツリーまたは同期ツリーとローカルツリーの差異を特定し、ツリー間の任意の差異に基づいて第 1 の操作セットを生成し得る。同期ツリーは、サーバ状態とファイルシステム状態との間の既知の同期した状態を表す。

【 0 1 8 0 】

システムは、第 1 の操作セットの実施を開始し得る。たとえば、一部のケースでは、操作は、コンテンツ管理システム及び/またはクライアントデバイスに送信されて実行される準備ができていない形式である。他のケースでは、操作は、システムによって管理され得る 1 つまたは複数のタスク、スクリプト、または実行スレッドに変換され得る。システムは、サーバ状態とファイルシステム状態とを収束させるために、タスク、スクリプト、または実行スレッドに従ってコンテンツ管理システム及び/またはクライアントデバイスとインターフェースし得る。

20

【 0 1 8 1 】

この時間中、システムは、コンテンツ管理サービスに関連付けられたコンテンツアイテムに関して、コンテンツ管理システム及び/またはクライアントデバイスから修正データを受信し続け得る。修正データに基づいて、システムはリモートツリーまたはローカルツリーを更新し、ツリーデータ構造への更新に基づいて第 2 の操作セットを生成し得る。動作 1 2 1 0 において、システムは第 2 の操作セットを受信し得る。

30

【 0 1 8 2 】

動作 1 2 1 5 において、システムは、第 2 の操作セットにない第 1 の操作セットの第 1 の操作が存在する場合に、これを特定する。システムが、第 2 の操作セットにない第 1 の操作セット内の操作を見つけた場合、この操作は、修正データで示された変更の結果、古くなり、最新でなくなり得る。したがって、システムは、動作 1 2 2 0 における第 1 の操作のキャンセルを開始する。第 1 の操作のキャンセルには、いくつかのステップ、それらのステップのいくつかの確認受信、及び些細でない処理時間が含まれ得る。

40

【 0 1 8 3 】

動作 1 2 2 5 において、システムは、第 1 の操作セット及び第 2 の操作セットの両方に含まれる第 2 の操作が存在する場合に、これを特定する。システムが第 1 の操作セット及び第 2 の操作セットの両方に含まれる操作を見つけた場合、この操作は、修正データで示された変更にもかかわらず、まだ有効であり得る。さらに、両方の操作セットにある操作は、セット内の他の操作に対して同時にまたは任意の順序で実行できるように構成されるので、第 2 の操作は、第 1 の操作がキャンセルされる間に実行を継続することができる。したがって、システムは、第 1 の操作がキャンセルを完了するのを待機することなく、動作 1 2 3 0 において第 2 の操作の実行を開始する。

【 0 1 8 4 】

50

動作 1 2 3 5 において、システムは、第 2 の操作セットにあるが、第 1 の操作セットにはない第 3 の操作が存在する場合、これを特定する。システムが第 1 の操作セットにない第 2 の操作セットにある操作を見つけた場合、この操作は、修正データで示された変更の結果としての新たな操作であり得る。意図しない結果を防ぐために、システムは第 1 の操作のキャンセルの完了の待機を開始する。動作 1 2 4 0 において、システムは、第 1 の操作がキャンセルを完了したと判定し、その結果、動作 1 2 4 5 において第 3 の操作の実行を開始し得る。

**【 0 1 8 5 】**

ツリーデータ構造ストレージ - ファイル名ストレージスペースの削減

クライアント同期サービス 1 5 6 は、たとえば、ハードディスク、ソリッドステートメモリ、または他のタイプのコンピュータ可読媒体などの永続的な記憶デバイス上にツリーデータ構造（たとえば、リモートツリー、同期ツリー、及びローカルツリー）を記憶し得る。パフォーマンスを改善し、処理時間を短縮し、期限切れの操作を削減するために、クライアント同期サービス 1 5 6 は、起動時にツリーデータ構造をメモリ（たとえば、ランダムアクセスメモリまたは高速メモリのキャッシュ）にロードし、メモリ内のツリーデータ構造に同期機能を実行し得る。永続的な記憶デバイスのデータ容量は限られており、これらのデータリソースの保存は重要である。メモリの場合、データ容量はさらに限られており、高価であり、これらのデータリソースの保存は非常に重要である。

10

**【 0 1 8 6 】**

オペレーティングシステムまたはクライアントアプリケーションに応じて、コンテンツアイテムのファイル名は、サイズが約 1 0 2 4 バイトである場合があり、ノード内で最大のデータコンポーネントであり得る。たとえば、1 0 0 万個のノードの場合、ノードのファイル名だけのサイズで 1 ギガバイト以上に達し得る。上記のように、クライアント同期サービス 1 5 6 は、サーバ状態とファイルシステム状態との同期を支援するように構成され、ローカルツリー、同期ツリー、及びリモートツリーは、3 つ全てのツリーが同等の場合に同期した状態を反映する。したがって、コンテンツアイテムのファイル名をノードに記憶する際には、いくらかの冗長性がある可能性がある。

20

**【 0 1 8 7 】**

本主題技術の様々な実施形態は、ツリーデータ構造のファイル名を記憶するために必要なメモリ量を削減し、ファイル名の重複を減らすことによりノードのサイズを減らすことを目標とする。ファイル名をノードに記憶する代わりに、クライアント同期サービス 1 5 6 は、ツリーデータ構造内のノードのファイル名をファイル名配列に記憶し、ファイル名への参照をノードに記憶するように構成される。その結果、ファイル名は一旦ファイル名配列に記憶され、そのファイル名を持つコンテンツアイテムを有する任意のノードは、ノードに記憶された参照を使用してファイル名にアクセスし得る。一部の実施態様では、ノードに記憶されているファイル名への参照は、ファイル名配列におけるファイル名のオフセット、場所、または位置を表す整数値であり得る。

30

**【 0 1 8 8 】**

図 1 3 に、本主題技術の様々な実施形態による、ファイル名配列 1 3 1 0 の実例を示す。ファイル名「P i c t u r e s」、「a . j p g」、及び「D o c u m e n t s」（完全には図示せず）を記憶したファイル名配列 1 3 1 0 を図示している。ファイル名配列 1 3 1 0 において、「P i c t u r e s」のファイル名は場所 0 に、「a . j p g」のファイル名は場所 9 に、「D o c u m e n t s」のファイル名は場所 1 6 に示されている。ファイル名配列 1 3 1 0 では、セパレータ 1 3 1 4（たとえばヌル文字）により各ファイル名が区切られている。

40

**【 0 1 8 9 】**

したがって、ノードのファイル名を調べるために、クライアント同期サービス 1 5 6 は、ファイル名配列 1 3 1 0 内のノードのファイル名の場所を表す、ノードに記憶されたファイル名への参照に単にアクセスし得る。クライアント同期サービス 1 5 6 は、参照によって示されたファイル名配列 1 3 1 0 内の場所にあるコンテンツアイテムのファイル名を

50

取り出し得る。たとえば、クライアント同期サービス 1 5 6 は、参照によって示された場所にあるファイル名の読み出しを開始し、セパレータ 1 3 1 4 に到達したときに停止し得る。

#### 【 0 1 9 0 】

一部のケースでは、クライアント同期サービス 1 5 6 はまた、ファイル名に基づいて参照を調べる必要があり得る。たとえば、新たなノードを追加するかまたはノードの名前を変更する場合、クライアント同期サービス 1 5 6 は、ノードのファイル名がファイル名配列 1 3 1 0 にすでに存在するか否かを判定することを望み得る。参照が見つかった場合、そのファイル名は存在し、参照に基づいて位置特定され得る。したがって、クライアント同期サービス 1 5 6 は、ファイル名への参照を使用し、参照を新たなノードまたは名前変更されたノードに記憶し得る。参照が見つからない場合、ファイル名はファイル名配列 1 3 0 に存在せず、クライアント同期サービス 1 5 6 はそのファイル名をファイル名配列 1 3 1 0 に追加し得る。

10

#### 【 0 1 9 1 】

ファイル名に基づいて参照を調べることは、ハッシュインデックス配列を使用することにより可能になる。図 1 3 に、本主題技術の様々な実施形態による、ハッシュインデックス配列 1 3 5 0 の実例を示す。ハッシュインデックス配列 1 3 5 0 は、様々な場所にあるファイル名への参照を記憶するように構成される。特に、ファイル名への参照は、そのファイル名のハッシュに基づいてハッシュインデックス配列 1 3 5 0 内の位置に記憶される。図 1 3 に示す一例では、使用されるハッシュ関数に基づいて、ファイル名「P i c t u r e s」のハッシュは 4 に等しくなり得る。したがって、ファイル名「P i c t u r e s」への参照は、ハッシュインデックス配列 1 3 5 0 内の位置 4 に記憶される。この参照値は 0 であり、これは上記のように、ファイル名「P i c t u r e s」のファイル名配列 1 3 1 0 内の場所を示す。同様に、ファイル名「D o c u m e n t s」のハッシュは 0 に等しくなり得る。したがって、ファイル名「D o c u m e n t s」への参照は、ハッシュインデックス配列 1 3 5 0 内の位置 0 に記憶される。この参照値は 1 6 であり、これは上記のように、ファイル名「D o c u m e n t s」のファイル名配列 1 3 1 0 内の場所を示す。

20

#### 【 0 1 9 2 】

一部のシナリオでは、2つのファイル名のハッシュが同じハッシュ値を生成する場合、衝突が発生し得る。図 1 3 に示す例では、「a . j p g」のハッシュ値は 4 にもなり得る。衝突が発生した場合、クライアント同期サービス 1 5 6 は、ハッシュインデックス配列 1 3 5 0 内の次に利用可能な位置を使用し得る。たとえば、ファイル名「P i c t u r e s」への参照はハッシュインデックス配列 1 3 5 0 の位置 4 に記憶されているので、クライアント同期サービス 1 5 6 は、ハッシュインデックス配列 1 3 5 0 内の次に利用可能な位置を探し、ファイル名「a . j p g」への参照（たとえば 9）を記憶し得る。

30

#### 【 0 1 9 3 】

図 1 4 に、本主題技術の様々な実施形態による、ファイル名を記憶するための例示的な方法を示す。本明細書に記載の方法及び処理は、特定の順序の特定のステップ及び動作と共に示す場合があるが、同様または代替の順序でまたは並行して実行される追加の、より少ない、または代替のステップ及び動作は、特に明記しない限り、様々な実施形態の範囲内にある。方法 1 4 0 0 は、たとえば、クライアントデバイス上で動作する図 2 のクライアント同期サービス 1 5 6 などのシステムによって実装され得る。

40

#### 【 0 1 9 4 】

動作 1 4 0 5 において、システムは、ツリーデータ構造内のノードへの修正を検出し得る。修正は、たとえば、ツリーデータ構造へのノードの追加、またはノードのファイル名の編集であり得る。次いで、システムは、動作 1 4 1 0 において、ファイル名がファイル名配列にすでに存在するか否かを判定し得る。システムは、ノードに関連付けられたファイル名を使用して、ファイル名配列内のファイル名の参照場所の問い合わせを行うことにより、ファイル名が既にファイル名配列内にあるか否かを確認し得る。参照場所が見つかった場合、ファイル名をファイル名配列に追加する必要はなく、動作 1 4 1 5 において、

50

システムは、ファイル名配列内のファイル名の場所を特定し、これは問い合わせによって返される参照であるべきであり、動作 1 4 2 0 において、ファイル名の場所をノードに記憶し得る。

【 0 1 9 5 】

問い合わせによってファイル名配列内のファイル名の参照が返されない場合、またはファイル名が別の形でファイル名配列内に存在しない場合、システムは動作 1 4 2 5 においてファイル名配列にファイル名及びセパレータを付加し、動作 1 4 3 0 においてファイル名配列内のファイル名の場所を特定し、動作 1 4 3 5 においてファイル名の場所をノードに記憶し得る。

【 0 1 9 6 】

ファイル名に基づいたその後の参照場所の問い合わせを可能にするために、システムはさらにファイル名の場所をハッシュインデックスに記憶し得る。ハッシュインデックス内のどの位置にファイル名の場所を記憶するかを決定するために、システムはファイル名のハッシュ値を計算し得る。このハッシュ値は、ファイル名の場所を記憶するハッシュインデックス内の位置を見つけるために使用され得る。

【 0 1 9 7 】

ファイル名の場所がノードに記憶されると、ファイル名を取り出すには、ノード内のファイル名の場所にアクセスし、その場所を使用してファイル名配列内でファイル名を調べるだけでよい。システムは、ノード内で指定された場所から開始し、セパレータに到達すると停止し得る。

【 0 1 9 8 】

図 1 5 に、本主題技術の様々な実施形態による、ファイル名が与えられた場合にファイル名の場所を取得するための例示的な方法を示す。本明細書に記載の方法及び処理は、特定の順序の特定のステップ及び動作と共に示す場合があるが、同様または代替の順序でまたは並行して実行される追加の、より少ない、または代替のステップ及び動作は、特に明記しない限り、様々な実施形態の範囲内にある。方法 1 5 0 0 は、たとえば、クライアントデバイス上で動作する図 2 のクライアント同期サービス 1 5 6 などのシステムによって実装され得る。上述のように、この方法を使用して、ファイル名がファイル名配列にすでに記憶されているか否かを判定し得る。

【 0 1 9 9 】

ファイル名の場所は、ハッシュインデックスまたはハッシュインデックス配列に記憶される。したがって、動作 1 5 0 5 において、システムは、ファイル名にハッシュ関数を実行することにより、ハッシュインデックス内の位置を生成し得る。これは、たとえば、新たなノードの名前、または名前変更されたノードの新たな名前であり得る。動作 1 5 1 0 において、システムは、ハッシュインデックス内の位置から、ファイル名の場所情報を取り出す。この場所情報は、ファイル名が記憶されているファイル名配列内の場所に関するものである。

【 0 2 0 0 】

一部の実施態様では、システムは、動作 1 5 1 5 において正しいファイル名がその場所に記憶されていることを確認し得る。たとえば、システムは、場所情報に基づいて、ファイル名配列から文字列を取り出し、文字列をファイル名と比較し得る。文字列とファイル名とが一致する場合、ファイル名の場所が確認され、動作 1 5 2 0 においてファイル名の場所がノードに記憶される。

【 0 2 0 1 】

文字列とファイル名とが一致しない場合、動作 1 5 3 0 においてハッシュインデックス内の次の位置の場所情報を取り出し、システムは動作 1 5 1 5 に戻って、次の位置の場所情報が正確であるか否かを確認し得る。システムは、正しい場所情報が見つかり、動作 1 5 2 0 においてノードに記憶されるまで継続し得る。

【 0 2 0 2 】

ツリー間の差異を効率的に特定する

10

20

30

40

50

上記のように、クライアント同期サービス156は、コンテンツ管理システムによって記憶されたコンテンツアイテムのサーバ状態を表すリモートツリーと、クライアントデバイスに記憶された対応するコンテンツアイテムのファイルシステム状態を表すローカルツリーと、サーバ状態及びファイルシステム状態の間の既知の同期した状態を表す同期ツリーとにおけるノード間の差異を特定するように構成される。これらの差異に基づいて操作セットが生成され得、操作セットは、実行された場合に、3つのツリーデータ構造が同一になる同期した状態へ向けてサーバ状態及びファイルシステム状態を収束させるように構成される。

#### 【0203】

多数のノードが存在する場合、ツリー間の差異を効率的に特定できることが重要である。たとえば、ツリーに数百万のノードが存在し得、処理時間及びリソース使用量の観点から、各ノードを個別に比較することを禁止することができる。

10

#### 【0204】

本主題技術の様々な実施形態は、ツリー間の差異を特定するより効率的な手段を提供することに関する。特に、クライアント同期サービス156は、他のツリーのノードと比較して、それらのノードが異なるか否かを判定するために使用可能な値を各ノードに割り当てるように構成される。これらの値はdiff値と呼ばれ得る。ツリー間の差異を特定する効率を改善するために、各リーフノードにdiff値が割り当てられ得、親ノードのdiff値が、それらの子ノードのdiff値に基づいて計算され得る。ツリーデータ構造の全てのレベルのdiff値が、ルートノードを含めてこの方法で計算され得る。

20

#### 【0205】

図16A及び図16Bに、様々な実施形態による、ツリーデータ構造の例を示す。説明の目的で、図16A及び図16Bに、同期ツリー及びローカルツリーを示す。しかしながら、リモートツリーも同様に実行し得る。図16A及び図16Bでは、同期ツリー及びローカルツリーのリーフノードのdiff値が、ノードにハッシュ関数を使用することによって計算される。各親ノード（すなわち、子を有するノード）は、それらの子ノードのdiff値に基づいて計算されたdiff値を有する。

#### 【0206】

たとえば、図16Aの同期ツリー1605では、ノードCのdiff値は325であり、ノードDのdiff値は742であり、ノードC及びDの親であるBのdiff値は、C及びDのdiff値の関数として計算される。換言すれば、 $DiffValue(B) = f(DiffValue(C), DiffValue(D)) = f(325, 742)$ となる。同様に、ルートノードのdiff値は、その子ノードであるノードA及びノードBのdiff値の関数である。別の言い方をすると、 $DiffValue(/root) = f(DiffValue(A), DiffValue(B)) = f(924, 789)$ となる。

30

#### 【0207】

ツリー間の差異を特定するために、クライアント同期サービス156は、対応するノードのdiff値を比較して、それらが異なるか否かを確認し得る。diff値が同じである場合、ツリー間に差異はない。diff値が異なる場合、ノードまたはノードに関連付けられたパスの下方に変更がある。その結果、クライアント同期サービス156は、子ノードを見て、反対側のツリー内の対応するノードを比較して、それらが異なるか否か、また、子ノードに関連付けられたパスの下方に差異があるか否かを判定することができる。

40

#### 【0208】

たとえば、図16Aでは、クライアント同期サービス156は、同期ツリー1605のルートノードとローカルツリー1610のルートノードとのdiff値を比較し、diff値が一致すると判定し得る。したがって、ルートノードの子に差異はない。その結果、クライアント同期サービス156は、ツリー内のありとあらゆるノードを反対側のツリー内の対応するノードと比較する必要なく、同期ツリー1605とローカルツリー1610とが一致し、同期していると判定し得る。

50

## 【0209】

図16Bに示す例では、クライアント同期サービス156は、同期ツリー1605のルートノードとローカルツリー1610のルートノードとのdiff値を同様に比較し得る。しかしながら、クライアント同期サービス156は、同期ツリー1605のルートノードとローカルツリー1610のルートノードとのdiff値が異なると判定し得、これはルートノードの子孫に差異が存在し得ることを示す。

## 【0210】

したがって、クライアント同期サービス156は、ルートノードの子の次のレベルに移動して、それらのdiff値を比較し得る。ノードAについて、同期ツリー1615のノードAのdiff値と、ローカルツリー1620のノードAのdiff値とが一致する。したがって、これら2つの間に差異はなく、ルートレベルで検出された差異は、ノードAによるものでも、ノードAの下のいかなるパスによるものでもない。

10

## 【0211】

ノードBに移動して、クライアント同期サービス156は、両方のツリーにおけるノードBのdiff値を比較し、同期ツリー1615のノードAのdiff値とローカルツリー1620のノードAのdiff値とが異なることを発見する。したがって、ルートレベルで検出された差異は、ノードBまたはノードBの下のパスによって生じている。クライアント同期サービス156は、ノードBの子の次のレベルに移動して、それらのdiff値を比較し、ルートレベルで検出された差異がノードDの削除によって生じたことを発見し得る。その結果、クライアント同期サービス156は、ツリー内のありとあらゆるノードを反対側のツリー内の対応するノードと比較する必要なく、同期ツリー1615とローカルツリー1620との差異を特定し得る。たとえば、ノードAは、両方のツリーにおけるノードAのdiff値が一致していたために分析される必要がなかった多くの子孫ノードを有していた場合がある。

20

## 【0212】

マークルツリーまたはハッシュツリーメカニズムが一部のケースで機能し得る。たとえば、各リーフノードのdiff値は、リーフノードのハッシュに基づいて計算され得、非リーフノードのdiff値は、子diff値の合計のハッシュに基づいて計算され得る。しかしながら、マークルツリーまたはハッシュツリーメカニズムでは、特定の状況でパフォーマンスメトリックが不利になる。リーフノードを追加または削除する場合に、リーフノードの全ての祖先ノードのdiff値は、それらのdiff値が再計算される必要があり、各祖先ノードを再計算するには、各祖先ノードの全ての子ノードのリストが必要になる。これは特に、各ノードが兄弟ノードの隣または近くのメモリに記憶されない方法でツリーデータ構造が記憶される場合に、計算コストが高くなる。

30

## 【0213】

本主題技術の様々な実施形態は、とりわけ、親ノードのdiff値を異なる方法で計算することにより、これら及び他の技術的欠点に対処する。各リーフノードのdiff値は、そのリーフノードのハッシュを計算することにより決定され得る。各親ノードのdiff値は、その全ての子のハッシュの排他的論理和演算すなわちXOR演算を実行することにより計算され得る。

40

## 【0214】

図17に、様々な実施形態による、ツリーデータ構造の一例を示す。図17のツリーデータ構造では、リーフノードA、C、及びDのdiff値は、これらのノードをハッシュ化することにより計算される。ノードBのdiff値は、その子ノードであるノードC及びノードDのハッシュのXOR関数である。ルートノードのdiff値は、その子ノードであるノードA及びノードBのハッシュのXOR関数である。XOR関数の重要な性質には、XOR関数は順序依存ではない、すなわち $m \text{ XOR } n$ が $n \text{ XOR } m$ に等しくなることが含まれる。また、 $m \text{ XOR } m$ は0に等しい。

## 【0215】

リーフノードを追加または削除する場合、リーフノードの全ての祖先ノードのdiff

50

値は、それらの `d i f f` 値が再計算される必要がある。しかしながら、全てのリストを必要とせずに、祖先ノードの `d i f f` 値の計算を行うことができる。たとえば、子ノードが削除される場合、親の古い `d i f f` 値と、削除される子ノードの `d i f f` 値との `X O R` 演算を実行することにより、親の新たな `d i f f` 値が計算され得る。子ノードが追加される場合、親の古い `d i f f` 値と、新たな子ノードの `d i f f` 値との `X O R` 演算を実行することにより、親の新たな `d i f f` 値が計算され得る。

#### 【0216】

図18に、本主題技術の様々な実施形態による、ファイル名が与えられた場合にファイル名の場所を取得するための例示的な方法を示す。本明細書に記載の方法及び処理は、特定の順序の特定のステップ及び動作と共に示す場合があるが、同様または代替の順序でまたは並行して実行される追加の、より少ない、または代替のステップ及び動作は、特に明記しない限り、様々な実施形態の範囲内にある。方法1800は、たとえば、クライアントデバイス上で動作する図2のクライアント同期サービス156などのシステムによって実装され得る。上述のように、この方法を使用して、ファイル名がファイル名配列にすでに記憶されているか否かを判定し得る。

10

#### 【0217】

動作1805において、クライアント同期サービス156は、リモートツリー、同期ツリー、またはローカルツリーなどのツリーデータ構造に対してノードを追加または除去し得る。ノードが削除された場合、ノードは、以前に計算された `d i f f` 値が関連付けられる必要がある。ノードが追加された場合、クライアント同期サービス156は、たとえばノードをハッシュ化することにより、新たなノードの `d i f f` 値を計算し得る。ツリーへの変更があるので、ノードの祖先の `d i f f` 値は更新される必要がある。

20

#### 【0218】

動作1810において、クライアント同期サービス156は、親ノードの現在の `d i f f` 値と、ノードの `d i f f` 値とに基づいて、ノードの親の新たな `d i f f` 値を計算し得る。動作1815において、親の新たな `d i f f` 値が親ノードに記憶される。

#### 【0219】

動作1820において、クライアント同期サービス156は、親ノード自体が親を有するか否かを判定する。換言すれば、親ノードがルートノードであるか、または `d i f f` 値を計算すべきさらなる祖先が存在するか否かである。さらなる親が存在する場合、処理は動作1810に戻り得、親ノードの親は、新たな `d i f f` 値が計算され、記憶される。さらなる親が存在せず、親ノードがルートノードである場合、処理は動作1825において停止し得る。

30

#### 【0220】

ルートノードに到達すると、ツリーデータ構造を他のツリーデータ構造と比較して差異を特定する準備が整う。上述のように、クライアント同期サービス156は、これらの差異に基づいて操作セットを生成し得、操作セットは、実行された場合に、3つのツリーデータ構造が同一になる同期した状態へ向けてサーバ状態及びファイルシステム状態を収束させるように構成される。

#### 【0221】

図19にコンピューティングシステム1900の一例を示し、これは、たとえば、クライアントデバイス150、コンテンツ管理システム110、またはその任意のコンポーネントを構成する任意のコンピューティングデバイスとすることができ、システムのコンポーネントは接続1905を使用して互いに通信する。接続1905は、バスを介した物理的な接続、またはチップセットアーキテクチャなどの、プロセッサ1910への直接接続とすることができる。接続1905は、仮想的な接続、ネットワーク接続、または論理接続とすることもできる。

40

#### 【0222】

一部の実施形態では、コンピューティングシステム1900は、本開示に記載の機能が1つのデータセンター、複数のデータセンター、ピアネットワークなどの中で分散させる

50

ことができる分散システムである。一部の実施形態では、説明するシステムコンポーネントのうちの1つまたは複数は、多くのそのようなコンポーネントを表し、各コンポーネントは、そのコンポーネントについて説明する機能の一部または全部を実行する。一部の実施形態では、これらのコンポーネントは物理デバイスまたは仮想デバイスとすることができる。

#### 【0223】

例示的なシステム1900は、少なくとも1つの処理ユニット(CPUまたはプロセッサ)1910と、読み取り専用メモリ(ROM)1920及びランダムアクセスメモリ(RAM)1925などのシステムメモリ1915を含む様々なシステムコンポーネントをプロセッサ1910に結合する接続1905とを含む。コンピューティングシステム1900は、プロセッサ1910に直接接続されるか、これに近接するか、またはその一部として統合される高速メモリ1912のキャッシュを含むことができる。

10

#### 【0224】

プロセッサ1910は、任意の汎用プロセッサと、プロセッサ1910を制御するように構成される、記憶デバイス1930に記憶されたサービス1932、1934、及び1936などのハードウェアサービスまたはソフトウェアサービスに加え、ソフトウェア命令が実際のプロセッサ設計に組み込まれる専用プロセッサを含むことができる。プロセッサ1910は、本質的に、複数のコアまたはプロセッサ、バス、メモリコントローラ、キャッシュなどを含む完全に自己完結型のコンピューティングシステムであり得る。マルチコアプロセッサは、対称または非対称であり得る。

20

#### 【0225】

ユーザインタラクションを可能にするために、コンピューティングシステム1900は、たとえば、音声用のマイクロフォン、ジェスチャまたはグラフィカル入力用のタッチ感応画面、キーボード、マウス、モーション入力、音声などの、任意の数の入力メカニズムを表すことができる入力デバイス1945を含む。コンピューティングシステム1900は出力デバイス1935も含むことができ、これは当業者に知られているいくつかの出力メカニズムのうちの1つまたは複数とすることができる。一部の例では、マルチモーダルシステムにより、ユーザは複数のタイプの入力/出力を提供して、コンピューティングシステム1900と通信することができる。コンピューティングシステム1900は通信インターフェース1940を含むことができ、これはユーザ入力及びシステム出力を全体的に統括し、管理することができる。特定のハードウェア構成での動作に制限されないので、ここでの基本的な特徴は、改良されたハードウェアまたはファームウェア構成が開発されたときに、それらと簡単に置き換えられ得る。

30

#### 【0226】

記憶デバイス1930は、不揮発性メモリデバイスとすることができる。また、コンピュータによってアクセス可能なデータを記憶できるハードディスクまたは他のタイプのコンピュータ可読媒体とすることができる。たとえば、磁気カセット、フラッシュメモリカード、固体メモリデバイス、デジタル多用途ディスク、カートリッジ、ランダムアクセスメモリ(RAM)、読み取り専用メモリ(ROM)、及び/またはこれらのデバイスの何らかの組み合わせなどがある。

40

#### 【0227】

記憶デバイス1930は、ソフトウェアサービス、サーバ、サービスなどを含むことができ、これらは、そのようなソフトウェアを定義するコードがプロセッサ1910によって実行された場合、システムに機能を実行させる。一部の実施形態では、特定の機能を実行するハードウェアサービスは、その機能を実行するために、プロセッサ1910、接続1905、出力デバイス1935などの必要なハードウェアコンポーネントに関連するコンピュータ可読媒体に記憶されたソフトウェアコンポーネントを含むことができる。

#### 【0228】

説明を明確にするために、一部の例では、本技術は、デバイス、デバイスコンポーネント、ソフトウェアで具現化される方法のステップもしくはルーチン、またはハードウェア

50

とソフトウェアとの組み合わせを含む機能ブロックを含む個々の機能ブロックを含むものとして提示され得る。

【0229】

本明細書に記載のステップ、動作、機能、または処理はいずれも、ハードウェア及びソフトウェアサービスまたはサービスの組み合わせによって、単独でまたは他のデバイスと組み合わせて実行または実装され得る。一部の実施形態では、サービスは、クライアントデバイス及び/またはコンテンツ管理システムの1つまたは複数のサーバのメモリに常駐するソフトウェアとすることができ、プロセッサがサービスに関連付けられたソフトウェアを実行した場合に1つまたは複数の機能を実行することができる。一部の実施形態では、サービスは、特定の機能を実行するプログラムまたはプログラムの集合体である。一部の実施形態では、サービスはサーバと見なすことができる。メモリは非一時的コンピュータ可読媒体とすることができる。

10

【0230】

一部の実施形態では、コンピュータ可読記憶デバイス、媒体、及びメモリは、ビットストリームなどを含む有線または無線信号を含むことができる。しかしながら、言及する場合、非一時的コンピュータ可読記憶媒体は、エネルギー、搬送波信号、電磁波、及び信号自体などの媒体を明示的に除外する。

【0231】

上述の例による方法は、コンピュータ可読媒体に記憶されるか、またはそこから別の方法で利用可能なコンピュータ実行可能命令を使用して実装することができる。そのような命令は、たとえば、汎用コンピュータ、専用コンピュータ、または専用処理デバイスに、特定の機能または機能のグループを実行させるか、または別の方法でそれらをそのように構成する命令及びデータを含むことができる。使用されるコンピュータリソースの一部は、ネットワーク経由でアクセス可能にすることができる。コンピュータ実行可能命令は、たとえば、バイナリ、アセンブリ言語などの中間フォーマット命令、ファームウェア、またはソースコードであり得る。記載の例による方法の間の、命令、使用される情報、及び/または作成された情報を記憶するために使用され得るコンピュータ可読媒体の例には、磁気または光ディスク、ソリッドステートメモリデバイス、フラッシュメモリ、不揮発性メモリが設けられたUSBデバイス、ネットワーク型ストレージデバイスなどが含まれる。

20

【0232】

これらの開示による方法を実装するデバイスは、ハードウェア、ファームウェア及び/またはソフトウェアを含むことができ、多様なフォームファクタのいずれかを取ることができる。そのようなフォームファクタの典型的な例には、サーバ、ラップトップ、スマートフォン、スモールフォームファクタのパーソナルコンピュータ、携帯情報端末などが含まれる。本明細書に記載の機能は、周辺機器またはアドインカードでも具現化することができる。そのような機能は、さらなる例として、回路基板上に、異なるチップまたは単一のデバイスで実行される異なるプロセスの間で実装することもできる。

30

【0233】

これらの命令、そのような命令を伝えるための媒体、それらを実行するためのコンピューティングリソース、及びそのようなコンピューティングリソースをサポートするための他の構造は、これらの開示に記載の機能を提供するための手段である。

40

【0234】

多様な例及び他の情報を使用して、添付の特許請求の範囲内の態様を説明したが、そのような例の特定の特徵または構成に基づいて特許請求の範囲への制限が暗示されるべきではなく、その理由は、当業者であれば、これらの例を使用して、多種多様な実施態様を導出できるためである。さらに、一部の主題は、構造的な特徴及び/または方法のステップの例に特有の文言で説明している場合があるが、添付の特許請求の範囲で定義する主題は、必ずしもこれらの説明した特徴または行為に限定されないことを理解されたい。たとえば、そのような機能は、異なる方法で配布したり、本明細書で特定したもの以外のコンポーネントで実行したりすることができる。むしろ、説明した特徴及びステップは、添付の

50

特許請求の範囲内のシステム及び方法のコンポーネントの例として開示している。

【0235】

条項

リモートツリーと同期ツリーとを比較して、前記リモートツリーと前記同期ツリーとの間のサーバの差異を特定することであって、前記リモートツリーは、第1の期間中の、コンテンツ管理システムによって記憶されたコンテンツアイテムのサーバ状態を表し、前記同期ツリーは、前記リモートツリーがローカルツリーと一致していた以前の期間中の同期した状態を表し、前記ローカルツリーは、前記クライアントデバイスによって記憶されたコンテンツアイテムのローカル状態を表す、前記特定することと、前記サーバの差異に基づいて、前記サーバ状態と前記ファイルシステム状態とが同期していないと判定することと、前記サーバの差異に基づいて、前記クライアントデバイス上に記憶された前記コンテンツアイテムのクライアント操作セットを生成することであって、前記クライアント操作セットは、前記サーバ状態と前記ファイルシステム状態とを収束させるように、前記クライアントデバイス上に記憶された前記コンテンツアイテムを操作するよう構成される、前記生成することと、前記クライアント操作セットを提供して、前記クライアントデバイス上で実行させることと、を含む、コンピュータ実装方法。

10

【0236】

命令を含む非一時的コンピュータ可読媒体であって、前記命令は、コンピューティングシステムによって実行された場合に、前記コンピューティングシステムに、ローカルツリーと同期ツリーとの間のクライアントの差異を特定することであって、前記ローカルツリーは、第1の期間中の、前記コンピューティングシステム上に記憶されたコンテンツアイテムのファイルシステム状態を表し、前記同期ツリーは、前記ローカルツリーがリモートツリーと一致していた以前の期間中の同期した状態を表し、前記リモートツリーは、コンテンツ管理システムによって記憶されたコンテンツアイテムのサーバ状態を表す、前記特定することと、前記ローカルツリーと前記同期ツリーとの間のクライアントの差異に基づいて、サーバ状態とファイルシステム状態とが同期していないと判定することと、前記クライアントの差異に基づいて、コンテンツ管理システムによって記憶された前記コンテンツアイテムのサーバ操作セットを生成することであって、前記サーバ操作セットは、前記サーバ状態と前記ファイルシステム状態とを収束させるように、前記コンテンツ管理システムによって記憶された前記コンテンツアイテムを操作するよう構成される、前記生成することと、前記サーバ操作セットを前記コンテンツ管理システムに送信して実行させることと、を行わせる、前記非一時的コンピュータ可読媒体。

20

30

【0237】

プロセッサと、命令を記憶する非一時的コンピュータ可読媒体と、を備えるシステムであって、前記命令は、前記プロセッサによって実行された場合に、前記プロセッサに、同期ツリーと、リモートツリー及びローカルツリーの少なくとも一方との比較を生成することであって、前記リモートツリーは、第1の期間中のサーバ状態を表し、前記ローカルツリーは、前記第1の期間中のファイルシステム状態を表し、前記同期ツリーは、以前の期間中の前記サーバ状態と前記ファイルシステム状態との間の同期した状態を表す、前記生成することと、前記比較に基づいて、同期ツリーと、リモートツリー及びローカルツリーの少なくとも一方との少なくとも1つの差異を特定することと、前記少なくとも1つの差異に基づいて、操作セットを生成することであって、前記操作セットは、前記サーバ状態と前記ファイルシステム状態とを収束させるように構成される、前記生成することと、前記操作セットの実行を管理することと、を行わせる、前記システム。

40

【0238】

コンテンツ管理システム上のユーザアカウントと、前記ユーザアカウントへのアクセスが許可されたクライアントデバイスとの間でコンテンツアイテムを同期させるためのコンピュータ実装方法であって、現在の期間中の前記コンテンツ管理システム上の前記ユーザアカウントに関連付けられたコンテンツアイテムのサーバ状態を表すリモートツリーと、前記現在の期間中の前記クライアントデバイス上の前記ユーザアカウントに関連付けられ

50

たコンテンツアイテムのファイルシステム状態を表すローカルツリーと、前記ローカルツリーと前記リモートツリーとが同一であった以前の期間中の既知の同期した状態を表す同期ツリーと、を含むツリーデータ構造のセットを取得することによって、前記同期ツリーは、前記以前の期間の前記ローカルツリー及び前記リモートツリーと一致する、前記取得することと、前記リモートツリーと前記同期ツリーとを比較することにより、前記コンテンツ管理システム上の前記ユーザアカウントが、前記クライアントデバイスへ同期されていない少なくとも1つの修正を含むと判定することと、前記同期ツリーと前記ローカルツリーとを比較することにより、前記クライアントデバイスが、前記コンテンツ管理システムへ同期されていない少なくとも1つの修正を含むと判定することと、を含む、前記コンピュータ実装方法。

10

**【0239】**

命令を含む非一時的コンピュータ可読媒体であって、前記命令は、コンピューティングシステムによって実行された場合に、前記コンピューティングシステムに、現在の期間中のコンテンツ管理システム上のユーザアカウントに関連付けられたコンテンツアイテムのサーバ状態を表すリモートツリーと、前記現在の期間中のクライアントデバイス上の前記ユーザアカウントに関連付けられたコンテンツアイテムのファイルシステム状態を表すローカルツリーと、前記ローカルツリーと前記リモートツリーとが同一であった以前の期間中の既知の同期した状態を表す同期ツリーと、を含むツリーデータ構造のセットを取得することによって、前記同期ツリーは、前記以前の期間の前記ローカルツリー及び前記リモートツリーと一致する、前記取得することと、前記リモートツリーと前記同期ツリーとを比較することにより、前記コンテンツ管理システム上の前記ユーザアカウントが、前記クライアントデバイスへ同期されていない少なくとも1つの修正を含むと判定することと、を行わせる、前記非一時的コンピュータ可読媒体。

20

**【0240】**

プロセッサと、命令を記憶する非一時的コンピュータ可読媒体と、を備えるシステムであって、前記命令は、前記プロセッサによって実行された場合に、前記プロセッサに、現在の期間中のコンテンツ管理システム上のユーザアカウントに関連付けられたコンテンツアイテムのサーバ状態を表すリモートツリーと、前記現在の期間中のクライアントデバイス上の前記ユーザアカウントに関連付けられたコンテンツアイテムのファイルシステム状態を表すローカルツリーと、前記ローカルツリーと前記リモートツリーとが同一であった以前の期間中の既知の同期した状態を表す同期ツリーと、を含むツリーデータ構造のセットを取得することによって、前記同期ツリーは、前記以前の期間の前記ローカルツリー及び前記リモートツリーと一致する、前記取得することと、前記同期ツリーと前記ローカルツリーとを比較することにより、前記クライアントデバイスが、前記コンテンツ管理システムへ同期されていない少なくとも1つの修正を含むと判定することと、を行わせる、前記システム。

30

**【0241】**

コンテンツ管理システム上のユーザアカウントへの修正を、前記ユーザアカウントへのアクセスが許可されたクライアントデバイスへ同期させるためのコンピュータ実装方法であって、前記コンテンツ管理システム上の前記ユーザアカウントに関連付けられたコンテンツアイテムのサーバ状態を表すリモートツリーデータ構造と、前記コンテンツ管理システムと前記クライアントデバイスとの間の既知の同期した状態を表す同期ツリーデータ構造との差異を計算することと、前記クライアントデバイス上で実行された場合に、前記クライアントデバイス上のファイルシステム状態と前記サーバ状態とを収束させるように、前記クライアントデバイス上に記憶された前記コンテンツアイテムを更新する操作セットを、前記差異に基づいて生成することと、を含む、前記コンピュータ実装方法。

40

**【0242】**

命令を含む非一時的コンピュータ可読媒体であって、前記命令は、コンピューティングシステムによって実行された場合に、前記コンピューティングシステムに、コンテンツ管理システム上のアカウントに関連付けられたコンテンツアイテムのサーバ状態を表すリモ

50

ートツリーデータ構造と、前記コンテンツ管理システムと前記コンピューティングシステムとの間の既知の同期した状態を表す同期ツリーデータ構造との差異を計算することと、前記コンピューティングシステム上で実行された場合に、前記コンピューティングシステム上のファイルシステム状態と前記サーバ状態とを収束させるように、前記クライアントデバイス上に記憶された前記コンテンツアイテムを更新する操作セットを、前記差異に基づいて生成することと、を行わせる、前記非一時的コンピュータ可読媒体。

【0243】

プロセッサと、命令を記憶する非一時的コンピュータ可読媒体と、を備えるシステムであって、前記命令は、前記プロセッサによって実行された場合に、前記プロセッサに、コンテンツ管理システム上のユーザアカウントに関連付けられたコンテンツアイテムのサーバ状態を表すリモートツリーデータ構造と、前記コンテンツ管理システムと前記システムとの間の既知の同期した状態を表す同期ツリーデータ構造との差異を計算することと、前記システム上で実行された場合に、前記システム上のファイルシステム状態と前記サーバ状態とを収束させるように、前記クライアントデバイス上に記憶された前記コンテンツアイテムを更新する操作セットを、前記差異に基づいて生成することと、を行わせる、前記システム。

10

【0244】

クライアントデバイス上のコンテンツアイテムへの修正を、コンテンツ管理システム上のユーザアカウントへ同期させるためのコンピュータ実装方法であって、前記クライアントデバイス上の前記ユーザアカウントに関連付けられたコンテンツアイテムのファイルシステム状態を表すローカルツリーデータ構造と、前記コンテンツ管理システムと前記クライアントデバイスとの間の既知の同期した状態を表す同期ツリーデータ構造との差異を計算することと、実行された場合に、前記コンテンツ管理システム上の前記ユーザアカウントに関連付けられたコンテンツアイテムのサーバ状態と前記ファイルシステム状態とを収束させるように、前記コンテンツ管理システム上に記憶された前記コンテンツアイテムを更新する操作セットを、前記差異に基づいて生成することと、を含む、前記コンピュータ実装方法。

20

【0245】

命令を含む非一時的コンピュータ可読媒体であって、前記命令は、コンピューティングシステムによって実行された場合に、前記コンピューティングシステムに、前記コンピューティングシステム上のユーザアカウントに関連付けられたコンテンツアイテムのファイルシステム状態を表すローカルツリーデータ構造と、コンテンツ管理システムと前記コンピューティングシステムとの間の既知の同期した状態を表す同期ツリーデータ構造との差異を計算することと、実行された場合に、前記コンテンツ管理システム上の前記ユーザアカウントに関連付けられたコンテンツアイテムのサーバ状態と前記ファイルシステム状態とを収束させるように、前記コンテンツ管理システム上に記憶された前記コンテンツアイテムを更新する操作セットを、前記差異に基づいて生成することと、を行わせる、前記非一時的コンピュータ可読媒体。

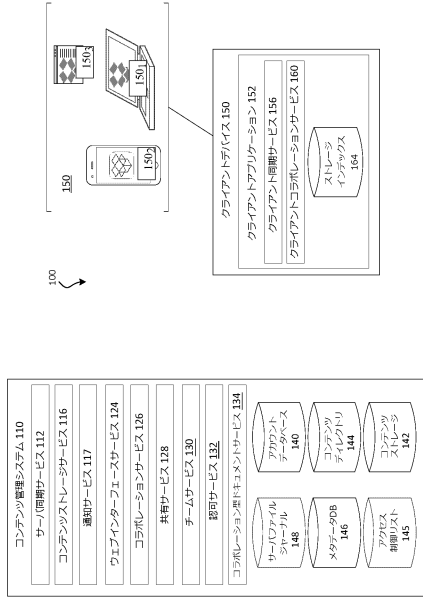
30

【0246】

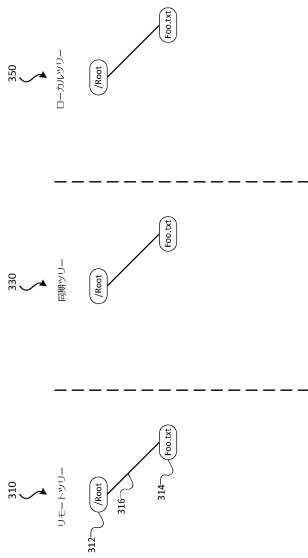
プロセッサと、命令を記憶する非一時的コンピュータ可読媒体と、を備えるシステムであって、前記命令は、前記プロセッサによって実行された場合に、前記プロセッサに、前記システム上のユーザアカウントに関連付けられたコンテンツアイテムのファイルシステム状態を表すローカルツリーデータ構造と、コンテンツ管理システムと前記システムとの間の既知の同期した状態を表す同期ツリーデータ構造との差異を計算することと、実行された場合に、前記コンテンツ管理システム上の前記ユーザアカウントに関連付けられたコンテンツアイテムのサーバ状態と前記ファイルシステム状態とを収束させるように、前記コンテンツ管理システム上に記憶された前記コンテンツアイテムを更新する操作セットを、前記差異に基づいて生成することと、を行わせる、前記システム。

40

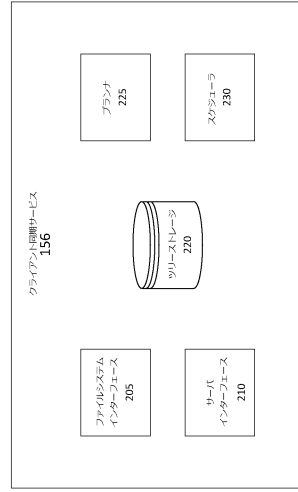
【図面】  
【図 1】



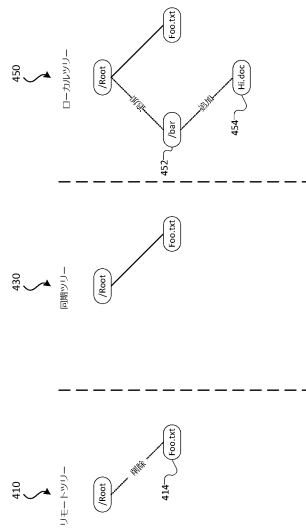
【図 3】



【図 2】



【図 4】



10

20

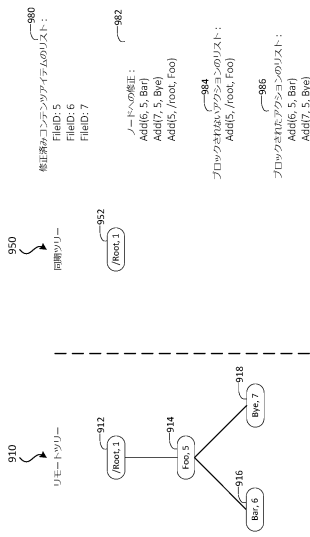
30

40

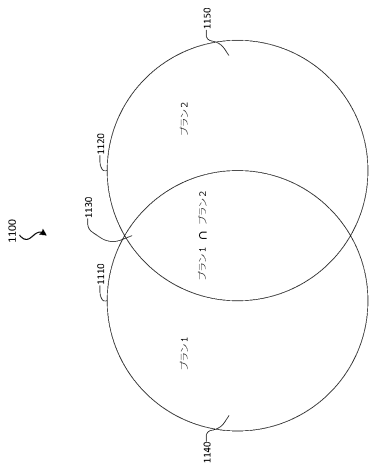
50



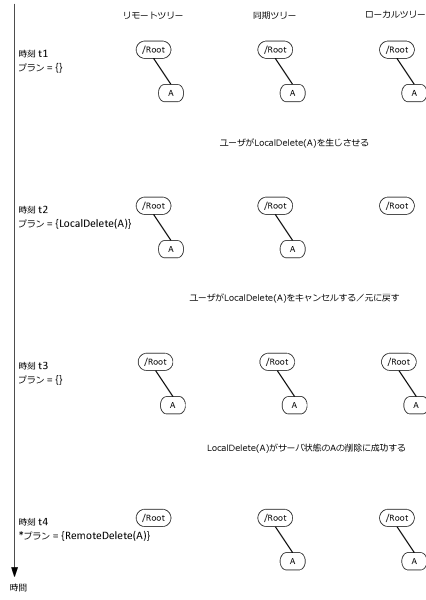
【 図 9 】



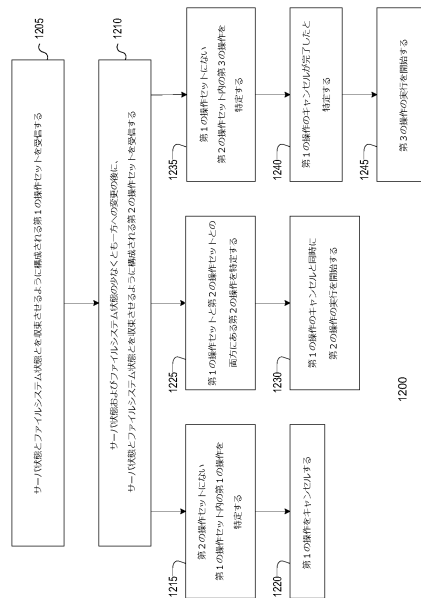
【 図 11 】



【 図 10 】

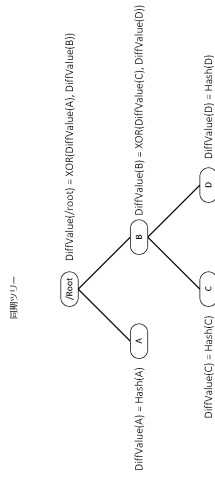


【 図 12 】

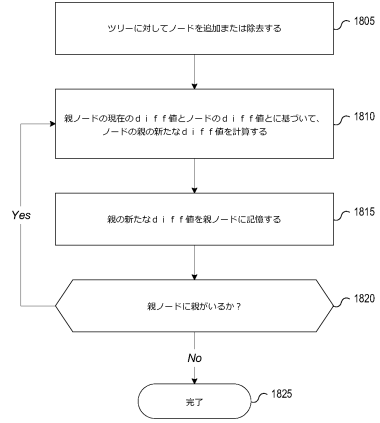




【 17 】



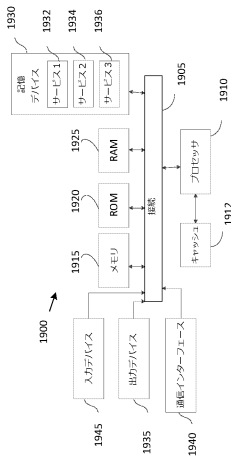
【 18 】



10

20

【 19 】



30

40

50

## フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 15/868,511

(32)優先日 平成30年1月11日(2018.1.11)

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 62/611,473

(32)優先日 平成29年12月28日(2017.12.28)

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 15/868,505

(32)優先日 平成30年1月11日(2018.1.11)

(33)優先権主張国・地域又は機関

米国(US)

アメリカ合衆国, カリフォルニア州 9 4 1 0 7, サンフランシスコ, ブラナン ストリート 3 3  
3, ドロップボックス, インコーポレイテッド 内

(72)発明者 レイ, ジョン

アメリカ合衆国, カリフォルニア州 9 4 1 0 7, サンフランシスコ, ブラナン ストリート 3 3  
3, ドロップボックス, インコーポレイテッド 内

(72)発明者 イン, ロバート

アメリカ合衆国, カリフォルニア州 9 4 1 0 7, サンフランシスコ, ブラナン ストリート 3 3  
3, ドロップボックス, インコーポレイテッド 内

(72)発明者 コオラパティ, ニブン

アメリカ合衆国, カリフォルニア州 9 4 1 0 7, サンフランシスコ, ブラナン ストリート 3 3  
3, ドロップボックス, インコーポレイテッド 内

(72)発明者 グプタ, ガウタム

アメリカ合衆国, カリフォルニア州 9 4 1 0 7, サンフランシスコ, ブラナン ストリート 3 3  
3, ドロップボックス, インコーポレイテッド 内

(72)発明者 ソン, ジョフリー

アメリカ合衆国, カリフォルニア州 9 4 1 0 7, サンフランシスコ, ブラナン ストリート 3 3  
3, ドロップボックス, インコーポレイテッド 内

(72)発明者 ジュブ, エルマー チャールズ

アメリカ合衆国, カリフォルニア州 9 4 1 0 7, サンフランシスコ, ブラナン ストリート 3 3  
3, ドロップボックス, インコーポレイテッド 内

審査官 原 秀人

(56)参考文献 国際公開第2016/160994(WO, A1)

特開平11-045201(JP, A)

特開2005-063124(JP, A)

特表2017-529625(JP, A)

アンドリュー・S・タネンバウム 外, 分散システム, 初版, 日本, 株式会社ピアソン・エ  
デュケーション, 2003年10月20日, pp. 621-628

(58)調査した分野 (Int.Cl., DB名)

G 0 6 F 1 6 / 0 0 - 1 6 / 9 5 8