

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5819416号  
(P5819416)

(45) 発行日 平成27年11月24日(2015.11.24)

(24) 登録日 平成27年10月9日(2015.10.9)

(51) Int.Cl.	F I
GO 6 F 12/00 (2006.01)	GO 6 F 12/00 5 2 O P
GO 6 F 13/00 (2006.01)	GO 6 F 13/00 5 2 O R

請求項の数 15 (全 20 頁)

(21) 出願番号	特願2013-515377 (P2013-515377)	(73) 特許権者	314015767
(86) (22) 出願日	平成23年6月6日(2011.6.6)		マイクロソフト テクノロジー ライセンシング, エルエルシー
(65) 公表番号	特表2013-534007 (P2013-534007A)		アメリカ合衆国 ワシントン州 98052 レッドモンド ワン マイクロソフト ウェイ
(43) 公表日	平成25年8月29日(2013.8.29)		
(86) 国際出願番号	PCT/US2011/039318	(74) 代理人	100107766
(87) 国際公開番号	W02011/159517		弁理士 伊東 忠重
(87) 国際公開日	平成23年12月22日(2011.12.22)	(74) 代理人	100070150
審査請求日	平成26年6月5日(2014.6.5)		弁理士 伊東 忠彦
(31) 優先権主張番号	12/818,515	(74) 代理人	100091214
(32) 優先日	平成22年6月18日(2010.6.18)		弁理士 大貫 進介
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 データ記憶とデータ送信の最適化

(57) 【特許請求の範囲】

【請求項 1】

クライアントとデータ記憶サーバを含むコンピューティング環境において、前記データ記憶サーバ内の記憶最適化の詳細が前記クライアントに見えるようにする方法であって、

前記クライアントにより、前記データ記憶サーバ上のファイルデータの記憶を記述したメタデータにアクセスするステップであって、メタデータにアクセスすることは、前記メタデータに対するリクエストを前記データ記憶サーバに送信することを有し、前記ファイルデータは、前記ファイルデータのネイティブ形体とは異なる形体で前記データ記憶サーバ上にストアされており、前記メタデータは、前記データ記憶サーバ上にストアされている前記ファイルデータの前記異なる形体を見るようにする、ステップと、

前記クライアントにて、前記データ記憶サーバ上の前記ファイルデータの記憶を記述したメタデータを含む情報を、前記データ記憶サーバから受信するステップであり、前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータは、前記ファイルデータを展開するために使用可能な前記ファイルデータに関する圧縮情報を記述し、前記データ記憶サーバは前記クライアントから遠隔に位置する、ステップと、

前記クライアントにより、前記ファイルデータに対する後続リクエストを送信するステップであり、前記ファイルデータに対する前記後続リクエストは少なくとも部分的に、前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータに基づき、該リクエストは、前記データ記憶サーバにストアされている前記異なる形体の前記ファイルデータに対するものである、ステップと、

10

20

前記クライアントにて、前記ファイルデータを受信するステップであり、前記ファイルデータは、前記データ記憶サーバにストアされている前記異なる形体の前記ファイルデータにて受信される、ステップと、

前記クライアントにより、前記異なる形体の前記ファイルデータを、前記メタデータに基づいて、前記ネイティブ形体の前記ファイルデータに変換するステップとを含む方法。

【請求項 2】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータは更に、前記データ記憶サーバ上の前記ファイルデータの非重複の結果得られる前記ファイルデータの記憶を記述したデータを含む、請求項 1 に記載の方法。

10

【請求項 3】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータは更に、前記ファイルデータのサブセットの暗号ハッシュを含む、請求項 1 に記載の方法。

【請求項 4】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータは更に、前記ファイルデータの複数のサブセットの各々の暗号ハッシュを含むことを特徴とする請求項 1 に記載の方法。

【請求項 5】

当該方法は更に、ファイルデータに対する後続リクエストを有し、ファイルデータに対する該リクエストは、前記ファイルデータのサブセットの暗号ハッシュを含む、請求項 1 に記載の方法。

20

【請求項 6】

クライアントとデータ記憶サーバを含むコンピューティング環境において、前記データ記憶サーバ内の記憶最適化の詳細が前記クライアントに見えるようにする方法であって、前記データ記憶サーバにて、ファイルデータの記憶を記述したメタデータにアクセスするリクエストを受信するステップと、

前記リクエストを受信したことに応答して、前記データ記憶サーバから、前記データ記憶サーバ上のファイルデータの記憶を記述したメタデータを送信するステップであって、前記ファイルデータは、前記ファイルデータのネイティブ形体とは異なる形体で前記データ記憶サーバ上にストアされており、前記メタデータは、前記データ記憶サーバ上にストアされている前記ファイルデータの異なる形体に見えるようにする、ステップと、

30

前記データ記憶サーバ上にストアされている前記異なる形体の前記ファイルデータでの前記ファイルデータに対する前記クライアントからのリクエストを前記データ記憶サーバで受信するステップであり、前記ファイルデータに対する前記リクエストは少なくとも部分的に、前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータに基づき、前記クライアントは前記データ記憶サーバから遠隔に位置する、ステップと、

前記ファイルデータ及び前記データ記憶サーバ上の前記ファイルデータの記憶を記述した追加メタデータを含む情報を前記データ記憶サーバから送信するステップであり、前記追加メタデータは、前記ファイルデータを展開するために使用可能な前記ファイルデータに関する圧縮情報を記述し、前記追加メタデータは、前記異なる形体の前記ファイルデータを前記ネイティブ形体の前記ファイルデータに変換するのに使用可能である、ステップと

40

を含む方法。

【請求項 7】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータは、前記データ記憶サーバ上の前記ファイルデータの非重複の結果得られる前記ファイルデータの記憶を記述したデータを含む、請求項 6 に記載の方法。

【請求項 8】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータは、前記ファイルデータのサブセットの暗号ハッシュを含む、請求項 6 に記載の方法。

50

## 【請求項 9】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータは、前記ファイルデータの複数のサブセットの各々の暗号ハッシュを含む、請求項 6 に記載の方法。

## 【請求項 10】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータは、前記ファイルデータの圧縮したサブセットを記述したデータを含む、請求項 6 に記載の方法。

## 【請求項 11】

1 つ以上のコンピュータプロセッサ上で実行されるときに、データ記憶サーバ内の記憶最適化の詳細がクライアントに見えるようにする方法を前記 1 つ以上のコンピュータプロセッサに実行させるコンピュータ実行可能命令、を有するコンピュータプログラムであって、前記方法は、

前記クライアントから、ファイルデータの記憶を記述したメタデータにアクセスするリクエストを、前記データ記憶サーバに送信するステップと、

前記クライアントにて、前記データ記憶サーバ上の前記ファイルデータの記憶を記述したメタデータを含む情報を、前記データ記憶サーバから受信するステップであり、前記ファイルデータは前記データ記憶サーバ上で圧縮されており、前記ファイルデータの記憶を記述した前記メタデータは、前記ファイルデータを展開するために使用可能な前記ファイルデータに関する圧縮情報を記述したデータを有する、ステップと、

前記クライアントにより、前記ファイルデータに対する後続リクエストを送信するステップであり、前記ファイルデータに対する前記後続リクエストは少なくとも部分的に、前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータに基づき、前記ファイルデータに対する前記後続リクエストは、前記データ記憶サーバにストアされている前記圧縮されたファイルデータに対するものである、ステップと、

前記クライアントにて、前記圧縮されたファイルデータを受信するステップと、前記クライアントにより、前記圧縮されたファイルデータを前記メタデータに基づいて展開するステップと

を含む、

コンピュータプログラム。

## 【請求項 12】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータを含む前記情報は、前記データ記憶サーバ上の前記ファイルデータの非重複の結果得られる前記ファイルデータの記憶を記述したデータを含む、請求項 11 に記載のコンピュータプログラム。

## 【請求項 13】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータを含む前記情報は、前記ファイルデータのサブセットの暗号ハッシュを含む、請求項 11 に記載のコンピュータプログラム。

## 【請求項 14】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータを含む前記情報は、前記ファイルデータの複数のサブセットの各々の暗号ハッシュを含む、請求項 11 に記載のコンピュータプログラム。

## 【請求項 15】

前記データ記憶サーバ上の前記ファイルデータの記憶を記述した前記メタデータを含む前記情報は、前記ファイルデータの圧縮したサブセットを記述した情報を含む、請求項 11 に記載のコンピュータプログラム。

## 【発明の詳細な説明】

## 【背景技術】

## 【0001】

記憶最適化機能性(storage optimization functionality)は、ファイルサーバおよびデータ記憶マーケットにおいて競争力を付けるためにますます重要になっている。ネットワークトラフィックの最適化も、コンピュータ環境とネットワーク環境、および既存のネットワークインフラストラクチャに統合される機器(appliance)において重要になっており、リアルタイムでネットワークトラフィックの最適化を行なうことによって有用な便益が得られる。

【 0 0 0 2 】

コンピュータ上で生成され、送信され、ストアされるデータの量は急速なペースで増加を続けている。コンシューマ（消費者）およびコンペティタ（競争企業）は、保存データに必要とされる記憶量を削減するためにデータ最適化手法を使用する傾向を高めている。例えば、データは圧縮されることがあり、ストアされたデータ内の冗長性は、データのストアのために必要なスペースを削減するために低減されることがある。ネットワーク上を転送されるデータ量を減少するために類似の手法も適用されつつある。しかし、データ記憶とデータ送信に対する現在のソリューションは大部分が別々で行われており、統一されたソリューションは知られていない。記憶手法と送信手法が別々であるために、データ記憶とデータ送信を一緒に見た場合に、冗長性、非互換性および不必要なオーバーヘッドが生じている。

【 0 0 0 3 】

一例として、サーバ（すなわち、データストア）上にストアされるファイルは、データ記憶サーバ上にストアされる場合に、別々のセグメント（すなわち、チャンク(chunk)）にて圧縮され、ストアされることがある。クライアントに送信されるファイルをクライアントがサーバに要求するとき、サーバはチャンクを再アセンブル(reassemble)し、ファイルを再構築(reconstitute)するためにファイルを展開(decompress)してからそのファイルをクライアントに送信しなければならない。

【 0 0 0 4 】

同様に、送信バンド幅（例えば、ネットワーク上の）、レイテンシ(latency)または送信コストを低減するためには、ネットワークエージェントはファイルを受け取ったあと、送信前にそのファイルを再び圧縮し、圧縮したファイルを他方のエンドポイントに送信し、そのあと送信経路の他方のエンドでファイルを展開することもある。

【 0 0 0 5 】

有用であるのは、統一されたデータ最適化ツールおよび手法であり、その中には、記憶と送信プロトコル、ファイルシステムAPI、データストア、サーバ、クライアント、アプリケーションおよびクラウドが含まれる。このようなツールと手法を使用すると、データ転送と送信プロトコルによって利用できる、保存データに対して最適化された記憶を提供することにより、ばらばらで別々の既存データ記憶とデータ送信ソリューションを拡張し、向上することができる。

【 発明の概要 】

【 0 0 0 6 】

本発明は、データ記憶とデータ送信をエンドツーエンド(end-to-end)で最適化するための方法、システム、デバイスおよびコンピュータプログラムプロダクトに適用される。例えば、本明細書に記載の実施形態は、データ記憶とデータ送信の両方の効率化と最適化を強化し、向上させることを可能にする。

【 0 0 0 7 】

一例示実施形態は、データ記憶サーバ内の記憶最適化の詳細がクライアントに見えるように(expose)する方法を提供している。この方法は、データ記憶サーバ上のファイルデータの記憶を記述したメタデータにアクセスするステップを含み、そこではファイルデータは、そのファイルデータのネイティブ形体(form)とは異なる形体でデータ記憶サーバ上にストアされている。このメタデータは、ファイルデータの記憶形体がデータ記憶サーバ上にストアされている通りに見えるようにする。

【 0 0 0 8 】

クライアントは、ファイルデータのリクエストを記憶サーバに送信することができ、ファイルデータ、データ記憶サーバ上のファイルデータの記憶を記述した追加メタデータ、および/またはファイルデータの少なくとも一部を表わしているデータを含む情報をデータ記憶サーバから受信することができる。

【0009】

別の例示実施形態によれば、データ記憶サーバ内の記憶最適化の詳細がクライアントに見えるようにする方法が提供されている。この方法は、データ記憶サーバ上の記憶を記述したメタデータを送信するステップを含んでいる。ファイルデータは、そのファイルデータのネイティブ形体とは異なる形体でデータ記憶サーバ上にストアされており、メタデータは、ファイルデータの記憶形体がデータ記憶サーバ上にストアされた通りに見えるようにする。

10

【0010】

データ記憶サーバは、ファイルデータのリクエストをコンピューティングシステムから受信し、ファイルデータ、データ記憶サーバ上のファイルデータの記憶を記述した追加メタデータ、および/またはファイルデータの少なくとも一部を表わしているデータを含む情報を送信する。

【0011】

別の例示実施形態によれば、データ記憶サーバ内の記憶最適化の詳細がクライアントに見えるようにするコンピュータプログラムプロダクトが提供されている。このコンピュータプログラムプロダクトは、特に、ファイルデータのリクエストをコンピューティングシステムからデータ記憶サーバに送信し、データ記憶サーバ上のファイルデータの記憶を記述した情報を含む情報をデータ記憶サーバから受信するためのコンピュータ実行可能命令を含んでいる。

20

【0012】

本発明の追加の特徴と利点は以下の説明の中で明らかにされるが、その一部は以下の説明から自明であることもあれば、本発明の実施によって習得されることもある。本発明の特徴と利点は、特に特許請求の範囲の中で記載されている手段およびその組み合わせによって実現され、取得されることがある。本発明のこれらの特徴およびその他の特徴は、以下の説明および特許請求の範囲の記載からより詳細に明らかになるが、以下に記述されている通りに本発明を実施することにより習得されることがある。

30

【0013】

なお、この概要は、以下の詳細な説明に詳しく記載されているコンセプト（概念）を選択して簡単に紹介したものである。この概要は、請求項に記載の主題のキーとなる特徴または基本的特徴を特定するものでも、請求項に記載の主題の範囲を判断する際の一助として使用されるものでもない。

【図面の簡単な説明】

【0014】

本発明の上記およびその他の利点のある特徴がどのようにして取得できるかを説明するために、以下では、添付図面に示した特定の実施形態を参照して上記に簡単に記載した本発明についてより具体的に説明する。これらの図面は本発明の代表的な実施形態のみを図示しており、従って本発明の範囲を限定するものではないとの理解の下で、以下では、添付図面の使用を通してさらに具体的にかつ詳細に本発明を記載し、解説する。

40

【0015】

【図1】データ記憶とデータ送信をエンドツーエンドで最適化する例を示す図である。

【図2】データ記憶とデータ送信をエンドツーエンドで最適化するためのアーキテクチャの例を示す図である。

【図3】データ記憶サーバ内の記憶最適化の詳細がクライアント側から見た通りにクライアントに見えるようにする方法の例を示す図である。

【図4】データ記憶サーバ内の記憶最適化の詳細がサーバ側から見た通りにクライアントに見えるようにするための方法の例を示す図である。

50

## 【発明を実施するための形態】

## 【0016】

本発明は、データ記憶とデータ送信をエンドツーエンドで最適化するための方法、システム、デバイスおよびコンピュータプログラムプロダクトに適用される。例えば、本明細書に記載の実施形態によれば、データ記憶とデータ送信の両方の効率化と最適化を強化することを可能にしている。本発明は、データ記憶サーバ内の記憶最適化の詳細がクライアントに見えるようにする方法、システムおよびコンピュータプログラムプロダクトに適用される。本発明の実施形態は特殊目的または汎用コンピュータを含むことがあり、その中には、本明細書の中で詳しく説明されているように、種々のコンピュータハードウェアまたはモジュールが含まれている。

10

## 【0017】

一例示実施形態によれば、データ記憶サーバ内の記憶最適化の詳細がクライアントに見えるようにする方法が提供されている。この方法は、データ記憶サーバ上のファイルデータの記憶を記述したメタデータにアクセスするステップを含み、そこではファイルデータはそのファイルデータのネイティブ形体とは異なる形体でデータ記憶サーバ上にストアされている。このメタデータは、ファイルデータの記憶形体がデータ記憶サーバ上にストアされている通りに見えるようにする。

## 【0018】

クライアントはファイルデータのリクエストを記憶サーバに送信することができ、ファイルデータ、データ記憶サーバ上のファイルデータの記憶を記述した追加メタデータ、および/またはファイルデータの少なくとも一部を表わしているデータを含む情報をデータ記憶サーバから受信することができる。

20

## 【0019】

別の例示実施形態によれば、データ記憶サーバ内の記憶最適化の詳細がクライアントに見えるようにする方法が提供されている。この方法は、データ記憶サーバ上のファイルデータの記憶を記述したメタデータを送信するステップを含んでいる。ファイルデータは、そのファイルデータのネイティブ形体とは異なる形体でデータ記憶サーバ上にストアされており、メタデータはそのファイルデータの記憶形体がデータ記憶サーバにストアされている通りに見えるようにする。

## 【0020】

データ記憶サーバは、ファイルデータのリクエストをコンピューティングシステムから受信し、ファイルデータ、データ記憶サーバ上のファイルデータの記憶を記述した追加メタデータ、および/またはファイルデータの少なくとも一部を表わしているデータを含む情報を送信する。

30

## 【0021】

別の例示実施形態によれば、データ記憶サーバ内の記憶最適化の詳細がクライアントに見えるようにするコンピュータプログラムプロダクトが提供されている。このコンピュータプログラムプロダクトは、特に、ファイルデータのリクエストをコンピューティングシステムからデータ記憶サーバに送信し、データ記憶サーバ上のファイルデータの記憶を記述した情報を含む情報をデータ記憶サーバから受信するためのコンピュータ実行可能命令を含んでいる。

40

## 【0022】

本発明の実施形態によれば、以下で詳しく説明されるように、例えば、1つまたは2つ以上のプロセッサおよびシステムメモリのようなコンピュータハードウェアを含む特殊目的または汎用コンピュータが含まれているか、または利用されることがある。本発明の範囲内の実施形態は、コンピュータ実行可能命令を実行またはストアするための物理的およびその他のコンピュータ可読媒体および/またはデータ構造も含んでいる。このようなコンピュータ可読媒体は、汎用または特殊目的コンピュータシステムによってアクセス可能ならば、利用可能などのような媒体であってもよい。コンピュータ実行可能命令をストアするコンピュータ可読媒体は物理的記憶媒体であることがある。コンピュータ実行可能命

50

令を実行するコンピュータ可読媒体は伝送媒体であることがある。従って、例を挙げると、本発明の実施形態は、少なくとも2つの異なる別種のコンピュータ可読媒体、すなわち、コンピュータ記憶媒体と伝送媒体を含むことができるが、これらに限定されない。

【0023】

コンピュータ記憶媒体としては、RAM、ROM、EEPROM、CD-ROMまたはその他の光ディスク記憶装置、磁気ディスク記憶装置またはその他の磁気記憶デバイス、または望みのプログラムコード手段をコンピュータ実行可能命令またはデータ構造の形体でストアするために使用可能で、汎用または特殊目的コンピュータによってアクセス可能であるその他の任意の媒体がある。

【0024】

コンピュータプログラムプロダクトは、1つまたは2つ以上のプロセッサ上で実行されるとき、方法、ステップおよび動作を本明細書に記載されているように実行するコンピュータ実行命令が符号化されてそこに格納されている1つまたは2つ以上のコンピュータ可読記憶媒体を含むこともある。

【0025】

「ネットワーク」とは、コンピュータシステムおよび/またはモジュールおよび/または他のエレクトロニックデバイスプログラムの間で電子データをトランスポートするのを可能にする1つまたは2つ以上のリンクとして定義されている。ネットワークまたはその他の通信コネクション(ハードワイヤード、ワイヤレスまたはハードワイヤードとワイヤレスの組み合わせのいずれか)を利用して情報がコンピュータに転送または提供されるとき、コンピュータはそのコネクションを伝送媒体として正しく見ている。伝送媒体として可能なものには、コンピュータ実行可能命令またはデータ構造の形体で望みのプログラムコード手段を伝播するために使用可能であって、汎用または特殊目的コンピュータによってアクセス可能であるネットワークおよび/またはデータリンクがある。上に挙げたものの組み合わせも、当然にコンピュータ可読媒体の範囲内に含まれる。

【0026】

さらに、種々のコンピュータシステムコンポーネントに到着したとき、コンピュータ実行可能命令の形体をしたプログラムコード手段またはデータ構造を伝送媒体からコンピュータ記憶媒体に(またはその逆に)自動的に転送することができる。例えば、ネットワークまたはデータリンクを利用して受信したコンピュータ実行可能命令またはデータ構造は、ネットワークインタフェースモジュール(例えば、“NIC”)内のRAMにバッファリングしたあとで、最終的にコンピュータシステムのRAMおよび/またはコンピュータシステムにある低揮発性(less volatile)のコンピュータ記憶媒体に転送することができる。従って、当然に理解されるように、コンピュータ記憶媒体は、伝送媒体も利用する(または主として)コンピュータシステムコンポーネントに含まれることができる。

【0027】

コンピュータ実行可能命令は、例えば、プロセッサで実行されると、汎用コンピュータ、特殊目的コンピュータまたは特殊目的処理デバイスにある種の関数(function)または関数グループを実行させる命令とデータを含んでいる。コンピュータ実行可能命令は、例えば、バイナリ、アセンブリ言語のような直接形式命令であることも、ソースコードであることもある。構造上の特徴および/または方法上の動作に特有の表現で主題を記載してきたが、当然に理解されるように、請求項に定義した主題は上述した記載の特徴または動作に必ずしも限定されない。むしろ、記載した特徴と動作は、請求項を実現する例示形態として開示されたものである。

【0028】

この分野の当業者ならば理解されるように、本発明は様々なタイプのコンピュータシステム構成と共にネットワークコンピューティング環境で実施されることがあり、その中にはパーソナルコンピュータ、デスクトップコンピュータ、ラップトップコンピュータ、メッセージプロセッサ、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またはプログラマブルコンシューマエレクトロニクス、ネットワークPC、

10

20

30

40

50

ミニコンピュータ、メインフレームコンピュータ、モバイル電話、PDA、ページャ、ルータ、スイッチなどが含まれる。本発明は分散システム環境で実施されることもあり、ここではローカルコンピュータシステムとリモートコンピュータシステムはネットワークを通して（ハードワイヤードデータリンク、ワイヤレスデータリンクによって、またはハードワイヤードデータリンクとワイヤレスデータリンクの組み合わせによって）リンクされており、共にタスクを実行している。分散システム環境では、プログラムモジュールは、ローカルとリモートの両方のメモリ記憶デバイスに置かれていることがある。

【0029】

本明細書で用いられている「モジュール」または「コンポーネント」という用語は、コンピューティングシステム上で実行可能なソフトウェアオブジェクトまたはルーチン指すことができる。本明細書に記載の異なるコンポーネント、モジュール、エンジンおよびサービスはコンピューティングシステム上で実行されるオブジェクトまたはプロセスとして（例えば、別々のスレッドとして）実現されることがある。本明細書に記載のシステムとモジュールは好ましくはソフトウェアで実現されるが、ハードウェアまたはソフトウェアとハードウェアの組み合わせで実現することも可能であり、これは本発明の意図していることである。本明細書の記載において、「コンピューティングエンティティ」は本明細書に前述したように任意のコンピューティングシステムであることも、コンピューティングシステム上を走行する任意のモジュールまたはモジュールの組み合わせであることもある。

【0030】

図1は、本発明が動作することができる環境の例を示す図である。図1は、クライアント110、データストア120、およびクライアント110とデータストア120との間のデータ送信130を示している。データは多数の異なる形体でデータストア120上にストアされることがある。

【0031】

本明細書に提示した実施形態は、図1に図示するような環境でデータ記憶140とデータ送信130を統合し、最適化する方法、システムおよびコンピュータプログラムプロダクトを記載している。

【0032】

ファイルは、連続ファイルとしてそのネイティブ形体でデータストア内にストアされることがある。例えば、ファイルA150は、アプリケーションによって提示されるか、あるいはアプリケーションによって予想される通りに、ファイルのすべてのビット、バイトおよびデータを含む未変更の生(raw)フォーマットまたはネイティブフォーマットでデータストアにストアされている。データは、種々の代替フォーマットでストアされることもある。例えば、データは必要とする記憶スペースを低減するために圧縮フォーマットでストアされることもあれば、データは冗長性を低減し、データストアにストアされたデータを非重複(de-duplicate)にするための手法を使用してストアされることもある。

【0033】

データは、ファイルが別々で異なるデータのサブセットに分割されているチャンク(chunk)またはブロックでデータストア上にストアされることがある。例えば、ファイルは、チャンク160C1 - Cnとしてデータストア内にストアされることがある。ファイルからのデータのサブセットであるチャンクはブロックと呼ばれることもあり、チャンクおよびブロックという2つの用語は本明細書では同意義で使用されている（なお、本明細書で用いられているファイルの用語は論理的に関連する任意のデータグループまたはデータ量を表わしている）。

【0034】

データストアでは、ファイルをチャンクに分割してデータ記憶を最適化するアルゴリズムが使用されることがある。例えば、ファイルは、より効率的にまたはコンパクトにファイルをデータストアにストアするためにチャンク160C1 - Cnに分割されることがある。チャンクに分割されたファイルは、ファイル内の冗長性を減少するとによってより効

率的にストアされることもある。例えば、チャンク C 1 は 2 回以上ファイル内に現れることがある。ファイルをチャンクに分割すると、チャンク C 1 は一度だけデータストアに書くことで済むので、ファイル内に繰り返して現れる各々のチャンク C 1 は、そのチャンク C 1 への参照 (reference) またはポインタ (pointer) で置き換えることが可能になる。

#### 【 0 0 3 5 】

以上から理解されるように、チャンクまたはブロックは任意の固定長である必要でなく、任意の長さ、データ量またはファイルの任意の部分であることもあり、その中にはファイル全体も含まれる。ファイルのチャンクまたはブロックはファイルの任意の長さおよび/またはオフセットであることがある。ファイルをチャンクまたはブロックに分割することは任意のアルゴリズムまたは手法に従って行なわれることがあり、チャンクのサイズは、データがそこに存続するデータストアまたはデータがそこを送信される送信経路の特定の考慮事項によって影響されることも、左右されることもある。

#### 【 0 0 3 6 】

データは、圧縮フォーマットでデータストア内にストアされることもある。例えば、ファイル C 1 7 0 は、オリジナルファイルが圧縮アルゴリズムを使用して圧縮されたときの圧縮フォーマットでストアされており、圧縮で得られたファイル、つまり、ファイル C 1 7 0 は、データストア内に占める記憶スペースがオリジナルの未圧縮ファイルデータよりも少なくなっている。ファイルおよびデータの圧縮は、Lempel-Ziv (LZ)、Lempel-Ziv-Welch (LZW) および M P E G 圧縮のような業界で周知の手法によって行なわれることがある。

#### 【 0 0 3 7 】

圧縮とチャンキング (chunking) (またはブロッキング) を組み合わせたものが、データストアで採用されることもある。例えば、ファイルはチャンクに分割されたあとで、これらのチャンクが圧縮され、圧縮チャンク 1 8 0 C H 1 - C H n としてストアされることがある。

#### 【 0 0 3 8 】

データストア内にストアされたファイルおよびデータを非重複にする (de-duplicating) ことにより、別の圧縮化が得られることがある。非重複にすると、データストアにストアされた別々のファイルに現れることのある同一ファイルまたは同一部分が特定され、重複するファイルまたはデータ部分の 1 つを除くすべてがファイルまたはデータ部分の参照コピー (reference copy) への参照で置き換えられる。ファイルを非重複にすると、特定のファイルまたはデータ部分の 1 つのコピーだけがデータストアにストアされるので、多数の重複ファイルまたはデータ部分が占めるはずであった記憶スペースが節減されることになる。

#### 【 0 0 3 9 】

非重複 (de-duplication) は、ファイルのチャンクレベルで行なわれることもある。例えば、2 つまたはそれ以上のファイルがデータチャンクにチャンキングされた場合、重複するチャンクはデータストア内で冗長チャンクのコピーへの参照で置き換えられることがある。例えば、ファイルは、チャンク C 1 およびデータストア 1 2 0 内にチャンクフォーマットでストアされた他のファイルに関連してストアされた他のチャンクへの参照としてデータストア 1 2 0 上にストアされることがある。例えば、ファイル X はチャンク C 1 - C n への参照としてストアされることがあり、ファイル Y はチャンク C H 1、C 1 および C 2 への参照としてストアされる可能性があり、ファイル Z はチャンク C 1 および圧縮チャンク C H 2 - C H n への参照のリストとしてストアされる可能性がある。

#### 【 0 0 4 0 】

ファイルの非重複、チャンキングおよび圧縮は、これらの組み合わせで行なわれることもある。例えば、ファイルは、1 つまたは 2 つ以上のチャンクとしてデータストア上にストアされることがあり、そこではチャンクの各々が圧縮されている。ファイルデータは任意の組み合わせでストアされることもあり、そこでは一部のファイルは未圧縮でストアされ、一部のファイルは圧縮でストアされ、一部のファイルはチャンクフォーマットでストアされ、一部のファイルはチャンクとしてストアされ、これにより一部のチャンクは圧縮

され、一部のチャンクは未圧縮になっている。

【 0 0 4 1 】

一般的に、あるクライアントがデータストアにデータを要求するとき、そのクライアントは、ファイル全体についてまたはファイルの一部のロジック部分についてデータを要求することになる。例えば、クライアントはファイルシステムを通して`get(fileX)`を要求することもある。ファイルシステムを通して`getFileBytes(fileX; bytes=100-1000)`を要求することもある。ファイルまたはそのファイルの一部がデータストア 1 2 0 からクライアント 1 1 0 に送信 1 3 0 されるとき、要求データを再アセンブルして、クライアントまたはアプリケーションが期待するフォーマットでクライアントに送信するために圧縮データを展開したり、データのチャンクを再アセンブルしたりする負担がデータストアに生じることになる。

10

【 0 0 4 2 】

本明細書に記載の実施形態によれば、クライアントは、データストア上のファイルデータの記憶に関する情報を要求またはその情報にアクセスできるので、データストア上にストアされたデータの記憶詳細に関する情報をクライアントに与えることによって効率化と最適化が得られることがある。例えば、クライアント 1 1 0 は、ファイル X がどのようにデータストアにストアされているかをクライアントに通知するようにデータストア 1 2 0 に要求することがある。データストア 1 2 0 は、ファイル X が圧縮チャンク C H 1 と C H 3 としてストアされていることをクライアントに通知することがある。圧縮チャンクを圧縮形体で送信するとより効率的になるように、この場合には、クライアントは`get(fileX)`を要求するのではなく（これを要求すると、データストアはチャンク C H 1 と C H 3 を展開してファイルを再アセンブルしてから、そのファイルをクライアントに送信する必要が生じる）、これらのチャンク C H 1 と C H 3 をクライアントに送信することをデータストアに要求することができる。

20

【 0 0 4 3 】

実施形態によれば、クライアントは、データストア上のファイルデータの記憶に関する情報にアクセスできるので、データストア上にストアされたデータの記憶詳細に関する情報をクライアントに与えることによって効率性と最適化が得られることもある。例えば、クライアント 1 1 0 は、ファイル X がどのようにデータストアにストアされているかを示す、ローカルにキャッシュまたはストアされた情報にアクセスすることがある。この情報は先行リクエスト(previous request)によってすでに得られていることもある。クライアントとデータストアとの間の先行トランザクションの過程ですでにキャッシュされていることもある。

30

【 0 0 4 4 】

ローカルにストアされているか、またはデータストア 1 2 0 よりも低いレイテンシまたは送信コストで記憶ロケーションから得られるチャンクのコピーをクライアントがすでに持っていれば、追加の効率性が得られることがある。このようなケースでは、そのあとクライアントは`getChunk(CH3)`だけをデータストアに要求することができる。

【 0 0 4 5 】

本明細書に記載の実施形態は、クライアントとデータストアとの間および/またはクライアントと集中化サーバ(centered servers)との間の冗長 L A N および/または W A N トラフィックを減少している。本明細書に記載の実施形態によれば、種々のネットワークファイルシステムプロトコルについて記憶と送信の最適化が得られる。例えば、S M B プロトコルと H T T P プロトコルのどちらも、本明細書に記載のデバイスと手法によって拡張され、強化されることができる。

40

【 0 0 4 6 】

標準ファイルシステムプロトコル（例えば、S M B と H T T P ）は、A P I を提供するように拡張可能であり、その A P I は、データストアによって与えられたときファイルまたはデータ部分がどのようにデータストア上にストアされているかの詳細が見えるようにするデータストアからデータをクライアントが要求するのを可能にする。例えば、クライ

50

アント 1 1 0 は、ファイル X がどのようにストアされているのかに関するデータをデータストア 1 2 0 に要求することがある。例えば、クライアント 1 1 0 は、getStorageDetails(fileX)のようなファイルシステムエクステンション(file system extension)をコールすることがあり、そのレスポンスとしてデータストアは {fileX:= chunks CH1, CH3} を返すことがある。ファイル X がどのようにストアされているかの詳細は分かっているため、クライアントはファイル X に関連するデータをどのようにデータストアに要求するかを決定することができる。クライアントは、その生フォーマットまたはネイティブフォーマットでファイル全体を標準的方法で要求できる。これとは対照的に、本明細書に記載の実施形態は、クライアントが圧縮チャンク C H 3 をクライアントに送信することをデータストアに要求するのを可能にしている。

10

#### 【 0 0 4 7 】

一実施形態では、図 3 に示すように、クライアントは、データ記憶サーバ上のファイルデータを記述したメタデータにアクセスすること 3 1 0 ができ、そこではファイルデータはそのファイルデータのネイティブ形体とは異なる形体でデータ記憶サーバ上にストアされ、メタデータはファイルの記憶形体がデータ記憶サーバ上にストアされた通りに見えるようにしている。データ記憶サーバ上のファイルデータの記憶を記述したメタデータは、ファイルデータがデータストア上でどのようにチャンキングされたか、ファイルデータがデータストア上でどのように圧縮されたか、またはファイルデータがデータストア上でどのようにチャンキングされると共に圧縮されているかを記述した情報であることがある。

#### 【 0 0 4 8 】

20

ファイルがどのようにチャンキングされているかの詳細には、ファイルのどの部分がサーバ上にストアされた各チャンクに対応しているかが含まれていることがある。また、チャンキングの詳細には、ファイルを構成するチャンクの各々の暗号ハッシュ(cryptographic hash)も含まれていることがある。チャンクの暗号ハッシュは、クライアント、アプリケーションおよびデータストアが各チャンクをユニークに識別することを可能にしている。この情報を使用すると、クライアント、アプリケーションおよび他のデータストアは、そのチャンクに同一チャンクが暗号ハッシュで識別されたようにすでに使用可能になっているかどうかを識別できることがある。

#### 【 0 0 4 9 】

ファイルまたはデータの部分(例えば、チャンク)がどのように圧縮されているかの詳細には、データをユニークに識別するためのオリジナル未圧縮データの暗号ハッシュが含まれていることがある。この詳細には、圧縮データをユニークに識別するために圧縮データの暗号ハッシュが含まれていることもある。また、この詳細には、圧縮を行なうために使用された圧縮のタイプ(これは、圧縮データをデータストアから別のエンドポイントに送信したあとその圧縮データを展開するために必要なことがある)も含まれていることもある。圧縮のタイプとしては、例えば、L Z、L Z W、M P E G などがある。

30

#### 【 0 0 5 0 】

メタデータにアクセスすることにより、クライアントはデータストア上のデータの記憶詳細を知ることができる。クライアントがデータストア上のデータの記憶詳細を知ったとき、クライアントはファイルデータのリクエストを記憶サーバに送信すること 3 2 0 がある。本明細書に記載の実施形態を採用すると、クライアントはファイル全体を要求しないで済むので、クライアントは、必要とするファイルのチャンクだけを要求することもあれば、ファイルの圧縮バージョンまたはファイルのチャンクの圧縮バージョンを要求することもある。ファイルデータのリクエストを送信したあと 3 2 0、クライアントは、要求したファイルデータ、記憶サーバ上のファイルデータの記憶を記述した追加メタデータおよび/またはファイルデータの少なくとも一部を表わしているデータを含む情報を記憶サーバから受信すること 3 3 0 ができる。

40

#### 【 0 0 5 1 】

ファイルデータ情報を受信すること 3 3 0 には、データファイル、記憶サーバ上のファイルデータの記憶を記述した追加メタデータおよび/またはファイルデータの少なくとも

50

一部を表わしているデータ、の少なくとも1つが含まれていることがある。この情報には、クライアント側のレガシ(legacy)アプリケーションが予想するように標準フォーマットのファイルデータが含まれていることがある。この情報には、データストア上のファイルデータの記憶を記述した情報が含まれていることがある。この情報には、ファイルデータの少なくとも一部を表わしているデータが含まれていることがある。

#### 【0052】

ファイルデータの記憶を記述したメタデータにアクセスすること310には、ファイルデータの記憶を記述した情報のリクエストをサーバに送信することが含まれることがある。このようなリクエストは、クライアントがファイルシステム(またはネットワークファイルシステム)に対してコールを行なって、ファイル、ファイルデータまたはデータの一部分がどのようにデータストア上にストアされているかの詳細を要求するのを可能にするファイルシステムエクステンションの形体になっていることがある。

10

#### 【0053】

上記とは別に、ファイルデータの記憶を記述したメタデータにアクセスすること310には、ファイルデータの記憶を記述した情報についてローカルストアにアクセスすることが含まれていることがある。ローカルストア内の情報は、先行リクエストに対するレスポンスとしてファイルサーバから以前に受信されていることもあれば、進行中(ongoing)の一連のファイルシステムトランザクションとしてローカルにすでにキャッシュされていることもある。ファイルデータの記憶を記述したメタデータにアクセスすること310には、ファイルデータの記憶形体がデータ記憶サーバ上にストアされているように見えるようにするか、またはローカルにキャッシュされたコピーがどのようにストアされているかがクライアントに見えるようにする詳細を返すファイルシステムコール(file system call)(通常ファイルシステムAPIのエクステンションによって導入された)が含まれていることがある。

20

#### 【0054】

例えば、データ記憶サーバ上のファイルデータの記憶を記述したメタデータには、データ記憶サーバ上のファイルデータの非重複の結果得られるファイルデータの記憶を記述したデータが含まれていることがある。このメタデータには、ファイルを構成するチャンクのチャンクリストが含まれることもあれば、ファイルを構成するチャンクの各々の暗号ハッシュのハッシュリストが含まれていることもある。そのあと、クライアントは、送信されるチャンクの1つまたは2つ以上のリクエストを組み立てるために返されたチャンクリストまたはハッシュリストを使用することもあれば、すでに受信されたまたはローカルにキャッシュされたチャンクのリストを比較してデータストアに要求する必要があるチャンクがあるかどうかを判断するためにハッシュリストを使用することもある。

30

#### 【0055】

例えば、ファイルをダウンロードするとき、クライアントはハッシュリストをファイルサーバに要求し、必要とするデータについてピアクライアント(peer clients)および/またはピアファイルサーバにクエリすることもある。クライアントはハッシュリストを含む情報をそのクエリに対するレスポンスとして受信すること330がある。ハッシュリストはデータがデータストア上にストアされた通りにデータを表わしていることもあれば、クライアントは必要とするデータの部分(例えば、チャンク)だけを要求できることもある。また、データは、必要とするデータがピアにあるときそのピアから読み取られることもあるので、ピアとクライアント間のデータ送信の場合の送信コストまたはレイテンシは、クライアントとデータストア間の送信コストまたはレイテンシよりも低くなっている。

40

#### 【0056】

データ記憶サーバ上のファイルデータの記憶を記述したメタデータには、ファイルデータの圧縮したサブセットを記述したデータまたはファイルデータの圧縮したバージョンを記述したデータが含まれていることもある。この情報を使用すると、クライアントはファイルデータの圧縮したサブセットのリクエストを組み立てる(formulate)ことも、ファイルデータの圧縮したバージョンのリクエストを組み立てることもある。このようにする

50

と、ファイルデータのリクエストに対するレスポンスしてデータを送信する前にファイルデータまたはそのファイルデータのサブセットを展開しないで済むデータストアの効率性が得られる。

【 0 0 5 7 】

一実施形態では、クライアントはファイル全体のリクエストまたはファイルの一部のリクエストを含むことのあるファイルデータのリクエストを送信すること 3 2 0 がある。例えば、ファイルのリクエスト `get(fileX)` またはファイルの一部分のリクエスト `getFileBytes(fileX; bytes=100-1000)` はファイルシステムを通してデータ記憶サーバに送信されることがある。このリクエストに対するレスポンスとして、データ記憶サーバはファイルまたはそのファイルの一部ではなく、要求したファイルまたはそのファイルの一部を含んでいるデータを恐らく異なる形体で送り返すことがある。

10

【 0 0 5 8 】

例えば、データ記憶サーバは、要求したファイルまたはそのファイルの要求した一部を完全にカバーしている一定範囲の圧縮チャンクを含むデータを返す可能性がある。さらに、データ記憶サーバは、返されたチャンクが要求したデータ（および可能性として要求よりも多いデータ）を含むことを示しているチャンクと一緒にファイル記憶メタデータを返す可能性がある。

【 0 0 5 9 】

さらに、返されたチャンクが圧縮されていれば、データ記憶サーバは、返されたデータ（またはデータのチャンク）が圧縮されていることを示すファイル記憶メタデータを返すこともあれば、データを圧縮するためにどの圧縮手法またはアルゴリズムが使用されたか、またはデータを展開するためにどの展開手法またはアルゴリズムを使用する必要があるかを示していることもある。以上から理解されるように、特定の圧縮または展開手法を示すメタデータを返すこともなしに、圧縮データおよび / または圧縮チャンクが返される場合に想定されることのあるデフォルトの圧縮または展開手法が存在することもある。

20

【 0 0 6 0 】

次いで、クライアントはこのデータおよび / またはメタデータをデータ記憶サーバから受信し 3 3 0、該当する展開および / またはチャンクアセンブリをクライアント側で行なって要求データを再構成することがある。以上から理解されるように、このようにすると、クライアントによって実際に要求された特定のデータをデータ記憶サーバに展開および / またはアセンブルさせてからクライアントに送信またはクライアントによって受信する場合よりもデータ送信コストまたは送信レイテンシに起因する効率性が向上する。

30

【 0 0 6 1 】

ファイル記憶メタデータには、チャンクまたは圧縮チャンクの暗号ハッシュリストおよびどのチャンクがファイルデータのどの部分を含んでいるかを示す ID (identification) を含んでいることがある。チャンクまたは圧縮チャンクの暗号ハッシュリストおよびどのチャンクがファイルデータのどの部分を含んでいるかを示す ID を使用すると、クライアントは圧縮データを適当な方法で展開し、および / またはクライアントによって望まれるまたは要求される一定範囲のデータのすべてまたはより多くを含むチャンクを再アセンブルすることがある。

40

【 0 0 6 2 】

ファイル記憶と送信に対する統合化アプローチのための例示アーキテクチャは、図 2 に図示されている。クライアントとサーバ 2 1 0 には、最適化を意識した (aware) アプリケーションおよび / またはサービスが含まれていることがある。クライアントとサーバは、ファイルシステムのアプリケーションプログラミングインタフェース (API) を含むと共に最適化 API を含んでいることがあるファイルシステムインタフェース 2 5 0 と通信することがある。ファイルシステム API には、通常ファイルシステムおよび / またはネットワークファイルシステムのすべての通常コールと関数が含まれていることがある。最適化 API は、データストアにストアされているデータ 2 6 0、2 7 0 および 2 8 0 の記憶詳細が見えるようにする拡張 API エlement (extended API elements) (例えば、関

50

数コールとインタフェース)を含んでいる。

【0063】

ファイルシステムインタフェース250を使用すると、クライアントはデータ記憶サーバ上のファイルデータの記憶を記述したメタデータを要求することができる。ファイルシステムインタフェース250を使用すると、クライアントは、いくつかのフォーマットでデータをデータ記憶サーバに要求することもできる。クライアントは、通常ファイルシステムAPI(例えば、標準またはレガシファイルシステムAPI)を使用してデータを要求して、ファイルをその生またはネイティブフォーマットのままにしておくことがある。また、クライアントは、ファイルの圧縮形体であるファイルの特定のチャンクだけをサーバ上にストアされた通りに要求するために最適化APIを使用してデータを要求することもあり、ファイルの圧縮チャンクをサーバ上にストアされた通りに要求することもある。

10

【0064】

強化された(enhanced)および/または拡張(extended)されたファイルシステムインタフェース250を意識していない(unaware)クライアント、アプリケーションおよびサービス220は、レガシファイルシステムAPIの全機能をそのまま残しているファイルシステムAPIにコールを行なうことにより、依然として通常に、未変更のまま、妨害されることなく動作することがある。

【0065】

最適化を意識しているクライアント、アプリケーションおよびサービス230は、最適化APIにコールを行なって、本明細書に記載の実施形態の全機能を起動(involve)することがある。最適化を意識したクライアント、アプリケーションおよびサービスはハッシュリスト、チャンクリスト、圧縮ファイルなどをデータストアまたはサーバに要求することがある。例えば、ファイルfoo.vhdは、チャンクストア/インデックス270をポイントするチャンクリストとしてデータストア上にストアされることがある。このチャンクリスト/インデックスはチャンク(例えば、チャンク160C1 - Cn)を含むことも、圧縮チャンク(例えば、チャンク180CH1 - CHn)を含むことも、ストアされたチャンクを指してファイルとデータ記憶の非重複やその他の最適化を可能にする参照、ポインタおよびインデックスを含むこともある。

20

【0066】

クライアントは、最適化APIを通してfoo.vhdの記憶を記述したメタデータを要求し、foo.vhdがどのようにストアされているかを記述したメタデータをデータストアから受信することがある。クライアントがそのメタデータにアクセスしたあと、そのクライアントは最適化APIを通してファイルデータのリクエストを記憶サーバに送信することがある。このリクエストはネイティブフォーマットのファイル全体に対することもあれば、1つまたは2つ以上のチャンクだけのこともあれば、チャンクストア/インデックス270にストアされた通りのファイルの圧縮チャンクに対することもある。

30

【0067】

次いで、クライアントは、ファイルデータ、データ記憶サーバ上のファイルデータの記憶を記述した追加メタデータおよびファイルデータの少なくとも一部を表わしているデータ、の1つまたは2つ以上を含む情報をデータ記憶サーバから受信することがある。クライアントはそのネイティブフォーマットでファイル全体を受信することがある。クライアントはファイルのチャンクを受信することがある。クライアントはファイルの圧縮チャンクを受信することがある。クライアントはファイルデータの記憶を記述した追加メタデータを受信することも、ファイルデータの一部を含むデータを受信することもある。クライアントによって受信されたレスポンスは、クライアントおよびアプリケーションがデータストア内のデータの記憶の詳細を意識しているリクエストを行なうのを可能にする拡張最適化APIを通して行なわれたリクエストに対応していることもある。

40

【0068】

別の例において、ファイルbar.docは、最適化サービス240によって圧縮され

50

、チャンクにされ、非重複化されて、チャンクストア/インデックスを指すポインタとしてストアされていることがある。本明細書に記載の実施形態では、クライアントはデータストア上の `bar.doc` の記憶を記述したメタデータを要求することがあり、データストア上の `bar.doc` の記憶を記述した情報を受信したあと、チャンクストア/インデックス 270 にストアされた `bar.doc` の圧縮チャンクの 1 つまたは 2 つ以上のリクエストを送信することがある。圧縮チャンクはクライアントによって要求されているので、データストアは `bar.doc` のチャンクを展開する必要もなければ、データストアは、クライアントからの `bar.doc` に対するリクエストに応答するために `bar.doc` のチャンクを再アセンブルする必要もない。

【0069】

別の実施形態では、データストア内の記憶最適化の詳細がクライアントに見えるようにする方法が提供されている。この方法は、データ記憶サーバ上のファイルデータの記憶を記述したメタデータを送信するステップを含み、そこではデータは、ファイルデータのネイティブ形体とは異なった形体でデータ記憶サーバ上にストアされると共に、メタデータはファイルデータの記憶形体がデータ記憶サーバ上にストアされた通りに見えるようにしている。また、この方法は、コンピューティングシステムからのファイルデータのリクエストをデータ記憶サーバで受信するステップも含んでいる。また、この方法は、ファイルデータ、データストア上のファイルデータの記憶を記述した追加メタデータおよびファイルデータの少なくとも一部を表わしているデータ、の少なくとも 1 つを含む情報をデータ記憶サーバから送信するステップも含んでいる。

【0070】

図 4 に図示するように、サーバまたはデータストアは、データ記憶サーバまたはデータストア上のファイルデータの記憶を記述したメタデータを送信すること 410 がある。ファイルデータはそのファイルデータのネイティブ形体とは異なる形体でデータ記憶サーバ上にストアされている。例えば、ファイルデータは、チャンクフォーマットでも、圧縮フォーマットでも、圧縮フォーマットとチャンクフォーマットの組み合わせでも記憶サーバ上にストアされることがある。

【0071】

送信されるメタデータは、ファイルデータの記憶形体がデータ記憶サーバ上にストアされている通りに見えるようにする情報を提供している。例えば、メタデータは、データがチャンクフォーマット、圧縮フォーマットまたはチャンクフォーマットと圧縮フォーマットの組み合わせでストアされていることを見えるようにする情報を含んでいることもある。メタデータは、ファイルデータを構成するチャンクのハッシュリストをデータストアにストアされているように示す情報を含んでいることもある。データストア上にストアされたチャンクは、記憶サーバ上にストアされたファイルデータ（他のデータファイルと共に）非重複の結果得られたチャンクであることもある。

【0072】

メタデータは、ファイルデータのサブセットの暗号ハッシュを含む情報を含んでいることもある。データのサブセットの暗号ハッシュは、チャンクが別のチャンクと同一であるかどうかを示すためにクライアントによっても、送信デバイスによっても、別データストアによっても使用されることがある。ファイルデータのサブセットの暗号ハッシュを使用することにより、クライアント、送信デバイスおよび他のデータストアは、データの特定サブセットがローカルに使用可能であるか、低いレイテンシまたは送信コストでソースから使用可能であるかを判断する能力を備えている。データの同一サブセットを特定することにより、データの特定サブセットを要求する必要があるか、送信する必要があるかが判断されることもある。

【0073】

ファイルデータのサブセットは、ファイル全体であることも、ファイルデータであることもある。データのサブセットは、記憶最適化または非重複方式の一部としてデータストアによってすでにチャンキングされているファイルデータ、の 1 つまたは 2 つ以上のチャ

10

20

30

40

50

ンクであることもある。

【0074】

データ記憶サーバまたはデータストア上のファイルデータの記憶を記述したメタデータには、ファイルデータの一部または全部がデータ記憶サーバまたはデータストア上に圧縮されていることを記述したデータが含まれていることもある。メタデータには、ファイルデータのチャンクフォーマットの1つまたは2つ以上が圧縮されているとの情報が含まれていることがある。ファイルデータのある部分が圧縮されていることを示す情報を使用することにより、クライアントは、クライアントに対するレスポンスの中でデータストア内にストアされている通りにチャンクフォーマットまたは圧縮フォーマットで返されるファイルまたはファイルの1つまたは2つ以上のチャンクを要求することがある。ファイルの特定チャンクまたは圧縮したチャンクを要求することにより、データストアは、ファイルまたはファイルのチャンクを要求したクライアントに送信する前にファイルまたはファイルのチャンクを展開する必要がないのでオーバーヘッドが減少される。

10

【0075】

図4は、ファイルデータのリクエストをコンピューティングシステムから受信すること420も示している。このリクエストはクライアントからも、別の記憶サーバからも、リモートコンピューティングシステム上で実行されるアプリケーションなどからも受信されることがある。このリクエストは、標準ネットワークファイルシステムAPIを拡張および/または強化する最適化APIに対応するプロトコルを使用してフォーマット化されることがある。

20

【0076】

ファイルデータのリクエストには、要求されるファイルの特定チャンクを示す情報が含まれていることがある。このリクエストには、要求されたファイルデータが圧縮または展開フォーマットで送られるべきかどうかを示す情報が含まれていることもある。このリクエストには、他のチャンクがローカルですでに使用可能であるのでファイルのチャンクのサブセットだけが送信されるはずであるとの情報が含まれていることがある。

【0077】

図4は、ファイルデータの少なくとも1つ、データ記憶サーバ上のファイルデータの記憶を記述した追加メタデータおよびファイルデータの少なくとも一部を表わしているデータを含んでいるファイルデータ情報を送信するステップ430も示している。ファイルデータ情報を送信すること430は、ファイルデータについて受信されたリクエストに対するレスポンスであることがある。上述したように、ファイルデータのリクエストは、圧縮フォーマットでまたはいずれかの組み合わせでチャンクとしてデータストアにストアされている通りのファイルデータに対するものであることがある。

30

【0078】

ファイルデータ情報を送信すること430には、ファイルデータ、データ記憶サーバ上のファイルデータの記憶を記述した追加メタデータおよびファイルデータの少なくとも一部を表わしているデータ、の少なくとも1つが含まれていることがある。この情報は、クライアント側のレガシアプリケーションが予想するように標準フォーマットのファイルデータを含むことがある。この情報は、データストア上のファイルデータの記憶を記述した情報を含んでいることもある。また、この情報はファイルデータの少なくとも一部を表わしているデータを含んでいることもある。

40

【0079】

受信したリクエストは、クライアントによって望まれるデータの特定チャンクをすでに特定していることがある。このリクエストに対するレスポンスとして、データストアはデータの要求されたチャンクを要求側のクライアントに送信することがある。受信したリクエストは、クライアントによって望まれるデータの特定の圧縮チャンクをすでに特定していることがある。このリクエストに対するレスポンスとして、データストアはデータの要求した圧縮サブセットを要求側クライアントに送信することがある。受信したリクエストは、クライアントによって望まれるデータのチャンクを特定する特定暗号ハッシュをすで

50

に特定していることがある。このリクエストに対するレスポンスとして、データストアは、暗号ハッシュによって特定されたデータの特定チャンクを要求側クライアントに送信することがある。

#### 【0080】

一実施形態では、データストアは、ファイルまたはファイルの一部のリクエストを受信すること420がある。例えば、データストアはファイルのリクエスト`get(fileX)`を受信することもある。ファイルの一部のリクエスト`getFileBytes(fileX, bytes=100-1000)`を受信することもある。データストアはそのリクエストに対するレスポンスを構築し、データストア上にストアされたままのファイルデータを含む情報を送信し、ストアされたままのファイルデータの記憶詳細を特定したメタデータを含んでいることがある。例えば、データストアは、チャンクのセットおよびどのチャンクが要求したチャンクのどの部分を含んでいるかを特定したメタデータを返すことがある。さらに、データストアは、圧縮フォーマットで返されたデータを展開するために適切と思われる圧縮および/または展開情報を含むメタデータを返すこともある。

10

#### 【0081】

一実施形態では、メタデータを送信する先行ステップ410を実行することなくリクエストが受信されること420があり、ファイルデータ情報が送信されること430がある。例えば、最適化を意識したクライアントはファイルデータを要求するだけのことがあり、データストアはその要求を受信する420可能性があり、データストアはレスポンスを構築し、クライアントが返されたファイルデータおよび/またはメタデータを適切に処理し、チャンクを適当に再アセンブルしおよび/またはデータを必要に応じて展開できることを想定してクライアントにそのレスポンスを送信する可能性がある。

20

#### 【0082】

実施形態によれば、データ記憶とデータ送信のための書き込み経路(write path)最適化もサポートされている。例えば、ファイルに対しローカルで変更するクライアントは、変更したファイルのハッシュリスト表現を生成することがある。そのあと、このハッシュリストはデータ記憶サーバに送信されることがある。そのあと、データ記憶サーバは変更したファイルを表わす受信したハッシュリストを、データ記憶サーバ上にストアされたファイルチャンクを特定したデータ記憶サーバ上に維持された包括的ハッシュリストと比較することがある。

30

#### 【0083】

この比較に基づいて、そのあとデータ記憶サーバは、クライアントがデータ記憶サーバ上にすでにストアしていたチャンクのリストを返すことがある。データ記憶サーバは、データ記憶サーバ上にストアされていないチャンクのリストをクライアントに返すこともある。データ記憶サーバ上にストアされていて返されたチャンクのリスト(またはストアされていないチャンクのリスト)に基づいて、そのあとクライアントはデータ記憶サーバ上にまだストアされていないチャンクをデータ記憶サーバに送信することが可能である。

#### 【0084】

変更ファイルを表わしているハッシュリストを受信したあとおよびデータ記憶サーバ上にまだストアされていない変更ファイルを受信したあと、データ記憶サーバは、完全な変更ファイル(これはサーバ上にすでにストアされている一部のチャンク、新たにサーバによって受信された一部のチャンクおよび完全な変更ファイルを表わしているハッシュリスト(またはチャンクリスト)で構成されている)をストアすることがある。完全なファイルを表わしているハッシュリスト(またはチャンクリスト)を送信し、データ記憶サーバ上にまだストアされていないチャンクだけを送信することにより、クライアントからデータストアへのデータ送信の最適化が実現されることが可能である。

40

#### 【0085】

例えば、データ記憶サーバはクライアントからハッシュリストを受信し、ファイルを表わしているその送信ハッシュリストを、データ記憶サーバ上にストアされたチャンクおよびデータ記憶サーバ上にストアされたチャンクに対する暗号ハッシュのインデックスを含

50

むチャンクストア/インデックス270にストアされたハッシュリストと比較することがある。そのあとデータストアは、チャンクストアとインデックス270にまだストアされていないチャンクを表わしているハッシュリストをクライアントに返すことがある。そのあと、クライアントはチャンクストアにまだストアされていないチャンクをデータストアに送信することがある。そのあと、データストアは受信したチャンクを、完全な変更ファイルを表わしているハッシュリストと共にチャンクストア270にストアすることがある。このようにして、データ記憶サーバは、変更ファイルの完全な表現（ファイルを表わしているチャンクリストおよび対応するチャンクで表わしている）をストアすることがあるが、ファイルを構成するすべてのチャンクをクライアントが送信する必要はない。

【0086】

別の例では、5個のチャンク、すなわち、チャンクC1 - C5で構成されたファイルはチャンクC4内だけでクライアントによって変更されることがある（その結果、変更したチャンクCm4が得られる）。クライアントはチャンクC1 - C3、Cm4およびC5を表わしているハッシュリストをデータ記憶サーバに送信することがある。このハッシュリストは完全な変更ファイルを表わすことになる。そのあと、データ記憶サーバはサーバ上にストアされたチャンクC1 - C3およびC5をすでに有しているが、チャンクCm4が抜けているクライアントに応答することがある。そのあと、クライアントはチャンクCm4をデータ記憶サーバに送信することが可能である。そのあと、データ記憶サーバは、チャンクC1 - C3、Cm4およびC5を表わしている受信したハッシュリスト及びすでにストアされたチャンクC1 - C3およびC5と共にチャンクCm4をデータ記憶サーバにストアすることあり、データストア上にストアされた完全な変更ファイルを持つことになる。

【0087】

以上から理解されるように、この書き込み経路の実施形態は、変更ファイルについても新しく作成されたファイルについても同じように使用可能である。変更ファイルであるか、新たに作成されたファイルであるかに関係なく、クライアントはどのファイルについてもチャンクリストを作成し、そのチャンクリストをデータ記憶サーバに送信することがあるので、データ記憶サーバは受信したチャンクリストを、サーバ上にすでにストアされているチャンクのリストと比較することができる。さらに、チャンクリストは、ファイルを構成するチャンクの各々をユニークに識別する暗号ハッシュリストであることがある。本明細書で説明したチャンクそれ自体は、圧縮チャンクであることも、生のデータフォーマットのチャンクであることも、暗号または他のなんらかの方法で変更されたチャンクでさえあることもある。

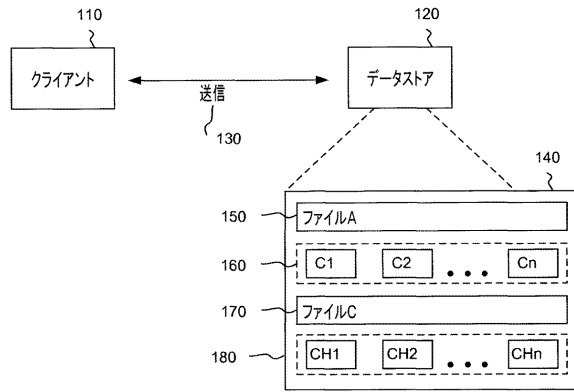
【0088】

チャンクは、その送信時に、生のデータフォーマットでも、圧縮フォーマットでも、その他のフォーマットでも送信されることがある。以上から理解されるように、ファイルデータ部分が圧縮フォーマットで送信されるときは、その結果として、送信の効率性を得るために送信インフラストラクチャがデータを圧縮する必要がないとする最適化が得られることがあるので、データ記憶サーバはデータ記憶サーバ上の記憶を最適化するためにデータを圧縮する必要がない。送信の受信側にまだストアされていないか、あるいは存在しないチャンクだけを送信することにより、ファイルデータの送信と記憶の両方で最適化が実現されることがある。

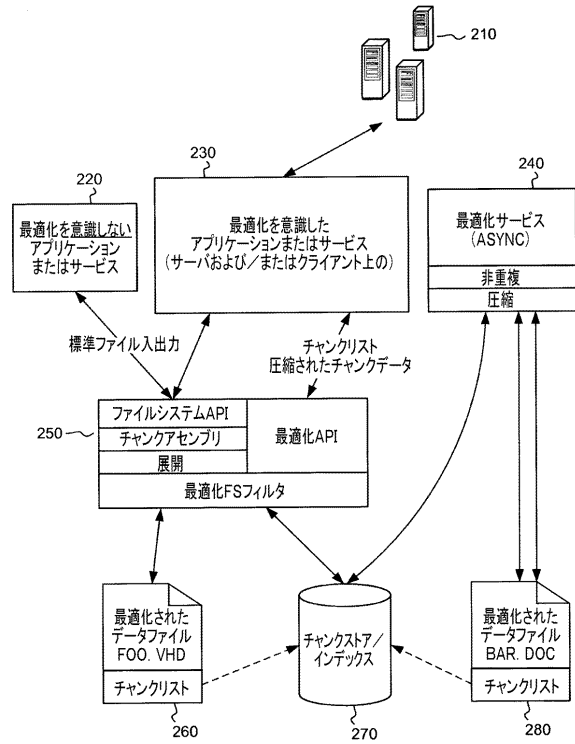
【0089】

本発明は、本発明の趣旨または基本的特徴から逸脱することなく、他の特定の形態で具現化されることがある。本明細書に記述した実施形態は、すべての側面において例示であって限定されるものではないと考慮されるものである。従って、本発明の範囲は上述してきた説明によってではなく特許請求の範囲の請求項によって示されている。請求項の等価の意味と範囲内に属するすべての変更はこれらの範囲内に包含されるものである。

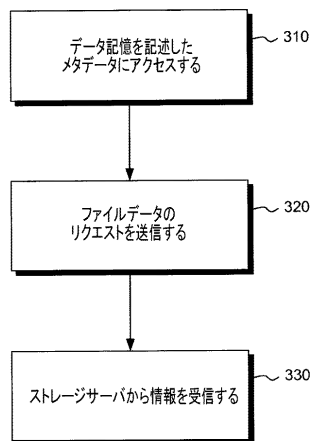
【図 1】



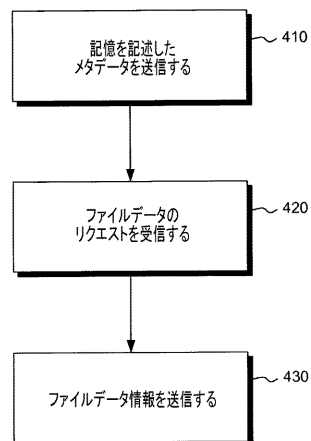
【図 2】



【図 3】



【図 4】



## フロントページの続き

- (72)発明者 アイリーン シー・ブラウン  
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 エルシーエー・インターナショナル パテント内
- (72)発明者 トーマス イー・ジョリー  
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 エルシーエー・インターナショナル パテント内
- (72)発明者 イエルク・トーマス フェニング  
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 エルシーエー・インターナショナル パテント内

審査官 加内 慎也

- (56)参考文献 特開平11-272575(JP,A)  
特開2003-345698(JP,A)  
特表2009-543198(JP,A)  
特表2009-530756(JP,A)  
特表2003-524243(JP,A)  
特表2009-516939(JP,A)  
特開2009-70361(JP,A)  
特開2000-285060(JP,A)

- (58)調査した分野(Int.Cl., DB名)  
G06F 12/00  
G06F 13/00