

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4510028号
(P4510028)

(45) 発行日 平成22年7月21日 (2010. 7. 21)

(24) 登録日 平成22年5月14日 (2010. 5. 14)

(51) Int. Cl.

F I

G O 6 F 12/00 (2006. 01)

G O 6 F 3/06 (2006. 01)

G O 6 F 13/10 (2006. 01)

G O 6 F 12/00 5 1 4 E

G O 6 F 3/06 3 0 2 A

G O 6 F 12/00 5 4 5 A

G O 6 F 13/10 3 4 0 A

G O 6 F 12/00 5 1 4 M

請求項の数 17 (全 34 頁)

(21) 出願番号 特願2006-541719 (P2006-541719)
 (86) (22) 出願日 平成16年11月24日 (2004. 11. 24)
 (65) 公表番号 特表2007-519088 (P2007-519088A)
 (43) 公表日 平成19年7月12日 (2007. 7. 12)
 (86) 国際出願番号 PCT/US2004/039591
 (87) 国際公開番号 W02005/052800
 (87) 国際公開日 平成17年6月9日 (2005. 6. 9)
 審査請求日 平成18年9月1日 (2006. 9. 1)
 (31) 優先権主張番号 10/721, 596
 (32) 優先日 平成15年11月25日 (2003. 11. 25)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 303039534
 ネットアップ、インコーポレイテッド
 アメリカ合衆国 カリフォルニア 940
 89, サニーヴェール, イースト ジ
 ャバ ドライブ 495
 (74) 代理人 100087642
 弁理士 古谷 聡
 (74) 代理人 100076680
 弁理士 溝部 孝彦
 (74) 代理人 100121061
 弁理士 西山 清春
 (72) 発明者 フェアー, ロバート, エル
 アメリカ合衆国 ノースカロライナ州 275
 13, ケアリー, スレジャー・コート・1
 01

最終頁に続く

(54) 【発明の名称】 複数読み出しストリームのための適応先読み技術

(57) 【特許請求の範囲】

【請求項 1】

ストレージシステムに記憶された 1 以上のファイル、ディレクトリ、v d i s k、又は、L U N において確立された複数の異なる読み出しストリームに対して先読み処理を同時に実施するように構成された、ストレージシステムにおいて実施されるストレージオペレーティングシステムのための方法であって、

ストレージシステムにおいてクライアント読み出し要求を受信するステップであって、該クライアント読み出し要求が、ストレージシステムに記憶されたファイル、ディレクトリ、v d i s k、又は、L U N からのクライアント要求データの読み出しをストレージオペレーティングシステムに知らせるものである、クライアント読み出し要求を受信するステップと、

受信した前記クライアント読み出し要求が、前記クライアント要求データを有する前記ファイル、ディレクトリ、v d i s k、又は、L U N に対する複数の読み出しストリームに対応する複数のリードセットデータ構造（「リードセット」）のうちのいずれかに一致するか否かを判定するステップと、

受信した前記クライアント読み出し要求に一致するものと判定されたリードセットに記憶された先読みメタデータのセットに従って、先読み処理を実施するステップとを含み、

前記クライアント読み出し要求は、開始データブロック (704) を含み、

前記複数のリードセットデータ構造はそれぞれ、最終読み出しオフセット (608)、及び

10

20

最終読み出しサイズ(610)を記憶し、

前記判定するステップは、

前記開始データブロック(704)が、前記複数のリードセットデータ構造のいずれかに記憶された前記最終読み出しオフセット(608)と前記最終読み出しサイズ(610)の合計に等しいとき、当該リードセットデータ構造を前記クライアント読み出し要求に一致するものとして判定するステップと、

前記開始データブロック(704)が、前記複数の読み出しストリームのいずれかにおいて読み出された最後のデータブロックから前後に延びる所定の範囲内にあるとき、当該リードセットデータ構造を前記クライアント読み出し要求に一致するものとして判定するステップと

を含み、

前記複数のリードセットデータ構造はそれぞれ、ネクスト先読み値(612)、及び先読みサイズ(614)を記憶し、

前記先読み処理を実施するステップは、前記ネクスト先読み値(612)が示すデータブロックの次のデータブロックから開始して、前記先読みサイズ(614)により指定される数のデータブロックに対して先読み処理を実施することを含む、方法。

【請求項 2】

前記受信するステップの前に、

複数の異なる読み出しストリームが内部に確立された 1 以上のファイル、ディレクトリ、v d i s k、又は、L U N のそれぞれに対し、少なくとも 1 つのリードセットを割り当てるステップと、

前記複数の異なる読み出しストリームのそれぞれについて個別の先読みメタデータのセットを生成するステップと、

生成された前記先読みメタデータのセットのそれぞれを、その先読みメタデータのセットに関連する読み出しストリームが内部に確立された前記ファイル、ディレクトリ、v d i s k、又は、L U N に割り当てられた異なるリードセットに記憶するステップとを更に含む、請求項 1 に記載の方法。

【請求項 3】

前記割り当てるステップと前記生成するステップの前に、割り当てられたリードセットをそれぞれ、所定の値のセットを記憶するように初期化するステップを更に含む、請求項 2 に記載の方法。

【請求項 4】

ファイル、ディレクトリ、v d i s k、又は、L U N に割り当てられる前記リードセットの数は、そのファイル、ディレクトリ、又は、L U N のサイズによって決まる、請求項 2 に記載の方法。

【請求項 5】

ファイル、ディレクトリ、v d i s k、又は、L U N に割り当てられる前記リードセットの数は、そのファイル、ディレクトリ、v d i s k、又は、L U N のサイズが増大するのに応じて動的に増加される、請求項 4 に記載の方法。

【請求項 6】

前記所定の範囲は、受信したクライアント読み出し要求の中に指定されたクライアントが要求するデータブロックの数の倍数に基づいて導出される、請求項 1 に記載の方法。

【請求項 7】

第 3 のリードセットが未使用であるものと判定された場合、該第 3 のリードセットは、受信したクライアント要求に一致するものと判定される、請求項 1 に記載の方法。

【請求項 8】

前記第 3 のリードセットに記憶されたレベル値が特殊な指示値に等しい場合、前記第 3 のリードセットは未使用であるものと判定される、請求項 7 に記載の方法。

【請求項 9】

前記クライアント要求データを有するファイル、ディレクトリ、又は、L U N が、ラン

10

20

30

40

50

ダムアクセススタイルを使用してアクセスされることを、前記ストレージオペレーティングシステムが判定した場合、先読み処理は実施されない、請求項 1 に記載の方法。

【請求項 10】

受信した前記クライアント読み出し要求に含まれる D A F S キャッシュヒントが、前記クライアント要求データを有するファイル、ディレクトリ、又は、L U N が、ランダムアクセススタイルを使用してアクセスされることを示す、請求項 9 に記載の方法。

【請求項 11】

(i) リードセットが、受信したクライアント読み出し要求に一致するものと判定され、且つ

(ii) その一致するリードセットが、前記クライアント要求データによって所定のデータブロック又はメモリアドレスを越えて延長されるリードストリームに関連する先読みメタデータのセットを記憶している場合を除き、

先読み処理は実施されない、請求項 1 に記載の方法。

【請求項 12】

前記先読み処理を実施するステップの前に、

受信した前記クライアント要求が、前記クライアント要求データを有するファイル、ディレクトリ、v d i s k、又は、L U N に割り当てられたリードセットのうちのいずれにも一致しない場合、

受信したクライアント読み出し要求を新たな読み出しストリームにおける前記第 1 の読み出し要求として識別するステップと、

前記新たな読み出しストリームに関連する先読みメタデータのセットを生成するステップと、

前記クライアント要求データを有するファイル、ディレクトリ、v d i s k、又は、L U N に割り当てられたリードセットのうちの 1 つを再使用のために選択するステップと、

生成された前記新たな読み出しストリームに関連する先読みメタデータのセットを、前記再使用のために選択されたリードセットに記憶するステップと

を実施することを更に含む、請求項 1 に記載の方法。

【請求項 13】

前記再使用のために選択されたリードセットは、前記クライアント要求データを有するファイル、ディレクトリ、v d i s k、又は、L U N に関連する残りのリードセットのそれぞれに記憶されたレベル値以下のレベル値を記憶する、請求項 12 に記載の方法。

【請求項 14】

前記ストレージシステムにおいて受信される前記クライアント読み出し要求は、ファイルベースのクライアント読み出し要求である、請求項 1 に記載の方法。

【請求項 15】

前記ストレージシステムにおいて受信される前記クライアント読み出し要求は、ブロックベースのクライアント読み出し要求である、請求項 1 に記載の方法。

【請求項 16】

ストレージオペレーティングシステムを使用して、ストレージシステムに記憶された 1 以上のファイル、ディレクトリ、v d i s k、又は、L U N において確立された複数の異なる読み出しストリームに対し、先読み処理を同時に実施するストレージシステムであって、該方法が、

ストレージシステムにおいてクライアント読み出し要求を受信する手段であって、該クライアント読み出し要求が、ストレージシステムに記憶されたファイル、ディレクトリ、v d i s k、又は、L U N からのクライアント要求データの読み出しをストレージオペレーティングシステムに知らせるものである、クライアント読み出し要求を受信する手段と、

受信した前記クライアント読み出し要求が、前記クライアント要求データを有する前記ファイル、ディレクトリ、v d i s k、又は、L U N に対する複数の読み出しストリー

10

20

30

40

50

ムに対応する複数のリードセットデータ構造（「リードセット」）のうちのいずれかに一致するか否かを判定する手段と、

受信した前記クライアント読み出し要求に一致するものと判定されたリードセットに記憶された先読みメタデータのセットに従って、先読み処理を実施する手段と

を含み、

前記クライアント読み出し要求は、開始データブロック(704)を含み、

前記複数のリードセットデータ構造はそれぞれ、最終読み出しオフセット(608)、及び最終読み出しサイズ(610)を記憶し、

前記判定する手段は、

前記開始データブロック(704)が、前記複数のリードセットデータ構造のいずれかに記憶された前記最終読み出しオフセット(608)と前記最終読み出しサイズ(610)の合計に等しいとき、当該リードセットデータ構造を前記クライアント読み出し要求に一致するものとして判定し、

10

前記開始データブロック(704)が、前記複数の読み出しストリームのいずれかにおいて読み出された最後のデータブロックから前後に延びる所定の範囲内にあるとき、当該リードセットデータ構造を前記クライアント読み出し要求に一致するものとして判定するように構成され、

前記複数のリードセットデータ構造はそれぞれ、ネクスト先読み値(612)、及び先読みサイズ(614)を記憶し、

前記先読み処理を実施する手段は、前記先読み値(612)が示すデータブロックの次のデータブロックから開始して、前記先読みサイズ(614)により指定される数のデータブロックに対して先読み処理を実施するようにさらに構成される、ストレージシステム。

20

【請求項 17】

ストレージシステムに記憶された1以上のファイル、ディレクトリ、v d i s k、又は、L U Nにおいて確立された複数の異なる読み出しストリームに対し、先読み処理を同時に実施するストレージシステムにおいて実施される、ストレージオペレーティングシステムのための方法を実施するための、プロセッサで実施するための命令を含むコンピュータ読取可能媒体であって、該方法が、

ストレージシステムにおいてクライアント読み出し要求を受信するステップであって、該クライアント読み出し要求が、ストレージシステムに記憶されたファイル、ディレクトリ、v d i s k、又は、L U Nからのクライアント要求データの読み出しをストレージオペレーティングシステムに知らせるものである、クライアント読み出し要求を受信するステップと

30

受信した前記クライアント読み出し要求が、前記クライアント要求データを有する前記ファイル、ディレクトリ、v d i s k、又は、L U Nに対する複数の読み出しストリームに対応する複数のリードセットデータ構造（「リードセット」）のうちのいずれかに一致するか否かを判定するステップと、

受信した前記クライアント読み出し要求に一致するものと判定されたリードセットに記憶された先読みメタデータのセットに従って、先読み処理を実施するステップとを含み、

40

前記クライアント読み出し要求は、開始データブロック(704)を含み、

前記複数のリードセットデータ構造はそれぞれ、最終読み出しオフセット(608)、及び最終読み出しサイズ(610)を記憶し、

前記判定するステップは、

前記開始データブロック(704)が、前記複数のリードセットデータ構造のいずれかに記憶された前記最終読み出しオフセット(608)と前記最終読み出しサイズ(610)の合計に等しいとき、当該リードセットデータ構造を前記クライアント読み出し要求に一致するものとして判定するステップと、

前記開始データブロック(704)が、前記複数の読み出しストリームのいずれかにおいて読み出された最後のデータブロックから前後に延びる所定の範囲内にあるとき、当該リ

50

ードセットデータ構造を前記クライアント読み出し要求に一致するものとして判定するステップと

を含み、

前記複数のリードセットデータ構造はそれぞれ、ネクスト先読み値(612)、及び先読みサイズ(614)を記憶し、

前記先読み処理を実施するステップは、前記先読み値(612)が示すデータブロックの次のデータブロックから開始して、前記先読みサイズ(614)により指定される数のデータブロックに対して先読み処理を実施することを含む、コンピュータ読取可能媒体。

【発明の詳細な説明】

【技術分野】

10

【0001】

発明の分野

本発明はストレージシステムに関し、特に、複数の読み出しストリームの先読み処理を同時に実施するストレージシステムのための技術に関する。

【背景技術】

【0002】

発明の背景

ストレージシステムは、ディスクのような記憶装置における情報の編成に関するストレージサービスを提供するコンピュータである。ストレージシステムは、情報をディスクに記憶される一連のデータブロックとして論理編成するためのストレージオペレーティングシステムを有する。従来のストレージ・エリア・ネットワーク(SAN)のようなブロックベースのデプロイメントでは、データブロックは、ストレージシステムにおいて直接アドレス指定される場合がある。しかしながら、ネットワーク・アタッチド・ストレージ(NAS)のようなファイルベースのデプロイメントでは、オペレーティングシステムは、ディスク上でアドレス指定可能なファイルやディレクトリの階層構造としてデータブロックを論理編成するためのファイルシステムを実施する。この場合、ディレクトリは、他のファイルやディレクトリに関する情報を記憶する特殊形式のファイルとして実施される場合がある。

20

【0003】

ストレージシステムはクライアント/サーバモデルの情報配送に従って動作するように構成される場合があり、それによって多数のクライアントシステム(クライアント)が、そのストレージシステムに記憶されたファイルのような共有資源にアクセスすることができる。ストレージシステムは通常、地理的に分散されたイーサネット(R)リンクのような相互接続通信リンクの集まりからなるコンピュータネットワーク上に配備され、それによってクライアントはストレージシステム上の共有情報(例えば、ファイル)に遠くからアクセスすることができる。クライアントは通常、トランスミッション・コントロール・プロトコル/インターネット・プロトコル(TCP/IP)のような所定のネットワーク通信プロトコルに従ってフォーマットされた個々のデータフレーム又はデータパケットをやりとりすることにより、ストレージシステムと通信する。この場合、プロトコルは、相互接続されたコンピュータシステムが互いに通信する方法を規定するルールの集まりから構成される。

30

40

【0004】

ファイルベースのデプロイメントでは、クライアントは、セマンティックレベルのアクセスを使用して、ストレージシステムに記憶されたファイルやファイルシステムにアクセスする。例えば、クライアントは、ストレージシステム上の特定のファイルに記憶された情報の取り出し(「読み出し」)や、そのファイルへの情報の記憶(「書き込み」)を要求する場合がある。クライアントは通常、コモン・インターネット・ファイル・システム(CIFS)プロトコル、ネットワーク・ファイル・システム(NFS)プロトコル、及び、ダイレクト・アクセス・ファイル・システム(DAFS)プロトコルといった従来のファイルベースのアクセス・プロトコルに従ってフォーマットされたファイルシステム・

50

プロトコル・メッセージ（パケットの形をしている）を発行することにより、そうしたサービスをファイルシステムベースのストレージシステムに要求する。クライアントは、要求するデータがディスク上に記憶されている例えばデータブロックのような特定の位置を考慮することなく、アクセスしようとする1以上のファイルの識別を要求する。クライアント要求が「読み出し」要求である場合、クライアントが要求したデータが格納されているデータブロックが取り出され、要求されたデータがクライアントに返される。

【0005】

ブロックベースのデプロイメントでは、クライアント要求は、ストレージシステムにおける特定の具体的なデータブロックを直接アドレス指定することができる。ブロックベースのストレージシステムの中には、データブロックをデータベースの形に編成するものもある。データがファイルとして編成される場合、クライアントが要求する情報は、その独自のマッピングを有し、ファイルセマンティックを管理する一方、ストレージシステムへのその要求（及び、応答）は、要求された情報をディスク上のブロックアドレスとしてアドレス指定する。このように、ブロックベースのストレージシステムにおけるストレージ・バスは、リモート・クライアント・システムにまで拡張されているように見える場合がある。この「拡張バス」は通常、SCSI over FC (FCP) プロトコルや、SCSI over TCP/IP / イーサネット(R) (iSCSI) プロトコルのようなブロックベースのアクセスプロトコルを使用するように構成されたファイバ・チャネル (FC) やイーサネット(R) メディアとして実施される。

【0006】

ブロックベースのシステムにおける各記憶装置には通常、一意の論理ユニット番号 (LUN) が割り振られ、例えばリモートクライアントは、その論理ユニット番号を使用して記憶装置を指定することができる。従って、「イニシエータ」クライアントシステムは、「ターゲット」のLUNに記憶された特定範囲のデータブロックのデータ転送を要求する場合がある。例えば、クライアントは、目標記憶装置における開始データブロックや、クライアント要求に従って記憶又は取り出しすべき連続したブロックの数を指定することができる。例えば、クライアント要求が「読み出し」要求である場合、要求された範囲のデータブロックが取り出され、要求元のクライアントへ返される。

【0007】

一般に、ファイルシステムは、「ディスク上」のデータブロック、例えば、ディスクブロック番号アドレス空間において割り当てられた個々のディスク番号 (d b n) にはアクセスしない。そうではなく、例えば d b n アドレス空間においてディスク上に記憶される幾つかのデータブロックと、例えばボリュームブロック番号 (v b n) 空間においてファイルシステムによって編成される同じ幾つかのデータブロックとの間には、一対一のマッピングが存在する。例えば、ファイルシステムにおいて、ディスク上のN個のデータブロックは、各データブロックにゼロからN - 1までの一意のv b nを割り当てることによって管理される場合がある。また、ファイルシステムは、幾つかのデータブロック（すなわち、v b n）のセットをファイルシステムによって管理されるファイルやディレクトリに関連付ける場合がある。その場合、ファイルシステムは、ファイル又はディレクトリ内の各データブロックに対し、対応する「ファイルオフセット」又はファイルブロック番号 (f b n) を与える場合がある。例えば、ファイル又はディレクトリにおけるファイルオフセットは、固定サイズの幾つかのデータブロックを単位として、例えば、4キロバイト (k B) を単位として測定される場合がある。従って、ファイルオフセットはそのファイル又はディレクトリにおけるf b n番号に一対一にマッピングすることができる。従って、各ファイル又はディレクトリは、ファイルシステムにおいて連続した番号の幾つかのf b nが割り当てられた一連のデータブロックとして定義される。例えば、各ファイル又はディレクトリにおける最初のデータブロックには、ゼロのような所定の開始f b n番号が割り当てられる。ただし、ファイルシステムは、一連のf b n番号をファイルごとに割り当てるのに対し、ファイルシステムは、さらに広いボリュームアドレス空間全体にわたって

v b n 番号を割り当てる。

【 0 0 0 8 】

読み出しストリームは、要求ファイル内における幾つかのファイルオフセットからなる論理的に連続した範囲からデータを取り出すことをストレージシステムに対して命じる、1以上のクライアント要求のセットとして定義される。換言すれば、読み出しストリームの最初の要求が受信された後、その読み出しストリームにおけるその後の要求はすべて、そのストリームの前回の要求によりアクセスされるファイルにおけるファイルオフセットの連続的シーケンスを論理的に「延長」する。従って、読み出しストリームは、連続した番号の幾つかの f b n が割り当てられた一連のデータブロックの読み出しをストレージシステムに対して命じる一連のクライアント要求として、ファイルシステムによって構成される場合がある。例えば、読み出しストリームにおける第1の要求は、f b n 1 0 ~ 1 9 が割り当てられたデータブロックの第1の集合を読み出し、読み出しストリームの第2の要求は、f b n 2 0 ~ 2 5 を有するデータブロックを読み出し、第3の要求は、f b n 2 6 ~ 4 2 が割り当てられたデータブロックを読み出す、等々である。なお、読み出しストリーム中のクライアント要求は、その要求が読み出しストリームの論理的に連続した範囲のファイルオフセットからのデータの読み出しをストレージシステムに対して命じるものであれば、ファイルベースのセマンティックを使用するものであっても、ブロックベースのセマンティックを使用するものであってもよい。

10

【 0 0 0 9 】

動作上、ストレージオペレーティングシステムは通常、同じファイルに対する幾つかのクライアントアクセスからなる順序付きシーケンスに基づいて、読み出しストリームを識別する。以下で使用されるように、ファイルとは、ゼロ以上の読み出しストリームを確立することが可能な任意のデータセットとして解釈される。従ってファイルは、ファイルベースのストレージシステム上に記憶された従来のファイルやディレクトリである場合もある。そのためストレージシステムは、クライアントの現在の要求ファイルデータを得るために、ストレージシステムが、ファイル内に既に確立された読み出しストリームを論理的に延長するデータブロックのセットを読み出さなければならないか否かを判定する。そうであれば、そのクライアント要求は読み出しストリームに関連付けられ、読み出しストリームは、読み出したデータブロックの数だけ延長される場合がある。

20

【 0 0 1 0 】

ストレージシステムは、読み出しストリームを識別すると、推測的先読み処理を使用して、将来のクライアント読み出し要求によって要求される可能性があるデータブロックを読み出す。このような「先読み」ブロックは通常、ストレージシステムのディスクやメモリ（すなわち、バッファキャッシュ）から読み出され、各読み出しデータブロックは、異なるファイルシステム f b n に関連付けられる。従来の先読みアルゴリズムは、読み出しストリームを延長する所定数のデータブロックを「プリフェッチ」するように構成されることが多い。例えば、連続した番号の f b n が割り当てられた一連のデータブロックを読み出すためのクライアント読み出し要求を含む読み出しストリームの場合、その読み出しストリーム中のクライアント要求が先読みブロックをまだ要求していない場合であっても、ファイルシステムは、先読み処理を実施し、一連のデータブロックをさらに延長する f b n が割り当てられたデータブロックを更に読み出す場合がある。

30

40

【 0 0 1 1 】

一般に、先読み処理は、ファイルの読み出しストリームが、1セットのファイルオフセット又はメモリアドレスのうちの1つに到達したときに常に、それを「トリガ」として開始される。例えば、ファイルオフセットの所定のセットが、ファイル内の32番目ごと（31個置き）にあるファイルオフセット（すなわち、ファイルブロック番号0、32、61、・・・）から構成されるものと仮定する。さらに、既存の読み出しストリームが、f b n 番号4から始まり、f b n 番号27までであるものと仮定する。f b n 番号28 ~ 34の読み出しをストレージシステムに命じるクライアント読み出し要求を受信した場合、その要求は、所定の f b n 番号32を超えて延長され、それに応じて先読み処理が開始され

50

る。従って、従来の先読みアルゴリズムは、読み出しストリームにおける将来の読み出し要求を見込んで、f b n 番号 3 5 から始まる所定数のデータブロック、例えば 2 8 8 個のデータブロックをディスクからキャッシュに記憶する。

【 0 0 1 2 】

現在のストレージシステムの 1 つの欠点は、他のクライアント要求によって「中断」された読み出し要求の順序付きシーケンスを含む読み出しストリームを識別する機能を持たないことである。例えば、1 以上のランダム読み出し要求や他の読み出しストリームにおける要求によって順序付きシーケンスが中断された場合でも、ストレージシステムは、読み出しストリームに含まれる要求を非読み出しストリーム要求から区別することができない。従って、ストレージシステムは、識別されていない読み出しストリームについて、先読み処理を実施することができない。例えば、クライアントが、異なる読み出しストリームにおいて「重複する」読み出し要求を発行するものと仮定する。従来のストレージシステムにとって、そのインタリーブされたクライアント要求は、個別の読み出しストリームに属しているようには見えず、ランダムで無秩序な要求であるかのように見える。その場合ストレージシステムは、インタリーブされたいずれの読み出しストリームに対しても、先読み処理を実施することができない。同様に、ストレージシステムは、ランダムクライアント書き込み要求によってインタリーブされた要求を含む読み出しストリームに対して、先読み処理を実施しない場合がある。

10

【 0 0 1 3 】

従来のストレージシステムの別の欠点は、「ほぼ連続して」受信された複数の読み出し要求を含む読み出しストリームを識別する機能を持たないことである。読み出しストリーム中の要求のうちの 1 以上の無秩序化は、種々の理由から発生する。例えば、クライアントは、ストリーム読み出し要求を非連続に発行する場合がある。あるいは、ストレージシステムはクライアント要求を連続的に受信していても、クライアントが要求するデータを読み出す際の生来的な待ち時間によって、ストリーム読み出し要求を非連続的に処理する場合がある。一般に、ストレージシステムは、ほぼ連続した複数の読み出し要求を同じ読み出しストリームに属するものとして識別するようには構成されていない。従って、まだ識別されていない読み出しストリームに対しては、先読み処理が実施されない。

20

【 0 0 1 4 】

従って、要求が非読み出しストリーム要求とインタリーブされていたり、ほぼ連続的に並んでいたりする場合であっても、読み出し要求の順序付きシーケンスを識別するストレージシステムが必要とされている。また、ストレージシステムは、システムの性能に悪影響を及ぼすことなく、複数の読み出しストリームに対する先読み処理を同時に管理できるものでなければならない。

30

【 発明の開示 】

【 課題を解決するための手段 】

【 0 0 1 5 】

発明の概要

本発明は、各ファイルについて複数の異なる読み出しストリームの推測的先読みを同時に実施するように構成されたファイルシステムを実施するストレージシステムを提供する。従来の実施形態とは異なり、このファイルシステムは、複数の読み出しストリームのそれぞれについて、先読みメタデータの独立したセットを管理する。同時に、このファイルシステムは次いで、読み出しストリームに含まれる関連メタデータのセットに従って、要求の先読み処理を実施する。要求ごとに、受信したクライアント読み出し要求が、読み出しストリームに対応付けられるので、読み出し要求が非読み出し要求とインタリーブされていたり、ほぼ連続的に並んでいる場合であっても、ファイルシステムは先読み処理を実施することができる。

40

【 0 0 1 6 】

一実施形態によれば、先読みメタデータの各セットは、対応するリードセットデータ構造に記憶される。ファイルシステムは、ストレージシステムにおける各要求ファイルに対

50

し、ゼロ以上のリードセットからなる異なるセットを割り当てる。このように、各要求ファイルは、そのファイルに割り当てられたリードセット（すなわち、リードセット1つあたり1つ）の数に等しい数の同時読み出しストリームをサポートする。例示的实施形態において、ファイルのリードセットは、ファイルのデータを読み出すための初期要求の受信に応答して、動的に割り当てられる。割り当てられるリードセットの数は、ファイルサイズが大きくなるのに従って増大させることが望ましい。

【0017】

ファイルシステムは、ファイル読み出し要求を受信すると、その要求を、要求されたファイルに関連するリードセットに対応付けようとする。そのため、要求はファイルのリードセットと次々と比較される。この比較は好ましくは、最も最近アクセスされたリードセットから始めて、少なくとも1つの判断基準を満たすリードセットが見つかるまで続けられる。第1の判断基準は例えば、受信した要求が、そのファイルの前回識別された読み出しストリームのうちの1つを延長するものであるか否かのテストである。そうであれば、その延長された読み出しストリームに関連するリードセットは、「完全一致」であるものと判定される。第2の判断基準は例えば、受信した要求が、前回識別された読み出しストリームにおいて実施された最後の読み出しから所定の距離内に位置するデータを読み出すものであるか否かのテストである。第2の基準を満たしている場合、その読み出しストリームに関連するリードセットは、「曖昧一致」であるものと判定される。さらに第3の判断基準は例えば、要求ファイルのリードセットのうちのいずれかが「空」（すなわち、未使用）であるか否か、従って「空一致」であるとみなしてよいか否かを判定するものである。一致するリードセット（完全、あいまい、空）を見付けた後、ファイルシステムは、一致するリードセットに記憶された先読みメタデータに基づいて、先読み処理を実施する。

【0018】

ファイルシステムは、受信した読み出し要求に対して一致するリードセットが見つからなかった場合、要求ファイルの既存のリードセットのうちのどれが「再使用」できるかを判定するように構成される。その場合、受信した読み出し要求は、新たな読み出しストリームを開始することを仮定し、再使用されるリードセットは、その新たな読み出しストリームに関連する先読みメタデータを記憶するように構成される。各リードセットは、リードセットが再使用に適しているか否かの判定に使用されるエージング手段を有する。例えば、受信したクライアント読み出し要求が、クライアントが要求するファイルに関連するいずれのリードセットにも、完全一致、曖昧一致、又は空一致しないものと判定された場合常に、そのクライアント要求ファイルは、他のリードセットに対して「古い」。既存のリードセットの過剰な「スラッシング」を防止するために、エージング手段は、リードセット再使用ポリシーとともに使用される場合がある。例えば、再使用ポリシーは、ファイルに関連する少なくとも所定数の他のリードセットが再使用されるまで、リードセットが再使用されないことを保証するものである。従って、再使用ポリシーによれば、リードセットの内容が時期尚早に上書きされることが防止される。

【0019】

有利なことに、本発明によれば、ファイルシステムは、読み出しストリームに含まれるファイル読み出し要求が、ストレージシステムによって順番に受信されているか、ほぼ順番に受信されているか、それとも、無作為な順番で受信されているかに関わらず、複数の読み出しストリームに対して、先読み処理を同時に実施することができる。また、ファイルシステムは、読み出しストリームが異なる先読みアルゴリズムを使用する場合であっても、複数の異なる読み出しストリームに対して推測的先読みを同時に実施することができる。本発明は、ファイルベースのストレージシステムによっても、ブロックベースのストレージシステムによっても、あるいは、それらの組み合わせによっても実施することができる。

【0020】

本発明の上記の利点及び他の利点は、添付の図面とともに下記の説明を読むことにより

、よりよく理解できるであろう。図中、同じ参照符号は機能的に同一の要素又は類似の要素を示している

【発明を実施するための最良の形態】

【0021】

実施例の詳細な説明

A. ストレージシステム

図1は、ディスク160のような記憶装置上での情報の編成に関するストレージサービスを提供するように構成された、マルチプロトコル・ストレージ・アプライアンス100を示すブロック図である。ストレージディスクは、RAID (Redundant Array of Independent disks)のような種々の構成で配置される場合がある。ストレージ・アプライアンス100は例えば、システムバス115によって相互接続されたプロセッサ110、メモリ150、複数のネットワークアダプタ120、140、及び、ストレージ・アダプタ130からなるストレージシステムとして実施される。

【0022】

図示の実施形態において、メモリ150は、本発明に関連するソフトウェアプログラムコードやデータ構造を記憶するために、プロセッサ110やアダプタ120～140によってアドレス指定可能な記憶場所を有する。例えば、メモリは、1以上のinodeデータ構造を格納するinode「プール」152を記憶する場合がある。同様に、このメモリは、リードセットデータ構造を格納するリードセットプール154、及び、データバッファを格納するバッファプール156を記憶する場合がある。プロセッサ及びアダプタは、ソフトウェアコードを実行し、メモリ150に記憶されたデータ構造を操作するように構成された処理要素、及び/又は、論理回路を含む場合がある。ストレージオペレーティングシステム200は、通常その一部がメモリに常駐し、処理要素によって実行され、とりわけストレージアプライアンスによって実施されるストレージサービスを支援する記憶処理を実施することにより、ストレージアプライアンスの機能を構成する。当業者には明らかなように、本明細書に記載する本発明のシステムに関連するプログラム命令の記憶及び実行には、他の処理手段や、種々のコンピュータ読取可能媒体を含む他の記憶手段を使用してもよい。

【0023】

ディスク160に対するアクセスを容易にするために、ストレージオペレーティングシステム200は、仮想化モジュールと協働するwrite-anywhereファイルシステムを実施し、ディスク160によって得られる記憶空間を「仮想化」する。ファイルシステムは、情報を名前付きディレクトリ及びファイルの階層構造としてディスク上に編成する。「ディスク上」の各ファイルは、データのような情報を記憶するように構成された一連のデータブロックとして実施される一方、ディレクトリは、他のファイルやディレクトリの名前やそれらへのリンクが記憶される特殊形式のファイルとして実施される場合がある。仮想化モジュールによれば、ファイルシステムは、情報をブロックの階層構造としてディスク上でさらに論理編成し、それを名前付き論理ユニット番号(LUN)としてエクスポートすることができる。

【0024】

本明細書で使用されるように、「ストレージオペレーティングシステム」という用語は、コンピュータ上で動作するコンピュータ実行可能コードであって、データアクセスを管理し、さらに、マルチプロトコル・ストレージ・アプライアンスの場合は、データ・アクセス・セマンティックを実施するコンピュータ実行可能コードを一般に意味する。ストレージオペレーティングシステムは、カリフォルニア州サニーベイルにあるネットワーク・アプライアンス・インコーポレイテッドから市販されているData ONTAPオペレーティングシステムのように、マイクロカーネルとして実施される場合がある。また、ストレージオペレーティングシステムは、UNIXやWindows(R) NTのような汎用オペレーティングシステム、あるいは、本明細書に記載するストレージ・アプリケーションのために構成された、設定機能を備えた汎用オペレーティングシステム上で動作するアプリケ

ーションプログラムとして実施してもよい。当然ながら、本明細書に記載する本発明の原理に従って使用するために、任意の適当なストレージオペレーティングシステムに拡張を施してもよい。

【0025】

ストレージアダプタ130は、ストレージアプライアンス上で実行されるストレージオペレーティングシステム200と協働し、クライアント190によって要求された情報にアクセスする。情報は、ディスク160に記憶される場合もあれば、情報を記憶するように構成された他の同様の媒体に記憶される場合もある。ストレージアダプタは、従来のファイバ・チャネル(FC)シリアル・リンク・トポロジのようなI/O相互接続構成を介してディスクに接続された入出力(I/O)インタフェース回路を含む。情報はストレージアダプタによって取得され、システムバス115を介してネットワークアダプタ120、140へ転送される前に、必要であれば、プロセッサ110(又はアダプタ130自体)によって処理される。ネットワークアダプタ120、140において、情報はパケット又はメッセージとしてフォーマットされ、クライアントへ返される。

【0026】

ネットワークアダプタ120は、例えば、ポイント・ツー・ポイントリンク、ワイドエリアネットワーク(WAN)、公共ネットワーク(例えば、インターネット)上で実施される仮想私設ネットワーク(VPN)、又は、図示のイーサネット(R)ネットワーク175のような共有ローカルエリアネットワーク(LAN)を介して、ストレージアプライアンス100を複数のクライアント190a, bに接続する。従って、ネットワークアダプタ120は、ストレージアプライアンスを従来のイーサネット(R)スイッチ170のようなネットワークスイッチに接続するのに必要とされる機械的、電気的、又は、信号回路を備えたネットワークインタフェースカード(NIC)からなる場合がある。このNASベースのネットワーク環境の場合、クライアントは、マルチプロトコルアプライアンス上に格納されたファイルのような情報にアクセスするように構成される。クライアント190は、トランスミッション・コントロール・プロトコル/インターネット・プロトコル(TCP/IP)のような所定のプロトコルに従って個々のフレームデータ又はパケットデータをやりとりすることにより、ネットワーク175を介してストレージアプライアンスと通信する。

【0027】

クライアント190は、UNIXオペレーティングシステムやMicrosoft Windows(R)オペレーティングシステムのような種々のオペレーティングシステム上でアプリケーションを実行するように構成された汎用コンピュータであってもよい。クライアントは一般に、NASベースのネットワークを介して情報(ファイルやディレクトリの形をしている)にアクセスする場合、ファイルベースのアクセスプロトコルを使用する。従って、各クライアント190は、ネットワーク175を介してストレージアプライアンス100にファイルアクセスプロトコルメッセージ(パケットの形をしている)を発行することにより、ストレージアプライアンス100のサービスを要求する。例えば、Windowsオペレーティングシステムを実行しているクライアント190aは、CIFS over TCP/IPプロトコルを使用してストレージアプライアンス100と通信する場合がある。一方、UNIXオペレーティングシステムを実行しているクライアント190bは、NFS over TCP/IPプロトコル、又は、RDMA over TCP/IPプロトコルによるDAFS over VIプロトコルを使用して、マルチプロトコル・アプライアンスと通信する場合がある。当業者には明らかなように、他のタイプのオペレーティングシステムを実行するクライアントは、他のファイルアクセスプロトコルを使用して、統合マルチプロトコルストレージアプライアンスと通信する場合もある。

【0028】

ストレージネットワーク「ターゲット」アダプタ140は、マルチプロトコル・ストレージ・アプライアンス100をクライアント190に接続する。クライアント190は、

ブロック、ディスク又は論理ユニットとして記憶された情報にアクセスするように構成される場合がある。S A Nベースのネットワーク環境の場合、ストレージアプライアンスは、図示のF Cネットワーク185に接続される。F Cは、S A Nデプロイメントに主に見られるプロトコル式及び媒体を記載するネットワーク規格である。ネットワークターゲットアダプタ140は、アプライアンス100を従来のF Cスイッチ180のようなS A Nネットワークスイッチに接続するのに必要とされる機械的、電氣的、又は、信号回路を備えたF Cホストバスアダプタ(H B A)からなる場合がある。F C H B Aによれば、F Cアクセスが可能になる上、ストレージアプライアンスのファイバーチャネル・ネットワークプロセス・オペレーションの負荷も軽減される場合がある。

【0029】

クライアント190は一般に、S A Nベースのネットワークを介して例えばブロックやディスクの形をした情報にアクセスするときに、スモールコンピュータ・システム・インタフェース(S C S I)プロトコルのようなブロックベースのアクセスプロトコルを使用する。S C S Iは、ディスク160のような種々の周辺機器をストレージ・アプライアンス100に取り付けることが可能な標準的なデバイス非依存のプロトコルを備えた周辺機器I/Oインタフェースである。S C S I用語において、S A N環境で動作しているクライアント190は、データの要求およびコマンドを開始する「イニシエータ」である。従って、マルチプロトコル・ストレージ・アプライアンスは、要求/応答プロトコルに従ってイニシエータにより発行された要求に対して応答する「ターゲット」である。クライアントがS A Nベースのデータアクセス要求をストレージ・アプライアンスに送信するとき、クライアントは通常、ディスク160に記憶された個々のデータブロックに対応する論理ブロックアドレスを使用する。

【0030】

マルチプロトコル・ストレージ・アプライアンス100は、T C P / I P上にカプセル化されたS C S Iプロトコル(i S C S I)や、F C上にカプセル化されたS C S I(F C P)のような、S A Nデプロイメントにおいて使用される種々のS C S Iベースのプロトコルをサポートする。従ってイニシエータ(以後クライアント190)は、ネットワーク175、185を介してi S C S IメッセージやF C Pメッセージを送信することにより、ディスクに記憶された情報にアクセスする。当業者には明らかなように、クライアントは、他のブロックアクセスプロトコルを使用して、統合マルチプロトコルストレージアプライアンスのサービスを要求する場合もある。複数のブロックアクセスプロトコルをサポートすることにより、マルチプロトコル・ストレージ・アプライアンスは、異種S A N環境におけるディスクや論理ユニットに対する統一的で整然としたアクセス手段を提供する。

【0031】

B. ストレージオペレーティングシステム

図2は、本発明と共に使用するのに都合がよいストレージオペレーティングシステム200の一例を示す略ブロック図である。ストレージオペレーティングシステムは、統合ネットワークプロトコルスタック、すなわち、より一般的には、マルチプロトコルエンジンを形成するように編成された一連のソフトウェア層から構成され、ブロックアクセスプロトコルとファイルアクセスプロトコルを使用して、マルチプロトコル・ストレージ・アプライアンス100に記憶された情報にアクセスするためのデータパスをクライアントに提供する。プロトコルスタックは、I P層212のネットワークプロトコル層、並びに、I P層が支えている搬送手段であるT C P層214及びユーザ・データグラム・プロトコル(U D P)層216とのインタフェースとして機能するネットワークドライバ(例えば、ギガビット・イーサネット(R)・ドライバ)のメディアアクセス層210を含む。ファイルシステムプロトコル層はマルチプロトコルファイルアクセスを提供し、その目的のためにD A F Sプロトコル218、N F Sプロトコル220、C I F Sプロトコル222、及び、ハイパーテキスト・トランスファー・プロトコル(H T T P)プロトコル224を含む。V I層226は、D A F Sプロトコル218に必要とされるようなダイレクト・ア

クセス・トランスポート (D A T) 機能を提供する V I アーキテクチャを実施する。

【 0 0 3 2 】

i S C S I ドライバ層 2 2 8 は、 T C P / I P ネットワークプロトコル層を介したブロックベースのプロトコルアクセスを提供する一方、 F C ドライバ層 2 3 0 は、 F C H B A 1 4 0 と共に動作し、ブロックアクセス要求の送受信を行うとともに、クライアント 1 9 0 a , b との間で応答のやりとりをする。 F C ドライバ及び i S C S I ドライバは、 F C 固有のアクセス制御及び i S C S I 固有のアクセス制御をストレージディスク 1 6 0 及び他の論理ユニットに提供する。更に、ストレージオペレーティングシステム 2 0 0 は、 R A I D プロトコルのようなディスクストレージプロトコルを実施するだけでなく、例えば S C S I プロトコルのようなディスクアクセスプロトコルに従ってストレージディスク 1 6 0 からデータブロックを読み出すための、ディスクドライバサブシステム 2 5 0 も実施する。

10

【 0 0 3 3 】

ディスクソフトウェア層 2 4 0 及び 2 5 0 を統合ネットワークプロトコルスタック層 2 1 0 ~ 2 3 0 に橋渡しするのは、仮想化システムである。仮想化システムは、例えば仮想ディスク (「 v d i s k 」) モジュール 2 7 0 及び S C S I ターゲットモジュール 2 3 5 として実施される仮想化モジュールと通信するストレージマネージャ又はファイルシステム 2 6 0 によって実施される。 v d i s k モジュール 2 7 0 は、ファイルシステム 2 6 0 の上に層として形成され、ユーザがストレージシステムに対して発行したコマンドに応答して、ユーザインタフェース (U I) 2 7 5 のような管理インタフェースによってアクセスすることができる。 S C S I ターゲットモジュール 2 3 5 は、 F C 2 3 0 及び i S C S I 2 2 8 と、ファイルシステム 2 6 0 との間に配置され、ブロック (L U N) 空間とファイルシステム空間との間に仮想化システムの変換層を提供する。ファイルシステム空間において、 L U N は仮想ディスクとして表わされる。 U I 2 7 5 は、ストレージオペレーティングシステムの上に配置され、種々の層、及び、 R A I D サブシステム 2 4 0 のようなサブシステムに対する管理アクセスやユーザアクセスを可能にする。

20

【 0 0 3 4 】

ファイルシステム 2 6 0 は例えば、ディスク 1 6 0 のような記憶装置に記憶された情報にアクセスするためのボリューム管理機能を提供する。すなわち、ファイルシステム 2 6 0 は、ファイルシステムセマンティックを提供する他に、ボリュームマネージャに通常関連する機能も提供する。そうした機能には、 (i) ディスクの集合化、 (i i) ディスクのストレージ帯域幅の集合化、 (i i i) ミラーリング、及び / 又は、パリティの信頼性確保 (R A I D) などがある。ファイルシステム 2 6 0 は例えば、 Write Anywhere File Layout (WAFL) ファイルシステムを実施し、 「ディスク上」のデータを固定サイズのブロック、例えば 4 キロバイト (k B) のブロックを使用して編成する。図示のファイルシステム 2 6 0 は、インデックス i n o d e (「 i n o d e 」) を使用してファイルを識別し、ファイル属性 (作成時刻、アクセス許可、サイズ、ブロック位置など) を記憶する。 i n o d e ファイルを含めて、 i n o d e の使用の詳細については、 1 9 9 8 年 1 0 月 6 日に発行された David Hitz 他による 「 Method for Maintaining Consistent Status of a File System and for Creating User-Accessible Read-Only Copies of a File System 」 と題する米国特許第 5 , 8 1 9 , 2 9 2 号に記載されており、この文献は参照により、本明細書の中で完全に説明されたものとして本明細書に援用される。

30

40

【 0 0 3 5 】

図 3 は、ファイル 3 3 0 のバッファツリーを示す略ブロック図である。このバッファツリーは、メモリに記憶されたファイルのブロックの内部表現である。バッファツリーは、ファイル 3 3 0 を表わすメタデータ及びファイルのサイズに係るメタデータを含み、更にそのファイルの実際のデータが格納された例えば 4 キロバイトブロックのデータブロック 3 2 0 を参照するポインタを含む最上位 i n o d e 3 0 0 を有する。特に、大きなファイル (例えば、 6 4 k B よりも大きなデータ) の場合、 i n o d e 3 0 0 にある各ポインタは、最大 1 0 2 4 個のポインタが格納される間接 (レベル 1) ブロック 3 1 0 を参照する

50

場合があり、各ポインタはデータブロック 320 を参照することができる。一例として、間接ブロック 310 にある各ポインタは、ファイルシステム 260 におけるデータブロック 320 に対応する *vbn* を識別する値を記憶する場合がある。

【0036】

動作に関し、ファイルシステム 260 は、統合ネットワークプロトコルスタックの種々のソフトウェア層によって処理されたクライアント要求を受信する。例えば、ネットワークアダプタ 120 又は 140 において受信されるクライアント要求は、(層 210 又は 230) のネットワークドライバによって処理され、適宜、その要求が更にネットワークプロトコル及びファイルアクセス層 212 ~ 228 へ転送され、処理される。次にクライアント要求はファイルシステム 260 へ渡すことが可能なファイルシステム「メッセージ」としてフォーマットされる。このメッセージは、とりわけ、クライアントが要求したファイル又はディレクトリ(例えば、通常は *inode* 番号によって表される)、要求されたファイル又はディレクトリ内の開始オフセット、及び、開始オフセットの後に書き込み又は読み出しするデータの長さを指定するものである。

【0037】

ファイルシステム 260 は、ディスク上のデータを固定サイズのデータブロック、例えば 4 キロバイトブロックを単位として操作するため、受信したファイルシステムメッセージがまだフォーマットされていない場合、その中の値(*inode*、オフセット、長さ)をデータブロック(例えば、*fbn*)単位に変換しなければならない場合がある。例えば、8 キロバイトのクライアント要求ファイルが、ディスク上で、*fbn* 11 及び 12 がそれぞれ割り当てられた 2 つの連続した 4 キロバイトブロックを占めるものと仮定する。更に、それら 2 つのデータブロックが、*inode* 番号 17 を有する *inode* に記憶された 1 セットのポインタを使用してアクセス可能であるものと仮定する。そして、クライアントは、そのファイルデータのうちの後ろ 6 キロバイト分に対するアクセス、すなわち、*fbn* 番号 11 における最後の 2 キロバイトと *fbn* 番号 12 における 4 キロバイト全部に対するアクセスを要求しているものと仮定する。その場合、ファイルシステム 260 は、要求データが(*inode* = 17、ファイルオフセット = 2 kB、長さ = 6 kB)として指定されたファイルシステムメッセージを受信する場合がある。ファイルシステムはデータブロック単位でデータを操作するため、受信したファイルオフセットおよび長さ値をデータブロック単位に変換し、どのデータブロックがクライアントの要求するデータを有しているかを識別できるようにする。例えば、(*inode* = 17、開始データブロック = *fbn* 11、読み出すべきデータブロック = 2 ブロック)というように。

【0038】

どのデータブロックがクライアントの要求データを記憶しているかを識別した後(例えば、*fbn* 11 及び 12 といったように)、ファイルシステム 260 は、クライアントの要求するデータブロックが、「コア内」バッファのうちの 1 以上から得られるか否かを判定する。得られる場合、ファイルシステムは、要求されたデータをメモリ 150 から読み出し、読み出したデータをクライアント要求に従って処理する。しかしながら、要求されたデータがコアメモリ内にない場合、ファイルシステム 260 は、要求されたデータをストレージディスク 160 からロード(読み出し)する処理を発生する。ファイルシステムは、クライアントの要求するデータブロックに割り当てられた *vbn* 番号(すなわち、*vbn* 11 と 12)を RAID サブシステム 240 に渡し、RAID サブシステム 240 は、それらの *vbn* を対応するディスクブロック番号(*dbn*)にマッピングし、後者をディスクドライバサブシステム 250 のディスクドライバのうちの適当なドライバ(例えば SCSI ドライバ)に送信する。ディスクドライバはディスク 160 の要求された *dbn* にアクセスし、要求されたデータブロック(複数の場合もあり)をメモリ 150 にロードし、それをファイルシステム 260 によって処理する。

【0039】

また、ファイルシステム 260 は、クライアントが要求するデータを含むデータブロックの読み出しの他に、更に、ディスクソフトウェア層 240 及び 250 に対し、ディスク

10

20

30

40

50

160からの「先読み」データブロックの読み出しを命じる場合がある。こうした先読みデータブロックは、受信したクライアント要求に含まれる読み出しストリームを論理的に延長する範囲のデータブロック（例えば、f b n）に対応する。ただし、それらの先読みブロック自体はまだ要求されてはいない。クライアントが要求するデータブロックと同様に、先読みデータブロックも、ディスクソフトウェア層240及び250によって読み出され、ファイルシステム260にとってアクセス可能な適当なメモリバッファにコピーされる。そのようなメモリバッファは、バッファプール156から得ることができる。ファイルシステムは、クライアントの要求に従って読み出されたデータブロックにおけるクライアントの要求するデータにアクセス（すなわち、読み出し又は書き込み）し、適宜、要求されたデータ、及び/又は、受領応答メッセージを要求元クライアント190に返すことができる。

10

【0040】

C. リードセット

本明細書で使用されるように、「読み出しストリーム」は、要求されたファイル内の論理的に連続した範囲の幾つかのファイルオフセット（例えば、f b n）からのデータの読み出しをストレージオペレーティングシステム200に対して命じる1以上のクライアント要求のセットとして定義される。オペレーティングシステムは、将来のクライアント読み出し要求による読み出しストリームにおいて要求される可能性がある1以上のデータブロックをプリフェッチするための推測的先読み処理を使用する場合がある。一実施形態によれば、ストレージオペレーティングシステム200は、同時に管理される複数の読み出しストリームのそれぞれについて個別に、先読みメタデータのセットを有する。例示的实施形態において、ストレージオペレーティングシステムは、各読み出しストリームのメタデータを個別の「リードセット」データ構造に記憶する（すなわち、1つのリードセットあたり1つの読み出しストリーム）。従って、複数の読み出しストリームを同時にサポートするファイルやディレクトリは、複数の異なるリードセットに関連する場合があり、例えば、ファイルやディレクトリに関連するi n o d eを使用してアクセスされる場合がある。

20

【0041】

図4は、例示的なi n o d e 400及びそれに関連するリードセット600 a ~ cのセットを示している。i n o d e 400は、とりわけ、i n o d e 番号402（又は、他の識別子）、リードセットポインタ404、読み出しアクセススタイル406、デフォルト先読み値408、ファイルメタデータ410、及び、データセクション412を含む。i n o d e 400は、そのi n o d eに関連するファイルやディレクトリ中のデータに対するアクセスを求めるクライアント要求をストレージシステム200が受信するのに応じて、i n o d e プール152から割り当てられ、すなわち、取得される。i n o d e 400に関連するファイルやディレクトリを一意に識別するために、i n o d e 番号402（例えば、この例では17）が使用される場合がある。例えば、クライアント要求は、クライアントがアクセスを要求している特定範囲のデータを含むファイルやディレクトリに関連するi n o d e 番号を指定する場合がある。クライアントが指定するi n o d e 番号は、そのファイル内の開始オフセットの指示、又は、その開始オフセットから始まるアクセスしたいデータの長さに結合される場合がある。

30

40

【0042】

リードセットポインタ404は、ゼロ以上のリードセットデータ構造600の記憶場所を指定する値を記憶する。動作に関し、ファイルシステム260は、リードセットを動的に割り当ててもよいし、あるいは、リードセットプール154から事前に割り当てられたリードセットを取得してもよい。i n o d e 400に割り当てられる各リードセットは、所定の値のセットを記憶するように初期化される場合がある。例えば、i n o d e 400に関連するリードセット600 a ~ cは、連結リストとして構成され、各リードセットは、リスト上の隣りのリードセットの記憶場所を示す値を記憶する。リストに含まれる最後のリードセット、例えばリードセット600 cにおけるネクストポインタは、そのリード

50

セットがリスト上で末尾に位置することを示す所定の「ヌル（空）」値を記憶する場合がある。図示の実施形態におけるリードセットは連結リストとして構成されているが、当業者であれば、リードセットをサーチツリーのような他の構成にしてもよいことは明らかであろう。

【 0 0 4 3 】

読み出しアクセススタイル 4 0 6 は、i n o d e 4 0 0 に関連するファイルやディレクトリからデータを読み出す方法を表わす読み出しアクセスパターンを示す値を記憶する。例えば、読み出しアクセススタイルは、例えば、通常アクセスパターン、連続アクセスパターン、又は、ランダムアクセスパターン等に従って、i n o d e のファイル又はディレクトリにあるデータが読み出されることを示す場合がある。ストレージオペレーティングシステム 2 0 0 は、クライアント読み出し要求を処理する際に、読み出しアクセスパターンを動的に識別し、更新する場合がある。あるいは、ストレージオペレーティングシステムは、受信したクライアント読み出し要求に含まれる「キャッシュヒント」等に基づいて、その読み出しアクセス値を設定する場合がある。キャッシュヒントは、要求元クライアントがファイルやディレクトリからのデータの読み出しに使用する可能性がある読み出しアクセスパターンを示す。例えば、ストレージオペレーティングシステムは、クライアントから送信された D A F S 読み出し要求からキャッシュヒントを得る場合がある。D A F S キャッシュヒントを含む D A F S キャッシュプロトコルの詳細については、2 0 0 1 年 9 月 1 日に出版された「DAFS: Direct Access File System Protocol, Version 1.00」に記載されており、この文献は、参照により完全に説明されたものとして本明細書に援用される。

【 0 0 4 4 】

デフォルト先読み値 4 0 8 は、i n o d e 4 0 0 に関連するファイルやディレクトリに記憶されたデータを読み出すための将来のクライアント読み出し要求を想定してプリフェッチ（例えば、前もっての読み出し）されることがある所定数のデータブロックを示す。例えば、デフォルト先読み値 4 0 8 は、クライアントが要求するデータを有する 1 以上のデータブロックを読み出した後、将来のクライアント読み出し要求を想定して、ファイルシステムがデータブロック、例えば 2 8 8 個のデータブロックを更に読み出さなければならないことを示す場合がある。当業者には分かるように、クライアント読み出し要求がある度に「先読み」データブロックを読み出す必要はなく、先読みデータブロックは、所定の先読みアルゴリズムに基づいて取得してもよい。図示の実施形態によれば、デフォルト先読み値 4 0 8 は、読み出しアクセススタイル 4 0 6 に応じて決まる場合がある。例えば、ランダム読み出しアクセスパターンの場合、デフォルト先読み値はゼロに設定され、通常読み出しアクセスの場合よりも、連続読み出しアクセスの場合の方が、デフォルト先読み値は大きな値に設定される。

【 0 0 4 5 】

ファイルメタデータ 4 1 0 は、i n o d e 4 0 0 に関連するファイルやディレクトリに関連する他のメタデータ情報を記憶する。こうしたメタデータ情報には、とりわけ、ユーザ識別子、グループ識別子、アクセス制御リスト、フラグ、及び、他のデータ構造へのポインタ等のようなセキュリティ証明がある。i n o d e 4 0 0 は、その i n o d e に関連するファイルやディレクトリを含むデータブロック 3 2 0 の記憶場所を（直接的又は間接的に）示す 1 セットのポインタを含むデータセクション 4 1 2 を更に有する。この例の場合、データセクション 4 1 2 にあるポインタは 1 以上の間接データブロック（図示せず）を指し、更に、それらの間接データブロックが、ファイルやディレクトリを含む 1 セットの連続したデータブロックの記憶場所を示すポインタを有する。以後、i n o d e 4 0 0 からアクセス可能な各データブロックには、対応する f b n が割り当てられ、i n o d e 4 0 0 に関連するファイル（又はディレクトリ）は、連続的な f b n 値が割り当てられた 1 セットのデータブロックからなるものと仮定する。例えば、データセクション 4 1 2 にある幾つかのポインタは、ファイルのうち、f b n 9 ~ 1 8 が割り当てられたデータブロックに記憶された部分を指す場合がある。

【 0 0 4 6 】

有利なことに、`inode 4 0 0`のファイルやディレクトリを含む複数のデータブロック 3 2 0 において、複数の読み出しストリームを同時に確立することができる。例えば図示のように、データブロック 9 ~ 1 8 のセットには、並列な 2 本の読み出しストリーム 4 3 0 及び 4 3 5 が識別される。読み出しストリーム 4 3 0 は、ファイルシステム 2 6 0 によって読み出されたファイルブロック番号 9 まで（ただし 9 は含まない）の `fbn` の論理的に連続したシーケンスに対応する。図示の実施形態によれば、各読み出しストリームは、リードセット 6 0 0 a ~ c のうちの異なる 1 つに記憶された対応する先読みメタデータのセットに関連する場合がある。

【 0 0 4 7 】

10

上記のように、各リードセットは、対応する読み出しストリームに関連するメタデータを記憶するように構成される。従って、図示の `inode 4 0 0` は 3 つのリードセット 6 0 0 a ~ c に関連しているため、この `inode` に関連するファイルやディレクトリは、最大で 3 つの異なる読み出しストリームをサポートする。ただし、当然ながら、`inode` には、0 以上の任意数のリードセットを割り当てるのが可能であるものと考えられる。`inode 4 0 0` に割り当てられるリードセットの数は、その `inode` に関連するファイルやディレクトリのサイズに基づいて決定されることが好ましい。例えば、ファイルのサイズが増大するほど、`inode` に割り当てられるリードセットの数も増大するといったように。

【 0 0 4 8 】

20

図 5 は、列 5 1 0 に記憶されたファイルサイズを列 5 2 0 に記憶された対応する割り当てリードセット数に関連付けるために使用されるテーブル 5 0 0 の例を示している。この例の場合、「非常に小さな」ファイル（例えば 6 4 k B 未満）は、読み出しストリームを形成できるほどのデータを有していないので、ゼロのリードセットが関連付けられる。一方「小さな」ファイル（例えば 6 4 k B ~ 5 1 2 k B）は、1 つの読み出しストリームを形成することが可能なサイズを有しているので、単一のリードセットが関連付けられる。一般に、ファイルサイズが大きくなるほど、ファイルがサポートする読み出しストリームの数も多くなるので、そのファイルの `inode` に割り当てられるリードセットの数も増える。例えば、1 以上のクライアント「書き込み」要求を処理した結果、ファイルサイズが動的に増大するのに従って、ファイルシステム 2 6 0 は更なるリードセットを動的に割り当てることができる。

【 0 0 4 9 】

30

図 6 は、リードセットポインタ 4 0 4 によってアクセス可能なリードセット 6 0 0 の例を示している。このリードセットは、読み出しストリーム 4 3 0 又は 4 3 5 のような対応する読み出しストリームに関連するメタデータを有する。リードセット 6 0 0 は、とりわけ、ネクストポインタ 6 0 2、レベル値 6 0 4、カウント値 6 0 6、最終読み出しオフセット値 6 0 8、最終読み出しサイズ 6 1 0、ネクスト先読み値 6 1 2、先読みサイズ 6 1 4、及び、種々のフラグ 6 1 6 を含む。当業者には分かるように、リードセット 6 0 0 は、図示したものの他にも、他の情報を更に記憶するように構成される場合がある。既に説明したように、ネクストポインタ 6 0 2 は、リードセットのリスト（又は他のデータ構造）上にある次のリードセットの記憶場所を示す値を記憶する。

【 0 0 5 0 】

40

レベル値 6 0 4 は、リードセット 6 0 0 の相対的「古さ」を示す。好ましくは、レベル値は、所定の上限值と所定の下限値を境界とする範囲内の整数値である。例えば、レベル値 6 0 4 は、所定のゼロの下限值と所定の 2 0 の上限値の間の整数に制限される場合がある。リードセット 6 0 0 を最初に割り当てるとき、レベル値 6 0 4 は、リードセット 6 0 0 が未使用（すなわち、空）であることを示す例えばマイナス 1 のような特殊な指示値に設定される。新たに識別された読み出しストリームにリードセットを割り当てるとき、レベル値 6 0 4 は、上限値と下限値の間の範囲内の所定の「初期」値に設定される。例えば、所定下限値と所定の上限值がそれぞれ 0 と 2 0 である場合、初期レベル値は、それらの

50

間にある10のような任意の値に設定される。リードセットが時期尚早に古くなることを防止するために、リードセット600に関連する大きなファイル（又は、ディレクトリ）に使用される初期レベル値は、小さなファイルに使用される初期レベル値よりも大きくすることが望ましい。例えば、非常に大きなファイル（例えば10GBよりも大）の場合、初期レベル値は15に設定され、他の場合は10に設定される場合がある。当然ながら、本発明のコンテキストでは、他の上限値、下限値、及び、初期レベル値を使用してもよい。

【0051】

クライアント読み出し要求がファイルシステム260によって処理されるたびに、ファイルシステムは、そのクライアント読み出し要求を含む読み出しストリームに関連するリードセットにおけるレベル値604をインクリメントする。例えば、レベル値604をインクリメントする場合、レベル値604は、例えば1に等しい第1の所定ステップサイズだけ増加される。また、後で説明するようにリードセットを「再使用」する場合、ファイルシステムは、各リードセットに記憶されたレベル値が再使用のために選択されていないものと判断する。レベル値604は、例えば1に等しい第2の所定ステップサイズだけデクリメントされる場合がある。

【0052】

例えば、リードセット600が、12に等しいレベル値604を記憶し、ファイルシステム260が、クライアント読み出し要求に対応するファイルシステムメッセージを受信するものと仮定する。ファイルシステムは、そのクライアント要求がリードセット600に関連する読み出しストリームに属しているものと判断した場合、レベル値604を13にインクリメントする。一方、そのクライアント要求が、別の読み出しストリーム、すなわち、リードセット600に関連しない読み出しストリームに属しているものと判断した場合、ファイルシステムは、レベル値604を変更せずに12のまま維持する。更に、受信したクライアント読み出し要求によって、ファイルシステムが別のリードセットを再使用しなくればならなくなった場合、ファイルシステムは、レベル値604を例えば11にデクリメントする。このように、クライアント読み出し要求がファイルシステム260によって処理されるたびにレベル値604は調節され、以下で説明するように、リードセット600が割り当て解除、又は、再使用されるまで調節され続ける。このエージングプロセスは、種々の条件の影響を受ける場合がある。例えば、レベル値604が、所定の初期値（例えば、10）よりも小さいものと判定された場合、次回、レベル値はインクリメントされ、レベル値が所定の初期レベル値に等しくなるように設定される場合がある。さらに、ファイルシステム260は、レベル値604が、所定の上限値を超えてインクリメントされたり、所定の下限値を超えてデクリメントされたりしないようにする。

【0053】

カウント値606は、リードセット600に関連する読み出しストリームにおいて処理されたクライアント読み出し要求の数を記憶する。図示の実施形態の場合、カウント値は最初にゼロに設定される。次に、そのリードセットに関連する読み出しストリームに含まれるクライアント読み出し要求をファイルシステム260が一回処理するたびに、そのカウント値はインクリメントされる。レベル値604の場合と同様に、マルチプロトコル・ストレージ・アプライアンス100のメモリリソースを節約するために、カウント値604は、所定の上限値、例えば 2^{16} によって制限される。

【0054】

最終読み出しオフセット608と最終読み出しサイズ610は合わせて、リードセット600に関連する読み出しストリームにおいて最後に処理された（すなわち、最も最近処理された）クライアント読み出し要求を表わす。最終読み出しオフセット608及び最終読み出しサイズ610は、リードセット600に関連する読み出しストリームにおいて受信された最後のクライアント読み出し要求の処理に応答して、ファイルブロック番号6から始まる3つのデータブロック（すなわち、f b n 番号6、7及び8）を読み出すことが望ましい。その場合、最終読み出しオフセット608はf b n 番号6に設定され、最終読

み出しサイズ 6 1 0 は 3 データブロックに設定される。従って、将来のクライアント読み出し要求がファイルシステムに対し、ファイルブロック番号 9 から始まる論理的に連続したデータブロックの他のシーケンスの読み出しを要求するものである場合、そのクライアント読み出し要求は、リードセット 6 0 0 に関連する読み出しストリームを「延長」する場合がある。

【 0 0 5 5 】

ネクスト先読み値 6 1 2 は、リードセット 6 0 0 に関連する読み出しストリームに対し、ファイルシステム 2 6 0 が次の先読み処理のセットを実施する場所である所定のファイルオフセット又はメモリアドレスの指示を記憶する。具体的には、クライアント読み出し要求が、ネクスト先読み値 6 1 2 が示すファイルオフセット又はメモリアドレスを越えて読み出しストリームを延長するものである場合、ファイルシステムは、将来のクライアント読み出し要求を見越して、読み出しストリームを更に延長する先読みデータブロックのセットを更に推測的に読み出す場合がある。先読みサイズ値 6 1 4 は、プリフェッチされる先読みデータブロックの数を記憶する。先読みサイズ値 6 1 4 は、デフォルト先読み値 4 0 8 であってもよいし、あるいは、先読みアルゴリズムに従って決定されるものであってもよい。先読みデータブロックを読み出した後、ファイルシステム 2 6 0 は、その読み出しストリームに対して先読み処理が実施される場所であるネクストファイルオフセット又はメモリアドレスを示すようにネクスト先読み値 6 1 2 を更新する。先読みデータブロックを読み出した後、ファイルシステムは、それらをメモリ 1 5 0 における適当なコア内メモリバッファにコピーし、ファイルシステムはそのクライアント読み出し要求の処理を完了する。

【 0 0 5 6 】

各リードセットは、ファイルシステム 2 6 0 がそのリードセットに関連する読み出しストリームに対して特殊な先読み処理を実施できるようにするための 1 以上のフラグ値 6 1 6 を含む場合がある。例えば、そうしたフラグ値の 1 つは、ファイルシステムがその読み出しストリームに対して推測的にデータブロックを読み出すべき「方向」を示すものである場合がある。すなわち、ファイルシステムは、論理的に「前進」方向（すなわち、データブロック番号が増大する方向）にデータブロックを読み出すように構成される場合もあれば、論理的に「後退」方向（すなわち、データブロック番号が減少する方向）にデータブロックを読み出すように構成される場合もある。他のフラグ値 6 1 6 は、先読みデータブロックが「一回読み出し」データを有している否か、すなわち、延長された時間の間メモリ 1 5 0 に記憶しておかなければならないか否かを示すものである場合がある。

【 0 0 5 7 】

D . リードセットに対するクライアント要求のマッチング

クライアント読み出し要求を受信すると、ファイルシステム 2 6 0 は、その要求が既存のリードセット 6 0 0 に「一致」するものであるか否かを判定する。例示的な実施形態によれば、要求が少なくとも 1 つの所定の基準を満たす場合、その要求はリードセットに一致するものであると判定される。例えば、受信した要求が既に識別された読み出しストリームの「延長」を要求するものであるか否かを調べるために、第 1 の基準をテストする場合がある。延長を要求するものである場合、延長読み出しストリームに関連するそのリードセットは、「完全一致」であるものと判定される。受信した要求が既に識別された読み出しストリーム中で実施された最後の読み出しから所定の距離内に位置するデータの読み出しを要求するものであるか否かを調べるために、第 2 の基準をテストする場合がある。第 2 の基準を満たしている場合、読み出しストリームに関連するリードセットは、「曖昧一致」であるものとみなされる。クライアントが要求するファイル又はディレクトリに関連するリードセットのうちのいずれかが「空」（すなわち、未使用）であるか否か、すなわち、「空一致」であるか否かを判定するために、さらに第 3 の基準をテストする場合がある。一致するリードセット（完全、曖昧、又は、空）を見つけた後、オペレーティングシステムは、一致するリードセットに記憶された先読みメタデータに基づいて、先読み処理を実施する。

【 0 0 5 8 】

図7は、読み出しストリーム435を論理的に延長する例示的なクライアント読み出し要求700を示している。具体的には、クライアント読み出し要求は、マルチプロトコル・ストレージ・アプライアンス100において受信され、ストレージオペレーティングシステム200により実施される統合ネットワークプロトコルスタックの1以上の層によって処理される。プロトコルエンジン218～230のうちの1つのようなファイルシステムプロトコルエンジンは、受信したクライアント要求を、ファイルシステム260へ転送されるファイルシステムメッセージとしてフォーマットする。ファイルシステムメッセージは、ファイルシステムがクライアントの要求するデータを読み出すための種々の情報を含む。例えば、ファイルシステムメッセージは、とりわけ、i n o d e 番号の指示、ファイルオフセット、及び、読み出すべきデータの長さを含む場合がある。この例では、ファイルシステムメッセージがクライアント読み出し要求700として実施され、その中に、ファイルオフセットや読み出すべきデータの長さがデータブロック単位で指定される。具体的には、読み出し要求700は、とりわけ、i n o d e 番号702、開始データブロック704、及び、読み出すべきデータブロックの数706を含む。

10

【 0 0 5 9 】

説明のために、i n o d e 番号は17、開始データブロック番号（例えばf b n）は15、読み出すべきデータブロック数は2であるものと仮定する。従って、クライアント読み出し要求700はファイルシステム260に対し、i n o d e 番号17に関連するファイル又はディレクトリ内のファイルデータブロック15及び16を探すように命じる。ファイルシステムはまず、自分のコア内のメモリバッファにあるデータブロックを調べ、以前に処理されたクライアント要求の結果としてそれらのデータブロックが読み出されたか否かを判定する。データブロック15及び15のうちの一方又は両方がメモリバッファに存在しない場合、ファイルシステム260は、ストレージサブシステム250（例えば、R A I D層及びディスクドライバ層）と協働し、見付からなかったデータブロックをストレージディスク160から読み出す。その場合、ディスクから読み出したデータブロックは、例えばバッファプール156から得られる1以上のメモリバッファにコピーされる。

20

【 0 0 6 0 】

図7は、読み出しストリーム435に関連するリードセット600の部分を示している。リードセットに記憶された値608及び610によって示されるように、読み出しストリーム435において処理された最後のクライアント読み出し要求の結果、ファイルシステムは、ファイルブロック番号13から始まる論理的に連続した並びの2つのデータブロック（すなわち、f b n 番号13と14）を読み出している。クライアント読み出し要求700における開始データブロック値704が、ファイルシステムに対してf b n 番号15及び16の読み出しを命じるものであることから、ファイルシステムは、その要求が、読み出しストリーム435を論理的に延長するものであることを判断することができる。従って、読み出しストリーム435に関連するリードセット600は、クライアント読み出し要求700に対して完全一致であるものとみなされる。

30

【 0 0 6 1 】

ファイルシステムが、受信したファイルシステム読み出し要求700に応答して、ファイルブロック番号15及び16（影付きデータブロックとして描かれている）を読み出すため、読み出しストリーム435は、ネクスト先読み値612によって指定されたf b n 番号の開始点16を超えて延長される。従って、ファイルシステム260は、読み出しストリーム435における次の論理データブロック（すなわち、f b n 番号17）から開始して、先読みサイズ値614によって指定されるような50個の先読みデータブロックを読み出す。読み出される先読みデータブロックの数は先読みサイズ値614を使用して決定することが好ましいが、先読みデータブロックの数は代わりに、i n o d e 番号17に記憶されたデフォルト先読みサイズ406のような他の情報を使用して決定してもよい。

40

【 0 0 6 2 】

ファイルシステム260は、クライアントが要求するデータブロック15及び16を読

50

み出すときと同じ又は同様のやり方で、先読みデータブロックを読み出す。すなわち、ファイルシステムはまず、コア内メモリバッファからの先読みデータブロックの読み出しを試行し、次に、コア内バッファに存在しない先読みデータブロックを、ストレージサブシステム 250 と協働して、ストレージディスク 160 から読み出す。ディスクから読み出されたクライアントの要求するデータブロックと同様に、先読みデータブロックも、コア内データバッファにコピーされる。ただし、先読みデータブロックの推測的性質から、すなわち、先読みデータブロックはクライアント 190 によって明示的に要求されたものではないので、先読みデータを保持するコア内メモリバッファは、クライアントから明示的に要求されたデータブロックを保持する時間に比べて短い時間しか先読みデータブロックをメモリ 150 に保持しないように構成される場合がある。

10

【0063】

また、ファイルシステム 260 は、先読みデータブロックを読み出す際に、読み出しストリーム 435 に関連する他の情報にも依存する場合がある。例えば、読み出しストリーム 435 がネクスト先読み値 612 によって指定されたデータブロック番号又はメモリアドレスを超えて延長された場合でも、フラグ 616 の値は、先読みデータブロックの読み出しの省略をファイルシステムに伝える場合がある。この状況の場合、フラグ 616 の値は、クライアントの要求するファイル又はディレクトリに関連する読み出しアクセススタイル 406 が、そのファイル又はディレクトリが、例えばランダムアクセススタイルを使用してアクセスされていることを示すものであることを反映する場合がある。

【0064】

20

ファイルブロック番号 15 及び 16 並びに対応する先読みデータブロックの読み出しに加えて、ファイルシステムは、読み出しストリーム 435 に関連するリードセット 600 の内容の更新も行う。例えば、開始データブロック番号 704 に対応して、最終読み出しオフセット値 608 を変更する場合がある。同様に、読み出し要求 700 において指定されたデータブロック数 706 に等しくなるように、最終読み出しサイズ値 610 を更新する場合がある。更に、例えば、読み出しストリーム 435 に関連する所定の先読みアルゴリズムに従って、先読み値 612 ~ 616 を変更する場合がある。

【0065】

図 8 は、リードセット 600 が、受信したクライアント読み出し要求 700 に対して完全一致するか否かを判定するための一連のステップを示している。シーケンスはステップ 800 から開始され、ステップ 810 へ進み、そこで、クライアント読み出し要求がファイルシステム 260 によって受信される。ステップ 820 では、受信したクライアント要求において指定された開始データブロック 704 を、リードセット 600 に記憶された最終読み出しオフセット 608 と最終読み出しサイズ 610 の合計と比較する。それらの値が等しいと判定された場合、シーケンスはステップ 840 へ進む。この場合、要求はリードセット 600 に関連する読み出しストリームを論理的に延長するものであるから、ファイルシステムは、リードセット 600 が、受信したクライアント読み出し要求 700 に対して完全一致するものと判定する。ステップ 820 における判定が否であった場合、ステップ 830 において、リードセット 600 は受信した要求 700 に対して完全一致するものとは判定されない。後者の場合、完全一致が見付かるまで、又は、テストすべきリードセットがなくなるまで、クライアントの要求したファイル又はディレクトリに関連する他のリードセット 600 について、ステップ 820 ~ 840 が行われる。

30

40

【0066】

図 9 は、読み出しストリーム 435 を論理的に延長するものではないが、代わりに、「ほぼ連続的」なクライアント読み出し要求に対応するクライアント読み出し要求 900 の例を示している。すなわち、要求 900 は、連続読み出し要求のように読み出しストリーム 435 を正確に延長しないが、ほぼ連続であるともみなせるくらいに読み出しストリームの延長に似ている。例示的な実施形態によれば、ファイルシステム 260 は、読み出しストリーム 435 において読み出された最後のデータブロック（すなわち、ファイルブロック番号 14）に対して前進方向と後退方向の両方に延びる「曖昧」範囲 910 を識別する

50

場合がある。曖昧範囲 9 1 0 は、読み出し要求 9 0 0 において指定された読み出すべき複数のブロック 9 0 6 に基づいて導出されることが望ましい。例えば図示のように、曖昧範囲 9 1 0 は、クライアントが要求するデータブロック数の 3 倍の数にわたって、前進方向と後退方向の両方に広がる場合がある。具体的には、要求 9 0 0 では 2 つのデータブロックが要求されているので、曖昧範囲は後退方向に 6 データブロック（例えばファイルブロック番号 9 ~ 1 4）にわたって広がり、前進方向に 6 データブロック（例えばファイルブロック番号 1 5 ~ 2 0）にわたって広がる。

【 0 0 6 7 】

図示の実施形態は、読み出しストリーム 4 3 5 において読み出された最終データブロック（すなわち、ファイルブロック番号 1 4）を中心とした対称な曖昧範囲 9 1 0 になっているが、当然ながら、曖昧範囲は、最後に読み出されたデータブロックを中心として非対称な形であってもよいと考えられる。換言すれば、曖昧範囲 9 1 0 は通常、後退方向に第 1 の数のデータブロックだけ延び、前進方向に第 2 の数のデータブロックだけ延びる。例えば、クライアントが要求するデータブロック数 9 0 6 に他の乗算係数を使用して、前進方向と後退方向のそれぞれにおける曖昧範囲 9 1 0 の長さを導出してもよい。

【 0 0 6 8 】

図 9 に示すように、クライアント読み出し要求 9 0 0 は、i n o d e 番号 = 1 7、開始データブロック番号 = 1 6、及び、読み出すべきデータブロック数 = 2 を指定する。このクライアント要求に回答して、ファイルシステム 2 6 0 は、i n o d e 番号 1 7 に関連するファイル又はディレクトリにおいて f b n 1 6 及び 1 7 を探す。図 7 に関して説明したように、ファイルシステムはまず、コア内メモリバッファにおけるそれらのデータブロックの探索を試みる。次にファイルシステムはストレージサブシステム 2 5 0（例えば、R A I D 層及びディスクドライバ層）と協働し、コア内メモリバッファにおいて見付からなかったデータブロックをストレージディスク 1 6 0 から読み出す。

【 0 0 6 9 】

リードセット 6 0 0 は、読み出しストリーム 4 3 5 において処理された最後の読み出し要求によって、ファイルシステム 2 6 0 が、ファイルブロック番号 1 3 から始まる 2 つのデータブロック（例えば、ファイルブロック番号 1 3 と 1 4）を読み出すことになることを示している。読み出される f b n 番号 1 6 と 1 7 は、読み出しストリーム 4 3 5 を延長しない。実際、ファイルブロック番号 1 5 は、読み出しストリーム 4 3 5 において事実上「スキップ」される。従って、読み出し要求 9 0 0 は、読み出しストリーム 4 3 5 に関連するリードセット 6 0 0 に対して完全一致にはならない。ただし、要求 9 0 0 において指定された開始ブロック番号 1 6 は、図示の曖昧範囲 9 1 0 内にある。そのため、要求 9 0 0 は、読み出しストリーム 4 3 5 に関連するリードセット 6 0 0 に「曖昧」一致するものと判定され、次いでファイルシステム 2 6 0 は、そのクライアント読み出し要求を完全一致であったかのように処理する。

【 0 0 7 0 】

ファイルシステム 2 6 0 は、クライアント読み出し要求 9 0 0 が、読み出しストリーム 4 3 5 に関連するリードセット 6 0 0 に対して曖昧一致であるものと判定した場合、その要求をトリガとして読み出しストリームにおける先読み処理を開始するか否かを決定する。例えば、読み出されたデータブロック 1 6 及び 1 7 が、ネクスト先読み値 6 1 2 によって指定された f b n 番号 1 6 を上回る場合、ファイルシステムは、たとえクライアント要求が曖昧一致であり、完全一致ではない場合であっても、読み出しストリーム 4 3 5 に対する先読み処理を実施する。ファイルシステムは、読み出しストリーム中の次の論理データブロック（すなわち、ファイルブロック番号 1 8）から開始して、先読みサイズ値 6 1 4 によって指定された数のデータブロックをプリフェッチする。読み出される先読みデータブロックの数は先読みサイズ値 6 1 4 によって決定されることが好ましいが、先読みデータブロックの数は、代わりに、i n o d e 番号 1 7 に記憶されたデフォルト先読みサイズ 4 0 6 のような他の情報を使用して決定してもよい。

【 0 0 7 1 】

ファイルシステムは、f b n 番号 1 6 及び 1 7 に対応するデータブロック並びに対応する先読みデータブロックの読み出しの他に、読み出しストリーム 4 3 5 に関連するリードセット 6 0 0 の内容の更新も行う。例えば、開始データブロック番号 9 0 4 に対応して、最終読み出しオフセット値 6 0 8 を更新する場合がある。同様に、読み出し要求 9 0 0 において指定されたデータブロック数に等しくなるように、最終読み出しサイズ値 6 1 0 を更新する場合がある。更に、例えば、読み出しストリーム 4 3 5 に関連する所定の先読みアルゴリズムに従って、先読み値 6 1 2 ~ 6 1 6 を更新する場合がある。

【 0 0 7 2 】

図 1 0 は、リードセット 6 0 0 が受信したクライアント読み出し要求 9 0 0 に曖昧一致するか否かを判定するための一連のステップを示している。シーケンスはステップ 1 0 0 0 から開始され、ステップ 1 0 1 0 へ進み、そこで、クライアント要求がファイルシステム 2 6 0 によって受信される。ステップ 1 0 2 0 では、受信した読み出し要求における指定された開始データブロック 9 0 4 を、曖昧範囲 9 1 0 に含まれる所定範囲のブロック番号と比較する。この曖昧範囲は、読み出し要求 9 0 0 において指定されたデータブロック数 9 0 6 の倍数として導出される場合がある。開始データブロック値が所定の曖昧範囲内にあるものと判定された場合、シーケンスはステップ 1 0 4 0 へ進み、そこで、ファイルシステムは、受信したクライアント読み出し要求が曖昧一致であるものと判定する。そうでなければ、ステップ 1 0 3 0 において、受信したクライアント読み出し要求は、リードセット 6 0 0 に曖昧一致しないものと判定される。後者の場合、曖昧一致が見付かるか、又は、テストすべきリードセットが無くなるまで、クライアントの要求したファイル又はディレクトリに関連する他のリードセット 6 0 0 に対し、ステップ 1 0 2 0 ~ 1 0 4 0 が繰り返し実施される。

【 0 0 7 3 】

受信したクライアント読み出し要求が、クライアントの要求するデータに関連するリードセット 6 0 0 に対して、完全一致でも曖昧一致でもない場合、ファイルシステム 2 6 0 は、受信した要求を新たな読み出しストリームに関連付ける。従って、ファイルシステム 2 6 0 は、その新たな読み出しストリームに関連するメタデータの記憶に使用可能な未使用の、すなわち「空き」のリードセット 6 0 0 があるか否かをチェックする。この例の場合、リードセットのレベル値 6 0 4 が、リードセットが使用されていないことを示すマイナス 1 のような特殊な指示値に等しい場合、そのリードセットは空であるものと判断される。既に述べたように、レベル値 6 0 4 が特殊な指示値に設定されるのは、それが割り当てられたときだけである。リードセット 6 0 0 が読み出しストリームメタデータの記憶に最初に使用された後、レベル値 6 0 4 はそれ以後、所定の上限値及び所定の下限値によって制限される。

【 0 0 7 4 】

図 1 1 は、リードセット 6 0 0 が空であるか否か、従って、リードセット 6 0 0 が新たな読み出しストリームを開始する受信したクライアント要求に一致するか否かを判定するための一連のステップを示している。シーケンスはステップ 1 1 0 0 から始まり、ステップ 1 1 1 0 へ進み、そこで、クライアント読み出し要求がファイルシステム 2 6 0 によって受信される。ステップ 1 1 2 0 において、ファイルシステムは、リードセット 6 0 0 に記憶されたレベル値 6 0 4 が特殊な指示値（例えば、- 1）であるか否かを判定する。特殊な指示値であれば、シーケンスはステップ 1 1 4 0 へ進み、そこで、ファイルシステムは、リードセット 6 0 0 が空であるか否か、従って、リードセット 6 0 0 が新たな読み出しストリームに関連するメタデータの記憶に使用可能であるか否かを判定する。その場合、リードセット 6 0 0 に記憶されたメタデータは、そのカウント値 6 0 6 が 1 になるように増加させるといったように、クライアント読み出し要求に基づいて適切に更新される。ただし、ステップ 1 1 2 0 において、リードセットにおけるレベル値 6 0 4 が特殊な指示値ではない場合、ステップ 1 1 3 0 において、受信したクライアント読み出し要求は、リードセット 6 0 0 に対して空一致であるものと判定される。後者の場合、空一致が見付かるまで、又は、テストすべきリードセットが無くなるまで、クライアントの要求するファ

イル又はディレクトリに関連する他のリードセット600について、ステップ1120～1140が繰り返し実施される。

【0075】

E. リードセットの再使用

ファイルシステム260は、受信した読み出し要求に一致（すなわち、完全、曖昧、又は、空）するリードセットが見付からなかった場合に、要求されたファイルのリードセットのうちのいずれかが再使用できるか否かを判定するように構成される場合がある。具体的には、再使用されるリードセット600におけるメタデータを上書きすることにより、受信した読み出し要求に関連する新たなメタデータのセットを記憶するように、そのリードセットを再構成する場合がある。一実施形態によれば、各リードセット600に記憶されるレベル値604は、そのリードセットが再使用に適しているか否かを判定するためのエージング手段として使用される場合がある。具体的には、受信したクライアント要求がリードセット600に完全一致又は曖昧一致する場合は常に、レベル値604がインクリメントされる。また、受信したクライアント要求がリードセット600に対して空一致である場合は、レベル値604をレベル値の所定の初期レベル値に設定する。クライアント要求がリードセットに対して完全一致でも、曖昧一致でも、空一致でもない場合、ファイルシステムは、あるリードセットを再使用のために選択し、再使用のために選択されていないあらゆるリードセットにおけるレベル値604をデクリメントする場合がある。既存のリードセットの過剰な「スラッシング」を防止するために、リードセット再使用ポリシーを使用してもよい。再使用ポリシーによれば、リードセットの内容が時期尚早に上書きされることが防止される。

【0076】

図示の実施形態に関連し、ファイルシステム260は、リードセット600が次の条件のうちのいずれかを満たした場合に、そのリードセットを再使用すべきものとして選択する。(i)リードセットが、クライアントの要求するファイル又はディレクトリに関連する全てリードセットの中で最小レベルの値604を記憶している場合、又は、(ii)リードセットが、クライアントの要求するファイル又はディレクトリに関連する最も最近アクセスされたリードセットであり、そのリードセットが1に等しいカウント値606を記憶している場合。ファイルシステムは、リードセット600がそれらの条件のうちのいずれかを満たしていることを識別すると、そのリードセットを再使用し、それによってそのリードセットの内容を上書きし、受信したクライアント読み出し要求を開始する新たな読み出しストリームに関連するメタデータが記憶されるようにする。例えば、再使用されるリードセットのカウント値606はゼロで上書きされ、そのリードセットのレベル値604は、所定の上限レベル値（例えば、20）と所定の下限レベル値（例えば、ゼロ）の間の所定の初期値（例えば、10）で上書きされる場合がある。

【0077】

図12A～図12Bは、inode400の例、及び、そのinodeに割り当てられたリードセット600a、600b及び600cを示す略ブロック図である。説明のために、リードセット600a及び600cは当初、読み出しストリームAとCのそれぞれについて、レベル値604のようなメタデータ、及び、カウント値606を記憶しているものとする。また、リードセット600bは、まず、そのレベル値604がゼロに設定されてから「時間が経っている」ものとする。従って、リードセット600bに記憶されたメタデータは、少なくとも所定数のクライアント要求（例えば10個の要求）の結果、ファイルシステム260が他のリードセット600a又は600cを再使用した後、新たなクライアント読み出し要求を何も受信していない読み出しストリームに対応する。図示の実施形態の場合、最も最近アクセスされたリードセット、例えばリードセット600aは、inode400に最も近い位置にある。

【0078】

ステップ(i)において、ファイルシステムは、inode400に関連するファイル又はディレクトリに格納されたデータを求めるクライアント読み出し要求を受信し、ファイ

ルシステムは、受信した要求が、読み出しストリーム A に関連するリードセット 6 0 0 a に対して完全一致であるか、又は、曖昧一致であるかを判定する。それに応じて、ファイルシステムは、リードセット 6 0 0 a におけるカウント値 6 0 6 をインクリメントし、更に、リードセット 6 0 0 a におけるレベル値 6 0 4 のインクリメントを試みる。ただし、リードセット 6 0 0 a のレベル値 6 0 4 は、所定の上限値（例えば、2 0）に等しいので、このレベル値は、それ以上インクリメントされない。受信したクライアント要求がリードセット 6 0 0 a に関連する読み出しストリームに一致したので、リードセット 6 0 0 b 及び 6 0 0 c におけるレベル値 6 0 4 及びカウント値 6 0 6 は変更されることなく維持される。なお、図示の実施形態におけるファイルシステムは、レベル値 6 0 4 のインクリメントやデクリメントを 1 ずつ試行しているが、本発明の他の実施形態では、他のインクリメントステップサイズ又はデクリメントステップサイズを使用して、レベル値をインクリメント又はデクリメントする場合もある。

10

【 0 0 7 9 】

ステップ(ii)において、ファイルシステムは、i n o d e 4 0 0 に関連するファイル又はディレクトリにおいて、新たな読み出しストリーム B に関連するクライアント読み出し要求を受信する。この場合、受信する要求は、リードセット 6 0 0 a ~ c のうちのいずれかに完全一致するものでも、曖昧一致するものでもない。また、図示のリードセット 6 0 0 a ~ c の中には、マイナス 1 に等しいレベル値 6 0 4（すなわち、図示の実施形態における空のリードセットの特殊な指示値）を記憶しているものがないので、受信した要求は更に、リードセット 6 0 0 a ~ c のいずれかに空一致することもない。要求がリードセット 6 0 0 のいずれにも一致しないので、ファイルシステム 2 6 0 は、リードセット再使用ポリシーを使用して、内容を上書きして新たに識別された読み出しストリーム A 及び B に関するメタデータを記憶してもよいリードセットを探す。

20

【 0 0 8 0 】

ファイルシステムはまず、最も最近アクセスされたリードセット 6 0 0 a におけるカウント値 6 0 6 が 1 に等しいか否かを判定する。この場合、リードセット 6 0 6 a におけるカウント値は 1 ではなく 1 3 である。次にファイルシステムは、どのリードセット 6 0 0 a ~ c が最小レベル値 6 0 4 を記憶しているか、すなわち、他のリードセットに記憶されたレベル値以下のレベル値を有するリードセットを判定する。リードセット 6 0 0 b に記憶されたレベル値 6 0 4 が、リードセット 6 0 0 a ~ c の中で最小であるから、リードセット 6 0 0 b が再使用のために選択される。従って、リードセット 6 0 0 b におけるレベル値 6 0 4 は、所定の初期レベル値（例えば、1 0）に初期化され、そのカウント値 6 0 6 は 1 に等しくなるように上書きされる。更に、リードセット 6 0 0 b におけるメタデータが、新たな読み出しストリーム B に対応するように更新される。ファイルシステムは、残りのリードセット 6 0 0 a 及び 6 0 0 c に記憶されたレベル値 6 0 4 をデクリメントし、リードセット 6 0 0 b を i n o d e 4 0 0 に関連するリードセットの連結リストの先頭に移動させることにより、そのリードセット 6 0 0 b が現在、リスト上にある最も最近アクセスされたリードセットであることを示す。

30

【 0 0 8 1 】

ステップ(iii)において、ファイルシステムは、i n o d e 4 0 0 に関連するファイル又はディレクトリにおいて、新たな読み出しストリーム D に関連するクライアント読み出し要求を受信する。受信した要求は、リードセット 6 0 0 a ~ c のうちのいずれにも一致（完全、曖昧、又は、空）しないので、ファイルシステムは、その新たな読み出しストリーム D のために再使用可能なリードセットを探す。ファイルシステムはまず、最も最近アクセスされたリードセット 6 0 0 b におけるカウント値が 1 に等しいか否かを判定する。最も最近アクセスされたリードセット 6 0 0 b が 1 に等しいカウント値を記憶しているので、ファイルシステムは、そのリードセット 6 0 0 b を再使用のために再び選択する。従って、リードセット 6 0 0 b におけるレベル値 6 0 4 は、所定の初期レベル値（例えば、1 0）に設定され、そのカウント値 6 0 6 は 1 に等しくなるように設定される。この場合、リードセット 6 0 0 b におけるメタデータは、新たな読み出しストリーム D に対応する

40

50

ように更新される。次にファイルシステムは、残りのリードセット600a及び600cに記憶されたレベル値604をデクリメントする。リードセット600bは既にinode400に関連するリードセットの連結リストの先頭にあるので、リスト上のリードセットの順番は変更されずに維持される。

【0082】

ステップ(iv)において、ファイルシステムは、inode400に関連するファイル又はディレクトリに格納されたデータを求めるクライアント読み出し要求を受信し、ファイルシステムは、その受信した要求が、読み出しストリームCに関連するリードセット600cに完全一致又は曖昧一致するか否かを判定する。リードセット600cにおけるレベル値604は所定の初期レベル値(例えば、10)よりも小さいので、リードセット600cにおけるレベル値604は初期レベル値に等しくなるように設定される。更に、リードセット600cにおけるカウント値606がインクリメントされ、リードセット600cがリードセットリストの先頭に移動され、そのリードセット600cがinode400に関連する最も最近アクセスされたリードセットであることが示される。一致しないリードセット600a及び600bにおけるレベル値604及びカウント606は変更されずに維持される。

【0083】

最後にステップ(v)において、ファイルシステムは、inode400に関連するファイル又はディレクトリにおいて、新たな読み出しストリームEに関連するクライアント読み出し要求を受信する。この場合も、受信したクライアント要求は、リードセット600a~cのうちのいずれにも一致(完全、曖昧、又は、空)しないので、ファイルシステムは、その新たな読み出しストリームEに関するメタデータを記憶するために再使用可能なリードセットを探す。ファイルシステムはまず、最も最近アクセスされたリードセット600cにおけるカウント値606が1に等しいかを判定する。図示のように、リードセット600cにおけるカウント値は3であるから、次にファイルシステムは、リードセット600a~cのうちのいずれが最小レベル値604を記憶しているかを判定する。この場合、リードセット600cと600bが両方とも10に等しい最小レベル値604を記憶している。例えば、ファイルシステムは、リードセット600cが最も最近アクセスされたリードセットであるため、リードセット600cを再使用のために選択することができる。ただし、他の実施形態では、2以上のリードセットがそれぞれ最小レベル値を記憶している場合、他の基準を使用する場合がある。リードセット600cにおけるメタデータは、新たな読み出しストリームEに対応するように更新される。リードセット600cは既にリードセットのリスト上の先頭に位置しているので、ファイルシステムは、リードセット600cを移動させる必要はない。リードセット600cにおけるレベル値604は、所定の初期レベル値に等しくなるよう設定され、そのカウント値606は1に等しくなるよう設定される。残りのリードセット600a及び600bにおけるレベル値604は適宜デクリメントされる。

【0084】

図13A~図13Bは、ファイルシステム260がクライアント読み出し要求を受信したときに実施される一連のステップを含むフロー図である。ここで、受信したクライアント読み出し要求は、少なくとも1つのリードセットに関連するファイル又はディレクトリに含まれるデータセットを指定するものと仮定する。ただし、クライアントの要求するデータがゼロリードセットに関連するファイル又はディレクトリに格納されている場合、例えば、ファイルサイズが小さすぎて読み出しストリームを形成することができない場合、ファイルシステム260は、受信したクライアント読み出し要求を、例えば従来の先読み技術を使用して、従来のやり方で処理する。

【0085】

シーケンスはステップ1300から開始され、ステップ1305へ進み、そこで、ファイルシステムはクライアント読み出し要求を受信する。ただし、受信したクライアント要求が既にデータブロック単位にフォーマットされている場合、ファイルシステムは、例え

ば開始データブロックと、読み出すべき連続したデータブロックとを指定することにより、その要求をデータブロック単位に再フォーマットする場合がある。また、受信した要求がファイルシステム 260 に対し、新たに（すなわち、最近）作成されたファイル又はディレクトリに記憶されたデータの読み出しを命じるものである場合、ファイルシステムは、そのファイル又はディレクトリに対して新たな `inode` を割り当てなければならない場合がある。さらに、ファイルシステムは、新たに割り当てられた `inode` に対し、ゼロ以上のリードセットを割り当てる場合がある。このリードセットの数は、クライアントが要求するファイル又はディレクトリのサイズに応じて決まる。例えば、新たな `inode` は `inode` プール 152 から得られ、それに対応するリードセットのセットはリードセットプール 154 から得られる場合がある。

10

【0086】

ステップ 1310 において、ファイルシステムは、受信した要求が、クライアントが要求したファイル又はディレクトリに関連するリードセット 600 に対して完全一致、曖昧一致、又は、空一致するか否かを判定する。一致するリードセットが見付かると、ステップ 1315 においてファイルシステム 260 は、一致するリードセットに記憶されたカウント値をインクリメントする。次に、ステップ 1320 においてファイルシステムは、その一致するリードセットに記憶されたレベル値 604 が所定の上限値（すなわち、最大値）に等しいか否かを判定する。上限値に等しい場合、シーケンスはステップ 1355 へ進み、一致するリードセットに記憶されたレベル値 604 を変更せずにそのまま維持する。そうでなければ、ステップ 1325 においてファイルシステムは、そのレベル値 604 が所定の初期レベル値（例えば、10）未満であるか否かを判定する。未満であれば、ステップ 1330 においてレベル値 604 は所定の初期値に設定され、シーケンスはステップ 1360 へ進む。ステップ 1325 においてレベル値 604 が初期レベル値以上であると判定された場合、次にステップ 1335 において、一致するリードセットにおけるレベル値 604 が、例えば 1 だけインクリメントされる。そしてシーケンスは、以下で説明するようなステップ 1360 へ進む。

20

【0087】

ステップ 1310 において、受信したクライアント要求がいずれのリードセットにも一致しなかった場合、ステップ 1340 ~ 1355 に記載したように、ファイルシステム 260 は、クライアントが要求したファイル又はディレクトリに関連するリードセットの中から再使用可能なものを探す。具体的には、ステップ 1340 において、ファイルシステムは、リードセットの連結リストの例えば先頭に位置している最も最近アクセスされたリードセットが、1 に等しいカウント値を記憶しているか否かを判定する。記憶している場合、その最も最近アクセスされたリードセットが、再使用のために選択される。一方、最も最近アクセスされたリードセット 600 におけるカウント値 606 が 1 に等しくなかった場合は、ステップ 1345 において、最小レベル値 604 を記憶しているリードセットが、再使用のために選択される。最小レベル値 604 が 2 以上のリードセットに記憶され、それらのリードセットのうちの 1 つが最も最近アクセスされたリードセットであった場合、その最も最近アクセスされたリードセットが、再使用のために選択される。そうでなければ、ファイルシステムは、最小レベル値を記憶している複数のリードセットの中から、任意の 1 つを単純に選択する。

30

40

【0088】

次に、ステップ 1350 において、再使用のために選択されたリードセット 600 に記憶されたレベル値 604 を所定の初期レベル値に等しくなるよう設定し、そのリードセットのカウント値 606 を 1 に等しくなるよう設定する。ステップ 1355 において、ファイルシステムは、再使用のために選択されなかったリードセット、すなわち、受信したクライアント要求に一致しなかったリードセットに記憶された全ての非ゼロのレベル値 604 をデクリメントする。ステップ 1360 において、ファイルシステムは、必要であれば、最も最近アクセスされたリードセット（すなわち、一致する、すなわち再使用されるリードセット）を、クライアントが要求したファイル又はディレクトリに関連するリードセ

50

ットのリストの先頭に移動させる。

【 0 0 8 9 】

ステップ 1 3 6 5 において、ファイルシステムは、一致する、すなわち再使用されるリードセットにおける最終読み出しオフセット値 6 0 8、最終読み出しサイズ値 6 1 0、ネクスト先読み値 6 1 2 等のリードヘッド情報を更新する。ステップ 1 3 7 0 において、ファイルシステムは、クライアントが要求するデータを有するデータブロック 3 2 0 を読み出す。また、受信したクライアント読み出し要求が既存の読み出しストリームに対して完全一致又は曖昧一致し、その要求が、例えば一致するリードセット 6 0 0 に記憶された先読みサイズ 6 1 4 によって指定されるような所定のファイルオフセット又はメモリアドレスを越えて既存の読み出しストリームを延長するものである場合、ファイルシステムは、1 以上の先読みデータブロックを更に読み出す場合がある。読み出される先読みデータブロックの数は、一致するリードセット 6 0 0 に記憶される先読みサイズ 6 1 4 によって指定することができる。ファイルシステムはまず、メモリ 1 5 0 に記憶された 1 以上のコア内メモリバッファにおいてクライアントが要求するデータブロック及び先読みデータブロックを探す。ただし、それらのデータブロックがメモリバッファ中に見付からない場合、ファイルシステムはストレージサブシステム 2 5 0 と協働し、それらのデータブロックをストレージディスク 1 6 0 から読み出し、読み出したブロックを例えばバッファプール 1 5 6 から取得した 1 以上のコア内メモリバッファにコピーする。そして、クライアントの要求したデータは、ストレージオペレーティングシステム 2 0 0 によってパケット化され、要求元クライアント 1 9 0 へ返される。シーケンスは 1 3 7 5 にて終了する。

【 0 0 9 0 】

D. むすび

上記が、本発明の例示的实施形態に関する詳細な説明である。本発明の思想及び範囲から外れることなく、種々の変更をや追加を行うことが可能である。例えば、ファイルシステム 2 6 0 は、クライアント読み出し要求を受信したときに、クライアントが要求するファイル又はディレクトリに関連するリードセットの連結リスト（又は、他の検索可能なデータ構造）の中を 1 以上の回数だけ「走査」し、それらのリードセットの中に、受信した要求に完全一致、曖昧一致、又は、空一致するものがあるか否かを判定するように構成してもよい。例えば、ファイルシステムは、リストを一回だけ走査することにより、完全一致、曖昧一致、又は、空一致するリードセットが見付かるまで、リスト内の各リードセットを順番にテストしてもよい。この実施形態の場合、ファイルシステムが、リードセットが完全一致、曖昧一致、又は、空一致するか否かに関するテストを行う順序は、様々であってよい。代替実施形態では、リードセットのリスト全体にわたって個別の走査を実施し、完全一致するリードセットと、曖昧一致するリードセットと、空一致するリードセットとを個別に探してもよい。

【 0 0 9 1 】

例示した `inode 400` に関連するリードセットのリストは、最も最近アクセスされたリードセットがリストの「先頭」にくるように構成されているが、ファイルシステムは、代わりに、最も最近アクセスされたリードセットを別の方法で探すように構成してもよい。例えば、リードセットがリスト上で最も最近アクセスされたリードセットである場合に第 1 の値を有し、それ以外の場合に第 2 の値を有するフラグ 6 1 6 を各リードセット 6 0 0 に含めてもよい。また、ファイルシステム 2 6 0 は、あらゆるクライアント読み出し要求が処理された後に、レベル値 6 0 4 をインクリメント又はデクリメントすることにより、リードセットを老化させているが、当然ながら、レベル値 6 0 4 は所定の時間間隔の後に適宜老化させてもよい。このシナリオの場合、レベル値 6 0 4 は、処理されたクライアント要求の数に対する相対年齢ではなく、時間に対するリードセットの相対年齢を示すものであってもよい。

【 0 0 9 2 】

上記のように、例示したファイルやディレクトリは、ゼロ以上の読み出しストリームを確立することが可能な任意のデータセットとして広く定義される。従って、この文脈にお

けるファイルは、ブロックベースのクライアント 190b に対して単一の論理ユニット番号 (LUN) としてエクスポートすることが可能な所定のデータブロックのセットに対応する「仮想ディスク」(vdisk) として実施される場合がある。ただし、仮想ディスクにおけるデータブロックは、マルチプロトコル・ストレージ・アプライアンス 100 内部のファイルベースのセマンティックを使用してアクセスされる場合もある。従って、ブロックベースのクライアントは、FCP や iSCSI プロトコルのような従来のブロックベースのプロトコルに従って自分の要求をフォーマットすることができ、それらの要求は、ストレージオペレーティングシステム 200 によって実施される仮想化システムにより、ファイルベースのセマンティックを使用して処理される。

【0093】

図示の実施形態は、例えば読み出しストリーム 430 及び 435 のように、「前方」方向 (すなわち、データブロック番号が増大する方向) に延びる読み出しストリームを示しているが、当業者には明らかなように、本明細書に記載した本発明の概念は、「後退」方向 (すなわち、データブロック番号が減少する方向) に延びる読み出しストリームにも等しく適用可能である。その目的のために、各リードセットにおけるフラグ 616 は、そのリードセットに関連する読み出しストリームが前方方向に延びていることを示す第 1 の値に等しくなるよう設定されるか、又は、そのリードセットに関連するリードストリームが後退方向に延びていることを示す第 2 の値に等しくなるよう設定される。それに従ってファイルシステムは、読み出しストリームに対する先読みデータブロックを、読み出しストリームが延びている方向、例えば、適当なフラグ 616 によって指定されるような方向に読み出す。

【0094】

本明細書の説明は、マルチプロトコル・ストレージ・アプライアンスを参照して書かれているが、その原理は、ブロックベースのストレージシステム (ストレージ・エリア・ネットワーク等)、ファイルベースのストレージシステム (ネットワーク・アタッチド・ストレージ・システム等)、両方のタイプのストレージシステムの組み合わせ (マルチプロトコル・ストレージ・アプライアンス等)、又は他の形態のコンピュータシステムとして構成されるものを含めて、あらゆるタイプのコンピュータに等しく適用される。また、当然ながら、本発明の教示は、コンピュータ上で実行されるプログラム命令を含むコンピュータ読取可能媒体を含めて、ソフトウェア、ハードウェア、ファームウェア、又は、それらの組み合わせのいずれの形で実施できるものと考えられる。更に、当業者には分かるように、本明細書に記載する教示は、特定のオペレーティングシステム (OS) 実施形態に限定されることなく、種々の OS プラットフォームによって実施することができる。従って、本明細書の説明は、単なる例として解釈すべきものであり、本発明の範囲を制限するものではない。

【図面の簡単な説明】

【0095】

【図 1】本発明に従って使用可能なマルチプロトコル・ストレージ・アプライアンスの例を示す略ブロック図である。

【図 2】本発明とともに有利に使用されるストレージオペレーティングシステムの例を示す略ブロック図である。

【図 3】例示的なマルチプロトコル・ストレージ・アプライアンスにおいてファイル又はディレクトリに関連付けられるバッファツリーの例を示す略ブロック図である。

【図 4】inode に関連するファイル又はディレクトリに確立される読み出しストリームのための先読みメタデータの記憶に使用されるリードセットデータ構造のセットの例を示す略ブロック図である。

【図 5】ファイルやディレクトリのサイズに基づいてファイルやディレクトリに割り当てられるリードセットの数を決めるために使用される表の例を示す略ブロック図である。

【図 6】本発明に従って有利に使用されるリードセットの例を示す略ブロック図である。

【図 7】既存の読み出しストリームに関連するリードセットに対して「完全一致」するも

10

20

30

40

50

のと判定された受信したクライアント要求を示す略ブロック図である。

【図 8】受信したクライアント要求が既存の読み出しストリームに関連するリードセットに対して完全一致するか否かを判定するために実施される一連のステップを示すフロー図である。

【図 9】既存の読み出しストリームに関連するリードセットに対して「曖昧一致」であるものと判定された受信したクライアント要求を示す略ブロック図である。

【図 10】受信したクライアント要求が既存の読み出しストリームに関連するリードセットに対して曖昧一致するか否かを判定するために実施される一連のステップを示すフロー図である。

【図 11】既存の読み出しストリームに関連するリードセットに対して「空一致」であるものと判定された受信したクライアント要求を示す略ブロック図である。

10

【図 12 A】クライアント読み出し要求を処理する際に使用される例示的な老化手段、及び、再使用ポリシーを示す略ブロック図である。

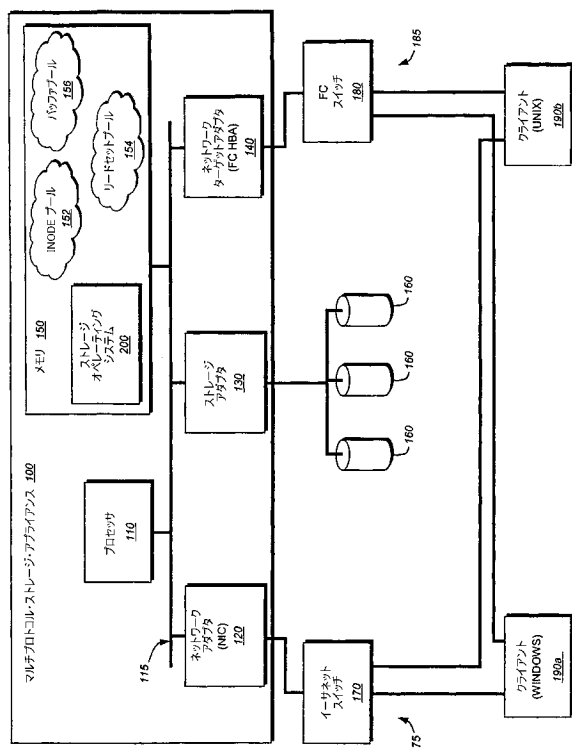
【図 12 B】クライアント読み出し要求を処理する際に使用される例示的な老化手段、及び、再使用ポリシーを示す略ブロック図である。

【図 13 A】図示のマルチプロトコル・ストレージ・アプライアンスにおいて受信したクライアント読み出し要求を処理するために実施される一連のステップを示すフロー図である。

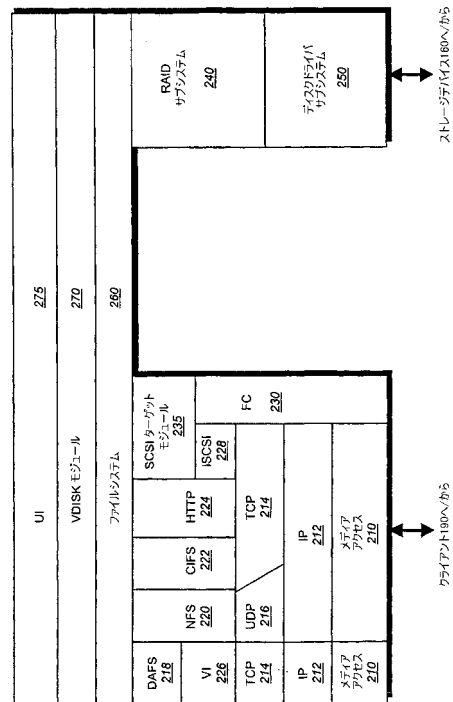
【図 13 B】図示のマルチプロトコル・ストレージ・アプライアンスにおいて受信したクライアント読み出し要求を処理するために実施される一連のステップを示すフロー図である。

20

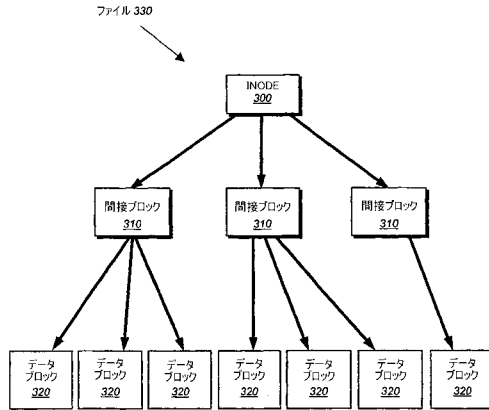
【図 1】



【図 2】



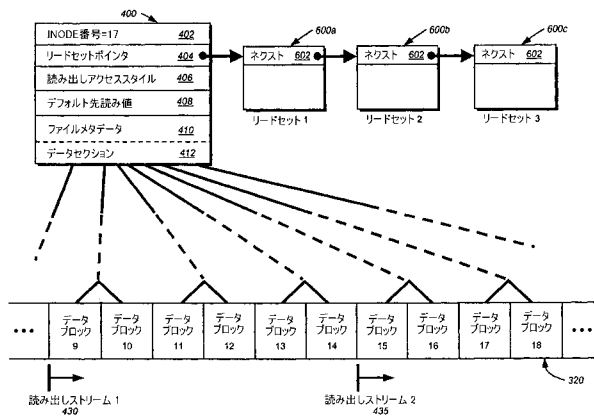
【図 3】



【図 5】

ファイルサイズ	リードセット数
< 64 kB	0
64 kB-512 kB	1
512 kB-50 MB	2
50 MB-1 GB	5
1 GB-10 GB	10
> 10 GB	15

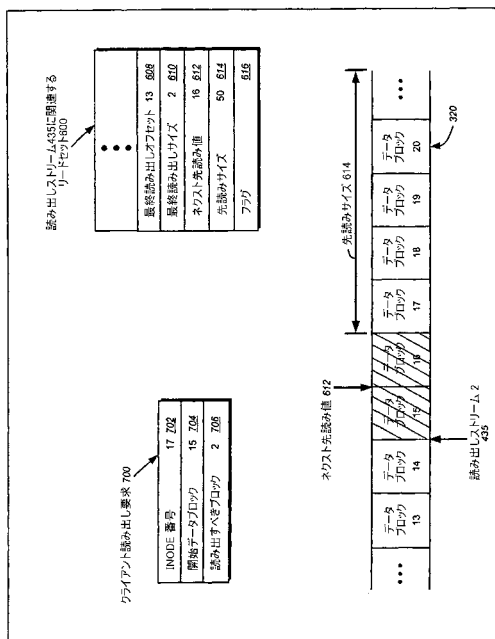
【図 4】



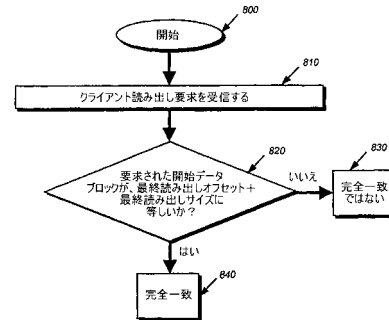
【図 6】

リードセット 600	
ネクストポイント	602
レベル	604
カウンタ	606
最終読み出しオフセット	608
最終読み出しサイズ	610
ネクスト先読み値	612
先読みサイズ	614
フラグ	616

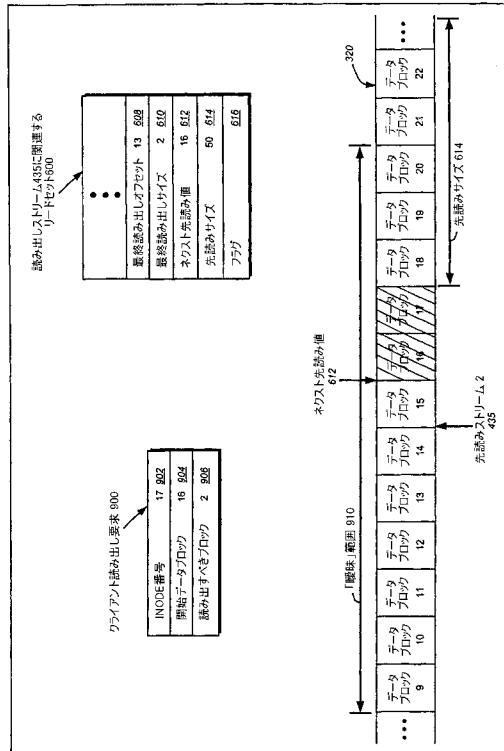
【図 7】



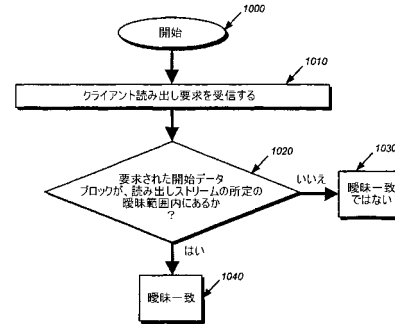
【図 8】



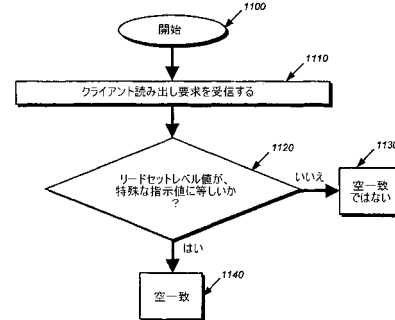
【図 9】



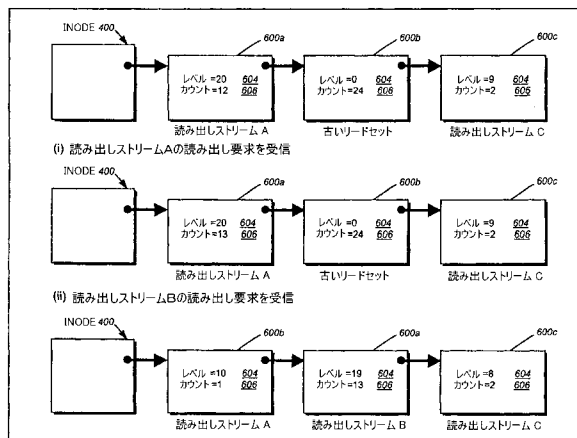
【図 10】



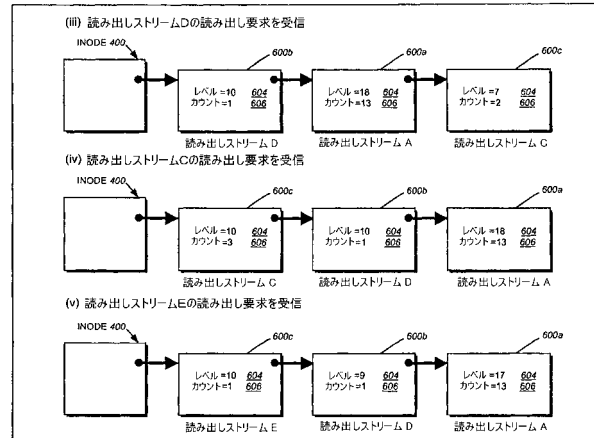
【図 11】



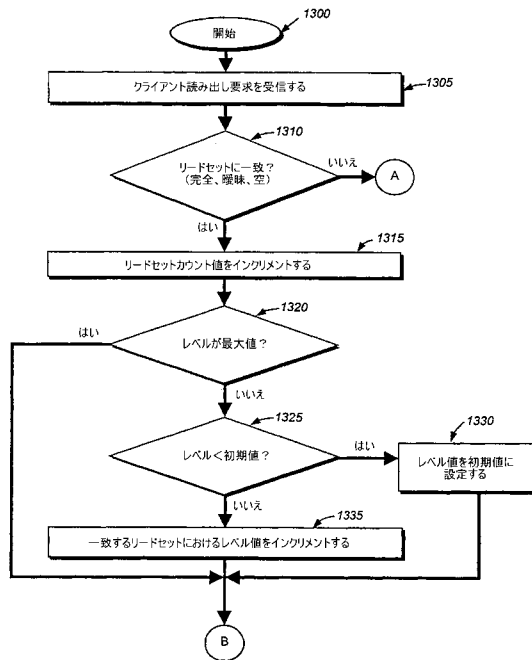
【図 12 A】



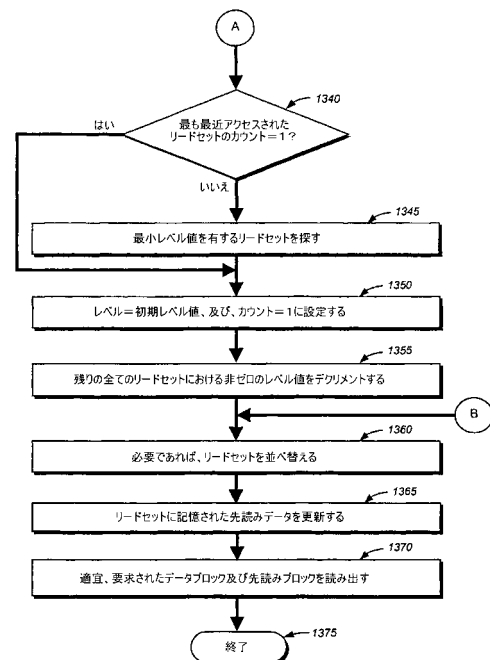
【図 12 B】



【図 13 A】



【図 13 B】



フロントページの続き

審査官 工藤 嘉晃

(56)参考文献 特開平 1 1 - 2 8 2 6 3 1 (J P , A)
特開 2 0 0 3 - 2 3 3 9 7 6 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

G06F 12/00

G06F 3/06

G06F 13/10

JSTPlus(JDreamII)