(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6141189号
(P6141189)
(45)発行日　平成29年6月7日(2017.6.7)　(24)登録日　平成29年5月12日(2017.5.12)

(51)Int.Cl.　F I
G06F 13/00 (2006.01)　G06F 13/00 353C

請求項の数 10　(全 16 頁)

(21)出願番号	特願2013-543293 (P2013-543293)
(86)(22)出願日	平成23年12月6日 (2011.12.6)
(65)公表番号	特表2014-500559 (P2014-500559A)
(43)公表日	平成26年1月9日 (2014.1.9)
(86)国際出願番号	PCT/US2011/063618
(87)国際公開番号	WO2012/078693
(87)国際公開日	平成24年6月14日 (2012.6.14)
審査請求日	平成26年12月5日 (2014.12.5)
(31)優先権主張番号	12/964,749
(32)優先日	平成22年12月10日 (2010.12.10)
(33)優先権主張国	米国(US)

(73)特許権者　314015767
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
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
(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】ファイルシステムにおける透過的なフェイルオーバーの提供

(57)【特許請求の範囲】
【請求項１】
コネクションを再開することを促進するようファイルシステムの状態情報を捕捉するために、コンピューターによって実施される方法であって、
サーバー上に格納されたリモートリソースへのアクセス要求を第１のクライアントから受信するステップと、
前記アクセス要求に関連するクライアントセッションを特定する識別子を決定するステップと、
前記サーバーの外部にあり、かつ前記サーバーとは別個の再開サーバーによりアクセス可能な状態データストアにおいて、レジュームレコードを作成するステップであって、該レジュームレコードは、前記識別子によって検索可能であり、該レジュームレコードは、前記第１のクライアントによって要求された動作により作成される状態情報を、前記識別子に関連付けて格納する、ステップと、
前記第１のクライアントからファイル動作を受信するステップであって、該ファイル動作は、前記サーバーを通じてアクセス可能なファイルへのアクセスを要求する、ステップと、
前記状態データストアにおいて、レジューム状態情報を前記レジュームレコードに格納するステップであって、前記レジューム状態情報は、前記第１のクライアントが前記サーバーとのコネクションを失った場合に、前記受信されたファイル動作を再開するための情報を提供する、ステップと、

前記サーバーが利用不可能になる条件を検知し、前記再開サーバーに対して障害中の前記サーバーの代わりに作動するように通知するステップと、
前記第１のクライアントから、以前に接続したセッションに再度接続するための要求を受け取ると、該要求に関連するセッションのセッション識別子に基づいて、前記状態データストアから、格納された状態情報を取り出すステップであって、前記状態情報は、前記検知された条件により中断された、前記以前に接続したセッションのファイル動作を、前記再開サーバーが再開することを可能にする、ステップと、
前記レジュームレコードを前記再開サーバーにロードし、前記再開サーバーが前記第１のクライアントによって以前に要求されたファイル動作を継続することを可能にするステップと、
を含み、前記第１のクライアントによる前記再開サーバーへのコネクションの再開と競合して第２のクライアントが前記リモートリソースと干渉することを防ぐ所定のブラックアウト期間を、前記リモートリソースへのアクセスに対して施行する、
方法。
【請求項２】
前記アクセス要求は、コネクションに障害が生じた場合、複数の潜在的なコネクションにわたって前記クライアントセッションを特定するレジュームキーを含む、１又は複数のパラメータを含み、前記レジュームキーは前記決定された識別子の少なくとも一部である、請求項１に記載の方法。
【請求項３】
前記第１のクライアントが前記サーバーから切断されると、前記再開サーバーにおいて、該再開サーバーが、元のアクセス要求と相互に関連付けることができる新たなアクセス要求を受信するステップを更に含み、前記サーバーによって保持される状態情報を複数のクライアントコネクションの間で相互に関連付けることによって、コネクションの障害の後に、前記再開サーバーが前記第１のクライアントに応答することを助ける、請求項１に記載の方法。
【請求項４】
ネットワークファイルシステム（ＮＦＳ）サーバーが、前記第１のクライアントからレジュームキーを受信することなく前記識別子を自動的に決定するか、
前記識別子は、前記第１のクライアントが、切断されたセッションを再開するのを可能にする永続的なハンドルのために提供する、サーバーメッセージブロック（ＳＭＢ）レジュームキーである、請求項１に記載の方法。
【請求項５】
前記ファイル動作を受信するステップは、ファイルを開くこと、ファイルを閉じること、ファイルを読み取ること、ファイルを書き込むこと、ファイル上のリソースを得ること及びファイル上のロックを得ることから成るグループから選択される動作を実行するという要求を備える、請求項１に記載の方法。
【請求項６】
前記要求されたファイル動作を実行するステップは、前記サーバーによって格納された状態を修正し、前記格納されたレジューム状態情報を更新するステップは、前記修正された状態を捕捉し、
前記レジューム状態情報を更新するステップは、前記第１のクライアントが前記状態情報の少なくとも一部を再度確立する必要なく、別のサーバーが前記サーバーの状態を再度確立してクライアントの要求を前記サーバーの代わりに対処することを可能にするよう、前記レジュームレコード内の前記サーバーの状態のビューを最新の状態に保つステップを備える、請求項１に記載の方法。
【請求項７】
コネクションを再開することを促進するようファイルシステムの状態情報を捕捉するためのコンピューターシステムであって、
プロセッサと、該プロセッサによって実行されるソフトウェア命令を格納するように構

成されたメモリと
を備え、前記ソフトウェア命令は、
サーバーにおいて動作するように構成される状態収集コンポーネントであって、前記サーバーの外部にあり、かつ前記サーバーとは別個の再開サーバーによりアクセス可能な状態データストアにおいて、各ファイルハンドルについて状態レコードを作成し、前記ファイルハンドルを使用して第１のクライアントが動作を要求すると、状態情報を収集する、状態収集コンポーネントと、
前記第１のクライアントによって提供されるセッション識別子に関連して、収集された状態情報を格納する、状態格納コンポーネントと、
前記サーバーが障害を生じている場合、前記サーバーにより格納された状態情報を再作成するために前記再開サーバーが使用する、ファイルシステムの状態情報を持続的に格納する、前記状態データストアと、
前記サーバーが利用不可能になる条件を検知し、前記再開サーバーに対して障害中の前記サーバーの代わりに作動するように通知する、レジューム検知コンポーネントと、
前記第１のクライアントから、以前に接続したセッションに再度接続するための要求を受け取ると、該要求に関連するセッションのセッション識別子に基づいて、前記状態データストアから、格納された状態情報を取り出す、状態取り出しコンポーネントであって、前記状態情報は、前記検知された条件により中断された、前記以前に接続したセッションのファイル動作を、前記再開サーバーが再開することを可能にする、状態取り出しコンポーネントと、
１つ又は複数のリソースに対するアクセスに対して、前記第１のクライアントによる前記再開サーバーへのコネクションの再開と競合して第２のクライアントが前記リソースに干渉することを防ぐブラックアウト期間を施行する、ブラックアウト施行コンポーネントと、
前記再開サーバーが前記第１のクライアントによって以前に要求された動作を継続することを可能にするよう、前記取り出した状態情報を前記再開サーバーにロードする、状態リストアコンポーネントと
において具現化される、システム。
【請求項８】
前記状態収集コンポーネントは更に、前記第１のクライアントが前記サーバーに接続するときに、前記第１のクライアントからレジュームキーを受信し、収集された状態情報を前記状態データストア内の前記レジュームキーに関連付けるように構成される、請求項７に記載のシステム。
【請求項９】
前記状態データストアは、前記サーバーが動作を実行しているときに前記状態情報を受信し、障害が発生すると、状態を再開して、完了しなかったいずれかの動作の実行を継続するように、以前に受信した状態情報へのアクセスを前記再開サーバーに提供するように構成される、請求項７に記載のシステム。
【請求項１０】
コネクションを再開することを促進するようファイルシステム状態情報を捕捉するためのコンピュータプログラムであって、プロセッサによって実行されると、該プロセッサに、
サーバー上に格納されたリモートリソースへのアクセス要求を第１のクライアントから受信するステップと、
前記アクセス要求に関連するクライアントセッションを特定する識別子を決定するステップと、
前記サーバーの外部にあり、かつ前記サーバーとは別個の再開サーバーによりアクセス可能な状態データストアにおいて、レジュームレコードを作成するステップであって、該レジュームレコードは、前記識別子によって検索可能であり、該レジュームレコードは、前記第１のクライアントによって要求された動作により作成される状態情報を、前記識別

子に関連付けて格納する、ステップと、
前記第１のクライアントからファイル動作を受信するステップであって、該ファイル動作は、前記サーバーを通じてアクセス可能なファイルへのアクセスを要求する、ステップと、
前記状態データストアにおいて、レジューム状態情報を前記レジュームレコードに格納するステップであって、前記レジューム状態情報は、前記第１のクライアントが前記サーバーとのコネクションを失った場合に、前記受信されたファイル動作を再開するための情報を提供する、ステップと、
前記サーバーが利用不可能になる条件を検知し、前記再開サーバーに対して障害中の前記サーバーの代わりに作動するように通知するステップと、
前記第１のクライアントから、以前に接続したセッションに再度接続するための要求を受け取ると、該要求に関連するセッションのセッション識別子に基づいて、前記状態データストアから、格納された状態情報を取り出すステップであって、前記状態情報は、前記検知された条件により中断された、前記以前に接続したセッションのファイル動作を、前記再開サーバーが再開することを可能にする、ステップと、
前記レジュームレコードを前記再開サーバーにロードし、前記再開サーバーが前記第１のクライアントによって以前に要求されたファイル動作を継続することを可能にするステップと、
を含む方法を実行させ、前記第１のクライアントによる前記再開サーバーへのコネクションの再開と競合して第２のクライアントが前記リモートリソースと干渉することを防ぐ所定のブラックアウト期間を、前記リモートリソースへのアクセスに対して施行する、
コンピュータプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明はファイルシステムにおける透過的なフェイルオーバーの提供に関する。
【背景技術】
【０００２】
ネットワーク上の二つのコンピューターの間でファイル、プリンタ、およびその他リソースを共有するために様々な技術が存在する。例えば、リソース共有のための、二つのアプリケーションレイヤーネットワークプロトコルは、ＳＭＢ(Server Message Block)およびＮＦＳ(Network File System)である。ＳＭＢはMICROSOFT（登録商標） WINDOWS（登録商標）およびその他オペレーティングシステムによって使用され、二つのコンピューターまたはその他リソースが通信し、リソースへのアクセスを要求し、意図されたリソースのアクセスを指定し（例えば、読み取り、書き込み等）リソースをロックする、等々を可能とする。MICROSOFT（登録商標） WINDOWS（登録商標） VistaはＳＭＢ２．０を導入し、ＳＭＢ１．０のコマンドセットを簡素化しおよび多くのその他改良点を加えた。MICROSOFT（登録商標） WINDOWS（登録商標） ７およびサーバー２００８ Ｒ２はＳＭＢ２．１を導入し、オポチュニスティックロック(oplocks)およびその他改良点を加えた。
【０００３】
大多数のリモートのリソース共有のためのプロトコルは、コネクションとセッションとの一対一の関係であると考えられている。セッションは、リソースにアクセスするための任意の単一の要求の存続期間を表し、およびコネクションが終了するまでの上記リソースへの連続したアクセスを表す。セッションはまた、特定のセキュリティプリンシパルと関連付けられおよび当該セッションの間認証された動作を決定するセキュリティ証明を有効化することができる。コネクションは、ＴＣＰ(Transmission Control Protocol)、ＵＤＰ(User Datagram Protocol)、またはＳＭＢおよびＮＦＳのような上位プロトコルが、コマンドを実行するために通信することができるように、コネクションのその他タイプを含むことができる。ＳＭＢあるいはＮＦＳセッションは、典型的に要求ソースと要求ターゲットとの間のＴＣＰまたはＵＤＰコネクションを開き、一つまたは複数のＳＭＢあるいは

ＮＦＳコマンドを送信してターゲットリソースへアクセスし、次いで当該セッションを閉じることを含む。時々、セッションの間にコネクションが失われ（例として、ネットワーク障害によって）、当該コネクションの間に確立したクライアントとサーバーの何れの状態を壊す。クライアントとサーバーとのコネクションを再度確立するために、典型的には最初にコネクションを確立するために使用したステップの全てを再び繰り返さなければならない。

【０００４】

ＳＭＢ２プロトコルは、クライアントがサーバーから切断された場合、クライアントがサーバーに対するファイルハンドルをすばやく再度確立することができるようにするレジュームキー(resume key)を提供し、クライアントがサーバーに対するネットワークのラウンドトリップを減らし、およびクライアントが再度接続する時にサーバー上の負荷を減らすことができるようにする。しかしながら、今日レジュームキーは、ＳＭＢ２サーバーが、サーバーのリブートまたはクラスタのフェイルオーバーの間、急変する状態を見失うようなサーバーフェイルオーバーが発生した場合に状態のリストアを提供しない。既存の導通と関連付けられた状態情報は失われそして再度確立されなければならない。加えて、レジュームキーはアプリケーションの境界内でのみ作成されおよび使用されることができるが、共有されることができないアプリケーションレベルの概念である。

【発明の概要】

【０００５】

本明細書で説明されるコネクション状態システムは、レジュームキーに関連したクライアントの状態情報をリモートに格納することによって、クライアントがサーバーまたは異なる代替サーバーとのコネクションを再開することを可能とする。当該システムは、サーバーで動作し変わりやすいサーバー状態情報の格納を促進するレジュームキーフィルタを提供する。当該状態情報はオポチュニスティックロック(oplocks)、クライアントに対して保証されたリース、およびファイルハンドル上のインフライト動作等を含むことができる。レジュームキーフィルタドライバはファイルシステム上に存在するが、レジュームキーフィルタドライバによって複数ファイルアクセスプロトコルが当該フィルタを使用することを可能し、並びに当該フィルタがこの機能性を複数ファイルシステムにわたって提供することを可能とする。システムは、実際のプロトコルとは別個にプロトコルに対する状態情報を提供する。サーバーがダウンするかまたはクライアントとの接続を失う等のフェイルオーバーイベントが発生すると、システムは別のサーバーまたは同じサーバーを割り当てることができ、そして様々なクライアントによって保持されるファイルハンドルのためにレジュームキーフィルタを使用して状態を再度確立する。当該フィルタは、動作中のファイル状態が一貫してリストアされることができ、およびその他のクライアントがその間にファイルにアクセスするために介在しないことを保証するフェイルオーバーの後で、動作中のファイル上にブラックアウトウインドウを強制的に出力する。再開フェーズにおいては、レジュームキーは、既存のフェイルオーバー以前のファイルハンドルを、レジュームキーフィルタによって格納されたフェイルオーバー後に保存されたファイル状態へマッピングするために使用される。このように、コネクション状態システムによって、同じまたは別のサーバーがフェイルオーバーイベント後クライアントに対して可能な限り中断することなく、クライアントとの以前のセッション状態を再開することができる。

【０００６】

本発明の概要は、後述の発明の詳細な説明においてより詳細に説明される技術思想を、簡潔な態様での抜粋を紹介するために示された。本発明の概要は、特許請求の範囲に期された発明の技術的範囲の中核となる特徴や本質的な特徴を特定することを意図したものでなく、発明の概要は特許請求の範囲に記載された発明の技術的範囲を制限するために使用されるものでもない。

【図面の簡単な説明】

【０００７】

【図１】本発明の一実施形態にかかる、コネクション状態システムのコンポーネントを図

示するブロック構成図である。
【図２】本発明の一実施形態にかかる、ファイルシステム状態情報を捕捉するためのコネクション状態システムの処理を図示するフローチャートである。
【図３】本発明の一実施形態にかかる、フェイルオーバー後にコネクションを再開するためのコネクション状態システムの処理を図示するフローチャートである。
【図４】本発明の一実施形態にかかる、コネクション状態システムの動作環境を図示するブロック構成図である。
【発明を実施するための形態】
【０００８】
本明細書で説明されるコネクション状態システムは、レジュームキーに関連したクライアントの状態情報をリモートに格納することによって、クライアントがサーバーまたは異なる代替サーバーとのコネクションを再開することを可能とする。当該システムは、サーバーで動作し変わりやすいサーバー状態情報の格納を促進するレジュームキーフィルタを提供する。当該状態情報はオポチュニスティックロック(oplocks)、クライアントに対して保証されたリース、およびファイルハンドル上のインフライト動作等を含むことができる。レジュームキーフィルタドライバはファイルシステム上に存在するが、レジュームキーフィルタドライバによって複数ファイルアクセスプロトコルが当該フィルタを使用することを可能し、並びに当該フィルタがこの機能性を複数ファイルシステムにわたって提供することを可能とする。システムは、実際のプロトコルとは別個にプロトコルに対する状態情報を提供する。サーバーがダウンするかまたはクライアントとの接続を失う等のフェイルオーバーイベントが発生すると、システムは別のサーバーまたは同じサーバーを割り当てることができ（例として、冗長化されたイーサネット（登録商標）コネクション等の異なるコネクションを介して）、そして様々なクライアントによって保持されるファイルハンドルのためにレジュームキーフィルタを使用して状態を再度確立する。当該フィルタは、動作中のファイル状態が一貫してリストアされることができ、およびその他のクライアントがその間にファイルにアクセスするために介在しないことを保証するフェイルオーバーの後で、動作中のファイル上にブラックアウトウインドウを強制的に出力する。再開フェーズにおいては、レジュームキーは、既存のフェイルオーバー以前のファイルハンドルを、レジュームキーフィルタによって格納されたフェイルオーバー後に保存されたファイル状態へマッピングするために使用される。このように、コネクション状態システムによって、同じまたは別のサーバーがフェイルオーバーイベント後クライアントに対して可能な限り中断することなく、クライアントとの以前のセッション状態を再開することができる。
【０００９】
システムは、サーバーがクライアントに対する自身のコネクションを失った後で、透過的なフェイルオーバーのために使用されることができるレジュームキーフィルタを提供する。レジュームキーフィルタはファイルシステムの最上位に存在し、および従ってファイルシステムに対してアクセスするために使用されるプロトコルとは独立している。レジュームキーフィルタは動作中のファイル状態を記録し、そして次いでフェイルオーバー後に動作中のファイル状態をリストアする。レジュームキーフィルタは様々な状態情報を捕捉することができる。例えば、当該フィルタはオープンハンドル（レジュームキーによって静的に参照された）、確定していないファイル状態（クローズ時の削除、保留中の削除およびロック状態等）、ならびにある特定のインフライト／中断されたファイル動作を備える動作中のファイルシステム状態を記録する。当該フィルタは、フェイルオーバー後に動作中のファイルシステム状態をリストアして、オープンハンドルが再開され、フェイルオーバー以前のオープンハンドルに整合させ、そしてインフライト動作は一貫して再び行われることができる。フェイルオーバー以前のオープンハンドルに整合させるために、オープンハンドルが再開されるようにフェイルオーバー後に動作中のファイルシステム状態をリストアし、そしてインフライト動作は一貫して再び行われることができる。当該フィルタは複数のリモートファイルシステム（ＲＦＳ）のための手段を提供し、レジュームキー

を通じて参照されるオープンファイルハンドルに関連付けられたプライベートな不透明データを格納しおよび取り出す。当該フィルタは、動作中のファイル状態が一貫してリストアされることができ、およびその他のクライアントがその間にファイルにアクセスするために介在しないことを保証する、フェイルオーバーの後で、動作中のファイルの上にブラックアウトウインドウを強制的に出力する。また、当該フィルタによって、現在動作中のファイルが「一時停止」され、そして次いでノードがフェイルオーバーするクラスタシナリオにおいてＳＭＢをサポートするためにフェイルオーバーなしで再開される。

【００１０】

リモートファイルシステム（ＲＦＳ）は、作成中の剰余のパラメータとして全てのファイル作成動作についてレジュームキーを提供する。当該キーはＲＦＳに対して固有のものである。レジュームキーフィルタは、レジュームキーおよびＲＦＳ識別キーをファイルハンドルのためにＧＵＩＤ(Globally Unique IDentifier)として一緒に使用する。再開フェーズにおいては、レジュームキーは、既存のフェイルオーバー以前のファイルハンドルを、レジュームキーフィルタによって格納されたフェイルオーバー後に保存されたファイル状態へマッピングするために使用される。このように、コネクション状態システムによって、同じまたは別のサーバーがフェイルオーバーイベント後クライアントに対して可能な限り中断することなく、クライアントとの以前のセッション状態を再開することができる。

【００１１】

図１は本発明の一実施形態にかかる、コネクション状態システムのコンポーネントを図示するブロック構成図である。システム１００は状態収集コンポーネント１１０、状態格納コンポーネント１２０、状態データストア１３０、レジューム検知コンポーネント１４０、状態取り出しコンポーネント１５０、状態リストアコンポーネント１６０、ブラックアウト施行コンポーネント１７０、およびリソースサスペンションコンポーネント１８０を含む。これらのコンポーネント各々は、本明細書においてさらに詳細に説明される。

【００１２】

状態収集コンポーネント１１０は、各々のファイルハンドルのための状態記録を作成しおよびファイルハンドルを使用するクライアント要求動作としての状態情報を収集する。コンポーネント１１０は、サーバーで動作しおよびサーバーの外部に状態情報を格納することができ、故にサーバーが利用不可能な場合でも状態情報はアクセスされることができる。例えば、コンポーネント１１０は、本明細書においてさらに説明される状態データストア１３０に状態情報を格納することができる。状態収集コンポーネント１１０は、クライアントがサーバーに対して接続する時に、クライアントからレジュームキーを受信することができ、そして当該コンポーネント１１０は収集された状態情報と状態データストア１３０におけるレジュームキーとを関連付ける。クライアントがフェイルオーバーイベント後に再度接続をしている場合、当該クライアントは最初のコネクションを開くために使用されたものと同じレジュームキーを提供し、ならびに現在のサーバーは以前のサーバーによって格納された状態情報を見つけおよび状態情報からサーバー状態を再度作成することができる。

【００１３】

状態格納コンポーネント１２０は、収集される状態情報をクライアントによって提供されるレジュームキーに関連して格納する。コンポーネント１２０は、状態情報を状態データストア１３０に格納し、およびフェイルオーバーイベントの場合にはリストアされるであろうレジュームキーに関連する動作の記録を保管する。状態情報は、オープンファイルハンドル、オポチュニスティックロック(oplocks)保証、リースおよびリース情報、進行中のファイル動作、バイト範囲のロック、ならびにクライアントが以前の状態の全てを再度確立することなく、別のサーバーがクライアントの要求を実行するために使用するであろう、その他何れの情報を含むことができる。

【００１４】

状態データストア１３０は、再開するサーバーが、障害中のサーバーによって格納され

た状態情報を再度作成するために使用する、ファイルシステム状態情報を持続的に格納する。ある場合において、再開するサーバーおよび障害中のサーバーは、クライアントに対して異なるコネクションを使用しまたは短時間の停止の後に復帰する同じサーバーであることができる。その他の場合において、再開するサーバーおよび障害中のサーバーは、異なるサーバーであり、および状態データストア１３０は状態情報を共有するために双方のサーバーに対してアクセス可能な位置に提供される。状態データストア１３０は、一つまたは複数のファイル、ファイルシステム、ハードドライブ、データベース、ストレージエリアネットワーク（ＳＡＮ）、クラウドベースのストレージサービス、またはデータを持続的に格納しならびに情報を交換するために、障害中および再開するサーバー双方に対してアクセス可能とするその他ストレージ機能を含むことができる。障害中のサーバーが動作を実行していると、動作の進行についての状態情報を状態データストア１３０に格納する。障害が発生すると、障害中のサーバーは中断され、そして再開するサーバーは、当該状態情報にアクセスして、状態を再開しおよび完了しなかった何れの動作を実行することを継続する。

【００１５】

レジューム検知コンポーネント１４０は、障害中のサーバーが利用不可能となる条件を検知し、および再開するサーバーに対して障害中のサーバーの代わりに作動するように通知する。当該検知をクライアント駆動型とすることができ、クライアントがシステムに再度接続しおよび以前に使用されたレジュームキーを提供するまでは、システムは何れの再開するステップを実行しないようにする。システムはキーおよび当該キーに関連して格納された何れの状態情報を特定し、ならびにコネクションの設定の一部としてその状態情報をリストアする。再開するサーバーは、同じまたは障害中のサーバーとは異なるサーバーであることができ、およびレジューム検知コンポーネント１４０は、クライアントの要求をハンドルするために再開するサーバーが動作中となることを確実に行う。その他の実施形態において、当該検知は、サーバー駆動型であることができ、および障害中のサーバーがダウンしたことを検知するとシステムは率先して再開するサーバーを起動することができる。システムはまた、クライアントがサーバーに対してコネクションを要求する前であっても、格納された状態情報に再開するサーバーを予め指定することができる。

【００１６】

状態取り出しコンポーネント１５０は、再開するサーバーに対してアクセス可能な位置から格納された状態情報を取り出すが、当該状態情報によって、再開するサーバーは、検知された障害条件によって中断された、以前に要求された何れのファイルシステム動作を再開することができる。状態取り出しコンポーネント１５０は、状態情報を状態データストア１３０から取り出し、および状態リストアコンポーネント１６０を呼び出し、再開するサーバーに情報をロードすることにより、再開するサーバーがクライアントによって要求された動作を継続することができる。

【００１７】

状態リストアコンポーネント１６０は、取り出された状態情報を再開するサーバーにロードし、再開するサーバーがクライアントによって以前に要求された動作を継続することができる。当該リストアはまた、何れのオプロック(oplocks)および／またはクライアントによって保持されたリースを最新の情報に更新することを含むことができ、その他のクライアントが以前に要求されたアクセスレベルおよび／またはクライアントに対して認められた排他性を遵守することを確実に行うことができる。状態リストアコンポーネント１６０によって、新たなサーバーまたはノードは、クライアントへの重い負荷をかけることなく、障害中のサーバーまたはノードに代わって、過去の動作を繰り返すことによって状態情報をリストアすることができる。ＳＭＢ２．０のようなプロトコルを使用しているクライアントは、同じサーバーに対してコネクションをリストアするためのレジュームキーの使用方法を既に備えており、およびコネクション状態システムによって代替サーバーはクライアントに対して透過的に障害中のサーバーに代わることができる。レジュームキーはまた、ＮＦＳと共に使用されることができる。ＮＦＳの場合において、レジュームキー

の概念は、クライアントに対して完全に不透明である。クライアントは、レジュームキー生成、管理、および関連付けに明示的に参照または参加をしない。むしろ、レジュームキーはサーバー側の概念である。

【００１８】

ブラックアウト施行コンポーネント１７０は、一つまたは複数のファイルあるいはその他のリソースに対するアクセス時に、二番目のクライアントが、再開するサーバーに対してコネクションを再開する最初のクライアントと競合するであろう方法でリソースへ干渉することを防止するブラックアウト期間を施行する。コンポーネント１７０は、大多数の競合動作を避けるのに充分な長さ（例として、１５または３０秒）ではあるが、最初のクライアントがコネクションを再開しない場合はその他のクライアントがリソースへアクセスをすることを防止する程の長さではないと認められる期間を、自動的に選択することができる。最初のクライアントが選ぶ場合、当該期間によって、最初のクライアントがコネクションを再開するための時間を持つことができる。ある実施形態において、システムによって、管理者またはユーザがブラックアウト期間の存続期間を構成し、アプリケーション特有の目的のためにシステムを調整することができる。また、システムによって、個々のクライアントが、作成／オープン要求に対するパラメータまたはその他ＡＰＩ(Application Programming Interface)としてのブラックアウト期間を要求することができる。ブラックアウトされたリソースへのアクセス試行に応答して、コンポーネント１７０は特定の期間の後に再試行する指示を提供することができ、または単純に要求を失敗とすることができる。ブラックアウト期間の後に、コネクションを再開したクライアントが無い場合、ブラックアウトが終了し、そしてリソースへのアクセス要求は通常通りに成功する。

【００１９】

リソースサスペンションコンポーネント１８０により、現在動作中のリソースがフェイルオーバーイベントなしで一時停止および再開され、クラスタが計画された態様において別のノードへフェイルオーバーすることができる。一例は負荷分散処理である。一時停止することにより、新しいノードに移行されつつある状態のサブセットのシナリオを可能とする。例えば、クラスタにおける一つのノードがオーバーロードの場合、管理者は当該ノードの半分のクライアントを、新しいノードに移行したいかもしれない。一時停止することにより、移行されつつあるオープン状態を捕捉することを可能とし、および同じオープンの継続として、クライアントが新しいノードに接続することを可能とする（例として、サーバー状態を再度確立することなく）。別の例として、ＳＭＢは、一般的なノードがクラスタ化され、および交互にクライアント要求へのサービスを行うために使用されることができる、クラスタリングシナリオをサポートする。時々、メンテナンス目的のように、特定のノードをダウンさせる理由があるが、および現在のノードが正常に一時停止し、新しいノードが作動し、古いノードを作動停止し、そして次いで作動停止されたノード上で、何れのメンテナンス動作を実行するのが望ましい。これはクライアントに望ましくない影響を有する可能性があるが、本明細書において説明される技術を使用して、システム１００は組織化された態様においてノードを一時停止することができ、およびクライアントが新しいノードとの動作を効率的に再開することができる。

【００２０】

コネクション状態システムが実装されるコンピューティング装置は、ＣＰＵ(Central Processing Unit)、メモリ、入力装置（例として、キーボードおよびポインティング装置）、出力装置（例として、ディスプレイ装置）、およびストレージ装置（例、ディスクドライブまたはその他不揮発性ストレージ媒体）を含むことができる。メモリおよびストレージ装置は、システムを実装または作動させるコンピューターが実行可能な命令（例として、ソフトウェア)としてエンコード可能な、コンピューターが読み取り可能なストレージ媒体である。加えて、データ構造およびメッセージ構造は、通信リンクの信号のようにデータ送信媒体を経由して格納されまたは送信され得る。通信リンクは、インターネット、ローカルエリアネットワーク、ワイドエリアネットワーク、ポイントトゥーポイントダイアルアップコネクション、携帯電話ネットワーク等のように様々な通信リンクを使用す

ることができる。

【００２１】

システムの実施形態は、様々な動作環境において実装することができる。当該動作環境は、パーソナルコンピューター、サーバーコンピューター、ハンドヘルドまたはラップトップ装置、マルチプロセッサシステム、マイクロプロセッサベースのシステム、プログラム可能な家電、デジタルカメラ、ネットワークＰＣ、マイクロコンピューター、メインフレームコンピューター、上記システムあるいは装置、セットトップボックス、ＳＯＣ(System On Chips)等々の、何れを含む分散コンピューティング環境を含む。コンピューターシステムは携帯電話、ＰＤＡ(Personal Digital Assistants)、スマートフォン、パーソナルコンピューター、プログラム可能な家電製品、デジタルカメラ、等々であることができる。

【００２２】

システムは、一つまたは複数のコンピューターあるいはその他の装置によって実行されるプログラムモジュール等の、コンピューターが実行可能な命令の一般的なコンテキストで説明されることができる。一般的に、プログラムモジュールは、特定のタスクを実行または特定の抽象的なデータタイプを実装する、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造等々を含む。典型的に、プログラムモジュールの機能は、様々な実施形態において所望の態様で集約されまたは分散されることができる。

【００２３】

図２は本発明の一実施形態にかかる、ファイルシステム状態情報を捕捉するためのコネクション状態システムの処理を図示するフローチャートである。まずブロック２１０において、システムはサーバー上に格納されたリモートリソースに対するアクセス要求を受信する。アクセス要求は一つまたは複数のパラメータを含むことができるが、当該パラメータはコネクションが失敗した場合に、複数の潜在的なコネクションの中からセッションを特定するために使用されるレジュームキーを含んでいる。リソースアクセス要求は、クライアントから送信された一連のアクセス要求における最初となることができ、およびクライアントがサーバーからこれまでに切断された場合、クライアントは、続いて発生するオープン要求において、同じまたは新しいサーバーに対して同じレジュームキーを提供して、コネクションを再開することができる。レジュームキーは、サーバーによって（またはサーバー間にわたって）、管理される状態情報を、独立したクライアントコネクションとして見える別なものと、相互に関連させることにより、サーバーがクライアントに対してより速く応答することに役立つ。

【００２４】

続いてブロック２２０において、システムは要求に関連したクライアントセッションを特定する識別子を決定する。ある場合における識別子は、クライアントが様々な理由により切断されたセッションを再開することを可能とする、永続的なハンドルのために提供するレジュームキーである。アクセス要求は、プロトコルにおけるはっきりと定義された位置で一つまたは複数のパラメータを含むことができ、故にシステムは要求における適切な位置を読み取ることにより当該キーを引き出すことができる。あるいはまたもしくはそのうえ、サーバーは、クライアントによって明白に提供された情報を含まない、識別子を決定するための自動化された処理を含むことができる。例えば、サーバーは、ＩＰ(Internet Protocol)アドレス、またはクライアントコネクションが以前のセッションと相互に関連することを、サーバーに対して示すその他推測されるデータによって、クライアントを特定することができる。

【００２５】

続いてブロック２３０において、システムは、クライアントによって要求された動作によって作成された状態情報と、引き出された識別子とを関連付ける、引き出された識別子によって検索可能なレジュームレコードを作成する。レジュームレコードは、現在のアクセスする要求をハンドリングしているサーバーの外部の位置で格納されることができ、サーバーが障害を起こした場合、別のサーバーが動作を再開するための記録を読み取りおよ

び元のサーバーの代わりに作動することができる。レジュームレコードはファイル、データベースレコード、またはその他ストレージの形式を含むことができる。当該レコードは、オープンファイルハンドルのリスト、クライアントによって得られるオプロック(oplocks)、リース(leases)、またはその他ファイルシステム状態情報を包含することができる。

【００２６】

続いてブロック２４０において、システムは、サーバーを通じてアクセス可能なファイルに対するアクセスを要求する、クライアントからファイル操作を受信する。当該ファイル操作は、ファイルを開き、ファイルを閉じ、ファイルを読み取り、ファイルに書き込み、共有プリンタへの印刷すること、またはその他ファイルシステム操作の要求であることができる。受信された操作は、サーバー上で作成されているある特定の量の状態情報を含むことができる。例えば、クライアントがファイルに対するハンドルを開いた場合、サーバーはファイルに関連したその他クライアントの要求を管理するため、ならびに存続期間および／または当該ハンドルのためのクリーンアッププロセスを管理するためのハンドルを追跡する。

【００２７】

続いてブロック２５０において、システムは、クライアントがサーバーとのそのコネクションを失った場合、受信されるファイル動作を再開するための情報を提供するレジューム状態情報を、作成されたレジュームレコードに格納する。クライアントコネクションが障害を起こした場合、再びリモートリソースを開くことによって、および同じレジュームキーまたはその他セッション識別子を指定することによって、クライアントはコネクションを再開することを試行する。上記により、サーバーまたは別のサーバーが、格納されたレジュームレコードにアクセスし、および以前の状態情報を再度確立することを可能とする。

【００２８】

続いてブロック２６０において、システムは要求されたファイル動作を実行する。当該動作は、ファイルを開き、ファイルのコンテンツを読み取り、ファイルに対してデータを書き込み、ファイルに対するアクセス権限を変更することができ、またはその他何れのファイルシステム動作であることができる。当該動作の結果は、サーバーによって格納される状態を変更することができる。例えば、クライアントがハンドルを閉じることを試行し、およびサーバーがうまくハンドルを閉じる場合、サーバー状態を更新して、サーバーによって追跡されるハンドルのリストからハンドルを削除する。

【００２９】

続いてブロック２７０において、システムは、作成されたレジュームレコードにおける格納されたレジューム状態情報を、実行されたファイル動作の結果に基づいて更新する。システムは、いつフェイルオーバーを引き起こす障害が発生するか事前に知り得ず、そのため、システムはサーバーができるだけ以前のサーバーの状態に忠実に状態を再度確立することを可能とする、レジュームレコードにおけるサーバー状態の最新のビュー(view)を保つ。完了しなかった動作は、当該動作を完了するために再び行われることができ、一方で、完了した動作は繰り返される必要がない（しかし、サーバーは結果をクライアントに対して再度送信するかもしれない）。このように、システムは、サーバー状態情報を変更する様々なファイルシステム動作の間におよび動作の後に必要とされるように、状態を更新する。

【００３０】

続いてブロック２８０において、システムは、要求されるファイル動作の結果を示す応答を、クライアントに対して送信する。クライアントとサーバーが依然として接続されている場合、クライアントによって要求されるように動作が継続し、およびサーバーは、更新された状態情報を追跡することを継続する。コネクションが失われた場合はいつでも、別のサーバーが割り当てられまたは既存のサーバーが修理されることができ、そして状態情報は状態ストアからロードされ、以前のサーバー状態を再度確立することができる。ク

ライアントからセッションを再開するための新しい要求を受信すると、クライアントは、フェイルオーバーが発生したことおよびクライアントが潜在的に元のものとは異なるサーバーと対話していることを、クライアントは認識する必要がない。ブロック２８０の後に、これらステップは完結する。

【００３１】

図３は一実施形態にかかる、フェイルオーバー後にコネクションを再開するためのコネクション状態システムの処理を図示するフローチャートである。まずブロック３１０において、システムはクライアントからサーバー上に格納されたリモートリソースを開くための要求を受信する。アクセス要求は、コネクションが障害を起こした場合に複数の潜在的なコネクションにわたるセッションを特定するために使用されるレジュームキーを含む、一つまたは複数のパラメータを含むことができる。図２を参照して検討されたリソースアクセス要求とは異なり、当該要求は以前に接続されたセッションに対して再度接続するための要求である。クライアントは、もともと提供されたものと同じレジュームキーを提供し、故にサーバーは現在のセッション要求と以前のセッションとを相互に関連させることができる。

【００３２】

続いてブロック３２０において、システムは要求に関連したクライアントのセッションを特定するセッション識別子を決定する。ある場合における識別子は、クライアントが様々な理由のために切断されたセッションを再開することを可能とする、永続的なハンドルのために提供するＳＭＢ２レジュームキーである。アクセス要求は、プロトコルにおけるはっきりと定義された位置で一つまたは複数のパラメータを含むことができ、故にシステムは要求における適切な位置を読み取ることにより当該キーを引き出すことができる。その他の場合において、サーバーはクライアントについての情報に基づいて自動的に識別子を決定することができる。

【００３３】

続いてブロック３３０において、システムは状態ストアにおいて受信されたセッション識別子を検索し、セッション識別子に関連付けられたレジュームレコードを特定する。再開可能なセッションを使用し、クライアントとの対話をする何れの以前のサーバーは、クライアントとの対話を通じて状態情報を継続的に格納する。クライアントがコネクションを再度確立することを試行するとき、状態情報は元のサーバーの代わりとなるフェイルオーバーサーバーに対して利用可能である。状態情報は、元のサーバーの外部に格納され、当該情報が元のサーバーの障害後にアクセス可能になる。

【００３４】

続いてブロック３４０において、システムは状態ストアからレジュームレコードに関連付けられた以前の状態情報を受信する。状態情報は、オープンファイルハンドル、得られたリース、得られたオプロック(oplocks)、等々のような、静的状態、並びに完了していないかもしれないインフライト動作のような動的状態を特定する。格納された状態情報により、フェイルオーバーサーバーが、クライアントよって特定の処理をすることなく元のサーバーの代わりとなることができる。クライアントは、再開可能なハンドルおよびコネクションを再開可能にするステップの実行を理解するが、どのサーバーが何れの特定の時間でコネクションハンドリングを終了することを認識することができない。クライアントは、フェイルオーバーサーバーを含む幾つかのサーバーのうち、何れかひとつのアドレスを解決することができる、ドメインネームまたはネットワークファイル共有を介してサーバーにアクセスすることができる。

【００３５】

続いてブロック３５０において、システムは、ファイルシステム状態を追跡するファイルシステムコンポーネント内に情報をロードすることによって、受信された以前の状態情報をリストアする。当該状態をロードした後に、フェイルーバサーバーのローカル状態は、もし以前の動作の全てがフェイルオーバーサーバー上で発生していた場合に、当該状態が見えていたであろう態様と同じである。このように、フェイルオーバーサーバーは、元

のサーバーが障害を起こしていないコネクションを有していたであろうと同じように、クライアントが一連の動作を継続するために有用である。

【００３６】

続いてブロック３６０において、システムは、サーバーがレジュームレコードを見つけたことを示すクライアントアクセス要求、および以前のセッションに関連するクライアント動作を受信する準備ができているという応答を行う。サーバーの応答に基づいて、セッションは再開されたか、またはクライアントは以前の動作を繰り返すステップを取る必要があるかを決定することができる。セッションがうまく再開された場合、クライアントは以前の動作が完了したことを継続して知ることができ、またはサーバーが再開された後に再び行われ、完了する。ある場合において、システムは、クライアントにフェイルオーバー以前のファイルハンドルと同じ状態を有する新しいファイルハンドルを渡すことができる。ブロック３６０の後に、これらステップは完結する。

【００３７】

図４は一実施形態にかかる、コネクション状態システムの動作環境を図示するブロック構成図である。当該環境は、ファイルシステムと対話をする一つまたは複数のオペレーティングシステムサービスまたはアプリケーション含む。例えば、MICROSOFT（登録商標）WINDOWS（登録商標）は、ＳＲＶとして知られるサーバーサービス４２０、およびＮＦＳとして知られるネットワークファイルシステムサービス４１０を含む。ネットワークファイルシステムサービス４１０およびサーバーサービス４２０は、コンピューターシステム間の、ファイルおよびプリンタ等の共有されるリソースへのアクセスを提供する。サーバーサービス４２０は、WINDOWS（登録商標）ネットワークに対して一般的なＳＭＢプロトコルを使用し、一方でネットワークファイルシステムサービス４１０は、より一般的にＮＦＳを使用するＵＮＩＸ（登録商標）ベースのシステムに対してアクセスを提供する。プロトコルにかかわらず、レジュームキーフィルタ４３０は、リモートデータストアにおける動作を再び行うための、ファイル動作および状態情報の格納を捕捉する。動作は、ファイルシステムレベル４４０（例として、ＮＴＦＳまたはその他ファイルシステム）を通過して、および一つまたは複数のユーザデータファイル４５０に作用する。この間、レジュームキーフィルタ４３０は、状態情報をログファイル４６０またはその他データストアに対して書き込み、これにより別のサーバーが状態情報を取り出しおよびクライアントに対するコネクションを再開することが出来る。システムは、特定のプロトコルまたは関係するファイルシステムと独立して動作することができ、および様々なコンポーネントは更新され、それらの独自の特定の状態情報を状態データストアにおいて保存することができる。

【００３８】

ある実施形態において、コネクション状態システムは、ファイルシステムコンポーネントの代わりにデータの不透明ブロブ(blobs)を格納し、システムが、コンポーネントが特定するナレッジなしにコネクションを再開することを可能とする。例えば、本明細書において説明されるレジュームキーフィルタは、サーバーサービスがその現在の状態を再度作成するために必要とするであろう何れのデータのためのサービスを、サーバーに問い合わせることができる。フィルタは、次いで何れの受信されたデータを不透明ブロブ（すなわち、フィルタはブロブ内に何があるかまたはそのセマンティックな意味を知る必要はない）として状態ストアに格納することができる。フェイルオーバー条件が発生すると、新しいサーバー上で動作しているレジュームキーフィルタは、格納された状態情報にアクセスし、格納されたブロブを取り出し、そしてサーバーサービスに対して当該ブロブを提供することができ、故にサーバーサービスはその独自の状態をリストアすることができる。このように、当該システムは、サーバーの各々のプロトコルを実装するコンポーネントの内部動作に関する特定のナレッジなしに、多くのタイプのプロトコルで動作することができる。

【００３９】

ある実施形態において、コネクション状態システムは、その他のクライアントが一定の

時間の間に（すなわち、ブラックアウト期間）再開可能なハンドルに関連したファイルまたはその他のリソースにアクセスすることをブロックする。ブラックアウト期間中に元のクライアントが再度接続した場合、元のクライアントは以前の状態のすべてがある状態でそのコネクションを取り戻し、そして動作を再開することができる。別のクライアントが接続を試行した場合、サーバーは、一定時間待つことおよびリトライをすることを指示するメッセージを提供することができる。レジュームを認識したクライアントは、この情報を使用して、ブラックアウト期間の後までリトライを送らせることができる一方で、古いクライアントは単にコネクションに失敗しおよびユーザの要求で手作業によるリトライをするかもしれない。元のクライアントがブラックアウト期間内に応答しない場合、サーバーはレジューム状態情報をクリーンアップし、および新しいクライアントが通常通りにリソースにアクセスすることを可能にする。

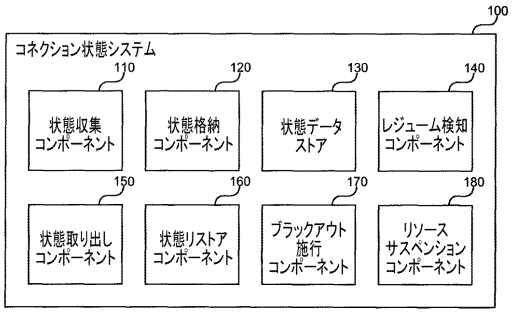
【００４０】

ある実施形態において、コネクション状態システムは様々なストレージ装置またはレジュームの速度を向上させるための方法を使用することができる。例えば、システムはレジューム状態情報を格納するために、高速の、不揮発性のストレージ装置（例として、ＳＳＤ(Solid State Disk)）を使用することができ、これによりレジュームがデータに対してより高速にアクセスし、何れの更なる障害によって既に中断された動作を遅らせることを避ける。別な例として、システムは各々のサーバーによって作成される全ての変化を、サーバーのグループに対してブロードキャストすることができ、これにより各々のサーバーは状態情報の独自のコピーを管理することができ、および元のサーバーの障害が発生した場合にはフェイルオーバーサーバーに選択されることができる。

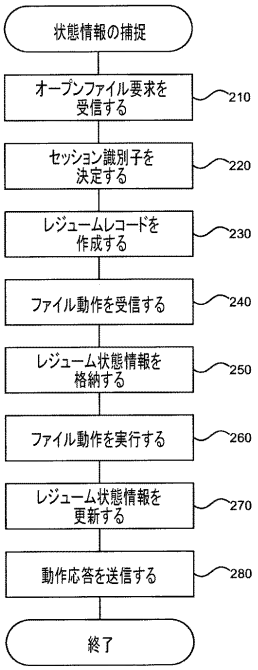
【００４１】

前述より、コネクション状態システムの特定の実施形態は、図示の目的のために本明細書に記載されたものであるが、本発明の技術思想および技術範囲から逸脱することなく様々な修正がなされることは理解されるであろう。従って、本発明は添付された特許請求の範囲によって以外は限定されない。

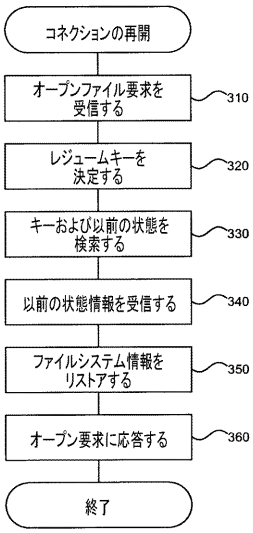
【図１】


100
コネクション状態システム
110
状態収集コンポーネント
120
状態格納コンポーネント
130
状態データストア
140
レジューム検知コンポーネント
150
状態取り出しコンポーネント
160
状態リストアコンポーネント
170
ブラックアウト施行コンポーネント
180
リソースサスペンションコンポーネント

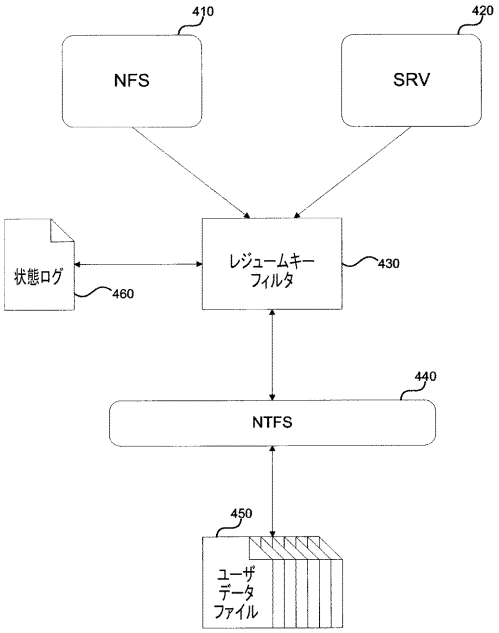
【図２】


状態情報の捕捉
オープンファイル要求を受信する
210
セッション識別子を決定する
220
レジュームレコードを作成する
230
ファイル動作を受信する
240
レジューム状態情報を格納する
250
ファイル動作を実行する
260
レジューム状態情報を更新する
270
動作応答を送信する
280
終了

【図３】


コネクションの再開
オープンファイル要求を受信する
310
レジュームキーを決定する
320
キーおよび以前の状態を検索する
330
以前の状態情報を受信する
340
ファイルシステム情報をリストアする
350
オープン要求に応答する
360
終了

【図４】


410
NFS
420
SRV
状態ログ
460
レジュームキーフィルタ
430
440
NTFS
450
ユーザデータファイル

フロントページの続き

(72)発明者 ポール アール. スワン
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 パテンツ内
(72)発明者 マシュー ジョージ
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 パテンツ内
(72)発明者 デービッド エム. クルーズ
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 パテンツ内
(72)発明者 ルーペシュ シー. バッテパティ
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 パテンツ内
(72)発明者 マイケル シー. ジョンソン
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 パテンツ内

審査官 坂東 博司

(56)参考文献 特開平１０－１３３９７１（ＪＰ，Ａ）
米国特許出願公開第２００５／００９１２１２（ＵＳ，Ａ１）
特開平０７－０３６７６０（ＪＰ，Ａ）
特開２００３－１９６１７８（ＪＰ，Ａ）
特開２０００－０６６９２２（ＪＰ，Ａ）
特開平１１－２６５３６１（ＪＰ，Ａ）
特開平０６－３４２３８２（ＪＰ，Ａ）
米国特許出願公開第２００５／０２２３０１４（ＵＳ，Ａ１）
米国特許出願公開第２００９／０３１９６６１（ＵＳ，Ａ１）
米国特許第０７６６４９９１（ＵＳ，Ｂ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １３／００