

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-293560
(P2005-293560A)

(43) 公開日 平成17年10月20日(2005.10.20)

(51) Int. Cl.⁷
G06F 12/00

F I
G06F 12/00 533 J

テーマコード(参考)
5B082

審査請求 未請求 請求項の数 28 O L (全 20 頁)

(21) 出願番号 特願2005-56021 (P2005-56021)
(22) 出願日 平成17年3月1日(2005.3.1)
(31) 優先権主張番号 60/558,388
(32) 優先日 平成16年4月1日(2004.4.1)
(33) 優先権主張国 米国 (US)
(31) 優先権主張番号 10/932,474
(32) 優先日 平成16年9月1日(2004.9.1)
(33) 優先権主張国 米国 (US)

(71) 出願人 500046438
マイクロソフト コーポレーション
アメリカ合衆国 ワシントン州 9805
2-6399 レッドモンド ワン マイ
クロソフト ウェイ
(74) 代理人 100077481
弁理士 谷 義一
(74) 代理人 100088915
弁理士 阿部 和夫
(72) 発明者 ビベク ジェイ. ジャベリ
アメリカ合衆国 98052 ワシントン
州 レッドモンド ワン マイクロソフト
ウェイ マイクロソフト コーポレーシ
ョン内

最終頁に続く

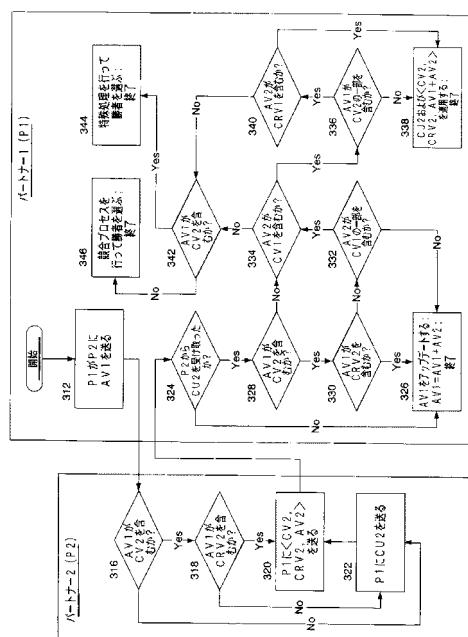
(54) 【発明の名称】 項目収束 (すなわち、データ収束) を施行するための競合解決を伝播するシステムおよび方法

(57) 【要約】

【課題】 手作業での、および自動的な競合解決動作中に競合を検出し、競合解決を適用するシステムおよび方法を提供する。

【解決手段】 いくつかの実施形態は、(a)最終的に、パートナーが、拡散した定常状態になり得る(すなわち、未解決の競合がない状態で、異なるデータを有する)場合の競合解決の非収束、および(b)パートナーの間で、競合解決のピンポンが終わらない状態(各競合解決のためのメタデータを絶え間なく無限にアップデートする、非効率な特定のソリューションの場合に起こり得る)両方を防ぐことを対象とする。本発明の具体的な特定の実施形態は、異なるパートナーが異なる競合解決を適用する場合でも、データの収束を保証する知識を維持する方法を対象とする。

【選択図】 図3



【特許請求の範囲】

【請求項 1】

同期コミュニティ内の少なくとも 2 つのピアコンピュータシステムの間で変更単位データを同期させる方法であって、

前記少なくとも 2 つのピアコンピュータシステムの間で前記変更単位データを同期させるステップと、

競合解決バージョンコンポーネントを含む、前記変更単位データに対応するメタデータコンポーネントを用いることによって、前記同期コミュニティ内での前記変更単位データの収束を確実にするステップとを含むことを特徴とする方法。

【請求項 2】

前記競合解決コンポーネントは、前記同期コミュニティを含む少なくとも 2 つのコンピュータシステムの中から、前記少なくとも 2 つのコンピュータシステムにおける前記変更単位の拡散状態を識別するために、同期中に使用されることを特徴とする請求項 1 に記載の方法。

10

【請求項 3】

前記変更単位の前記拡散状態を正し、前記同期コミュニティ全体に前記正された変更単位を伝播するステップをさらに含むことを特徴とする請求項 2 に記載の方法。

【請求項 4】

競合が起きたときにそれを解決する自動システムをさらに含むことを特徴とする請求項 3 に記載の方法。

20

【請求項 5】

同期コミュニティのためのピアツーピア同期システム、およびそのメンバーであるパートナーコンピュータシステムにおける変更単位データの拡散状態を解決する方法であって、前記データは、第 1 のコンピュータシステムにおける第 1 の変更単位、および第 2 のコンピュータシステムにおける第 2 の変更単位として具体化され、

第 1 の同期動作中に起こる競合解決を追跡するステップと、

後続同期動作中に、前記第 1 の変更単位と前記第 2 の変更単位の間の変更単位データの拡散状態を識別するステップと、

前記同期コミュニティにおける変更単位データを収束させるために、変更単位ソリューションを決定論的に選択するステップとを含むことを特徴とする方法。

30

【請求項 6】

前記第 1 の変更単位は、第 1 のバージョン番号をさらに含み、前記第 2 の変更単位は、第 2 のバージョン番号をさらに含み、前記同期コミュニティにおける変更単位データを収束させるための変更単位ソリューションを決定論的に選択する前記ステップは、前記第 1 のバージョン番号と前記第 2 のバージョン番号それぞれの比較から得られたより低いバージョン番号に基づく、前記第 1 の変更単位または前記第 2 の変更単位の選択を含むことを特徴とする請求項 5 に記載の方法。

【請求項 7】

第 1 のコンピュータシステム (P 1) と第 2 のコンピュータシステム (P 2) の間の変更単位データを同期させる方法であって、前記第 1 のコンピュータシステムは、前記変更単位の第 1 の複製 (C U 1)、および前記第 1 の複製に対応する第 1 のメタデータ (M 1) を含み、前記メタデータは、第 1 の変更バージョン (C V 1)、第 1 の競合解決バージョン (C R V 1)、および第 1 の認識ベクトル (A V 1) をさらに含み、前記第 2 のコンピュータシステムは、前記変更単位の第 2 の複製 (C U 2)、および前記第 2 の複製に対応する第 2 のメタデータ (M 2) を含み、前記メタデータは、第 2 の変更バージョン (C V 2)、第 2 の競合解決バージョン (C R V 2)、および第 2 の認識ベクトル (A V 2) をさらに含み、

40

P 1 から P 2 に A V 1 を送るステップと、

A V 1 が、C V 2 および C R V 2 を含むかどうか判定し、含む場合、P 2 から P 1 に C U 2 を送るステップと、

50

P 2 から P 1 に M 2 を送るステップと、

M 2 を M 1 と比較し、必要とされるどの同期アップデートも完了させ、起こったどの競合も特殊な処理状況も解決するステップと、

C U 1 および M 1 をアップデートし、競合が起きた場合、競合解決を反映するように C R V 1 および A V 1 をさらにアップデートし、そうすることによって、拡散状態が起こるのを防ぐステップとを含むことを特徴とする方法。

【請求項 8】

同期コミュニティ内の少なくとも 2 つのピアコンピュータシステムの間で変更単位データを同期させるシステムであって、

前記少なくとも 2 つのピアコンピュータシステムの間で前記変更単位データを同期させ

、競合解決バージョンコンポーネントを含む、前記変更単位データに対応するメタデータコンポーネントを用いることによって、前記同期コミュニティ内での前記変更単位データの収束を確実にする少なくとも 1 つのサブシステムを備えることを特徴とするシステム。

【請求項 9】

前記競合解決コンポーネントが、前記同期コミュニティを含む少なくとも 2 つのコンピュータシステムの中から、前記少なくとも 2 つのコンピュータシステムにおける前記変更単位の拡散状態を識別するために、同期中に使用されるようにするための少なくとも 1 つのサブシステムをさらに備えることを特徴とする請求項 8 に記載のシステム。

【請求項 10】

前記変更単位の前記拡散状態を正し、前記同期コミュニティ全体に前記正された変更単位を伝播する少なくとも 1 つのサブシステムをさらに備えることを特徴とする請求項 9 に記載のシステム。

【請求項 11】

競合が起きたときにそれを自動的に解決する少なくとも 1 つのサブシステムをさらに備えることを特徴とする請求項 10 に記載のシステム。

【請求項 12】

同期コミュニティのためのピアツーピア同期システム、およびそのメンバーであるパートナーコンピュータシステムにおける変更単位データの拡散状態を解決するシステムであって、前記データは、第 1 のコンピュータシステムにおける第 1 の変更単位、および第 2

のコンピュータシステムにおける第 2 の変更単位として具体化され、第 1 の同期動作中に起こる競合解決を追跡し、

後続同期動作中に、前記第 1 の変更単位と前記第 2 の変更単位の間の変更単位データの拡散状態を識別し、

前記同期コミュニティにおける変更単位データを収束させるために、変更単位ソリューションを決定論的に選択する少なくとも 1 つのサブシステムを備えることを特徴とするシステム。

【請求項 13】

前記第 1 の変更単位が第 1 のバージョン番号をさらに含み、前記第 2 の変更単位が第 2 のバージョン番号をさらに含むための少なくとも 1 つのサブシステムをさらに備え、前記同期コミュニティにおける変更単位データを収束させるための変更単位ソリューションを決定論的に選択する前記ステップが、前記第 1 のバージョン番号と前記第 2 のバージョン番号それぞれの比較から得られたより低いバージョン番号に基づく、前記第 1 の変更単位または前記第 2 の変更単位を選択を含むための少なくとも 1 つのサブシステムをさらに備えることを特徴とする請求項 12 に記載のシステム。

【請求項 14】

第 1 のコンピュータシステム (P 1) と第 2 のコンピュータシステム (P 2) の間の変更単位データを同期させるシステムであって、前記第 1 のコンピュータシステムは、前記変更単位の第 1 の複製 (C U 1)、および前記第 1 の複製に対応する第 1 のメタデータ (M 1) を含み、前記メタデータは、第 1 の変更バージョン (C V 1)、第 1 の競合解決バ

10

20

30

40

50

ージョン（C R V 1）、および第1の認識ベクトル（A V 1）をさらに含み、前記第2のコンピュータシステムは、前記変更単位の第2の複製（C U 2）、および前記第2の複製に対応する第2のメタデータ（M 2）を含み、前記メタデータは、第2の変更バージョン（C V 2）、第2の競合解決バージョン（C R V 2）、および第2の認識ベクトル（A V 2）をさらに含み、

P 1からP 2にA V 1を送り、

A V 1が、C V 2およびC R V 2を含むかどうか判定し、含む場合、P 2からP 1にC U 2を送り、

P 2からP 1にM 2を送り、

M 2をM 1と比較し、必要とされるどの同期アップデートも完了させ、起こったどの競合も特殊な処理状況も解決し、

C U 1およびM 1をアップデートし、競合が起きた場合、競合解決を反映するようにC R V 1およびA V 1をさらにアップデートし、そうすることによって、拡散状態が起こるのを防ぐ少なくとも1つのサブシステムを備えることを特徴とするシステム。

【請求項15】

同期コミュニティ内の少なくとも2つのピアコンピュータシステムの間で変更単位データを同期させるコンピュータ可読命令を含むコンピュータ可読媒体であって、

前記少なくとも2つのピアコンピュータシステムの間で前記変更単位データを同期させ

、競合解決バージョンコンポーネントを含む、前記変更単位データに対応するメタデータコンポーネントを用いることによって、前記同期コミュニティ内での前記変更単位データの収束を確実にする命令を含むことを特徴とするコンピュータ可読媒体。

【請求項16】

前記競合解決コンポーネントが、前記同期コミュニティを含む少なくとも2つのコンピュータシステムの中から、前記少なくとも2つのコンピュータシステムにおける前記変更単位の拡散状態を識別するために、同期中に使用されるようにするための命令をさらに含むことを特徴とする請求項15に記載のコンピュータ可読媒体。

【請求項17】

前記変更単位の前記拡散状態を正し、前記同期コミュニティ全体に前記正された変更単位を伝播する命令をさらに含むことを特徴とする請求項16に記載のコンピュータ可読媒体。

【請求項18】

競合が起きたときにそれを自動的に解決する命令をさらに含むことを特徴とする請求項17に記載のコンピュータ可読媒体。

【請求項19】

同期コミュニティのためのピアツーピア同期システム、およびそのメンバーであるパートナーコンピュータシステムにおける変更単位データの拡散状態を解決するコンピュータ可読命令を含むコンピュータ可読媒体であって、前記データは、第1のコンピュータシステムにおける第1の変更単位、および第2のコンピュータシステムにおける第2の変更単位として具体化され、

第1の同期動作中に起こる競合解決を追跡し、

後続同期動作中に、前記第1の変更単位と前記第2の変更単位の間の変更単位データの拡散状態を識別し、

前記同期コミュニティにおける変更単位データを収束させるために、変更単位ソリューションを決定論的に選択する命令を含むことを特徴とするコンピュータ可読媒体。

【請求項20】

前記第1の変更単位が第1のバージョン番号をさらに含み、前記第2の変更単位が第2のバージョン番号をさらに含むための命令をさらに含み、前記同期コミュニティにおける変更単位データを収束させるための変更単位ソリューションを決定論的に選択する前記ステップが、前記第1のバージョン番号と前記第2のバージョン番号それぞれの比較から得

10

20

30

40

50

られたより低いバージョン番号に基づく、前記第1の変更単位または前記第2の変更単位の選択を含むための命令をさらに含むことを特徴とする請求項19に記載のコンピュータ可読媒体。

【請求項21】

第1のコンピュータシステム(P1)と第2のコンピュータシステム(P2)の間の変更単位データを同期させるコンピュータ可読命令を含むコンピュータ可読媒体であって、前記第1のコンピュータシステムは、前記変更単位の第1の複製(CU1)、および前記第1の複製に対応する第1のメタデータ(M1)を含み、前記メタデータは、第1の変更バージョン(CV1)、第1の競合解決バージョン(CRV1)、および第1の認識ベクトル(AV1)をさらに含み、前記第2のコンピュータシステムは、前記変更単位の第2の複製(CU2)、および前記第2の複製に対応する第2のメタデータ(M2)を含み、前記メタデータは、第2の変更バージョン(CV2)、第2の競合解決バージョン(CRV2)、および第2の認識ベクトル(AV2)をさらに含み、

10

P1からP2にAV1を送り、

AV1が、CV2およびCRV2を含むかどうか判定し、含む場合、P2からP1にCU2を送り、

P2からP1にM2を送り、

M2をM1と比較し、必要とされるどの同期アップデートも完了させ、起こったどの競合も特殊な処理状況も解決し、

CU1およびM1をアップデートし、競合が起きた場合、競合解決を反映するようにCRV1およびAV1をさらにアップデートし、そうすることによって、拡散状態が起こるのを防ぐ命令を含むことを特徴とするコンピュータ可読媒体。

20

【請求項22】

同期コミュニティ内の少なくとも2つのピアコンピュータシステムの間で変更単位データを同期させるハードウェア制御装置であって、

前記少なくとも2つのピアコンピュータシステムの間で前記変更単位データを同期させる手段と、

競合解決バージョンコンポーネントを含む、前記変更単位データに対応するメタデータコンポーネントを用いることによって、前記同期コミュニティ内での前記変更単位データの収束を確実にする手段とを備えることを特徴とするハードウェア制御装置。

30

【請求項23】

前記競合解決コンポーネントが、前記同期コミュニティを含む少なくとも2つのコンピュータシステムの中から、前記少なくとも2つのコンピュータシステムにおける前記変更単位の拡散状態を識別するために、同期中に使用されるようにするための手段をさらに備えることを特徴とする請求項22に記載のハードウェア制御装置。

【請求項24】

前記変更単位の前記拡散状態を正し、前記同期コミュニティ全体に前記正された変更単位を伝播する手段をさらに備えることを特徴とする請求項23に記載のハードウェア制御装置。

【請求項25】

競合が起きたときにそれを自動的に解決する手段をさらに備えることを特徴とする請求項24に記載のハードウェア制御装置。

40

【請求項26】

同期コミュニティのためのピアツーピア同期システム、およびそのメンバーであるパートナーコンピュータシステムにおける変更単位データの拡散状態を解決するハードウェア制御装置であって、前記データは、第1のコンピュータシステムにおける第1の変更単位、および第2のコンピュータシステムにおける第2の変更単位として具体化され、

第1の同期動作中に起こる競合解決を追跡する手段と、

後続同期動作中に、前記第1の変更単位と前記第2の変更単位の間の変更単位データの拡散状態を識別する手段と、

50

前記同期コミュニティにおける変更単位データを収束させるために、変更単位ソリューションを決定論的に選択する手段とを備えることを特徴とするハードウェア制御装置。

【請求項 27】

前記第1の変更単位が第1のバージョン番号をさらに含み、前記第2の変更単位が第2のバージョン番号をさらに含むための手段をさらに備え、前記同期コミュニティにおける変更単位データを収束させるための変更単位ソリューションを決定論的に選択する前記ステップが、前記第1のバージョン番号と前記第2のバージョン番号それぞれの比較から得られたより低いバージョン番号に基づく、前記第1の変更単位または前記第2の変更単位を選択を含むための手段をさらに備えることを特徴とする請求項26に記載のハードウェア制御装置。

10

【請求項 28】

第1のコンピュータシステム(P1)と第2のコンピュータシステム(P2)の間の変更単位データを同期させるハードウェア制御装置であって、前記第1のコンピュータシステムは、前記変更単位の第1の複製(CU1)、および前記第1の複製に対応する第1のメタデータ(M1)を含み、前記メタデータは、第1の変更バージョン(CV1)、第1の競合解決バージョン(CRV1)、および第1の認識ベクトル(AV1)をさらに含み、前記第2のコンピュータシステムは、前記変更単位の第2の複製(CU2)、および前記第2の複製に対応する第2のメタデータ(M2)を含み、前記メタデータは、第2の変更バージョン(CV2)、第2の競合解決バージョン(CRV2)、および第2の認識ベクトル(AV2)をさらに含み、

20

P1からP2にAV1を送る手段と、

AV1が、CV2およびCRV2を含むかどうか判定し、含む場合、P2からP1にCU2を送る手段と、

P2からP1にM2を送る手段と、

M2をM1と比較し、必要とされるどの同期アップデートも完了させ、起こったどの競合も特殊な処理状況も解決する手段と、

CU1およびM1をアップデートし、競合が起きた場合、競合解決を反映するようにCRV1およびAV1をさらにアップデートし、そうすることによって、拡散状態が起こるのを防ぐ手段とを備えることを特徴とするハードウェア制御装置。

【発明の詳細な説明】

30

【技術分野】

【0001】

本発明は概して、コンピュータシステムに渡るデータ同期に関し、より詳細には、ピアツーピア同期システムにおける競合の解決、および項目収束(item convergence)(すなわち、データ収束(data convergence))(つまり、同期用に用いられる変更単位に基づくデータ収束)を確実にするやり方での競合解決の伝播(propagating)に関する。

【背景技術】

【0002】

同期中に起こる競合は、ピアツーピア同期環境内の異なる同期パートナーによって異なるやり方で解決され得る。このようなピアツーピア同期環境の1つおよびその基底にあるハードウェア/ソフトウェアインターフェイスシステムは、(オペレーティングシステムなどだがそれに限定されない)ハードウェア/ソフトウェアインターフェイスシステムであり、このシステムでは、他のものの中でも、格納可能な情報(項目)からなる自立単位が、同期のための基本データ変更要素を構成するいくつかのコンポーネント変更単位(CU)を含む。各CUは、ピアコンピュータシステム上でどのような変更がアップデートされる必要があるか、ならびに他のピアコンピュータシステムから受け取ったどのような変更が適用されるべきかを同期システムが追跡することを可能にするメタデータを関連づけている。

40

【0003】

50

同期のための変更単位およびメタデータ（または類似の構造）を用いるシステム用の基本的な同期の例示的な一形式は概して、2つのピアコンピュータシステム上に存在する項目用の単一の変更単位に関して説明されることができる。この単一変更単位に関して、第1のピア、すなわちパートナーA（またはより簡単に「A」）が、第2のピア、すなわちパートナーB（またはより簡単に「B」）に対して、変更単位の複製（CUa）に関する、Aの現在のメタデータ（Ma）をBに送信することによって、変更を要求する。パートナーBは、（CUaに対応する）Maを受信すると、このメタデータを、同じ変更単位の複製（Cub）に関するB自体のメタデータ（Mb）と比較する。次いで、Mbが、Maで反映されていない、Cubに対するいずれかの変更を反映している場合、Bは、AにCubおよびMb両方を送信する。それ以外の場合、Bは、AにMbのみを送信する（あるいは、「アップデートなし」というインジケータをAに送信する）。というのは、CUaは、Cubに対して行われたすべてのアップデートを既に反映しており、したがって、BからAにCubをコピーする必要はないからである。

10

20

30

40

50

【0004】

同期要求に対するBの応答に基づいて、Aが、CubではなくMbのみ（あるいは、「アップデートなし」インジケータ）をBから受信した場合、Aは、CUaが、Cubと同じであるか、それともCubよりもアップデートされているかを知る。しかし、Aが、MbおよびCub両方をBから受信した場合、Aは、MaをMbと比較することによって、Cubが、CUaよりもアップデートされているかどうか、またはCUaおよびCubが、競合する変更を有しているかどうか、判定しなければならない。CubがCUaよりもアップデートされている場合、Aは、CubでCUaをアップデートし、また、MbでMaをアップデートする。反対に、CUaおよびCubが、競合する変更を有している場合、Aは、そのローカルな競合解決手順によってその競合を解決する。たとえば、この場合のAの競合解決手順が、リモートなソリューションよりもローカルなソリューションを選ぶものである場合、Aは、（Cubの変更を拒否して）CUaをもち続け、MaおよびMbに既に反映されているすべてのアップデート（MaおよびMbの合併）を含むようにMaをアップデートする。その後、Bが同じ手順を用いてAと同期する（かつ、追加の変更が介入しないと仮定する）と、Bは、CUaがそのCubよりもアップデートされていると結論づけ、CubをCUaでアップデートする。

【0005】

同期に対する、この基本的な変更単位およびメタデータ（CUAM）手法は、2つのピアからなる同期コミュニティにとって容易であり有効である。しかし、3つ以上のピアの場合、この容易な手法は、2つのシステムが異なる競合解決ソリューションに到達するが同一のメタデータをもち得るようなデータの拡散（*divergence*）を生じる可能性があるため、この論理的な矛盾が、ピアコミュニティ内で認識されることも、適切に改められることもできない。（本明細書において、この現象の詳細な例が後で提示される。）

【発明の開示】**【発明が解決しようとする課題】****【0006】**

すべてのピア上のデータを収束させるために、当該分野において必要とされるのは、すべての関連ピアに及ぶ1つの競合解決法に一致して収束させるために、同期コミュニティ内のピアの間で特定の競合解決を正しく識別し伝播させ、同じ競合に対する異なる解決法を調停する、様々なタイプの同期システムのための同期方法である。

【課題を解決するための手段】**【0007】**

本発明の様々な実施形態は、手作業での、および自動的な競合解決動作中に競合を検出し、競合解決を適用するシステムおよび方法を対象とする。いくつかの実施形態は、（a）最終的に、パートナーが、拡散したままの定常状態になり得る（すなわち、未解決の競合がない状態で、異なるデータを有する）場合の競合解決の非収束、および（b）パート

ナーの間で、競合解決のピンポン (ping-pong) が終わらない状態 (各競合解決のためのメタデータを絶え間なく無限にアップデートする、非効率な特定のソリューションの場合に起こり得る) 両方を防ぐことを対象とする。本発明の具体的な特定の実施形態は、異なるパートナーが異なる競合解決を適用する場合でも、データの収束を保証する知識を維持する方法を対象とする。

【0008】

上述の要約、ならびに以下の好ましい実施形態の詳細な説明は、添付の図面と併せ読むとよりよく理解されよう。本発明を説明するために、本発明の例示的な構成が図面に示される。ただし、本発明は、開示される特定の方法および手段に限定されない。

【発明を実施するための最良の形態】

10

【0009】

本主題が、法的要件を満たすために明確に説明される。ただし、説明自体は、本特許の範囲を限定することを意図していない。そうではなく、本発明者は、特許請求の対象が、本文書において説明されるものと類似した、異なるステップまたはステップの組合せを含むように、現在または将来の他の技術とともに、他の方法でも実施されることを企図している。さらに、「ステップ」という用語は、本明細書において、利用される方法の異なる要素を意味するのに用いられ得るが、この用語は、個々のステップの順序が明確に記述されない限り、本明細書において開示される様々なステップの間のいかなる具体的な順序も含意すると解釈されるべきでない。

【0010】

20

(コンピュータ環境)

本発明の数多くの実施形態が、コンピュータ上で実行されることができる。図1および以下の説明は、本発明が実装され得る適切な計算機環境の簡潔かつ全体的な説明を提供することを意図している。そうすることが必要なわけではないが、クライアントワークステーションやサーバなどのコンピュータによって実行されるプログラムモジュールなどのコンピュータ実行可能命令という一般的なコンテキストにおいて本発明が説明される。概して、プログラムモジュールは、特定のタスクを実施し、または特定の抽象データ型を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含む。さらに、本発明は、ハンドヘルドデバイス、マルチプロセッサシステム、マイクロプロセッサベースの家電製品またはプログラム可能な家電製品、ネットワークPC、ミニコンピュータ、メインフレームコンピュータなど他のコンピュータシステム構成とともに実施され得ることを当業者は理解するであろう。本発明は、通信ネットワークを介してリンクされるリモート処理デバイスによってタスクが実施される分散型計算機環境でも実施されることができる。分散型計算機環境では、プログラムモジュールは、ローカルおよびリモートメモリ記憶装置両方に置かれ得る。

30

【0011】

図1に示すように、例示的な汎用計算機システムは、従来のパーソナルコンピュータ20などを含み、このコンピュータは、処理ユニット21、システムメモリ22、およびシステムメモリなど様々なシステムコンポーネントを処理ユニット21に結合するシステムバス23を含む。システムバス23は、様々なバスアーキテクチャのいずれかを使用するメモリバスまたはメモリコントローラ、周辺バス、およびローカルバスなどいくつかのタイプのバス構造のいずれでもよい。システムメモリは、ROM (読出し専用メモリ) 24 およびRAM (ランダムアクセスメモリ) 25を含む。BIOS 26 (基本入出力システム) は、たとえば起動中にパーソナルコンピュータ20内部の要素間の情報の転送を助ける基本ルーチンを含み、ROM 24に格納される。パーソナルコンピュータ20は、図示していないハードディスクからの読出しおよびそこへの書込みを行うハードディスクドライブ27、着脱式磁気ディスク29からの読出しまたはそこへの書込みを行う磁気ディスクドライブ28、および、たとえばCD-ROMや他の光学媒体などの着脱式光ディスク31からの読出しまたはそこへの書込みを行う光ディスクドライブ30をさらに含むことができる。ハードディスクドライブ27、磁気ディスクドライブ28、および光ディスク

40

50

ドライブ30は、それぞれハードディスクドライブインターフェイス32、磁気ディスクドライブインターフェイス33、および光ドライブインターフェイス34によって、システムバス23に接続される。こうしたドライブおよびそれに関連するコンピュータ可読媒体は、コンピュータ可読命令、データ構造、プログラムモジュール、および他のデータの揮発性記憶をパーソナルコンピュータ20にもたらず。本明細書で説明される例示的な環境は、ハードディスク、着脱式磁気ディスク29、および着脱式光ディスク31を使用するが、コンピュータによってアクセス可能なデータを格納することができる、コンピュータ可読な他のタイプの媒体も、例示的な動作環境において使われ得ることが、当業者には理解されるべきである。このような他のタイプの媒体として、磁気カセット、フラッシュメモリカード、デジタルビデオディスク、ベルヌーイカートリッジ、RAM（ランダムアクセスメモリ）、ROM（読み出し専用メモリ）などがある。 10

【0012】

オペレーティングシステム35、1つまたは複数のアプリケーションプログラム36、他のプログラムモジュール37、およびプログラムデータ38などいくつかのプログラムモジュールが、ハードディスク、磁気ディスク29、光ディスク31、ROM24、またはRAM25に格納されることができる。ユーザは、キーボード40などの入力デバイス、およびポインティングデバイス42を介して、コマンドおよび情報をパーソナルコンピュータ20に入力することができる。他の入力デバイス（図示せず）は、マイクロホン、ジョイスティック、ゲーム用パッド、衛星パラボラアンテナ、スキャナなどを含み得る。こうしたおよび他の入力デバイスはしばしば、システムバスに結合されるシリアルポート 20
インターフェイス46を介して処理ユニット21に接続されるが、他のインターフェイス、たとえば並列ポート、ゲームポート、USB（ユニバーサルシリアルバス）によって接続されることもできる。モニター47または他のタイプの表示デバイスも、ビデオアダプタ48などのインターフェイスを介してシステムバス23に接続される。モニター47に加え、パーソナルコンピュータは通常、他の周辺出力デバイス（図示せず）、たとえばスピーカおよびプリンタを含む。図1の例示的なシステムは、ホストアダプタ55、SCSI（小型計算機システムインターフェイス）バス56、およびSCSIバス56に接続された外部記憶装置62も含む。

【0013】

パーソナルコンピュータ20は、リモートコンピュータ49など1つまたは複数のリモートコンピュータへの論理接続を使用してネットワーク接続された環境において動作することができる。リモートコンピュータ49は、別のパーソナルコンピュータ、サーバ、ルータ、ネットワークPC、ピアデバイスまたは他の共通ネットワークノードでよく、通常、パーソナルコンピュータ20に関連して上述した要素の多くまたはすべてを含むが、図1にはメモリ記憶装置50のみが示されている。図1に示した論理接続は、LAN（ローカルエリアネットワーク）51およびWAN（ワイドエリアネットワーク）52を含む。このようなネットワーク環境は、会社、企業規模のコンピュータネットワーク、イントラネットおよびインターネットにおいてよく見られる。 30

【0014】

LANネットワーク環境において使われる場合、パーソナルコンピュータ20は、ネットワークインターフェイスまたはアダプタ53を介してLAN51に接続される。WANネットワーク環境において使われる場合、パーソナルコンピュータ20は通常、モデム54、または、たとえばインターネットなどのワイドエリアネットワーク52を介した通信を確立する他の手段を含む。モデム54は、内部にあっても外部にあってもよく、シリアルポートインターフェイス46を介してシステムバス23に接続される。ネットワーク接続された環境では、パーソナルコンピュータ20に関連して図示したプログラムモジュールまたはその一部は、リモートメモリ記憶装置に格納されることができる。図示したネットワーク接続は例示的なものであり、コンピュータ間の通信リンクを確立する他の手段も使われ得ることが理解されよう。 40

【0015】

本発明の数多くの実施形態は、コンピュータによるシステムに特に適合すると想像されるが、本文書のどの部分も、本発明をこのような実施形態に限定することを意図していない。反対に、本明細書で使用する「コンピュータシステム」という用語は、デバイスが本質的に電子、機械、論理、または仮想であるかに関わらず、情報を格納し処理することができ、かつ/またはこのようなデバイス自体の動作または実行を制御するのに、格納された情報を使うことができるあらゆるデバイスを包含することを意図する。

【0016】

(基本同期スキーマおよび項目拡散)

本発明の特定の実施形態、たとえば項目ベースのシステムにとって、「変更単位」とは、本質的に、常に一単位として同期される1組のプロパティである。たとえば、連絡先項目の名前および名字というプロパティは合わせて、1個の変更単位を構成することができる。さらに、「バージョン」が、各変更単位に関連づけられて、アップデートが行われたときに対応するタイムスタンプとともに、その変更単位へのある特定の変更を最後に行ったパートナーを表す。したがって、表記を簡単にするため、本明細書で使用するパートナーは文字で表され、タイムスタンプは、整数を増やすことによって表され、その結果、パートナー(たとえば「A」)およびタイムスタンプ(たとえば「1」)を単に対にすることによって、各変更単位はそれ自体を、各パートナーによって特定の時間に行われた最新の変更を反映する一意の識別番号、たとえば「A1」に関連づけている。

【0017】

本明細書において上で説明されたような(データの非収束を生じ得る)基本同期スキーマの場合、各変更単位に関するメタデータは、2つのコンポーネントを含む。すなわち、変更単位の現在の「アップデート」状況を示す現在のバージョン(c v)、ならびにピアコミュニティ内の各パートナー(あるいは少なくとも、何らかの変更を行ったパートナー)に関する、前記変更単位に起こったアップデートに対応する認識ベクトル(a v)(別名、「バージョン配列」)である。変更単位のメタデータ用の一般的な形式は、<変更バージョン, 認識ベクトル>、またはより簡単には、<c v, [a v]>である。この情報はともに、たとえば、<A13, [A13, B11]>というメタデータ(M a)をもつ、パートナーAにおける変更単位(C U a)が、<B11, [A12, B11]>というメタデータ(M b)をもつ、パートナーBにおける変更単位(C U b)よりもアップデートされていることを容易に判定するのに用いられ得る。というのは、各バージョン番号のタイムスタンプ部分が、時間とともに徐々に増加することを考えると、M aは、M bに反映されていない変更(すなわちA13)を含むからである。

【0018】

図2は、最終的に競合解決の非収束(パートナーが拡散定常状態になり得る)を生じる、(3つのパートナーによる同期コミュニティのための)3つのピアによる同期動作を示す、テーブルに基づくタイムフロー図である。テーブル「T」の第1列は、イベントの相対的な時系列に対応し、第2、第3、および第4の列は、同期コミュニティ内の3つのパートナーシステムA、B、およびCそれぞれにおけるアクティビティに対応する。便宜上、図に示されるように、各パートナーA、B、およびCにおける変更単位の複製に対する時間0での初期状態は、完全に同期した状態で始まり、この状態において、各パートナーA、B、およびCに関する変更単位の複製は、対応する各列のデータによって個別に表されている。

【0019】

時間1で、Aは、変更単位の複製(これ以降、C U a)中のデータをアップデートし、そのアップデートを、新しい値(A11)でc vを増分しアップデートされた状態を反映するようにa vをアップデートすることによって反映させる。時間2で、Bも、変更単位の複製(これ以降、C U b)中のデータを独自にアップデートし、c vおよびa vを適切にアップデートする。

【0020】

時間3で、CがBと同期して、第1の同期動作が起こる。動作の際、本明細書において

10

20

30

40

50

上で説明した論理を用いて、Cは、そのメタデータ(Mc)をBに送り、Bは、応答として、その変更単位のコピー(CUb)およびそれに関連するメタデータ(Mb)両方を、さらに処理させるためにCに送る。Cは、CUbおよびMbを受け取ると、AVcのすべてがAVbに反映されているのにBの認識ベクトル(AVb)中のバージョン「B11」がCの認識ベクトル(AVc)に反映されていないので、CUbおよびMbがCUcおよびMcより現在のものに近いことを突き止め、したがってこのことは、CUbが、CUcに関して「最終アップデート済み」である(より最新のものである)ことを意味する。その結果、Cは、CUbのデータでCUcをアップデートし、Mbのメタデータで、関連するメタデータMcをさらにアップデートする。

【0021】

10

時間4で、Bは次いで、パートナーAと同期し、Aの変更単位(CUa)およびメタデータ(Ma)を(ここでは便宜上説明が省略されるが、後で論理的に説明されるやり方によって)受け取った後、Bは、BおよびA両方がそれぞれの変更単位の複製CUbおよびCUaに対して個別に変更を行っているので、競合が存在することを突き止める。Bは次いで、その競合解決プロセス(この例では、「リモートを選択」、つまり、そのローカルバージョンよりもリモートバージョンを選ぶことである)を利用して、CUbよりもCUaを選び(CUaのデータをCUbにコピーし)、その現在のバージョン(CVb)をA11に変更しBおよびA両方の認識ベクトル(AVbおよびAVa)をマージさせることによってこの解決法をメタデータに反映させる。

【0022】

20

時間5で、Cは次いで、パートナーAと同期し、Aの変更単位(CUa)およびメタデータ(Ma)を受け取った後、Cもやはり、CおよびA両方がそれぞれの変更単位の複製CUcおよびCUaに対して個別に変更を行っているので、競合が存在することを突き止める(しかし実際には、CUcの変更は、時間3でCがBと同期した結果によるが、この違いは重要でない)。競合により、Cは次いで、その競合解決プロセス(この例では、「ローカルを選択」、つまり、リモートバージョンよりもCのローカルバージョンを選ぶことである)を利用して、CUaよりもCUcを選び(変更単位データをさらにコピーする必要はない)、その現在のバージョン(CVc)をB11としてもち続けCおよびA両方の認識ベクトル(AVcおよびAVa)をマージさせることによってこの解決法をメタデータに反映させる。

30

【0023】

時間6で、Aは次いで、Bと同期し、交換したメタデータ(AがBにMaを送り、次いでBが、CUaとともにMbをAに戻す)に基づいて、Aは、(MbはMaのすべてを反映し、バージョンB11も反映しているが、Maは反映していないので)CUbおよびMbがCUaおよびMaよりもアップデートされていると認識し、したがってAは、そのCUaおよびMaにCUbおよびMbを適用する(ただし、この場合、CUbおよびCUaは実際には同じもの、つまりA11であり、したがって代替実施形態では、Bは、この事実を認識し、CUbを送らなくてもよく、Aも、CUaにCUbにコピーしなくてもよいことが留意されるべきである)。

【0024】

40

最後に、時間7で、Aは次いで、Cと同期するが、MaがMcと同一なので、Cは、AにCUcを送らず、したがって、AおよびCそれぞれにおけるCUaおよびCUcのデータは拡散しており、現在の同期システムによって認識されることができず調停されにくい状態にある。

【0025】

(競合解決の伝播)

本発明のいくつかの実施形態は、手作業での、および自動的な競合解決動作中に競合を検出し、競合解決を適用するシステムおよび方法を対象とする。というのは、たとえば、バージョンA1とB2の間に、A1のためにはパートナーXによって、かつB2のためにはパートナーYによって解決される競合が存在する場合などに、異なるピア(「パートナ

50

ー」)は、同じ競合を異なるやり方で解決し得るからである。

【0026】

本発明の様々な実施形態は、(未解決の競合がない場合に異なるデータをもつことなどによって、パートナーが拡散した定常状態になり得る)競合解決の非収束、およびパートナーの間の競合解決の無限ピンポンを防止する1つまたは複数の方法を利用する。

【0027】

いくつかの本発明の実施形態の場合、同期用メタデータは、変更バージョン(c v)および認識ベクトル(a v)を含むだけでなく、拡散状態につながり得る競合解決を特に追跡する競合解決バージョン(c r v)をさらに含む。変更単位のメタデータへのc r vの追加に基づいて、同期プロセスは、本明細書において説明されるように、メタデータ中のこの新しい要素を使うように修正される。便宜上、c v、c r v、およびa vを含む前記同期メタデータは、変更単位の各複製(たとえば、< A 1 0 , B 1 2 , [A 1 1 , B 1 2 , C 1 0] > に対して、< c v , c r v , [a v] > の形で表されることができる。

10

【0028】

図3は、本発明のいくつかの実施形態で用いられる、メタデータへのc r vの追加に基づく改良された同期プロセス、および本明細書において説明されるようにメタデータを修正する他のプロセスを示すプロセスフロー図である。図3において、プロセスは、ステップ312で始まり、第1のパートナーP1(別のピアとの同期を捜し求めているパートナー)が、そのローカルな変更単位の複製(C U 1)の認識ベクトル(A V 1)をP2に送ることによって、第2のパートナーP2との同期動作を開始する。P2は、P1からA V 1を受け取った後、ステップ316で、A V 1が、それ自体の、ローカルであるが対応する変更単位の複製(C U 2)の現在のバージョン(C V 2)を含むかどうか最初に判定し、含む場合、ステップ318で、P2は、A V 1がC U 2(存在する場合)の競合解決バージョン(C R V 2)も含むかどうかも判定する。A V 1が、こうした条件両方を満たす場合、次いで、ステップ320で、P2は、C U 2自体を送らずに、C V 2、C R V 2、およびA V 2を含む、P2がもつC U 2のメタデータ(M2)を直ちにP1に送る。A V 1がC R V 2を含まない場合、ステップ322で、P2は、P1にC U 2を送り、かつ、ステップ320で、M2を送る。

20

【0029】

P1に戻ると、ステップ324で、P1は最初に、P2からC U 2を受け取ったかどうか判定する。受け取っていない場合、次いで、P1は、その変更単位の複製(C U 1)がP1とP2の間に関して最も新しいアップデートを反映していると結論づけることができ、したがってステップ326で、P1は、(必要な場合)A V 2でA V 1を単にアップデートし、そうすることによって、A V 1は、(まだ反映していない場合)A V 2中のすべてのアップデートバージョンを反映し、同期プロセスは終わる。

30

【0030】

逆に、ステップ324で、P1がP2からC U 2を実際に受け取っていた場合、ステップ328で、P1は、C V 2によって反映されるバージョンをA V 1が含むかどうか最初に判定しなければならず、含む場合、ステップ330で、C R V 2によって反映されるバージョンをA V 1が含むかどうかも判定しなければならない。含む場合、次いで、P1は、やはりステップ326に進んで、(必要な場合)A V 1をアップデートし、同期プロセスは終わる。反対に、A V 1が、ステップ328で、C V 2によって反映されるバージョンを含むが、ステップ330で、C R V 2によって反映されるバージョンを含まないとP1が判定した場合、ステップ332で、P1は、C V 1によって反映されるバージョンの少なくとも一部をA V 2が含むかどうか判定する。「一部」という言葉は、特定の代替実施形態の場合、C V 1が1組の複数のc v値であり得ることを期待しているが、この例のように1個のc v値の場合は、問題となっているc v値を含むことと同じであり(したがって、1個のc vの組に対してステップ334がある程度重複する)、したがって、1個のc vの組においてステップ334に自動的に進むことになるが、これは全体を通して用いられるケースであり、A V 2がC V 1の一部を含まない場合、ステップ326で、P1

40

50

は、(必要な場合)AV2でAV1をやはりアップデートし、同期プロセスは終わる。

【0031】

対照的に、ステップ328で、AV1がCV2を含まない場合、またはステップ332で、AV2がCV1の一部を実際を含む場合、ステップ334で、P1は次に、AV2がCV1を含むかどうか判定する。含む場合、ステップ336で、P1は、CV2によって反映されるバージョンの少なくとも一部をAV1が含むかどうか突き止め、含まない場合、ステップ338で、P1は、CU1にCU2を適用し(またはCU2からCU1にデータをコピーし)、(CV1、CRV1、AV1を含む)対応するメタデータM1を、<CV2, CRV2, AV1+AV2>という値(AV1+AV2は、各認識ベクトルの最新バージョンの合併であり、たとえば、AV1=[A11, B10, C10]、かつAV2=[A10, B11, C12]の場合、AV1+AV2=[A11, B11, C12]となる)に設定し、同期プロセスは終わる。しかし、ステップ336で、AV1がCV2の一部を実際を含むとP1が判定した場合、ステップ340で、P1は、AV2がCRV1を含むかどうかも判定し、含む場合、P1は、やはりステップ338に進んで、上で説明したようにCU2を適用し、同期プロセスは終わる。

10

【0032】

別法では、ステップ334で、AV2がCV1を含まないとP1が判定した場合、またはステップ340で、AV2がCRV1を含まないとP1が判定した場合、いずれの場合も、P1は、ステップ342で、AV1がCV2を含むかどうか突き止め、含む場合、ステップ344で、P1は、「特殊な処理」を利用して、ソリューション(競合が存在しないので)を選び、その後、同期プロセスを終える。AV1がCV2を含まない場合、ステップ346で、P1は、「競合プロセス」(または競合解決プロセス)を開始して、競合を解決し、2組のデータの間の勝者を選び、同期プロセスを終える。

20

【0033】

(高度な同期スキーマおよび項目収束(すなわち、データ収束))

図3で説明し図示した上記の方法を用いることによって、(基本同期スキーマでの場合のように)異なる変更単位値をもつが同じメタデータをもつ2つの変更単位の複製が認識されず、拡散状態のまま放置されるという状況が回避され、図4は、図2に示されるのと同様のシナリオを使用するが図3に示される本発明の方法を適用することによって、この点を説明する。

30

【0034】

より具体的には、図4は、最終的に競合解決の収束を生じる、3つのピアによる同期動作を示す、テーブルを基にしたタイムフロー図である。さらに、図5は、図3に示すプロセスの結果生じる特定の最終状態に対する具体的な処理および競合プロセスに関する、結果として生じるメタデータ、ならびに(新規ローカル変更は同期動作の直接の一部ではないので、比較のためだが、時間1および2において後で説明されるように、ローカルなアップデートに対してメタデータがどのように変わるかの理解のために)新規ローカル変更に関するメタデータを示すテーブルである。

【0035】

再度図4を参照すると、テーブル「T」の第1列は今回も、イベントの相対的な時系列に対応し、第2、第3、および第4の列は、同期コミュニティ内の3つのパートナーシステムA、B、およびCそれぞれにおけるアクティビティに対応する。便宜上、図に示されるように、各パートナーA、B、およびCにおける変更単位の複製に対する時間0での初期状態は、完全に同期した状態で始まり、この状態において、各パートナーA、B、およびCに対する変更単位の複製は、対応する各列のデータによって個別に表されている。

40

【0036】

時間1で、Aは、変更単位の複製(これ以降、CUa)中のデータをアップデートし、そのアップデートを、新しい値(A11)でその現在のバージョン(cv)を増分しアップデートされた状態を反映するように認識ベクトル(av)をアップデートすることによって反映させ、競合解決バージョン(crv)には、(競合が解決されなかったので)ア

50

アップデートが行われない。時間2で、Bも、変更単位の複製（これ以降、C U b）中のデータを独自にアップデートし、対応するそのメタデータ（M b）コンポーネントを図に示すように適切にアップデートする。（これ以降、便宜上、各パートナーc v、c r v、およびa vは、パートナーごとに、たとえば、ピアAのメタデータコンポーネントに対してはC V a、C R V a、およびA V aと示され、パートナーBおよびCにおける対応するメタデータの場合も同様の名称となる。同じことが集合メタデータにも適用され、たとえば、システムA用はM a、変更単位に関しては、たとえばシステムA用はC U a、などとなる。）

時間3で、CがBと同期して、第1の同期動作が起こる。図3のフローチャートで示される方法を用いて、Cは、BにA V cを送り、Bは、CにC U bおよびM b両方を返す。M bおよびM cを分析することにより、Cは、C U bがC U cよりもアップデートされていると判定することができ、最終的に、C U bを適用し（すなわち、C U cにC U bをコピーし）、そのメタデータM cコンポーネントC V c、C R V c、およびA V cを、 $\langle C V b, C R V b, A V c + A V b \rangle$ に等しい値にアップデートする（ここで、 $A V c + A V b$ は、本明細書において上で説明したように、A V cとA V bの合併である）。より具体的には、図3で示されるプロセスに関して、論理プロセスは、ステップ312、316、322、320、324、328、334、336、340、および338をこの順序でトラバースする。（結果は、本明細書において上で説明した基本同期システムから予期されるであろう結果と同様であることに留意されたい。）

再度図4を参照すると、時間4で、Bは次いで、パートナーAと同期し、Aの変更単位（C U a）およびメタデータ（M a）を（やはり図3のフローチャートで示される方法に従って）受け取った後、Bは、BおよびA両方がそれぞれの変更単位の複製C U bおよびC U aに対して個別に変更を行っているので、競合が存在することを突き止め、その競合解決プロセスを利用する。より具体的には、図3で示されるプロセスに関して、論理プロセスは、ステップ312、316、322、320、324、328、334、342、および346をこの順序でトラバースする。競合を解決するために、Bは、その競合解決手順に従って、ローカルなアップデートよりもリモートなアップデートを選び、その結果得られるM bのメタデータの値は、図5に示されるように、 $\langle C V a, C R V n e w, A V a + A V b + C R V n e w \rangle$ であり、ここで、C R V n e wの「new」は、パートナーB用の次の増分アップデートバージョンであり（たとえば、この例では、時間2での変更に対する最終アップデートバージョンはB11であったので、この競合解決用の次の増分アップデートバージョンはB12である）、C R V bおよびA V b両方（ただしC V bは除く）が、時間4でのM bの最終状態に対して、図4に示すようにこの新しい値を反映する。

【0037】

時間5で、Cは次いで、パートナーAと同期し、Aの変更単位（C U a）およびメタデータ（M a）を受け取った後、Cもやはり、（312、316、322、320、324、328、334、342、および346をこの順序でトラバースする図3のプロセスによって）CおよびA両方がそれぞれの変更単位の複製C U cおよびC U aに対して個別に変更を行っているので、競合が存在することを突き止める（しかし実際には、C U cの変更は、時間3でCがBと前に同期していた結果によるが、この違いは重要でない）。競合により、Cは次いで、その競合解決プロセス（この例では、リモートなアップデートよりもローカルなアップデートを選択することである）を利用して、C U aよりもC U cを選び、この選択は、図5に示されるように、 $\langle C V c, C R V n e w, A V a + A V c + C R V n e w \rangle$ としてメタデータに反映され、ここで、やはりC R V n e wは、パートナーC用の次の増分アップデートバージョン（たとえば、この例ではC11）であり、時間5でのM cの最終状態に関して図4に示すように、得られたC R V cおよび得られたA V c両方（ただしC V cを除く）によって反映されている。

【0038】

時間6で、Aは次いで、Bと同期し、交換したメタデータ（AがBにM aを送り、次い

10

20

30

40

50

でBが、CUaとともにMbをAに戻す)に基づいて、Aは、(MbはMaのすべてを反映し、バージョンB11も反映しているが、Maは反映していないので)CubおよびMbがCUaおよびMaよりもアップデートされていると認識し、したがってAは、図に示されるようにそのCUaおよびMaにCubおよびMbの値を適用する(ただし、この場合、CubおよびCUaは実際には同じもの、つまりA11であり、したがって代替実施形態では、Bは、この事実を認識し、Cubを送らなくてもよく、Aも、CUaにCubにコピーしなくてもよいことが留意されるべきである)。図3に関して、この同期のための具体的なトラバースは、312、316、318、322、320、324、328、334、336、340、および338の順序である。

【0039】

再度図4に戻ると、時間7で、Aは次いで、Cと同期するが、競合が起きたときに、本明細書において上で説明した、競合解決バージョン(crv)と認識バージョン(av)へのcrvの論理的な組み込みとを使用したことにより、Maは、もはやMcと同一ではない(標準同期スキームを用いる場合に起こる)ので、Cは、CucおよびMcを、(図3で説明されたプロセスに従って)さらに処理させるためにAに実際に送り、Aは最終的に、(図3のプロセスを、ステップ312、316、318、322、320、324、328、330、332、334、336、340、342、および344の順序でトラバースすることによって)特殊な処理をとる。この特殊な処理は、具体的には項目収束(すなわち、データ収束)の施行を意図しており、次いで、同期コミュニティ内で、他のアップデートよりもある1つのアップデート(たとえば、この例では、アップデート番号が最も小さいアップデートであり、このアップデートはやはり、一意のコンピュータシステムid、および前記アップデートに対するタイムスタンプの組合せである)が全域において選択されるようにする決定論的なソリューションに到達する方法を利用する。このソリューションは次いで、図5に示す、アップデートされたメタデータ<CVdet, CRVdet, AV1 + AV2>と結合され、あるいは、「A11」の対応する値が「B11」の対応する値より小さく、したがってA11が決定されたソリューションである場合、メタデータMaは、図4に時間7で示す値をもつ<Cva, CRVa, AVa + AVc>となる。

【0040】

図4の時間8および時間9で、次いで、時間7でのこの特殊な処理によるソリューションがどのようにして同期コミュニティに伝播し、最終的に項目収束(すなわち、データ収束)(各システムにおける変更単位はやはり同一である)をもたらすかが容易に明らかになる。時間8で、方法の(図3に示されるプロセスのステップ312、316、318、320、324、および326による)論理プロセスのトラバースは、MbのAVbのアップデートを生じる。時間9で、方法の(図3に示されるプロセスのステップ312、316、318、322、320、324、328、330、332、334、336、340、および338による)論理プロセスのトラバースは、CucおよびMcへのCubおよびMbの適用を生じ、ここで、得られるメタデータMcは、<Cub, CRVb, AVb + AVc>に等しい値を含む。

【0041】

(項目収束(すなわち、データ収束)のための代替方法)

本発明の代替実施形態では、項目収束(すなわち、データ収束)は、競合時にすべての競合解決に対して特殊処理の決定論的手法を利用することによっても確実に行われる。ただし、この方法は、項目収束(すなわち、データ収束)を確実にする基本同期スキームも可能にするが、必ずしも拡散し得るわけではない競合に対する恣意的な解決ももたらずので、本明細書において上で説明した高度な同期方法ほど堅牢ではない。

【0042】

(結論)

本明細書で説明した様々なシステム、方法、および技術は、ハードウェアまたはソフトウェア、あるいはそうすることが適切な場合には、両方の組合せを用いて実装されること

10

20

30

40

50

ができる。したがって、本発明の方法および装置、あるいは本発明の特定の態様または部分は、実体のある媒体、たとえばフロッピー（登録商標）ディスク、CD-ROM、ハードドライブ、または他のどのマシン可読記憶媒体においても実施されるプログラムコード（すなわち、命令）の形をとることができ、プログラムコードがコンピュータなどのマシンにロードされるとともにそれによって実行されると、マシンは本発明を実施する装置となる。プログラム可能なコンピュータ上のプログラムコードの実行の場合、コンピュータは概して、プロセッサ、プロセッサによって読取り可能な記憶媒体（揮発性、不揮発性メモリおよび/または記憶要素など）、少なくとも1つの入力デバイス、ならびに少なくとも1つの出力デバイスを含む。1つまたは複数のプログラムは好ましくは、高度な手続き型プログラミング言語またはオブジェクト指向プログラミング言語で実装されて、コンピュータシステムと通信する。ただし、プログラム（群）は、そうすることが望まれる場合、アセンブリでも機械言語でも実装されることができる。いずれの場合でも、言語は、コンパイル型言語でもインタープリト型言語でもよく、ハードウェアの実装と組み合わせることができる。

10

【0043】

本発明の方法および装置は、たとえば、電氣的接続またはケーブル布線、光ファイバ、あるいは他のどの伝送の形でもよい何らかの伝送媒体を介して伝送されるプログラムコードの形で実現されることもでき、プログラムコードが受信され、たとえばEPROM、ゲートアレイ、PLD（プログラム可能な論理回路）、クライアントコンピュータ、ビデオレコーダなどのマシンにロードされるとともにそれによって実行されると、マシンは、本発明を実施する装置となる。汎用プロセッサ上で実装されると、プログラムコードは、プロセッサと結合して、本発明の索引づけ機能を実施するように動作する独自の装置を提供する。

20

【0044】

本発明は、様々な図面の好ましい実施形態に関連して説明されたが、本発明から逸脱することなく本発明の同じ機能を実施するために、他の類似の実施形態が使用されることもでき、説明した実施形態に修正および追加が行われ得ることが理解されるべきである。たとえば、例示的な本発明の実施形態は、パーソナルコンピュータの機能をエミュレートするデジタルデバイスという状況で説明されているが、本発明は、このようなデジタルデバイスに限定されず、本出願において述べたように、有線または無線に関わらず、任意の数の既存または新興の計算装置または環境、たとえばゲーム用コンソール、ハンドヘルドコンピュータ、可搬型コンピュータなどにも当てはまり、通信ネットワークを介して接続され、ネットワーク経由で相互作用するこのような任意の数の計算装置に適用され得ることを当業者は理解するであろう。さらに、ハンドヘルドデバイスのオペレーティングシステムおよび他のアプリケーション特有のハードウェア/ソフトウェアインターフェイスシステムを含む様々なコンピュータプラットフォームが本明細書において企図され、特に、無線ネットワーク接続された装置の数は増加し続けることが強調されるべきである。したがって、本発明は、どの1つの実施形態にも限定されるべきではなく、添付の特許請求の範囲による広がりおよび範囲内で解釈されるべきである。

30

【図面の簡単な説明】

40

【0045】

【図1】本発明の態様が組み込まれ得るコンピュータシステムを表すブロック図である。

【図2】最終的に競合解決の非収束（パートナーが拡散定常状態になり得る）を生じる、3つのピアによる同期動作を示す、テーブルを基にしたタイムフロー図である。

【図3】本発明のいくつかの実施形態で用いられる、改良された同期プロセスを示すプロセスフロー図である。

【図4】最終的に競合解決の収束を生じる、3つのピアによる同期動作を示す、テーブルを基にしたタイムフロー図である。

【図5】図3に示すプロセスの結果生じる特定の最終状態に対する具体的な処理および競合プロセスに関する、結果として生じるメタデータ、ならびに（新規ローカル変更は同期

50

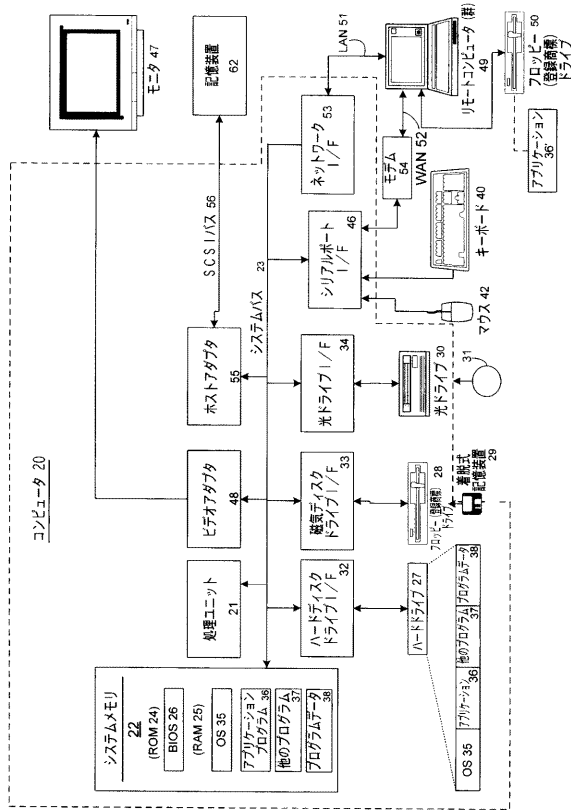
動作の直接の一部ではないので、比較のために) 新規ローカル変更に関するメタデータを示すテーブルである。

【符号の説明】

【 0 0 4 6 】

2 0	コンピュータ	
2 1	処理ユニット	
2 2	システムメモリ	
2 3	システムバス	
2 7	ハードドライブ	
2 8	フロッピー (登録商標) ドライブ	10
2 9	着脱式記憶装置	
3 0	光ドライブ	
3 2	ハードディスクドライブ I / F	
3 3	磁気ディスクドライブ I / F	
3 4	光ドライブ I / F	
3 6	アプリケーションプログラム	
3 6	アプリケーション	
3 7	他のプログラム	
3 8	プログラムデータ	
4 0	キーボード	20
4 2	マウス	
4 6	シリアルポート I / F	
4 7	モニタ	
4 8	ビデオアダプタ	
4 9	リモートコンピュータ (群)	
5 0	フロッピー (登録商標) ドライブ	
5 3	ネットワーク I / F	
5 4	モデム	
5 5	ホストアダプタ	
5 6	S C S I バス	30
6 2	記憶装置	

【図 1】

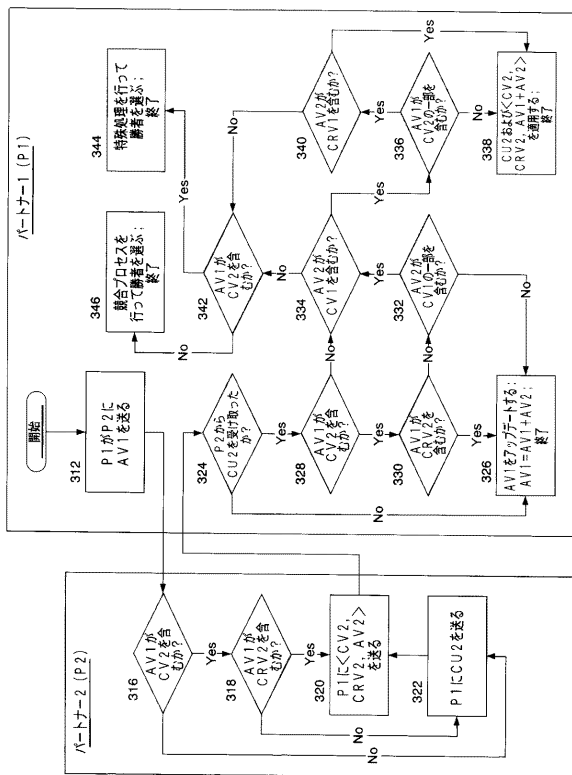


【図 2】

T	パートナーA	パートナーB	パートナーC
0	初期状態: A10 [A10, B10, C10]	初期状態: A10 [A10, B10, C10]	初期状態: A10 [A10, B10, C10]
1	ローカルアップデート: A11 [A11, B10, C10]		
2		ローカルアップデート: B11 [A10, B11, C10]	
3			パートナーBと同期 (競合なし、最終アップデート済み): B11 [A10, B11, C10]
4		パートナーAと同期 (競合あり、リモートを選択): A11 [A11, B11, C10]	
5			パートナーAと同期 (競合あり、ローカルを選択): B11 [A11, B11, C10]
6	パートナーBと同期 (競合なし、最終アップデート済み): A11 [A11, B11, C10]		
7	パートナーCと同期 (変更未検出):		

論理エラー!
競合未検出!

【図 3】



【図 4】

T	パートナーA	パートナーB	パートナーC
0	初期状態: A10, null, [A10, B10, C10]	初期状態: A10, null, [A10, B10, C10]	初期状態: A10, null, [A10, B10, C10]
1	ローカルアップデート: A11, null, [A11, B10, C10]		
2		ローカルアップデート: B11, null, [A10, B11, C10]	
3			パートナーBと同期 (競合なし、最終アップデート済み): B11, null, [A10, B11, C10]
4		パートナーAと同期 (競合あり、リモートを選択): A11, B12, [A11, B12, C10]	
5			パートナーAと同期 (競合あり、ローカルを選択): B11, C11, [A11, B11, C11]
6	パートナーBと同期 (競合なし、最終アップデート済み): A11, B12, [A11, B12, C10]		
7	パートナーCと同期 (競合、SpecDet→A): A11, B12, [A11, B12, C11]		
8		パートナーAと同期 (競合なし、MetaDのアップデート): A11, B12, [A11, B12, C11]	
9			パートナーBと同期 (競合なし、最終アップデート済み): B11, C11, [A12, B12, C11]

【 図 5 】

イベント:	解決	ステータ
特殊処理	決定論的に勝者が選ばれる	< CVdet, CRVdet, AV1+AV2 >
競合プロセス	ローカル変更が勝つか? リモート変更が勝つか? 新規変更が提案されるか?	< CV1, CRVnew, AV1+AV2+ CRVnew > < CV2, CRVnew, AV1+AV2+ CRVnew > < CV1 U CV2, CRVnew, AV1+ AV2+CRVnew >
(同期動作の一部でない) 新規ローカル変更		< CVnew, null, AV1+CVnew >

フロントページの続き

(72)発明者 ダグラス ビー・テリー

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ
マイクロソフト コーポレーション内

(72)発明者 レブ ノビク

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ
マイクロソフト コーポレーション内

Fターム(参考) 5B082 GB02