

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5349733号
(P5349733)

(45) 発行日 平成25年11月20日 (2013.11.20)

(24) 登録日 平成25年8月30日 (2013.8.30)

(51) Int.Cl.

F I

G 0 6 F 9/445 (2006.01)

G 0 6 F 13/00 (2006.01)

G 0 6 F 11/00 (2006.01)

G 0 6 F 9/06 6 4 0 A

G 0 6 F 13/00 5 3 0 B

G 0 6 F 9/06 6 1 0 R

G 0 6 F 9/06 6 3 0 B

G 0 6 F 9/06 6 5 0 J

請求項の数 31 外国語出願 (全 20 頁)

(21) 出願番号 特願2005-360913 (P2005-360913)
 (22) 出願日 平成17年12月14日 (2005.12.14)
 (65) 公開番号 特開2006-172472 (P2006-172472A)
 (43) 公開日 平成18年6月29日 (2006.6.29)
 審査請求日 平成20年12月12日 (2008.12.12)
 (31) 優先権主張番号 11/012,367
 (32) 優先日 平成16年12月14日 (2004.12.14)
 (33) 優先権主張国 米国 (US)

前置審査

(73) 特許権者 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 100107766
 弁理士 伊東 忠重
 (74) 代理人 100070150
 弁理士 伊東 忠彦
 (74) 代理人 100091214
 弁理士 大貫 進介

最終頁に続く

(54) 【発明の名称】 アップデートをダウンロードする方法およびシステム

(57) 【特許請求の範囲】

【請求項 1】

少なくとも1つのアップデートを記憶するためのキャッシュを有する第1のコンピューティングデバイスおよびホストを備えたネットワークにおける前記第1のコンピューティングデバイスを更新する方法であって、

(A) 前記ホストから、利用可能なアップデートのリストを、前記第1のコンピューティングデバイスが受け取るステップと、

(B) 前記(A)に回答して、前記利用可能なアップデートのリストから必要なアップデートを、前記第1のコンピューティングデバイスが判定するステップであって、該必要なアップデートは、複数の部分(ピース)に区分されるステップと、

(C) 前記ホストから、前記複数の部分(ピース)それぞれにつきその部分(ピース)をダウンロードできるようにしている少なくとも1つのコンピューティングデバイスを識別する前記ネットワーク内のコンピューティングデバイスのリストを、前記第1のコンピューティングデバイスが受け取るステップと、

(D) 前記キャッシュ内の利用可能なスペースを生成するように決定された場合、前記キャッシュ内のコンポーネントを、前記第1のコンピューティングデバイスが削除するステップと

(E) 前記コンピューティングデバイスのリストの少なくともいくつかのコンピューティングデバイスから前記複数の部分(ピース)を、前記第1のコンピューティングデバイスがダウンロードするステップであって、前記複数の部分(ピース)の少なくとも2つは

10

20

、異なるコンピューティングデバイスからダウンロードされるステップと、

(F) 前記ダウンロードされた複数の部分(ピース)を利用して、前記第1のコンピューティングデバイスがソフトウェアを更新するステップとを備えることを特徴とする方法。

【請求項2】

前記必要なアップデートのアップデートオプションのリストを、前記第1のコンピューティングデバイスが受け取るステップであって、各該アップデートオプションは、複数の部分に区分されるステップと、

前記アップデートオプションのリストからアップデートオプションを、前記第1のコンピューティングデバイスが選択するステップと

10

をさらに備え、

前記ホストから前記ネットワーク内のコンピューティングデバイスのリストを、前記第1のコンピューティングデバイスが受け取るステップは、前記選択されたアップデートオプションそれぞれについて、少なくとも1つのコンピューティングデバイスを識別するコンピューティングデバイスのリストを受け取るステップを含むことを特徴とする請求項1に記載の方法。

【請求項3】

前記アップデートオプションリストは、前記必要なアップデートについて、フルパッチと少なくとも1つの細部パッチを含むことを特徴とする請求項2に記載の方法。

【請求項4】

20

前記細部パッチは、ファイルのファイル名と予定ハッシュ値を含むことを特徴とする請求項3に記載の方法。

【請求項5】

前記(A)に応答して、前記利用可能なアップデートのリストから必要なアップデートを、前記第1のコンピューティングデバイスが判定するステップは、前記第1のコンピューティングデバイスをスキャンしてその現在の状態を判定し、前記現在の状態を前記利用可能なアップデートのリストと比較することを含み、

前記(D)は、

前記キャッシュ内に記憶された各アップデートに優先度を割り当てることによって、前記キャッシュ内に記憶されたアップデートのための優先度リストを生成するステップと、

30

前記複数の部分(ピース)をダウンロードするための利用可能なスペース持つキャッシュかどうかを判定するステップと、

前記キャッシュが利用可能なスペースを持っていない場合に、削除するためのコンポーネントであって、前記優先度リスト内の低い優先度を持つアップデートに対応するコンポーネントを選択するステップと、

を含み、

前記各アップデートのための優先度の割り当ては各アップデートに対してのリクエストの頻度に基づく

ことを特徴とする請求項1に記載の方法。

【請求項6】

40

前記コンピューティングデバイスのリストは、前記複数の部分の少なくとも1つについて、複数のコンピューティングデバイスの識別を含むことを特徴とする請求項1に記載の方法。

【請求項7】

前記コンピューティングデバイスは、インターネットプロトコルアドレスで識別されることを特徴とする請求項1に記載の方法。

【請求項8】

前記第1のコンピューティングデバイスが前記ソフトウェアを更新するステップは、前記ダウンロードされた部分を組み合わせてパッチにするステップと

前記パッチをインストールするステップと

50

を含むことを特徴とする請求項 1 に記載の方法。

【請求項 9】

前記ダウンロードされた各部分は、前記部分のハッシュに基づいて確認されることを特徴とする請求項 1 に記載の方法。

【請求項 10】

前記第 1 のコンピューティングデバイスが、前記ダウンロードされた複数の部分の識別を提供するステップと、

前記第 1 のコンピューティングデバイスが、前記識別されたダウンロードされた複数の部分を他のコンピューティングデバイスと共有するステップと
をさらに備えることを特徴とする請求項 1 に記載の方法。

10

【請求項 11】

前記ダウンロードされた複数の部分の識別は、ダウンロードされた各部分のハッシュ値を含むことを特徴とする請求項 10 に記載の方法。

【請求項 12】

所定の時点には、前記識別されたダウンロードされた複数の部分は共有されていないことを特徴とする請求項 1 に記載の方法。

【請求項 13】

キャッシュを有するクライアントとホストを備えたネットワークにおけるソフトウェアアップデートをダウンロードするために前記クライアントにピアリストを提供する方法であって、

20

利用可能なアップデートを識別するリストを前記ホストが前記クライアントに送信するステップであって、前記利用可能なアップデートは、複数の部分に分割される、ステップと、

前記クライアントから、前記利用可能なアップデートについてのピアリスト要求を前記ホストが受け取るステップであって、前記ピアリストは、前記複数の部分それぞれにつき少なくとも 1 つのピアを識別する、ステップと、

複数のピアを識別するピアリストを前記ホストが生成するステップであって、識別される各ピアは、前記利用可能なアップデートの少なくとも一部分をダウンロードできるようにしているステップと、

前記ピアリストを前記ホストが前記クライアントに送信するステップと、

30

前記ホストが前記キャッシュ内の利用可能なスペースを生成するように決定した場合、前記キャッシュ内のコンポーネントを前記 クライアント が削除するステップと
を備えることを特徴とする方法。

【請求項 14】

前記利用可能なアップデートを識別する前記リストは、複数の利用可能なアップデートを識別することを特徴とする請求項 13 に記載の方法。

【請求項 15】

前記リストは、前記利用可能なアップデートについて、複数のアップデートオプションを識別することを特徴とする請求項 13 に記載の方法。

【請求項 16】

40

前記複数のアップデートオプションは、フルパッチと少なくとも 1 つの細部パッチを含むことを特徴とする請求項 15 に記載の方法。

【請求項 17】

前記利用可能なアップデートを識別するリストを前記ホストが前記クライアントに送信するのに応答して、前記利用可能なアップデートのアップデートオプションの要求を前記ホストが受け取るステップと、

前記利用可能なアップデートについての複数のアップデートオプションを識別するリストを前記ホストが生成するステップと、

複数のアップデートオプションを識別する前記リストを前記ホストが送信するステップと

50

をさらに備えることを特徴とする請求項 13 に記載の方法。

【請求項 18】

前記クライアントから前記ピアリスト要求を前記ホストで受け取るステップは、
前記複数のアップデートオプションの 1 つについてのピアリスト要求を受け取るステップ
を含むことを特徴とする請求項 17 に記載の方法。

【請求項 19】

前記利用可能なアップデートのキャッシュされた部分の識別を前記ホストが受け取るステップと、

前記利用可能なアップデートのマスタピアリストを前記ホストが更新するステップと
をさらに備えることを特徴とする請求項 13 に記載の方法。

10

【請求項 20】

前記識別されるピアの 1 つはホストコンピュータであることを特徴とする請求項 1 に記載の方法。

【請求項 21】

アップデートを記憶するためのキャッシュを有するコンピュータシステムおよびホストを備えるネットワークにおけるアップデートを取得する方法を行うコンピュータ実行可能プログラムを含むコンピュータ可読媒体を有するコンピュータシステムであって、前記方法は、

(A) 前記ホストから、利用可能なアップデートのリストを前記コンピュータシステムが受け取るステップと、

20

(B) 前記(A)に回答して、前記利用可能なアップデートのリストから必要なアップデートを前記コンピュータシステムが識別するステップと、

(C) 前記必要なアップデートのピアリストを前記ホストに対し前記コンピュータシステムが要求するステップと、

(D) 前記必要なアップデートのピアリストを前記ホストから前記コンピュータシステムが受け取るステップであって、前記ピアリストは、前記必要なアップデートの複数の部分を識別し、各部分につき、その部分をダウンロードできるようにしている前記ネットワーク内のコンピューティングデバイスを識別するステップと、

(E) 前記キャッシュ内の利用可能なスペースを生成するように決定された場合、前記キャッシュ内のコンポーネントを前記コンピュータシステムが削除するステップと、

30

(F) 前記識別されたコンピューティングデバイスから前記複数の部分を前記コンピュータシステムがダウンロードするステップと、

(G) 前記ダウンロードされた複数の部分(ピース)を利用して、前記コンピュータシステムがソフトウェアを更新するステップと
を備えることを特徴とするコンピュータシステム。

【請求項 22】

前記ピアリストは、前記複数の部分それぞれにつき、複数の二次分割部分を識別し、各二次分割部分につき、その二次分割部分をダウンロードできるようにしているコンピューティングデバイスを識別することを特徴とする請求項 21 に記載のコンピュータシステム

40

【請求項 23】

前記ピアリストは、前記複数の二次分割部分それぞれにつき、複数の三次分割部分を識別し、各三次分割部分につき、その三次分割部分をダウンロードできるようにしているコンピューティングデバイスを識別することを特徴とする請求項 22 に記載のコンピュータシステム。

【請求項 24】

前記ピアリストは、各部分のハッシュ値を識別することを特徴とする請求項 21 に記載のコンピュータシステム。

【請求項 25】

50

前記（F）は、

前記ダウンロードされた部分のハッシュ値を計算して第2のハッシュ値を得るステップと、

前記ピアリストの第1のハッシュ値と前記第2のハッシュ値を比較して、前記ダウンロードされた部分の有効性を確認するステップと、

を含み、

前記（E）は、

前記キャッシュ内に記憶された各アップデートに優先度を割り当てることによって、前記キャッシュ内に記憶されたアップデートのための優先度リストを生成するステップと、

前記複数の部分（ピース）をダウンロードするための利用可能なスペース持つキャッシュかどうかを判定するステップと、

前記キャッシュが利用可能なスペースを持っていない場合に、削除するためのコンポーネントであって、前記優先度リスト内の低い優先度を持つアップデートに対応するコンポーネントを選択するステップと、

を含み、前記各アップデートのための優先度の割り当ては各アップデートに対してのリクエストの頻度に基づく

ことを特徴とする請求項24に記載のコンピュータシステム。

【請求項26】

前記コンピュータシステムが、前記ダウンロードされた部分それぞれを組み合わせるステップと、

前記コンピュータシステムが、前記アップデートをインストールするステップとをさらに備えることを特徴とする請求項21に記載のコンピュータシステム。

【請求項27】

前記ピアリストは、前記アップデートの第1のハッシュ値を含み、前記方法はさらに、前記コンピュータシステムが、前記組み合わせられた部分のハッシュ値を計算して第2のハッシュ値を得るステップと、

前記コンピュータシステムが、前記第1のハッシュ値と前記第2のハッシュ値を比較して前記アップデートの有効性を確認するステップと

を含むことを特徴とする請求項26に記載のコンピュータシステム。

【請求項28】

前記コンピュータシステムが、前記ダウンロードされた部分の少なくとも1つをキャッシュに記憶するステップと、

前記コンピュータシステムが、前記キャッシュされた部分の識別を送信するステップと、

前記コンピュータシステムが、前記キャッシュされた部分をダウンロードに供するステップと

をさらに備えることを特徴とする請求項21に記載のコンピュータシステム。

【請求項29】

前記識別されたコンピューティングデバイスから前記部分それぞれを前記コンピュータシステムがダウンロードするステップは、

ダウンロードされる部分の真正性を確認するステップ

を含むことを特徴とする請求項21に記載のコンピュータシステム。

【請求項30】

前記部分それぞれを前記コンピュータシステムがダウンロードするステップは、

ダウンロードする部分を無作為に選択するステップと、

前記部分をダウンロードするコンピューティングデバイスを無作為に選択するステップと

を含むことを特徴とする請求項21に記載のコンピュータシステム。

【請求項31】

前記部分それぞれを前記コンピュータシステムがダウンロードするステップは、

前記部分をダウンロードできるようにしている必要最小限のコンピューティングデバイスを、前記コンピュータシステムが識別するステップと、

前記部分をダウンロードできるようにしている必要最小限のコンピューティングデバイスから、前記コンピュータシステムが該部分をダウンロードするステップと

を含むことを特徴とする請求項 2 1 に記載のコンピュータシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般にはコンピュータのアップデートに関し、詳細には、ピアツーピアのネットワーク環境を利用してアップデート（更新）を取得し、インストールするシステムおよび方法に関する。

【背景技術】

【0002】

コンピュータ、特にコンピュータソフトウェアは、ユーザがインターネットなどのネットワークを介してソフトウェア提供者などのホストからソフトウェアパッチをダウンロードすることによってしばしば更新される。現在は、1つの供給源からパッチ全体がダウンロードされ、ダウンロードを行ったマシンにインストールされて、ソフトウェアを最新の状態に更新する。この技術は実際に用いられているが、いくつかの欠点がある。

【0003】

第1に、パッチをダウンロード可能な状態にするソフトウェア提供者は、複数の顧客が同時にアップデートをダウンロードできるように、十分な量の送出帯域幅（egress bandwidth；送信帯域幅ともいう）を提供しなければならない。多数の顧客を抱えるソフトウェア提供者の場合、そのような送出帯域幅は、非常に高コストになる可能性がある。

【0004】

第2に、顧客は、パッチをインストールし、ソフトウェアを更新する前に、1つの供給源からパッチ全体をダウンロードしなければならない。これは、サイズの小さいパッチには許容できるが、大きなパッチの場合は、1つの供給源からパッチ全体をダウンロードするのに必要とされる時間が、許容できない長さになる可能性がある。第3に、提供されるソフトウェアが、ダウンロードに十分な帯域幅を利用できるようにしていない場合（特にセキュリティアップデートなどの優先度の高いパッチが公開される場合は難しく、高コストになる）、顧客は、パッチのダウンロードを一時的に妨げられる可能性がある。

【0005】

ピアツーピア（peer to peer）ネットワーキングの近年の進歩で、複数のピアから大きなファイルの色々な部分をダウンロードする能力がもたらされている。一例として、図1に、特定のファイルをダウンロードするための典型的なピアツーピア環境のブロック図を示す。特定のファイルをダウンロードするために、クライアント101は、ホスト103にその特定のファイルを要求する。それに応答して、ホストは、それら要求されたファイルの様々な部分をダウンロードできるようにしている複数のピアを識別するピアリストを送信する。例えば、1つのビデオファイルを複数の部分（ピース）に細かく分割し、それらの部分をサブピース（sub-piece）に分割し、さらにそのサブピース（以後、「二次分割部分」と翻訳する）をサブサブピース（sub-sub-piece：以後、「三次分割部分」と翻訳する）に分割することができる。以前に1つの供給源または複数の他のピアから、そのビデオファイルをダウンロードしている各ピアは、ファイル全体の三次分割部分の1つまたは複数を、クライアント101などの他のピアがダウンロードできる状態にすることができる。ホスト103からクライアント101に送信されたピアリストは、要求された特定のファイルの三次分割部分のそれぞれと、その三次分割部分をダウンロードに供させることを可能にする付随ピア（associated peer）とを同定する（identify）。

【0006】

クライアント101は、ホスト103からピアリストを受け取ると、三次分割部分をダウンロードする、各三次分割部分に係わるピアを選択する。ピアを同定すると、クライア

10

20

30

40

50

ント１０１は、特定のファイルのすべての三次分割部分がダウンロードされるまで、選択したピアから各三次分割部分をダウンロードする。色々なピアから部分（ピース）をダウンロードすることにより、複数のダウンロードを同時に行うことができる。各三次分割部分がダウンロードされると、クライアント１０１は、その三次分割部分を、ダウンロードに供するピアにする（すなわちその三次分割部分を共有する）ことを希望するか否かを決定することができる。クライアントはその三次分割部分を共有することを決定する場合には、クライアント１０１は、自身が共有するのに異議がない三次分割部分をホスト１０３に対して明確にする。

【０００７】

図１に関して説明したようなピアツーピアの環境は、ビデオファイルなど、クライアントが要求する特定のファイルをダウンロードする能力を提供するが、クライアントが必要とする可能性のあるファイルを判定する技術は現在ない。例えば、ソフトウェアパッチのダウンロードから言及すると、パッチを要求する前に、クライアントは、どのアップデートが利用可能であるかだけでなく、そのクライアントが必要とするアップデートも知っていなければならない。さらに、その素材（material）がそのダウンロード元のコンピュータに完全に組み込まれる前に、ダウンロード元のコンピュータが、コンピュータ自身が共有しているものを明らかにすることによって、コンピュータが損なわれることから保護するセキュリティ機能が現在ない。さらに、クライアントがローカルのキャッシュスペースを解放（reclaim）する必要がある時に、クライアントのローカルキャッシュにある複数のパッチがソフトウェア提供者（ソフトウェアプロバイダー）によって優先順位を付けられることができる方法は存在しない。

【発明の開示】

【発明が解決しようとする課題】

【０００８】

したがって、クライアントが必要とされるソフトウェアアップデートを特定し、ピアツーピア環境を利用してそのアップデートを安全にダウンロードすることを可能にするシステムおよび方法が必要とされる。

【課題を解決するための手段】

【０００９】

本発明の実施形態は、ソフトウェア提供者が、ピアツーピア環境を利用していくつかのさまざまな顧客にソフトウェアアップデートを配布する能力を提供する。ここに記載される本発明を使用して、これらに限定しないが、オペレーティングソフトウェア、プログラミングソフトウェア、アンチウィルスソフトウェア、データベースソフトウェア等を含む、どのようなタイプのソフトウェアでも更新することができる。セキュリティが高められたピアツーピア環境の使用により、各ピアについてのダウンロード時間を最小にすることができ、また、クライアント（ピア）がアップデートを得られるようにするためにソフトウェア提供者によって提供されなければならない送出帯域幅の量も減らすことができる。

【００１０】

本発明の第１の態様によれば、第１のコンピューティングデバイスを更新する方法が提供される。この方法は、利用可能なアップデートのリストを受け取り、そのリスト中にアップデートに必要なものがあるかどうかを判定する。アップデートする必要とされるものがある場合は、そのアップデートの部分（ピース）と、その部分（ピース）をそこから入手できるコンピューティングデバイスとを同定するコンピューティングデバイスリストを受け取る。そして、それらの部分（ピース）がコンピューティングデバイスからダウンロードされ、複数の部分の少なくとも２つは、異なるコンピューティングデバイスからダウンロードされる。その部分（ピース）がダウンロードされると、ダウンロードされた部分（ピース）を利用して第１のコンピューティングデバイスのソフトウェアが更新される。

【００１１】

本発明の別の態様によれば、ソフトウェアアップデートをダウンロードするためにクラ

クライアントにピアリストを提供する方法が提供される。この方法は、利用可能なアップデートを同定するリストを送信し、利用可能なアップデートについてのピアリスト要求を受け取る。それに応答して、複数のピアを識別するピアリストが生成され、識別される各ピアは、利用可能なアップデートの少なくとも一部分をダウンロードできるようにしている。そしてそのピアリストが送信される。

【0012】

本発明の別の態様によれば、アップデートを取得する方法を行うコンピュータ実行可能プログラムを含むコンピュータ可読媒体を有するコンピュータシステムが提供される。コンピュータシステムは、利用可能なアップデートのリストを受け取り、利用可能なアップデートのリストから必要なアップデートを識別する。次いで必要なアップデートについてのピアリストが要求され、受け取られる。受け取られたピアリストは、必要なアップデートの複数の部分（ピース）を識別し、各部分（ピース）につき、その部分（ピース）をダウンロードできるようにしている1つまたは複数のコンピューティングデバイスを識別する。コンピュータシステムは次いで、識別されたコンピューティングデバイスから、各部分（ピース）をダウンロードする。

10

【0013】

上述の本発明の態様とそれに付随する利点・効果の多くは、以下の詳細な説明を添付図面と併せて参照することにより、よりよく理解されれば、より容易に認識されよう。

【発明を実施するための最良の形態】

【0014】

20

本発明の実施形態は、ここではホストと称するソフトウェア提供者が、ソフトウェアアップデートの複数の部分をさまざまな異なる受信者（ここではピアと称する）に配布する能力を提供する。ここで称されるピアは、ソフトウェアアップデートの一部（ピース）をダウンロードに供する任意のタイプのコンピューティングデバイスを含む。例えば、ピアは、これらに限定しないが、個々のコンピューティングデバイス、サーバ、ホストコンピューティングデバイス、クライアントコンピューティングデバイス等である。ソフトウェアアップデートの部分（ピース）を有するピアは、その部分（ピース）を他のピアがダウンロードに供する。追加的なピアが同じソフトウェア部分を取得すると、それらのピアも、その部分（ピース）を他のピアのダウンロードに供することができる。したがって、部分（ピース）を保持するピアの数が増えるのにしたがって、利用可能なその部分（ピース）のダウンロードサイトの数が増していく。

30

【0015】

図2A～図2Cに、本発明の一実施形態による、ピアツーピア環境を利用してソフトウェアアップデートをダウンロードする状態図を示す。このピアツーピア環境200は、クライアント201が必要とされるアップデートを識別し、そのアップデートの部分（ピース）をそこから取得することができるピアリストをホスト203から取得する能力を提供する。

【0016】

初めに図2Aを参照すると、初期の状態、クライアント201が、ホスト203から入手できるアップデートのリストをホストに要求する。代替実施形態では、ホスト203は、周期的に利用可能なアップデートのリストを公開し、そのリストがクライアント201に入手される。ホスト203は、利用可能なアップデートリストに対する要求を受け取ると、利用可能なアップデートのリストをクライアント201に送信する。クライアント201は、自身をスキャンして現在の状態を判定し、その状態を利用可能なアップデートリストと比較して、必要なアップデートを決定する。必要なアップデートを決定すると、クライアント201は、ホスト203に、その必要とされるアップデートのアップデートオプションのリストを要求する。ホスト203は、必要なアップデートについてのアップデートオプションの要求を受け取ると、その要求されたアップデートオプションリストを作成する。

40

【0017】

50

クライアントのプライバシーの問題のために、本発明の一実施形態では、必要最低限のクライアント情報がクライアントからホストに渡される。例えば、すぐ上で述べたように、ホストが、利用可能なアップデートのリストを提供し、クライアントがどのアップデートが必要であるかを判断する。代替実施形態では、クライアントが、自身の現在のステータスの識別、あるいは各種の動作状態に対応する情報をホストに提供し、ホストが、必要なアップデートをクライアントに通知することができる。

【0018】

図3を参照して、本発明の一実施形態による、必要なアップデートについての様々なオプションを識別する、ホストで生成されることができるアップデートオプションリストについて説明する。アップデートオプションリスト300は、アップデート1(UPDATE1)301などクライアント201から要求される各アップデートについて、そのアップデートを取得するために利用できる様々な選択肢を識別する。また、アップデートオプションリスト300は、いくつかの要求されるアップデートについてのアップデートオプションを含むこともできる。例えば、クライアント201がアップデート1(UPDATE1)301およびアップデート7(UPDATE7)303についてのアップデートオプションを要求したとすると、アップデートオプションリスト300は、要求されたアップデート301、303それぞれの識別と、そのアップデートを入手するために利用できる様々なオプションの識別を含む。例えば、アップデート1(UPDATE1)301は、フルパッチ1(Full Patch1)305、細部1.1(Delta1.1)パッチ307、細部1.2(Delta1.2)パッチ309、細部1.3(Delta1.3)パッチ311の識別を含む。同様に、アップデート7(UPDATE7)303は、フルパッチ7(Full Patch7)313、細部7.1(Delta7.1)パッチ315、細部7.2(Delta7.2)パッチ317、および細部7.3(Delta7.3)パッチ319の識別を含むことができる。

【0019】

フルパッチ(full patch; 完全パッチともいう)1305は、インストールしてソフトウェアを既存の状態から最新の状態にすることができる完全なアップデートである。各細部パッチ(delta patch; デルタパッチともいう)は、フルパッチを使用しなくとも、ソフトウェアを既知の既存の状態から最新の状態にするために使用することができる。アップデートしようとするソフトウェアは、多くの場合、識別されることが可能ないくつかの異なる既知の状態の1つにある。そのため、それら既知の状態それぞれについて、細部パッチを使用してそのソフトウェアに修正を加えてソフトウェアを最新の状態にすることができる。例えば、細部1.1パッチ307が使用されて、ある既知の既存状態にあるソフトウェアのバージョンを最新の状態にすることができる。

【0020】

クライアント201がソフトウェアを更新するために細部パッチの1つを使用できるかどうかを判定できるようにするために、各細部パッチは、ファイルのリストと、それらファイルの予定ハッシュ値(expected hash)を含む。予定ハッシュ値は、その細部パッチが既存のソフトウェアに該当するかどうかを判定するために使用することができる。例えば、細部1.1パッチ307は、予定ハッシュ値321を有するファイル1.1320、予定ハッシュ値323を有するファイル1.2322、および予定ハッシュ値325を有するファイル1.3324を含む。クライアントは、アップデートオプションリスト300を受け取ると、そのクライアントのシステム201で更新しようとするファイルの現在の状態のハッシュ値を計算し、そのハッシュ値が、アップデートリストに含まれる予定ハッシュ値と一致するかどうかを判定することができる。ハッシュ値が一致する場合は、対応する細部パッチを使用してクライアントのシステムを更新することができる。フルパッチ1305などのフルパッチを使用する代わりに、細部1.1307などの細部パッチを使用すると、全体的なアップデートではなく、部分的なアップデートになるので、結果としてダウンロードがより小さくなる。

【0021】

次いで図 2 B を参照すると、ホスト 2 0 3 が、要求されたアップデートオプションリストを生成すると、そのアップデートオプションリストがクライアント 2 0 1 に送信される。クライアント 2 0 1 は、アップデートオプションリストから、クライアント 2 0 1 の状態を最新の状態にする各アップデートのパッチを選択する。上記で述べたように、クライアントは、必要なアップデートのために、フルパッチまたは該当する細部パッチを選択することができる。当業者には理解されるように、1 つまたは複数の必要とされるアップデートがアップデートオプションリストから選択されることができる。該当する各パッチを選択すると、クライアント 2 0 1 は、必要なアップデートについてのピアリストを要求する。ホスト 2 0 3 は、必要なアップデートのピアリストの要求を受け取るのに応答して、要求されるピアリストを作成し、そのピアリストをクライアント 2 0 1 に送信する。

10

【 0 0 2 2 】

ホストは、あるアップデートの部分をダウンロードできるようにしている各ピアを識別するマスタピアリストを保持している。これまでにホストからアップデートがダウンロードされることがない場合、初期のピアリストは、そのホストを利用可能なピアとしてのみ識別することができる。そのような状況では、クライアントは、アップデートをダウンロードする最初のクライアントになり、ホストをピアとして選択し、ホストからファイル全体をダウンロードすることになる。下記で述べるように、クライアントがアップデートをダウンロードし、インストールすると、クライアントは、そのアップデートの 1 つまたは複数の部分をダウンロード（共有）可能にするつもりであることをホストに対して明らかにする。したがって、ホストは、そのクライアントを、そのクライアントが共有するアップデートの部分についてのマスタピアリストに加える。追加的なクライアントがホスト、あるいはピア、またはその両方からアップデートをダウンロードし、自身をそのアップデート部分をダウンロードできるようにしていると識別するにつれて、利用可能なピアの数が増えていく。

20

【 0 0 2 3 】

図 4 に、本発明の一実施形態による、クライアントからのピアリスト要求に応答して、必要とされたアップデートについてのピアを識別するピアリストのブロック図を示す。上記の例を続けると、クライアント 2 0 1 が、細部 1 . 1 パッチ 4 0 1 とフルパッチ 7 4 0 3 を要求したとすると、ピアリスト 4 0 0 は、それら各パッチの詳細とそれらパッチについてのピアの識別を含む。ダウンロードされるパッチの真正性を確認するためのハッシュ値も含まれる。

30

【 0 0 2 4 】

例えば、細部 1 . 1 (D e l t a 1 . 1) パッチ 4 0 1 は、そのパッチの細部ハッシュ (delta hash) と優先度値 (priority) を含む。下記でより詳細に述べるように、パッチのハッシュ値は、インストールの前にそのパッチの有効性を確認するために使用することができる。代替実施形態では、その優先度値は、パッチに含まなくてよい。そのような実施形態では、部分（ピース）はキャッシュに保存され、追加的な記憶スペースが必要になると、クライアントコンピュータは、現在キャッシュに置かれている部分についての優先度リストを要求することができる。これは、ホストが、時間の経過に伴って部分の優先度を動的にコントロールする能力を提供する。これも下記で述べるように、パッチの優先度値を使用して、パッチの部分をクライアントコンピュータのキャッシュに保持すべきか、削除すべきかを決定することができる。ファイル 1 . 1 (F i e l 1 . 1) 4 0 5 やファイル 1 . 2 (F i e l 1 . 2) 4 0 7 などのパッチの各ファイルについては、細部ハッシュも含まれる。パッチの細部ハッシュと同様に、ファイルの細部ハッシュを使用してファイルの真正性 (authenticity) を検証することができる。

40

【 0 0 2 5 】

また、ファイル 1 . 1 4 0 5 などの各ファイルは、部分 1 (P i e c e 1) 4 0 9 や部分 2 (P i e c e 2) 4 1 1 などの部分に分割することができる。ファイルの各部分はさらに、二次分割部分 (S u b - P i e c e) に分割し、その二次分割部分を三次分割部分 (S u b - S u b - P i e c e) に分割することができる。例えば、部分 1 4 0 9 を二次

50

分割部分 1 . 1 4 1 3 と、二次分割部分 1 . 2 4 1 5 などの二次分割部分に分けることができる。同様に、それらの二次分割部分をさらに、三次分割部分 1 . 1 . 1 4 1 7 と三次分割部分 1 . 1 . 2 4 1 9 のような三次分割部分に分割することができる。当業者には理解されるように、部分、二次分割部分、三次分割部分への分割は、任意の分割レベルまで、そして、説明のためだけであっても、ここに記載されるように三次分割部分の細分度まで行える。

【 0 0 2 6 】

三次分割部分のような最も小さい分割レベルは、その部分をダウンロードできるようにしている 1 つまたは複数のピアの識別を含む。例えば、三次分割部分 1 4 1 7 は、その部分をダウンロードできるようにしている 2 つのピアの識別を含む。詳細には、三次分割部分 1 . 1 . 1 4 1 7 は、ピア 1 4 2 1 またはピア 4 4 2 5 からダウンロードすることができる。ある三次分割部分をダウンロードできるようにしている各ピア(peer)は、インターネットプロトコルアドレスなどのアドレス(addr)で識別される。ピアリスト 4 0 0 は、三次分割部分をダウンロードできるようにしているすべてのピア、三次分割部分をダウンロードできるようにしているピアの一部、または各三次分割部分につき 1 つのピアの識別を含むことができる。また、ピアリストに掲載される各ピアについて、追加的な情報が提供されることができる。例えば、ピアについての追加的な情報には、これらに限定しないが、ピアの帯域幅(bandwidth)、そのピアの格付け(そのピアがどれだけ信頼できるか)、企業や組織とのピアの提携、ピアツーピアのダウンロードシステムのピアの使用履歴等が含まれる。ピアリストは、細部パッチ自体のインストール方法についての指示を提供するアップデートコマンドセット 4 2 7 も含むことができる。

【 0 0 2 7 】

再度図 2 B を参照すると、クライアント 2 0 1 は、ピアリストを受け取ると、そのピアリストに識別される三次分割部分についてのピアを選択する。ピアと部分の選択については下記で図 6 との関係でより詳しく説明する。三次分割部分と、その部分をダウンロードするピアを選択すると、クライアント 2 0 1 は、異なるピアとの間にいくつかの通信チャネルを開き、異なるアップデート部分を同時にダウンロードし始める。異なるピアから異なるアップデート部分をダウンロードする技術は、アップデートを取得することができる速度を増し、それにより、合計のダウンロード時間を短縮する。また、ピアツーピア環境を介してアップデートを提供すると、ホスト 2 0 3 に必要とされる送出帯域幅(egress bandwidth)が減る。本発明の実施形態を利用すると、クライアントがホストに接続し、アップデートをダウンロードし、他のクライアント(他のピア)とそれらアップデートの部分を共有し始めると、ホスト 2 0 3 の送出帯域幅は、クライアントがホストではなく他のピアからアップデート部分を取得するようになるのにつれて低減する。最もよい場合でも非常にコストが高くなり、最悪の場合には完全に負荷を受けて機能しない従来のファイルサービスに比べて、本発明のこの技術は、特定のコンテンツに対する需要の極度の増大に、はるかに効果的に対応する。

【 0 0 2 8 】

次いで図 2 C を参照すると、各三次分割部分のダウンロードが完了すると、その三次分割部分のハッシュ値が計算され、ピアリストに含まれるハッシュ値と比較されて、その三次分割部分に変更が加えられていないことを確認する。特定のパッチのすべての三次分割部分がダウンロードされ、認証されると、それらの部分が組み合わせられて(assemble)パッチを復元する。また、ダウンロードされた三次分割部分の 1 つまたは複数のコピーがキャッシュに保持される。そのコピーは、他のピアがダウンロードするのに供することができる。

【 0 0 2 9 】

組み合わせられたパッチがインストールされて、クライアントを最新の状態に更新する。インストールが完了すると、必要な場合はクライアントコンピュータシステムを再起動することができる。パッチのインストールとクライアントコンピュータ 2 0 1 の再起動(必要な場合)が完了すると、クライアント 2 0 1 は、キャッシュに記憶された三次分割部

10

20

30

40

50

分の識別(identification)をホスト203に送信する。この識別は、記憶された各三次分割部分のハッシュ値でよい。ホスト203は、キャッシュされた三次分割部分の識別を受け取ると、マスタピアリストを更新して、それらの部分をダウンロードできるようにしているピアとしてそのクライアントを追加する。マスタピアリストは、他のクライアントからダウンロードできるようにされたパッチの部分を保持している各ピアのリストを含む。

【0030】

図5は、本発明の一実施形態による、クライアントのソフトウェアを更新するクライアントソフトウェア更新ルーチンを説明する流れ図である。クライアントソフトウェア更新ルーチン500は、ブロック501で開始し、ブロック503でクライアントがアップデートリストを要求し、取得する。上記で述べたように、アップデートリストは、ダウンロードすることが可能なアップデートの識別を含む。ブロック505で、そのリストを利用して、クライアントが自身の既存のシステムのステータスを利用可能なアップデートと比較する。その比較に基づいて、判定ブロック507で、クライアントの既存のシステムを最新の状態にするためにアップデートが必要であるかどうか判定される。

【0031】

判定ブロック507でアップデートが必要であると判定された場合は、判定ブロック509で、必要なアップデートの細部パッチが利用できるかどうか判定される。上記で述べたように、必要なアップデートの細部パッチが利用できるかどうかの判定は、必要なアップデートのアップデートオプションを要求し、様々なアップデートオプションのリストを受け取ることによって確かめることができる。様々なアップデートオプションのリストは、細部パッチの識別と、その細部パッチで更新されるファイルの予定ハッシュ値を含む。細部パッチの予定ハッシュ値を、クライアントにあるファイルのハッシュ値と比較することにより、細部パッチを利用できるかどうかの判定がクライアントで行われることができる。代替実施形態では、利用可能なアップデートのリストは、利用可能な各アップデートのアップデートオプションも含むことができ、それにより、後にアップデートオプションを取得する必要をなくする。

【0032】

クライアントが自身の既存のシステムのハッシュ値をホストに提供するのではなく、細部パッチを利用できるかどうかをクライアントが判定できるようにすると、クライアントが自身に関するプライバシーを保つことができるようになる。判定ブロック509で細部パッチが利用できると判定された場合は、ブロック511で示すように、その細部パッチがダウンロードリストに追加される。一方、判定ブロック509で細部パッチが利用できないと判定される場合は、ブロック513で、そのアップデートの全パッチの識別がダウンロードリストに加えられる。

【0033】

判定ブロック515で、クライアントを最新の状態にするために更なるアップデートが必要とされるかどうか判定される。判定ブロック515で更なるアップデートが必要と判定される場合、クライアントソフトウェア更新ルーチン500は、判定ブロック509に戻り、ルーチンが継続する。一方、判定ブロック515で追加的なアップデートが必要でないと判定される場合は、ブロック517で、クライアントが、ダウンロードリストで識別されるパッチについてのピアリストを要求する。ブロック517でピアリストを要求するのに対応して、ブロック519で、クライアントは、ダウンロードリストで識別された各パッチの各三次分割部分をダウンロードできるようにしているピアの識別を含むピアリストを受け取る。ピアリストを受け取ると、ブロック521で示すダウンロードサブルーチンが行われる。ダウンロードサブルーチンについては図6に関してより詳細に説明する。ダウンロードサブルーチンが完了すると、ブロック523で、ダウンロードされた三次分割部分が組み合わせ(assemble)られて、要求されたパッチを復元し、そのパッチがクライアントシステムにインストールされて、クライアントシステムを最新の状態にする。

【0034】

ダウンロードされたパッチのインストールが完了し、必要な場合はクライアントシステムが再起動されると、クライアントソフトウェア更新ルーチン 500 は、ブロック 525 で示すように、キャッシュに格納されている三次分割部分のハッシュを送信する。キャッシュされた三次分割部分は、そのクライアントが他のピアからダウンロードに供する、クライアントのキャッシュに記憶された部分である。本発明の一実施形態では、クライアントは、ピアツーピア環境が確実に維持されるように、他のピアからダウンロードした部分を共有することを要求される。

【0035】

例えば、クライアントがピアツーピア環境を介して 50MB（メガバイト）のアップデートをダウンロードする場合、そのクライアントは、自身がダウンロードした量と同じ量のデータをダウンロード可能な状態にすることを求められる。本発明の実際の実施形態では、各部分が検証され、インストールされるまで、キャッシュされた三次分割部分のハッシュが送信され、それにより共有が可能であることを明らかにしなくてよい。ピアリストがホストから他のピアに提供されると、現在ダウンロード中のパッチなど、特定のパッチがピアにないことを知りながらピアのアドレスが使用されてピアマシンを損なう可能性があるため、パッチを検証およびインストールした後にパッチを共有することが必要である場合がある。ブロック 525 でキャッシュされた三次分割部分のハッシュが送信されると、または、判定ブロック 507 で必要なアップデートがないと判定される場合、クライアントソフトウェア更新ルーチン 500 は、ブロック 527 で完了する。

【0036】

クライアントソフトウェア更新ルーチン 500 は、クライアントが最新の状態に確実に保たれるように、手動で開始されても、あるいは代替実施形態では、周期的に実行されるようにスケジュールされてもよい。例えば、クライアントソフトウェア更新ルーチン 500 は、夜間に、あるいはクライアントコンピューティングデバイスが使用されていない時に実行されるようにスケジュールされることができる。また、追加されたセキュリティ機能として、利用可能アップデートリスト、アップデートオプションリスト、およびピアリストはそれぞれ、ホストによってデジタル的に署名して、変更を加えられないようにすることができる。文書のデジタル署名は、当技術分野で知られており、そのためここでは詳しく説明しない。

【0037】

図 6 は、本発明の一実施形態による、異なるピアからパッチの部分をダウンロードするためにクライアントソフトウェア更新ルーチンの一部として実行されることができるダウンロードサブルーチンの流れ図である。ダウンロードサブルーチン 600 は 1 つのパッチのダウンロードとの関係で説明するが、ダウンロードサブルーチン 600 を使用して、任意数のパッチが同時に、または異なる時にダウンロードされてよいことは理解されよう。

【0038】

ダウンロードサブルーチン 600 は、ブロック 601 で開始し、ブロック 603 で、1 つまたは複数のパッチの各三次分割部分についてのピアリストが受け取られる。上記で述べたように、ピアリストは、あるパッチの異なる三次分割部分をダウンロードできるようにしているピアのアドレスを識別する。三次分割部分をダウンロードする間に、ピアリストは、周期的に（例えば 2 分ごと）更新されて、クライアントが、利用可能なピアにダウンロードを要求することを保証する。

【0039】

そのピアリストを利用して、判定ブロック 605 で、インストールする前にパッチの三次分割部分を記憶するために追加的なキャッシュが必要となるかどうか判定される。判定ブロック 605 で追加的なキャッシュが必要であると判定される場合は、ブロック 607 で示すように優先度サブルーチンが実行される。追加的なキャッシュを得るための優先度サブルーチンについては、図 7 との関係で下記でより詳細に説明する。一方、判定ブロック 605 で追加的なキャッシュが必要でないと判定された場合は、ブロック 609 で、ダウンロードされるパッチのファイルが選択される。ブロック 611 で、ダウンロードを

開始する、選択されたファイルの部分（ピース）が選択される。その選択された部分（ピース）を利用して、判定ブロック 6 1 3 で、その部分の二次分割部分の 1 つのダウンロードが終了したかどうか判定される。その部分の二次分割部分の 1 つのダウンロードが終了していないと判定される場合、判定ブロック 6 1 5 で、その二次分割部分の三次分割部分の 1 つがダウンロードされたかどうか判定される。判定ブロック 6 1 5 で二次分割部分の三次分割部分の 1 つがまだダウンロードされていないと判定される場合は、ブロック 6 1 7 で、二次分割部分の三次分割部分が無作為に選択される。ブロック 6 1 9 で、無作為に選択された三次分割部分をダウンロードできるようにしているピアも無作為に選択される。三次分割部分とピアが選択されると、ブロック 6 2 1 で、選択されたピアからの三次分割部分のダウンロードが開始する。三次分割部分のダウンロードが開始すると、ダウンロードサブルーチン 6 0 0 は、ブロック 6 1 3 に戻り、継続する。

10

【 0 0 4 0 】

再度判定ブロック 6 1 5 を参照すると、三次分割部分の 1 つのダウンロードが完了したと判定される場合は、ブロック 6 2 3 でその三次分割部分をダウンロードすることができる最小量のピアを有する二次分割部分の三次分割部分が選択される。また、ブロック 6 2 5 で、選択された三次分割部分についてのピアの 1 つが無作為に選択される。ブロック 6 2 7 で、無作為に選択されたピアからの選択された三次分割部分のダウンロードが開始される。ブロック 6 2 7 で選択された三次分割部分のダウンロードが開始すると、ダウンロードサブルーチン 6 0 0 はブロック 6 1 3 に戻り、継続する。

【 0 0 4 1 】

20

一度に複数の三次分割部分がダウンロードされることができ、それにより、1 つのパッチまたは複数のパッチをダウンロードするのに必要な合計時間が短縮されるので、ブロック 6 2 1 またはブロック 6 2 7 でダウンロードが開始すると、ダウンロードサブルーチン 6 0 0 は、別の部分と別のピアを選択し、そのダウンロードを開始することによって継続することができる。したがって、アップデートをダウンロードしているクライアントについて帯域幅の完全な飽和が達成されることができ、アップデートのダウンロードが迅速に完了されることができ。また、各三次分割部分のダウンロードが完了すると、その三次分割部分のハッシュ値を計算し（hash）、そのハッシュ値をピアリストに含まれるハッシュ値と比較することにより、それら部分の真正性が検証されることができ。

【 0 0 4 2 】

30

次いでブロック 6 1 3 に戻ると、ある部分の二次分割部分のダウンロードが終了していると判定される場合は、判定ブロック 6 2 9 で、1 つのファイルのすべての部分のダウンロードが終了しているかどうかの判定が行われる。判定ブロック 6 2 9 でファイルのすべての部分のダウンロードが終了していないと判定される場合、ダウンロードサブルーチン 6 0 0 はブロック 6 1 1 に戻り、別の部分を選択し、プロセスが継続する。一方、判定ブロック 6 2 9 でファイルのすべての部分のダウンロードが終了していると判定される場合は、判定ブロック 6 3 1 で、そのパッチのすべてのファイルのダウンロードが終了したかどうか判定される。判定ブロック 6 3 1 でそのパッチのすべてのファイルのダウンロードが終了していないと判定される場合、ダウンロードサブルーチン 6 0 0 はブロック 6 0 9 に戻り、ダウンロードする別のファイルが選択される。一方、判定ブロック 6 3 1 でパッチのすべてのファイルのダウンロードが終了しており、したがって、そのパッチのすべての部分、二次分割部分、および三次分割部分もダウンロードされたと判定される場合は、ルーチンはブロック 6 3 3 で完了し、クライアントソフトウェア更新ルーチン 5 0 0 （図 5）に制御を戻す。

40

【 0 0 4 3 】

図 7 に、本発明の一実施形態による、クライアントシステムのキャッシュサイズを拡張してパッチのダウンロードを可能にする優先度サブルーチンの流れ図を示す。優先度サブルーチン 7 0 0 はブロック 7 0 1 で開始し、ブロック 7 0 3 で、現在キャッシュにある、共有されるすべての三次分割部分のリストがホストに送信される。共有される、キャッシュにある三次分割部分のリストの送信に回答して、ブロック 7 0 5 で、それら三次分割部

50

分それぞれの優先度が受け取られる。三次分割部分には、ダウンロードが要求される頻度に基づいて優先度が割り当てられることができる。例えば、新しいコンピュータウイルスからの防御を提供する新しいアップデートが公開された場合、そのアップデートのパッチと、したがって三次分割部分は、非常に高い頻度でダウンロードされる可能性がある。そのため、そのアップデートの三次分割部分は、高い優先度を受け取ることができる。

【0044】

キャッシュされた三次分割部分の優先度を受け取ると、ブロック707で、キャッシュされた三次分割部分の中で優先度の最も低いものがキャッシュから削除される。判定ブロック709で、共有される三次分割部分が削除されるとキャッシュにアップデートのために十分なスペースが空くかどうか判定される。判定ブロック709で十分なスペースが空くと判定される場合、ブロック723で示すように、ルーチンは、ダウンロードサブルーチン(図6)に制御を戻す。一方、判定ブロック709でまだ十分なスペースが空かないと判定される場合は、判定ブロック711で、共有される他のキャッシュされた三次分割部分で、まだ削除されていない三次分割部分があるかどうか判定される。

【0045】

判定ブロック711で共有される、キャッシュされた三次分割部分がまだ残っていると判定される場合、優先度サブルーチン700はブロック707に戻り、次に優先度が低い三次分割部分を削除し、プロセスが継続する。本発明の実際の実施形態では、特定のアップデートのパッチに関連付けられたキャッシュ内の三次分割部分の一部は、削除不可能と識別されることができる。例えば、三次分割部分が、極めて頻繁にダウンロードされるアップデートに関連付けられている場合、その部分は、削除不可能と識別されて、その三次分割部分をダウンロードすることができるピアが十分にあるようにすることができる。

【0046】

判定ブロック711で、キャッシュされた三次分割部分で削除可能なものが残っていないと判定される場合は、判定ブロック713で、すでに完全にダウンロードされたパッチがあるかどうか判定される。上記で述べたように、本発明の一実施形態によれば、複数のパッチが同時にダウンロードされることができる。そのパッチの1つがダウンロードを完了している場合、判定ブロック713で、そのパッチが識別され、ブロック717でそのパッチがインストールされる。そのようなインストールの結果、そのアップデートがインストールされる間、ダウンロードの残りは一時停止されることができる。しかし、既存の技術では、再開しなくとも、一時停止されたダウンロードが再開されることができる。したがって、ダウンロードの一時停止/再開の結果生じる、ダウンロードデータの実質的な損失がない。ダウンロードされたパッチがブロック717でインストールされると、ブロック719で示すように、削除不可能と識別されていない場合は、そのパッチの三次分割部分がキャッシュから削除される。ダウンロードが削除されると、判定ブロック721で、キャッシュにアップデートをダウンロードするのに十分なスペースがあるかどうか判定される。判定ブロック721でキャッシュに十分なスペースがないと判定される場合、優先度サブルーチン700は判定ブロック713に戻り、ダウンロードが完了した他のパッチがあるかどうかを判定し、プロセスが継続する。

【0047】

判定ブロック713に戻り、ダウンロードが完了したパッチがないと判定される場合は、ブロック715でキャッシュのサイズが拡大される。キャッシュのサイズは、クライアントシステムのユーザがキャッシュサイズを拡大するように要求することにより拡大することができる。あるいは、キャッシュサイズは、自動的に拡大されることもできる。キャッシュから既存の部分を削除するか、キャッシュサイズを拡大することにより、アップデートをダウンロードするために十分なキャッシュスペースが準備されると、優先度サブルーチン700はブロック723で完了し、ダウンロードサブルーチン600に制御を戻す。

【0048】

図8に、本発明の一実施形態による、ピアツーピア環境を利用してアップデートを利用

10

20

30

40

50

できるようにするホストのルーチンを示す。ホストのルーチン 800 は、ブロック 801 で開始し、ブロック 803 でホストがアップデートリストの要求を受け取る。上記で述べたように、アップデートリストは、ダウンロードするためにそのホストから提供される利用可能なアップデートのリストである。ブロック 805 で、ホストが、利用可能なダウンロードを識別するアップデートリストを送信する。

【0049】

アップデートリストが送信されると、ブロック 807 で示すように、後のある時点で、そのアップデートリストで識別される利用可能な 1 つまたは複数のアップデートについてのアップデートオプションの要求がクライアントから受け取られる。ブロック 809 で、ブロック 807 で受け取られた要求で識別されるアップデートについてのアップデートオプションリストが生成される。上記で述べたように、アップデートオプションリストは、そのアップデートを取得するために使用することができるパッチの識別を含む。そのリストが次いで、アップデートオプションを要求したクライアントに送信される。ブロック 811 で、ブロック 809 で送信されたオプションリストで識別されるパッチの 1 つまたは複数についてのピアリストを要求するピアリスト要求が受け取られる。ホストは、ブロック 813 で、要求されるパッチのピアリストを生成し、送信する。上記で述べたように、ピアリストは、各パッチについて、部分（ピース）、二次分割部分、および三次分割部分を識別する。また、その三次分割部分それぞれにつき、その三次分割部分をどこから取得することができるのかを明らかにする 1 つまたは複数のピアとそのピアのアドレスが含まれる。次いで、ピアリストがクライアントに送信され、ブロック 815 で、そのクライアントにダウンロードされ、記憶されて現在は他のピアと共有できるようになっている三次分割部分の識別が受け取られる。受け取られたリストが使用されて、利用可能なアップデートのすべての三次分割部分の識別と、その三次分割部分をダウンロードできるようにしているピアの識別を含むマスタピアリスト 817 を更新する。ホストルーチン 800 は、ブロック 819 で完了する。

【0050】

本発明の実施形態を図示し、説明したが、本発明の主旨および範囲から逸脱せずに本発明に各種の変更を加えてよいことは理解されよう。

【図面の簡単な説明】

【0051】

【図 1】特定のファイルをダウンロードするための典型的なピアツーピア環境のブロック図である。

【図 2 A】本発明の一実施形態による、ピアツーピア環境を利用してソフトウェアアップデートをダウンロードする状態図である。

【図 2 B】本発明の一実施形態による、ピアツーピア環境を利用してソフトウェアアップデートをダウンロードする状態図である。

【図 2 C】本発明の一実施形態による、ピアツーピア環境を利用してソフトウェアアップデートをダウンロードする状態図である。

【図 3】本発明の一実施形態による、ホストで生成されることができる、必要なアップデートのための様々なオプションを明らかにするアップデートオプションリストのブロック図である。

【図 4】本発明の一実施形態による、クライアントからのピアリストの要求に回答して必要なアップデートのピアを識別するピアリストのブロック図である。

【図 5】本発明の一実施形態による、クライアントのソフトウェアを更新するクライアントソフトウェア更新ルーチンを説明する流れ図である。

【図 6】本発明の一実施形態による、異なるピアから必要なアップデートの部分をダウンロードするためにクライアントソフトウェア更新ルーチンの一部として実行されることができるダウンロードサブルーチンの流れ図である。

【図 7】本発明の一実施形態による、パッチのダウンロードを可能にするためにクライアントシステムのキャッシュサイズを拡大する優先度サブルーチンの流れ図である。

【図 8】本発明の一実施形態による、ピアツーピア環境を利用してアップデートを取得できるようにするホストルーチンを説明する図である。

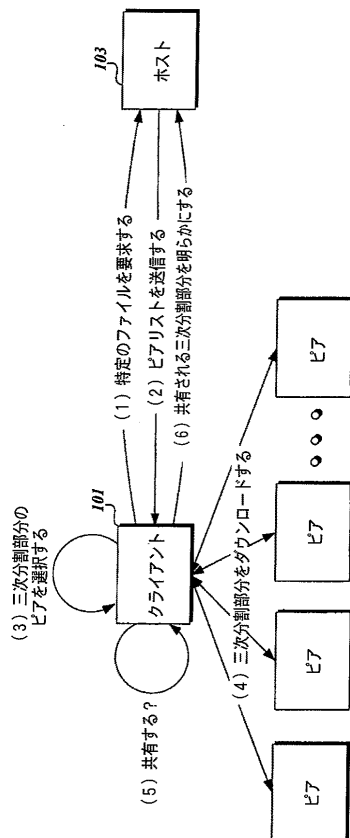
【符号の説明】

【 0 0 5 2 】

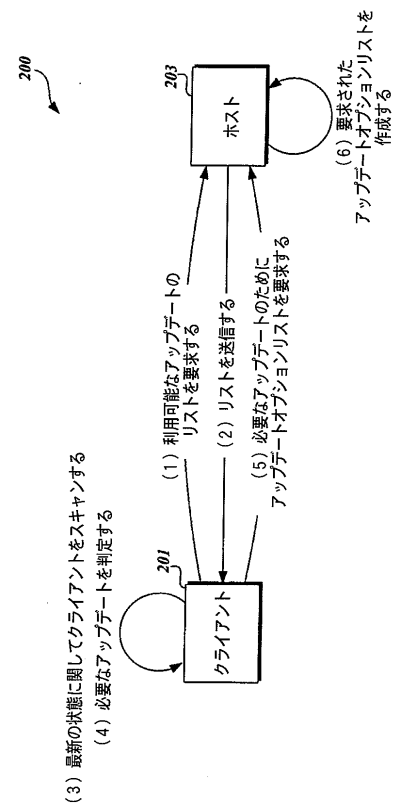
- 1 0 1 クライアント
- 1 0 3 ホスト
- 2 0 0 ピアツーピア環境
- 2 0 1 クライアント
- 2 0 3 ホスト
- 3 0 0 アップデートオプションリスト
- 4 0 0 ピアリスト

10

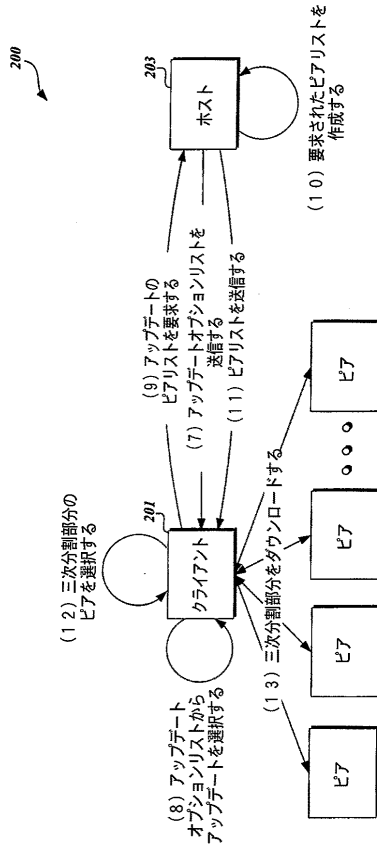
【 図 1 】



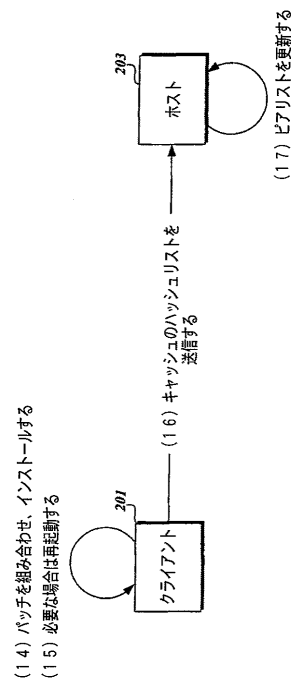
【 図 2 A 】



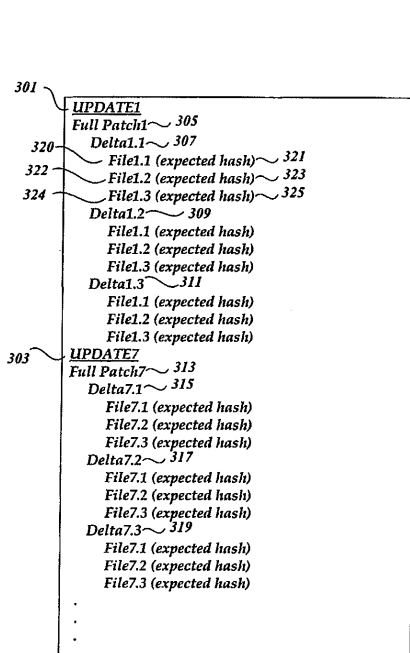
【図 2 B】



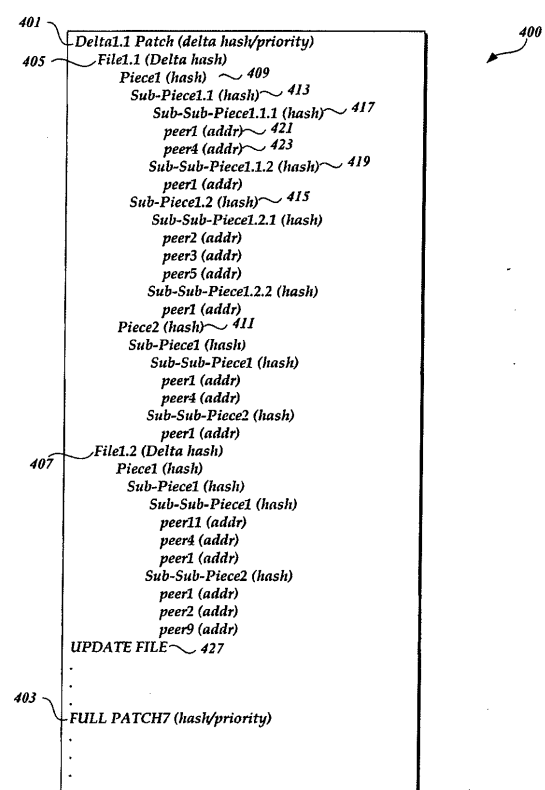
【図 2 C】



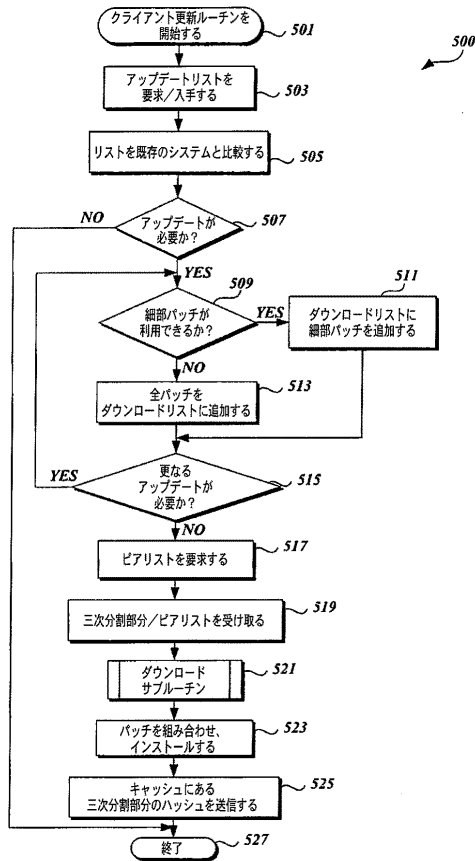
【図 3】



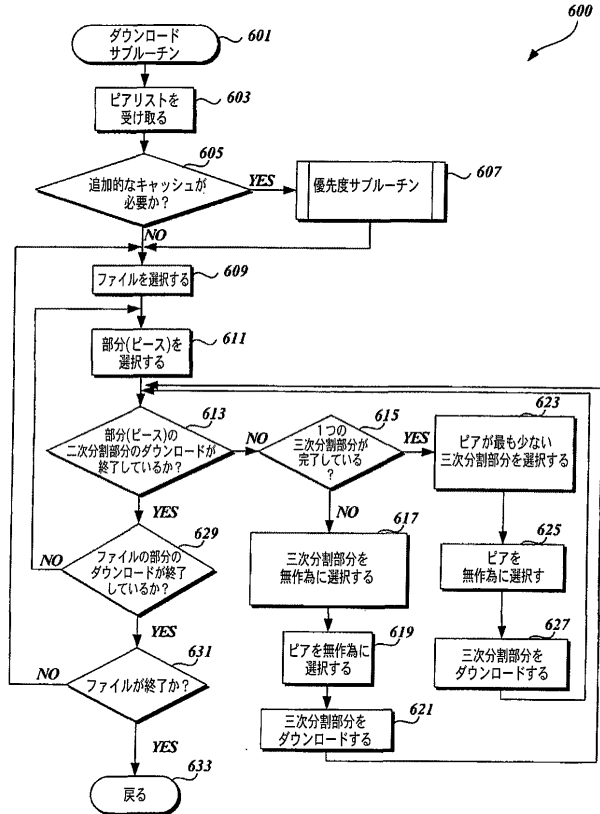
【図 4】



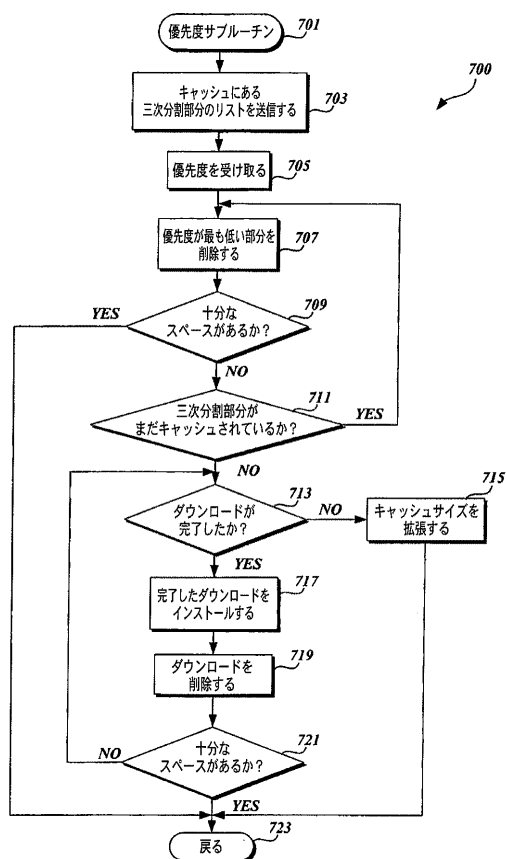
【図 5】



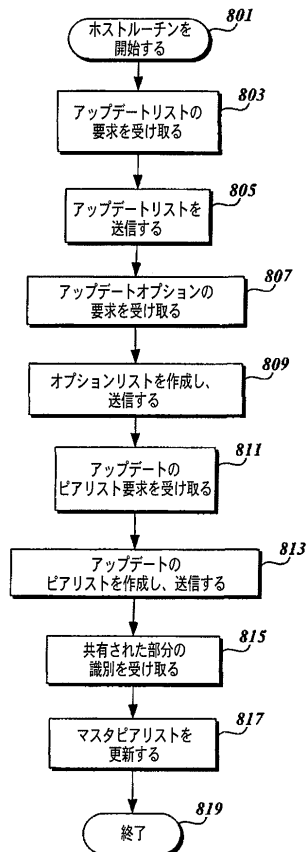
【図 6】



【図 7】



【図 8】



フロントページの続き

(72)発明者 ジェイソン エフ・マッカイ
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

審査官 久保 光宏

(56)参考文献 国際公開第03/015377(WO, A1)
特開2004-213508(JP, A)
特開2004-126912(JP, A)
太田純, 「FreeBSD Updateでらくらくアップデート」, Software Design(ソフトウェアデザイン), 日本, (株)技術評論社, 2004年6月18日, 2004年6月号(通巻第164号), 第162~167頁, ISSN:0916-6297
田口剛(外3名), 「企業向けP2P情報共有システムにおけるキャッシュ制御方式の提案」, DICO 2004シンポジウム論文集(情報処理学会シンポジウムシリーズ), 日本, 社団法人情報処理学会, 2004年7月7日, Vol.2004, No.7, 第627~630頁, ISSN:1344-0640
戸田智雄(外2名), 「VOD配信に適したPeer-to-Peerファイル共有システム」, 情報処理学会研究報告, 日本, 社団法人情報処理学会, 2004年10月28日, Vol.2004, No.104(2004-QA-13), 第33~38頁, ISSN:0919-6072

(58)調査した分野(Int.Cl., DB名)
G06F9/445,
G06F11/00,
G06F9/54,
G06F13/00,
CSDB(日本国特許庁)