

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2017-504924

(P2017-504924A)

(43) 公表日 平成29年2月9日(2017.2.9)

(51) Int.Cl.
G06F 12/00 (2006.01)F I
G06F 12/00 545A

テーマコード (参考)

審査請求 未請求 予備審査請求 未請求 (全 26 頁)

(21) 出願番号 特願2016-567483 (P2016-567483)
 (86) (22) 出願日 平成27年2月4日 (2015.2.4)
 (85) 翻訳文提出日 平成28年9月1日 (2016.9.1)
 (86) 国際出願番号 PCT/US2015/014492
 (87) 国際公開番号 W02015/120071
 (87) 国際公開日 平成27年8月13日 (2015.8.13)
 (31) 優先権主張番号 61/935, 618
 (32) 優先日 平成26年2月4日 (2014.2.4)
 (33) 優先権主張国 米国 (US)

(71) 出願人 514008826
 エグザプロックス・コーポレーション
 アメリカ合衆国カリフォルニア州94086,
 サニーベール, ソノラ・コート1156番
 (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】 ファイルシステムのコンテンツベースの編成

(57) 【要約】

コンテンツアドレス可能オブジェクトストア上のファイルシステムを編成し及び管理するための方法及びシステムが提供される。ユーザコンテンツが、コンテンツアドレス可能オブジェクトストアにおけるアドレスに関連付けられたコンテンツハッシュを含むレコードに関連付けられる。コンテンツハッシュは、ユーザコンテンツの関数であり、ファイルをデータオブジェクトに分割することと、データオブジェクト上にハッシュのポインタツリーを生成することと、ポインタツリーのルートのハッシュを算出することとによって決定される。レコード、ポインタツリー、及びファイルが、オブジェクトストアに記憶される。オブジェクトストアからユーザコンテンツを読み出すことが、レコードを読み出すことと、コンテンツハッシュを取り出してポインタツリーのルートを発見することと、ポインタツリーのリーフを下降して、ユーザコンテンツに関連付けられたデータオブジェクトのハッシュを読み出すことと、データオブジェクトを順次ファイルデータにおいて再組立することを含む。

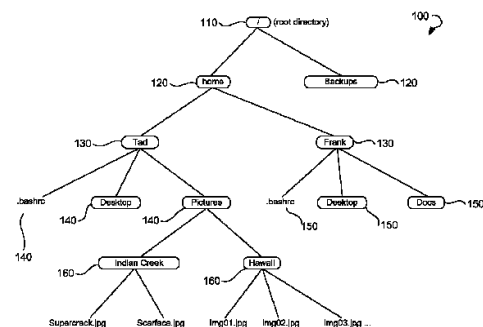


FIG. 1

【特許請求の範囲】

【請求項 1】

コンテンツアドレス可能オブジェクトストア上のファイルシステムを編成し及び管理する、コンピュータにより実施される方法であって、

レコードをユーザコンテンツに関連付けるステップであって、前記レコードは、前記ユーザコンテンツに関連付けられた属性を含み、前記属性は、前記コンテンツアドレス可能オブジェクトストアにおけるアドレスに関連付けられたコンテンツハッシュを少なくとも含む、ステップと、

前記ユーザコンテンツを複数のデータオブジェクトに分割するステップであって、各データオブジェクトは所定の最小サイズを有する、ステップと、

前記複数のデータオブジェクトの各データオブジェクトについて、それぞれのデータオブジェクトハッシュを生成するステップであって、前記データオブジェクトハッシュは、前記データオブジェクトに関連付けられたそれぞれのデータオブジェクトコンテンツの関数である、ステップと、

前記データオブジェクトハッシュに基づいて、前記コンテンツハッシュを決定するステップと、

を含む方法。

【請求項 2】

前記コンテンツハッシュを決定するステップは、

前記複数のデータオブジェクトの中のオブジェクトの数を算出することと、

前記オブジェクトの数が 1 である場合、前記コンテンツハッシュを前記データオブジェクトハッシュに設定することと、

前記オブジェクトの数が 2 以上である場合、前記データオブジェクトハッシュに基づいてポインタツリーを生成することであって、前記ポインタツリーを生成することは、

前記データオブジェクトハッシュを複数のポインタオブジェクトにグループ化することと、

前記ポインタオブジェクトを前記ポインタツリーにおけるレイヤのリーフとして割り当てることと、

前記レイヤが 2 つ以上のリーフを含む場合、前記ポインタツリーの次レイヤを構築することであり、前記次レイヤを構築することは、

前記ポインタオブジェクトの各々について、ポインタハッシュを生成することであり、前記ポインタハッシュは、前記ポインタオブジェクトに関連付けられたそれぞれのポインタオブジェクトコンテンツの関数である、こと、

前記ポインタハッシュを複数のさらなるポインタオブジェクトにグループ化することであり、前記さらなるポインタオブジェクトの各々は前記所定の最小サイズを有する、こと、

前記さらなるポインタオブジェクトを、前記ポインタツリーにおけるさらなるレイヤのリーフとして割り当てること、並びに、

前記ポインタハッシュを生成すること、前記ポインタハッシュをグループ化すること、及び前記さらなるポインタオブジェクトを割り当てることを、1 つのリーフを有する最終レイヤが受信されるまで繰り返すこと、

を含む、ことと、

前記最終レイヤのリーフを、前記ポインタツリーのルートとして割り当てることと、

前記ポインタツリーのルートのコンテンツに基づいて、ルートポインタハッシュを計算することと、

を含む、ことと、

前記レコードの前記コンテンツハッシュを前記ルートポインタハッシュに設定することと、

を含む、請求項 1 に記載の方法。

【請求項 3】

前記ポインタツリーの深さが、下記の式： $\text{MaxFileSize} = \text{BLKSZ} \times (\text{truncate}(\text{BLKSZ} \div \text{SMASHSZ}))^{\text{depth}}$ により決定され、MaxFileSizeはコンテンツエンティティのサイズであり、BLKSZは前記所定の最小サイズであり、SMASHSZは前記コンテンツハッシュのサイズであり、前記truncateは、実数のうちの整数部分を返す、実数の関数である、請求項2に記載の方法。

【請求項4】

前記ユーザコンテンツは、データファイルとフォルダとのうち1つ以上を含む、請求項2に記載の方法。

【請求項5】

前記フォルダは、さらなるレコードに関連付けられた少なくとも1つのフォルダエントリを含み、前記さらなるレコードは、さらなるユーザコンテンツに関連付けられ、前記少なくとも1つのフォルダエントリは、

10

前記さらなるユーザコンテンツのタイプに関連付けられた、エントリのタイプと、

前記さらなるユーザコンテンツの長さに関連付けられた、エントリの長さ、と、

前記さらなるユーザコンテンツの名前に関連付けられた、エントリの名前と、

さらなるコンテンツハッシュに関連付けられた、エントリの属性であって、前記さらなるコンテンツハッシュは、前記さらなるレコードの属性に基づいて生成される、エントリの属性と、

のうち少なくとも1つを含む、請求項4に記載の方法。

【請求項6】

20

前記フォルダは、さらなるレコードに関連付けられた1つ以上のエントリを含み、前記さらなるレコードの各々は、前記ユーザコンテンツの一意の時間バージョンに関連付けられる、請求項5に記載の方法。

【請求項7】

前記レコードの前記属性は、前のコンテンツハッシュを含み、前記前のコンテンツハッシュは、前のレコードの属性に関連付けられたコンテンツに基づいて生成され、前記前のレコードは、前記ユーザコンテンツのそれぞれの前のバージョンに関連付けられる、請求項4に記載の方法。

【請求項8】

前記ユーザコンテンツにおける変更に応答して、

30

前記ユーザコンテンツに関連付けられた前記ポインタツリーを少なくとも部分的に再生成するステップと、

さらなるレコードを生成するステップと、

前記さらなるレコードのコンテンツハッシュを、前記の再生成されたポインタツリーのルートに関連付けられたポインタハッシュに設定するステップと、

前記レコードの前記属性に関連付けられたコンテンツに基づいて、レコードコンテンツハッシュを計算するステップと、

前記新しいレコードの前のコンテンツハッシュを、前記レコードコンテンツハッシュに設定するステップと、

をさらに含む請求項7に記載の方法。

40

【請求項9】

前記データオブジェクトをクラスタストレージに書き込むステップ、をさらに含み、前記データオブジェクトを書き込むステップは、

前記データオブジェクトと前記データオブジェクトのタイプとを受信することと、

前記データオブジェクトハッシュを計算することと、

前記クラスタストレージの物理ノードを決定することであって、前記物理ノードは前記データオブジェクトのコピーを含む、ことと、

前記の決定された物理ノードの各々において、下記のステップ：

前記コンテンツアドレス可能オブジェクトストアにおける前記データオブジェクトハッシュをルックアップして、データオブジェクトメタデータを取得することであり、前記

50

メタデータは少なくともディスク識別子とディスク内の場所とを含む、こと、並びに、
前記データオブジェクトハッシュが発見されない場合、
前記物理ノードに前記データオブジェクトのコピーを書き込むこと、及び
前記コンテンツアドレス可能オブジェクトストアを、前記データオブジェクトに関連付けられた新しいメタデータで更新すること、
を実行することと、
を含む、請求項2に記載の方法。

【請求項10】

前記クラスタストレージにファイルを書き込むステップ、をさらに含み、前記ファイルを書き込むステップは、
前記ファイルを受信することと、
前記ファイルをファイルデータオブジェクトに分けることと、
前記ファイルデータオブジェクトのファイルデータオブジェクトハッシュを計算することと、

ファイルポインタツリーを、
前記ファイルデータオブジェクトハッシュを、前記ファイルポインタツリーのリーフに関連付けられたファイルポインタオブジェクトに挿入すること、

前記ファイルポインタオブジェクトがフルであり、あるいはすべてのファイルデータオブジェクトが処理されている場合、前記ファイルポインタオブジェクトのファイルポインタハッシュを計算すること、及び、前記ファイルポインタハッシュを親ファイルポインタオブジェクトに挿入すること、並びに、

前記ファイルポインタオブジェクトのファイルポインタハッシュを計算することを、前記ファイルポインタツリーが構築されるまで繰り返すこと、
により生成することと、

前記ファイルポインタツリーのルートのファイルポインタハッシュを、ファイルレコードのコンテンツハッシュに挿入することであって、前記ファイルレコードは前記ファイルに関連付けられる、ことと、

前記ファイルレコードの属性を、少なくともファイルサイズ及びタイムスタンプで更新することと、

前記ファイルレコードの属性に基づいて、レコードコンテンツハッシュを計算することと、

集合からの各オブジェクトを前記クラスタストレージに書き込むことであって、前記集合は、前記ファイルデータオブジェクト、前記ファイルポインタオブジェクト、及び前記ファイルレコードを含む、ことと、

前記書き込みの成功と失敗とのうち一方の指標を返すことと、
を含む、請求項9に記載の方法。

【請求項11】

クラスタストレージから前記データオブジェクトを読み出すステップ、をさらに含み、前記データオブジェクトを読み出すステップは、

前記データオブジェクトハッシュを受信することと、

前記クラスタストレージの物理ノードのセットを決定することであって、前記物理ノードは、前記データオブジェクトのコピーを有する可能性の最大から可能性の最小に順序付けられる、ことと、

各々の順序付けられた物理ノードについて、前記コンテンツアドレス可能オブジェクトストアにおける前記データオブジェクトハッシュをルックアップすることと、

前記データオブジェクトハッシュが発見される場合及びとき、

前記データオブジェクトハッシュに基づいて、ディスク識別子とディスク内の場所とを含むデータオブジェクトメタデータを受信することと、

前記データオブジェクトメタデータを用いて、前記物理ノードから前記データオブジェクトのコピーを読み出すことと、

10

20

30

40

50

を含む、請求項 2 に記載の方法。

【請求項 1 2】

前記クラスタストレージからファイルを読み出すステップ、をさらに含み、前記ファイルを読み出すステップは、

前記ファイルに関連付けられたファイルレコードのレコードコンテンツハッシュを受信することと、

前記データオブジェクトを読み出すステップを適用することによって、前記クラスタストレージから前記ファイルレコードを読み出すことと、

前記ファイルレコードの前記コンテンツハッシュを取り出すことによって、ファイルポインタツリーのルートに関連付けられたルートポインタハッシュを決定することと、

前記ファイルポインタツリーを前記ルートから始めて下降して、前記ファイルポインタツリーの子リーフの中のファイルポインタハッシュを取り出すこと、及び、前記データオブジェクトを読み出すステップを適用することによって、前記ファイルポインタハッシュに関連付けられたファイルデータオブジェクトを読み出すことと、

前記ファイルデータオブジェクトを順次ファイルデータに再組立てすることと、

前記順次ファイルデータを返すことと、

を含む、請求項 1 1 に記載の方法。

【請求項 1 3】

コンテンツアドレス可能オブジェクトストア上のファイルシステムを編成し及び管理するシステムであって、

少なくとも 1 つのプロセッサと、

前記少なくとも 1 つのプロセッサに通信可能に結合されたメモリと、

を含む、

前記メモリは、前記少なくとも 1 つのプロセッサにより実行されるときに方法を実行する命令を記憶し、前記方法は、

レコードをユーザコンテンツに関連付けるステップであって、前記レコードは、前記ユーザコンテンツに関連付けられた属性を含み、前記属性は、前記コンテンツアドレス可能オブジェクトストアにおけるアドレスに関連付けられたコンテンツハッシュを少なくとも含む、ステップと、

前記ユーザコンテンツを複数のデータオブジェクトに分割するステップであって、各データオブジェクトは所定の最小サイズを有する、ステップと、

前記複数のデータオブジェクトの各データオブジェクトについて、それぞれのデータオブジェクトハッシュを生成するステップであって、前記データオブジェクトハッシュは、前記データオブジェクトに関連付けられたそれぞれのデータオブジェクトコンテンツの関数である、ステップと、

前記データオブジェクトハッシュに基づいて、前記コンテンツハッシュを決定するステップと、

を含む、システム。

【請求項 1 4】

前記コンテンツハッシュを決定するステップは、

前記複数のデータオブジェクトの中のオブジェクトの数を算出することと、

前記オブジェクトの数が 1 である場合、前記コンテンツハッシュを前記データオブジェクトハッシュに設定することと、

前記オブジェクトの数が 2 以上である場合、前記データオブジェクトハッシュに基づいてポインタツリーを生成することであって、前記ポインタツリーを生成することは、

前記データオブジェクトハッシュを複数のポインタオブジェクトにグループ化することと、

前記ポインタオブジェクトを前記ポインタツリーにおけるレイヤのリーフとして割り当てることと、

前記レイヤが 2 つ以上のリーフを含む場合、前記ポインタツリーの次レイヤを構築す

10

20

30

40

50

ることであり、前記次レイヤを構築することは、

前記ポインタオブジェクトの各々について、ポインタハッシュを生成することであり、前記ポインタハッシュは、前記ポインタオブジェクトに関連付けられたそれぞれのポインタオブジェクトコンテンツの関数である、こと、

前記ポインタハッシュを複数のさらなるポインタオブジェクトにグループ化することであり、前記さらなるポインタオブジェクトは前記所定の最小サイズを有する、こと、

前記さらなるポインタオブジェクトを、前記ポインタツリーにおけるさらなるレイヤのリーフとして割り当てること、並びに、

前記ポインタハッシュを生成すること、前記ポインタハッシュをグループ化すること、及び前記さらなるポインタオブジェクトを割り当てることを、1つのリーフを有する最終レイヤが受信されるまで繰り返すこと、

を含む、ことと、

前記最終レイヤのリーフを、前記ポインタツリーのルートとして割り当てることと、

前記ポインタツリーのルートのコンテンツに基づいて、ルートポインタハッシュを計算することと、

を含む、ことと、

前記レコードの前記コンテンツハッシュを前記ルートポインタハッシュに設定することと、

を含む、請求項13に記載のシステム。

【請求項15】

前記ポインタツリーの深さが、下記の式： $\text{MaxFileSize} = \text{BLKSZ} \times (\text{truncate}(\text{BLKSZ} \div \text{SMASHSZ}))^{\text{depth}}$ により決定され、MaxFileSizeはコンテンツエンティティのサイズであり、BLKSZは前記所定の最小サイズであり、SMASHSZは前記コンテンツハッシュのサイズであり、前記truncateは、実数のうちの整数部分を返す、実数の関数である、請求項14に記載のシステム。

【請求項16】

前記ユーザコンテンツは、データファイルとフォルダとのうち1つ以上を含む、請求項14に記載のシステム。

【請求項17】

前記フォルダは、さらなるレコードに関連付けられた少なくとも1つのフォルダエントリを含み、前記さらなるレコードは、さらなるユーザコンテンツに関連付けられ、前記少なくとも1つのフォルダエントリは、

前記さらなるユーザコンテンツのタイプに関連付けられた、エントリのタイプと、

前記さらなるユーザコンテンツの長さに関連付けられた、エントリの長さ、と、

前記さらなるユーザコンテンツの名前に関連付けられた、エントリの名前と、

さらなるコンテンツハッシュに関連付けられた、エントリの属性であって、前記さらなるコンテンツハッシュは、前記さらなるレコードの属性に基づいて生成される、エントリの属性と、

のうち少なくとも1つを含む、請求項16に記載のシステム。

【請求項18】

前記フォルダは、さらなるレコードに関連付けられた1つ以上のエントリを含み、前記さらなるレコードの各々は、前記ユーザコンテンツの一意の時間バージョンに関連付けられる、請求項17に記載のシステム。

【請求項19】

前記レコードの前記属性は、前のコンテンツハッシュを含み、前記前のコンテンツハッシュは、前のレコードの属性に関連付けられたコンテンツに基づいて生成され、前記前のレコードは、前記ユーザコンテンツのそれぞれの前のバージョンに関連付けられる、請求項16に記載のシステム。

【請求項20】

前記ユーザコンテンツにおける変更に応答して、

前記ユーザコンテンツに関連付けられた前記ポインタツリーを少なくとも部分的に再生成するステップと、

さらなるレコードを生成するステップと、

前記さらなるレコードのコンテンツハッシュを、前記の再生成されたポインタツリーのルートに関連付けられたポインタハッシュに設定するステップと、

前記レコードの前記属性に関連付けられたコンテンツに基づいて、レコードコンテンツハッシュを計算するステップと、

前記新しいレコードの前のコンテンツハッシュを、前記レコードコンテンツハッシュに設定するステップと、

をさらに含む請求項 19 に記載のシステム。

10

【請求項 21】

前記データオブジェクトをクラスタストレージに書き込むステップ、をさらに含み、前記データオブジェクトを書き込むステップは、

前記データオブジェクトと前記データオブジェクトのタイプとを受信することと、

前記データオブジェクトハッシュを計算することと、

前記クラスタストレージの物理ノードを決定することであって、前記物理ノードは前記データオブジェクトのコピーを含む、ことと、

前記の決定された物理ノードの各々において、下記のステップ：

前記コンテンツアドレス可能オブジェクトストアにおける前記データオブジェクトハッシュをルックアップして、データオブジェクトメタデータを取得することであり、前記メタデータは少なくともディスク識別子とディスク内の場所とを含む、こと、並びに、

20

前記データオブジェクトハッシュが発見されない場合、

前記物理ノードに前記データオブジェクトのコピーを書き込むこと、及び

前記コンテンツアドレス可能オブジェクトストアを、前記データオブジェクトに関連付けられた新しいメタデータで更新すること、

を実行することと、

を含む、請求項 14 に記載のシステム。

【請求項 22】

前記クラスタストレージにファイルを書き込むステップ、をさらに含み、前記ファイルを書き込むステップは、

30

前記ファイルを受信することと、

前記ファイルをファイルデータオブジェクトに分けることと、

前記ファイルデータオブジェクトのファイルデータオブジェクトハッシュを計算することと、

ファイルポインタツリーを、

前記ファイルデータオブジェクトハッシュを、前記ファイルポインタツリーのリーフに関連付けられたファイルポインタオブジェクトに挿入すること、

前記ファイルポインタオブジェクトがフルであり、あるいはすべてのファイルデータオブジェクトが処理されている場合、前記ファイルポインタオブジェクトのファイルポインタハッシュを計算すること、及び、前記ファイルポインタハッシュを親ファイルポインタオブジェクトに挿入すること、並びに、

40

前記ファイルポインタオブジェクトのファイルポインタハッシュを計算することを、前記ファイルポインタツリーが構築されるまで繰り返すこと、

により生成することと、

前記ファイルポインタツリーのルートのファイルポインタハッシュを、ファイルレコードのコンテンツハッシュに挿入することであって、前記ファイルレコードは前記ファイルに関連付けられる、ことと、

前記ファイルレコードの属性を、少なくともファイルサイズ及びタイムスタンプで更新することと、

前記ファイルレコードの属性に基づいて、レコードコンテンツハッシュを計算すること

50

と、

集合からの各オブジェクトを前記クラスタストレージに書き込むことであって、前記集合は、前記ファイルデータオブジェクト、前記ファイルポインタオブジェクト、及び前記ファイルレコードを含む、ことと、

前記書き込みの成功と失敗とのうち一方の指標を返すことと、
を含む、請求項 2 1 に記載のシステム。

【請求項 2 3】

クラスタストレージから前記データオブジェクトを読み出すステップ、をさらに含み、前記データオブジェクトを読み出すステップは、

前記データオブジェクトハッシュを受信することと、

10

前記クラスタストレージの物理ノードのセットを決定することであって、前記物理ノードは、前記データオブジェクトのコピーを有する可能性の最大から可能性の最小に順序付けられる、ことと、

各々の順序付けられた物理ノードについて、前記コンテンツアドレス可能オブジェクトストアにおける前記データオブジェクトハッシュをルックアップすることと、

前記データオブジェクトハッシュが発見される場合及びとき、

前記データオブジェクトハッシュに基づいて、ディスク識別子とディスク内の場所とを含むデータオブジェクトメタデータを受信することと、

前記データオブジェクトメタデータを用いて、前記物理ノードから前記データオブジェクトのコピーを読み出すことと、

20

を含む、請求項 1 4 に記載のシステム。

【請求項 2 4】

前記クラスタストレージからファイルを読み出すステップ、をさらに含み、前記ファイルを読み出すステップは、

前記ファイルに関連付けられたファイルレコードのレコードコンテンツハッシュを受信することと、

前記データオブジェクトを読み出すステップを適用することによって、前記クラスタストレージから前記ファイルレコードを読み出すことと、

前記ファイルレコードの前記コンテンツハッシュを取り出すことによって、ファイルポインタツリーのルートに関連付けられたルートポインタハッシュを決定することと、

30

前記ファイルポインタツリーを前記ルートから始めて下降して、前記ファイルポインタツリーの子リーフの中のファイルポインタハッシュを取り出し、前記データオブジェクトを読み出すステップを適用することによって、前記ファイルポインタハッシュに関連付けられたファイルデータオブジェクトを読み出すことと、

前記ファイルデータオブジェクトを順次ファイルデータに再組立てすることと、

前記順次ファイルデータを返すことと、

を含む、請求項 2 3 に記載のシステム。

【請求項 2 5】

1 つ以上のプロセッサにより実行されるときにコンテンツアドレス可能オブジェクトストア上のファイルシステムを編成し及び管理する方法を実行する命令を具現化させた非一時的コンピュータ可読記憶媒体であって、前記方法は、

40

レコードをユーザコンテンツに関連付けるステップであって、前記レコードは、前記ユーザコンテンツに関連付けられた属性を含み、前記属性は、前記コンテンツアドレス可能オブジェクトストアにおけるアドレスに関連付けられたコンテンツハッシュを少なくとも含む、ステップと、

前記ユーザコンテンツを複数のデータオブジェクトに分割するステップであって、データオブジェクトの各々は所定の最小サイズを有する、ステップと、

前記複数のデータオブジェクトの各データオブジェクトについて、それぞれのデータオブジェクトハッシュを生成するステップであって、前記データオブジェクトハッシュは、前記データオブジェクトに関連付けられたそれぞれのデータオブジェクトコンテンツの関

50

数である、ステップと、

前記データオブジェクトハッシュに基づいて、前記コンテンツハッシュを決定するステップと、

を含む、非一時的コンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

本出願は、2014年2月4日に申請された米国仮出願第61/935,618号の利益を主張する。前述された出願の開示は、すべての目的について、本明細書において参照により援用される。

10

【0002】

本開示は、概してデータ処理に関し、より具体的には、コンピュータファイルシステムの編成及び管理のための方法及びシステムに関する。

【背景技術】

【0003】

本セクションに説明されるアプローチは、追求されたことがある可能性があるが、必ずしもこれまでに着想され、あるいは追求されてきたアプローチではない。それゆえに、他に示されない限り、本セクションに説明されるアプローチのいずれも、単に本セクションに含まれるという理由で従来技術と見なすと仮定されるべきではない。

20

【0004】

コンピュータシステムにおいて、ハードディスクなどの記憶装置上に保有されるデータは、ファイルシステムとして編成される。大抵、ファイルシステムはツリー構造を有し、ルートディレクトリ、ディレクトリ、サブディレクトリ、及びファイルから成る。ファイルシステム内の各オブジェクト、すなわち、ファイル又はディレクトリは、オブジェクト識別子（ID）と、“Iノード”と呼ばれる特別なレコードとを有し、上記Iノードは、オブジェクトに関連付けられた属性、例えば、バイト単位のファイルサイズ、ブロック割り振り、所有者ユーザID、所有者のグループID、最後の修正の時間などを保持する。

【0005】

コンテンツアドレス可能ファイルシステムにおいて、オブジェクトIDは、ユーザ又はメタデータコンテンツ上で数学的関数を実行することによって作成される。伝統的なファイルシステムにおいて、オブジェクトIDは、典型的に、使用されていない識別子のリストから選ばれる。

30

【0006】

大抵のファイルシステムは、ファイルの最新バージョンのみ保有する。ユーザがより古いバージョンのファイル又はディレクトリを保有したい場合、ユーザはこれらを手動で保存し、あるいは特別なバージョンングソフトウェアを使用して異なるバージョンから成る複数のバージョンの記録をつけないければならない。データのバックアップが管理者によって定期的に行われることがあり、ユーザは、より古いバージョンのファイルを得るためには、管理者にコンタクトをとる必要がある。

40

【発明の概要】

【0007】

本発明の概要は、以下に詳細な説明においてさらに説明される概念のうち選択されたものを簡素化された形式において導入するために提供される。本発明の概要は、請求される対象事項の重要な特徴又は必須の特徴を識別するものではなく、請求される対象事項の範囲を決定することの助けとして用いられるべきものでもない。

【0008】

本明細書に説明されるテクノロジーは、コンピュータ記憶装置におけるファイルシステムの編成及び管理の方法である。ファイルシステムは、コンテンツアドレス可能分散オブ

50

ジェクトストアの上に構築される。

【0009】

一例示的な実施形態に従い、方法が、レコードをユーザコンテンツに関連付けることを含む。レコードは、ユーザコンテンツに関連付けられた属性を含むことができる。属性は、コンテンツアドレス可能オブジェクトストア内のオブジェクトに関連付けられたコンテンツハッシュを少なくとも含むことができる。方法は、ユーザコンテンツを複数のデータオブジェクトに分割することを可能にすることができる。各データオブジェクトは、所定の最小サイズを有する。方法は、複数のデータオブジェクトのうちの各データオブジェクトについて、それぞれのデータオブジェクトハッシュを生成することを含む。データオブジェクトハッシュは、データオブジェクトに関連付けられたそれぞれのデータオブジェクトコンテンツの関数である。方法は、データオブジェクトハッシュに基づいて、コンテンツハッシュを決定することをさらに含む。

10

【0010】

いくつかの実施形態において、コンテンツハッシュを決定することは、複数のデータオブジェクトの中のオブジェクトの数を算出することを含む。オブジェクトの数が1である場合、コンテンツハッシュはデータオブジェクトハッシュに設定される。オブジェクトの数が2以上である場合、データオブジェクトハッシュに基づいて、ポインタツリーが生成される。ポインタツリーの生成は、データオブジェクトハッシュを複数のポインタオブジェクトにグループ化することを含む。ポインタオブジェクトは、ポインタツリーにおけるレイヤのリーフとして割り当てられる。レイヤが2つ以上のリーフを含む場合、ポインタツリーの次レイヤが構築される。次レイヤの構築は、ポインタオブジェクトの各々について、ポインタハッシュを生成することを含む。ポインタハッシュは、ポインタオブジェクトに関連付けられたそれぞれのポインタオブジェクトコンテンツの関数である。次レイヤの構築は、ポインタハッシュを複数のさらなるポインタオブジェクトにグループ化することを含む。次レイヤの構築は、さらなるポインタオブジェクトをポインタツリーにおけるさらなるレイヤのリーフとして割り当てることを含む。ポインタハッシュの生成、ポインタハッシュのグループ化、及びさらなるポインタオブジェクトの割り当ては、1つのリーフを有する最終レイヤが受信されるまで繰り返される。最終レイヤのリーフは、ポインタツリーのルートとして割り当てられる。ルートハッシュが、ポインタツリーのルートのコンテンツに基づいて計算され、レコードのコンテンツハッシュは、ルートポインタハッシュに設定される。

20

30

【0011】

いくつかの実施形態において、ポインタツリーの深さは、下記の式： $\text{MaxFileSize} = \text{BLKSZ} \times (\text{truncate}(\text{BLKSZ} \div \text{SMASHSZ}))^{\text{depth}}$ によって決定される。MaxFileSizeは、コンテンツエンティティのサイズである。BLKSZは、所定の最小サイズである。SMASHSZは、コンテンツハッシュのサイズである。truncateは、実数の関数であり、実数の整数部分を返す。

【0012】

いくつかの実施形態において、ユーザコンテンツは、下記のうち1つ以上を含む：データファイル、及びフォルダ。

40

【0013】

いくつかの実施形態において、フォルダは、さらなるレコードに関連付けられた少なくとも1つのフォルダエントリを含む。上記さらなるレコードは、さらなるユーザコンテンツに関連付けられる。フォルダエントリは、下記のうち少なくとも1つを含む：さらなるユーザコンテンツのタイプに関連付けられた、エントリのタイプ、さらなるユーザコンテンツに関連付けられた名前、さらなるユーザコンテンツに関連付けられた名前の長さ、及びさらなるレコードに関連付けられた、エントリの属性。さらなるレコードハッシュが、さらなるレコードの属性に基づいて生成される。

【0014】

いくつかの実施形態において、フォルダは、さらなるレコードに関連付けられた1つ以

50

上のエントリを含み、さらなるレコードの各々は、ユーザコンテンツについての一意の時間バージョンに関連付けられる。

【0015】

いくつかの実施形態において、レコードの属性は、前のレコードハッシュを含む。前のレコードハッシュは、前のレコードの属性に関連付けられたコンテンツに基づいて生成される。前のレコードは、ユーザコンテンツのそれぞれの前のバージョンに関連付けられる。

【0016】

いくつかの実施形態において、上記方法は、ユーザコンテンツの変更に応答して、ユーザコンテンツに関連付けられたポインタツリーを少なくとも部分的に再生成することをさらに含む。方法は、さらなるレコードを生成することと、さらなるレコードのコンテンツハッシュを、上記の再生成されたポインタツリーのルートに関連付けられたポインタハッシュに設定することを含む。方法は、さらに、レコードの属性に関連付けられたコンテンツに基づいて、レコードコンテンツハッシュを計算することを含む。方法は、この新しいレコードについての前のレコードハッシュを、レコードコンテンツハッシュに設定することを含む。

【0017】

いくつかの実施形態において、方法は、さらに、データオブジェクトをクラスタストレージに書き込むことを含む。データオブジェクトの書き込みは、データオブジェクトとデータオブジェクトのタイプとを受信することを含む。データオブジェクトハッシュが計算される。データオブジェクトのコピーを記憶すべきクラスタストレージの物理ノードが決定される。上記の決定された物理ノードの各々において、データオブジェクトハッシュがコンテンツアドレス可能オブジェクトストア内でルックアップされて、データオブジェクトメタデータが取得される。メタデータは、少なくともディスク識別子とディスク内の場所とを含む。データオブジェクトハッシュが発見されない場合、データオブジェクトのコピーがノードに書き込まれ、コンテンツアドレス可能オブジェクトストアは、データオブジェクトに関連付けられた新しいメタデータで更新される。

【0018】

いくつかの実施形態において、方法は、クラスタストレージにファイルを書き込むことをさらに含む。ファイルの書き込みは、ファイルをファイルデータオブジェクト分けることと、ファイルデータオブジェクトのためのファイルデータオブジェクトハッシュを計算することを含む。ファイルポインタツリーが、ファイルポインタツリーのリーフに関連付けられたファイルポインタオブジェクトにファイルデータオブジェクトハッシュを挿入することによって、生成される。ファイルポインタオブジェクトがフルであるか、あるいはすべてのファイルデータオブジェクトが処理されている場合、ファイルポインタオブジェクトのためのファイルポインタハッシュが計算され、このファイルポインタハッシュは親ファイルポインタオブジェクトに挿入される。ファイルポインタオブジェクトのためのファイルポインタハッシュの計算は、ファイルポインタツリーが構築されるまで繰り返される。ファイルの書き込みは、さらに、ファイルポインタツリーのルートのファイルポインタハッシュをファイルレコードのコンテンツハッシュ属性に挿入することを含む。ゆえに、ファイルレコードは、ファイルに関連付けられる。ファイルレコードの属性は、少なくともファイルサイズとタイムスタンプとで更新される。レコードコンテンツハッシュが、ファイルレコードの属性に基づいて計算される。ファイルデータオブジェクト、ファイルポインタオブジェクト、及びファイルレコードは、クラスタストレージに書き込まれる。書き込みの成功及び失敗のうち一方の指標が返される。

【0019】

いくつかの実施形態において、コンテンツアドレス可能オブジェクトストア上のファイルシステムを編成し及び管理する方法が、クラスタストレージからデータオブジェクトを読み出すことを含む。データオブジェクトの読み出しは、データオブジェクトハッシュを受信することを含む。データオブジェクトの読み出しは、さらに、クラスタストレージの

10

20

30

40

50

物理ノードのセットを決定することを含む。物理ノードは、データオブジェクトのコピーを含む可能性の最大から可能性の最小へ順序付けられる。各々の順序付けられた物理ノードについて、コンテンツアドレス可能オブジェクトストアにおけるデータオブジェクトハッシュがルックアップされる。データオブジェクトハッシュが発見される場合及びとき、データオブジェクトの読み出しは、データオブジェクトハッシュに基づいて、データオブジェクトメタデータを受信することをさらに含む。オブジェクトメタデータは、ディスク識別子とディスク内の場所とを含む。それから、データオブジェクトのコピーが、データオブジェクトメタデータを用いてノードから読み出される。

【0020】

いくつかの実施形態において、方法は、クラスタストレージからファイルを読み出すことをさらに含むことができる。ファイルの読み出しは、ファイルに関連付けられたファイルレコードのレコードコンテンツハッシュを受信することを含む。ファイルレコードは、読み出し処理を適用することによって、クラスタストレージから読み出される。上記処理は、ファイルレコードのコンテンツハッシュを取り出すことによって、ファイルポインタツリーのルートに関連付けられたルートポインタハッシュを決定することを含む。上記読み出しは、ファイルポインタツリーをルートから始めて下降して、ファイルポインタツリーの子リーフの中のファイルポインタハッシュを取り出すことを含む。ファイルポインタハッシュに関連付けられたファイルデータオブジェクトが、上記読み出し処理を適用することによって読み出されることができる。ファイルデータオブジェクトは、順次ファイルデータへと再組立てされる。クラスタストレージからファイルを読み出すことは、上記順次ファイルデータを返すことで完結する。

【0021】

本開示の別の例示的な実施形態に従い、コンテンツアドレス可能オブジェクトストア上のファイルシステムを編成し及び管理する方法のステップが、命令を含むマシン可読媒体上に記憶され、上記命令は、1つ以上のプロセッサにより実施されると、列挙されたステップを実行する。

【0022】

本願方法の他の特徴には、実行中の (on the fly) すべてのファイル及びディレクトリについてのバージョンの追跡の履歴を提供することが含まれる。ゆえに、より古いバージョンのファイル及びディレクトリが、ルートディレクトリを含め、任意のときにユーザに対して利用可能である。

【図面の簡単な説明】

【0023】

実施形態は、添付図面の図において限定ではなく例として例示され、同様の参照は類似の要素を示す。

【図1】ファイルシステムにおけるコンテンツの一例を示す。

【図2】所与のファイルのデータオブジェクトのためのスマートハッシュの生成を例示する。

【図3】ファイルシステム内のIノードエントリを説明する属性のテーブルを表示する。

【図4】Ichdataパラメータの属性の集合を表示する。

【図5】ファイルデータを直接指し示すIノードを例示する。

【図6】ファイル又はディレクトリを表現する1レイヤポインタツリーを例示する。

【図7A】ファイル又はディレクトリを表現する2レイヤポインタツリーを例示する。

【図7B】ファイル又はディレクトリを表現する3レイヤポインタツリーを例示する。

【図8】ファイルのバージョンに関連付けられた、リンクされたIノードのチェーンを例示する。

【図9】ディレクトリデータオブジェクトの構造、及び、Iノードと対応するファイルデータオブジェクトとに対するディレクトリエントリの接続を例示する。

【図10】タイプ1のディレクトリエンティティを有するディレクトリデータの一例を示す。

10

20

30

40

50

【図 1 1】より古いバージョンのファイル又はディレクトリに対するアクセスをユーザが有することを可能にするメカニズム及びスナップショットのための仮想ディレクトリのスキームを例示する。

【図 1 2】オブジェクトを書き込む方法を示す処理フロー図である。

【図 1 3】オブジェクトを読み出す方法を示す処理フロー図である。

【図 1 4】ファイルを書き込む方法を示す処理フロー図である。

【図 1 5】ファイルを読み出す方法を示す処理フロー図である。

【図 1 6】コンピュータシステムの例示的な電子形態におけるマシンのコンピューティング装置の概略表現を示し、これにおいて、本明細書に論じられる方法論のうち任意の 1 つ以上をマシンに実行させる命令セットが実行されることができる。

10

【発明を実施するための形態】

【0024】

下記の詳細な説明は添付図面に対する参照を含み、添付図面は本詳細な説明の一部を形成する。図面は、例示的な実施形態に従い例示を示す。こうした例示的な実施形態は、本明細書において「例」とさらに呼ばれ、十分詳細に説明されて、当業者が本対象事項を実施することを可能にする。実施形態は組み合わせられてもよく、他の実施形態が利用されてもよく、あるいは、構造的、論理的、及び電氣的な変更が、請求されるものの範囲から逸脱することなく行われてもよい。ゆえに、下記の詳細な説明は、限定的な意味においてとられるべきではなく、その範囲は、別記の請求項及びその均等物によって定義される。

【0025】

20

本明細書に説明されるテクノロジーは、コンテンツアドレス可能分散オブジェクトストア (content addressable distributed object store) の上に築かれるファイルシステムの編成及び管理を可能にする。ファイルシステム内の各可視オブジェクトが、オブジェクト ID のチェーンに関連付けられることができ、上記オブジェクト ID の各々は、所与の時点におけるオブジェクトのバージョンに対応する。ゆえに、オブジェクトストアは、様々な時点におけるファイルシステムの「スナップショット」を提供すると言われ得る。ファイルシステムは、すべてのファイルのすべてのバージョンを保有し、ガーベッジコレクション処理を実行して、もはや必要とされないバージョンを廃棄することができる。

【0026】

一例示的な実施形態に従い、上記方法は、レコード（さらに、i ノード (inode) オブジェクトとして参照される）をユーザコンテンツに関連付けることを含む。レコードは、ユーザコンテンツに関連付けられた属性を含むことができる。属性は、コンテンツアドレス可能オブジェクトストア (content addressable object store) 内のオブジェクトに関連付けられたコンテンツハッシュを少なくとも含むことができる。この例示的な方法は、ユーザコンテンツを複数のデータオブジェクトに分割することを可能にすることができる。各データオブジェクトは、所定の最小サイズを有する。方法は、複数のデータオブジェクトの各データオブジェクトについて、それぞれのデータオブジェクトハッシュを生成することを含むことができる。データオブジェクトハッシュは、対応するデータオブジェクトに関連付けられたそれぞれのデータオブジェクトコンテンツの関数である。方法は、データオブジェクトハッシュに基づいて、コンテンツハッシュを決定することを含むこ

30

40

【0027】

ファイルシステム 100 の一例が図 1 に示されている。ファイルシステムは階層ツリーとして見られ得る。ツリーのルートノードは、ルートディレクトリ 110、すなわち、ファイルシステムのルートフォルダ、いかなる他のフォルダの子でもない唯一のフォルダに対応する。所与のノードである子ノード 120、130、140、150、160 はファイル又はフォルダであり得る。ファイルシステムは P O S I X セマンティック (semantic) を利用することができる。

【0028】

すべてのオブジェクト ID がオブジェクトの内容に基づいて計算される。オブジェクト

50

内容が変わる場合、オブジェクト I D がさらに変わり、新しいオブジェクト I D を有し得る。オブジェクトの変更は、さらに、その親のオブジェクト I D の変更を引き起こし得る。上記変更は、ファイルシステム 1 0 0 を表現するツリーのルートまでずっと伝播する。こうして、一意のオブジェクト I D がファイルシステム 1 0 0 の各バージョンについて生成されることができる。

【 0 0 2 9 】

各オブジェクト I D のキー属性はタイプ Smash のパラメータであり、Smash はスマートハッシュ (smart hash) を表す。スマートハッシュは、さらに、オブジェクト識別子として参照される。図 2 は、ファイルの Smash オブジェクト I D を算出する一例示的な表現 2 0 0 を示している。

10

【 0 0 3 0 】

ファイル 2 1 0 は、BLKSZ サイズのオブジェクト 2 2 0 に分けられる。これらオブジェクト 2 2 0 の各々は、Smash タイプのパラメータを返すハッシュ関数のための入力オブジェクトである。

【 0 0 3 1 】

ハッシュ関数はいくつかの要件を有し得る。この関数は、あらゆるとり得る入力オブジェクトに対して出力される一意の Smash を生成しなければならない。2 つの入力オブジェクトがちょうど 1 ビットだけ異なる場合、ハッシュ関数は、高い確率で、2 つの異なる Smash 値を生成しなければならない。

20

【 0 0 3 2 】

ハッシュ関数は、常に、同じ長さのバイトのものである Smash 2 3 0 を生成する。Smash 2 3 0 の長さは、SMASHSZ として知られている。データオブジェクトの最大長は、BLKSZ として知られている。これらは、ファイルシステムのための所定のパラメータである。例えば、長さ 2 0 バイトの複数の Smash と 3 2 K B の BLKSZ 値とが使用されることができる。3 2 K B 長の 2 つの異なる入力オブジェクトが長さ 2 0 バイトの同じ Smash 値を結果としてもたらし得る確率は、非常に小さくなければならない。

【 0 0 3 3 】

ハッシュ関数の出力は、ハッシュ空間において公平に均等に分散されるべきである。ハッシュ関数は、毎回同じ入力を与えられるたび、同じ出力を生成しなければならない。

30

【 0 0 3 4 】

zzhash と呼ばれる特別な Smash 値がある。この値は、存在しない何かを表現することによって使用される。

【 0 0 3 5 】

ファイルシステム内の、ユーザに対して可視の各オブジェクトについて、I ノード (Inode) と呼ばれる少なくとも 1 つのエンティティが存在する。I ノードは、ファイルシステム内のファイル又はディレクトリを表現するのに必要とされるすべての情報を保持する。図 3 は、I ノード 3 0 0 の構造を示している。I ノードの属性 3 1 0 の多くが、P O S I X 標準により定義される通常のパラメータ、例えば、所有者 I D、グループ I D、バイト単位でのファイルのサイズなどである。いくつかのカスタム属性が追加されることができ、例えば、chash、xhash、及び ichdata などである。

40

【 0 0 3 6 】

chash は、コンテンツハッシュを表す。chash は Smash タイプのパラメータであり、ディレクトリデータオブジェクト、ディレクトリポインタオブジェクト、データオブジェクト、ポインタツリーオブジェクト、又は zzhash の Smash を含み得る。

【 0 0 3 7 】

xhash は、拡張された属性ハッシュである。このハッシュは、拡張された属性データオブジェクト、拡張された属性ポインタオブジェクト、又は zzhash の Smash パラメータを含み得る。

【 0 0 3 8 】

意図的チェーン化ヒストリデータ (Intentional Chained History Data) (ichdata

50

）400の構造が図4に示されている。これは、データストア内の変えられない（immutable）オブジェクトの中から変えられる（mutable）オブジェクトの履歴を作成すること及び追跡することについて責任を負うメカニズムにおいて利用される属性410のセットである。このパラメータのセットは、generation、revgen、snaptime、及びsnaptagを含む。generationは、スナップショット生成カウントであり、revgenは、スナップショット内における改訂（revision）カウントであり、snaptimeは、スナップショットが作成された時間であり、snaptagは、いずれのファイルシステム動作がスナップショットを引き起こしたかを示す列挙値であり、例えば、作成、削除、ファイルに対する書き込み、ファイルの名前変更などである。

【0039】

10

ichdata400は、Smashタイプの3つのパラメータ420をさらに含み得る。

a. ohashは、古いハッシュ（old hash）を表し、このIノードの前のバージョンのためのSmashを保持する。ohashは、このオブジェクトの前のバージョンがない場合、zzhashに等しい。

b. Mhash又はマージハッシュ（merge hash）は、このノードが由来する代替の履歴を表現する。Mhashは、こうした代替の履歴がない場合、zzhashに等しくなり得る。

c. Uuid又はユニバーサル一意識別子（Universally Unique Identifier）は、このオブジェクトのすべてのバージョンによって、その履歴の全体を通して継承されるパラメータである。uuidパラメータは、この特定のファイル又はディレクトリのためのIノードが最初に作成されるとき、生成される。

20

【0040】

Iノードは、ただ1つのchashパラメータを有する。ファイルサイズがBLKSZの予め定義されたシステムパラメータより小さいか又は等しいとき、必要なことは、ハッシュ関数をファイルに単に適用して、対応するSmashを生成することである。しかしながら、ファイルシステム内のファイルの大多数と、同様にディレクトリとは、BLKSZより大きく成長する可能性がある。ファイル又はディレクトリのサイズがBLKSZより大きいとき、ファイル又はディレクトリはポインタツリーを介して表現される。

【0041】

図5は、Iノード300についての表現500を示しており、Iノード300は、データオブジェクト520として示されるBLKSZより小さいか又は等しいサイズのファイルを指し示している。Iノード内のchashパラメータは、このファイルについて算出されたSmashである。

30

【0042】

図6は、BLKSZより大きいサイズのファイルに対応するIノード610と、上記ファイルを表現する対応した1レイヤポインタツリーとの表現600を示している。Iノード300内のchashパラメータは、今度はSmash640のレイヤ1を含むオブジェクトのSmashである。レイヤ1のSmash640は、ファイルデータ520のオブジェクトについて算出され、ファイルデータ520は、ポインタツリーにおけるリーフとして見られ得る。各データオブジェクト520は、サイズBLKSZのものである。最後のデータオブジェクト520は、サイズBLKSZか又はより小さいものであり得る。オブジェクトハッシュを含むオブジェクトが、ポインタオブジェクト620として参照される。

40

【0043】

各ポインタオブジェクト620は、ゆえに、最大で、 $N = \text{truncate}(\text{BLKSZ} \div \text{SMASHSZ})$ ハッシュまで保持し得る。例えば、BLKSZが32KBであり、SMASHSZが20バイトであるとき、1レイヤのSmashを有するポインタツリーは、サイズにおいて最大53,673,984バイトまでのファイルを表現することができる。

【0044】

図7Aは、2レイヤポインタツリー700を例示している。レイヤ1のポインタオブジェクト620は、他のポインタオブジェクトのSmashを保持する。第2のレイヤの各ポインタオブジェクトは、データオブジェクト520のためのSmash640を含む。例えば、B

50

LKSZが32KBであり、SMASHSZが20バイトであるとき、2レイヤポインタツリーは、サイズにおいて最大約83GBまでのファイルを表示することができる。

【0045】

図7Bは、3レイヤポインタツリー750を示している。図7Aの2レイヤポインタツリー700とは対照的に、3レイヤポインタツリー750は、さらなるレイヤのポインタオブジェクト620を含む。例えば、BLKSZが32KBであり、SMASHSZが20バイトであるとき、3レイヤポインタツリー750は、サイズにおいて最大約134TBまでのファイルに対処することができる。

【0046】

ポインタオブジェクトツリーの任意の与えられた深さに対するファイルの最大サイズは、下記のとおりに算出されることができる： $\text{MaxFileSize} = \text{BLKSZ} \times (\text{truncate}(\text{BLKSZ} \div \text{SMASHSZ}))^{\text{depth}}$ 。

【0047】

ポインタツリーを用いたデータの表現は、ファイルデータからSmashを通してIノードの後ろのレイヤに変更を迅速に伝播することに役立つ。ポインタツリーの同じメカニズムが、大きいディレクトリを表示することに使用されることができる。

【0048】

図8は、接続されたIノード300のリスト800を例示しており、Iノード300は、ファイル820として示される1つのエンティティ、ファイル、又はディレクトリを説明している。Iノード300は、エンティティに対する修正が行われた時点T0、T1、T2などに対応する。新しいIノードが、対応するコンテンツが毎回修正されるたび、作成され、チェーンの上部に追加される。新しいIノードのgeneration属性850は、1ずつ増加し得る。

【0049】

各Iノード300のchashパラメータ830は、所与の時点におけるエンティティ又はエンティティのポインタツリーについてのSmashを含む。ohashパラメータは、前の時点に対応するIノード300のSmashである。第1の時点のIノード300のohashパラメータは、このエンティティについて前のバージョンがないので、zzhashに向けられる。ohashパラメータ840は、所与のエンティティの「スナップショット」の時間逆行チェーン（back in time chain）を構築する、所与のエンティティのIノード300をリンクしている。チェーンの中の各Iノード300のchashパラメータ830は、そのバージョンのデータを指し示す。このタイプの接続は、データの異なるバージョンを通じた検索を可能にする。ガーベッジコレクタが、このチェーンを歩いて行く（walk down）ように設定され、上記バージョンのうちいずれが削除されることができるかを定めることができる。ガーベッジコレクションを実行する方法の一例が、2013年6月12日に申請され“Hybrid Garbage Collection”と題された米国仮特許出願第61/834,097号に説明されており、該出願は本明細書において参照により援用される。

【0050】

図9は、ディレクトリ910の内容の一例示的な表現900を示している。ディレクトリデータは、ディレクトリエントリ920のセットから成る。ディレクトリエントリ920は、下記の属性を含む可変サイズのレコードである：エントリのタイプ、エントリの名前の長さ、エントリの名前、及びタイプにより指定される属性。ディレクトリエントリのタイプは、少なくとも2つのとり得る値：0及び1を有することができる。

【0051】

タイプ0のディレクトリエントリは、Iノード300を識別する。ディレクトリエントリ920の中の名前フィールドは、Iノードが表現するファイル940又はディレクトリの、名前を保持する。ディレクトリエントリの中の指定された属性が、Iノード300のSmashである。

【0052】

タイプ1のディレクトリエンティティについて、名前ファイルは、仮想スナップショッ

10

20

30

40

50

ツリーのルートを識別する。ディレクトリエントリの特定属性は、この場合には使用されない。ユーザは、仮想スナップショットツリーの名前をサブディレクトリとして見ることができる。図10は、タイプ1のディレクトリエンティティ1010を有するディレクトリデータ1000の一例を示している。

【0053】

仮想ディレクトリが、ユーザがファイルシステムのスナップショットを調べたいとき、生成される。図11は、仮想ディレクトリの構造1100を示している。仮想ディレクトリの仮想ディレクトリデータ1110は、タイプ0のディレクトリエントリ1120を含むことができ、名前1130は、スナップショットの作成に関連付けられたタイムスタンプによって表現されることができる。ディレクトリエントリのハッシュ属性1140は、タイムスタンプに対応するIノード1150のSmashを含むことができる。上記Iノード1150を通して、ユーザは、対応するより古いバージョンに対するアクセスを有することができる。ファイルシステムのスナップショットを調べる方法のさらなる詳細は、2014年10月10日に申請され“Hierarchical Data Archiving”と題された米国特許出願第14/512,299号に説明されており、該出願は本明細書において参照により援用される。

【0054】

図12は、ストレージクラスタにオブジェクトを書き込む方法1200を示す「ライトオブジェクト」処理フロー図である。方法1200は、ステップ1210において、書き込むべきオブジェクト及びオブジェクトタイプを受信することから始まる。ステップ1220において、オブジェクトのSmashが計算される。ステップ1230において、オブジェクトのコピーを有するべき物理ノードが、一貫したハッシングアルゴリズム(Consistent Hashing Algorithm)を適用することによって発見される。一貫したハッシングアルゴリズムは、2012年4月6日に申請され“Consistent Ring Namespaces Facilitating Data Storage and Organization in Network Infrastructures”と題された米国特許出願第13/441,592号において説明されており、該出願は本明細書において参照により援用される。ステップ1240において、オブジェクト及びSmashが、前のステップ1230において決定された各物理ノードに対して送信される。各ノードは、そのハッシュデータベース(HDB)内でSmashのルックアップを実行する。ルックアップは、2012年4月6日に申請され“Cryptographic Hash Database”と題された米国特許出願第13/441,715号において説明される方法を用いて実行されてもよく、該出願は本明細書において参照により援用される。ルックアップはオブジェクトメタデータを返し、オブジェクトメタデータは、このSmashを有するオブジェクトについてのディスク識別子とディスク上の場所とを含む。ステップ1250において、ルックアップが成功する場合、オブジェクトが既に存在するので、オブジェクトの新しいコピーは書き込まれない。ルックアップが失敗する場合、ステップ1260において、オブジェクトの新しいコピーが、ステップ1230において一貫したハッシングアルゴリズムにより決定されたとおりのクラスタの中のノード及びディスクに書き込まれる。オブジェクトをディスクに書き込む方法は、2013年10月16日に申請され“Forward-Only Page Data Storage Management”と題された米国特許出願第14/055,662号において説明されており、該出願は本明細書において参照により援用される。ステップ1270において、オブジェクトを書き込んだ各ノードは、そのHDBを、オブジェクトの新しい属性を用いて更新する。HDBを更新する方法は、米国特許出願第13/441,715号(完全な引用文は上記)において説明されている。ステップ1280において、成功又は失敗の指標がライター(writer)に返される。

【0055】

図13は、一例示的な実施形態に従う、ストレージクラスタからオブジェクトを読み出す方法1300を示す「リードオブジェクト」処理フロー図である。方法1300は、ステップ1310において、オブジェクトのSmashを受信することから始まる。ステップ1320において、2012年4月6日に申請され“Consistent Ring Namespaces Facilitating Data Storage and Organization in Network Infrastructures”と題された、本明細書において参照により援用される米国特許出願第13/441,592号に説明されたとおり

の一貫したハッシングアルゴリズムが適用されて、オブジェクトのコピーを保持する可能性の最大から可能性の最小にソートされたノードのセットを発見する。ステップ1330において、各ノードにおいて、オブジェクトのSmashが、ノードに属するHDB内でルックアップされる。HDBを検索する方法は、米国特許出願第13/441,715号（完全な引用文は上記）において説明されている。Smashのレコードが発見される場合、方法1300のステップ1340に進む。Smashが発見されない場合、ステップ1330におけるルックアップが、次に最も可能性のあるノード上で繰り返される。オブジェクトを発見することなく、すべてのノードが検索された場合、「発見されない（not found）」の指標が要求元に返される。ステップ1340において、オブジェクトは、ステップ1330におけるルックアップにより受信されるデータから取得されるディスク識別子とディスク上の場所とを用いて、ディスクから読み出される。ディスクからオブジェクトを読み出す方法は、米国特許出願第14/055,662号（完全な引用文は上記）において説明されている。ステップ1350において、オブジェクトデータ及びメタデータが要求元に返される。このアルゴリズムは、1つ以上のノード上で同時的に実行され得ることが留意されるべきである。

【0056】

図14は、ストレージクラスタにファイルを書き込む一例示的な方法1400を示す処理フロー図である。方法1400は、ステップ1410において、ライタからファイルデータを受信することから始まる。ステップ1415において、ファイルデータはデータオブジェクトに分けられ、各データオブジェクトは最後のチャンクを除いてサイズBLKSZのものであり、最後のチャンクはBLKSZより小さい可能性がある。ステップ1420において、Smashが各データオブジェクトについて計算される。ステップ1425において、ポインタツリーが、図5、図6、図7A、及び図7Bに示されるとおり、データオブジェクトSmashをリーフポインタノードに挿入することによって構築される。ステップ1430において、ポインタオブジェクトがフル（full）であるか、あるいはすべてのデータオブジェクトが処理されている場合、ポインタオブジェクトのSmashが計算され、親ポインタオブジェクトに挿入される。ステップ1435において、ポインタオブジェクトの完全なツリーが構築されるまでステップ1430が繰り返される。ステップ1440において、ポインタオブジェクトツリーのルートのSmashが、Iノードオブジェクトのchashフィールドに挿入される。ステップ1445において、他のIノード属性、例えばファイルサイズ、タイムスタンプなどが、更新される。ステップ1450において、IノードオブジェクトのSmashが計算される。ステップ1455において、本処理により作成され又は修正された各オブジェクトについて、図12に説明されるとおりの「ライトオブジェクト」処理が呼び出される。最後、ステップ1460において、成功又は失敗がライタに返される。

【0057】

図15は、ストレージクラスタからファイルを読み出す方法1500を示す処理フロー図である。方法1500は、ステップ1510において、リーダ（reader）からIノードオブジェクトのSmashを受信することから始まる。ステップ1520において、図13に説明される「リードオブジェクト」処理が使用されて、Iノードオブジェクトを読み出す。ステップ1530において、ポインタツリーのルートが、Iノードオブジェクトのchash属性からSmashを取り出すことによって発見される。ステップ1540において、ポインタツリーが、chashから始め、子のSmashを取り出すことと図13に説明される「リードオブジェクト」処理を適用してオブジェクトを読み出すこととによって、リーフノードへ下降される。ステップ1550において、データオブジェクトが、順次（sequential）ファイルデータへと再組立てされる。ステップ1560において、ファイルデータがリーダに返される。

【0058】

図16は、コンピュータシステム1600の例示的な電子形態におけるマシンのコンピューティング装置の概略表現を示しており、これにおいて、本明細書に論じられる方法論のうち任意の1つ以上をマシンに実行させる命令セットが実行されることができ、様々な例示的な実施形態において、マシンはスタンドアロン装置として動作し、あるいは他の

マシンに接続され（例えば、ネットワーク化され）てもよい。ネットワーク化された展開において、マシンは、サーバクライアントネットワーク環境の中でサーバ又はクライアントマシンの能力において、又は、ピアツーピア（又は、分散型）ネットワーク環境においてピアマシンとして動作し得る。マシンは、サーバ、パーソナルコンピュータ（PC）、タブレットPC、セットトップボックス（STB）、PDA、セルラー電話、デジタルカメラ、ポータブル音楽プレーヤ（例えば、MP3（Moving Picture Experts Group Audio Layer 3）プレーヤなどのポータブルハードドライブオーディオ装置）、ウェブ家電製品、ネットワークルータ、スイッチ、ブリッジ、又は、マシンにより行われるべきアクションを指定する命令セット（順次の、又はその他のの）を実行することができる任意のマシンであり得る。さらに、単一のマシンだけが例示されているが、用語「マシン」は、さらに、命令の一セット（又は、複数のセット）を個々に又は共同で実行して本明細書に論じられる方法論のうち任意の1つ以上を遂行するマシンの任意の集合を含むものとみなされるべきである。

10

【0059】

例示的なコンピュータシステム1600は、一のプロセッサ又は複数のプロセッサ1602、ハードディスクドライブ1604、メインメモリ1606、及びスタティックメモリ1608を含み、これらは、バス1610を介して互いに通信する。コンピュータシステム1600は、ネットワークインターフェース装置1612をさらに含み得る。ハードディスクドライブ1604は、コンピュータ可読媒体1620を含み、コンピュータ可読媒体1620は、本明細書に説明される方法論又は機能のうち任意の1つ以上を具現化し又は該1つ以上により利用される1つ以上の命令セット1622を記憶する。命令1622は、さらに、コンピュータシステム1600による命令1622の実行の間、メインメモリ1606内及び／又はプロセッサ1602内に、完全に又は少なくとも部分的に存在し得る。メインメモリ1606及びプロセッサ1602もまた、マシン可読媒体を構成する。

20

【0060】

コンピュータ可読媒体1620が、一例示的な実施形態において単一の媒体であるように図示されているが、用語「コンピュータ可読媒体」は、1つ以上の命令セットを記憶する単一の媒体又は複数の媒体（例えば、集中型又は分散型のデータベース及び／又は関連付けられたキャッシュ及びサーバ）を含むようにみなされるべきである。用語「コンピュータ可読媒体」は、さらに、マシンによる実行のための命令セットを記憶し、エンコードし、又は運ぶことができ、本出願の方法論のうち任意の1つ以上をマシンに実行させ、あるいは上記の命令セットにより利用され又はこれに関連付けられたデータ構造を記憶し、エンコードし、又は運ぶことができる任意の媒体を含むようにみなされるべきである。用語「コンピュータ可読媒体」は、したがって、これらに限られないが、ソリッドステートメモリ並びに光及び磁気媒体を含むようにみなされるべきである。こうした媒体には、さらに、限定でなく、ハードディスク、フロッピーディスク、NAND又はNORフラッシュメモリ、デジタルビデオディスク、RAM、ROMなどが含まれ得る。

30

【0061】

本明細書に説明される例示的な実施形態は、コンピュータ上にインストールされるコンピュータ実行可能命令（例えば、ソフトウェア）を含む動作環境において、ハードウェアにおいて、又はソフトウェアとハードウェアとの組み合わせにおいて実装され得る。コンピュータ実行可能命令は、コンピュータプログラミング言語において書かれてもよく、あるいはファームウェアロジックにおいて具現化されてもよい。認識されている標準に合うプログラミング言語において書かれる場合、上記の命令は、様々なハードウェアプラットフォーム上で、及び、様々なオペレーティングシステムに対するインターフェースのために実行されることができる。これらに限定されないが、本願方法を実施するコンピュータソフトウェアプログラムは、任意数の適切なプログラミング言語、例えば、C、Python、JavaScript（登録商標）、Go、又は、他のコンパイラ、アセンブラ、インタプリタ、若しくは他のコンピュータの言語若しくはプラットフォームにおいて書か

40

50

【図 3】

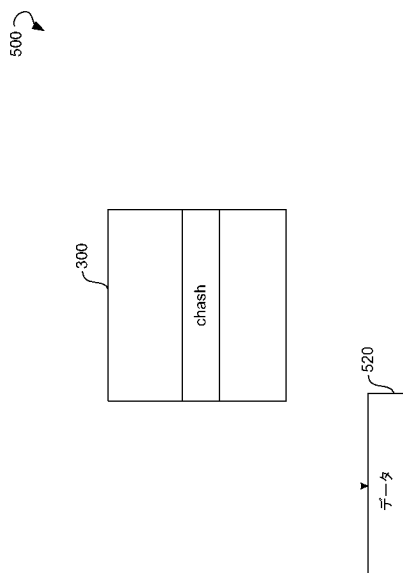
タイプ	名前	説明
ichdata	ichdata	図4を参照
u64	ino	iノード番号
u64	size	バイト単位でのファイルサイズ
u32	blksize	ブロック割り振り
u32	nlink	ハードリンクの数
u32	uid	所有者ユーザid
u32	gid	所有者グループid
u64	atimesec	最後のアクセスの時間（秒部分）
u64	mtimesec	最後のデータ修正の時間（秒部分）
u64	ctimesec	最後のファイルステータス変更の時間（秒部分）
u32	atimensec	最後のアクセスの時間（ナノ秒部分）
u32	mtimensec	最後のデータ修正の時間（ナノ秒部分）
u32	ctimensec	最後のファイルステータス変更の時間（ナノ秒部分）
u32	mode	タイプ及びパーミッション
u32	rdev	装置タイプ
Smash	chash	コンテンツハッシュ - このiノードが説明するコンテンツのsmash
Smash	chash	拡張された属性ハッシュ - 拡張された属性オブジェクトのsmash

【図 4】

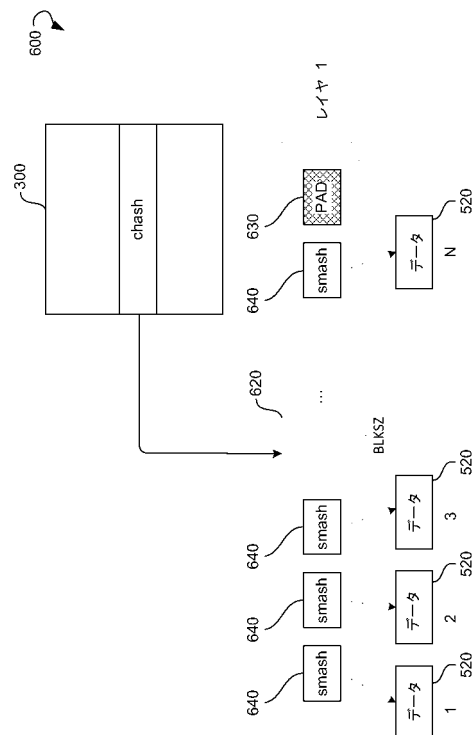
タイプ	名前	説明
u64	generation	スナップショット生成カウント（オリジナルバージョンに対する）
u64	revgen	改訂のスナップショット（履歴の間引き(thinning)のために書き込みされたとき）
u64	snaptime	UTCからの、64ビットNSとしての、このバージョンのこの生成の開始時間
u32	snaptag	このスナップショットを作成した動作を識別する列挙値
Smash	ohash	old hash - このノードの前のバージョンのsmash。前のバージョンが無い場合、zzhash
Smash	mhash	merge hash - このiノードが由来する代替の履歴（間引かれた履歴を含む）
Smash	uuid	ユニバーサル一意識別子 - 履歴を通してこのオブジェクトのすべてのバージョンによって継承される

POSIX属性
310

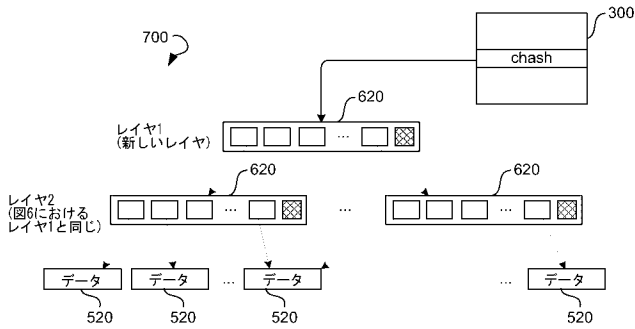
【図 5】



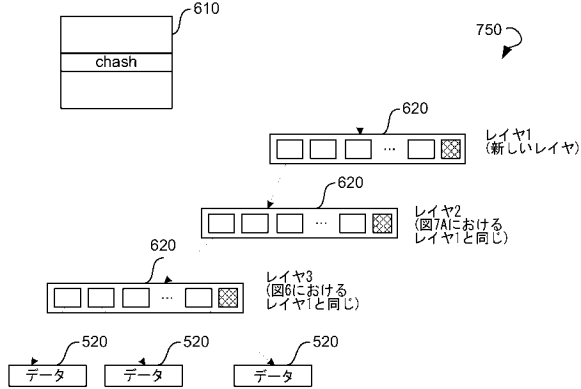
【図 6】



【図 7 A】



【図 7 B】



【図 8】

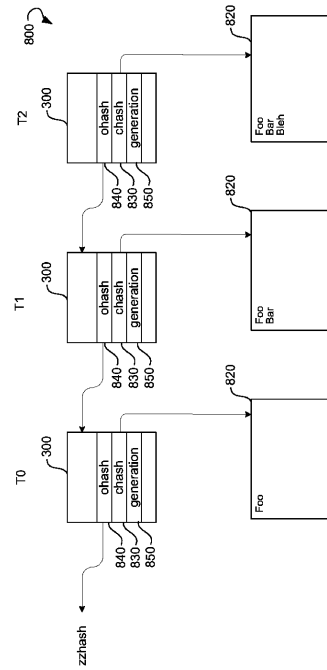
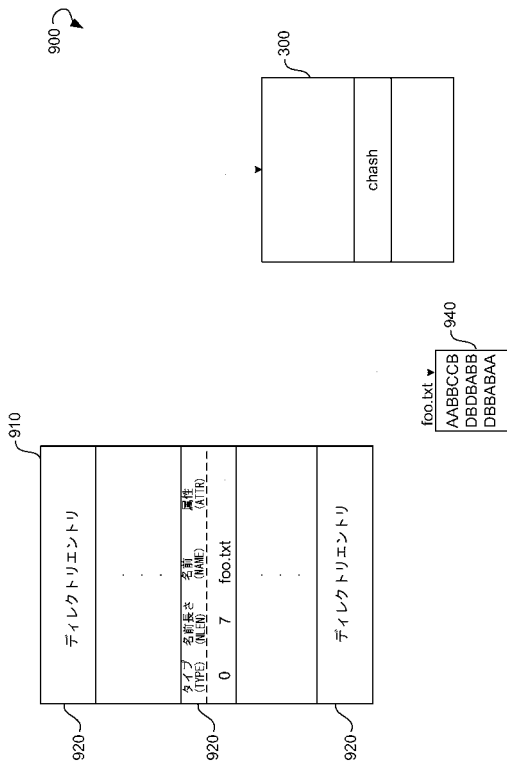
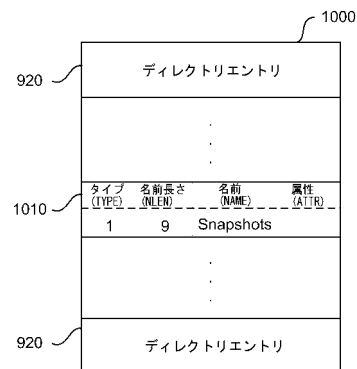


FIG. 8

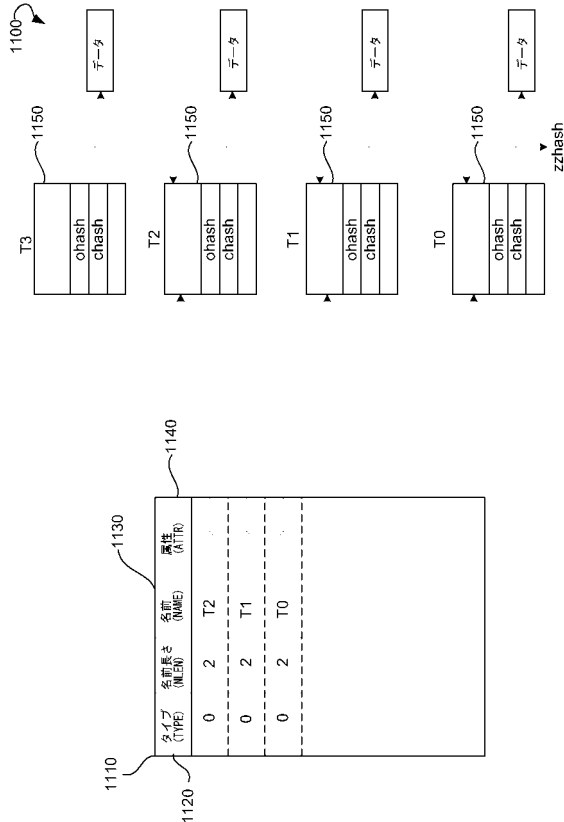
【図 9】



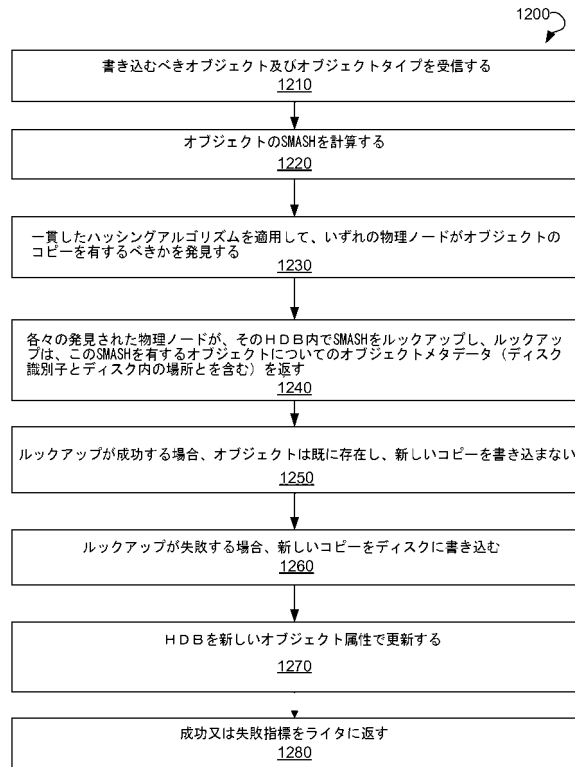
【図 10】



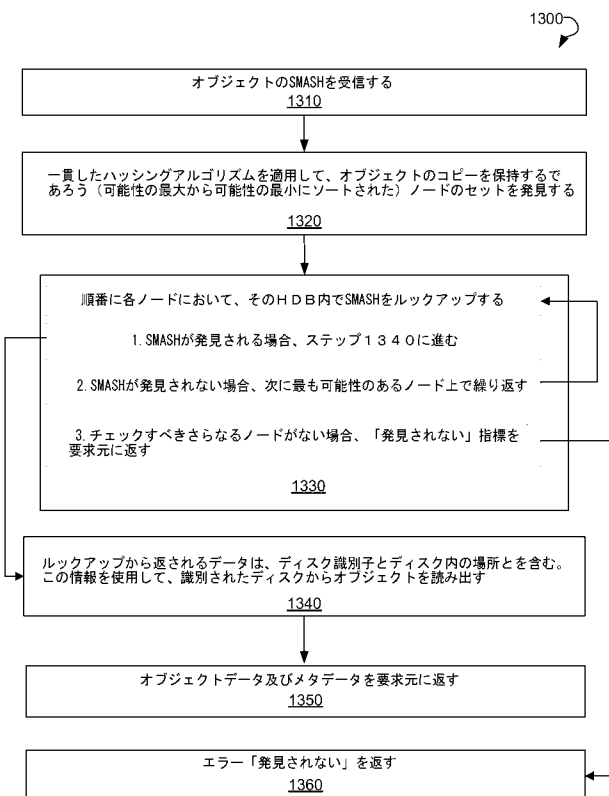
【図 1 1】



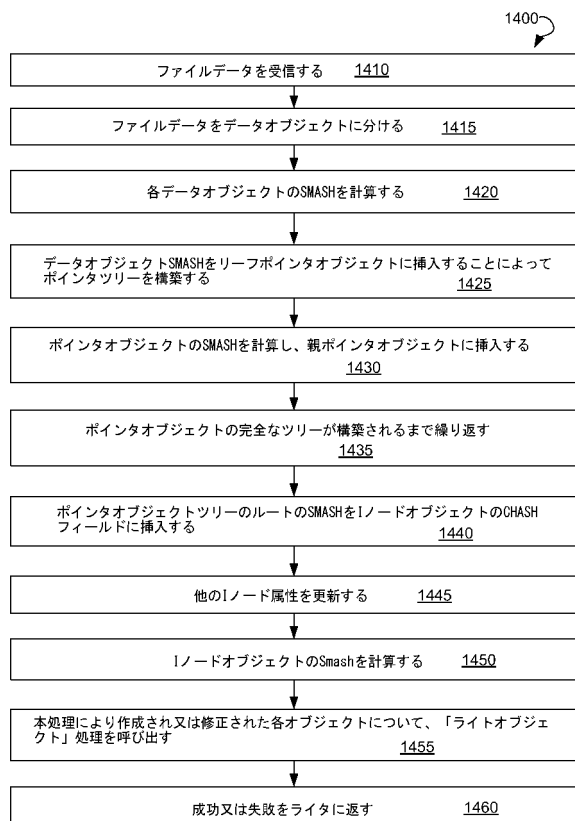
【図 1 2】



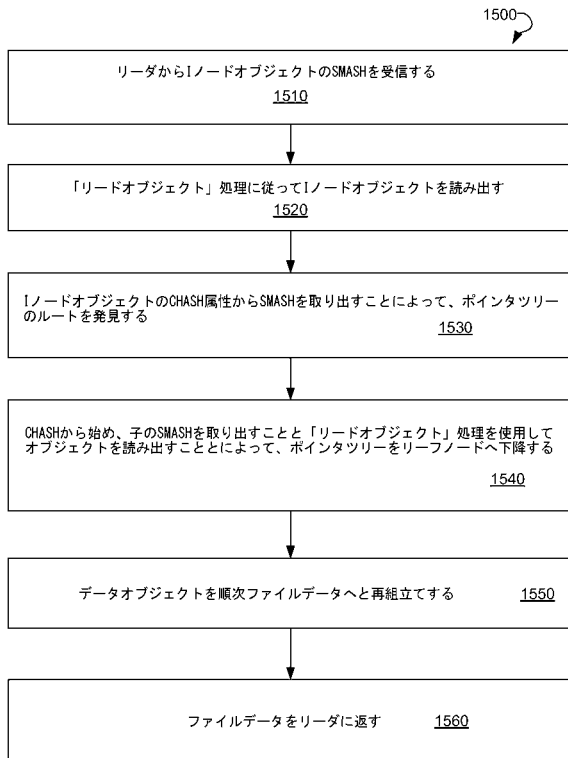
【図 1 3】



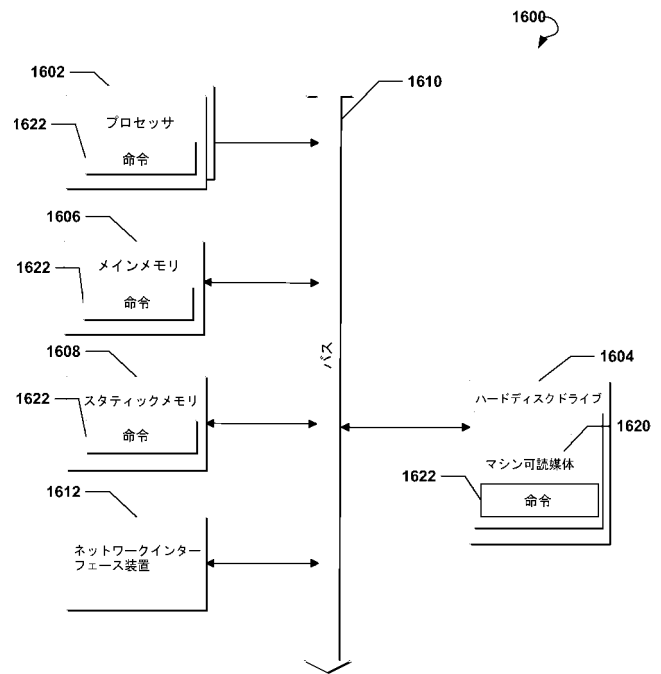
【図 1 4】



【図 15】



【図 16】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

15/014492 15.05.2015

International application No.

PCT/US 15/14492

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06F 17/00 (2015.01)

CPC - G06F17/30949, G06F17/30067, G06F17/30516, G06F17/30864, G06F17/30985

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC(8): G06F 17/00 (2015.01); CPC: G06F17/30949, G06F17/30067, G06F17/30516, G06F17/30864, G06F17/30985

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 707/747, 707/791, 707/802Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PatBase, ProQuest Dialog, Google Web, Google Patents (Search terms: Merkle tree, hash-based directed acyclic graph, HDAG, divide content, floor, integer, truncate, round, size, depth, file, document, content, folder, ratio, length, minimum, level, number, pieces, parts, group, block, key, compression factor, scaling factor, hierarchy, etc.)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X — Y	US 2007/0276843 A1 (Lillibridge et al.) 29 November 2007 (29.11.2007), para. [0020]-[0029], [0039], [0057]-[0058], [0077], [0089]-[0090], and [0094]-[0095], and Figs. 1 and 16, and claims 21-22.	1-2, 4-7, 13-14, 16-19, 25 3, 8-12, 15, 20-24
Y	US 2010/0114905 A1 (Slavik et al.) 06 May 2010 (06.05.2010), para. [0004], [0013]-[0014], [0022], [0050], and [0064].	3, 15
Y	US 7,725,437 B2 (Kirshenbaum et al.) 25 May 2010 (25.05.2010), col. 2, ln. 19-37 and 49-57, col. 3, ln. 4-9 and 15-24, col. 4, ln. 47-56, col. 5, ln. 1-6, 22-26, and 45-65, col. 10, ln. 13-15 and 28-41, col. 13, ln. 41-48 and 53-64, col. 14, ln. 39-52, col. 18, ln. 1-5, col. 19, ln. 23-34, col. 23, ln. 13-23, col. 25, ln. 53-55, col. 28, ln. 11-30, and col. 31 ln. 8-20, and Fig. 5.	8-12, 20-24
Y	US 8,019,882 B2 (Rao et al.) 13 September 2011 (13.09.2011), col. 5, ln. 13-20, 32-34, and 53-64, and col. 7, ln. 20-21.	12, 24

☐ Further documents are listed in the continuation of Box C.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

09 April 2015 (09.04.2015)

Date of mailing of the international search report

15 MAY 2015

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450

Facsimile No. 571-273-8300

Authorized officer:

Lee W. Young

PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US

(72)発明者 ハント, タッド
アメリカ合衆国 94086 カリフォルニア州, サニーヴェイル, ソノラ・コート 1156,
エグザブロックス コーポレーション

(72)発明者 バラス, フランク イー
アメリカ合衆国 94086 カリフォルニア州, サニーヴェイル, ソノラ・コート 1156,
エグザブロックス コーポレーション