

(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2014-524610

(P2014-524610A)

(43) 公表日 平成26年9月22日(2014.9.22)

(51) Int.Cl.

G06F 12/00 (2006.01)

F I

G06F 12/00 545F

テーマコード (参考)

審査請求 未請求 予備審査請求 未請求 (全 27 頁)

(21) 出願番号 特願2014-525031 (P2014-525031)
 (86) (22) 出願日 平成24年7月19日 (2012.7.19)
 (85) 翻訳文提出日 平成26年4月10日 (2014.4.10)
 (86) 国際出願番号 PCT/US2012/047261
 (87) 国際公開番号 W02013/022582
 (87) 国際公開日 平成25年2月14日 (2013.2.14)
 (31) 優先権主張番号 13/207,014
 (32) 優先日 平成23年8月10日 (2011.8.10)
 (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 100140109
 弁理士 小野 新次郎
 (74) 代理人 100075270
 弁理士 小林 泰
 (74) 代理人 100101373
 弁理士 竹内 茂雄
 (74) 代理人 100118902
 弁理士 山本 修
 (74) 代理人 100120112
 弁理士 中西 基晴

最終頁に続く

(54) 【発明の名称】 トークン・ベースのファイル動作

(57) 【要約】

トークン・ベース・ファイル動作を可能にする実施形態について記載する。クライアントは、ファイル・アクセス・プロトコルにしたがってフォーマットされた特殊なオフロード・ファイル動作を要求することができる。このファイル動作は、オフロード・リード動作でもオフロード・ライト動作でもよい。オフロード・リード動作では、クライアントは、格納されているファイルからデータまたはその一部を論理的に読み出すことを要求する。応答して、ファイル・サーバーは、論理的に読み出されたデータを表すトークンを含む応答を供給する。実施形態の中には、何らかの理由でデータの全てを表すトークンを供給することができない場合、ファイル・サーバーが要求されたデータの全て未満を表すトークンと共に応答を戻す場合もある。次いで、トークンは、後続のオフロード・ライト動作において、クライアントによって使用することができる。実施形態では、トークンは、サーバーおよびクライアントに跨がって安全にそして確実に使用することができる不変のデータを表す。

【選択図】 図 1

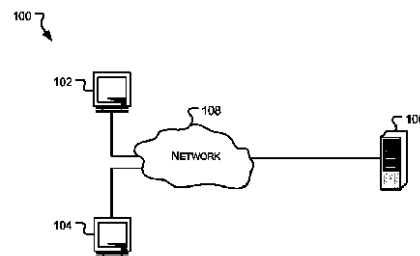


FIG. 1

【特許請求の範囲】**【請求項 1】**

トークン・ベース・ファイル動作を行うコンピューター実装方法であって、

ファイル・システムにおける情報にアクセスするために、ファイル・サーバーにおいて、前記ファイル・サーバーに接続することを求める第 1 要求を受けるステップと、

前記ファイル・サーバーから第 1 応答を送るステップであって、前記応答が、前記ファイル・システムにおける前記情報へのアクセスを可能にするために、クライアントとのセッションを作る、ステップと、

前記ファイル・サーバーにおいて、前記ファイルからのファイル情報にアクセスするために前記ファイル・システムにおいてファイルを開くことを求める第 2 要求を受けるステップと、

前記第 2 要求を受けたことに応答して、前記ファイル・サーバーが前記クライアントに第 2 応答を送り、前記ファイルへのアクセスを付与するステップと、

前記ファイル・サーバーにおいて、前記ファイルの一部からの第 1 データのオフロード・リードを求める第 3 要求を受けるステップであって、前記第 3 要求が、ファイル・アクセス・プロトコルにしたがってフォーマットされる、ステップと、

前記第 3 要求を受けたことに応答して、前記ファイル・サーバーが、前記第 1 データを表すトークンと共に第 3 応答を送るステップであって、前記第 1 データが論理的に前記ファイルの前記一部から読み出され、前記第 3 応答が前記ファイル・アクセス・プロトコルにしたがってフォーマットされる、ステップと、

を含む、方法。

【請求項 2】

請求項 1 記載の方法において、前記第 3 要求が、前記ファイルから読み出される第 1 データと、前記ファイルから読み出される第 2 データとを示し、前記第 3 応答が、前記第 1 データを表す前記トークンと、前記第 2 データを表す第 2 トークンとを含む、方法。

【請求項 3】

請求項 1 記載の方法において、前記第 3 要求が、第 1 ファイルの第 1 部分と第 2 ファイルの第 2 部分とを示し、前記第 3 応答が、前記第 1 ファイルの前記第 1 部分から論理的に読み出された第 1 データを表す前記トークンと、前記第 2 ファイルの前記第 2 部分から論理的に読み出された第 2 データを表す第 2 トークンとを含む、方法。

【請求項 4】

請求項 1 記載の方法であって、更に、

前記ファイル・サーバーにおいて、前記第 1 データの要求された部分の第 2 ファイルへのオフロード・ライトを求める第 4 要求を受けるステップであって、前記第 4 要求が、前記トークンを含み、前記ファイル・アクセス・プロトコルにしたがってフォーマットされる、ステップと、

前記第 4 要求を受けたことに応答して、前記ファイル・サーバーが、

前記第 1 データの前記要求された部分を前記第 2 ファイルに書き込むステップと、

前記第 1 データの前記要求された部分が前記第 2 ファイルに書き込まれたことを示す第 4 応答を送るステップであって、前記第 4 応答が前記ファイル・アクセス・プロトコルにしたがってフォーマットされる、ステップと、

を含む、方法。

【請求項 5】

請求項 1 記載の方法であって、更に、

前記ファイル・サーバーにおいて、前記第 1 データの要求された部分の第 2 ファイルへのオフロード・ライトを求める第 4 要求を受けるステップであって、前記第 4 要求が前記トークンを含み、前記ファイル・アクセス・プロトコルにしたがってフォーマットされる、ステップと、

前記第 4 要求を受けたことに応答して、前記ファイル・サーバーが、

前記第 1 データーの前記要求された部分の内第 1 部分を前記第 2 ファイルに書き込むステップであって、前記要求された部分の前記第 1 部分が、前記第 1 データーの前記要求された部分の全て未満である、ステップと、

前記第 1 データーの前記要求された部分の内前記第 1 部分が前記第 2 ファイルに書き込まれたことを示す第 4 応答を送るステップであって、前記第 4 応答が、前記ファイル・アクセス・プロトコルにしたがってフォーマットされる、ステップと、
を含む、方法。

【請求項 6】

コンピューター実行可能命令を含むコンピューター読み取り可能記憶媒体であって、前記コンピューター実行可能命令をプロセッサによって実行すると、トークン・ベース・ファイル動作を要求する方法を実行し、前記方法が、

ファイル・システムにおける情報にアクセスするために、ファイル・サーバーに接続することを求める第 1 要求をクライアントによって送るステップと、

第 1 応答を受信するステップであって、前記応答が、前記ファイル情報へのアクセスを可能にするために前記クライアントとのセッションを作る、ステップと、

前記ファイル・システムにおいてファイルを開くことを求める第 2 要求を送るステップと、

前記ファイルへのアクセスを付与する第 2 応答を受信するステップと、

トークンによって表されるデーターの第 1 部分のファイルへのオフロード・ライトを求める第 3 要求を送るステップであって、前記第 3 要求が、サーバー・メッセージ・ブロック (SMB) プロトコルのバージョンにしたがってフォーマットされ、前記データーを表す前記トークンを含む、ステップと、

応答を受信するステップと、

を含む、コンピューター読み取り可能記憶媒体。

【請求項 7】

請求項 6 記載のコンピューター読み取り可能記憶媒体において、前記応答が、前記データーの第 2 部分が前記ファイルに首尾良く書き込まれたことを示し、前記第 2 部分が、前記第 1 部分におけるデーターの全て未満を表す、コンピューター読み取り可能記憶媒体。

【請求項 8】

請求項 6 記載のコンピューター読み取り可能記憶媒体であって、更に、

第 2 ファイルの一部からの第 2 データーのオフロード・リードを求める第 4 要求を送るステップであって、前記第 4 要求が、前記 SMB プロトコルの前記バージョンにしたがってフォーマットされる、ステップと、

前記第 2 データーを表すトークンと共に第 4 応答を受信するステップであって、前記第 4 応答が、前記 SMB プロトコルの前記バージョンにしたがってフォーマットされる、ステップと、

を含む、コンピューター読み取り可能記憶媒体。

【請求項 9】

トークン・ベース・ファイル動作を可能にするシステムであって、

少なくとも 1 つのサーバーを含み、このサーバーが、

コンピューター実行可能命令を実行するように構成された少なくとも 1 つのプロセッサと、

前記コンピューター実行可能命令を格納する少なくとも 1 つのコンピューター読み取り可能媒体と、

を含み、前記少なくとも 1 つのプロセッサによって前記コンピューター実行可能命令を実行すると、ファイル・サーバーを設け、このファイル・サーバーが、

ファイルの一部からデーターのオフロード・リードを求める要求を受け、前記要求が、サーバー・メッセージ・ブロック (SMB) プロトコルのバージョンにしたがってフォーマットされ、

前記要求を受けたことに応答して、前記ファイル・サーバーが、前記データーを表すト

10

20

30

40

50

ークンと共に応答を送り、前記応答が、前記 S M B プロトコルの前記バージョンにしたがってフォーマットされる、
ように構成された、システム。

【請求項 10】

請求項 9 記載のシステムにおいて、前記システムが、更に、
少なくとも 1 つのクライアントを含み、このクライアントが、
コンピューター実行可能命令を実行するように構成された少なくとも 1 つのプロセッサと、

前記コンピューター実行可能命令を格納する少なくとも 1 つのコンピューター読み取り可能記憶媒体と、

を含み、前記少なくとも 1 つのプロセッサによって前記コンピューター実行可能命令を実行すると、

前記ファイルの前記一部からのデータの前記オフロード・リードを求める前記要求を送り、

前記データを表す前記トークンと共に前記応答を受信する、システム。

【発明の詳細な説明】

【従来技術】

【0001】

[0001] 大量のデータをコピーする従前の方法では、データはソース・ファイルからローカル R A M に読み込まれ、次いで同じバイトが R A M から逆に宛先ファイルに書き込まれる。

【発明の概要】

【発明が解決しようとする課題】

【0002】

このプロセスでは、コピーの完了にはデータがローカル R A M に入ることさえも本質的に必要でない場合でも、データがローカル R A M を含む経路を進むことが必要になる。最終的なソースと最終的な宛先との間でデータが辿ることができる経路があり、この経路が信頼があってこちらの方が速い場合、ローカル R A M を通る回り道は不要になる。この問題は、信頼があってより速い経路とローカル R A M 経由の経路との間における速度差が大きいと一層深刻になる。現在、サーバー・メッセージ・ブロック (S M B) ファイル・サーバーのような、一部のファイル・サーバーは、ソース・ファイルの範囲を宛先ファイルの範囲にコピーすることを、クライアントに許可している。しかしながら、同じファイル・サーバーにおいて開いているファイル間でデータをコピーするときの限度というような、複数の制限がある。また、 S M B ファイル・サーバーは、ソースおよび宛先範囲双方を指定する 1 つのコマンドを発行することしか、クライアントに許可しない。多くの場合、コピーを行うにはリードおよびライトを別々に使用するようにクライアント・コード構造が設定されるが、 S M B ファイル・サーバーがデータの範囲をコピーするために設ける現在の方法とは調和しない。

【0003】

[0002] 実施形態が作られたのは、これらおよびその他の考察に関してである。また、比較的具体的な問題について説明したが、実施形態は、背景において特定した具体的な問題を解決することに限定されるのではないことは言うまでもない。

【課題を解決するための手段】

【0004】

[0003] この摘要は、詳細な説明において以下で更に説明する概念から選択したものを、簡略化した形態で紹介するために設けられている。この摘要は、特許請求する主題の主要な特徴や必須の特徴を特定することを意図するのではなく、特許請求する主題の範囲を判断するときに補助として用いられることを意図するのでもない。

【0005】

[0004] トーク・ベースのファイル動作を可能にする実施形態について記載する。この

10

20

30

40

50

実施形態は、クライアントがファイル・サーバーとのセッションを作ること考慮に入れている。セッションは、任意のファイル・アクセス・プロトコルを使用して作ることができ、一例にはサーバー・メッセージ・ブロック (SMB) プロトコルが含まれる。セッションが作られた後、クライアントはこのファイル・アクセス・プロトコルにしたがってフォーマットされた特殊なオフロード・ファイル動作を要求することができる。このファイル動作は、リード動作でもライト動作でもよい。オフロード・リード動作では、クライアントは、ファイル・サーバーにアクセス可能なファイル記憶システムに格納されているファイルからファイル・データーを読み出すことを要求する。その代わりに、ファイル・サーバーは、そのデーター・ファイルを表すトークンを含む応答を供給する。実施形態の中には、何らかの理由でファイル・データーの全てを表すトークンを供給することができない場合、ファイル・サーバーがファイル・データーの全て未満を表すトークンと共に応答を戻す場合もある。次いで、トークンは、後続のオフロード・ライト動作、または他の関連動作 (例えば、必要になった場合、後にトークンによって表されるデーターを得る) において、クライアントによって使用することができる。実施形態では、トークンは、サーバーおよびクライアントに跨がって安全にそして確実に使用することができる不変のデーターを表す。

10

【0006】

【0005】 実施形態は、コンピューター・プロセス、計算システム、あるいはコンピューター・プログラム製品またはコンピューター読み取り可能媒体のような製造品目として実現することができる。コンピューター・プログラム製品は、コンピューター・システムによって読み取ることができ、コンピューター・プロセスを実行するための命令のコンピューター・プログラムをエンコードしたコンピューター記憶媒体とすることができる。また、コンピューター・プログラム製品は、計算システムによって読み取り可能であり、コンピューター・プロセスを実行するための命令のコンピューター・プログラムをエンコードした、搬送波上の伝搬信号であることもできる。

20

【図面の簡単な説明】

【0007】

【0006】 以下の図を参照して、非限定的かつ非網羅的な実施形態について説明する。

【図1】 図1は、実施形態を実現するために使用することができるシステムを示す。

【図2】 図2は、実施形態にしたがって、ファイル・アクセス・プロトコルを使用するトークン・ベース・ファイル動作に関与するクライアントおよびサーバーのブロック図を示す。

30

【図3】 図3は、実施形態にしたがって、オフロード・ファイル動作を処理する動作フローを示す。

【図4】 図4は、実施形態にしたがってオフロード・リード要求を処理する動作フローを示す。

【図5】 図5は、実施形態にしたがってオフロード・ライト要求を処理する動作フローを示す。

【図6】 図6は、実施形態にしたがってオフロード・ファイル動作を要求する動作フローを示す。

40

【図7】 図7は、実施形態を実現するのに適した計算環境のブロック図である。

【発明を実施するための形態】

【0008】

【0014】 添付図面を参照して種々の実施形態について更に詳しく説明する。添付図面は、本明細書の一部をなし、具体的な実施形態例を示す。しかしながら、実施形態は、多くの異なる形態でも実現することができ、本明細書において示される実施形態に限定されるように解釈してはならない。むしろ、本開示が周到で完全であり、実施形態の範囲を全て当業者に伝えるように、これらの実施形態が提示されるのである。実施形態は、方法、システム、またはデバイスとして実施することもできる。したがって、実施形態は、ハードウェアの実現例、全体的にソフトウェアの実現例、またはソフトウェアおよびハードウェア

50

アの形態を組み合わせた実現例という形態をなすことができる。以下の詳細な説明は、したがって、限定的な意味で捕らえてはならない。

【 0 0 0 9 】

[0015] 図 1 は、実施形態を実現するために使用することができるシステム 1 0 0 を示す。システム 1 0 0 は、クライアント 1 0 2 および 1 0 4 ならびにサーバー 1 0 6 を含む。クライアント 1 0 2 および 1 0 4 は、ネットワーク 1 0 8 を介して、サーバー 1 0 6 と通信する。サーバー 1 0 6 は、クライアント 1 0 2 および 1 0 4 におけるアプリケーションによってアクセスされる情報を格納する。クライアント 1 0 2 および 1 0 4 は、クライアント 1 0 2 および 1 0 4 における情報にアクセスするために、サーバー 1 0 6 とのセッションを作る。図 1 ではサーバー 1 0 6 と通信するものとしてクライアント 1 0 2 および 1 0 4 のみが示されているが、他の実施形態では、2 つよりも多いクライアントがサーバー 1 0 6 からの情報にアクセスすることもできる。

10

【 0 0 1 0 】

[0016] 実施形態では、クライアント 1 0 2 および 1 0 4 におけるアプリケーションが、アプリケーションに対して透過的なファイル・システムに、ファイル情報を要求する。ファイル情報は、サーバー 1 0 6 におけるファイル・システムから引き出される。一実施形態では、サーバー 1 0 6 におけるこのようなファイル・システムは、リモート・ファイル・システムである。他の実施形態では、サーバー 1 0 6 におけるファイル・システムは分散型ファイル・システムである。本明細書において開示する実施形態にしたがって、本開示の主旨および範囲から逸脱することなく、複数のタイプのファイル・システムを使用することができる。更に、図示されていないが、実施形態の中には、1 つのサーバー 1 0 6 の代わりに、サーバーが、例えば、サーバー・クラスターの一部である複数のサーバーの内の 1 つであってもよい場合もある。他の実施形態では、サーバーは、サーバー・クラスターの一部ではない複数のサーバーの内の 1 つでもよい。このような実施形態における複数のファイル・サーバーは、クライアント 1 0 2 および 1 0 4 に対して、冗長性および情報、例えば、ファイル情報の高い可用性を提供する。

20

【 0 0 1 1 】

[0017] 一実施形態では、クライアント 1 0 2 および 1 0 4 は、サーバー 1 0 6 におけるリモート・ファイル・システムに格納されているファイルに対して実行される複数のファイル動作を送ることができる。クライアント 1 0 2 および 1 0 4 は、ファイルに対して実行されるファイル動作の要求をフォーマットするために、ファイル・アクセス・プロトコルを使用する。このファイル・アクセス・プロトコルは、ネットワーク・ファイル・システム (N F S)、またはサーバー・メッセージ・ブロック (S M B) プロトコルのバージョンというような、任意のしかるべきプロトコルであればよい。実施形態によれば、クライアントは、正規のリードおよびライト・ファイル動作を送ることに加えて、オフロード・リード動作およびオフロード・ライト動作も要求することができる。これらは、トークン・ベース動作である。以下で更に詳しく説明するが、オフロード・ファイル動作は、実際のデータをネットワークを介してクライアント 1 0 2 または 1 0 4 のいずれかに送るローカル R A M に送る (transfer) ことを必要とせずに、大量のデータをクライアント 1 0 2 および 1 0 4 によって動かすことを可能にする。

30

40

【 0 0 1 2 】

[0018] 一実施形態を例示すると、クライアント 1 0 2 がサーバー 1 0 6 とのセッションを作る要求を送ることができる。例えば、クライアント 1 0 2 は、サーバー・メッセージ・ブロック (S M B) プロトコルのバージョンを使用して、サーバー 1 0 6 に格納されているファイル・システムにアクセスするために、サーバー 1 0 6 とのセッションを作ることができる。セッションを作るには、クライアント 1 0 2 とサーバー 1 0 6 との間において送信される複数のネゴシエート要求および応答の交換が必要になる場合がある。S M B プロトコルのバージョンでは、具体的に定められたネゴシエート・パケットがあり、これらを使用して、セッションの間に使用されるプロトコルの正確なバージョンをネゴシエートし、更にクライアント、例えば、1 0 2 およびサーバー、例えば、1 0 6 の双方の

50

能力を互いに通知する(advertise)。一実施形態では、ネゴシエート・パケットは、サーバー 106 がトークン・ベース・ファイル動作、即ち、オフロード・リードおよびオフロード・ライト・コマンドを扱うことができることの指示を含むこともできる。これによって、クライアントは、所望であれば、オフロード・ファイル動作をサーバーに要求してもよいことを知ることができる。

【0013】

[0019] 以上の例を続けると、セッションが作られた後、クライアント 102 は、サーバー 106 におけるファイル・システムにおいてファイルを開くために、SMB プロトコルにしたがってフォーマットされたメッセージを送ることができる。サーバーは、ファイル開放のためのハンドルで応答することができる。次いで、クライアント 102 は、SMB プロトコルにしたがってフォーマットされたオフロード・リード動作を要求し、ファイルからのファイル・データーを要求することができる。一実施形態では、クライアントは、オフロード・リード動作においてファイルの一部からのデーターを要求する。オフロード・リード動作要求は、トークン・ベース・リード動作である。

【0014】

[0020] クライアント 102 からの要求に応答して、サーバー 106 は、SMB プロトコルにしたがってフォーマットされた応答を、クライアント 102 が要求したファイル・データーを表すトークンと共に送る。実施形態の中には、サーバー 106 が、トークンを生成し、同じファイル・データーを要求しているかもしれないクライアント 104 のような、他のクライアントからの任意の要求にわたって、トークンがファイル・データーを一貫して表すことを保証する役割を担うことができる場合もある。他の実施形態では、ファイル・サーバーはクライアントからの任意の要求を、基礎ファイル記憶システムに渡すでもよい。これらの実施形態では、基礎ファイル記憶システムが、クライアント 102 によって要求されたファイル・データーを表すトークンを生成する役割を担う。いずれの実施形態でも、サーバー 106 は、クライアント 102 に対する応答をトークンと共に送る。トークンを生成するとき、実施形態は、例えば、トークンを作るために使用されるソース・ファイルの範囲が連続的でない場合でも、トークンが作られてもよいことを考慮に入れている。このような実施形態では、このようなソース範囲からのデーターは、論理的に繋がれ、トークンによって表される 1 つの論理範囲のデーターとなる。実施形態の中には、実現例が内部的にトークンを具体的なソース範囲と関連付けるとよい場合もあり、このようなソース範囲は互いに連続的でなくてもよい。

【0015】

[0021] トークンは、不変のデーター、即ち、クライアント 102 が要求するファイル・データーを表す。したがって、クライアント 102 は、サーバー 106 によって戻されたトークンを使用して、他のファイル動作を実行することができる。例えば、後の時点において、クライアント 102 は、トークンを使用してデーターを他のファイルに書き込むことができる。この例では、クライアント 102 は、SMB プロトコルにしたがってフォーマットされたオフロード・ライト動作を要求することができる。このオフロード・ライト動作もトークン・ベース・ファイル動作である。このオフロード・ライト動作は、クライアント 102 に以前に供給されたトークンを含むことができる。オフロード・ライト動作は、トークンによって表されるファイル・データーを、サーバー 106 における他のファイルに書き込むことを要求することができる。この要求を受けたことに応答して、サーバー 106 は、一実施形態では、次に要求を処理して、トークンによって表されるファイル・データーをサーバー 106 における他のファイルに書き込む。他の実施形態では、要求を受けたことに応答して、サーバー 106 は最初に受信したトークンの有効性を判断し、トークンが有効である場合、次にトークンによって表されるファイル・データーを、サーバー 106 における他のファイルに書き込む。先に注記したように、トークンが基礎ファイル記憶システムまたはそれよりも下位のレイヤー(1つまたは複数)によって生成される実施形態では、サーバー 106 は単にオフロード・ライト要求を基礎ファイル記憶システムに渡すだけであり、次いで基礎ファイル記憶システムがこの要求を処理して、ファイ

ル・データーを他のファイルに書き込む。サーバー 106 は、次に、応答をクライアント 102 に送り、オフロード・ライトが成功であったか否か示す。「ファイル」データーに言及するが、他の実施形態は、トークンが任意のタイプのデーターを表すことを考慮に入れている。例えば、実施形態は、トークンが、ファイル、ボリューム、ディスク、ボリューム・スナップショット、ディスク・スナップショット、プロブ・ストア等のような、任意の記憶コンテナから得られたものであることを考慮に入れている。「ファイル・データー」という用語は、本明細書では、例示の目的に限って使用されているのであり、限定を意図するのではない。更に、以上で説明した実施形態は、オフロード・ライト動作が、トークンによって表されるファイル・データーをサーバー 106 における他のファイルに書き込むことを要求することを考慮に入れているが、他の実施形態は、オフロード・ライト動作が、トークンによって表されるファイル・データーの一部をサーバー 106 における他のファイルに書き込むことを要求することを考慮に入れている。

10

【0016】

[0022] 加えて、以上で説明した実施形態は、オフロード・ライト動作が、以前にクライアント 102 に供給されたトークンを含むことを考慮に入れているが、他の実施形態は、クライアント 102 が、周知のトークンを、オフロード・ライト動作でのトークンとして使用することを考慮に入れている。例えば、ゼロ・トークンのような周知のトークンが、以前の対応するオフロード・リードが全くななくても、データー（ゼロのような）を書き込むために使用されてもよい。

【0017】

20

[0023] 実施形態の中には、ファイル・サーバー 106 がクライアントによって要求されたファイル・データーの全てに対してトークンを作ることができない場合もある。これが起こり得るのは、例えば、クライアント 104 のような他のクライアントが、クライアント 102 が要求したファイル・データーのファイルのある部分にロックをかけた場合である。これらの実施形態では、サーバー 106 は短縮応答(truncated response)を送ることもできる。即ち、応答は、クライアント 102 が要求したファイル・データーの一部のみを表すトークンを含む。つまり、実施形態では、この応答が示すのは、トークンが、オフロード・リード要求において要求された第 1 データーの全て未満を表すということである。実施形態の中には、このトークンが不連続なデーター範囲を表すことができる場合もある。しかしながら、他の実施形態では、トークンは連続するデーター範囲を表すことができるが、オフロード・リード要求においてクライアント 102 が要求したファイル・データーよりも少ない場合もある。これらの実施形態では、サーバーの応答は、短縮応答において送られたトークンによって表されるファイル・データーの部分の指示を含む。

30

【0018】

[0024] 実施形態の中には短縮オフロード・リード応答を考慮に入れているものもあるが、本開示の実施形態は短縮オフロード・ライトも考慮に入れている。先に説明したように、オフロード・ライト動作は、関連するトークンによって表されるデーターが、サーバー、例えば、サーバー 106 における他のファイルに書き込まれることを要求することができる。この要求を受けたことに応答して、サーバー 106 は次にこの要求を処理して、トークンによって表されるファイル・データーをサーバー 106 における他のファイルに書き込む。実施形態では、サーバー 106 は、トークンによって表されるデーターの全てを他のファイルに書き込むことができない。その結果、実施形態では、オフロード・ライトは短縮される場合があり、「書き込み長」は、例えば、書き込むことを要求された長さ未満になる。例えば、サイズの制約のために、トークンによって表されるデーターの全てを他のファイルに書き込む能力が制限される場合がある。他の実施形態では、ファイルの一部に書き込みのロックがかけられていると、このファイルの部分が書き込まれることが妨げられるであろう。また、処理エラー、または他のタイプのエラーも、トークンによって表されるデーターの一部が他のファイルに書き込まれない原因になり得る。更に、トークンが部分的に転化されていること、または部分的に無効であることもあり、その場合サーバー 106 はこのデーターの転化した部分 / 無効の部分を他のファイルに首尾良く書き

40

50

込むことはできない。本明細書に開示される実施形態によれば、本開示の主旨や範囲から逸脱することなく、他の理由でも、トークンによって表されるデータの全てが他のファイルに書き込まれることが妨げられることがある。短縮オフロード・ライトを必要とする実施形態では、短縮オフロード・ライト応答をサーバー 106 からクライアント 102 に送り、例えば、データの一部分が他のファイルに書き込まれなかったことを示すことができる。このような指示は、実施形態では、例えば、フラグまたは他のインディケーターの使用によって行うことができる。更に、実施形態では、短縮オフロード・ライト応答は、実際に書き込まれたデータ量を示す。

【0019】

[0025] 以上の説明は、図 1 に示した実施形態がどのように動作することができるかについての一例に過ぎない。以下で更に詳しく説明するように、実施形態は異なるステップまたは動作を伴うこともできる。これらは、任意の適したソフトウェアまたはハードウェア・コンポーネントを使用して実現することができる。

【0020】

[0026] これより図 2 に移ると、図 2 は、クライアント 202、クライアント 204、サーバー 206、およびサーバー 208 を含むソフトウェア環境 200 のブロック図を示す。また、ファイル情報が格納されるファイル・ストレージ 210 も示されている。

【0021】

[0027] 図 2 に示すように、クライアント 202 およびクライアント 204 は、各々、ファイル情報を要求することができるアプリケーションを含む。このアプリケーションは、例えば、ワープロ・アプリケーション、表計算アプリケーション、ブラウザ・アプリケーション、またはファイルに対するアクセスを要求する任意の他のアプリケーションとすることができる。図 2 に示す実施形態では、ファイルは、ファイル・ストレージ 210 内部に格納されているファイル・システムに配置される。図 2 は、本明細書において開示する一実施形態にしたがって、サーバー 206 および 208 に共有ストレージ能力を与えるファイル・ストレージ 210 を示すが、他の実施形態は他の記憶手段を有する。例えば、サーバー 206 およびサーバー 208 は、各々、それら自体の記憶手段を有することもでき、この記憶手段は、実施形態にしたがって、着脱型(detached)でも据え付け型(attached)でもよい。更に他の実施形態では、サーバー 206 およびサーバー 208 は、各々、それら自体の記憶手段を有し、ファイル・ストレージ 210 の使用によって記憶能力を共有させることもできる。本明細書において開示する実施形態によれば、本開示の主旨および範囲から逸脱することなく、複数のタイプのストレージを使用することができる。更に、クライアント 202 およびクライアント 204 は、各々、アプリケーションからのファイル要求をファイル・サーバーにリディレクトするリディレクターも含み、ファイル・サーバーはリモート・ファイル・システムへのアクセスを与える。リディレクターは、ファイル・アクセス・プロトコルを使用して、ファイル・サーバーと通信する。実施形態の中には、ファイル・アクセス・プロトコルが NFS または SMB プロトコルの一バージョンであるとよい場合がある。例示の目的に限って、クライアント 202 およびクライアント 204 におけるリディレクターが、SMB 2 のような、SMB プロトコルの一バージョンを使用してファイル・サーバーと通信することを仮定して、図 2 について説明する。しかしながら、実施形態は SMB プロトコルの使用に限定されるのではない。

【0022】

[0028] 図 2 では、サーバー 206 および 208 は、各々がファイル・サーバーを含むように示されている。先に注記したように、ファイル・サーバーは、クライアント 202 およびクライアント 204 におけるリディレクターと通信するために、SMB プロトコルの一バージョンを使用することができる。また、サーバー 206 および 208 の各々は、ファイル・データを表すトークンを生成するトークン生成モジュールも含む。加えて、ファイル・ストレージ 210 も、ファイル・データを表すトークンを生成するトークン生成モジュールを含む。

【0023】

10

20

30

40

50

【0029】 クライアントとサーバーとの間においてセッションを作るためのSMBプロトコルの使用は、クライアント202におけるリディレクターのようなリディレクターがネゴシエート要求をサーバー206のようなファイル・サーバーに送ることから開始する。リディレクターおよびファイル・サーバーは、セッションに使用されるSMBのバージョンをネゴシエートするために、ネゴシエート・パケットを交換する。加えて、このネゴシエーションの間に、能力(capabilities)も交換することができる。一実施形態では、サーバー206は、ファイル・サーバーからクライアントに送られるネゴシエート応答パケット内に能力フラグを含み、ファイル・サーバーがオフロード・ファイル動作の使用をサポートすることを、クライアントに示すことができる。他の実施形態では、クライアント202およびサーバー206は、単にSMBプロトコルのバージョンをネゴシエートすることもでき、このとき、当該バージョンがオフロード・ファイル動作の使用に対するサポートを含むことを理解している。更に他の実施形態では、オフロード・ファイル動作が試されるときに、プロトコルの一バージョンがオフロード・ファイル動作をサポートするという判断が行われる。例えば、クライアント202がオフロード・リード(またはオフロード・ライト)動作を要求するとしてもよい。サーバー206がオフロード・ファイル動作をサポートする場合、サーバー206はこの要求の処理を進める。サーバー206がオフロード・ファイル動作をサポートしない場合、サーバー206は、要求されたオフロード・ファイル動作を実行できないことを示す応答をクライアント202に送る。例えば、一実施形態では、サーバー206がオフロード・ファイル動作をサポートしない場合、サーバー206は、エラー・メッセージおよび/またはこれを示すフラグでクライアント202に応答する。

【 0 0 2 4 】

【0030】一旦ネゴシエーションが完了すると、クライアント２０２におけるリディレクターおよびファイル・サーバー２０６はセッションを作る。クライアント・リディレクターは、次いで、ファイル・アクセス要求をファイル・サーバーに送ることができる。一実施形態では、クライアント２０２におけるリディレクターは、ファイルにおいて開放を要求する。サーバー２０６は、この開放に対するハンドルを含む応答を供給する。次いで、クライアント２０２は、このハンドルを使用して、オフロード・リード動作を要求することができる。実施形態では、オフロード・リード動作は、ＳＭＢプロトコルにしたがってフォーマットされる。実施形態の中には、オフロード・リードおよびオフロード・ライト動作が、ＳＭＢプロトコル接続を使用して送られることもあり、他のファイル・システム制御コマンド（ＦＳＣＴＬ）と同様に、コマンドをＳＭＢ２入力／出力制御（ＩＯＣＴＬ）要求内にカプセル化する。以下に、実施形態においてオフロード・リード動作を要求するために使用することができる構造の一例を示す。

【 0 0 2 5 】

【表 1】

```
typedef struct _FSCTL_OFFLOAD_READ_INPUT {
    ULONG Size;
    ULONG Flags;
    ULONG TokenTimeToLive; // In milliseconds
    ULONG Reserved;
    ULONGLONG FileOffset;
    ULONGLONG CopyLength;
} FSCTL_OFFLOAD_READ_INPUT, *PF_FSCTL_OFFLOAD_READ_INPUT;
```

【 0 0 2 6 】

[0031] 先に示したように、オフロード・リード動作を要求するためにクライアントに

よって使用される構造は、複数のフィールドを含むことができる。実施形態では、サーバーに対する存続時間(time to live)示唆(suggestion)を含むことができる。言い換えると、このフィールドは、サーバーが送るトークンに対して示唆される寿命を示すことができる。また、この構造は、クライアントが要求するファイル・データーのファイル・オフセットおよびコピー長も含む。

【 0 0 2 7 】

[0032] この要求に回答して、サーバー 2 0 6 は回答を返送する。以下に、オフロード・リード動作に対して回答するために使用することができる構造の一例を示す。

【 0 0 2 8 】

【表 2】

10

```
typedef struct _FSCTL_OFFLOAD_READ_OUTPUT {
    ULONG Size;
    ULONG Flags;
    ULONGLONG TransferLength;
    UCHAR Token[512];
}FSCTL_OFFLOAD_READ_OUTPUT, *PFSTL_OFFLOAD_READ_OUTPUT;
```

【 0 0 2 9 】

[0033] 先に示したように、この回答は、実施形態では、クライアント 2 0 2 が要求したファイル・データーを表すトークンを含む。また、このトークンは、実施形態では、当該トークンによって表されるファイル・データーの長さも含む。

20

【 0 0 3 0 】

[0034] 実施形態の中には、サーバー 2 0 6 が、オフロード・リード要求に対して、失敗、即ち、このリード要求が十分に処理されなかったことの指示で回答することができる場合がある。以下に、追加情報をクライアントに示すためにオフロード・リード応答においてサーバーによって設定することができる 3 つのフラグを示す。

【 0 0 3 1 】

【表 3】

```
#define OFFLOAD_READ_FLAG_ALL_ZERO_BEYOND_CURRENT_RANGE
```

30

(1)

```
#define OFFLOAD_READ_FLAG_FILE_TOO_SMALL (2)
```

```
#define OFFLOAD_READ_FLAG_CANNOT_OFFLOAD_BEYOND_CURRENT_RANGE (4)
```

【 0 0 3 2 】

[0035] 以上に示す 2 番目のフラグは、ファイルが小さすぎたために要求が失敗したことを示す。小さなファイルに対するファイル・データー・ペイロードが、別個のクラスターに格納されるのではなく、直接ファイル・レコードに格納されるという状況がある可能性がある。このようなファイルに対しては、対応する小さなオフロード・リードから得られる効率は殆どないので、このようなファイルに対する小さなオフロード・リードを扱うのではなく、ソース・ファイル・システムは、オフロード・リードをやり過ごす(fail)ことができ、応答は、先に定められた 2 番目のフラグを含むことができる。このフラグは、ファイル・データーが小さすぎるために要求が失敗したことを示す。先に定められた最初のフラグは、オフロード・リード応答によって示される転送長(transfer length)を超える残りのデーターは、全てゼロを含むことを示す。最後に、3 番目のフラグは、サーバーが、オフロード・リード要求によって示される転送長を超えるオフロード・リード要求は成功しないと判断した状況において使用することができる。3 番目のフラグは、クライアントが要求するデーターの内、現在のオフロード・リード応答によって示される範囲を超

40

50

える分に対して、サーバーがトークンを供給できないことを示す。図示されていないが、オフロード・リード応答は、戻される一部(sub-portion)が不連続データ（単にオフセット0からの連続バイト・カウントの代わりに）を表すことの指示というような、情報を提供することもできる。実施形態では、オフロード・リード要求は、オフロード・リードが成功することができる次の後続ソース・オフセットに関するヒントも含むことができる。以上で示した3つのフラグは当該フラグに対して数値を含むが、これらの数値は例示の目的に限って提供されたに過ぎない。他の数値または一般的な値も、実施形態にしたがって、本開示の主旨および範囲から逸脱することなく、使用することができる。

【0033】

[0036] 一旦クライアント202が、以前のオフロード・リード要求によってまたは何らかの他の手段によって、トークンを受信したなら、クライアント202は、オフロード・ライト要求をサーバー206に発行することができる。以下に、実施形態においてクライアントがオフロード・ライト要求を送るために使用することができる構造の一例を示す。

【0034】

【表4】

```
typedef struct _FSCTL_OFFLOAD_WRITE_INPUT {
    ULONG Size;
    ULONG Flags;
    ULONGLONG FileOffset;
    ULONGLONG CopyLength;
    ULONGLONG TransferOffset;
    UCHAR Token[512];
}FSCTL_OFFLOAD_WRITE_INPUT, *PF_FSCTL_OFFLOAD_WRITE_INPUT;
```

【0035】

[0037] 以上の例において示したように、オフロード・ライトの構造は、コピーする宛先ファイルのオフセット、コピーするデータの長さ、およびトークンによって表される、データ内においてコピーを開始するオフセットを含む。また、トークンも含まれ、サーバーまたは他の手段によって受信されたのでもよい。

【0036】

[0038] オフロード・ライト要求に回答して、サーバー206はオフロード・ライト要求を発行する。オフロード・ライト応答においてサーバー206によって使用されるための構造の一例を以下に示す。

【0037】

【表5】

```
typedef struct _FSCTL_OFFLOAD_WRITE_OUTPUT {
    ULONG Size;
    ULONG Flags;
    ULONGLONG LengthWritten;
}FSCTL_OFFLOAD_WRITE_OUTPUT,
*PF_FSCTL_OFFLOAD_WRITE_OUTPUT;
```

【0038】

[0039] 実施形態の中には、サーバー206がオフロード・ライト要求に失敗で応答することができる場合もある。例えば、オフロード・ライト動作が失敗する可能性があるの

は、要求されたファイルが小さいファイルである、例えば、定められたサイズ閾値未満であるときである。以下に、ファイル情報が小さすぎたために要求が失敗したことを示すために、オフロード・ライト応答においてサーバーによって設定することができるフラグの例を示す。先に注記したように、小さいファイルは、大きいファイルとは異なる方法でそのデータを格納する場合があり、このような小さなファイルに対して発行されたオフロード要求は、やり過ごす方が適していることもあり得る。何故なら、実際のファイル・データの代わりにトークンを使用することから殆ど効率が得られないからである。他の実施形態では、オフロード・ライト動作が失敗する可能性があるのは、オフロード・ライトの受信先とデータ・ソースとの間におけるリンクが遅いときである。このような実施形態では、ファイル・システムは、オフロード・ライト動作に応答するか否か決定するとき

に、リンク・ステータス・チェックを実行することができる。このような事例では、サーバー 206 はオフロード・ライト要求をやり過ごすことができる。更に他の実施形態では、サーバー 206 は、トークンが無効であることから、要求をやり過ごすこともできる。例えば、トークンが期限切れである場合、これは無効であろう。無効 / 期限切れのトークンのためにサーバーが要求をやり過ごす場合、サーバー 206 は、オフロード・ライト要求に対して失敗で応答することができ、この場合、例えば、トークンが無効または期限切れであったために要求が失敗したことを示すために、オフロード・ライト応答においてサーバーによってフラグ（フラグ例として以下で示すような）を設定することができる。このタイプのフラグは、このライト動作の単なる再試行はうまくいかないこと、そして新たなトークンを生成するためには読み出し直さなければならないことのヒントとして、クライアントに対して作用することができる（即ち、失敗は、リンクの一時的な遅さによるのではなく、例えば、トークンが無効であったためである）。代替実施形態では、所与のトークンがもはや有効でないことを示すために、具体的なステータスを戻すこともできる。つまり、実施形態は、トークンがもはや有効でないことを示すために、リターン・ステータスを使用することを考慮に入れており、一方他の実施形態は、このような指示を与えるためにフラグを使用することを考慮に入れている。更に他の実施形態は、このような指示を与えるために、フラグおよびリターン・ステータスの双方を使用することを考慮に入れている。例えば、トークンの期限切れによる短縮(truncations)のときに、エラー・コードの代わりに、同じトークン（1つまたは複数）を使用してオフロード・ライトの残りを再試行しても無駄であることを示すために、フラグ、例えば、OFFLOAD_WRITE_FLAG_TOKEN_INVALIDと共に、短縮値を含む成功を、FSCTLのコラーに戻す動作を行うことができる。本開示の主旨および範囲から逸脱することなく、本明細書において開示する実施形態によれば、複数の他のタイプの条件のために、オフロード・ライトの失敗に至ることもある。

10

20

30

【0039】

【表 6】

#define OFFLOAD_WRITE_FLAG_FILE_TOO_SMALL (1)

#define OFFLOAD_WRITE_FLAG_TOKEN_INVALID (2)

【0040】

40

【0040】 以上で示したフラグは、当該フラグに対する数値を含むが、これらの数値は例示の目的に限って提供されたに過ぎない。他の数値または一般的な値も、実施形態にしたがって、本開示の主旨および範囲から逸脱することなく、使用することができる。

【0041】

【0041】 更に、実施形態の中には、先に説明したように、オフロード・ライトが短縮されることを考慮に入れているものもあり、この場合、書き込み長は、例えば、書き込みを要求された長さよりも短い。このような実施形態では、サーバーは、短縮オフロード・ライト応答で、オフロード・ライト要求に応答し、書き込むことが要求されたデータの一部だけが実際に書き込まれたことを示す。

【0042】

50

【0042】 実施形態では、オフロード・リードおよびライト動作において使用されるトークンは、ある規格にしたがってフォーマットされる。例えば、小型コンピューター・システム・インターフェース（SCSI）規格は、一実施形態にしたがって、使用することができるトークン・フォーマットの定義を提供することができる。とりわけ、高速コンピューター・インターフェース仕様および/または規格を含む複数のタイプの規格も、本明細書において開示する実施形態にしたがって、本開示の主旨および範囲から逸脱することなく使用することができる。SCSI規格は一例として示したに過ぎない。標準的なフォーマットの使用により、異なるデータ・アクセス・プロトコルを使用する他のサーバーとトークンを相互動作可能にすることができる。

【0043】

10

【0043】 実施形態では、サーバー206は、以前にオフロード・リードが（サーバーからまたは任意のソースから）なかった場合でも、オフロード・ライト・コマンドに指定された周知のトークン値を認識し使用することができる。例えば、サーバー206は、ゼロを含む範囲を表す周知のトークンを戻すことができる。クライアント202は、これを、ソース・ファイルの基礎範囲がゼロであること、そしてこのトークンに関連するデータが全てゼロであることを示すものとして解釈する。他の実施形態では、正常に読み取られたときにファイル・システムがゼロを戻す場合のように、オフロード・リード要求に応答して、ゼロ・トークンを戻すこともできる。このような状況が発生する可能性があるのは、例えば、ファイルの疎な範囲が読み取られる場合である。更に他の実施形態では、サーバー206は、割り当てを取り消されたトークンのような、他の周知のトークンも受け入れることができる。

20

【0044】

【0044】 サーバーにおいてトークンを生成するためにトークン・ジェネレーターを使用する実施形態では、オフロード・リード要求から戻されるトークンは、情報をサーバー206に要求する任意のクライアントによって使用可能である。つまり、クライアント204はクライアント202からトークンを受信するのでもよく、そしてオフロード・ライト動作をサーバー206に要求するためにこのトークンを使用することができる。クライアント間でトークンが渡される実施形態では、クライアントはそれらが選択した任意のプロトコルまたはトランスポートによってこのようなトークンを渡すことができる。クライアント間でトークンを渡すモードは、トークン自体には関係ない。したがって、トークンはサーバー206との異なる接続に跨がって使用可能であり、これらの接続は異なるクライアントによって作ることができる。このように、サーバー206は、ファイル・ストレージ210がオフロード・リードおよびライト動作をサポートするか否かには関係なく、オフロード・リード要求および一部のオフロード・ライト要求にサービスする(service)ことができる。オフロード・ライトにおいてサーバー206に供給されるトークンは、クライアントが現在開いているハンドルを有するファイルから得られたものである必要はなく、更にクライアントがアクセスすることができるファイルから得られたものである必要もない。クライアントは、少なくともトークンが得られた時点においてそのトークンが得られたファイル（1つまたは複数）にアクセスすることができる他のクライアントを介して間接的にトークンを得ることもできる。1つの接続を介して1つの共有(share)、たとえば、ファイル共有から読み取られたオフロード・リードによってクライアントによって得られたトークンは、クライアントによって異なる共有または異なる接続に対して発行されたオフロード・ライトにおいて、首尾良く使用することができる。

30

40

【0045】

【0045】 実施形態では、トークンは複数のサーバーに跨がって使用可能であるとよい。即ち、トークンは、本来そのトークンが発生したサーバーでなくても、そのサーバーにおいて使用されてもよい。言い換えると、トークンがサーバー206、サーバー208、またはファイル・ストレージ210において生成されるとき、このようなトークンは、そのトークンを受け入れることを決定した任意のサーバーにおいて使用することができる。例えば、先の実施形態では、クライアント202によって送られたオフロード・リード要求

50

がファイル・ストレージ 2 1 0 に渡されるのでもよく、ファイル・ストレージ 2 1 0 は、このオフロード・リード要求に対してトークンを生成する。後の時点において、クライアント 2 0 2 がサーバー 2 0 8 に接続した場合、サーバー 2 0 6 へのその接続によって既に供給されているトークンを使用して、オフロード・ライト動作のような他の動作を実行することができる。この例では、サーバー 2 0 8 は任意のオフロード・ライト動作を、元来このトークンを作ったファイル・ストレージ 2 1 0 に渡す。このように、トークンは、複数のサーバーに跨がって使用可能にすることができる。この実施形態は、例えば、共有ストレージを使用するサーバー・クラスターがファイル・サービスをクライアントに提供するために使用される状況において使用することができる。

【 0 0 4 6 】

10

[0046] 認めることができるであろうが、環境 2 0 0 についての以上の説明は、本明細書において説明する実施形態を限定することは意図していない。図 2 およびその説明は、単に一部の実施形態の実現例を例示することを意図するに過ぎない。他の実施形態では、オフロード動作が 1 つ以上のファイルおよび 1 つ以上のトークンを伴ってもよい。つまり、実施形態は、図 2 において示され説明されたものには限定されない。例えば、オフロード・リードが、1 つのファイルまたは複数のファイルの複数のセグメントを、1 つのトークンまたは複数のトークンを含むオフロード・リード応答によって読み取ることを考慮に入れてもよい。同様に、実施形態の中には、オフロード・ライト動作が、1 つ以上のファイルに関連する 1 つ以上のトークンを識別できる場合もある。

【 0 0 4 7 】

20

[0047] 図 3、図 4、図 5、および図 6 は、実施形態による動作フロー 3 0 0 および 4 0 0 を示す。動作フロー 3 0 0 および 4 0 0 は、任意の適した計算環境において実行することができる。例えば、これらの動作フローは、図 1 および図 2 に示したようなシステムおよび環境によって実行することができる。したがって、動作フロー 3 0 0 および 4 0 0 の説明は、図 1 および図 2 のコンポーネントの内少なくとも 1 つに言及することもある。しかしながら、このような図 1 および図 2 のコンポーネントに対する言及はいずれも、説明のために過ぎず、図 1 および図 2 の実現例は、動作フロー 3 0 0 および 4 0 0 の非限定的な環境であることは理解されてしかるべきである。

【 0 0 4 8 】

[0048] 更に、動作フロー 3 0 0 および 4 0 0 は、特定の順序で順次図示および説明されるが、他の実施形態では、これらの動作は異なる順序で、複数回、および / または並列に実行することもできる。更に、実施形態によっては、1 つ以上の動作を省略すること、または組み合わせることもできる。

30

【 0 0 4 9 】

[0049] 実施形態では、図 3 に示すフロー 3 0 0 は、少なくとも部分的に、サーバー、例えば、サーバー 2 0 6 (図 2) において実行しているファイル・サーバーによって実行することができる。フロー 3 0 0 は、動作 3 0 2 において開始し、ファイル・サーバーへの接続要求が受け取られる。動作 3 0 2 において受け取られる要求は、ファイル・サーバーを介してアクセス可能なリモート・ファイル・システムに格納されているファイル情報にアクセスするために、ファイル・サーバーとのセッションを作る要求である。この要求は、クライアント、例えば、クライアント 2 0 2 および 2 0 4 (図 2) によって送ることができる。動作 3 0 2 の後、フロー 3 0 0 は動作 3 0 4 に移り、セッションが作られたことを示す応答が送られる。実施形態の中には、動作 3 0 2 および 3 0 4 において送られる要求および応答が、クライアントとサーバーとの間でセッションをネゴシエートするために交換される複数のメッセージの内一部であってもよい場合がある。メッセージの交換は、オフロード・ファイル動作にサービスするファイル・サーバーの能力を含む、能力の交換を含むことができる。

40

【 0 0 5 0 】

[0050] 動作フロー 3 0 0 は動作 3 0 4 から動作 3 0 6 に移り、ファイルを開く第 2 要求が受け取られる。この要求は、ファイル内部の情報にアクセスするために、クライアン

50

トによって送られる。動作 3 0 6 から、フローは動作 3 0 8 に移り、応答がクライアントに送られ、ファイルへのアクセスを付与する。この応答は、応答においてファイル・サーバーによって供給されるファイル識別子を含むことができる。

【 0 0 5 1 】

[0051] 次いで、フロー 3 0 0 は動作 3 1 0 に移り、オフロード動作の要求が受け取られる。このオフロード動作は、トークンによって表されるファイル・データーを要求するオフロード・リード動作、または宛先ファイルに書き込まれるファイル・データーを表すトークンを含むオフロード・ライト動作であってもよい。この動作がオフロード・リード動作である場合、フローは A に移り、図 4 に続く。

【 0 0 5 2 】

[0052] 図 4 に示すように、フロー 3 0 0 は判断 3 1 2 に移り、オフロード・リード動作において要求されるデーターを全てトークンによって表すことができるか否か判定を行う。この判断 3 1 2 は、実施形態では、複数の異なる判定を伴うことができる。例えば、オフロード・リード動作において要求されたデーターの任意の部分が、他のクライアントによる排他的使用のためにロックされているか否かについて、判定を行うことができる。これらの状況では、サーバーが、要求されたデーターの全てを表すトークンを供給することができない場合もある。判断 3 1 2 において、要求されたデーターの全てを 1 つのトークンによって表すことは不可能であるという判定が行われた場合、フローは N O を通って動作 3 1 4 に移り、トークンを含む短縮応答が送られる。この短縮応答は、応答において供給されているトークンが、オフロード・リード要求において要求されたデーターの全てを表すのではないことを示す。また、この応答は、応答におけるトークンによって表されるデーターの範囲も示すことができる。動作 3 1 4 の後、フローは 3 1 6 において終了する。

【 0 0 5 3 】

[0053] 判断 3 1 2 において、要求されたデーターの全てが 1 つのトークンによって表すことができるという判定が行われた場合、動作 3 1 8 において、オフロード・リード要求において要求されたファイル・データーの全てを表すトークンを含む応答が送られる。次いで、フロー 3 0 0 は 3 1 6 において終了する。

【 0 0 5 4 】

[0054] 実施形態の中には、サーバーがトークンの供給元でなくともよい場合もある。これらの実施形態では、破線で示す代替動作が、判断 3 1 2、動作 3 1 4、および / または動作 3 1 8 の代わりに実行されるとよい。破線で示す動作は、トークンの生成が下位レイヤー、例えば、基礎ファイル記憶レベル（1 つまたは複数）において行われる実施形態において実行される。これらの実施形態では、フロー 3 0 0 は、判断 3 1 2 の代わりに、クエリー 3 2 0 に移る。クエリー 3 2 0 において、オフロード・リード動作が短縮されるか否か判定を行う。オフロード・リードが短縮される場合、例えば、要求されたデーターの長さに対する調節が行われ、プロセス 3 0 0 は Y E S に進んで、調節または短縮 3 2 2 を行う。サーバーが全く調節を行わない場合、プロセス 3 0 0 は N O に進んで、要求を変更しないままにしておく（3 2 1）。次に、プロセス 3 0 0 は動作 3 2 3 に進み、オフロード・リード要求がサーバーから、ファイル記憶コンポーネントまたはトークンを生成する役割を担う他のモジュールというような、下位レイヤーに渡される。

【 0 0 5 5 】

[0055] 動作 3 2 3 の後、フロー 3 0 0 はクエリー 3 2 4 に移り、ファイル・ストレージ（またはトークンを生成する役割を果たす他のコンポーネント）が短縮リード動作を実行するか否か判定を行う。短縮が行われない場合、プロセス 3 0 0 は N O に進み下位レイヤーによって要求を処理する（3 2 5）。次いで、トークンを含む応答がサーバーによって受信され（3 2 6）、動作 3 2 6 において受信される応答は、オフロード・リード動作において要求されたファイル・データーの少なくとも一部を表すトークン（1 つまたは複数）を含む。このトークンは、ファイル・ストレージによって使用される任意のしるべきフォーマットにしたがってフォーマットすることができる。一実施形態では、トークン

は、予め定められた S C S I フォーマットにしたがってフォーマットされる。次いで、一実施形態にしたがって、トークン（１つまたは複数）を含む応答は、サーバーによってクライアント 329 に送られる。他の実施形態では、トークン（１つまたは複数）を含む応答は、直接ファイル・ストレージ、または他の下位レイヤーから、例えば、クライアントに送られる。

【0056】

[0056] クエリー 324 に戻り、先に説明したように、トークンを生成する役割を果たすファイル・ストレージまたは他のコンポーネントが、要求されたデータの全てを含む応答を供給するか否か判定を行う。例えば、一実施形態では、ファイル・ストレージが、オフロード・リード要求において要求されたデータの全てを供給することができない場合もある。先に説明したように、完全な読み取りを妨げる記憶コンテナに対するロックを含む複数の理由により、短縮オフロード・リード応答になる場合があり得る。ファイル・ストレージが要求されたデータの一部分のみを供給する場合、プロセス 300 は YES から動作 327 に進み、下位レイヤー、例えば、ファイル・ストレージによって要求が処理され、短縮応答が供給される。一実施形態では、次に、トークンを含む短縮応答が、クライアントに送る（329）ために、サーバーにおいて受信される（328）。他の実施形態では、トークンを含む応答は直接ファイル・ストレージまたはトークンを生成する役割を果たす他のコンポーネントからクライアントに渡される。実施形態では、受信された短縮応答 328 は、要求が１つ以上のレイヤーによって短縮されたことを示す。例えば、応答は、トークンが、オフロード・リード要求において要求されたデータの全て未満を表すことを示す。他の実施形態では、短縮応答 328 は、要求が短縮されたことの指示を与えない。次いで、フロー 300 は動作 316 において終了する。他の実施形態（図示せず）では、サーバーが、ファイル・ストレージから受信したときに、そして既にファイル・ストレージによってデータが短縮されていると判断した場合であっても、更に応答におけるデータを短縮することを決定することもできる。このような実施形態では、このような短縮は、動作 326 および 328 の後で、トークンを含む短縮応答をクライアント動作 329 に渡す前に行うとよい。認めることができようが、動作 320 ~ 329 は、サーバーが単に下位のトークン供給元への受け渡し役として動作するときのみ実行される。図 4 に示す実施形態では、トークン供給元はファイル記憶システムである。実施形態の中には、ファイル記憶システムが、次に、オフロード要求を更に下位のトークン供給元に渡す場合もある。実施形態では、任意のレイヤーが、要求を下位層に送る前または後に、短縮することができる。例えば、実施形態は、サーバーが要求を下位レイヤー、例えば、ファイル・ストレージに渡し、この下位レイヤーから応答を受信し、次いでトークン（１つまたは複数）を含む応答をクライアントに送る前に、サーバーが更に短縮を実行するか否か判定することを考慮に入れている。しかしながら、実施形態によっては、下位レイヤー（１つまたは複数）に送る前に要求を短縮する方が効率的である場合もある。先に説明したように、フロー 300 は、実施形態にしたがって実行することができる動作フローの一例に過ぎない。実施形態は、図 3 ~ 図 5 に関して行った具体的な説明には限定されず、追加の動作を含むこともできる。例えば、図示した動作ステップは、他のステップに組み合わせること、および／または並び替えることも可能である。更に、例えば、もっと少ないステップまたはもっと多いステップを使用することもできる。

【0057】

[0057] 再度図 3 を参照すると、動作 310 において、動作がオフロード・ライト動作である場合、フローは B に移り、図 5 に続く。認めることができようが、オフロード・ライト動作は、データを表すトークンを含む。トークンは、本開示の主旨および範囲から逸脱することなく、本明細書において開示する実施形態にしたがって、複数のタイプの記憶コンテナから得ることができる。例えば、トークンは、ファイル、ボリューム、ディスク、ボリューム・スナップショット、ディスク・スナップショット、プロブ・ストア等から得ることができる。一実施形態では、トークンは、ソース・ファイルからこのトークンにデータをコピーすることによって作られる。他の実施形態では、トークンは、ソース

10

20

30

40

50

・ファイルから、トークンに関連する保持エリアにデータをコピーすることによって、作られる。つまり、作られたトークンは、ソース・ファイルとは独立であり、論理的にデータのそれ自体の読み取り専用コンテナとなる。図5に示すように、フローは動作310から動作330に移り、オフロード・ライト動作においてトークンに関連するデータが特定される。

【0058】

[0058] 動作330においてデータが特定された後、フロー300はクエリー331に移り、全ての要求データを、例えば、サーバーによって、宛先ファイルに書き込むことができるか否か判定を行う。全ての要求データを書き込むことができる場合、プロセス300はYESに進み、トークンによって表されるデータを宛先ファイル332に書き込む。即ち、トークンによって表されるデータの要求された部分が、オフロード・ライト要求において要求された特定の位置に書き込まれる。成功または失敗を示す応答、そして実施形態によっては、宛先ファイルに書き込まれるデータ量が、動作334において、クライアントに送られる。次いで、フロー300は316において終了する。一方、要求データの全てを宛先ファイルに書き込むことができない場合、例えば、プロセス300はNOに進んで、動作336において短縮データを書き込む。この場合、要求されたデータの一部が書き込まれる。要求データの一部が書き込まれたことを示す短縮ライト応答が、次に、動作338においてクライアントに送られる。次いで、フロー300は316において終了する。

10

【0059】

[0059] 先に注記したように、実施形態の中には、サーバーがトークンの供給元でなくともよい場合もある。これらの実施形態では、図5において破線で示す代替動作を、動作330～338の代わりに実行することができる。破線の動作は、トークンの生成が下位レイヤー、例えば、ファイル記憶レベル以下で行われる実施形態において実行される。これらの実施形態では、フロー300は、動作330の代わりに、クエリー340に移る。クエリー340において、下位レイヤー、例えば、基礎ファイル・ストレージに渡す前に、サーバーにおいてオフロード・ライトが短縮されるか否か判定が行われる。オフロード・ライトが短縮される場合、プロセス300はYESに進み調節または短縮を行う(344)。サーバーが何の調節も行わない場合、フロー300はNOに進み、オフロード・ライト要求を変更しないままにしておく(342)。次に、プロセス300は動作346に進み、オフロード・ライト要求(トークンを含む)が、サーバーから、ファイル記憶コンポーネントまたはオフロード・ライト要求を扱う役割を担う他のモジュールのような、下位レイヤーに渡される。

20

30

【0060】

[0060] オフロード・ライト要求を下位レイヤー(1つまたは複数)に渡した後(346)、フロー300はクエリー348に進み、下位レイヤー、例えば、ファイル・ストレージが短縮ライト動作を実行するか否か判定を行う。短縮が行われない場合、プロセス300はNOから動作350に移り、下位レイヤーがライト要求を処理する。この処理は、例えば、要求におけるトークンに関連するデータを特定し、トークンによって表されるデータを宛先ファイルに書き込むことを含む。下位レイヤーが短縮ライトを実行する場合、プロセス300はYESから下位レイヤー短縮および処理動作352に進み、要求データの一部が、例えば、宛先ファイルに書き込まれる。下位レイヤー(1つまたは複数)によるオフロード・ライト要求の処理に続いて、オフロード・ライト要求が、サーバーにおいて、下位レイヤーから受け取られる(354)。実施形態では、この応答は、トークンによって表されるデータが宛先ファイルに首尾良く書き込まれたか否か、そしてどの位のデータが書き込まれたかを示す。他の実施形態では、オフロード・ライト要求は、ファイル・ストレージ、またはオフロード・ライト要求を扱う役割を担う他のコンポーネントから直接クライアントに渡される。

40

【0061】

[0061] 図5に戻り、本開示の実施形態にしたがって、サーバーにおいてオフロード・

50

ライト応答が受信され(354)、クエリー356は、次に、データーの全てが宛先に書き込まれたか否か、またはライトが短縮されたか否か判定を行う。ライトが短縮された場合、要求データーの一部が宛先に書き込まれたことになり、プロセス300はYESに進み、短縮ライト応答358をクライアントに渡す。次いで、フロー300は316において終了する。クエリー356に戻り、要求データーの全てが宛先に書き込まれたとクエリー356において判定された場合、プロセス300はNOから動作360に進み、基礎ファイル記憶システムからの応答がクライアントに渡される。次いで、フロー300は316において終了する。

【0062】

[0062] このように、実施形態では、要求がファイル・ストレージのような下位レイヤーに渡される場合であっても、サーバー自体がライトを短縮することもできる。例えば、サーバーは、実施形態によれば、書き込み要求を処理する時間が所定の閾値を超える場合、ライトを短縮することができる。実施形態では、サーバーは、要求を下位レイヤーに渡す前にライトを短縮する。例えば、短縮は、オフロード・ライト要求をファイル・ストレージに送る前に行うことができる。他の実施形態では、サーバーは、下位レイヤー、例えば、ファイル・ストレージによる処理の後に、ライトを短縮する。先に論じたように、フロー300は、実施形態にしたがって実行することができる動作フローの一例に過ぎない。実施形態は、図3～図5に関して以上で行った具体的な説明に限定されるのではなく、追加の動作を含むこともできる。例えば、図示した動作ステップを他のステップに組み合わせること、および/または並び替えることもできる。更に、例えば、もっと少ないステップまたはもっと多いステップを使用することもできる。

【0063】

[0063] 図6に移ると、動作フロー400は、オフロード・ファイル動作を要求するステップを示す。実施形態では、フロー400は、クライアント202および204(図2)のようなクライアントにおけるリディレクターによって実行することができる。これらのリディレクターは、ファイル・システムにおけるファイルにアクセスするために、ファイル・サーバーと通信している。クライアントは、実施形態では、SMBプロトコルのバージョンまたはNFSの位置バージョンというような、ファイル・アクセス・プロトコルを使用して、ファイル・サーバーと通信する。

【0064】

[0064] フロー400は、動作402において開始し、ファイル・サーバーに接続する要求が送られる。動作402において送られる要求は、ファイル・サーバーを介してアクセス可能なファイル・システムに格納されているファイル情報にアクセスするために、ファイル・サーバーとのセッションを作るための要求である。この要求は、サーバー、例えば、サーバー206(図2)におけるファイル・サーバーに送ることができる。この要求は、SMBまたはNFSのバージョンのような、ファイル・アクセス・プロトコルにしたがってフォーマットされる。

【0065】

[0065] 動作402の後、フロー400は動作404に移り、セッションが作られたことを示す応答が受信される。実施形態の中には、動作402および404が、クライアントとサーバーとの間でセッションをネゴシエートするために交換される複数のメッセージの内一部であってもよい場合がある。メッセージの交換は、オフロード・ファイル動作にサービスするファイル・サーバーの能力を含む、能力の交換を含むことができる。

【0066】

[0066] 動作フローは動作404から動作406に移り、ファイルを開く要求が送られる。フロー400は動作406から動作408に移り、ファイルへのアクセスを付与する応答が受信される。動作408から、フローは動作410に移り、クライアントはオフロード・リード要求を送る。このオフロード・リード要求は、要求されたファイル・データーの一部を示す。また、このオフロード・リード要求は、本質的に、データーが、オフロード・リード要求に応答して送られるトークンによって表されることを要求する。動作4

1 2 において、トークンを含むオフロード・リード応答が受信される。

【 0 0 6 7 】

[0067] 実施形態では、動作 4 1 0 においてオフロード・リード要求を送ったクライアントは、リード要求においてそれが要求できるデーターについて、何らかの制限を受けることもある。例えば、実施形態では、クライアントが一部のデーターをローカルにキャッシュした場合、そのリード要求を送る前に、キャッシュされているあらゆる「ダーティ」データーをフラッシュする(flush)。オフロード・リードを送る前に、キャッシュされているダーティ・データーをサーバーにフラッシュし損ねると、オフロード・リードが、古い(stale)データーを表すトークンを供給するという結果になる可能性がある。実施形態の中には、クライアント自体がオフロード・リードを短縮できる場合もある。言い換えると、ファイル・データーの全範囲を要求せず、これによってダーティなキャッシュ・データーを除外することもできる。

10

【 0 0 6 8 】

[0068] 動作 4 1 2 に続いて、クライアントは動作 4 1 4 においてオフロード・ライト要求を送ることができる。フロー 4 0 0 は、動作 4 1 4 が動作 4 1 2 の直後にくるところを示すが、これは例示の目的のために過ぎないことは言うまでもない。他の実施形態では、フロー 4 0 0 を実行するクライアントが、データーを表すトークンを他の何らかの手段から受信する場合、動作 4 1 4 において送られたオフロード・ライト要求は、動作 4 1 0 において送られた要求のような、任意のオフロード・リード要求の前に実行することができる。

20

【 0 0 6 9 】

[0069] 実施形態では、動作 4 1 4 においてオフロード・ライト要求を送ったクライアントも、オフロード・ライト要求において送ることができるデーターに関して何らかの制限を受けることもある。オフロード・ライトが、トークン・データーを、ダーティ・データーをキャッシュしている宛先オフセットに書き込むことが許されると、キャッシュされているダーティ・データーがストレージに書き戻されるときに、このキャッシュされているダーティ・データーが、オフロード・ライトによって書き込まれたデーターを後に誤って上書きすることになる。オフロード・ライトを送るクライアントは、キャッシュされているダーティ・データーをクライアントが保持する位置のオフセットに対するライト要求はいずれも、送ることを回避すればよい。実施形態によれば、クライアントは、オフロード・ライト自体を短縮するのもよく、またはキャッシュされているダーティ・データーがオフロード・ライトの宛先のオフセットと重複するときは、キャッシュされているダーティ・データーを破棄するのもよく、またはクライアントがオフロード・ライトを実行しなくてもよい。

30

【 0 0 7 0 】

[0070] オフロード・ライト要求が動作 4 1 4 において送られた後、フロー 4 0 0 は動作 4 1 6 に移り、オフロード・ライト要求に対する応答が受信される。この応答は、オフロード・ライト要求において送られたトークンが首尾良く宛先ファイルに書き込まれたか否かを示すことができる。データーが部分的に書き込まれたかもしれない実施形態では、動作 4 1 6 において受信される応答は、首尾良く書き込まれたデーターの部分を注記し、データーの全てが宛先ファイルに書き込まれたのではないことを示す。

40

【 0 0 7 1 】

[0071] 破線で示す動作 4 1 8 は、実施形態によってはフロー 4 0 0 の実行中の任意の時点において実行される。図 6 に示す実施形態では、動作 4 1 8 は動作 4 1 6 の後に示されているが、実施形態は必ずしもこの順序に限定されとは限らない。フロー 4 0 0 を実行するクライアントが、トークンに関連する実際のデーターを要求する実施形態では、動作 4 1 8 に進む(sent)。即ち、このクライアントは、オフロード・リード要求を送ることによってまたは何らかの他の手段によってそれが受信したトークンを所持することができる。しかしながら、クライアントは、例えば、実際のデーターを要求しているアプリケーションにデーターを供給するために、トークンに関連する実際のデーターを必要とする

50

場合がある。これらの実施形態では、動作 4 1 8 は、トークンに関連するデータの一部を引き出す要求を送ることができる。クライアントは、実際のデータを供給することができるサーバーに、データを引き出す要求を送る。応答して、クライアントは実際のデータを受信し、このデータを、実際のデータを要求するアプリケーションに供給することができる。これは、実施形態において一部のクライアントによって実行することができる 1 つの追加動作に過ぎない。フロー 4 0 0 は 4 2 0 において終了する。

【 0 0 7 2 】

[0072] 先に注記したように、フロー 3 0 0 および 4 0 0 は、実施形態にしたがって実行することができる動作フローの一例に過ぎない。実施形態は、図 3 ~ 図 6 に関して先に行った具体的な説明には限定されず、追加の動作を含むこともできる。更に、図示した動作ステップを他のステップと組み合わせること、および / または並び替えることもできる。更に、例えば、もっと少ないステップまたはもっと多いステップを使用することもできる。

10

【 0 0 7 3 】

[0073] 図 7 は、一般的なコンピューター・システム 7 0 0 を示す。これは、本明細書において説明した実施形態を実現するために使用することができる。コンピューター・システム 7 0 0 は、計算環境の一例に過ぎず、コンピューターおよびネットワーク・アーキテクチャーの使用範囲や機能について何ら限定を示唆しようとするのでもない。また、コンピューター・システム 7 0 0 が、コンピューター・システム例 7 0 0 において図示されているコンポーネントのいずれの 1 つまたは組み合わせに関しても何らかの依存性や要件を有するように解釈してはならない。実施形態では、システム 7 0 0 は、図 1 に関して先に説明したクライアントおよび / またはサーバーとして使用することができる。

20

【 0 0 7 4 】

[0074] システム 7 0 0 は、その最も基本的な構成では、通例少なくとも 1 つの処理ユニット 7 0 2 とメモリー 7 0 4 とを含む。計算デバイスの正確な構成およびタイプに依存して、メモリー 7 0 4 は揮発性 (RAM のような)、不揮発性 (ROM、フラッシュ・メモリー等のような)、または何らかの組み合わせとすることができる。この最も基本的な構成は、図 7 において破線 7 0 6 によって示されている。システム・メモリー 7 0 4 は、ストレージ 7 0 8 のようなストレージを含むファイル記憶システムに格納することができるデータ 7 2 0 を表すトークン 7 2 3 のようなデータを格納する。

30

【 0 0 7 5 】

[0075] コンピューター読み取り可能媒体という用語は、本明細書において使用する場合、コンピューター記憶媒体を含むことができる。コンピューター記憶媒体は、揮発性および不揮発性、リムーバブルおよび非リムーバブル媒体を含むことができ、コンピューター読み取り可能命令、データ構造、プログラム・モジュール、または他のデータというような情報の格納のための任意の方法または技術で実現される。システム・メモリー 7 0 4、リムーバブル・ストレージ、および非リムーバブル・ストレージ 7 0 8 は全て、コンピューター記憶媒体の例である (即ち、メモリー・ストレージ)。コンピューター記憶媒体は、RAM、ROM、電子的消去可能なリード・オンリー・メモリー (EEPROM)、フラッシュ・メモリーまたは他のメモリー技術、CD-ROM、デジタル・バーサタイル・ディスク (DVD) または他の光ストレージ、磁気カセット、磁気テープ、磁気ディスク・ストレージまたは他の時期記憶デバイス、あるいは情報を格納するために使用することができる計算デバイス 7 0 0 によってアクセスすることができる任意の他の媒体を含むことができるが、これらに限定されるのではない。このようなコンピューター記憶媒体はいずれも、デバイス 7 0 0 の一部であってもよい。また、計算デバイス 7 0 0 は、キーボード、マウス、ペン、サウンド入力デバイス、タッチ入力デバイス等のような入力デバイス (1 つまたは複数) 7 1 4 も含むことができる。ディスプレイ、スピーカー、プリンター等のような出力デバイス (1 つまたは複数) 7 1 6 も含むことができる。以上のデバイスは例であり、他のものを使用してもよい。

40

【 0 0 7 6 】

50

【0076】 コンピューター読み取り可能媒体という用語は、本明細書において使用する場合、通信媒体も含むことができる。通信媒体は、コンピューター読み取り可能命令、データ構造、プログラム・モジュールまたは他のデータによって、搬送波のような変調データ信号または他の伝達メカニズム(transport mechanism)において具体化することができる。任意の情報配信媒体を含む。「変調データ信号」という用語は、当該信号において情報をエンコードするようなやり方で、1つ以上の特性が設定または変更されている信号を記述することができる。一例として、そして限定ではなく、通信媒体は、有線ネットワークまたは直接有線接続のような有線媒体と、音響、無線周波(RF)、赤外線、および他のワイヤレス媒体のようなワイヤレス媒体とを含むことができる。

【0077】

10

【0077】 本明細書にわたって、「一実施形態」または「実施形態」に言及したが、特定の説明された特徴、構造、または特性が少なくとも1つの実施形態に含まれることを意味する。つまり、このような句の使用は、単に1つよりも多い実施形態に言及することができる。更に、説明した特徴、構造、または特性は、1つ以上の実施形態では、任意の適したやり方で組み合わせることもできる。

【0078】

【0078】 しかしながら、関連技術における当業者であれば、実施形態は、前述の具体的な詳細の内1つ以上がなくても、あるいは他の方法、リソース、材料等を用いても実施できることを認めることができよう。その一方で、周知の構造、リソース、または動作を詳細に示さず説明しなかったのは、単に実施形態の形態を曖昧にするのを避けるためである。

20

【0079】

【0079】 以上、実施形態例および用途について例示および説明したが、実施形態は以上で説明した構成およびリソースそのものに限定されるのではないことは言うまでもない。当業者には明白な種々の変更、変化、および変形は、特許請求する実施形態の範囲から逸脱することなく、本明細書において開示した方法およびシステムの構成、動作、および詳細において行うことができる。

【図 1】

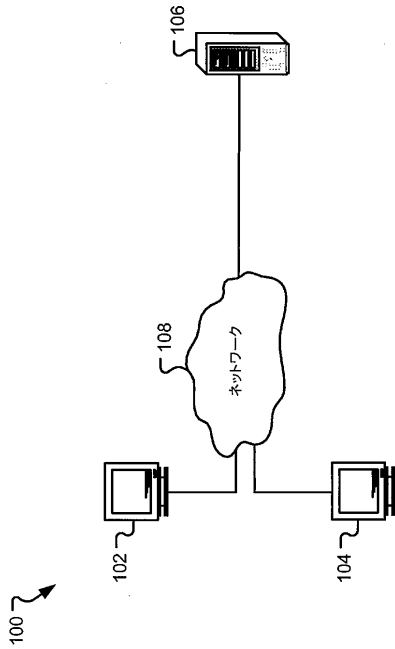


FIG. 1

【図 2】

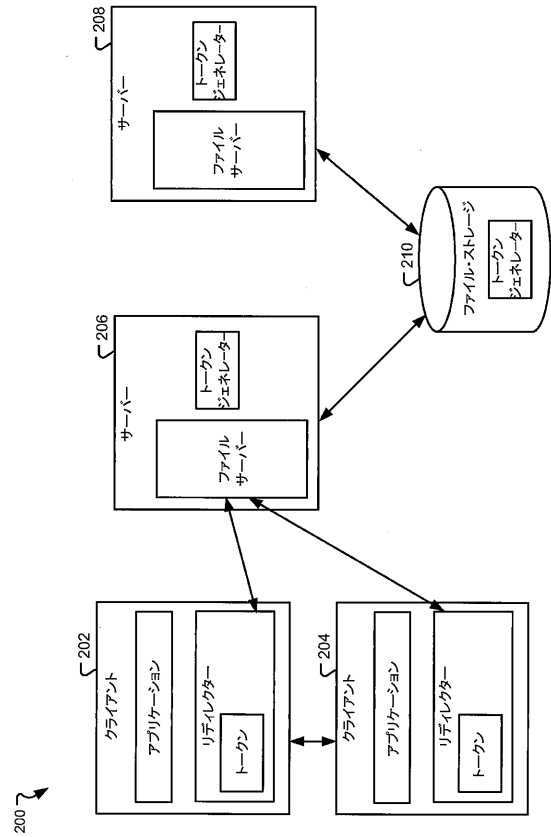


FIG. 2

【図 3】

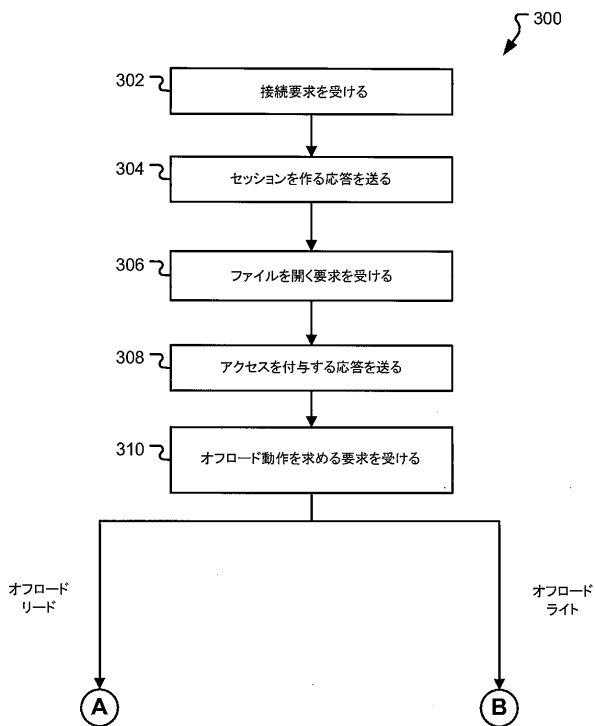


FIG. 3

【図 4】

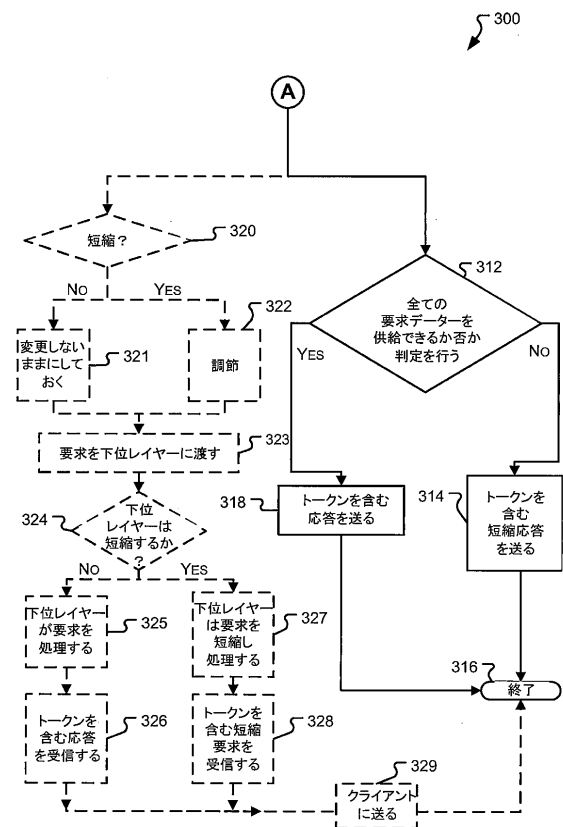


FIG. 4

【図 5】

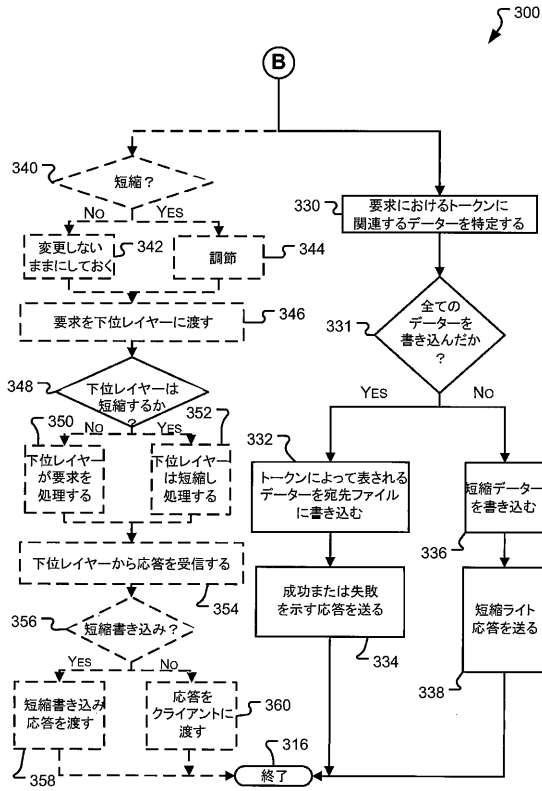


FIG. 5

【図 6】

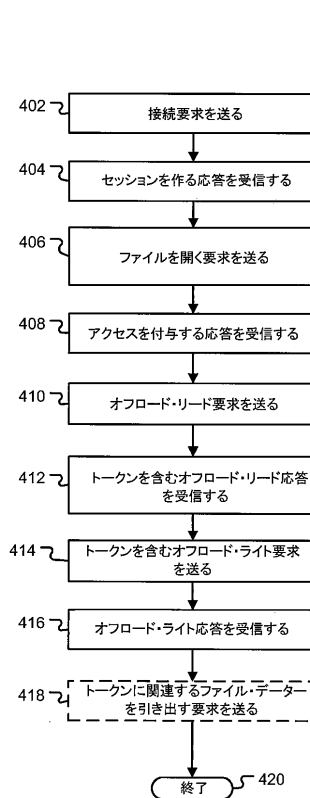


FIG. 6

【図 7】

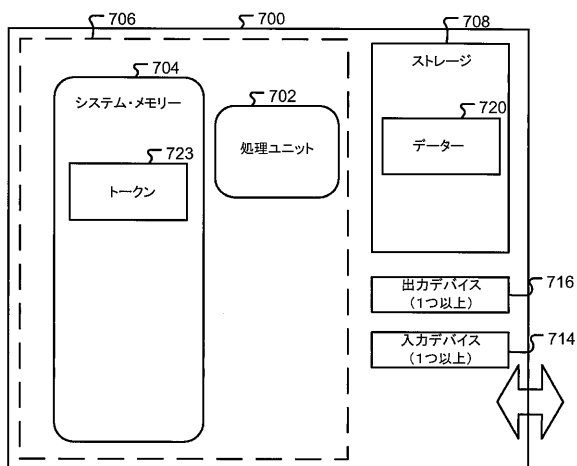




FIG. 7

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US2012/047261
A. CLASSIFICATION OF SUBJECT MATTER		
G06F 15/16(2006.01)i		
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 15/16; G06F 12/02; G06F 9/54; G06F 17/30		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) cKOMPASS(KIPO internal) & Keywords: token, file, offload		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2007-0055702 A1 (FRIDELLA, S. et al.) 08 March 2007 See abstract; and paragraphs [0007, 0044-0052].	1-10
A	US 2010-0115184 A1 (CHANG, CHING-WEN) 06 May 2010 See abstract; claims 1-19; and figures 1-3.	1-10
A	US 2004-0205202 A1 (NAKAMURA, T. et al.) 14 October 2004 See abstract; paragraphs [0032-0046]; and claims 1-16.	1-10
A	US 6275867 B1 (BENDERT; EDWARD JOSEPH et al.) 14 August 2001 See abstract; and claims 1-48.	1-10
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
<p>* Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"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</p> <p>"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</p> <p>"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</p> <p>"&" document member of the same patent family</p>		
Date of the actual completion of the international search 30 JANUARY 2013 (30.01.2013)		Date of mailing of the international search report 31 JANUARY 2013 (31.01.2013)
Name and mailing address of the ISA/KR  Korean Intellectual Property Office 189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea Facsimile No. 82-42-472-7140		Authorized officer LEE, Sang Hun Telephone No. 82-42-481-5914 

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2012/047261

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007-0055702 A1	08.03.2007	US 7617216 B2	10.11.2009
US 2010-0115184 A1	06.05.2010	TW 201019337 A	16.05.2010
		US 8250288 B2	21.08.2012
US 2004-0205202 A1	14.10.2004	JP 2004-272668 A	30.09.2004
		JP 4271967 B2	03.06.2009
		US 7613786 B2	03.11.2009
US 6275867 B1	14.08.2001	None	

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, 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, RS, 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, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA

(72)発明者 クリスチャンセン, ニール・アール

アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, 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, エルシーエイ - インターナショナル・パテンツ

(72)発明者 グリーン, ダスティン・エル

アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, 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, エルシーエイ - インターナショナル・パテンツ

(72)発明者 ピンカートン, ジェームズ・ティー

アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, 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, エルシーエイ - インターナショナル・パテンツ

(72)発明者 ナガル, ラジーヴ

アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, 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, エルシーエイ - インターナショナル・パテンツ

(72)発明者 マシュー, ブライアン・スティーヴン

アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, 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, エルシーエイ - インターナショナル・パテンツ

(72)発明者 アイタル, ジャイヴィル・ケイ

アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, 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, エルシーエイ - インターナショナル・パテンツ