(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4620457号
(P4620457)
(45)発行日　平成23年1月26日(2011.1.26)　(24)登録日　平成22年11月5日(2010.11.5)

(51)Int.Cl.　F I
G06F 12/00 (2006.01)　G06F 12/00 520J
G06F 12/00 531M

請求項の数 11　(全 15 頁)

(21)出願番号　特願2004-512028 (P2004-512028)
(86)(22)出願日　平成15年6月3日 (2003.6.3)
(65)公表番号　特表2005-529410 (P2005-529410A)
(43)公表日　平成17年9月29日 (2005.9.29)
(86)国際出願番号　PCT/US2003/017497
(87)国際公開番号　WO2003/105026
(87)国際公開日　平成15年12月18日 (2003.12.18)
審査請求日　平成17年12月15日 (2005.12.15)
(31)優先権主張番号　10/165,188
(32)優先日　平成14年6月7日 (2002.6.7)
(33)優先権主張国　米国(US)

前置審査

(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　９４０８９，　サニーヴェール，　イースト　ジャバ　ドライブ　４９５
(74)代理人　100101454
弁理士　山田　卓二
(74)代理人　100081422
弁理士　田中　光雄
(72)発明者　デイビッド・ヒッツ
アメリカ合衆国９４０２２カリフォルニア州ロス・アルトス、シルビアン・ウェイ１３７番

最終頁に続く

(54)【発明の名称】複数の同時にアクティブなファイルシステム

(57)【特許請求の範囲】
【請求項１】
複数のアクティブな、すなわち書き込み可能なファイルシステムを含むデータ記憶を行う方法であって、
第１のアクティブなファイルシステムに基づいて第２のアクティブなファイルシステムを作成し、当初は上記第１のアクティブなファイルシステムおよび上記第２のアクティブなファイルシステムがデータを共有し、
上記第１のアクティブなファイルシステムに変更が加えられたとき、変更されたデータが上記第１のアクティブなファイルシステム内の上記第２のアクティブなファイルシステムとは共有されない場所に記録され、
上記第２のアクティブなファイルシステムに変更が加えられたとき、変更されたデータが上記第２のアクティブなファイルシステム内の上記第１のアクティブなファイルシステムとは共有されない場所に記録され、
上記アクティブなファイルシステムの各々に加えられた変更が、上記変更されたアクティブなファイルシステムと上記データを共有するアクティブなファイルシステムに反映されず、
更に各々のアクティブなファイルシステムが第１のポインタと第２のポインタを含む組織データを有し、変更されたアクティブなファイルシステムにおける変更されていないデータへの第１のポインタは複数のアクティブなファイルシステムを記憶する１以上の記憶装置内で変更されず、既存のアクティブなファイルシステムにおける新しいアクティブな

ファイルシステムへの第２のポインタが変更される
方法。
【請求項２】
複数のスナップショットが上記複数のアクティブなファイルシステムの１つから作成され、上記スナップショットの各々が、過去の１整合点でのそのアクティブなファイルシステムのイメージを形成する請求項１に記載の方法。
【請求項３】
各スナップショットが、上記複数のアクティブなファイルシステムのためのアクティブなファイルシステムデータから分離した、ファイルシステムデータのための完全な階層を含む請求項２に記載の方法。
【請求項４】
上記スナップショットの少なくとも１つが、新しいアクティブなファイルシステムに変換される請求項２に記載の方法。
【請求項５】
上記スナップショットの１つが、上記スナップショットの１つを書き込み可能にすることにより変換される請求項４に記載の方法。
【請求項６】
いずれかの上記アクティブなファイルシステムから上記新しいアクティブなファイルシステムへのスナップショットポインタが切断される請求項５に記載の方法。
【請求項７】
複数のアクティブなファイルシステムを作成する方法であって、
第１のアクティブなファイルシステムのスナップショットを作成し、各々のアクティブなファイルシステムがスナップショットポインタと他のポインタを含む組織データを有し、上記スナップショットが当初は上記第１のアクティブなファイルシステムとデータを共有するステップ；および
上記スナップショットを書き込み可能にすることにより、上記スナップショットを第２のアクティブなファイルシステムに変換し、第１のアクティブなファイルシステムから第２のアクティブなファイルシステムへの第１のスナップショットポインタを切断するステップを含み、
上記第１のアクティブなファイルシステムに変更が加えられたとき、変更されたデータが、上記第１のアクティブなファイルシステム内の上記第２のアクティブなファイルシステムとは共有されない場所に記録され、
上記第２のアクティブなファイルシステムに変更が加えられたとき、変更されたデータが、上記第２のアクティブなファイルシステム内の上記第１のアクティブなファイルシステムとは共有されない場所に記録され、
ここに、上記第１のアクティブなファイルシステムに加えられた変更が上記第２のアクティブなファイルシステムに反映されず、上記第２のアクティブなファイルシステムに加えられた変更が上記第１のアクティブなファイルシステムに反映されず、変更された各々のアクティブなファイルシステムにおける変更されていないデータへの上記他のポインタが複数のアクティブなファイルシステムを記憶する１以上の記憶装置内で変更されない
方法。
【請求項８】
請求項７に記載の方法であって、さらに
上記第１のアクティブなファイルシステムの、当初は上記第１のアクティブなファイルシステムとデータを共有する新しいスナップショットを作成するステップ；および
上記新しいスナップショットを書き込み可能にすることにより、上記新しいスナップショットを第３のアクティブなファイルシステムに変換し、第１のアクティブなファイルシステムから第３のアクティブなファイルシステムへの第２のスナップショットポインタを切断するステップを含み、
ここに、上記第１のアクティブなファイルシステムまたは上記第２のアクティブなファ

イルシステムに加えられた変更が上記第３のアクティブなファイルシステムに反映されない
方法。
【請求項９】
上記第１のアクティブなファイルシステムまたは上記第２のアクティブなファイルシステムに変更が加えられたとき、変更されたデータが、上記第３のアクティブなファイルシステムとは共有されない場所に記録される請求項８に記載の方法。
【請求項１０】
コンピュータシステム上で実行された時、上記コンピュータシステムに請求項１から請求項９のいずれか１つのステップをもたらすプログラムコード手段を含むコンピュータプログラム。
【請求項１１】
少なくとも１つの記憶装置；
情報を送受信する少なくとも１つのコンピュータ・デバイスまたはネットワークへのインタフェース；および
上記記憶装置内の上記情報の記憶および取り出しを制御し、プログラム制御下で動作して請求項１から請求項９のいずれか１つのステップを実行するコントローラ
を含む記憶システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は複数の同時に書き込み可能なファイルシステムに関する。
【背景技術】
【０００２】
ファイルシステムは情報を記憶する構造を提供する。この情報は、例えば、ディスクドライブ、ＣＤ－ＲＯＭドライブなどの記憶装置上のアプリケーションプログラム、ファイルシステム情報、他のデータなど（今後、まとめて単にデータという）である。多くのファイルシステムについての１つの問題は、もし上記ファイルシステムがどういうわけか損傷すると、多量のデータが失われうることである。
【０００３】
そのようなデータの損失を防ぐため、ファイルシステムのバックアップがしばしば作成される。ファイルシステムのバックアップを生成する１つのとても効率的な方法は、上記ファイルシステムのスナップショットを作成することである。スナップショットは整合点での上記ファイルシステムのイメージであって、整合点とは上記ファイルシステムが、自己整合的である点である。もしファイルシステムの中に記憶されたデータが正当なファイルシステムイメージを構成するならば、そのファイルシステムは自己整合的である。
【０００４】
あるファイルシステム、例えばＷＡＦＬ（Ｗｒｉｔｅ　Ａｎｙｗｈｅｒｅ　Ｆｉｌｅ　ｓｙｓｔｅｍ　Ｌａｙｏｕｔ）ファイルシステムでは、ファイルシステムのスナップショットは、上記ファイルシステム内のデータの組織に関する情報をコピーすることにより作製可能である。次に、上記データ自体が上記記憶装置に保存されている限り、上記データは上記スナップショット経由でアクセス可能である。１つの仕組みが、例えばブロックマップ経由で、このデータを保存するため、これらのファイルシステム内で提供される。
【発明の開示】
【発明が解決しようとする課題】
【０００５】
従来は、スナップショットは読み取り専用である。読み取り専用スナップショットは、以前のバージョンのデータを再度読み出して、ファイルシステムへの損傷を修復する。これらの機能は非常に有用になりうる。しかし、これらの種類のスナップショットは、有用でありうる他の機能を提供しない。

【課題を解決するための手段】
【０００６】
もし、スナップショットを変更したい人がスナップショットを変更するように、スナップショットに書き込めたら、好都合である。これはいくつかの効果があるだろう。
【０００７】
・スナップショットに記憶されていた誤ったエントリの修正が可能になる。
【０００８】
・ファイルシステムから消去されることを所望されたデータが消去可能になる。
【０００９】
・ファイルシステム（またはファイルシステムにより維持されるデータについて）の「試験的な」バージョンに対する変更が可能になる。ファイルシステムの「試験的な」バージョンとは、上記ファイルシステムの壊滅的な誤りがその「実際の」アクティブなバージョンにおいてデータ損失を生じない上記ファイルシステムのバージョンである。
【００１０】
・上記ファイルシステムの動作、または、上記ファイルシステムの保護の下で動作する、プログラムまたはデータベース動作に対して誤った更新を無効にすることが可能になる。
【００１１】
書き込み可能なスナップショットは実はもう１つのアクティブなファイルシステムである。このアクティブなファイルシステムがもう１つのアクティブなファイルシステムからのデータに基づいているので、元のアクティブなファイルシステムに損害を与える危険を冒さずに、上記アクティブなファイルシステムに関する変更および修正を上記書き込み可能なスナップショットへ加えることができる。加えて、スナップショットは、単に組織情報をコピーして既存のデータを保存することにより作成できるので、書き込み可能なスナップショット（すなわち、新しいアクティブなファイルシステム）は、容易にかつわずかなシステム資源の利用で作成できる。
【００１２】
これらおよび他の効果は、ここに記載した、複数の上記アクティブなファイルシステムが維持される本発明の実施の形態において提供される。複数の上記アクティブなファイルシステムの各々が、その中のもう１つの上記アクティブなファイルシステムと共有されるデータに最初にアクセスし、また、上記アクティブなファイルシステムの各々に加えられる変更は、他のアクティブなファイルシステムに反映されない。
【００１３】
上記の好ましい実施の形態では、第２のアクティブなファイルシステムが第１のアクティブなファイルシステムに基づいて作成されたとき、上記第１のアクティブなファイルシステムおよび上記第２のアクティブなファイルシステムは、最初はデータを共有する。上記第１のアクティブなファイルシステムに変更が加えられたとき、変更されたデータが、上記第１のアクティブなファイルシステム内の上記第２のアクティブなファイルシステムとは共有されない場所に記録される。上記第２のアクティブなファイルシステムに変更が加えられたとき、変更されたデータが、上記第２のアクティブなファイルシステム内の上記第１のアクティブなファイルシステムとは共有されない場所に記録される。
【００１４】
好ましくは、さらに別のスナップショットが複数の上記アクティブなファイルシステムのスナップショットから作られ、各スナップショットは、過去の整合点におけるそのアクティブなファイルシステムのイメージを形成する。各スナップショットは、複数の上記アクティブなファイルシステムのアクティブなファイルシステムデータから分離し、ファイルシステムデータの完全な階層を含む。これらのスナップショットの１つは、次に、上記スナップショットを書き込み可能にすることにより、および、いずれかの上記アクティブなファイルシステムから新しいアクティブなファイルシステムへのスナップショットポインタを切断することにより、上記新しいアクティブなファイルシステムに変換可能である

。
【００１５】
本発明は、また、これらの動作を実行する命令を含むメモリおよび上記動作を実行する記憶システムを含む。
【発明を実施するための最良の形態】
【００１６】
以下、添付の図を参照して発明の実施の形態を説明する。
【００１７】
以下の語句は、以下に説明するように本発明の複数の面に関係する。これらの語句の一般的意味の説明は、限定することを意図せず、説明に役立つことのみを意図している。
【００１８】
・データとは、一般的にどれかの情報。記憶装置またはファイルシステムに関連して、アプリケーションプログラムまたはデータ、マルチメディアデータ、上記記憶装置またはファイルシステムの組織データなどを含むがそれらに限定されない上記記憶装置またはファイルシステム内に記憶されたどれかの情報。
【００１９】
・組織データとは、一般的に、ファイルシステム内の他のデータのレイアウトを指定するデータ。ＷＡＦＬ（Ｗｒｉｔｅ　Ａｎｙｗｈｅｒｅ　Ｆｉｌｅ　Ｓｙｓｔｅｍ）設計では、上記組織データは直接にまたは間接に（すなわち、他のｉノードを経由して）上記ファイルシステム内のすべてのファイルのためのデータのブロックを指すルートｉノードを含む。ＷＡＦＬ設計では、上記組織データを含むすべてのデータ（それ故、ルートｉノードおよび他のｉノード）は複数のブロックに記憶される。
【００２０】
・ｉノードとは、一般的に、情報ノード。ＷＡＦＬ設計では、上記ファイルシステム内での他のブロックについてのデータを含む情報ノード。
【００２１】
・自己整合的である（ファイルシステム内のコンテキスト（ｃｏｎｔｅｘｔ)の中で）とは、一般的に、ファイルシステムの組織に関するデータを含み、上記ファイルシステム内に記憶された上記データが有効なファイルシステムイメージを構成するとき、上記ファイルシステムは自己整合的であるという。
【００２２】
整合点とは、一般的に、整合点は、（ａ）ファイルシステムが自己整合的である時間、または、（ｂ）整合点の時間におけるファイルシステム内の１組のデータをいう。
【００２３】
スナップショットとは、一般的に、整合点の時間において上記ファイルシステムにより維持されるデータの書かれた記録である。好ましい実施の形態では、各スナップショットは（ａ）アクティブなファイルシステムと同じフォーマットに維持され、かつ、（ｂ）ファイルシステム名前空間を使用して参照可能であるけれども、これらの条件の一方を必要とすることに対して限定はない。
【００２４】
アクティブなファイルシステムとは、一般的に、アクティブなファイルシステムはアクセス可能で変更可能な１組のデータである。
【００２５】
ファイルシステム階層とは、一般的に、ファイルシステム階層とは、（ａ）名前空間へのデータの組織、または（ｂ）記憶装置に維持されている、データまたはメタデータの情報を記録し、その情報にアクセスするために使用される１組のデータブロックおよびそれらの相互結合をいう。
【００２６】
スナップショットおよびアクティブなファイルシステム
【００２７】

図１は、本発明によりアクティブなファイルシステムに変換可能なスナップショットの作成を示す。

【００２８】

図１に示すファイルシステム１００は１以上の記憶装置、例えばハードディスクドライブ、ＣＤ－ＲＯＭ、または他の装置に存在する。好ましい実施の形態では、ファイルシステム１００はＷＡＦＬシステムであるが、これは必須ではない。

【００２９】

ファイルシステム１００はルートｉノード１１０およびデータ１２０を他のデータとともに含む。ファイルシステム１００内のすべてのｉノードおよびデータは好ましくはブロック内に記憶されるが、これは必須ではない。

【００３０】

ルートｉノード１１０は、ファイルシステム１００のための組織データの複数部分を記憶する。特に、ルートｉノード１１０は、データを指し、他のｉノードおよびデータを指し、このデータは、次に、ファイルシステム１００内に記憶されたすべての情報に関するデータを指す。それ故、ファイルシステム１００内に記憶されたすべてのデータは、ルートｉノード１１０で開始することにより到達可能である。

【００３１】

スナップショット１３０はファイルシステム１００から形成された。図１では、スナップショット１３０の要素は、ファイルシステム１００からそれらの要素を区別するのを助けるために破線を使用して示される。本発明の好ましい実施の形態によれば、上記スナップショットは、ファイルシステム１００に関する整合点において、単にルートｉノード１１０をスナップショットのルートｉノード１４０にコピーすることにより、形成可能である。ある実施の形態では、追加の組織データがコピーされねばならないかもしれない。次に、ルートｉノード１１０により指されるすべてのデータおよびｉノード（およびすべての他のコピーされた組織データ）が保存される限りは、スナップショットのルートｉノード１４０はファイルシステム１００の有効なコピーを指す。

【００３２】

スナップショットのルートｉノード１４０が作成された後で、スナップショット１３０およびファイルシステム１００は１以上の記憶装置でのデータを実際に共有する。それ故、好ましくは、スナップショット１３０は、図１のデータ１２０のまわりの実線と破線の重なった境界により示されるように、ファイルシステム１００のような１以上の記憶装置での同じ物理データ１２０を含む。すなわち、上記スナップショットと上記ファイルシステムは重なる。これは、記憶容量および他のシステム資源の効率的な使用により、スナップショット１３０の短時間での作成を可能にする。

【００３３】

ファイルシステム１００は、好ましくは、ファイルシステム１００のスナップショットを指すスナップショットデータ１５０を含む。特に上記スナップショットデータ内のポインタ１６０は、好ましくは、それらのスナップショットのルートｉノードを指す。

【００３４】

スナップショット１３０は、好ましくは、他のスナップショットを指すスナップショットデータ１７０も含む。しかし、スナップショット１３０のスナップショットデータ１７０は、スナップショット１３０が好ましくはそれ自体を指さないので、ファイルシステム１００のスナップショットデータ１５０とは異なる。この違いは、図１で、ファイルシステム１００のスナップショットデータ１５０のまわりのスナップショット１３０の切欠により示される。

【００３５】

好ましくは、本発明によるファイルシステムのスナップショットは、アクティブなファイルシステムのアクティブなファイルシステムデータから分離したファイルシステムデータについての完全な階層を含む。この階層は、上記スナップショットのルートｉノード、および、可能ならば上記スナップショットに関してコピーされた他のノードおよびデータ

（図示されていない）に含まれる。

【００３６】

関連するアクティブなファイルシステムに元は使用された名前空間を複製するというスナップショットに関する上記ファイルシステム階層について、要求はない。１つの好ましい実施の形態では、スナップショットのルートｉノード内のファイル名（および他の組織データ）は、各スナップショットに関して記憶されねばならない組織データを最小にするため、ハッシュコードまたは他の技術を使用して圧縮可能である。しかし、代わりの実施の形態では、好ましいかも知れない状況で、人間のユーザにとって比較的読みやすい形で、各スナップショットに関する元の名前空間および他の組織データを維持することは、よりすぐれている。これは、そのようなスナップショットに基づくバックアップおよびリストア作業をする人間のユーザを助ける有益な効果をもつ。

【００３７】

図２は、アクティブなファイルシステムのスナップショットからのアクティブなファイルシステムの分岐を示す。

【００３８】

ファイルシステム１００はアクティブなので、上記ファイルシステム内のデータを変更するために、１つの仕組みが提供されねばならない。しかし、スナップショット１３０の整合性を維持するために、スナップショットのルートｉノード１４０により指されるデータは保存されねばならない。それ故、例えば、データ１２０がファイルシステム１００内で変更されたとき、変更されたデータ１２０’は１以上の記憶装置に記憶される。ファイルシステム１００のルートｉノード１１０およびすべての介在するｉノードならびに組織データは、変更されたデータ１２０’を指すように更新される。加えて、変更されなかったデータ１２０は１以上の記憶装置上に保存される。スナップショットのルートｉノード１４０はこの変更されなかったデータを指し続けるので、スナップショット１３０の整合性を維持する。

【００３９】

同様に、アクティブなファイルシステム１００からデータが削除されたとき、そのデータへのポインタは上記ファイルシステムから削除される。しかし、そのデータ自体は、もしスナップショット１３０の中に含まれれば、保存される（実際は、上記スナップショット自体が削除されたとき、このデータは削除できる。）。

【００４０】

実際には、ルートｉノード１１０、他のｉノード、およびファイルシステム１００に対する多くの変更に関するデータへの変更は、１以上の記憶装置に書き込まれる前に、累積される。そのような変更が書き込まれた後では、ファイルシステム１００は自己整合的である（すなわち整合点で）。好ましくは、スナップショットはそのような整合点でのみ作成される。

【００４１】

本発明によれば、スナップショット１３０は、そのスナップショットを書き込み可能にすることにより、新しいアクティブなファイルシステムに変換可能である。書き込み可能なスナップショット内のデータを変更するために、変更されたデータが１以上の記憶装置に書き込まれる。上記変更されたデータを指すルートｉノード１４０および介在するｉノードならびに組織データは更新される。さらに、上記データの変更されなかったコピーは、もしファイルシステム１００内にまだ含まれれば、保存される。この処理は、ファイルシステム１００に変更が加えられたときに発生する処理と実質的に同一であり、保存された上記未変更データのみが、ルートｉノード１１０が指すデータである。

【００４２】

すなわち、第１のアクティブなファイルシステム（すなわち、ファイルシステム１００）に変更が加えられたとき、変更されたデータが、第１のアクティブなファイルの、第２のアクティブなファイル（例えば、書き込み可能なスナップショット１３０）と共有されない場所に記録される。同様に、第２のアクティブなファイルシステムに変更が加えられ

たとき、変更されたデータが、第２のアクティブなファイルの、第１のアクティブなファイルと共有されない場所に記録される。結果として、第１のアクティブなファイルに加えられた変更は第２のアクティブなファイルに反映されず、第２のアクティブなファイルに加えられた変更は第１のアクティブなファイルに反映されない。

【００４３】

スナップショット１３０は、作成されるとき、実質的にファイルシステム１００と重なる。もし上記スナップショットがその作成のすぐ後に書き込み可能にされたら、上記書き込み可能なスナップショットより形成された新しいアクティブなファイルは、当初は元のアクティブなファイルシステムとそのデータのほとんどすべてを共有する。結果として、上記発明は、処理時間および記憶容量などの資源の効率的利用を用いて、新しいアクティブなファイルシステムの作成を可能にする。

【００４４】

変更されたデータを記憶する処理および変更されなかったデータを保存する処理は、ファイルシステム１００およびスナップショット１３０（読み取り専用か書き込み可能かに関わらず）を互いに分岐させる。この分岐は、ファイルシステム１００およびスナップショット１３０の間の重なりの減少により、図２に示されている。

【００４５】

図３は、図２のアクティブなファイルシステムおよびスナップショットの間の関係を示す。この種の図は、複数のファイルシステムおよびそれらのスナップショットの間の簡略化された図を提供する。図３では、ファイルシステム１００はスナップショット１３０を指す。加えてファイルシステム１００およびスナップショット１３０の両方は他のスナップショット（図示されていない）を指す。

【００４６】

図４は、本発明によりアクティブなファイルシステムに変換可能なスナップショットの連鎖を示す。この図では、第２のスナップショットがファイルシステム１００から作成される。上記第２のスナップショット作成時にスナップショット１００はまだスナップショット１３０を指しているので、スナップショット１８０は、スナップショット１３０を指すスナップショットデータ１９０を含む。

【００４７】

スナップショット１３０およびスナップショット１８０のいずれかまたは両方は、それらのスナップショットを書き込み可能にすることにより、アクティブなファイルシステムになることができる。１つのデータが上記アクティブなファイルシステム（すなわち、ファイルシステム１００、書き込み可能なスナップショット１３０または書き込み可能なスナップショット１８０）のいずれかに書き込まれると、上記ファイルシステムは互いに分岐する。

【００４８】

図５は、図４の上記アクティブなファイルシステムと上記スナップショットとの間の関係を示す。図５では、ファイルシステム１００はスナップショット１３０およびスナップショット１８０を指す。同様に、スナップショット１８０はスナップショット１３０を指し、スナップショット１３０は次に他の１以上のスナップショットを指すことができる。

【００４９】

図６は、本発明によりアクティブなファイルシステムに変換されたスナップショットを示す。この図では、スナップショット１８０は、書き込み可能にされることにより、アクティブなファイルシステム１８０’になった。この新しいアクティブなファイルシステムは変更可能であるので、もうファイルシステム１００の真のスナップショットを表さない。結果として、ファイルシステム１００のスナップショットデータ１５０の中のスナップショット１８０を指すスナップショットポインタは、例えば削除することにより、切断されている。

【００５０】

図７は、図６のアクティブなファイルシステム、新しいアクティブなファイルシステム

およびスナップショットの間の関係を示す。この図では、アクティブなファイルシステム１００はスナップショット１３０を指す。同様に、アクティブなファイルシステム１８０’はスナップショット１３０を指す。上に説明したように、ファイルシステム１００は、好ましくは、もはや、スナップショット１８０へのスナップショットポインタを含まない。しかし、ファイルシステム１００は、例えば１つのファイルシステムから他のファイルシステムへ移れるように、ファイルシステム１８０’へのポインタをまだ含むことができる。このファイルシステム間ポインタは、図７の破線で示されて、スナップショットポインタから区別される。

【００５１】

本発明によりアクティブなファイルシステムに変換可能なスナップショットのより複雑な連鎖を示す。図８では、ファイルシステム８００はアクティブなファイルシステムである。４つのスナップショットがこのファイルシステムから作成されている。スナップショット８１０が最も古く、スナップショット８２０は次に古く、スナップショット８３０はスナップショット８２０の次に古く、スナップショット８４０は最も新しい。スナップショット８１０より古いスナップショットが削除されていて、これにより、いずれかの他のスナップショットまたはアクティブなファイルシステムと重ならないデータにより占有される記憶容量を開放する。スナップショット８１０から８４０の各々は、書き込み可能にされることにより、アクティブなファイルシステムになることができる。

【００５２】

図９は、図８に示されている連鎖と、本発明によりアクティブなファイルシステムに変換されたスナップショットの１つとを示す。

【００５３】

図９では、スナップショット８３０は、データが変更可能、追加可能、および削除可能なアクティブなファイルシステム８３０’に変換された。結果として、ファイルシステム８００は、好ましくは、もう、スナップショットとしてスナップショット８３０を指さない。アクティブなファイルシステム８３０’はスナップショット８１０および８２０を指し続けていられる。

【００５４】

図１０は、複数のアクティブなファイルシステムとそれらに関連するスナップショットとの間の、本発明による他の可能な関係を示す。

【００５５】

追加のスナップショットが図９のアクティブなファイルシステムから作成されたことを除いて、図１０の上部は図９に対応する。それ故、スナップショット１０００はファイルシステム８００から作成されていて、スナップショット１０１０はファイルシステム８３０’から作成されている。さらに、スナップショット８１０は、１以上の記憶装置上の容量を開放するために削除されている。

【００５６】

アクティブなファイルシステム８００および８３０’の両方とも、共通のスナップショット８２０にさかのぼれる。しかし、そのスナップショットが削除されるとき、上記アクティブなファイルシステムは、もはや、共通のスナップショットを共有しない。この状況はファイルシステム１０２０およびスナップショット１０３０から１０５０に関して発生する。この配置は、すべて１つの記憶装置または１組の記憶装置で、複数のアクティブファイルシステムとそれらに関連するスナップショットとの間のリンクにより形成される「森」（すなわち、接続されていない木の集まり）をもつことが可能であることを示す。上記ファイルシステムおよびそれらのスナップショットが、もはや、共通のスナップショットを指していないという事実にもかかわらず、これらのスナップショットおよび上記アクティブなファイルシステムでさえ、まだ、データを共有している（すなわち、重なっている）ので、これにより本発明の効率性を維持している。

【００５７】

以下の説明では、新しいアクティブなファイルシステムはスナップショットから作成さ

れる。しかし、本発明は新しいアクティブなファイルシステムを作成するために、スナップショットの実際の作成を要求していない。むしろ、要求していることは、スナップショットの中で見出される構造のラインに沿った構造、すなわち、スナップショットのルートｉノードの中で見出される構造のラインにしたがった組織データの作成と、上記組織データにより指されたデータの保存である。

【００５８】

さらに、本発明は上に説明した特定の配置に限定されない。むしろ、それらの配置は、アクティブなファイルシステム、スナップショットおよび新しいアクティブなファイルシステムの間の関係の複数の可能な型を示す。他の配置は可能であり本発明の範囲に含まれる。

【００５９】

システム要素

【００６０】

図１１は、本発明による複数のアクティブなファイルシステムを含む記憶システムのブロック図を示す。

【００６１】

システム１１００は、少なくとも１つのファイルシステムプロセッサ１１１０（すなわち、コントローラ）およびハードディスクドライブまたはＣＤ－ＲＯＭドライブなどの少なくとも１つの記憶装置１１２０を含む。このシステムは、また、好ましくは、情報を送受信するための、少なくとも１つのコンピュータ・デバイスまたは、ネットワークへのインタフェース１１３０を含む。別の実施の形態では、プロセッサ１１１０は、インタフェース１１３０経由で記憶システムに接続されたコンピュータ・デバイスのためのプロセッサである。

【００６２】

プロセッサ１１１０は、ここで説明するように、プログラムおよびデータメモリに制御されて、上記ファイルシステムに関連するタスクを実行する。上記プログラムおよびデータメモリは、記憶装置１１２０で演算を実行する制御プロセッサ１１１０のための（さらに可能ならば、プロセッサ１１１０と協調して記憶装置１１２０を制御するための）適切なソフトウェアを含む。

【００６３】

好ましい実施の形態では、少なくとも１つのそのような記憶装置１１２０が１以上のブートレコード１１４０を含む。各ブートレコード１１４０は、アクティブファイルシステムに関するファイルシステム階層におけるルートデータブロック（すなわち、ｉノード）を指定する２以上（好ましくは２）のエントリを含む。１つのアクティブなファイルがある場合は、好ましくは１つのそのようなブートレコードがあり、１より多いそのようなアクティブなファイルがある場合は、好ましくは１より多いそのようなブートレコードがある。

【００６４】

上で述べたように、１以上のアクティブなファイルシステムが記憶装置１１２０に存在しうる。そのような場合、上記ファイルシステム維持装置（すなわち、プログラムの制御で動作するプロセッサ１１１０）は、好ましくは、そのようなアクティブなファイルシステムごとに、１より多いブートレコードを指定し、整然と維持する。

【００６５】

読み取り専用スナップショットも記憶装置１１２０の中に存在できる。この場合、アクティブなファイルシステムからスナップショットへのポインタおよびスナップショットから他のスナップショットへのポインタは、上に説明したように、上記記憶装置の中に記憶される。

【００６６】

高い可用性

【００６７】

本発明により、複数の同時動作のファイルサーバにより使用されている複数の同時にアクティブなファイルシステムを含むファイルシステムクラスタのブロック図を示す。

【００６８】

１つのファイルシステムクラスタは複数のファイルシステムプロセッサ１２００および１以上のファイルシステムディスク１２１０を含む。好ましい実施の形態では、そのような各プロセッサ１２００は、ファイルサーバとして動作するために配置されていて、例えば既知のファイルサーバプロトコルを使用して、ファイルサーバ要求を受け付け、ファイルサーバ応答を返すことができる。好ましい実施の形態では、１以上のファイルシステムディスク１２１０はそのような複数のディスクを含むので、全体の高い可用性クラスタに関し、どの個々のディスク１２１０も単一位置の故障を発生させない。好ましくは本発明と共に使用されるＷＡＦＬ（Ｗｒｉｔｅ　Ａｎｙｗｈｅｒｅ　Ｆｉｌｅ　Ｓｙｓｔｅｍ）はそのような配置を組み込む。

【００６９】

上に説明したように、複数のプロセッサ１２００は、複数の並列に書き込み可能なアクティブなファイルシステムを、それらの並列に書き込み可能なアクティブなファイルシステムに関連するスナップショットとともに維持できる。上記アクティブなファイルシステムおよびスナップショットは、同じ１組のディスク１２２０に維持可能である。こうして、１組のプロセッサ１２００および１組のディスク１２２０は、資源の実質的な重複の必要なしに高い可用性クラスタを提供可能である。

【００７０】

別の実施の形態

【００７１】

本発明は、複数のアクティブなファイルシステムを作成および維持するための方法だけでなく、本発明の方法を実行するソフトウェアおよび／または１以上の記憶装置などのハードウェア、そして種々の他の実施の形態において、具体化できる。

【００７２】

以上の説明では、本発明の好ましい実施の形態は、好ましい処理ステップおよびデータ構造に関して説明されている。しかし、当業者は、本出願の熟読後、本発明の実施の形態が、プログラムの制御で動作する特定の処理ステップおよびデータ構造に適応可能な１以上の汎用プロセッサまたは専用プロセッサを使用して実行可能であること、そのような処理ステップおよびデータ構造はメモリ（例えば、ＤＲＡＭ、ＳＲＡＭ、ハードディスク、キャッシュなどの固定メモリおよびフロッピーディスク、ＣＤ－ＲＯＭ、データテープなどの着脱式メモリ）に記憶されるか入出力され、そのようなプロセッサで実行可能な命令（例えば、直接実行可能なオブジェクトコード、コンパイル後に実行可能なソースコード、インタプリタを使用して実行可能なコードなど）を含む情報により実施可能であること、また、そのような装置を使用してここで説明されている好ましい上記処理ステップおよびデータ構造の実施は必要以上の実験またはさらなる発明を必要としないことを理解するだろう。

【図面の簡単な説明】

【００７３】

【図１】本発明によりアクティブなファイルシステムに変換可能なスナップショットの作成を示す図

【図２】アクティブなファイルシステムのスナップショットからのアクティブなファイルシステムの分岐を示す図

【図３】図２におけるアクティブなファイルシステムおよびスナップショットの間の関係を示す図

【図４】本発明によりアクティブなファイルシステムに変換可能なスナップショットの連鎖を示す図

【図５】図４におけるアクティブなファイルシステムおよびスナップショットの間の関係を示す図

【図６】本発明によりアクティブなファイルシステムに変換可能なスナップショットを示す図
【図７】図６におけるアクティブなファイルシステム、新しいアクティブなファイルシステム、およびスナップショットの間の関係を示す図
【図８】本発明によりアクティブなファイルシステムに変換可能なスナップショットのより複雑な連鎖を示す図
【図９】図８に示されている連鎖と本発明によりアクティブなファイルシステムに変換されたスナップショットの１つとを示す図
【図１０】本発明による複数のアクティブなファイルシステムおよびそれらに関係するスナップショットの間のより多くの別の可能な関係を示す図
【図１１】本発明による複数のアクティブなファイルシステムを含む記憶装システムのブロック図
【図１２】本発明により、複数の同時作動のファイルサーバにより使用される複数の同時にアクティブとなるファイルシステムを含むファイルシステムクラスタのブロック図
【符号の説明】
【００７４】
１００　ファイルシステム
１１０　ルートｉノード
１２０　（変更されなかった）データ
１２０’　変更されたデータ
１３０　スナップショット
１４０　スナップショットのルートｉノード
１５０　ファイルシステム（１００）のスナップショットデータ
１６０　スナップショットデータ内のポインタ
１７０　スナップショット１３０の他のスナップショットを指すスナップショットデータ
１８０　スナップショット
１８０’アクティブなファイルシステム
１９０　スナップショット（１３０）を指すスナップショットデータ
８００　ファイルシステム
８１０　スナップショット
８２０　スナップショット
８３０　スナップショット
８３０’アクティブなファイルシステム
８４０　スナップショット
１０００　スナップショット
１０１０　スナップショット
１０２０　ファイルシステム
１０３０　スナップショット
１０４０　スナップショット
１０５０　スナップショット
１１００　システム
１１１０　ファイルシステムプロセッサ
１１２０　記憶装置
１１３０　インタフェース
１１４０　ブートレコード
１２００　ファイルシステムプロセッサ
１２１０　ファイルシステムディスク
１２２０　ディスク

【図１】

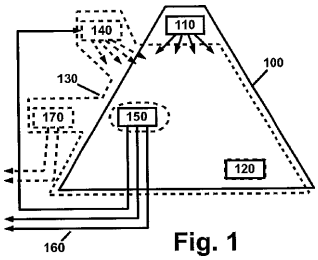

Fig. 1

【図２】

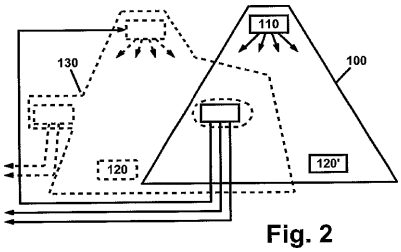

Fig. 2

【図３】

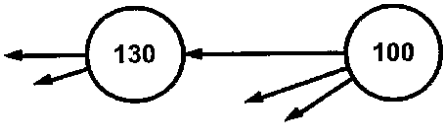

Fig. 3

【図４】

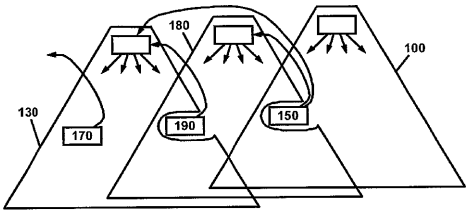

Fig. 4

【図５】

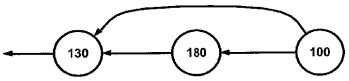

Fig. 5

【図６】

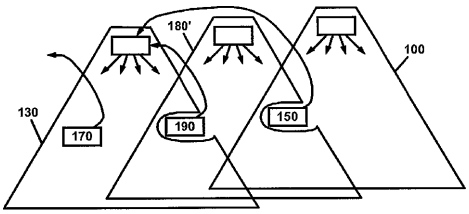

Fig. 6

【図７】

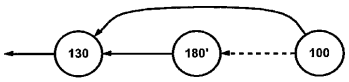

Fig. 7

【図８】

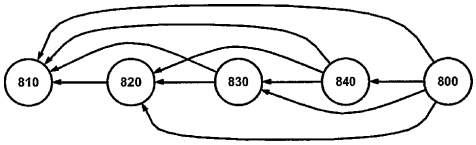

Fig. 8

【図９】

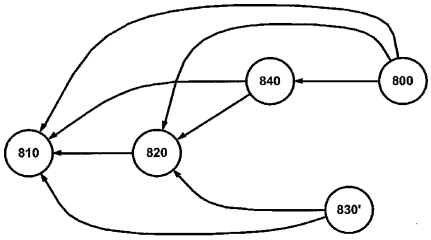

Fig. 9

【図１０】

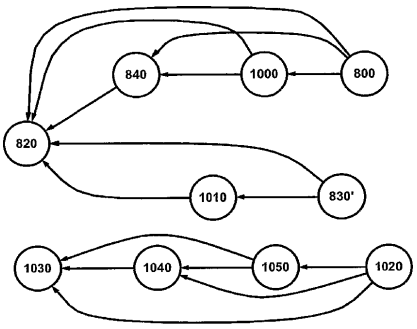

Fig. 10

【図１１】

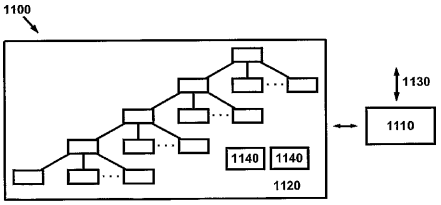

Fig. 11

【図１２】

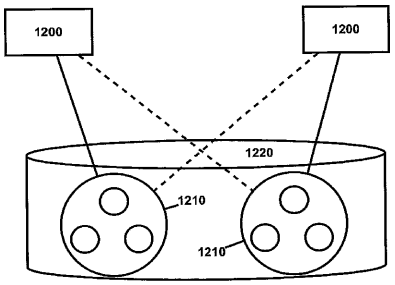

Fig. 12

フロントページの続き

(72)発明者　ジョン・エドワーズ
アメリカ合衆国９４０８７－２０７６カリフォルニア州サニーベイル、クランダノ・コート１１７３番
(72)発明者　ブレイク・ルイス
アメリカ合衆国９４０２２カリフォルニア州ロス・アルトス・ヒルズ、ビア・フェリス２７８７８番

審査官　田川　泰宏

(56)参考文献　実良　栗富，ＩＴプロのための解説，日経コンピュータ，日本，日経ＢＰ社，２００２年　５月２０日，ｎｏ．５４８，p.158-164

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　12/00