

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2012-516503

(P2012-516503A)

(43) 公表日 平成24年7月19日(2012.7.19)

(51) Int.Cl. F I テーマコード (参考)
G 0 6 F 12/00 (2006.01)
 G 0 6 F 12/00 5 3 3 J
 G 0 6 F 12/00 5 4 5 Z

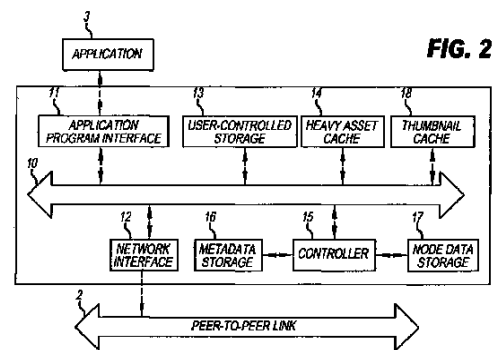
審査請求 未請求 予備審査請求 未請求 (全 29 頁)

| | |
|---|---|
| (21) 出願番号 特願2011-547954 (P2011-547954) (86) (22) 出願日 平成22年1月15日 (2010.1.15) (85) 翻訳文提出日 平成23年7月25日 (2011.7.25) (86) 国際出願番号 PCT/US2010/000101 (87) 国際公開番号 W02010/087940 (87) 国際公開日 平成22年8月5日 (2010.8.5) (31) 優先権主張番号 12/362, 762 (32) 優先日 平成21年1月30日 (2009.1.30) (33) 優先権主張国 米国 (US) | (71) 出願人 590000846 イーストマン コダック カンパニー アメリカ合衆国 ニューヨーク州 ロチェスター スター ステート ストリート 343 (74) 代理人 110001210 特許業務法人 Y K I 国際特許事務所 (72) 発明者 ブロムシュテット リンダ シー アメリカ合衆国 ニューヨーク ロチェスター スター ステイト ストリート 343 (72) 発明者 シュワルツ マイケル エス アメリカ合衆国 ニューヨーク ロチェスター スター ステイト ストリート 343 (72) 発明者 パリー チャールズ イー アメリカ合衆国 ニューヨーク ロチェスター スター ステイト ストリート 343 最終頁に続く |
|---|---|

(54) 【発明の名称】 分散された資産とメタデータを管理するシステム

(57) 【要約】

ネットワーク内に分散されたデジタル資産および関係メタデータを管理するためのシステムは、相互接続された複数のノードを含み、相互接続されたノードの各々は、ユーザ制御によるストアとキャッシングサブシステムを介して、分散された資産および関係メタデータへのアクセス可能性を提供し、資産と関係メタデータの管理が所定の基準に基づいて行われるようになっている。各ノードは、キャッシングサブシステム、メタデータストア、コントローラ、ユーザ制御による資産ストレージコンポーネント、および各ノード間の通信のためのネットワークリングサブシステムと、を有する。相互接続されたノードは、クライアントアプリケーションのプラットフォーム運用をサポートする。



【特許請求の範囲】**【請求項 1】**

あるネットワークにおける分散されたデジタル資産と関係メタデータを管理するシステムであって、

相互接続された複数のノードを有し、前記相互接続されたノードの各々は、ユーザ制御によるストアとキャッシングサブシステムを通じて、前記分散された資産へのアクセス可能性を提供し、前記資産と関係メタデータの管理が所定の基準に基づいて行われるようになっており、前記相互接続された複数のノードの各ノードは、キャッシングサブシステムと、メタデータストアと、コントローラと、ユーザ制御による資産コンポーネントと、前記各ノード間の通信のためのネットワーキングサブシステムと、を含み、前記相互接続された複数のノードは、クライアントアプリケーションのプラットフォーム運用をサポートすることを特徴とするシステム。

10

【請求項 2】

請求項 1 に記載のシステムであって、

前記分散されたデジタル資産の管理のための前記所定の基準は、ノードの可用性、ノードの能力、資産の人気と需要、資産安全要求、および資産可用性優先性要求を含むことを特徴とするシステム。

【請求項 3】

請求項 1 に記載のシステムであって、

前記相互接続された複数のノードは、個別に構成されることを特徴とするシステム。

20

【請求項 4】

請求項 1 に記載のシステムであって、

前記コントローラは、前記クライアントアプリケーションのためのアプリケーションプログラムインタフェース (API) を提供して、前記分散されたデジタル資産と関係メタデータにアクセスし、これを管理することを特徴とするシステム。

【請求項 5】

請求項 1 に記載のシステムであって、

1 つまたは複数の相互接続されたノードの間の同期が、少なくとも 1 つのノードへの変更の結果として開始されることを特徴とするシステム。

【請求項 6】

請求項 2 に記載のシステムであって、

前記関係メタデータの管理はさらに、1 つまたは複数のノードを同期させ、前記関係メタデータが、前記相互接続されたノードの各々に関連する挙動に適用される前記所定の基準に従って、個別に調整されるようにすることを含むことを特徴とするシステム。

30

【請求項 7】

請求項 2 に記載のシステムであって、

前記資産の管理はさらに、1 つまたは複数のノードを同期させ、前記資産が、前記相互接続されたノードの各々に関連する挙動に適用される所定の基準に従って、個別に調整されるようにすることを含むことを特徴とするシステム。

【請求項 8】

請求項 5 に記載のシステムであって、

実質的な数のノードペアが相互に同期されることを特徴とするシステム。

40

【請求項 9】

請求項 8 に記載のシステムであって、

所定の期間だけオフラインであったノードは、前記ノードがオフラインに戻った時に、新しいメタデータでそれを更新することを特徴とするシステム。

【請求項 10】

複数のユーザ間のデータ資産へのアクセスを提供する方法であって、

少なくとも第一のユーザと第二のユーザの間で前記データ資産を共有するためのネットワークを提供するステップと、

50

前記データ資産の共有への招待を有効にするステップであって、前記招待が前記第一のユーザから前記第二のユーザに送信されるステップと、

前記第二のユーザによる前記招待の受入を有効にするステップと、

共有されるデータ資産に対応する共有メタデータが作成されるようにするステップと、

前記第一と第二のユーザ間の前記データ資産の共有を有効にするために、前記共有メタデータのコピーを送信するステップと、

前記共有メタデータが、少なくとも、最近のメタデータ変更が反映されるように変更されている場合、前記共有メタデータを同期させるステップと、
を含むことを特徴とする方法。

【請求項 11】

10

請求項 10 に記載の方法であって、

前記第二のユーザによって、データ資産のレプリカを要求するステップと、

要求されたデータ資産のレプリカを有する少なくとも 1 つのノードが利用可能か否かを判断するステップと、

データ資産の新しいコピーを前記第二のユーザに伝送するステップと、

指定されたノードに位置する追加のデータ資産の存在を、前記データ資産の新しいコピーを反映させるように記録するステップと、
をさらに含むことを特徴とする方法。

【請求項 12】

20

請求項 11 に記載の方法であって、

前記同期させるステップは、すべてのレプリカの位置を含むことを特徴とする方法。

【請求項 13】

請求項 11 に記載の方法であって、

前記データ資産の新しいコピーは、キャッシュ内に設置されることを特徴とする方法。

【請求項 14】

複製のデータ資産を判断する方法であって、

モニタされた切断ノードにおける追加資産を判断するステップと、

前記追加資産が既存の資産の複製であるか否かを評価するステップと、

前記モニタされた切断ノードにおいて見つめられた複製資産を追加レプリカとして分類するステップであって、前記切断ノードに複製資産が見つからなかった場合は、前記切断ノードを再接続するステップと、

30

わかっているその他すべてのユーザノードに連絡して、前記複製資産がこれらのノードに存在するか否かを判断するステップと、

前記わかっているその他のユーザノードにおいて見つめられた複製資産を追加レプリカとして分類するステップと、

前記わかっているユーザノードにおいて前記追加資産が存在しない場合、前記追加資産を新規資産に分類するステップと、
を含むことを特徴とする方法。

【請求項 15】

40

請求項 14 に記載の方法であって、

前記わかっているユーザノードのいくつかがオフラインであり、前記追加資産が暫定新規資産として記録されることを特徴とする方法。

【請求項 16】

請求項 15 に記載の方法であって、

わかっているユーザノードの各々がオンラインに戻ったとき、前記オンラインに戻った、わかっているユーザノード上の前記複製資産の存在を判断するステップをさらに含むことを特徴とする方法。

【請求項 17】

請求項 16 に記載の方法であって、

検出された複製資産をレプリカとして分類するステップをさらに含むことを特徴とする

50

方法。

【請求項 18】

複数のノードにわたって、共有デジタル資産に関連するメタデータを管理するシステムであって、

- a) 共有資産に関連する前記メタデータと前記共有デジタル資産へのアクセスを制御するノードコントローラと、
 - b) ローカルノードに関するメタデータを保有し、ユーザの集合における資産に対応するメタデータストアと、
 - c) 前記ノードコントローラの処理を制御するワークフローマネージャと、
- を備えることを特徴とするシステム。

10

【請求項 19】

請求項 18 に記載のシステムであって、

共有資産に関する前記メタデータには、ユーザの分散されたデジタル集合の中の資産に関する情報が含まれることを特徴とするシステム。

【請求項 20】

請求項 18 に記載のシステムであって、

いずれか 1 つのノードで生成されるメタデータは、ノード同期プロセスを通じて他のノードに通信され、前記メタデータには、画像とオーディオクリップがサムネイルとパスワードとして含まれることを特徴とするシステム。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数のデバイス、サーバおよびシステムの間でデータを管理するためのアーキテクチャ、サービスおよび方法に関する。特に、本発明は、いずれかのシステムノードまたはデバイスからのメタデータを含むあるユーザのデータ関連資産について、論理的に一元化した統合ビューを提供することに関する。

【背景技術】

【0002】

データ関連資産には、個人で楽しむために制作されたり、パーソナルコンピュータ（PC）のストレージにダウンロードされたりする画像、ビデオおよび音楽ファイルがある。一般に、今日のデータ関連資産は、たとえ当初はそうでなくても、デジタルフォーマットであることが多い（すなわち、「デジタル資産」）。通常、これらのデジタル資産には、見る、聴く、または再生するために必要な時だけアクセスされる。さまざまなデバイスやインターネットサービスがこれらの資産を提供、利用し、たとえば携帯情報端末（PDA）、デジタルカメラ、パーソナルコンピュータ（PC）、メディアサーバ、端末およびウェブサイト等がある。これらのデバイスやサービスプロバイダに保存された資産の集合は一般に、緩く結合されているか、まったく結合されず、現行同期プロセスは通常、2 つのデバイス、たとえばメディアプレイヤーと PC との間で行われる。

30

【0003】

多くの既存のソリューションでは、分散されたデジタル資産の集合の統合ビューが、すべてのノードで各資産を複製することによって（すなわち、その資産に関連付けられたすべてのデータファイルを複製する）ことによって提供される。Syncplify、Sharpcast の Sugar Sync および Microsoft の Live Mesh は、この複製方式を用いた、現時点での市販品の例である。このようなシステムの 1 つの欠点は、ストレージの非効率性である。新しい資産が他のノードに追加されるたびに、すべてのノードにおいて、保存された資産がそのノードに必要なか否かに関係なく、追加のストレージ容量が必要となる。

40

【0004】

すべてのノードで新しい資産を複製する上記のシステムの別の欠点は、ユーザが資産のアップロードを待っている間、ユーザの使用体験上に与えられる大きな影響である。多く

50

の一般的なワークフローでは、関心対象の資産(asset interest)をアップロードする(たとえば、共有または印刷する)必要があるため、ユーザの使用体験は、アップロード作業の実行中、実質的に遅延される。

【0005】

従来のシステムのさらに別の欠点は、過去に共有された資産へのアクセスを制限する能力に関する。共有された資産が他のユーザと共有されると、その資産は、受領者のノードでも複製される。既存のソリューションの多くにおいて、その資産の新しい、同一のレプリカが受領者のノードまたはコンピュータ上に存在するとき、当初の共有者はそのレプリカを削除することができない。

【0006】

他の従来のソリューションは、分散されたデジタル資産の集合について、資産を保存している各ノードが、要求された資産のあるバージョンを、要求しているノードに放送できるようにすることによってその統合ビューを提供する。その一例がOrb Systemsである。各アクセスポイントノードにインストールされたクライアントアプリケーションは、要求された資産を、要求された要件(たとえば、解像度基準)にしたがって、他のノード上に流す。このタイプのシステムの欠点は、アクセスポイントがシステムに接続されていないときに(すなわち、オフライン状態では)、アクセスできないことである。さらに、オフラインノードは事実上、動作せず、ローカルに保存された資産に関する数値しか提供しない。さらに、他の制約は、ある資産を保存するノードがオフラインであると、他のシステムの中のどのノードも、保存された資産の存在に気付かないことである。

【0007】

既存のシステムの別の欠点は、集合に新たに導入された資産がすでに存在するかもしれないことを認識する能力が限られていることであり、したがって、システムは通常、集合の中に不要な複製を生成し、これが動作のための追加の資源(たとえば、保存スペースや処理時間)を必要とすることによって、システムに影響を与える。これはまた、そのシステムに同じ資産のコピーがいくつも含まれていることに気付かないユーザにも影響を与える。

【0008】

最後に、分散されたデジタル資産の集合の従来の統合ビューの別の欠点は、所有者が自分の資産をローカルで(たとえば、ピクチャを整理するWindows(登録商標)フォルダシステム)整理しているときに生成される情報が、破壊されるか、無視されるかもしれないという点である。

【先行技術文献】

【特許文献】

【0009】

【特許文献1】国際公開第2007/024380号

【非特許文献】

【0010】

【非特許文献1】アンドロウトセリス・セオトキスら(ANDROUTSELLIS - THE OTOKIS S ET AL)、「ピアツーピアコンテンツ分配技術の概説(survey of peer-to-peer distribution technologies)」、エーシーエムコンピューティングサーベイ(ACM COMPUTING SURVEYS), エーシーエム(ACM)、ニューヨーク(NEWYORK)、(米国)、Vol. 36、No. 4、2004年12月1日、p. 335 - 371

【非特許文献2】マーチン・ブイラ(MARTINS V ET AL)、「P2Pシステムにおけるデータ複製の概説(Survey of data replication in P2P systems)」、[online]、2006年12月1日、p. 1 - 45、[2008年10月17日検索]、インターネット<URL: http://hal.inria.fr/docs/00/12/22/82/PDF/Survey_of_data_replication_in_P2P_systems.ODF

10

20

30

40

50

>

【非特許文献3】デブルム サッコル ディーら (DE BRUM SACCOL D ET AL)、 「ピアツーピア環境におけるレプリカとバージョンの検出、管理及び検索 (Detecting, managing and querying replicas and versions in a peer-to-peer environment)」、第77回クラスタコンピューティングとグリッドのIEEE国際シンポジウム (CCGRID 2007) IEEE ビスカタウェイ (PISCATAWAY)、ニュージャージー (NJ)、(米国)、2007年、p. 833 - 838

【発明の概要】

【発明が解決しようとする課題】

10

【0011】

したがって、当業界では依然として、今日存在する、従来の分散型デジタル資産集合システムに関わる欠陥や制約を克服することに対するニーズがある。

【課題を解決するための手段】

【0012】

上記のニーズは、あるネットワークにおいて分散されたデジタル資産と関係メタデータを管理するシステムを開示する本発明によって満たされ、このシステムは、いくつかの相互接続されたノードを含み、相互接続されたノードの各々は、ユーザ制御によるストアとキャッシングサブシステムを介して、分散されたデジタル資産へのアクセス可能性を提供し、資産と関係メタデータの管理が所定の基準に基づいて行われるようになっている。相互接続された各ノードは、キャッシングサブシステム、メタデータストア、コントローラ、ユーザ制御による資産ストレージコンポーネント、および各ノード間の通信のためのネットワークサブシステムと、を有する。相互接続されたノードは、クライアントアプリケーションのプラットフォーム運用をサポートする。

20

【0013】

本発明はさらに、モニタされた切断されたノードにおいて追加された資産を判断し、追加された資産が既存の資産の複製であるか否かを判断することによって、複製データ資産を判断する方法を提供する。追加された資産が複製であると直ちに評価された場合、この資産は、追加されたレプリカとして分類される。切断されたノードで何の複製資産も見つからない場合、ノードが再接続されたときに、他のすべてのわかっているユーザノードに連絡して、それらのノードに複製資産が存在するか否かを判断する。もし存在すれば、その複製資産は追加されたレプリカとして分類される。わかっているユーザノードに追加された資産が存在しない場合、その追加された資産は新規資産として分類される。

30

【0014】

本発明のさらに1つの他の態様では、このシステムの1人のユーザは、ある資産を第二のユーザと、第二のユーザがその資産を見て、その資産とそのメタデータの変更を見ることができ、また、資産の所有者からそれを行うための許可を受ければ、その資産とそのメタデータに変更を加えることができるような方法で共有できる。

【0015】

したがって、複数のユーザ間でデータ資産のアクセスを提供する方法が開示され、この方法は、少なくとも第一のユーザと第二のユーザとの間でデータ資産を共有するためのネットワークを提供するステップを含む。第二のステップは、データ資産共有への招待を有効にする。招待が第一のユーザから第二のユーザに送信される。第三のステップは、第二のユーザによる招待の受入を有効にし、したがって、その後、共有されるデータ資産に対応する共有メタデータが生成される。別のステップは、メタデータのコピーを第二のユーザに送信して、第一と第二のユーザの間のデータ資産の共有を有効にする。最後のステップでは、共有メタデータが変更されている場合に、共有メタデータを同期させ、少なくとも、最近のメタデータの変更が反映されるようにする。

40

【図面の簡単な説明】

【0016】

50

【図 1】本発明による完全分散型システムの一例を、ピアツーピアネットワークによって相互接続されたノード群として示す図である。

【図 2】1つのノードのコンポーネントを示す図である。

【図 3】バージョンベクタの成分を示す図である。

【図 3 a】ユーザ間である資産を共有し、その資産の変更を同期させるステップの例を示す図である。

【図 4】切断されたノードによる複製検出のステップの例を示す図である。

【図 5】同じメタデータを保存するノードを同期させるステップの例を示す図である。

【図 6】メタデータレベルの異なるノードを（能力の高いものから能力の低いものへ）同期させるステップの例を示す図である。

10

【図 7】メタデータレベルの異なるノードを（能力の高いものから能力の低いものへ）同期させるステップの例を示す図である。

【図 8】同じメタデータレベルのノードを同期させるステップの例を示す（部分同期を記入）図である。

【発明を実施するための形態】

【0017】

本明細書では、本発明による分散デジタル資産管理システムについて述べる。分散デジタル資産管理システムは時々、連合資産管理システム（フェデレイテッド資産マネジメントシステム(federated asset management system)）と呼ばれる。分散デジタル資産管理システムは、本発明の文脈において、異なるソースからのデータを、システムのユーザにとって、すべてのデータと一緒に保存されているかのように見える方法で結合する。これらのソースには、たとえば、パーソナルコンピュータ、サーバ、携帯電話、ゲーム機等がある。データには、デジタルデータ、オーディオデータ、画像処理データ等がある。したがって、本発明の文脈において、「資産」は、当初からデジタル化された、または後にデジタルフォーマットに変換される画像、ビデオ、音楽、その他のオーディオデータおよびファイルに関する。

20

【0018】

本発明のいくつかの態様の1つは、所有され、共有されるすべてのメディア資産の統合ビュー（1つのノードまたは多数のノードを通じたもの）とアクセスを提供する。あるユーザによって所有または共有されるデジタル/メディア資産の全部が統合され、あるユーザの仮想集合と呼ばれる。本発明は、そのユーザのデジタル資産およびこの仮想集合に関連付けられるメタデータの全部または選択された量を管理するための分散システムについて述べ、これにはそのユーザが使用したい、またはそのユーザの資産が存在するかもしれない、かつその他のノードと通信する各種のノードの上で実行されるソフトウェアサービス群が含まれる。

30

【0019】

本発明による、一例としての分散デジタル資産管理システム（以下、「システム」とも称する）には、図 1 に示すように、ノード群（1、4、5 および 6）が含まれる。ノード 1、4、5 および 6 には、コンピュータ上で実行されるソフトウェアサービスまたはサービス群が含まれる。このコンピュータは、あるユーザに帰属するワークステーションであってもよいが、ノードは携帯情報端末（PDA）またはその他のネットワーク対応デバイス、たとえば携帯電話またはゲーム機等であってもよい。ノード 1、4、5 および 6 はネットワーク上で、ピアツーピアリンク 2 を使って相互に通信する。このネットワークは一般にインターネットであるが、Bluetooth（登録商標）等、他のプロトコルに基づくアドホックサブネットワークを含んでいてもよい。いくつかのノードは、アプリケーション 3 と通信する。典型的なアプリケーションは、ユーザ（「メンバ」とも称する）に閲覧させ、共有させ、またさもなければ、他のノードにある自分の仮想集合の中の資産を含む自分の仮想集合およびその集合に属する資産とメタデータのすべてを操作させる。あるコンピュータ上の各ノードは、そのコンピュータを使用するすべてのユーザをサポートする。この分散システムの構成は、好ましくは中央ノード 5 を含む。中央ノード 5 は、ネ

40

50

ットワークサーバ群の上にある。中央ノード5は、分散システム上の異なるユーザとの最初のコンタクトを支援し、ウェブアプリケーションをサポートし、写真仕上げ等の外部サービスをサポートする。中央ノード5はまた、可用性が保証される必要のある資産のためのキャッシングストレージを提供することができる。この分散システムのほとんどのユーザについて、相互作用と通信は主として中央以外の（縁辺）ノードと、またそれらの間で行われる。

【0020】

ノードの一例の内部構造を図2に示す。典型的なノードは、相互にメッセージ10を伝えることによって通信する、多数のコンポーネントを含む。あるシステムでは、メッセージはMicrosoftのMSMQまたは同様のソフトウェアによって実装される同期キューを伝えるが、これらのコンポーネントはまた、メッセージを直接伝えることもできる。Application Program Interface (API)であるコンポーネント11は、ユーザアプリケーションに、そのユーザの仮想集合およびそれに関する対応の情報へのアクセスを与える。ネットワークインタフェース12により、各ノードは、メッセージの送受によって他のすべてのノードと通信できる。ユーザ制御によるメモリ/ストレージであるコンポーネント13は、ユーザ制御による資産を管理する。一般に、ユーザ制御によるメモリ/ストレージであるコンポーネント13によって管理される資産には、そのノードが位置するコンピュータにあるファイルが含まれる。しかしながら、ユーザ制御によるメモリ/ストレージであるコンポーネント13は、Flickr（登録商標）等の遠隔ネットワークサイトにおける資産を管理することも可能である。大量資産キャッシュ（heavy asset cache）14には、このノードに一時的に保存されている資産およびその他のファイルが含まれる。コントローラ15は、ノードによって処理されるタスクを実行するワークフローを起動する。これらのワークフローは、ノードデータベース17に保存されているワークフローの説明とポリシーによって制御される。ノードは、これらのワークフローの説明とポリシーを変更することによって、個別に構成することができる。たとえば、PDAとワークステーションのワークフローは大きく異なるであろう。コントローラによって使用されるワークフローとポリシーには、大量資産キャッシュの管理に適用されるポリシーが含まれる。

【0021】

メタデータストア16は、ユーザの集合の中の資産に対応するメタデータを含む。一般に、あるユーザのためのメタデータストア16には、そのユーザの集合の中のすべての資産のためのメタデータが含まれるであろう。資産に対応するサムネイル（たとえば、画像やオーディオクリップ）もまた、この分散システムではメタデータと考えられる。サムネイルは、サムネイルキャッシュ18の中に保存される。本発明の他の実施形態では、サムネイルをメタデータストア16の中に保存することができる。あるノードに保存されるメタデータの量は十分であり、コンピュータがネットワークから分離されても、ユーザは集合全体を閲覧し、これを使って何らかの活動を行うことができる。各ノードにどの資産メタデータが保存されるかに関する正確な仕様は、ノードデータストア17に保存された基準によって決定される。

【0022】

ネットワークインタフェース12は、ノード間のピアツーピア接続の管理を担当する。これは、ノード間でメッセージとファイルを伝送し、どのノードがオンラインかを発見する。ネットワークインタフェース12は、2つの部分から構成される。1つの部分は、Network Talker / Listenerと呼ばれ、残りのノードからメッセージを受け取る。これらのメッセージは、そのシステム特定のものであり、伝送とは無関係のフォーマットである。これらのメッセージは、特定のノード上の特定のユーザに宛てられる。ノードの識別は、特定のネットワークアドレスを参照しない方法で特定される。第二の部分は、Talker / Listenerがメッセージの送受信に使用するインタフェースである。Network Talker / Listenerが使用する伝送インタフェースは、そのシステムに利用できる各種の伝送に関するすべての情報をカプセル化する

。ノード間の通信のための最も好ましいメカニズムは、ルータとファイアウォールによって分離されるホームシステム間でデータを伝送する。理想的には、ネットワークインタフェース12は、ユーザがそのルータまたはファイアウォールに穴を開けることを必要とせず、この伝送を行うべきである。そのために、このシステムの1つの実施形態において最も一般的に使用されるノード間でデータとメタデータを伝送するのに使用される方法は、Googleにより開発されたlibjingleと呼ばれるソフトウェアパッケージを使う。このパッケージは、コンピュータ間でメッセージを送信するチャットメカニズムを、民生用ネットワークルータを通じてデータを受け渡すことのできるピアツーピアファイル伝送メカニズムと組み合わせる。コンピュータ間でピアツーピア接続を確立するためのその他のよく知られた方法もまた、このシステムで使用できる。これに加えて、1つのホームネットワーク内に位置するノードは、単純なTCP/IPベースのプロトコルを使って通信でき、外部チャットサーバの使用を必要としない。

10

【0023】

ユーザは、アプリケーションを通じてこのシステムにアクセスし、そのアプリケーションはグラフィカルユーザインタフェース(すなわち、GUI)を提供し、それによってユーザは集合を見る/編集することができる。これらのアプリケーションは、ローカルの資産管理ノードによって提供されるサービスを利用して、あるユーザに属する分散された集合にアクセスし、アプリケーションはその資産またはそのメタデータの位置の詳細を知っていなくてよい。アプリケーションは、分散され、システムのアプリケーションの一部としてロードされたライブラリを使用する。このライブラリは、ローカルノード上の、そのシステムのApplication Program Interface (API)であるコンポーネント11と通信する。ライブラリは、標準的なプロセス間通信方式を用いてAPIであるコンポーネント11にメッセージを送信することによって、通信する。システムAPIであるコンポーネント11は、アプリケーションからのメッセージに応答して、システムノードの他のコンポーネントと通信する。

20

【0024】

アプリケーションは、システムAPIを使って、システムメタデータ(たとえば、メンバーの情報、集合の情報および資産の情報)の読取りと書込みを行うことができる。アプリケーションはまた、状態およびエラーの両方の情報のための通知と、アプリケーション外で集合が変更されたときの通知を予約することができる。アプリケーションは、システムサービスを使って、大量資産キャッシュ14内の大量資産(すなわち、フル解像度資産ファイル)にアクセスする。大量資産へのアクセスは、その資産のためのローカルパスを要求するか、その資産へのストリームを開くことによって実行できる。システムサービスは、大量資産の実際のストレージの位置に関係なく、使用可能なローカルパスまたはストリームを提供する。

30

【0025】

中央ノード5では、APIがネットワーク環境にとって適当である必要がある。インターネットAPIのために最も一般的に使用されるプロトコルは、Simple Object Access Protocol (SOAP)とRepresentational State Transfer (REST)プロトコルであり、その各々にHTTPパケット内のAPIパラメータと応答の伝送が関わる。中央ノード5の上にあるAPIプロセッサは、RESTコマンドに応答する。このAPIを通じて大量資産を要求すると、その資産のコピーが中央の大量資産キャッシュ14に移動し、中央の大量資産キャッシュ14からその資産の読出しに使用できるプロキシURLを戻す。

40

【0026】

システムメタデータには、本発明によれば、集合内の資産に関する情報のほか、ユーザ情報(たとえば、パスワード、ファイル名、ビデオプレビュー、資産作成日、ユーザ提供のタグ)と、システム動作の状態/ステータス(たとえば、ファイルコピー、ファイル共有)が含まれる。資産情報には、ファイルから抽出されるExifタグ等の情報、ユーザにより追加されるキャプション等の情報および、イメージインテリジェンス(image intel

50

ligence)の作業を用いて資産のコンテンツから抽出されるデータを含む。ある資産に関するメタデータにはまた、その資産に関するサムネイル、たとえば、その資産から抽出され、ユーザがその資産を容易に視覚的に識別できるようにする小さな画像も含む。ある資産のメタデータには、その資産のわかっているすべてのコピーへのリファレンスおよび、その資産から作成されたその他すべての資産へのリファレンスも含まれる。システムが、ある資産がどの資産から導出されたものかを知っていれば、資産メタデータには、その資産が導出された元の資産へのリファレンスも含まれる。ある資産のアクセス制御情報もまた、その資産のメタデータの一部である。システムメタデータは、特性を有するサブジェクトとしてオーガナイズされ、そのサブジェクトと特性はどちらも、それ自体がメタデータの一部である。

10

【0027】

1つのノードで発生されるメタデータは、ノード同期プロセスを通じて他のノードに伝えられる。本発明のある一面において、あるシステムのノードは、そのノードを使用するユーザの集合に関連付けられるメタデータひとつひとつのすべてを有しているわけではない。しかしながら、各ノードは、それが「アクセスクリティカル(access-critical)」と考えるメタデータを保存する。アクセスクリティカルなメタデータとは、そのノードがオフラインのときに、ローカルアプリケーションが機能するのに必要なメタデータである。ノードにより、何が「アクセスクリティカルなメタデータ」かの考え方は異なり、たとえば、携帯電話は資産IDのリストしか保存しないかもしれない。PCは、資産ID、日付、キャプション、説明等を含む、ずっと大きなメタデータ群を有するかもしれない。ノード同期プロセスの中で、各ノードは、それが「アクセスクリティカル」であるとする特定のメタデータ群を他のノードに伝える。また、そのローカルノードにのみ関係するメタデータを保持する別のメタデータストア16があってもよい。別のメタデータストア16は、たとえば、物事の局所状態に関する項目を保持する。

20

【0028】

集合のメタデータには、リファレンスのほか数値もまた含めることができる。たとえば、資産メタデータの1つは、その資産のプロパティである。ある資産の、異なるノード上に位置する各種のコピーもまた、ある資産のプロパティであり、これらのコピーのリファレンスがメタデータストアに保存される。システムは、これらのリファレンスをシステムのUniversal Resource Identifier (URI)として保存する。システム資源の特定に加え、システムURIは、その基本的アクセスメカニズム(たとえば、そのネットワークの「位置」)を説明することによって、その資源の位置を特定する手段を提供するため、システムのUniversal Resource Locator (URL)でもある。URLをこのように使用できるのは、それが、各ノードを一意的に特定し、ネットワークインタフェースコンポーネントによって解釈可能なGlobally Unique Identifier (GUID)であるシステムネットワークノードIDを含んでいるからである。あるノードに、それがローカルで保存していないメタデータまたは資産が求められると、このノードはシステムURLを使って、必要なデータを確かに有するのはどのノードかを見つけ出し、これらのノードに対し、所望のデータを求めるメッセージを送信する。

30

40

【0029】

メタデータストア16には、さまざまなデータベース技術を含めることができる。集合メタデータの多くが、Resource Description Framework (RDF)を実装するデータベースの中に保存される。RDFは、セマンティックウェブが、アプリケーション、企業および社会の境界を越えたデータの共有と再使用を可能にする共通のフレームワークを提供するための手段である。これに加えて、過去および進行中のセマンティックウェブの作業の多くは、RDFを使って表されるデータに関する「推論」を含む。メタデータの共有と再使用およびメタデータに関する推論はシステム、ひいてはオープンインタフェースとイメージインテリジェンスの使用にとって重要である。従来のリレーショナルデータベースと比較したRDFのその他の利点は、新しい種類の情報を

50

設計の中に容易に追加できることである。しかしながら、R D Fに伴う、および、E x i fファイル等の特定の種類のデータに関してはコストも発生するため、リレーショナルデータベースのほうがよりよい選択肢である。したがって、本発明の1つの実施形態は、複合的なデータベースを使用する。各種類のメタデータは、ストレージの種類に関連付けられる。メタデータストア16は、特定の種類のメタデータにアクセスする必要がある場合、その種類のメタデータに関するそのストレージモードを見て、どのデータベースに行くべきかを決定する。

【0030】

あるノード上のノードコントローラ15は、システムメタデータと資産へのアクセスを制御する。その動作は、ノードデータストア17に保存されているそのノードの能力に関する情報によって制御される。ノードコントローラ15は、ワークフローマネージャを使ってその処理の多くを制御する。他のシステムコンポーネントからのメッセージの一連の送受信を行わなければ、メッセージの処理が完了したとは考えられないワークフローが数多くある。ワークフローマネージャは、これらのワークフローの全部の管理を担当する。ワークフローマネージャはまた、メッセージ間のワークフローの状態を保持するために、その状態を説明するパラメータをノードデータストア17に保存することも担当する。

【0031】

ワークフローマネージャは、ポリシーにしたがって動作する。これは、それが動作しているノードの種類に関する知識を持ち、あるノードの能力と要求に見合ったワークフローのセットアップと実行においてこの情報を使用するコンポーネントである。ワークフローマネージャはまた、メタデータの一部を使って、ローカルで保存されていない資産とメタデータを発見する。ワークフローそのものは、Business Process Execution Language (BPEL) によるステートメントで具体的に説明される。ノードコントローラ15は、BPELによるワークフローの説明をノードデータストア17に保存し、これらを、ノードコントローラ15の一部である実行エンジンを使って実行する。そのノードに関するワークフローはそのノードに保存されるため、ワークフローはノードごとに違うこともあり、能力レベルの異なるノード上で違うこともある。

【0032】

各ノードのシステムストレージは、そのノード上の資産の全部を含む。資産ストレージは、資産マネージャによって管理される。これらの資産マネージャは、そのシステムの他のコンポーネントと通信し、分散システムに対して、資産にアクセスし、これを管理する方法を提供する。各資産マネージャは、それが担当するデータに関する情報を管理または要求するためのメッセージを受け入れる。これらは、システムと、そのシステムがサポートできる各種の資産ストレージの間の標準的インタフェースとなる。

【0033】

資産ストレージの種類に対応する資産マネージャには、3つの基本的な種類がある。

1) ユーザ資産マネージャ (User Asset Manager) (UAM) - UAMは、ユーザが選択した保存と整理の方法にしたがって位置付けられ、整理される、ローカルコンピュータ上に保存された各種の資産ファイルを管理する。このシステムソフトウェアをインストールしたユーザは、ローカル資産ファイルがどこにあるかをソフトウェアに伝えるが、ローカル資産をいずれかの特定の場所に移す必要も、またいずれかの特定の階層に従って整理する必要もない。ユーザは、ローカルコンピュータファイルシステムユーティリティを使って、これらのファイルをまったく自由に追加、削除または移動することができる。このことから、UAMにより管理されるストレージは、ユーザ制御によるストレージと呼ばれる。

2) サムネイル資産マネージャ (Thumbnail Asset Manager) (TAM) - TAMは、ある人物の仮想集合の中のすべての資産に関するサムネイルファイルを含むサムネイルキャッシュを管理する。各ノードのサムネイルキャッシュは、そのノードで実行されるシステムサービスによって完全に制御される。

3) キャッシュ資産マネージャ (Cache Asset Manager) (CAM) - CAMは大量資産キ

10

20

30

40

50

キャッシュを管理するもので、あるノード上のシステムコントローラによって、コントローラが他のノードからコピーした資産のコピーを保存するのに使用される。これは、そのノード上で実行されるシステムサービスによって完全に管理される。UAMはまた、仮想集合と各種の実物資産ストレージの間の統一されたインタフェースにもなる。新しい種類のUAMを追加することにより、システムを将来開発される種類の資産ストレージ媒体にも拡張できる。

【0034】

より具体的には、ユーザ資産マネージャは、資産データをユーザの資産管理仮想集合に供給することを担当する。UAMは一般に、ファイルスニファ(file sniffer)または、それが管理するディレクトリの中の資産の追加、削除および変更を検出するその他何らかの方法を含む。UAMは、ある資産の変更を検出すると、ノードコントローラにメッセージを送信する。ノードコントローラはそのメッセージに応答して、ワークフローを開始する。新たに発見された資産に関して、このワークフローは、その資産が新しいか、それとも既存の資産の複製かを判断する。新しい資産であれば、ワークフローはその資産ファイル内に含まれるすべてのメタデータを抽出し、その資産のサムネイルを作成し、その資産に関するすべてのメタデータをメタデータストアの中に保存する。サムネイルを作成させたワークフローはまた、異なる解像度のいくつかのサムネイルを作成させることもできる。ワークフローがこれを行うか否かは、それぞれのノードに実装されているワークフローポリシーによる。

【0035】

サムネイル資産マネージャは、ノードによって保存されたサムネイルを管理する。これらのサムネイルの中には、ローカルで生成されて、新しい資産の発見に適用されるワークフローによってサムネイルキャッシュの中に移されたものがあれば、他のノードとのメタデータ同期を制御するワークフローによってサムネイルキャッシュに移動されたものもある。ユーザがある資産を削除すると、資産削除を制御するワークフローがそのサムネイルを削除する。

【0036】

キャッシュ資産マネージャは、ノード上の一時的な保存を制御する。キャッシュ資産マネージャは、ローカルの大量資産キャッシュ14の中の資産を管理する。この一時的ストレージには、そのノードにコピーされ、そこに一時的に保存されている、その他のノードからの資産が含まれる。キャッシュ資産マネージャは、ローカルキャッシュの中の資産のコピーを追跡し、必要に応じてこれらを削除して、そのキャッシュのディスクスペース割当を管理する。ワークフローポリシーが、キャッシュからの資産の削除を制御する。たとえば、変更された資産は、同期または、ローカルのレプリカが存在する場合にはローカルワークフローを介して、その資産の少なくとも1つの他のレプリカにその変更が適用されるまで削除されない。別の例として、「人気のある」資産(すなわち、多くの他のユーザと共有される資産)は、より人気の低い資産より優先的にキャッシュに保持されるかもしれない。

【0037】

同様に、需要の高い資産(すなわち、ユーザが頻繁にアクセスする資産)を優先的に保持することができる。別の例として、喪失するリスクがより高いと判断される資産(たとえば、バックアップコピーが少ないもの、携帯電話のように保存の信頼性が低いデバイスに保存されているもの)は、よりリスクが低いと判断される資産(たとえば、信頼性の高いオンラインサービス上で保存されるバックアップコピー)より優先的に保持することができる。資産をキャッシュから削除するワークフローは、ノードごとに異なることがある。CAMは、そのキャッシュの中のある資産に加えられた変更を検出すると、ワークフローマネージャに通知する。すると、ワークフローマネージャは、この変更に対処するための適当なステップを開始する。

【0038】

大量資産キャッシュ14は、当初は空である。大量資産は、ワークフローが、別のノード

10

20

30

40

50

ドに保存された資産のコピーをローカルでも保存するように要求した時に、キャッシュに追加される。ある資産のコピーをキャッシュに保存することにはさまざまな理由がある。1つの理由は、あるノード上のアプリケーションが、そのレプリカのすべてが他のノードにあるような資産にアクセスしたいと望むことである。その場合、ノードコントローラ15が、その資産をローカルキャッシュに移し、その後、アプリケーションに、その新しいローカルのレプリカへのアクセスを提供する。大量資産キャッシュ14を使用する、より興味深い理由は、便宜、性能、安全、または可用性の検討事項に関わる。資産の中には、ユーザにとって他の資産より重要なものがある。ある人物は、その資産に重要であるとマークするかもしれない。あるいは、その重要性は、この資産に関するユーザの行動の記録から、またはその資産のコンテンツの解析から導き出すことができる。いずれの場合も、資産、たとえば、特定のピクチャの重要性は、そのピクチャのメタデータプロパティとなる。あるユーザが、その資産を誤って削除されないように保護するべきであると指示した場合、分散システムは確実に、その資産のレプリカを複数カ所に置くようにすることができる。重要な資産の唯一のコピーが、時々しかオンラインとまらない機械の上にある場合、本願による分散システムは、この重要な資産を、より高い信頼性でオンラインとなるノードに、または常時オンラインの中央ノード5に移すことができる。このような場合、選択されたピクチャは、他のノードの上の大量資産キャッシュに移されるであろう。あるノードのワークフローポリシーが、その資産のコピーを別のノード上にあるユーザ制御によるストレージに移すように指示することさえ可能である。ここでも、ワークフローがこれらの資産の移動を管理する。ワークフローは、システムのAPIを通じて伝えられるユーザの行動によって開始することができ、また、同期命令によっても開始することができ、これは同期サイクルの完了によってユーザの集合の状態の変化が示されるからである。これらのワークフローを容易にするために、各ノードに関連付けられるノード可用性メタデータがある。このノード可用性メタデータは、そのノードが連絡する他のノードと共有される。ノード可用性メタデータによって、ノードは、可用性の理由から、資産を他のノードに移すべきか否かを知ることができる。

10

20

30

40

50

【0039】

このシステムは、システムメンバデータのコピー（レプリカ）を、特定の時点でオンラインかもしれないし、オフラインかもしれないシステムノードに保存する。このシステムにより、各ノードは、そのデータを他のノード上のデータと調和させて、分散システムが全体として自己矛盾のない状態になるようにすることができる。このシステムにより、メンバは、現時点でオンラインか否かに関係なく、その集合を用いた作業を行うことができる。これは、他のノードを更新することができないときに、データをノード上で変更できることを意味する。また、同じデータの異なるコピーに異なる変更を加えることも可能である。同期は、異なるノード上に存在する同じデータのコピー間の違いを調整することによって、すべてのシステムノード間でメンバデータをコヒーレント状態にするためにシステムが使用するプロセスである。

【0040】

システムデータは、多くの従来の複製システムとは、各資産に関する情報が異なるノードに多かれ少なかれ保存されうるという点で異なる。（従来の同期メカニズムは、すべてのデータがすべてのノード上で複製されるフルレプリカ用に設計されている。）本明細書において、「より低能力のノード」とは、必ずしもすべてのメタデータを保存するとはかぎらないノードを指す。すべてのノードに少なくともいくつかのメタデータが保存される一方で、いずれかのノードにいくつかの大量資産が存在してもよいし、しなくてもよい。本発明において、システム同期の実装は、1つのメカニズムによるメタデータと大量資産の両方の同期をサポートする。

【0041】

2つのノードに同じデータの異なるバージョンがあるとき、それは3つの異なる理由で正しいといえる。第一のノードに、そのデータのより最近のバージョンがあるかもしれず、第二のノードに、そのデータのより最近のバージョンがあるかもしれず、あるいは両方

のノードに、そのデータの同程度に最近ではあるが異なるバージョンがあるかもしれず、これは、ノードが最後に同期されて以降、各ノードのユーザらが同じデータを変更したからである。このシステムでは最後の可能性が特に高いが、それは、このシステムではノードがオフラインで動作でき、これらを常に同期できるとはかぎらないからである。このシステムで、どの可能性が発生したかを効率的に判断できるようにするために、システムはバージョンベクタを使用する。バージョンベクタと変更リストは、システムがシステムノード上のレプリカとシステムデータのバージョンを追跡する手段であり、また、同期が必要か否かを判断するための主要なメカニズムである。

【0042】

システム変更は、1つのノードの上のシステムデータに加えられた1つまたは複数の変更のまとめであり、システムはこれに連続するバージョン番号を付与する。システム変更の例としては、写真のタイトルの変更、1つのアプリケーションからの複数のエディットの取入れ、およびシステムユーザまたは資産マネージャによって発見された複数の資産の追加等がある。1つより多いシステム変更が同期動作中に行われることがある。複数のシステム変更は、システムの変更リスト(Change List)の中に保存される。各ノードは、システムの変更リスト内に自己の変更を保存するだけでなく、それが同期中に他のノードから受け取った変更のコピーも保存する。特定のノードに関するすべてのシステム変更から最も高いバージョン番号がそのノードの、その各々のバージョンベクタの要素の中に保存される。システムのバージョンベクタは、メタデータへの変更を明示的に追跡し、大量資産レプリカへの変更は黙示的に追跡する。バージョンベクタ30(図3aに示す)は、各ノードのための要素31を含むアレイである。1つの実装例において、メタデータまたは資産を保存する各システムノードは、そのノード上にデータを有する各システムメンバに関してバージョンベクタを1つだけ有する。したがって、バージョンベクタは、そのシステムメンバがデータを保存する各ノードについて1つの要素を含む。

【0043】

1つの実施の形態において、バージョンベクタは、これまでにその関連するメタデータに変更を行ったことのあるノードだけを説明する。システムは、最上位のバージョンベクタをシステム変更リストとともに使用して、すべてのシステムメンバノードを通じたシステムメンバデータを追跡し、同期する。バージョンベクタの各要素31は、特定のノードに関するユーザに対応する同期ID32(たとえば、図3aの“A”)を、そのユーザのためのそのノードのデータが変更されるたびにソフトウェアがカウントを増やすカウンタ33(たとえば、図3Aの“9”)(すなわち、「論理バージョン」)とともに保有するあるノードは、あるユーザのシステムデータについて同じ同期IDを保持するが、それがそのデータベースを喪失し、完全に更新しなければならない場合、または他のノードがその変更リストの中に必要な変更のコピーを持たなくなるために、通常の同期をサポートできない場合を除く。特定のメンバノードに関するバージョンベクタには、そのメンバノードに関する現行バージョン情報と、最後に同期が行われた時点でそのメンバが使用していた他のノードに関する改訂情報が含まれる。各ノード上の集合の各共有部分については、また別のバージョンベクタがある。

【0044】

同期は少なくとも1対のノードの間で行われる。同期サイクル(“sync cycle”)は、あるノードに、その最後の同期サイクル以降、所定の回数だけ変更が行われたとき、または最後の同期サイクル以降、所定の時間が経過した時に行われる。あるノードでそれが行われると、そのノードはあるユーザのためのバージョンベクタを、それがユーザの他のノードから過去に受け取った、そのユーザのためのバージョンベクタのコピーと比較して、これらの他のノードのうちのどれが、それより低い改訂レベルにあり、したがってそのデータで更新される必要があるかを判断する。ノードは、これらのその他ノードに「同期サイクル必要」というメッセージを送信する。このメッセージを受け取ったノードは、「同期サイクル必要」のメッセージを送信したノードとの同期サイクルを開始する。同期サイクルは、1つのノード、すなわちそのバージョンベクタを他のノード(ソース

）に送信する「要求者」から始まる。ソースは、要求者の中のバージョン番号とそのバージョンベクタを比較する。目的は、ソースが要求者に対して、最後に要求者と同期して以降、ソースにおいて発生したすべての変更（要求者が、その能力レベルに基づいて要求するもの）を送信することである。バージョンベクタの比較により、ソースが目的地に変更を送信すべきであることが示された場合、ソースは、その変更リストを見て、そのバージョンベクタ内の改訂レベルの数値に応じて、要求者に送信する必要がある変更はどれかを判断する。同期サイクルが完了したら、要求者は新しい変更レベルを反映させるように、そのバージョンベクタを更新し、その後、それが受け取った、同期サイクルを求める他のメッセージに応答する。

【 0 0 4 5 】

10

図 5、図 6、図 7、図 8 は、異なるノード能力を異なる組み合わせにしたノードに関する同期サイクルの中のステップを示す。これらのステップは好ましくは、プロセッサ、コンピュータまたは、特定用途集積回路（ASIC）によって実行される。これらのステップは、コンピュータにより読取可能な記憶媒体、たとえば、磁気ディスク（たとえば、ハードドライブまたはフロッピー（登録商標）ディスク）または磁気テープ等の磁気記憶媒体、光ディスク、光テープまたは機械により読取可能なバーコード等の光記憶媒体、ランダムアクセスメモリ（RAM）またはリードオンリメモリ（ROM）等のソリッドステート電子記憶デバイス、またはコンピュータプログラムの保存に利用されるその他の物理的デバイスまたは媒体等に保存可能なコンピュータプログラムであってもよい。

【 0 0 4 6 】

20

それぞれの場合において、同期サイクルは、バージョンベクタの交換から始まる。図 5 では、バージョンベクタは同一であり、したがって、それ以上の処理は行われない。具体的には、ノード 2 とノード 3 の交換について記されているステップ 5 0 1 で、ノード 2 は、そのバージョンベクタ__N 2（VV__N 2）で同期開始（Start Sync）メッセージをノード 3 に送信する。ステップ 5 0 2 で、ノード 3 は VV__N 2 を自己のバージョンベクタと比較する。比較の結果、ノード 3 は変更を検出して、ノード 2 にはない、ノード 3 が有する変更リストを編集する。ステップ 5 0 3 で、ノード 3 は VV__N 3 と変更リストで、同期開始（Start Sync）応答メッセージを送信する。

【 0 0 4 7 】

30

ステップ 5 0 4 で、ノード 2 はノード 3 を自己のバージョンベクタと比較する。比較の結果、ノード 2 は変更を検出し、ノード 3 によって送信された変更リストを見る。ノード 2 はその後、自己の変更リストとノード 3 の変更リストの間に矛盾がないかチェックする。この例においては、矛盾はない。ステップ 5 0 5 で、ノード 2 はその変更リストをノード 3 に送信する。ステップ 5 0 6 で、ノード 2 はノード 3 の変更を自己のデータストアの中に組み込み、追加の変更を反映させるようにその変更リストを更新する。これに加え、VV__N 2 が更新される。同様に、ステップ 5 0 7 で、ノード 3 はノード 2 の変更を自己のデータストアの中に組み込み、その変更リストを、追加の変更を反映させるように更新し、VV__N 3 が更新される。この例では、最終ステップ 5 0 8 において、VV__N 2 と VV__N 3 は同じである。

【 0 0 4 8 】

40

図 6 において、1つのノード（ノード 0）は、バージョンベクタから、2つのノード（ノード 0 とノード 2）の同期が必要であることを検出する。これらのノードは変更リストを交換し、変更リストを使って、どのメタデータを変更する必要があるかを判断し、2つのノードを同期させるために必要なすべてのメタデータを交換する。特に、ステップ 6 0 1 で、ノード 2 は、そのバージョンベクタ__N 2（VV__N 2）でノード 0 に同期開始（Start Sync）メッセージを送信する。ステップ 6 0 2 で、ノード 0 は VV__N 2 を自己のバージョンベクタと比較する。したがって、ノード 0 は、ノード 2 にノード 0 が必要とする変更があるが、ノード 0 にはノード 2 が必要とする変更がないことを知る。ステップ 6 0 3 で、ノード 0 は VV__N 3 で同期開始（Start Sync）応答メッセージを送信する。その後、ステップ 6 0 4 で、ノード 2 はノード 0 を自己のバージョン

50

ベクタと比較する。ノード 2 は、ノード 0 に送信する必要がある変更があることを知るか、これを検出する。したがって、ステップ 605 で、ノード 2 はその変更リストをノード 0 に送信する。

【0049】

ステップ 606 において、ノード 0 はノード 2 からの各変更を調べ、評価し、それがノード 0 によって保存されたものに影響を与えるかどうか判断する。保存された資産に影響がある場合、ノード 0 はそのストアと変更リストを更新する。保存された資産に影響がない場合、ノード 0 は変更を廃棄し、変更リストの中にプレースホルダとして空値 (null) を追加する。最終ステップのステップ 607 では、VV_N2 と VV_N0 は同じであるが、ノード 0 には、そのバージョンベクタに部分更新フラグがセットされている。

10

【0050】

図 7 と図 8 において、1 つのノードはもう一方のノードより能力が低く、すなわち、そのノードは必ずしもすべてのメタデータを含んでいない。携帯電話に位置するノードはデスクトップ PC に位置するノードより能力が低い、これは、携帯電話はデスクトップ PC ほど各資産に関するメタデータを多く保存できないからである。携帯電話の能力データには、携帯電話のノードにどの種類のメタデータが保存されていないかの説明が含まれる。これらの図は、同期サイクルがより高い能力のノードとより低い能力のノードのどちらから始まっているかが異なる。プロセスは、能力が等しいノードの場合と、能力の低いノードがそのデータのすべてを更新するとは限らないという点において異なる。図はまた、同期サイクルが起こった前後のバージョンベクタを示す。これらのバージョンベクタは括弧の中に記されている。注記は、当該のノードが知っているすべてのその他のノード、これらのノードの各々のバージョンレベル、およびこれらのノードのいずれかが部分的に更新されているか否かを示す。

20

【0051】

具体的には、図 7 に示される実施の形態に関して、ステップ 701 で、ノード 3 はその VV_N3 で同期開始 (Start Sync) メッセージをノード 0 に送信する。ステップ 702 で、ノード 0 は VV_N3 を自己のバージョンベクタと比較する。ノード 0 は、それにノード 3 が必要とする変更があることを検出する。ノード 3 には、それが必要とする変更がない。ステップ 703 で、ノード 0 は VV_N0 で同期開始 (Start Sync) 応答メッセージと、そのバージョンベクタの N2 : 6 と N2 : 6 に配置するための変更リストを送信する。ステップ 704 で、ノード 3 はノード 0 を自己のバージョンベクタと比較する。ノード 3 は、それがノード 0 に送信する必要がある変更があることを知る。

30

【0052】

ステップ 705 で、ノード 3 はノード 0 からの変更リストを見直す。ノード 0 は、場所 N2 : 5 に空値があることを知り、これはノード 0 が変更を保存していないことを示す。これに加えて、ノード 3 は N2 : 6 の変更を知り、したがってノード 3 のデータベースが更新される。特に、ノード 0 のバージョンベクタは N2 に “p” を有し、したがってノード 3 も、ノード 3 について “p” を設定する。ステップ 706 で、VV_N3 と VV_N0 は同じであり、どちらもノード 2 のバージョンベクタを有し、VV には部分更新フラグがセットされている。

40

【0053】

図 8 に示した実施の形態において、ステップ 801 は、ノード 3 がその VV_N3 で同期開始 (Start Sync) メッセージをノード 0 に送信することを示している。ステップ 802 において、ノード 0 は VV_N3 を自己のバージョンベクタと比較して、ノード 3 が必要とする変更を検出する。この例において、ノード 3 には、それが必要とする変更がない。続いて、ステップ 804 で、ノード 3 はノード 0 を自己のバージョンベクタと比較して、それがノード 0 に送信する必要がある変更があるか否かを検出する。ステップ 805 で、ノード 3 はノード 0 からの変更リストを見直す。N2 : 5 には空値があり、これはノード 0 が変更を保存しなかったことを示す。変更が N2 : 6 にあり、したがって

50

、対応するデータベースが更新される。ノード0のバージョンベクタはN2に“p”があるため、ノード3もノード3について“p”を設定する。最終ステップ806では、同じバージョンベクタ、VV__N3とVV__N0があり、どちらもノード2のバージョンベクタを有し、VVには部分更新フラグがセットされている。

【0054】

共有データの同期サイクルは、若干異なる方法で実行される。第一のユーザがアカウントを有するノードに他のユーザと共有されていたデータが含まれていれば、それは共有データに対応するバージョンベクタを有する。共有データが第一のユーザのための特定のノードで変更した場合、そのノードは、第二のユーザのためのアカウントを有する別のノードと同期サイクルを開始する。第二のユーザのノードが同期された後、それは通常の同期プロセスを使って、それ自身を第二のユーザのアカウントを有する他のノードと同期させる。

10

【0055】

変更リストは時間の経過と共に増大するため、これらは時々、ブルーニング(pruned)を行う必要がある。すべてのノードが常にオンラインであれば、ブルーニング(pruning)は容易であり、ある変更が他のすべてのノードに伝搬されるとすぐに、その変更リスト項目は不要となり、削除できる。事実、ノードのペアが同期されると、システム全体が実質的に同期され、これは、ノードの実質的な部分が相互に同期されるからである。例外は、オフラインのノードであり、これはオンラインになるまで同期することができない。あるノードが所定の時間だけオフラインであると、他のノードは、オフラインのノードがそれらの変更を受け取るまで待たず、その他すべての、オフライン以外のノードに伝搬された変更に関する変更リスト項目を削除する。この削除プロセスがブルーニングと呼ばれる。オフラインであったノードがオンラインに戻ると、これは再び他のノードとの同期サイクルに加わる。これらの同期サイクルは通常通りに進めることができず、これは、他のノードに、そのノードがオフラインになった後にこれらのノードに発生した変更に対応する変更リスト項目は、ブルーニングを行ってしまったために、なくなっているからである。したがって、以前にオフラインであったノードはその後、修正後の同期サイクルに参加する。まず、このノードは、オフラインであった間に発生した変更を第二のノードに送信する。これらのオフラインでの変更はすべて、矛盾する可能性がある想定され、ユーザには、これらの変更を受け入れるべきか否かを判断するように求められる。次に、矛盾が解消された後に、以前にオフラインであったノードは、そのメタデータストア全体と変更リストを、それがその変更を送信した先である、同じ第二のノードから更新する。これの、より単純な例は、新しいノードがオンラインで加わり、その他すべてのノードがその変更リストをすでにブルーニングした場合である。新しいノードは、そのメタデータストア全体を他のノードから更新する。特に、この例では、新しいノードには何の変更もない可能性が非常に高い。

20

30

【0056】

デジタル資産管理における共有は、他のユーザ(時々、「共有受信者」と呼ばれる)は、あるユーザの集合の中のデジタル資産にアクセスできることを意味する。他のユーザが有するアクセスの量は、その資産を所有するユーザが管理する。部分的にのみ信頼される共有受信者には、その資産に関するメタデータの一部への読取アクセスしか認めなくてもよく、全面的に信頼される共有受信者には、資産そのものを修正する権利を与えることができる。共有は、1人のユーザによって開始または提供され、他のユーザに向けられる。第二のユーザには共有招待のメッセージが送られ、共有招待を受け入れても、拒絶してもよく、応答メッセージを返す。このメッセージは、他のノード間通信と同じピアツーピアネットワークを使用する。共有招待が受け入れられた場合、それはまた、共有資産に関する情報をメタデータに追加するワークフローと、提供ユーザによる共有資産の追跡を開始する。ここでも、共有資産の所有者は、共有資産に関して、どれだけのアクセスが共有受信者に付与されるかを管理する。本システムの実施の形態において、共有者は、各共有受信者に役割を割り当てる。この割り当てられた役割は、共有受信者の識別情報とともに共

40

50

有者のメタデータストア 16 の中に保存される。ノードコントローラ 15 は、この割り当てられた役割を使って、共有受信者が共有資産を読み、または変更することを許可するかどうかを判断する。

【0057】

資産を第二のユーザと共有するために、提供するユーザはまず、自分のコンタクトリストに第二のユーザを入れる必要がある。システムネットワーク内のどのノードも、中央ノード 5 に尋ねることによって、他のユーザを利用できるか否かがわかるが、これはすべてのユーザが中央ノード 5 と連絡しているからである。中央ノード 5 が利用できない場合は、ノードは少なくとも、それがそれらのユーザのために連絡しているノードに尋ねることができる。システムメンバが潜在コンタクトのリストを入手したら、そのメンバは、コンタクトとなるように招待するメッセージの送信先となるべきユーザを選択することができる。メッセージは、他のユーザメッセージと同様に、メッセージの宛先となる人物のアカウントを含むことがわかっている少なくとも 1 つのノードに送信され、同期プロセスを通じて、そのユーザを含む他のノードに伝搬される。第二のユーザがコンタクトとなると、提供するユーザは、第二のユーザと資産を共有することができる。提供するユーザは、ユーザメッセージを潜在的な共有受信者に送信し、選択された資産の共有にその共有受信者を招待する。共有受信者が共有招待を受け入れると、共有受信者が作業しているノードは、提供する使用者のノードの一つに、その資産を説明するメタデータを照会し、そのメタデータを、それが共有されていることを示す追加の属性とともにその集合メタデータに追加する。

10

20

【0058】

図 3 は、「ボブ」と「テッド」が関わる例としてのフローチャート 300 における共有資産の同期に関するステップを示す。これらのステップは好ましくは、プロセッサ、コンピュータまたは、特定用途集積回路 (ASIC) によって実行される。これらのステップは、コンピュータにより読取可能な記憶媒体、たとえば、磁気ディスク (たとえば、ハードドライブまたはフロッピー (登録商標) ディスク) または磁気テープ等の磁気記憶媒体、光ディスク、光テープまたは機械により読取可能なバーコード等の光記憶媒体、ランダムアクセスメモリ (RAM) またはリードオンリメモリ (ROM) 等のソリッドステート電子記憶デバイス、またはコンピュータプログラムの保存に利用されるその他の物理的デバイスまたは媒体等に保存可能なコンピュータプログラムであってもよい。

30

【0059】

最初のステップ 310 において、ボブはテッドと資産 “A” を共有する。ステップ 315 で、システムは共有資産 A のメタデータをテッドの集合にコピーし、ボブを所有者として特定する。ステップ 320 で、ボブは資産 A と B のためのメタデータを修正する。ステップ 325 で、システムは資産 A と B に関する変更をボブのノードの全部に同期させる。これに加えて、システムは資産 A の変更をテッドのノードの全部に同期させる (すなわち、‘s y n c h s’)。ステップ 330 で、テッドは資産 A と C のメタデータを修正する。

【0060】

ステップ 335 で、システムは A と C の変更をテッドのノードの全部に同期させ、その一方で、資産 A に関する変更をボブのノードの全部に同期させる。ステップ 340 で、テッドはそのノードを分散システムから切断する (すなわち、オフラインにする)。ステップ 345 で、テッドは、ボブから切断されている間に資産 A のメタデータを修正する。続いて、ステップ 350 で、テッドは自分のノードを分散システムに再接続する。ステップ 355 で、システムは A の変更をテッドのノードの全部とボブのノード全部とに同期させる。

40

【0061】

ステップ 360 で、テッドは資産 A の画像データを編集することに決める (ボブがテッドにその許可を与えている)。ステップ 365 で、システムはテッドの仮想集合の中の資産 A のメタデータを使ってレプリカを見つけ、それをコピーして、テッドが編集するため

50

に、テッドのシステムキャッシュの中に新しいレプリカを作成する。システムは資産 A のメタデータを更新して、新たに作成されたレプリカを含める。ステップ 370 で、システムは A のメタデータの変更をテッドのすべてのノードとボブのすべてのノードと同期させる。

【0062】

ステップ 375 で、テッドは資産 A の画像データを編集する。ステップ 380 で、システムは画像の変更を資産 A のレプリカの全部に同期させる。ステップ 385 で、テッドは再び、分散システムから切断する。最後のステップ、ステップ 390 で、テッドは資産 A からの画像データを自分のプリンタで印刷する（システムは、資産ファイルのキャッシュに入っているコピーを使用する）。

10

【0063】

本発明の別の特徴において、あるノードの UAM が新たな資産を検出すると、システムはそれが完全に新しい資産であるか、すでにシステムに知られている資産のコピーであるかを判断する必要がある。その資産がすでにシステムに知られている資産のコピーである場合、分散システムはその資産ファイルを、新規資産ではなく、既存資産のレプリカと分類する。資産が、ノードがすでにその資産のためのメタデータを含んでいる場合に固有であるか否かを判断する手段は知られている。しかしながら、本明細書に記載されている分散システムは、オフラインのノードもこれを使用することができるため、相互に連絡していない 2 つのノードはどちらも、同じ資産を追加して、両方の複製が個別に集合データベースに追加されるまで、複製を検出できないようにすることが可能である。ノードが連絡を取ると、いずれかの資産がノード間で複製であると特定される必要があるかないかを判断する効率的な手段が必要である。本発明の分散システムは、オフラインノードによって発見された新しい資産を「暫定新規資産」として記録することにより、複製特定の問題を解決する。分散システムは、これを行うために、オフラインノードで新たに発見された資産に、「暫定新規資産プロパティ」というある種の資産メタデータを割り当てる。暫定新規資産プロパティは、その資産について複製チェックが完了しておらず、まだ行う必要があること示す、明確なインディケータとなる。図 4 は、一例としてのフローチャート 400 の中の暫定新規資産プロパティと「マイピクチャ」とラベリングされたファイルを割り当てて、使用するステップを示す。

20

【0064】

これらのステップは好ましくは、プロセッサ、コンピュータまたは、特定用途集積回路 (ASIC) によって実行される。これらのステップは、コンピュータにより読取可能な記憶媒体、たとえば、磁気ディスク（たとえば、ハードドライブまたはフロッピー（登録商標）ディスク）または磁気テープ等の磁気記憶媒体、光ディスク、光テープまたは機械により読取可能なバーコード等の光記憶媒体、ランダムアクセスメモリ (RAM) またはリードオンリメモリ (ROM) 等のソリッドステート電子記憶デバイス、またはコンピュータプログラムの保存に利用されるその他の物理的デバイスまたは媒体等に保存可能なコンピュータプログラムであってもよい。

30

【0065】

ステップ 410 で、ユーザは「マイピクチャ」というファイルをそのシステム仮想集合に追加する。ステップ 420 で、システムは資産 (ID、種類、サムネイル等) とレプリカ (ID、種類、サムネイル等) のメタデータを作成し、これをユーザの仮想集合メタデータに追加する。ステップ 430 で、システムは「暫定新規資産」プロパティを、ユーザの集合の中に新たに追加された資産のメタデータに追加する。ステップ 440 で、システムは現在接続されているユーザの全ノードに照会することによって、複製検出方法を適用する。

40

【0066】

ステップ 445 で問い合わせを行い、複製が見つかったか判断する。複製が見つかった場合、ステップ 450 で、「マイピクチャ」を既存の資産のレプリカとしてラベリングする。ステップ 460 で、システムは新規に追加された資産に関するレプリカのメタデータ

50

を既存の資産に移動させる。言い換えれば、既存の資産が追加のレプリカを獲得する。ステップ 470 で、システムは、ユーザの集合から、そこに残っている新規に追加された資産のメタデータを削除して、ワークフローが完了する。

【0067】

あるいは、レプリカが見つからない場合、ノードが再接続されると、ステップ 475 でまた別の問い合わせにより、そのユーザのすべてのノードに照会して複製の有無を確認する。ユーザの全ノードへの照会が終了したら、その資産は新しいと考えられ、ステップ 480 で暫定プロパティの指定が削除され、その後ワークフローが完了する。

【0068】

ユーザのすべてのノードの照会が済んでいない場合、ステップ 485 でさらに問い合わせを行うことによって、特定のノードがオンラインであるか否かを判断し、オンライン状態になるまでポーリングを続ける。ステップ 490 で、システムは、反復サイクル中に新規に接続されたノードに複製の有無を照会するのに複製検出方法を適用する。

10

【0069】

オフラインであったノードが連絡した時に、複製資産を検出する別の方法を以下に説明する。ノードは、連絡すると、それらのメタデータを同期させる。このプロセスによって、1つのノードに追加された資産が他のノードのメタデータに追加される。2つのノードの同期では、これらのノード間の資産変更リストが交換されるため、また、あるノードの変更リストによってどの資産が新しいかが示されるため、特定の資産に関する複製チェックを実行する必要性は、同期中に交換される資産変更リストから推定される。

20

【符号の説明】

【0070】

- 1 ノード
- 2 ピアツーピアリンク
- 3 アプリケーション
- 5 中央ノード
- 10 通信するコンポーネント
- 11 APIコンポーネント
- 12 ネットワークインタフェース
- 13 ストレージコンポーネント
- 14 資産キャッシュ
- 15 ノードコントローラ
- 16 メタデータストア
- 17 ノードデータベース
- 18 サムネイルキャッシュ
- 30 バージョンベクタ
- 31 バージョンベクタの要素
- 32 同期ID
- 33 バージョンカウンタ
- 300 フローチャート
- 310 最初のステップ
- 315 ステップ
- 320 ステップ
- 325 ステップ
- 330 ステップ
- 335 ステップ
- 340 ステップ
- 345 ステップ
- 350 ステップ
- 355 ステップ

30

40

50

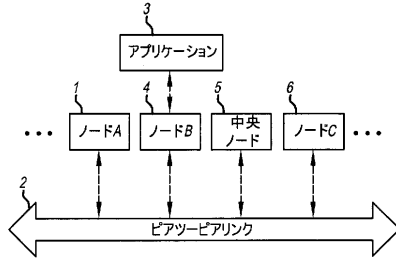
3 6 0 ステップ
3 6 5 ステップ
3 7 0 ステップ
3 7 5 ステップ
3 8 0 ステップ
3 8 5 ステップ
3 9 0 ステップ
5 0 1 ステップ
5 0 2 ステップ
5 0 3 ステップ
5 0 4 ステップ
5 0 5 ステップ
5 0 6 ステップ
5 0 7 ステップ
5 0 8 ステップ
6 0 1 ステップ
6 0 2 ステップ
6 0 3 ステップ
6 0 4 ステップ
6 0 5 ステップ
6 0 6 ステップ
6 0 7 ステップ
7 0 1 ステップ
7 0 2 ステップ
7 0 3 ステップ
7 0 4 ステップ
7 0 5 ステップ
7 0 6 ステップ
8 0 1 ステップ
8 0 2 ステップ
8 0 3 ステップ
8 0 4 ステップ
8 0 5 ステップ
8 0 6 ステップ

10

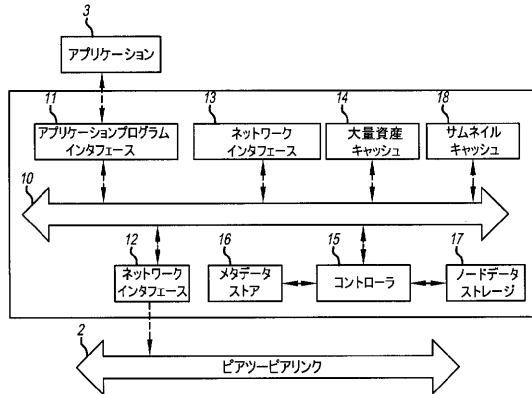
20

30

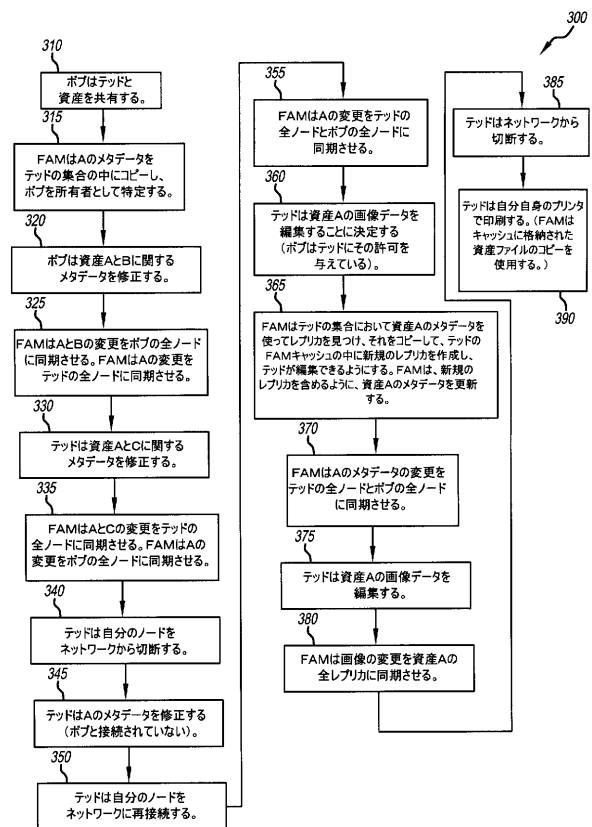
【図 1】



【図 2】



【図 3】



【図 3 a】

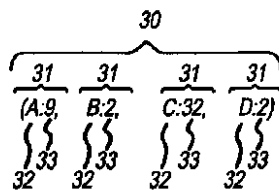
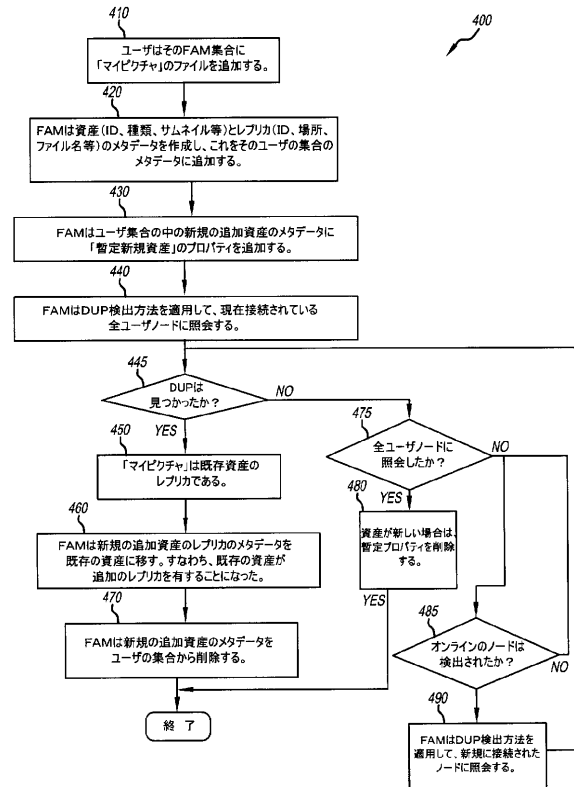
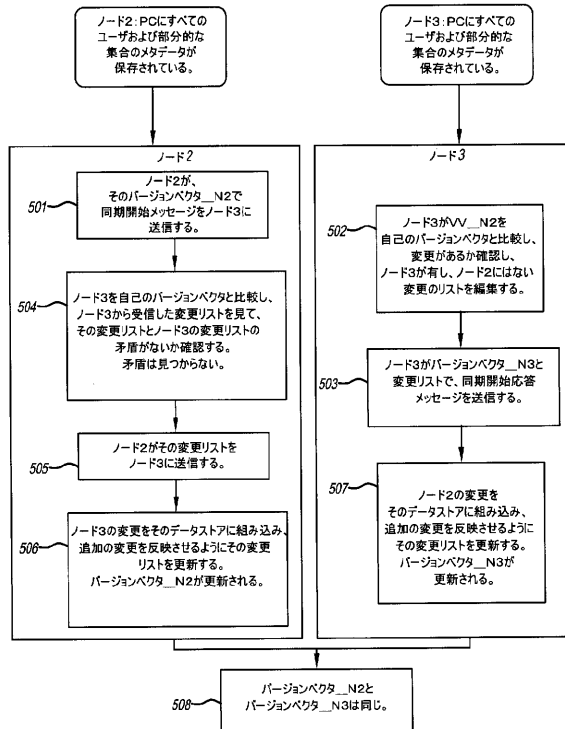


FIG. 3a

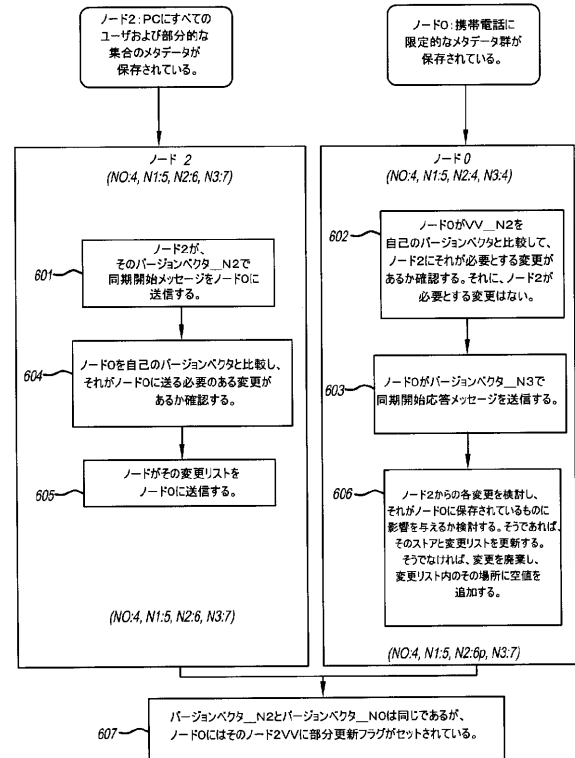
【図 4】



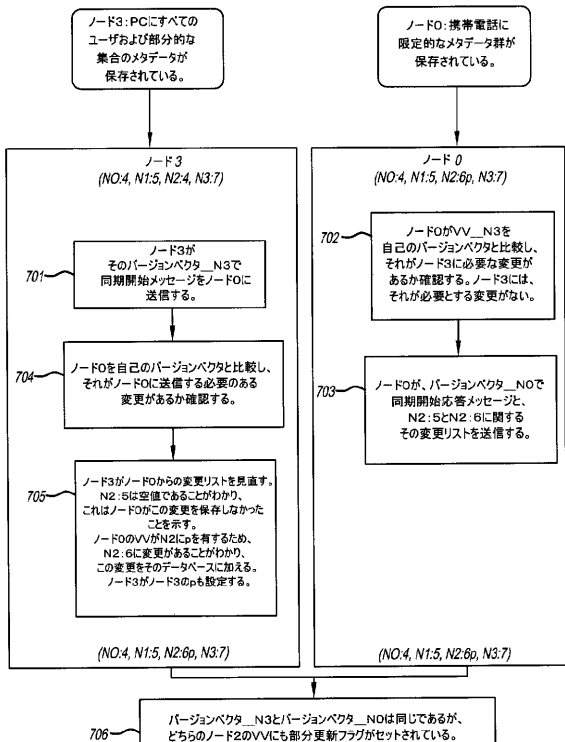
【図 5】



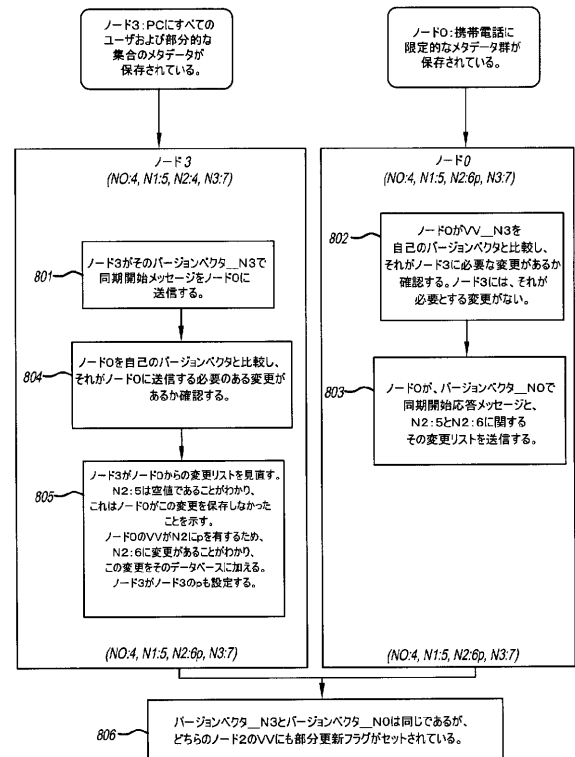
【図 6】



【図 7】



【図 8】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2010/000101

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F17/30

ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|-----------|---|-----------------------|
| X | ANDROUTSELLIS-THEOTOKIS S ET AL: "A survey of peer-to-peer content distribution technologies" ACM COMPUTING SURVEYS, ACM, NEW YORK, NY, US, US, vol. 36, no. 4, 1 December 2004 (2004-12-01), pages 335-371, XP002412530 ISSN: 0360-0300 the whole document ----- -/- | 1-9 |

☒ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

27 May 2010

Date of mailing of the international search report

21/06/2010

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Bernardi, Luca

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2010/000101

| C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT | | |
|--|--|-----------------------|
| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
| X | MARTINS V ET AL: "Survey of data replication in P2P systems" INTERNET CITATION 1 December 2006 (2006-12-01), pages 1-45, XP002501325 Retrieved from the Internet: URL: http://hal.inria.fr/docs/00/12/22/82/PDF/Survey_of_data_replication_1n_P2P_systems.PDF [retrieved on 2008-10-17] the whole document | 1-9 |
| X | WO 2007/024380 A2 (MICROSOFT CORP [US]) 1 March 2007 (2007-03-01) the whole document | 10-13, 18-20 |
| X | DE BRUM SACCOL D ET AL: "Detecting, managing and querying replicas and versions in a peer-to-peer environment" 7TH IEEE INTERNATIONAL SYMPOSIUM ON CLUSTER COMPUTING AND THE GRID (CCGRID '07) IEEE PISCATAWAY, NJ, USA, 2007, pages 833-838, XP002584320 ISBN: 978-0-7695-2833-5 the whole document | 14-17 |
| A | RONALDO A FERREIRA ET AL: "Randomized Protocols for Duplicate Elimination in Peer-to-Peer Storage Systems" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE SERVICE CENTER, LOS ALAMITOS, CA, US LNKD-DOI:10.1109/TPDS.2007.351710, vol. 18, no. 5, 1 May 2007 (2007-05-01), pages 686-696, XP011179651 ISSN: 1045-9219 * abstract page 686 - page 687 | 14-17 |

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2010/000101**Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)**

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☒ No protest accompanied the payment of additional search fees.

International Application No. PCT/US2010/000101

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-9

Distributed system to manage distributed files in a network.

2. claims: 10-13, 18-20

Method for enabling by invitation the sharing of data among users.

3. claims: 14-17

Method for determining duplicate files in a distributed system for data sharing.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2010/000101

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---------------------|----------------------------|---------------------|
| WO 2007024380 A2 | 01-03-2007 | AU 2006284414 A1 | 01-03-2007 |
| | | CA 2616242 A1 | 01-03-2007 |
| | | CN 101243443 A | 13-08-2008 |
| | | EP 1917608 A2 | 07-05-2008 |
| | | JP 2009506423 T | 12-02-2009 |
| | | KR 20080047361 A | 28-05-2008 |
| | | US 2007067349 A1 | 22-03-2007 |
| | | ZA 200800961 A | 29-04-2009 |

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW

(72)発明者 ルーク アンネ シー
 アメリカ合衆国 ニューヨーク ロチェスター ステイト ストリート 3 4 3

(72)発明者 セイキイ ルイス ジェイ
 アメリカ合衆国 ニューヨーク ロチェスター ステイト ストリート 3 4 3

(72)発明者 スワンベック ウェンディ
 アメリカ合衆国 ニューヨーク ロチェスター ステイト ストリート 3 4 3

(72)発明者 シュレッケンガスト ジェームズ オーエン
 アメリカ合衆国 ニューヨーク ロチェスター ステイト ストリート 3 4 3