

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号
特許第6352308号
(P6352308)

(45) 発行日 平成30年7月4日 (2018.7.4)

(24) 登録日 平成30年6月15日 (2018.6.15)

(51) Int. Cl.

G 0 6 F 3 / 0 6 (2006.01)

F I

G 0 6 F 3 / 0 6 3 0 1 W

G 0 6 F 3 / 0 6 3 0 1 N

請求項の数 44 (全 50 頁)

(21) 出願番号	特願2015-556178 (P2015-556178)	(73) 特許権者	515208784
(86) (22) 出願日	平成26年1月31日 (2014.1.31)		シンボリック アイオー コーポレーショ
(65) 公表番号	特表2016-509309 (P2016-509309A)		ン
(43) 公表日	平成28年3月24日 (2016.3.24)		アメリカ合衆国, ニュージャージー州 O
(86) 国際出願番号	PCT/US2014/014209		7733, ホルムデル, 101 クローフ
(87) 国際公開番号	W02014/121102		ォーズ コーナー ロード
(87) 国際公開日	平成26年8月7日 (2014.8.7)	(74) 代理人	100114775
審査請求日	平成29年1月19日 (2017.1.19)		弁理士 高岡 亮一
(31) 優先権主張番号	13/756, 921	(74) 代理人	100121511
(32) 優先日	平成25年2月1日 (2013.2.1)		弁理士 小田 直
(33) 優先権主張国	米国 (US)	(74) 代理人	100202751
(31) 優先権主張番号	13/797, 093		弁理士 岩堀 明代
(32) 優先日	平成25年3月12日 (2013.3.12)	(74) 代理人	100191086
(33) 優先権主張国	米国 (US)		弁理士 高橋 香元

最終頁に続く

(54) 【発明の名称】 データを格納および取得するための方法およびシステム

(57) 【特許請求の範囲】

【請求項 1】

非キャッシュ記録媒体上にデータを格納するための方法であって、前記方法が、コンピュータソフトウェアにより指令されたコンピュータにより自動的に行われ、前記方法が、

- i . N バイトの I / O ストリームを受信することと、
- i i . 前記 N バイトを、 X バイトのフラグメント化ユニットにフラグメント化することと、
- i i i . 暗号化ハッシュ関数を X バイトの各フラグメント化ユニットに適用して、 X バイトの各フラグメント化ユニットに対して生成されたハッシュ関数値を形成することと、
- i v . 相関ファイルにアクセスすることであって、前記相関ファイルが、 Y ビットの格納されたハッシュ関数値を、複数の格納された X バイトのシーケンスの各々と関連付け、各フラグメント化ユニットに対して

(a) X バイトのフラグメント化ユニットに対して前記生成されたハッシュ関数値が前記相関ファイル内にある場合は、非キャッシュ記録媒体上に格納するために、前記 Y ビットの格納されたハッシュ関数値を使用し、

(b) 前記 X バイトのフラグメント化ユニットに対して前記生成されたハッシュ関数値が相関ファイル内にない場合は、 Y ビットの生成されたハッシュ関数値を前記 X バイトのフラグメント化ユニットとともに前記相関ファイル内に格納し、前記非キャッシュ記録媒体上に格納するために、前記生成されたハッシュ関数値を使用することと、を含む、方法。

【請求項 2】

i i iにおいて、前記暗号化ハッシュ関数がMD5メッセージダイジェストアルゴリズムを含む、請求項1に記載の方法。

【請求項 3】

前記方法が、矛盾解決モジュールを適用することをさらに含み、前記矛盾解決モジュールが、Xバイトのフラグメント化ユニットに対して、前記相関ファイル内の格納されたハッシュ関数値と同じ、ハッシュ関数値が生成されるが、前記Xバイトのフラグメント化ユニットが、前記格納されたハッシュ関数値と関連付けられたXバイトと異なると判断する場合、異なるZビットを、前記格納されたハッシュ関数値および前記生成されたハッシュ関数値と関連付ける、請求項1に記載の方法。

10

【請求項 4】

前記相関ファイル内で、矛盾が存在しない全てのハッシュ関数値に対して、Zビットが関連付けられており、前記Zビットが8乃至16個のゼロである、請求項3に記載の方法。

【請求項 5】

N = 4096 および X = 512 である、請求項4に記載の方法。

【請求項 6】

Y = 128 または 256 である、請求項5に記載の方法。

【請求項 7】

複数のハッシュ関数値を前記非キャッシュ記録媒体上に格納することをさらに含む、請求項1に記載の方法。

20

【請求項 8】

格納のための各ハッシュ関数値に対してビットマーカを生成するために、格納のための各ハッシュ関数値をビットマーカテーブルの使用を通して符号化することと、各ビットマーカを前記非キャッシュ記録媒体上に格納することをさらに含む、請求項1に記載の方法。

【請求項 9】

前記方法が、格納のための各ハッシュ関数値を頻度変換器の使用を通して符号化することをさらに含み、格納のための各ハッシュ関数値に対して、変換されたビット列が生成され、少なくとも2つの異なるYビット列に対して、出力される変換されたビット列が、異なる長さを有する、請求項1に記載の方法。

30

【請求項 10】

異なるサイズの複数の変換されたビット列すべてに対して、Aビット長の第1の変換されたビット列および、Bビット長の第2の変換されたビット列があり、A < Bで、かつ前記第1の変換されたビット列のAビットの識別が、前記第2の変換されたビット列のAビットの識別と同じでない、請求項9に記載の方法。

【請求項 11】

前記I/Oストリームがホストから受信されて、前記方法が、前記I/Oストリームの受信を前記ホストに対して確認することをさらに含む、請求項9に記載の方法。

【請求項 12】

40

前記ホストが、前記I/Oストリームを第1の記憶アドレスに格納されているとして記録し、かつ前記変換されたビット列が、第2の記憶アドレスにおいて前記非キャッシュ記録媒体内に格納されており、前記第1の記憶アドレスが前記第2の記憶アドレスと同じでない、請求項11に記載の方法。

【請求項 13】

前記第1の記憶アドレスが、第1のファイルサイズのファイルに対応し、かつ前記第2の記憶アドレスが、第2のファイルサイズのファイルに対応し、前記第1のファイルサイズが前記第2のファイルサイズの少なくとも4倍の大きさである、請求項12に記載の方法。

【請求項 14】

50

前記方法が、前記第 2 の記憶アドレスをメディアータ上に格納することをさらに含み、前記メディアータが持続性記憶媒体である、請求項 1 2 に記載の方法。

【請求項 1 5】

前記第 2 の記憶アドレスがビットフィールド内に格納される、請求項 1 4 に記載の方法。

【請求項 1 6】

非キャッシュ記録媒体上にデータを格納するための方法であって、前記方法が、コンピュータソフトウェアにより指令されたコンピュータにより自動的に行われ、前記方法が、

i . N バイトの I / O ストリームを受信することと、

i i . 前記 N バイトを、X バイトのフラグメント化ユニットにフラグメント化することと、

i i i . 暗号化ハッシュ関数値を X バイトの各フラグメント化ユニットと関連付けることであって、前記関連付けることが、相関ファイルにアクセスすることを含み、前記相関ファイルが、Y ビットの格納されたハッシュ関数値を、複数の格納された X バイトのシーケンスの各々と関連付けて、

(a) X バイトのフラグメント化ユニットのシーケンスが前記相関ファイル内にある場合は、非キャッシュ記録媒体上に格納するために、前記 Y ビットの格納されたハッシュ関数値を使用し、

(b) X バイトのフラグメント化ユニットのシーケンスが前記相関ファイル内にない場合は、Y ビットの新しいハッシュ関数値を前記 X バイトのフラグメント化ユニットとともに前記相関ファイル内に格納し、前記非キャッシュ記録媒体上に格納するために、Y ビットの新しいハッシュ関数値を使用することと、

を含む、方法。

【請求項 1 7】

非キャッシュ記録媒体上にデータを格納するための方法であって、前記方法が、コンピュータソフトウェアにより指令されたコンピュータにより自動的に行われ、前記方法が、

i . N バイトの I / O ストリームを受信することと、

i i . 暗号化ハッシュ関数値アルゴリズムを N バイトの各ユニットに適用して、N バイトの各ユニットに対して生成されたハッシュ関数値を形成することと、

i i i . 相関ファイルにアクセスすることであって、前記相関ファイルが、Y ビットの格納されたハッシュ関数値を、複数の格納された N バイトのシーケンスの各々と関連付けて、

(a) N バイトのユニットに対して前記生成されたハッシュ関数値が前記相関ファイル内にある場合は、非キャッシュ記録媒体上に格納するために、前記 Y ビットの格納されたハッシュ関数値を使用し、

(b) N バイトのユニットに対して前記生成されたハッシュ関数値が前記相関ファイル内にない場合は、Y ビットの生成されたハッシュ関数値を N バイトのユニットとともに前記相関ファイル内に格納し、前記非キャッシュ記録媒体上に格納するために、前記生成されたハッシュ関数値を使用することと、

を含む、方法。

【請求項 1 8】

記録媒体上にデータを格納するための方法であって、前記方法が、コンピュータソフトウェアにより指令されたコンピュータにより自動的に行われ、前記方法が、

i . 複数のデジタル 2 値信号を受信することであって、前記デジタル 2 値信号が複数のチャンクレットに編成されており、各チャンクレットが N ビット長で、N が 1 より大きい整数であり、かつ前記チャンクレットが順序を有することと、

i i . 各チャンクレットを均一なサイズのサブユニットに分割して、X 個のマーカのセットから、1 つのマーカを各サブユニットに割り当てて、複数のマーカのセットを形成することであって、X がサブユニット内のビットの異なる考えられる組合せの数以下であり、同一のサブユニットが同じマーカを割り当てられて、少なくとも 1 つのマーカ

10

20

30

40

50

ーがサブユニットのサイズよりも小さいことと、

i i i . 前記複数のマーカーのセットを、前記チャンクレットの順序に対応する順序で、持続性記録媒体上に格納することと、
を含む、方法。

【請求項 1 9】

前記割り当てが、ビットマーカーテーブルにアクセスすることを含み、前記ビットマーカーテーブル内で、各一意のマーカーが一意的ビット列に対応するものとして識別される、請求項 1 8 に記載の方法。

【請求項 2 0】

各サブユニットが第 1 の端部および第 2 の端部を有し、前記マーカーを割り当てる前に、前記方法が、各チャンクレットの各サブユニット内の 1 つ以上のビットを分析して、前記第 2 の端部におけるビットが値 0 を有するかを判断することと、前記第 2 の端部におけるビットが値 0 を有する場合には、前記第 2 の端部におけるビットおよび値 0 を有して、前記第 2 の端部におけるビットとともに連続的なビット列を形成する全てのビットを除去し、それにより、前記第 2 の端部に 0 を有する任意のサブユニットに対する修正されたサブユニットを形成することをさらに含む、請求項 1 9 に記載の方法。

【請求項 2 1】

コンピュータアルゴリズムが、

(a) 各サブユニットを再検討して、前記第 2 の端部に 0 があるかどうかを判断し、そうである場合は 0 を除去して、前記サブユニットの第 2 の端部に隣接していた位置に、修正された第 2 の端部をもつ、修正されたサブユニットを形成し、

(b) 各修正されたサブユニットを再検討して、前記修正された第 2 の端部に 0 があるかどうかを判断し、そうである場合は前記 0 を除去して、さらに修正された第 2 の端部を形成し、

(c) 各修正されたサブユニットに対して、その第 2 の端部に 1 をもつ、短縮されたサブユニットが生成されるまで (b) を繰り返す、請求項 2 0 に記載の方法。

【請求項 2 2】

各サブユニットが第 1 の端部および第 2 の端部を有し、前記マーカーを割り当てる前に、前記方法が、各チャンクレットの各サブユニット内の 1 つ以上のビットを分析して、前記第 2 の端部におけるビットが値 1 を有するかを判断することと、前記第 2 の端部におけるビットが値 1 を有する場合には、前記第 2 の端部におけるビットおよび前記値 1 を有して、前記第 2 の端部におけるビットとともに連続的なビット列を形成する全てのビットを除去し、それにより、前記第 2 の端部に 1 を有する任意のサブユニットに対する修正されたサブユニットを形成することをさらに含む、請求項 1 9 に記載の方法。

【請求項 2 3】

コンピュータアルゴリズムが、

(a) 各サブユニットを再検討して、前記第 2 の端部に 1 があるかどうかを判断し、そうである場合は 1 を除去して、前記サブユニットの第 2 の端部に隣接していた位置に、修正された第 2 の端部をもつ、修正されたサブユニットを形成し、

(b) 各修正されたサブユニットを再検討して、前記修正された第 2 の端部に 1 があるかどうかを判断し、そうである場合は前記 1 を除去して、さらに修正された第 2 の端部を形成し、

(c) 各修正されたサブユニットに対して、その第 2 の端部に 0 をもつ、短縮されたサブユニットが生成されるまで (b) を繰り返す、請求項 2 2 に記載の方法。

【請求項 2 4】

前記マーカーが頻度変換器内に格納され、前記マーカーが複数の異なるサイズで、より小さいサイズのマーカーがより高い頻度のサブユニットと相互に関連付けられている、請求項 1 9 に記載の方法。

【請求項 2 5】

複数の異なるマーカーが異なる数のビットから形成される、請求項 1 8 に記載の方法。

10

20

30

40

50

【請求項 26】

記録媒体からデータを取得するための方法であって、

i . 記録媒体にアクセスすることであって、前記記録媒体が複数のマーカ―をある順序で格納することと、

i i . 前記複数のマーカ―をビットチャンクレットのセットに翻訳することであって、各ビットチャンクレットがNビット長で、Nが1より大きい整数であり、前記ビットチャンクレットが、前記複数のマーカ―の順序に対応する順序を有し、前記翻訳がビットマーカ―テーブルにアクセスすることにより達成され、前記ビットマーカ―テーブル内で各一意のマーカ―が一意のビットチャンクレットに対応するものとして識別されることと、

i i i . 前記ビットチャンクレットのセットを含むデータ出力を生成することと、
を含む、方法。

10

【請求項 27】

前記記録媒体上に格納された際の複数のマーカ―が、XからYまでのサイズを有し、 $Y > X$ であり、少なくとも1つのマーカ―がサイズXを有し、かつ少なくとも1つのマーカ―がサイズYを有する、請求項26に記載の方法。

【請求項 28】

前記翻訳することが、

長さZより小さいマーカ―の全てを、0を前記マーカ―の第1の端部に追加することにより、長さZのマーカ―にレンダリングすることであって、ZがY以上であることと、

前記長さZのマーカ―をビットチャンクレットに翻訳することであって、前記ビットチャンクレットが長さZより長いことと、
を含む、請求項27に記載の方法。

20

【請求項 29】

前記長さZのマーカ―をビットチャンクレットに前記翻訳することが、前記長さZのマーカ―をサブユニットに翻訳することと、前記サブユニットをビットチャンクレットに結合することとを含む、請求項28に記載の方法。

【請求項 30】

データを記録媒体上に格納するための方法であって、前記方法が、コンピュータソフトウェアにより指令されたコンピュータにより自動的に行われ、前記方法が、

i . 複数のデジタル2値信号を受信することであって、前記デジタル2値信号がチャンクレットに編成されており、各チャンクレットがNビット長で、各チャンクレットが第1の端部および第2の端部を有し、Nが1より大きい整数であり、前記チャンクレットが順序を有することと、

30

i i . 各チャンクレットを複数のサブユニットに分割することであって、各サブユニットがAビット長であることと、

i i i . 各サブユニットを分析して、前記第2の端部におけるビットが値0を有するかを判断し、前記第2の端部におけるビットが値0を有する場合に、前記第2の端部におけるビットおよび値0を有して、前記第2の端部におけるビットとともに連続的なビット列を形成する全てのビットを除去し、それにより前記第2の端部に0を有する任意のチャンクレットに対して修正されたチャンクレットを形成することと、

40

i v . 持続性記録媒体上に、前記順序で、各修正されたサブユニット、およびAビット長であって、その第2の端部に1を有する、各サブユニットを格納することと、
を含む、方法。

【請求項 31】

データを記録媒体上に格納するための方法であって、前記方法が、コンピュータソフトウェアにより指令されたコンピュータにより自動的に行われ、前記方法が、

i . 複数のデジタル2値信号を受信することであって、前記デジタル2値信号がチャンクレットに編成されており、各チャンクレットがNビット長で、各チャンクレットが第1の端部および第2の端部を有し、Nが1より大きい整数であり、前記チャンクレットが順序を有することと、

50

i i . 各チャンクレットを複数のサブユニットに分割することであって、各サブユニットがAビット長であることと、

i i i . 各サブユニットを分析して、前記第1の端部におけるビットが値0を有するかを判断し、前記第1の端部におけるビットが値0を有する場合に、前記第1の端部におけるビットおよび値0を有して、前記第1の端部におけるビットとともに連続的なビット列を形成する全てのビットを除去し、それにより前記第1の端部に0を有する任意のサブユニットに対して第1の修正されたサブユニットを形成することと、

i v . 各サブユニットを分析して、前記第2の端部におけるビットが値0を有するかを判断し、前記第2の端部におけるビットが値0を有する場合に、前記第2の端部におけるビットおよび値0を有して、前記第2の端部におけるビットとともに連続的なビット列を形成する全てのビットを除去し、それにより前記第2の端部に0を有する任意のサブユニットに対して第2の修正されたサブユニットを形成することと、

v . 各サブユニットに対して、

(a) 前記第1の修正されたサブユニットおよび前記第2の修正されたサブユニットのサイズが同じである場合、前記第1の修正されたサブユニットまたは前記第2の修正されたサブユニットを格納することと、

(b) 前記第1の修正されたサブユニットが前記第2の修正されたサブユニットよりも小さい場合、前記第1の修正されたサブユニットを格納することと、

(c) 前記第2の修正されたサブユニットが前記第1の修正されたサブユニットよりも小さい場合、前記第2の修正されたサブユニットを格納することと、

(d) 修正されたサブユニットがない場合、前記サブユニットを格納することと、

(e) 第1の修正されたサブユニットはないが、第2の修正されたサブユニットがある場合には、前記第2の修正されたサブユニットを格納することと、

(f) 第2の修正されたサブユニットはないが、第1の修正されたサブユニットがある場合には、前記第1の修正されたサブユニットを格納することであって、格納される各修正されたサブユニットが、1つ以上のビットが前記第1の端部または前記第2の端部から除去されたかを示す情報とともに格納されることと、

を含むことと、

を含む、方法。

【請求項32】

記録媒体からデータを取得するための方法であって、前記方法が、コンピュータソフトウェアにより指令されたコンピュータにより自動的に行われ、前記方法が、

i . 記録媒体にアクセスすることであって、前記記録媒体が複数のデータユニットを複数の位置に格納し、各データユニットが複数のビットを含み、前記データユニットの最大サイズがNビットであり、少なくとも1つのデータユニットがNビットよりも少ないビットを含み、前記データユニットが順序を有することと、

i i . 前記データユニットを取得し、1つ以上のビットを、Nビット長よりも少ない任意のデータユニットの端部に追加して、前記データユニットに対応するビットチャンクレットのセットを生成することであって、各ビットチャンクレットのビット数が互いに等しいことと、

i i i . 前記ビットチャンクレットのセットを含む出力を、前記データユニットの順序に対応する順序で生成することと、

を含む、方法。

【請求項33】

i i において、値0のビットが追加される、請求項32に記載の方法。

【請求項34】

請求項26および33の一項に記載の方法を含む、文書を記憶装置から取得するための方法であって、前記出力をファイルタイプと関連付けることと、前記出力を、前記ビットチャンクレットを前記ファイルタイプの文書に変換することが可能なオペレーティングシステムに伝送することとをさらに含む、方法。

【請求項 3 5】

電子データの 1 つ以上のファイルを格納するための方法であって、前記方法が、コンピュータソフトウェアにより指令されたコンピュータにより自動的に行われ、前記方法が、

i . パラメータのセットを受信することであって、前記パラメータが、ファイルシステム情報、ブート可能性情報、およびパーティション情報のうちの 1 つ以上を含むことと、

i i . メタデータを受信することと、

i i i . 前記電子データの 1 つ以上のファイルを受信することであって、各ファイルがファイル名を有することと、

i v . 前記パラメータおよびメタデータをメディアータ上に格納することと、

v . 前記ファイルの各々をある位置において非キャッシュ媒体上に格納することと、

v i . 前記メディアータ上に、各ファイル名の、前記非キャッシュ媒体上の位置との相関関係を格納することと、

を含む、方法。

【請求項 3 6】

前記ファイルを前記非キャッシュ媒体上に格納する前に、前記ファイルを符号化することをさらに含む、請求項 3 5 に記載の方法。

【請求項 3 7】

前記符号化が、変換されたファイルを作成するためにビットマーカータブルおよび頻度変換器のうちの 1 つを使用することを含み、前記変換されたファイルが、ファイルシステム情報、ブート可能性情報、およびパーティション情報のいずれも含まない、請求項 3 6 に記載の方法。

【請求項 3 8】

前記パラメータが、複数の予約トラック内に格納されている、請求項 3 5 に記載の方法。

【請求項 3 9】

前記複数の予約トラックが、予約トラックの第 1 のセットであり、前記方法が、前記パラメータを予約トラックの第 2 のセットにコピーすることをさらに含む、請求項 3 8 に記載の方法。

【請求項 4 0】

前記予約トラックの第 1 のセット内のエラーを検査するために、前記予約トラックの第 2 のセットを使用することをさらに含む、請求項 3 9 に記載の方法。

【請求項 4 1】

前記メタデータが、シンプロビジョニングのための命令に対応する、請求項 4 0 に記載の方法。

【請求項 4 2】

前記ファイルが、前記ファイルを仮想アドレスに格納されているとして記録するホストから受信されて、前記仮想アドレスが、前記ファイルの位置と同じではない、請求項 3 5 に記載の方法。

【請求項 4 3】

データをバックアップするための方法であって、前記方法が、請求項 3 5 に記載の方法を含み、前記メディアータが第 1 のメディアータで、位置が第 1 の位置であり、前記方法が第 2 のメディアータを生成することをさらに含み、前記第 2 のメディアータが、時間 T 1 における前記第 1 のメディアータのコピーであり、T 1 の後である、時間 T 2 において：

i . 第 1 の位置に置かれている修正されたファイルを保存するための命令を受信することと、

i i . 新しいファイルを第 2 の位置に格納することであって、前記新しいファイルが前記修正されたファイルに対応することと、

i i i . 前記ファイル名を前記第 2 の位置と相互に関連付けるように前記第 1 のメディアータを更新することであって、前記第 2 のメディアータ上で、前記ファイル名が前記第

10

20

30

40

50

1 の位置と相互に関連付けられていることと、
を行う、方法。

【請求項 4 4】

データをバックアップするための方法であって、前記方法が、コンピュータソフトウェアにより指令されたコンピュータにより自動的に行われ、前記方法が、

i . 第 1 のメディアータ上で、複数のファイル名を複数のデータファイルの位置と相互に関連付けることであって、前記データファイルの位置が第 1 の非キャッシュ媒体上の位置に対応し、前記第 1 のメディアータが、特定のファイル名を識別するユーザーが、前記特定のファイル名に対応するデータファイルを取得するのを可能にするように構成されていることと、

i i . 前記複数のデータファイルを第 2 の非キャッシュ媒体にコピーすることと、

i i i . 第 2 のメディアータを生成することであって、前記第 2 のメディアータが時間 T 1 における前記第 1 のメディアータのコピーであり、前記第 2 のメディアータ内で、前記第 2 の非キャッシュ媒体上の複数のデータファイルの位置が前記ファイル名と相互に関連付けられていることと、

i v . データファイルに対する修正を保存するための命令を受信することと、

v . T 1 の後である、時間 T 2 において、前記第 1 の非キャッシュ媒体内で、前記データファイルに対する修正を保存することと、

v i . 修正されたデータファイルを前記第 2 の非キャッシュ媒体にコピーすることと、
を含む、方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データ記憶および取得の分野に関する。

【背景技術】

【0002】

21 世紀は、人々および企業が生成および格納するデジタル化された情報の量において急激な増加を示してきた。この情報は、通常、ディスクなどの磁気面上に格納されている電子データから成り、磁気面は、サイズがサブマイクロメータで、情報の個々のバイナリ断片を格納可能な、小さな領域を含む。

【0003】

多数のエンティティが生成する大量のデータのために、データ記憶産業は、ネットワークベースの記憶システムに方向転換している。これらのタイプの記憶システムは、1 つ以上のエンティティの代わりに、データを格納および取得するように構成されている、処理システムを形成するか、または処理システムの一部である、少なくとも 1 つのストレージサーバーを含み得る。データは、ブロックおよび / またはファイルなどの、記憶オブジェクトとして格納および取得され得る。

【0004】

格納のために使用される 1 つのシステムは、ネットワーク接続ストレージ (NAS : Network Attached Storage) システムである。NAS のコンテキストでは、ストレージサーバーは、1 つ以上のクライアントの代わりに、データを格納して、データに対するファイルレベルのアクセスを管理するように動作する。ファイルは、磁気もしくは光ディスクまたはテープなどの、大容量記憶装置の 1 つ以上のアレイを含む、記憶システム内に格納され得る。加えて、このデータ記憶モデルは、独立ディスクの冗長アレイ (RAID : Redundant Array of Independent Disks) 技術を採用し得る。

【0005】

格納のために使用される別のシステムは、ストレージエリアネットワーク (SAN) である。SAN システムでは、通常、ストレージサーバーは、クライアントに、格納されたデータに対するファイルレベルのアクセスではなく、ブロックレベルのアクセスを提供す

10

20

30

40

50

る。しかし、いくつかのストレージサーバーは、クライアントに、ファイルレベルアクセスおよびブロックレベルアクセスの両方を提供することが可能である。

【 0 0 0 6 】

N A Sを使用するか、S A Nを使用するかに関わらず、データを生成する全ての産業は、そのデータを格納および取得するための費用を考慮する必要がある。それ故、データを経済的に格納および取得するための新しい技術に対する必要性がある。

【発明の概要】

【課題を解決するための手段】

【 0 0 0 7 】

本発明は、データを格納および取得する効率を向上するための方法、システムおよびコンピュータプログラム製品を提供する。本発明の様々な実施形態を使用することにより、任意選択で変換または符号化されているデータを効率的に格納してアクセスすることができる。追加または代替として、本発明の様々な実施形態は、データの効率的な格納およびデータへのアクセスを容易にするメディエータ (m e d i a t o r) を使用する。

【 0 0 0 8 】

本発明の様々な実施形態は、生データおよび/または生データとは別個のメタデータで動作する。その結果として、本発明に関連して格納および/または取得できるファイルのタイプに基づく制限がない。使用され得るファイルタイプの例は、J P E G、P D F、W O R D文書、M P E GおよびT X T文書を含み得るが、それらに制限されない。本発明の様々な実施形態を通して、データを変形し得、かつ/または変形もしくは変換されたデータが格納される物理装置を変更し得る。これらの実施形態は、実行される際に、本発明の方法またはプロセスのステップの実行を引き起こす、コンピュータプログラム製品を含むか、またはそれに動作可能に結合されているコンピュータを採用する自動化プロセスを通して実施され得る。これらの方法またはプロセスは、例えば、コンピュータアルゴリズムもしくはスクリプトで具現化されるか、またはそれを含み、任意選択で、本発明のシステムによって実施され得る。

【 0 0 0 9 】

第1の実施形態によれば、本発明は、記録媒体上にデータを格納するための方法を対象とし、(i) 複数のデジタル2値信号を受信することであって、デジタル2値信号が複数のチャンクレットに編成されており、各チャンクレットがNビット長で、Nが1より大きい整数であり、チャンクレットが順序を有する、複数のデジタル2値信号を受信すること、(i i) 各チャンクレットを均一なサイズのサブユニットに分割して、X個のマーカーのセットから、1つのマーカーを各サブユニットに割り当てて、複数のマーカーのセットを形成することであって、Xがサブユニット内のビットの異なる組合せの数に等しく、同一のサブユニットが同じマーカーに割り当てられて、少なくとも1つのマーカーがサブユニットのサイズよりも小さい、複数のマーカーのセットを形成すること、および(i i i) 複数のマーカーのセットを、チャンクレットの順序に対応する順序、またはチャンクレットの順序の再作成を可能にする別の構成のいずれかで、持続性記録媒体上に格納すること：を含む。

【 0 0 1 0 】

第2の実施形態によれば、本発明は、記録媒体からデータを取得するための方法を対象とし、(i) 記録媒体にアクセスすることであって、記録媒体が複数のマーカーをある順序で格納する、記録媒体にアクセスすること、(i i) 複数のマーカーをチャンクレットのセットに翻訳することであって、各チャンクレットがNビット長で、Nが1より大きい整数であり、チャンクレットが、複数のマーカーの順序に対応する順序を有し、翻訳がビットマーカーテーブルにアクセスすることによって達成され、ビットマーカーテーブル内で、各一意のマーカーが一意のビット列に対応するとして識別される、複数のマーカーをチャンクレットのセットに翻訳すること、および(i i i) チャンクレットのセットを含む出力を生成すること：を含む。マーカーは、チャンクレットの順序と同じ順序で格納されることもあれば、格納されないこともあるが、それらが格納される順序に関わらず、チ

チャンクレットの順序およびそれらが導出された任意のファイルを再作成することができる。

【0011】

第3の実施形態によれば、本発明は、データを記録媒体上に格納するための方法を対象と、(i)複数のデジタル2値信号を受信することであって、デジタル2値信号がチャンクレットに編成されており、各チャンクレットがNビット長で、各チャンクレットが第1の端部および第2の端部を有し、Nが1より大きい整数であり、チャンクレットが順序を有する、複数のデジタル2値信号を受信すること、(ii)各チャンクレットを複数のサブユニットに分割することであって、各サブユニットがAビット長である、各チャンクレットを複数のサブユニットに分割すること、(iii)各サブユニットを分析して、第2の端部におけるビットが値0を有するかを判断し、第2の端部におけるビットが値0を有する場合に、第2の端部におけるビットおよび値0を有して、第2の端部におけるビットとともに連続的なビット列を形成する全てのビットを除去し、それにより第2の端部に0を有する任意のチャンクレットに対して修正されたチャンクレットを形成すること、ならびに、(iv)持続性記録媒体上に、Aビット長で、その第2の端部に1を有する、各修正されたサブユニットおよび各サブユニットを、その順序でチャンクレットの再構築を可能にする方法で格納すること：を含む。例えば、修正されたサブユニット（および修正されなかった任意のサブユニット）は、修正される前に、各チャンクレット内のサブユニットの順序に対応する順序で編成され得、データの格納は、元のファイル内の対応するチャンクレットと同じ順序であり得る。

【0012】

第4の実施形態によれば、本発明は、記録媒体上にデータを格納するための方法を提供し、(i)複数のデジタル2値信号を受信することであって、デジタル2値信号がチャンクレットに編成されており、各チャンクレットがNビット長で、各チャンクレットが第1の端部および第2の端部を有し、Nが1より大きい整数であり、チャンクレットが順序を有する、複数のデジタル2値信号を受信すること、(ii)各チャンクレットを分析して、第1の端部におけるビットが値0を有するかを判断し、第1の端部におけるビットが値0を有する場合に、第1の端部におけるビットおよび値0を有して、第1の端部におけるビットとともに連続的なビット列を形成する全てのビットを除去し、それにより第1の端部に0を有する任意のチャンクレットに対して第1の修正されたチャンクレットを形成すること、(iii)各チャンクレットを分析して、第2の端部におけるビットが値0を有するかを判断し、第2の端部におけるビットが値0を有する場合に、第2の端部におけるビットおよび値0を有して、第2の端部におけるビットとともに連続的なビット列を形成する全てのビットを除去し、それにより第2の端部に0を有する任意のチャンクレットに対して第2の修正されたチャンクレットを形成すること、(iv)各チャンクレットに対して、(a)第1の修正されたチャンクレットおよび第2の修正されたチャンクレットのサイズが同じである場合、第1の修正されたチャンクレットまたは第2の修正されたチャンクレットを格納すること、(b)第1の修正されたチャンクレットが第2の修正されたチャンクレットよりも小さい場合、第1の修正されたチャンクレットを格納すること、(c)第2の修正されたチャンクレットが第1の修正されたチャンクレットよりも小さい場合、第2の修正されたチャンクレットを格納すること、(d)修正されたチャンクレットがない場合、そのチャンクレットを格納すること、(e)第1の修正されたチャンクレットはないが、第2の修正されたチャンクレットがある場合には、第2の修正されたチャンクレットを格納すること、(f)第2の修正されたチャンクレットはないが、第1の修正されたチャンクレットがある場合には、第1の修正されたチャンクレットを格納することであって、格納される各修正されたチャンクレットが、1つ以上のビットが第1の端部または第2の端部から除去されたかを示す情報とともに格納されること：を含む。1つ以上のビットが第1の端部または第2の端部から除去されたかを示す情報は、例えば、サブユニットの一意性の形であり得る。任意の比較において比較される、第1および第2の修正されたチャンクレットの各々を生成する際に、好ましくは、0が、両方の端部ではなく、

10

20

30

40

50

いずれかの端部から除去されている。

【 0 0 1 3 】

第5の実施形態によれば、本発明は、記録媒体上にデータを格納するための方法を提供し、(i) 複数のデジタル2値信号を受信することであって、デジタル2値信号がチャンクレットに編成されており、各チャンクレットがNビット長で、各チャンクレットが第1の端部および第2の端部を有し、Nが1より大きい整数であり、チャンクレットが順序を有する、複数のデジタル2値信号を受信すること、(i i) 各チャンクレットを複数のサブユニットに分割することであって、各サブユニットがAビット長である、各チャンクレットを複数のサブユニットに分割すること、(i i i) 各サブユニットを分析して、第1の端部におけるビットが値0を有するかを判断し、第1の端部におけるビットが値0を有する場合に、第1の端部におけるビットおよび値0を有して、第1の端部におけるビットとともに連続的なビット列を形成する全てのビットを除去し、それにより第1の端部に0を有する任意のサブユニットに対して第1の修正されたサブユニットを形成すること、(i v) 各サブユニットを分析して、第2の端部におけるビットが値0を有するかを判断し、第2の端部におけるビットが値0を有する場合に、第2の端部におけるビットおよび値0を有して、第2の端部におけるビットとともに連続的なビット列を形成する全てのビットを除去し、それにより第2の端部に0を有する任意のサブユニットに対して第2の修正されたサブユニットを形成すること、ならびに、(v) 各サブユニットに対して、(a) 第1の修正されたサブユニットおよび第2の修正されたサブユニットのサイズが同じである場合、第1の修正されたサブユニットまたは第2の修正されたサブユニットを格納すること、(b) 第1の修正されたサブユニットが第2の修正されたサブユニットよりも小さい場合、第1の修正されたサブユニットを格納すること、(c) 第2の修正されたサブユニットが第1の修正されたサブユニットよりも小さい場合、第2の修正されたサブユニットを格納すること、(d) 修正されたサブユニットがない場合、そのサブユニットを格納すること、(e) 第1の修正されたサブユニットはないが、第2の修正されたサブユニットがある場合には、第2の修正されたサブユニットを格納すること、(f) 第2の修正されたサブユニットはないが、第1の修正されたサブユニットがある場合には、第1の修正されたサブユニットを格納することであって、格納される各修正されたサブユニットが、1つ以上のビットが第1の端部または第2の端部から除去されたかを示す情報とともに格納されること：を含む。1つ以上のビットが第1の端部または第2の端部から除去されたかを示す情報は、例えば、サブユニットの一意性の形であり得る。

【 0 0 1 4 】

第6の実施形態によれば、本発明は、記録媒体からデータを取得するための方法を提供し、(i) 記録媒体にアクセスすることであって、記録媒体が複数のデータユニットを複数の位置に格納し、各データユニットが複数のビットを含み、データユニットの最大サイズがNビットであり、少なくとも1つのデータユニットがNビットよりも少ないビットを含み、データユニットが順序を有する、記録媒体にアクセスすること、(i i) データユニットを取得し、1つ以上のビットを、Nビット長よりも少ない任意のデータユニットの端部に追加して、データユニットに対応するチャンクレットのセットを生成することであって、各チャンクレットが同じビット数を含む、チャンクレットのセットを生成すること、および(i i i) チャンクレットのセットを含む出力を、データユニットの順序に対応する順序で生成すること：を含む。

【 0 0 1 5 】

第7の実施形態によれば、本発明は、電子データを格納するための方法を対象とし、前記方法は、(i) パラメータのセットを受信することであって、パラメータがファイルシステム情報、ブート可能性情報、およびパーティション情報の1つ以上を含む、パラメータのセットを受信すること、(i i) メタデータを受信すること、(i i i) 1つ以上のファイルを受信することであって、各ファイルがファイル名を有する、1つ以上のファイルを受信すること、(i v) パラメータおよびメタデータをメタデータ上に格納すること、(v) ファイルの各々をある位置において非キャッシュ媒体上に格納すること、なら

びに (v i) メディエータ上に、各ファイル名と非キャッシュ媒体上の位置との相関関係を格納すること：を含む。本発明のメディエータは、次の目的：(1) データを符号化するためのプロトコルを格納すること、(2) 記録媒体上に物理空間を割り当てること、(3) ホストイニシエータのディスクジオメトリに対する中心点として機能すること、(4) セキュリティを追加すること、(5) システム内部装置が、ログをとること、読み取ること、および1つまたは2つの予約 (R_1 および R_2) とやりとりすることを可能にすること、(6) スナップショットをとる (すなわち、特定の時間において格納されたデータを凍結する) ため、および/またはディスクを複製するための新しい方法に対する枠組みを提供すること、ならびに(7) メタデータを提供すること：の1つ以上を果たし得る。全部ではないにしろ、これらの特徴の1つ以上を実現すると、データの格納、データの権限のないアクセスからの保護、および/またはデータの取得のための方法の効率性に寄与できる。

10

【0016】

第8の実施形態によれば、本発明は、データをバックアップするための方法を対象とし、前記方法は、(i) 第1のメディエータ上で、複数のファイル名を複数のデータファイルの位置と相互に関連付けることであって、データファイルの位置が第1の非キャッシュ媒体上の位置に対応し、第1のメディエータが、特定のファイル名を識別するユーザーが、その特定のファイル名に対応するデータファイルを取得するのを可能にする、複数のファイル名を複数のデータファイルの位置と相互に関連付けること、(ii) 複数のデータファイルを第2の非キャッシュ媒体にコピーすること、(iii) 第2のメディエータを生成することであって、第2のメディエータが時間 T_1 における第1のメディエータのコピーであり、第2のメディエータ内で、第2の非キャッシュ媒体上の複数のデータファイルの位置がファイル名と相互に関連付けられている、第2のメディエータを生成すること、(iv) データファイルに対する修正を保存するための命令を受信すること、および(v) T_1 の後である、時間 T_2 において、第1の非キャッシュ媒体上で、データファイルに対する修正を保存すること：を含む。好ましくは、修正は、第2の非キャッシュ媒体上の対応するデータファイル内に保存されない。

20

【0017】

第9の実施形態によれば、本発明は、データ格納および取得システムを提供し、(i) 非キャッシュデータ記憶媒体、(ii) メディエータであって、メディエータが非キャッシュデータ記憶媒体からリモートに保存され、メディエータが：(a) トラックの第1のセット、(b) トラックの第2のセット、(c) トラックの第3のセット、および(d) トラックの第4のセット、を含む、メディエータ、ならびに(ii) マネージャであって、マネージャが：(a) ファイルシステム情報、ブート可能性情報、およびパーティション情報の1つ以上を含むデータをトラックの第1のセット内に格納すること、(b) メタデータをトラックの第3のセット内に格納すること、(c) 1つ以上のファイルを非キャッシュ媒体上に格納することであって、1つ以上のファイルが非キャッシュ媒体上に、ファイルシステム情報、ブート可能性情報、およびパーティション情報のいずれもなしで保存される、1つ以上のファイルを非キャッシュ媒体上に格納すること、(d) トラックの第4のセット内に、非キャッシュ媒体内の各ファイルの位置を格納すること、ならびに(e) 非キャッシュ媒体内の各ファイルの位置の、ファイルに対するホスト名との相関関係を格納すること、を行うように構成されている、マネージャ：を含む。従って、マネージャは、マネージャ自身上に格納されない可能性がある、情報をメディエータ上に格納させ得る。

30

40

【0018】

第10の実施形態によれば、本発明は、データを非キャッシュ記録媒体上に格納するための方法を対象とし、(i) N バイトの I/O ストリームを (例えば、 I/O プロトコルを使用して) 受信すること、(ii) N バイトを、 X バイトのフラグメント化ユニットにフラグメント化すること、(iii) (値アルゴリズムである) 暗号化ハッシュ関数を X バイトの各フラグメント化ユニットに適用して、 X バイトの各フラグメント化ユニットに

50

対して生成されたハッシュ関数値を形成すること、(i v) 相関ファイルにアクセスすることであって、相関ファイルが、Yビットの格納されたハッシュ関数値を、複数の格納されたXバイトのシーケンスの各々と関連付けて、(a) Xバイトのフラグメント化ユニットに対して生成されたハッシュ関数値が相関ファイル内にある場合は、非キャッシュ記録媒体上に格納するために、Yビットの格納されたハッシュ関数値を使用し、また(b) Xバイトのフラグメント化ユニットに対して生成されたハッシュ関数値が相関ファイル内にはない場合は、Yビットの生成されたハッシュ関数値をXバイトのフラグメント化ユニットとともに相関ファイル内に格納し、非キャッシュ記録媒体上に格納するために、生成されたハッシュ関数値を使用する、相関ファイルにアクセスすること：を含む。

【 0 0 1 9 】

10

第11の実施形態によれば、本発明は、データを非キャッシュ記録媒体上に格納するための方法を対象とし、(i) NバイトのI/Oストリームを受信すること、(i i) Nバイトを、Xバイトのフラグメント化ユニットにフラグメント化すること、(i i i) 暗号化ハッシュ関数値をXバイトの各フラグメント化ユニットと関連付けることであって、前記関連付けることが、相関ファイルにアクセスすることを含み、相関ファイルが、Yビットの格納されたハッシュ関数値を、複数の格納されたXバイトのシーケンスの各々と関連付けて、(a) Xバイトのフラグメント化ユニットのシーケンスが相関ファイル内にある場合は、非キャッシュ記録媒体上に格納するために、Yビットの格納されたハッシュ関数値を使用し、また(b) Xバイトのフラグメント化ユニットのシーケンスが相関ファイル内にはない場合は、Yビットの新しく生成されたハッシュ関数値をXバイトのフラグメント化ユニットとともに相関ファイル内に格納し、非キャッシュ記録媒体上に格納するために、Yビットの新しく生成されたハッシュ関数値を使用する、暗号化ハッシュ関数値をXバイトの各フラグメント化ユニットと関連付けること：を含む。

20

【 0 0 2 0 】

第12の実施形態によれば、本発明は、データを非キャッシュ記録媒体上に格納するための方法を対象とし、(i) NバイトのI/Oストリームを受信すること、(i i) 暗号化ハッシュ関数をNバイトの各ユニットに適用して、Nバイトの各ユニットに対して生成されたハッシュ関数値を形成すること、(i i i) 相関ファイルにアクセスすることであって、相関ファイルが、Yビットの格納されたハッシュ関数値を、複数の格納されたNバイトのシーケンスの各々と関連付けて、(a) Nバイトのユニットに対して生成されたハッシュ関数値が相関ファイル内にある場合は、非キャッシュ記録媒体上に格納するために、Yビットの格納されたハッシュ関数値を使用し、また(b) Nバイトのユニットに対して生成されたハッシュ関数値が相関ファイル内にはない場合は、Yビットの生成されたハッシュ関数値をNバイトのユニットとともに相関ファイル内に格納し、非キャッシュ記録媒体上に格納するために、生成されたハッシュ関数値を使用する、相関ファイルにアクセスすること：を含む。

30

【 0 0 2 1 】

本発明の様々な方法が、例えば、電子データを格納するように記憶システムを構成するための次の方法に関連して使用され得る：(i) パラメータのセットを受信することであって、パラメータが、ファイルシステム情報、ブート可能性情報、およびパーティション情報のうちの、全部ではないにしろ、1つ以上を含む、パラメータのセットを受信すること、(i i) メタデータを受信すること、ならびに(i i i) パラメータおよびメタデータをメディアータ上に格納すること。記憶システムの構成後、システムは、各ファイルがファイル名を有する、1つ以上のファイルを受信し；ファイルの各々を(任意選択で、翻訳するための前述の方法の1つを通して処理されたとおり)非キャッシュ媒体上のある位置に格納して；メディアータ上に、各ファイル名と、非キャッシュ媒体上の位置との相関関係を格納する準備ができている可能性がある。

40

【 0 0 2 2 】

本発明のある方法を採用する際に、命令が、持続性コンピュータ可読記憶媒体上で符号化されて、データ処理装置に、(a) サーバーから、NバイトのI/OストリームをI/

50

Ｏプロトコルを通して受信すること、（ｂ）Ｎバイトの各ユニットに対して、または複数のＮバイトのフラグメント化ユニットに対して、ハッシュ関数値を取得すること、および（ｃ）複数のハッシュ値または複数の変換されたハッシュ値のいずれかを含む、データを格納させること：を含む動作を実行させるように動作可能な、コンピュータプログラム製品上に格納され得る。ハッシュ値を変換されたハッシュ値に変換することにより、格納されたファイル内の情報への権限のないアクセスに対する保護が高められ得る。任意選択で、コンピュータプログラム製品は、Ｎバイトの異なるユニットまたはＮバイトの異なるフラグメント化ユニットに割り当てられている同じハッシュ関数値の識別を可能にして、矛盾するハッシュ関数値の各々に対応する、異なるデータを格納させる、矛盾解決モジュールをさらに含む。コンピュータプログラム製品は、実行される際に、前述した特徴の開始および自動実行を引き起こすことができる。これらの命令は、１つのモジュール内、または動作可能に互いに結合されている、複数の別個のモジュール内に格納され得る。

10

【００２３】

加えて、本発明の様々な実施形態は、システム上でも実装され得る。本発明のシステムの一例は、（ａ）（永続的または持続性記憶装置であり得る）非キャッシュ記憶媒体、および（ｂ）非キャッシュ記憶媒体とやりとりするように動作可能な１つ以上のプロセッサ：を含む。任意選択で、システムは、ユーザーが１つ以上のプロセッサとやりとりするのを可能にできる１つ以上のユーザーインタフェース（例えば、グラフィックユーザーインタフェース）をさらに含む。１つ以上のプロセッサは、本発明の方法の１つ以上を実施するようにさらに動作可能であり、それらは、例えば、持続性媒体内に格納されるコンピュータプログラム製品上に格納され得る。

20

【００２４】

いくつかの実施形態では、システムは、（ｉ）メディエータであって、非キャッシュデータ記憶媒体からリモートに保存され、メディエータが：（ａ）トラックの第１のセット、（ｂ）トラックの第２のセット、（ｃ）トラックの第３のセット、および（ｄ）トラックの第４のセット、を含む、メディエータ、（ｉｉ）非キャッシュ媒体、ならびに（ｉｉ）マネージャであって、マネージャが：（ａ）ファイルシステム情報、ブート可能性情報、およびパーティション情報の１つ以上を含むデータをトラックの第１のセット内に格納させること、（ｂ）メタデータをトラックの第３のセット内に格納させること、（ｃ）１つ以上のファイルを非キャッシュ媒体上に格納させることであって、１つ以上のファイルが非キャッシュ媒体上に、ファイルシステム情報、ブート可能性情報、およびパーティション情報のいずれもなしで保存される、１つ以上のファイルを非キャッシュ媒体上に格納させること、（ｄ）トラックの第４のセット内に、非キャッシュ媒体内の各ファイルの位置を格納させること、ならびに（ｅ）非キャッシュ媒体内の各ファイルの位置の、ファイルに対するホスト名および／またはアドレス（複数可）との相関関係を格納させること、を行うように構成されている、マネージャ：を含む。

30

【発明の効果】

【００２５】

本発明の様々な実施形態を通して、データの格納および取得の効率が向上できる。効率の向上は、一般に使用される方法で使われるよりも少ない記憶空間を使用し、情報を格納する活動において、より少ない時間および努力を投資することにより、実現され得る。いくつかの実施形態では、データファイルの権限のない取得に対する保護も改善できる。これらの利益は、データを、リモートまたはローカルのいずれかで格納する際に実現され得、本発明の様々な実施形態は、ＲＡＩＤ技術と併せて、またはＲＡＩＤ技術とは無関係に使用され得る。

40

【図面の簡単な説明】

【００２６】

【図１】本発明のシステムの概要の表現である。

【図２】メディエータおよび非キャッシュ媒体（ＮＣＭ：non-cache-medium）の表現である。

50

【図 3】メディアータを使用して、情報を格納するためのシステムの表現である。

【図 4】格納されている情報をバックアップするために 2 つのメディアータを使用するためのシステムの表現である。

【図 5】二分木の表現である。

【図 6】本発明の方法を表す流れ図である。

【発明を実施するための形態】

【0027】

ここで、本発明の様々な実施形態に対する参照を詳細に行うが、その例が付随する図に示されている。以下の詳細な記述では、多数の具体的詳細が、本発明の完全な理解を提供するために記載されている。しかし、別段の指示がないか、またはコンテキストから暗黙的でない限り、詳細は例であることを意図し、本発明の範囲をいかなる方法でも制限すると見なされるべきでない。さらに、見出しは、本発明の実施形態のいずれかの範囲を制限すること、または、任意の特徴を特定の見出し内の節に制限することを意図するのではなく、読者の便宜のために使用される。

【0028】

定義

【0029】

別段の指示がないか、またはコンテキストから暗黙的でない限り、次の用語および句は、以下に規定する意味を有する。

【0030】

用語「ビット」は、二進数字を指す。それは、0 または 1 のいずれかによって表すことができる、2 つの値の 1 つを有することができる。

【0031】

用語「ブロック」は、所定の長さを有するデータのバイトまたはビットのシーケンスを指す。

【0032】

句「ブート可能性コード」、「ブート可能性情報」および「ブート可能性特徴」は、ブート可能状態に入るための手段を提供する情報を指し、ブートセクター上に格納され得る。ブートセクターは、ファームウェアによって RAM（ランダムアクセスメモリ）内にロードされるように構成されている機械コードを含み得、それは、その結果として、ブートプロセスがプログラムを記憶装置から、または記憶装置にロードするのを可能にする。例として、マスターブートレコードが、アクティブなパーティションを見つけて、ボリュームブートレコードを呼び出すコードを含み得、ボリュームブートレコードは、オペレーティングシステムまたは他のスタンドアロンプログラムをロードして呼び出すためのコードを含み得る。

【0033】

用語「バイト」は、8 ビットのシーケンスを指す。

【0034】

用語「キャッシュ」は、データに対する将来の要求がより迅速に対応されるため、またはバッファリングを目的として、データが一時的に格納される位置を指す。L1 キャッシュ（レベル 1 キャッシュ）は、例えば、プロセッサコアと統合されている、スタティックメモリを指す。L1 キャッシュは、CPU（汎用処理装置）が同じデータを複数回アクセスする場合にデータアクセス速度を向上するために使用され得る。L2 キャッシュ（レベル 2 キャッシュ）は、通常、L1 キャッシュよりも大きく、データファイルが検索されたが、L1 内で見つからなかった場合に、検索は、外部メモリを目指す前に、L2 キャッシュで行われ得る。いくつかの実施形態では、L1 キャッシュは、中央処理装置内ではない。代わりに、それは、DDR、DIMM または DRAM 内に置かれ得る。追加または代替として、L2 キャッシュは、PCI 2.0 / 3.0 の一部であり得、それは、マザーボードに入る。従って、L1 キャッシュおよび L2 キャッシュの各々は、マザーボードの別々の部分にあり得る。サイズに関して、本発明のいくつかの実施形態では、L1 キャッシュは

、2ギガバイトと128テラバイトの間、または2ギガバイトと4テラバイトの間であり、L2キャッシュは、16ギガバイトと1ペタバイトの間、または16ギガバイトと3.2テラバイトの間である。いくつかの実施形態では、本発明の方法が実装される際に、ビットマーカータブル、頻度変換器 (frequency converter) またはハッシュ値テーブルの1つ以上が、L2キャッシュ内に常駐する。

【0035】

用語「チャンクレット」は、セクタークラスタに対応し得るビットのセットを指す。チャンクレットのサイズは、記憶システムによって決定され得、チャンクレットサイズを有し得る。慣例的に、チャンクレットサイズは、CHS方式によって導出されたが、CHS方式は、シリンダー、ヘッドおよびセクターを、ハードディスク上にそれらが出現したところで定義するタプルによってブロックをアドレス指定した。ごく最近では、チャンクレットサイズは、論理ブロックアドレス (logical block addressing) を指す、LBA測定から導出されており、コンピュータ記憶装置上に格納されるデータのブロックの位置を指定するための別の手段である。例として、チャンクレットサイズは、512B、1K、2K、4K、8K、16K、32K、64Kまたは1MBであり得る。当業者として、1K=1024Bが分かっている。チャンクレットは、生データとしてホストから受信され得る。

【0036】

「ファイル」は、長さを有する関連したバイトまたはビットの集合である。ファイルは、チャンクレットよりも小さいか、チャンクレットと同じサイズか、またはチャンクレットよりも大きい可能性がある。

【0037】

句「ファイル名」は、コンピュータが特定のファイルを識別して、そのファイルを他のファイルから区別できるようにする表記法またはコードを指す。

【0038】

句「ファイルシステム」は、ファイルのセットを格納、取得、および更新するために使用される抽象化を指す。従って、ファイルシステムは、ファイルのデータおよびメタデータへのアクセス、ならびにデータを含む記憶装置上の利用可能な空間を管理するために使用されるツールである。いくつかのファイルシステムは、例えば、サーバー上に常駐し得る。ファイルシステムの例は、Unix (登録商標) ファイルシステムおよびその関連したディレクトリテーブルおよびiノード、Windows (登録商標) FAT16およびFAT32ファイルシステム (FATは、ファイルアロケーションテーブルを指す)、マスターファイルテーブルに基づく、Windows (登録商標) NTFS、ならびに、HFSまたはHFSプラスを使用する、Apple Mac OS Xを含むが、それらに制限されない。

【0039】

句「ハッシュ関数」、「暗号化ハッシュ関数値アルゴリズム」、および「ハッシュ関数値アルゴリズム」は、大きなデータセット (任意選択で可変長の) を、特定のハッシュ関数に対して固定長を有する、より小さいデータセットにマッピングする、アルゴリズムまたはサブルーチンを指す。「ハッシュ関数値」は、ハッシュ関数アルゴリズムの適用後に返される出力を指す。アルゴリズムが返す値は、ハッシュ値、ハッシュコード、ハッシュ合計、チェックサムまたはハッシュとも呼ばれ得る。例えば、MD5を使用する場合、出力は128ビットであり、他方、SHA-1を使用する場合、出力は160ビットである。

【0040】

用語「ホスト」および「イニシエータ」は、区別しないで使用され得、データを格納のために、本発明のデータ記憶および取得仲介システムに送信するエンティティまたはシステムを指す。ホストは、1つ以上のタイプの文書またはファイルに対応するデータおよび受信したデータを送信し得る。好ましくは、任意の入力/出力 (I/O) ストリーム内で、データは、単一の文書タイプのファイルに対応する。便宜上、句「I/Oストリーム」

は、データが1つのエンティティから別のエンティティに伝送されることを指すために使用される。第1のエンティティに対して出力されるものは、第2のエンティティに対する入力であり得る。

【0041】

略語「LBA」は、論理ブロックアドレス(logical block addressing)を指す。LBAは、リニアアドレス指定方式であり、ある記憶媒体、例えば、ハードディスク、内に格納されるデータブロックの位置を指定するために使用されるシステムである。LBA方式では、ブロックは、整数によって見つけられ、1つだけの番号がデータをアドレス指定するために使用される。通常、第1のブロックはブロック0である。

10

【0042】

略語「LUN」は、論理ユニット番号(logical unit number)を指し、論理ユニットを識別するために使用される。LUNは、通常、SANを介して共有されるブロックストレージレイを管理するために使用される。

【0043】

用語「マネージャ」は、コンピュータプログラム製品、例えば、持続性媒体内に格納され得、かつ1つ以上の他の動作(例えば、データの受信、送信、格納、および処理)をとらせるコードを指す。マネージャは、ハードウェア、ソフトウェアまたはそれらの組合せ上に格納され得る。いくつかの実施形態では、マネージャは、マネージャがその意図する機能を実施するのを可能にするように構成されている、コンピュータおよび/またはシス

20

【0044】

用語「メディアータ」は、ハードウェア、ソフトウェアまたはそれらの組合せ上に格納され得、かつ少なくとも1つの非キャッシュ媒体内の記憶空間の1つ以上のユニットをファイル名と相互に関連付ける、コンピュータプログラム製品を指す。メディアータは、それがポイントする非キャッシュ媒体よりも数桁小さい可能性がある。例えば、メディアータは、おおよそ、通常のシリンダーのサイズの約0.2%の大きさしかない可能性がある。いくつかの実施形態では、メディアータは、コンピューティングクラウド内に存在し得、一方、他の実施形態では、メディアータは、持続性有形的記録媒体内に存在する。メディアータは、データが実際には、記録媒体の異なるトラック内に出現している間に、ホストが、記録媒体のあるトラック内にあると認識する位置内のデータの、編成、変換、翻訳および格納の制御を行うことが可能であり得るか、または、全てではないにしろ、これらの機能の1つ以上に対応するマネージャに動作可能に結合され得る。さらに、メディアータは、セクターマップ、テーブル、または物理装置もしくは構造内に配置され得るデータの他の編成を含み得、従って、メディアータの内容は、物理装置もしくは構造にあるジオメトリを持たせ得る。いくつかの実施形態では、メディアータは、L2キャッシュ上に永続的に常駐する。他の実施形態では、メディアータは、ソリッドステートドライブ上に常駐する。

30

【0045】

用語「メタデータ」は、データのコンテナに関する管理情報を指す。メタデータの例は、読み取られているファイルの長さまたはバイトカウント；ファイルが修正された最後の時間に関する情報；ファイルタイプおよびアクセス許可を記述する情報；ならびにLUN、QoS、VMおよびWORMを含むが、それらに制限されない。他のタイプのメタデータは、オペレーティングシステム情報、自動初期化情報、グループ許可、および文書タイプ内のビットの頻度を含む。いくつかの実施形態では、格納されたメタデータは、例えば、イニシエータが格納しようとする文書の数およびサイズが縮小または増大するにつれて、イニシエータに対して記憶空間の効率的な収縮または拡張を可能にするために使用され得る。

40

【0046】

略語「NAS」は、ネットワークエリアストレージ(network area st

50

orage)を指す。NASシステムでは、ディスクアレイは、ローカルエリアネットワークトランスポートにアクセスできるコントローラに接続され得る。

【0047】

句「動作可能に結合された」は、システム、装置、および/またはモジュールが、互いにまたは相互に通信するように構成されて、通信時または通信した後に、それらの意図する目的を実施することが可能である。

【0048】

句「オペレーティングシステム」は、コンピュータハードウェア資源を管理するソフトウェアを指す。オペレーティングシステムの例は、Microsoft Windows (登録商標)、Linux (登録商標)、およびMac OS Xを含むが、それらに制限されない。

10

【0049】

用語「パーティション」は、記憶媒体、例えば、ディスクドライブをユニットに分割するフォーマットを指す。従って、パーティションは、ディスクパーティションとも呼ばれ得る。パーティションの例は、GUIDパーティションテーブルおよびAppleパーティションマップを含むが、それらに制限されない。

【0050】

略語「RAID」は、独立ディスクの冗長アレイ(redundant array of independent disks)を指す。関連したサーバーに対して、ディスクのこのグループは、単一ボリュームのように見え得る。RAID技術は、データの単一ストリップを複数のディスクからプルすることにより性能を向上させて、(1)データのミラーリング、(2)データのストライプ化、または(3)データのミラーリングとストライプ化の組合せ：などの1つまたは複数のプレミスタイプ上に構築される。

20

【0051】

句「記録媒体」は、ビットに対応する磁気信号をその中に格納できる、持続性有形的コンピュータ可読記憶媒体を指す。例として、記録媒体は、ハードディスクおよびソリッドステートドライブなどの非キャッシュ媒体を含むが、それらに制限されない。当業者であれば分かっているように、ソリッドステートドライブもキャッシュを有し、スピンする必要がない。持続性有形的コンピュータ可読記憶媒体の例は、ハードドライブ、ハードディスク、フロッピーディスク、コンピュータテープ、ROM、EEPROM、不揮発性RAM、CD-ROMおよびパンチカードを含むが、それらに制限されない。

30

【0052】

略語「SAN」は、ストレージエリアネットワーク(storage area network)を指す。このタイプのネットワークは、コンピューティング装置を、ディスク、テープアレイおよび他の記録媒体にリンクするために使用できる。データは、例えば、ブロックの形でSANを介して伝送され得る。

【0053】

略語「SAP」は、システムアシストプロセッサ(system assist processor)を指し、それは、オペレーティングシステムによって使用されるI/O(入力/出力)エンジンである。

40

【0054】

略語「SCSI」は、小規模コンピュータシステムインタフェース(small computer systems interface)を指す。

【0055】

用語「セクター」は、ディスク、例えば、磁気ディスク上のトラックの下位区分を指す。各セクターは、一定量のデータを格納する。ディスクに対する一般的なセクターサイズは、512バイト(512B)、2048バイト(2048B)、および4096バイト(4K)である。チャンクレットが4Kのサイズであり、各セクターが512Bのサイズである場合、各チャンクレットは8セクターに相当する($4 * 1024 / 512 = 8$)。セクターは、トラックを有し、プラッター上に配置される。一般に、2つまたは4つのブ

50

ラッターが1つのシリンダーを構成し、255のシリンダーがハードディスクおよび媒体装置を構成する。

【0056】

句「セクターマップ」は、ホストから要求を受信して、ファイルが格納されている記憶装置内の位置を相互に関連付けるツールを指す。セクターマップは、例えば、SCSIプロトコルによって定義されるパラメータ下で動作し得る。本発明のいくつかの実施形態では、セクターマップは、メディアータのビットフィールド内に配置され得る。

【0057】

用語「トラック」は、全てのセクターをトラバースするディスク内の円形の単位を指す。「トラックセクター」は、任意の1つのセクター内のトラックである。「トラックラスタ」は、2つ以上のセクターに及ぶ。

10

【0058】

好ましい実施形態

【0059】

本発明は、データを非キャッシュ記録媒体上に格納するための方法、これらの方法を実施するためのコンピュータプログラム製品、およびこれらの方法を実施するように構成されているシステムを提供する。本発明の様々な実施形態を通して、データを効率的に格納および取得することができる。

【0060】

ビットマーカ

20

【0061】

一実施形態によれば、本発明は、ビットマーカを、生データ内のビット列の表現として使用する、記録媒体上にデータを格納するための方法を対象とする。従って、生データは、生データを表す一連のマーカに翻訳され、本方法は、格納のために、信号のセット（ビットマーカと呼ばれ得る）に変換されるデータを含む、ファイルまたはストリームを受信することおよびそれらの信号を格納することを提供する。

【0062】

ファイルは、ホストと呼ばれる人またはエンティティから受信され得る。好ましくは、ホストは、信号を生データの形で送信する。例えば、ホストは、JPEG、PDF、TIFFまたはWORD文書などの、1つ以上のファイルを個々に、または全体として形成する1つ以上のチャンクレットを送信し得る。本発明の様々な実施形態は、チャンクレットを受信するか、チャンクレットのサブユニットを受信するか、またはチャンクレットに変換される1つ以上のファイルを受信すると、開始する。好ましくは、受信されるチャンクレットまたはチャンクレットのサブユニットは、全て均一なサイズである。任意選択で、本方法は、サイズの均一性を検証し、非均一性が検出される、例えば、1つ以上のチャンクレットが少なすぎるビットしか有していない場合、本方法は、全てのチャンクレットが均一のサイズになるまで、ゼロを追加させる。

30

【0063】

一例として、データは、Nビット長のチャンクレットで受信され得、Nは1よりも大きい整数である。例えば、Nは、128バイトから16Kまで、例えば、4Kであり得、それは、4096Bに相当する。

40

【0064】

チャンクレットは順番に受信される。例えば、ファイルが、システムによって連続的に受信される10のチャンクレットを含み得る。あるいは、所与のファイルに対する複数のチャンクレットが、それらが、ホストのオペレーティングシステムによるファイルの再作成および使用を可能にするような方法で、それらの相互の再関連付けを可能にする情報を含む場合、並行して、または一緒に伝送され得る。従って、いくつかの実施形態では、本発明の方法は、マーカを、チャンクレットが受信されるのと同じ順序で格納する。それに応じて、ホストがファイルの取得を要求する場合、対応する取得手段が、符号化されたデータを同じ順序で呼び戻して、それを適切な順序でチャンクレットに復号するであろう

50

。

【 0 0 6 5 】

任意選択で、符号化の前に、システムは、チャンクレットをビットのグループ（サブユニットとも呼ばれる）に分割し得、その各々はAビット長である。システムがチャンクレットをサブユニットに分割する場合、サブユニットがビットマーカータブルと比較され得る。システムがチャンクレットをサブユニットに分割しない場合、各チャンクレットがビットマーカータブルと比較され得る。

【 0 0 6 6 】

ビットマーカータブルは、ビットの一意のセットを一意のマーカータブルと相互に関連付ける。いくつかの実施形態では、ビットマーカータブルは、サブユニットが使用される場合にはサイズA、またはサブユニットが使用されない場合にはサイズNの、各一意のビット列に対するマーカータブルを含む。それ故、この方法では、コンピュータプログラムは、チャンクレットのセットを入力として受信し得る。それは、次いで、各チャンクレットを、同じサイズで、各々Aビット長のY個のサブユニットに分割し得、 $A/8$ は整数である。各一意のAに対して、テーブル内に1つのマーカータブルがあり得る。

【 0 0 6 7 】

このように、自動化プロトコルを通して、チャンクレットの受信後、コンピュータプログラム製品は、ビットマーカータブルにアクセスさせる。それに応じて、各チャンクレットまたはサブユニットは、入力として機能し得、各ビットマーカータブルは、出力として機能し得、それによりマーカータブルの出力セットを形成する。マーカータブルの出力セットは、翻訳されたか、コード化されたか、または符号化されたデータと呼ばれ得る。各チャンクレットが細分されない実施形態では、各チャンクレットは1つのマーカータブルを受信するであろう。チャンクレットが2つのサブユニットに分割される場合、それは、2つのマーカータブルに翻訳されるか、または符号化されるであろう。従って、コンピュータプログラム製品は、マーカータブルを入力と相互に関連付けるビットマーカータブルを使用して、各チャンクレットに対応する少なくとも1つのマーカータブルを割り当てる。コンピュータプログラム製品は、各個々のマーカータブルに対応する異なる出力が生成されるように、各チャンクレットに対応するマーカータブルのセットを含む異なる出力が生成されるように、または、完全なファイルに対応するマーカータブルのセットを含む異なる出力が生成されるように、設計され得る。

【 0 0 6 8 】

ビットマーカータブルは、X個のマーカータブルを含む。いくつかの実施形態では、Xは、本方法がチャンクレットをサブユニットに分割しない場合には、長さNのチャンクレット内のビットの異なる組合せの数、または本方法がチャンクレットを分割する場合には、長さAのサブユニット内のビットの異なる組合せの数のいずれかに等しい。文書タイプが分かっているか、または所与の長さのサブユニットもしくはチャンクレットに対するビットの全ての組合せよりも少ないと予測される場合、X（マーカータブルの数）は、ビットの考えられる組合せの実数よりも小さい可能性がある。例えば、いくつかの実施形態では、全てのビットマーカータブルが同じサイズであり、ビットマーカータブル内のビットマーカータブルの数が、サイズNまたはAのビット列内のビットの組合せの数のいずれかに等しい。他の実施形態では、全てのビットマーカータブルが同じサイズであり、ビットマーカータブル内のビットマーカータブルの数が、サイズNまたはAのビット列内のビットの組合せの数の、90%未満、80%未満、70%未満、もしくは60%未満である。

【 0 0 6 9 】

例として、いくつかの実施形態では、各チャンクレットは、複数の0および/または1から成るコード（すなわち、マーカータブル）を割り当てられる。他の実施形態では、各チャンクレットは、各々が、複数の0および1から成るコード（すなわち、マーカータブル）を割り当てられている複数のサブユニットに分割される。サブユニットは、長さAによって定義され得、 $N/A = Y$ で、Yは整数である。サブユニットがその数のビットを有していない場合、例えば、1つ以上のサブユニットが、システムが入力として受信するように構成されているビット数よりも少ないビット数しか持たない場合、システムはビット、例えば、ゼ

口を、全てのサブユニットが同じサイズになるまで、追加し得る。このステップは、例えば、チャンクレットがサブユニットに分割された後、全てのチャンクレットが同じサイズであるかどうかをまず確かめない場合に、実行され得る。あるいは、前述のように、それは、チャンクレットをサブユニットに分割する前に、チャンクレットレベルで実行され得る。

【 0 0 7 0 】

上の説明で示唆するように、アルゴリズムは、ビット列をコード化されたデータのセットに翻訳するように構成され得、アルゴリズムは、ビット列が、チャンクレットまたは、チャンクレットのサブユニットのいずれかに対応するように、設計され得る。好ましくは、コード化されたデータのセットは、ホストまたはクライアントから受信される際のファイルよりも小さい。しかし、コード化されたデータのセットが元のデータよりも小さいかどうかに関わらず、それは、変換してファイルのチャンクレットに戻すことが可能である。当業者であれば認識するように、格納のためにホストから受信されるデータは、生データであり、従って、任意の文書タイプに対応できる。

【 0 0 7 1 】

符号化は、2つの独立した目的に役立ち得る。第1に、格納のためにデータを符号化することにより、セキュリティが向上する。コードを知っている（すなわち、ビットマーカーターブルにアクセスできる）人またはエンティティだけが、それを復号して文書を再構築することができる。第2に、コードが元の文書よりも少ないビットを使用して作成される場合、必要な記憶空間が少なくなつて、費用を節約できる。1つ以上のファイルに対する記憶空間が削減されると、NCMは、もっと多くの数のファイルおよび/またはもっと大きなファイルに相当するデータを格納できる。

【 0 0 7 2 】

テーブル内のビットの少なくとも複数の一意の組合せに対して、好ましくは、システムがチャンクレットをサブユニットに分割しない場合、マーカーは、チャンクレット長Nよりも小さいか、またはシステムがチャンクレットをサブユニットに分割する場合には、サブユニット長Aよりも小さい。好ましくは、システムがチャンクレットをサブユニットに分割しない場合、どのマーカーもチャンクレット長Nよりも大きくないか、またはシステムがチャンクレットをサブユニットに分割する場合、どのマーカーもサブユニット長Aよりも大きくない。いくつかの実施形態では、全てのマーカーは、Nよりも小さいか、またはAよりも小さい。加えて、いくつかの実施形態では、各マーカーは、同じサイズであり得るか、または2つ以上のマーカーが異なるサイズであり得る。異なるサイズのマーカーがある場合、これら異なるサイズのマーカーは、例えば、テーブル内にあり得る。あるいは、テーブル内で全てのマーカーが同じサイズであるが、格納前に、全ての0がマーカーの一方または両方の端部から除去される。

【 0 0 7 3 】

コンピュータプログラム製品がチャンクレットを複数のマーカーに翻訳した後、コンピュータプログラム製品は、複数のマーカー（0が端部から除去されているかいないかに関わらず）を、持続性記録媒体上に、チャンクレットの順序に対応する順序で、またはチャンクレットが別の方法で再作成され得る順序で、格納させる。最終的に、マーカーは、非キャッシュ媒体である、持続性媒体内に格納される。しかし、任意選択で、それらは、まず、キャッシュ媒体、例えば、L1および/またはL2に送信され得る。従って、情報が失われないように、ビットマーカータブルが、永続的メモリ内に格納され得るが、使用時には、コピーが、例えば、L2キャッシュ内にあり得、ブートまたは再ブート時に、L2キャッシュ内のテーブルは永続的メモリから生成され得る。

【 0 0 7 4 】

一実施形態では、記憶装置は、所与のファイルに対応する、複数のビットマーカータブルを、非キャッシュ媒体内に格納する。ビットマーカータブルは、X～Yのサイズ範囲であり、XはYよりも小さく、少なくとも2つのマーカーが異なるサイズを有する。ビットマーカータブルの取得時または取得後に、コンピュータアルゴリズムは、所定のサイズのZよりも小さい、全

10

20

30

40

50

てのビットマーカ－の一方の端部に 0 を追加するが、ここで、Z は Y よりも大きい、または等しい。サイズ Z の各マーカ－が長さ A のビット列に翻訳される、ルックアップテーブルが、参照され得るが、ここで、A は Z 以上である。限定されない例では、 $X = 4$ 、 $Y = 20$ 、 $Z = 24$ 、 $A = 32$ である。いくつかの実施形態では、A は少なくとも Z より 50 % 大きい。A に対応するビット列は、チャンクレットに結合されるサブユニットであり得るか、またはチャンクレット自身であり得る。

【0075】

頻度変換器

【0076】

前述のように、ビットマーカ－テーブルは、マーカ－をビット列に、生データに対してランダムまたは非ランダムに、割り得て得、ビットマーカ－は、均一または不均一なサイズであり得る。しかし、前述のようなビットマーカ－テーブルの代わりに、頻度変換器を使用し得る。

【0077】

当業者であれば認識するように、本開示の例の節におけるテーブル 1 およびテーブル 2 は、ビットマーカ－（すなわち、変換されたビット）を、生データ内のビット列の頻度とは無関係な方法で割り当てる。しかし、前述し、以下の例 3 で説明するように、ある文書タイプまたは文書のセット内により頻繁に出現すると見込まれる生データに対して、より小さいマーカ－を割り当て得る。この方策は、全ての情報のほぼ 80 % が、最も頻出するサブユニットのほぼ上位 20 % 内に含まれるという事実を利用する。言い換えれば、データに対応するサブユニットは、非常に繰返しが多い。

【0078】

頻度変換器が異なるサイズのビット列をどのように使用できるかをさらに示すために、図 5 が参照され得る。図 5 は、5 ユニット長であって、数 1 から始まる、全ての 2 進シーケンスに対する二分木を示す。その木が示すように、最上位行から始まって、木を最下層まで下方に移動する、長さが 5 桁のシーケンスを作成するための 16 の経路がある。しかし、木を下方に移動する際に、例えば、第 3、第 4、または第 5 の行への分岐を進むことができる。それに応じて、本方法は、一片のデータに対して、木を一部下方に移動することに対応するコードを割り当て得、他方、他のデータに対しては、本方法は、異なる分岐に沿って、もっと多くの数のユニットを移動することに対応するコードを割り当てる。限定されない例として、変換されたデータの特定の一片に対して、第 3 行で止まって、例えば、101 のシーケンスを生成し得、他方、データの他の全てに対して、最初の 3 桁は 101 ではない。従って、本方法は、（図 5 のボックス内にある）1010、1011、10100、10101、10110 および 10111 の使用を決して許可しないであろう。メモリからデータを取得すると、システムは、数字を、それらが一意であると認識するまで読み取り、それらが一意であると認識すると、その一意の文字列を、復号を可能にするプログラムへの入力として使用し、そのプログラムは、ホストに対して生データファイルの再作成を可能にし得る。

【0079】

本発明の様々な実施形態は、（最も短い一意のコードに対応する）最小数のビットおよびその後の最小数の読取りに対する取得要求の間またはその後に、データファイルまたはテーブルと比較することにより一意性をチェックし得、コードがデータファイルまたはテーブル内で一意でない場合は、ビット数の拡張を継続して、シーケンスが一意であると認識されるまで各個々のビットを追加した後に、その拡張されたコードの一意性をチェックする。上記の例は、3、4、または 5 ビット列に関して記述しているが、それらのサイズは、例示のみを目的として使用されている。いくつかの実施形態では、頻度変換器に関連して使用するための一意のビットの最も短いシーケンスは、10 ~ 15 ビット長であり、ビットの最も長いシーケンスは 25 ~ 40 ビット長である。

【0080】

いくつかの実施形態では、異なるサイズの複数の変換されたビット列の全てに対して、

Aビット長の第1の変換されたビット列、およびBビット長の第2の変換されたビット列があり、ここで、 $A < B$ であって、第1の変換されたビット列のAビットの識別が、第2の変換されたビット列の最初のAビットの識別と同じでない。

【0081】

テーブル1、2および3（例を参照）が示すように、ビットのグループを表すためにマーカーが使用されるので、情報が変換され得、出力コードが、入力よりも少ない空間を占めるように構成できる。従って、好ましくは、テーブル内で、少なくとも1つの、複数の、少なくとも50%の、少なくとも60%の、少なくとも70%の、少なくとも80%の、少なくとも90%の、または少なくとも95%の、マーカーが、サブユニットよりもサイズが小さい。しかし、変換されたデータを同じサイズにするか、またはホストから受信したか、もしくはハッシュ関数値アルゴリズムから生成したデータよりも長くするのを妨げる技術的障害はない。

10

【0082】

フラグメンテーション

【0083】

前述のように、データストリームの受信後、いくつかの実施形態では、本方法は、受信されるデータのフラグメント化を必要とする。受信されるデータが大規模なファイルの形式である場合、それは、チャンクレットにフラグメント化され得る。受信されるデータがチャンクレットの形式である場合、チャンクレットはサブユニットにフラグメント化され得る。大規模なファイルは、まず、チャンクレットにフラグメント化され得、次いで、それらのチャンクレットがサブユニットにフラグメント化され得る。従って、ホストからの任意の入力に対して、0、1、または2つのフラグメンテーションのステップがあり得る。

20

【0084】

I/Oストリームをフラグメント化することにより、本方法は、各I/Oストリームを、所定の、好ましくは、均一サイズのもっと小さい（ブロックとも呼ばれ得る）ユニットに分割する。当業者であれば認識するように、より小さいブロックサイズを使用することの1つの利益は、小さいブロックは、データのさらに容易な処理を可能にすることである。限定されない例として、I/Oストリームは、4096バイト長であり得、本方法は、そのストリームを、1024、512、256、または128バイト長のフラグメント化されたユニットにフラグメント化し得る。データが受信された後、またはデータが受信されているときに、チャンクレットを、例えば、32ビットのサブユニットに分割する、アルゴリズムが実行され得る。

30

【0085】

サブユニットのサイズは、データをホストから受信するシステムの設計者の選択である。しかし、サブユニットのサイズは、チャンクレットが、一貫したサイズのサブユニットに分割されるように、選択されるべきであり、サブユニットは、ビットマーカーテーブルまたは頻度変換器の参照に関連して、容易に使用できる。

【0086】

チャンクレットのいずれかが他よりも小さい場合、任意選択で、小さいサイズのチャンクレットの受信時に、アルゴリズムが、その小さいチャンクレットを他のチャンクレットと同じサイズにするために、ゼロを追加する。代替として、システムは、チャンクレットをサブユニットに分割し、所望の長さより小さいサブユニットの取得時には、そのサブユニットの端部にゼロを追加し得る。

40

【0087】

前処理

【0088】

（符号化プロセスとも呼ばれ得る）翻訳プロセス中に、アルゴリズムが、ビットマーカーテーブルまたは頻度変換器に対する入力として使用する、ビット列（すなわち、チャンクレットまたはサブユニット）が、前処理され得る。これらのビット列の各々は、第1の

50

端部および第2の端部によって画定され得、マーカーを割り当てる前に、本方法は、各ビット列を分析して、第2の端部におけるビットが0の値を有するかを判断することをさらに含む。第2の端部におけるビットが0の値を有する場合、本方法は、第2の端部におけるビット、および0の値を有して、第2の端部におけるビットとともに連続的なビット列を形成する全ての後続ビットを除去し、それにより、削減されたサイズのビット列を形成する。前処理ステップの利益は、より小さいビットマーカーテーブルまたは頻度変換器が使用できることである。例えば、同じコード化されたデータを生成するために、テーブル1の代わりに、テーブル2が使用できる。当業者であれば認識するように、この前処理は、第2の端部からゼロの代わりに1を探して除去することにより達成できる。

【0089】

切捨て

【0090】

本発明の別の実施形態では、ビットマーカーテーブルまたは頻度変換器の使用を通して翻訳するのではなく、切り捨てられたサブユニットを、それらがチャンクレット内に存在するのと同じ順序で格納できる（または、サブユニットが使用されない場合には、チャンクレットが切り捨てられて格納できる）。従って、短くされた文字列を、例えば、ビットマーカーテーブル内で入力として使用するために、前処理されたデータとして機能させる代わりに、それら自体が格納され得る。

【0091】

このように、いくつかの実施形態では、データを記録媒体上に格納するための別の方法がある。この方法によれば、複数のデジタル2値信号を受信し、デジタル2値信号が、前述したフォーマットのチャンクレットに編成される。任意選択で、各チャンクレットは、前述のように、サブユニットに分割され得る。

【0092】

各チャンクレットまたはサブユニットは、その長さによって定義され得、各チャンクレットまたはサブユニットは、第1の端部および第2の端部を有する。各チャンクレットまたはサブユニットを分析して、第2の端部におけるビットが値0を有するかを判断し得、第2の端部におけるビットが値0を有する場合、第2のビットにおけるビットおよび両方が値0を有して、第2の端部におけるそのビットとともに連続的なビット列を形成する全てのビットを除去し、それにより、第2の端部に0を有する任意のチャンクレットまたはサブユニットに対して、修正されたチャンクレットまたは修正されたサブユニットを形成する。これらの修正されたビット列は元のビット列よりも短いので、それらは切り捨てられたと呼ばれ得る。

【0093】

チャンクレットまたはサブユニットが切り捨てられた後、その切り捨てられた情報を持続性記録媒体内に格納し得る。切り捨てられた情報を格納することにより、そうでなければ、切り捨てられなかったビット列で格納されていた同じ情報を格納するために、少ないビットしか使用されない。

【0094】

前述の様々な実施形態では、各サブユニットまたは各チャンクレットの、両方ではなく、第1の端部または第2の端部から、桁（複数可）を除去できる。しかし、各サブユニットまたはチャンクレットの第1の端部から桁の除去を検討し、各サブユニットまたはチャンクレットの第2の端部から桁を除去することを別に検討して、各サブユニットまたはチャンクレットに対して、第1の端部および第2の端部の一方または両方のいずれかで、切捨てが生じているかどうかを分析し、切捨てが一方の端部のみで生じている場合、切り捨てられたチャンクレットまたはサブユニットを保存し、切捨てが両方の端部で生じている場合には、切り捨てられたユニットの小さい方を保存する、方法を実施することは、本発明の様々な実施形態の範囲内である。チャンクレットまたはサブユニットの両方の端部から桁が除去できる方法を実施することも、本発明の範囲内である。

【0095】

10

20

30

40

50

このように、複数のデジタル2値信号を受信し得る。2値信号は、ユニットで、例えば、チャンクレットまたはチャンクレットのサブユニットで、受信され得る。各ユニットは、同じビット数長であり得、各ユニットは、第1の端部および第2の端部を有する。ユニット内のビット数は、1より大きい整数であり、ビットは、ユニット内に順序を有し、ユニットは順序を有する。

【0096】

次いで、第1の端部におけるビットが値0を有するかを判断するために各ユニットを分析し得、第1の端部におけるビットが値0を有する場合は、第1の端部におけるビットおよび両方が値0を有して、そのビットとともに連続的なビット列を形成する全てのビットを除去し、それにより、第1の端部に0を有する任意のユニットに対して第1の修正されたユニットを形成する。

10

【0097】

第2の端部におけるビットが値0を有するかも判断するために各ユニットを分析し得、第2の端部におけるビットが値0を有する場合は、第2の端部におけるビットおよび両方が値0を有して、そのビットとともに連続的なビット列を形成する全てのビットを除去し、それにより、第2の端部に0を有する任意のユニットに対して第2の修正されたユニットを形成する。

【0098】

各ユニットに対して、以下のデシジョンツリーが適用され得る：(a)第1の修正されたユニットおよび第2の修正されたユニットのサイズが同じである場合は、第1の修正されたユニット(または第2の修正されたサブユニット)を格納する；(b)第1の修正されたユニットが第2の修正されたユニットより小さい場合は、第1の修正されたユニットを格納する；(c)第2の修正されたユニットが第1の修正されたユニットより小さい場合は、第2の修正されたユニットを格納する；(d)修正されたユニットがない場合は、そのユニットを格納する；(e)第1の修正されたユニットはないが、第2の修正されたユニットがある場合は、第2の修正されたユニットを格納する；ならびに、(f)第2の修正されたユニットはないが、第1の修正されたユニットがある場合は、第1の修正されたユニットを格納する。

20

【0099】

1つ以上のビットが第1の端部または第2の端部から除去されているか、またはビットが第1の端部から除去されるユニットに対して第1のビットマーカータブルを、またビットマーカータブルが第2の端部から除去されるユニットに対して第2のビットマーカータブルを使用できるかを示す情報も格納し得、2つのビットマーカータブル間には、ビットマーカータブルの重複がない。これらの2つの異なるビットマーカータブルは、同じテーブルのセクションとして編成でき、修正されていないユニットに対するビットマーカータブルを含む。テーブルまたは複数のテーブルでは、第1の修正されたユニット、第2の修正されたユニットおよび修正されていない任意のユニットに対して、例えば、それらは両方の端部に1を有しているので、ビットマーカータブルの重複がない。

30

【0100】

当業者であれば認識するように、前述の方法はゼロの除去に関連して説明されているが、システムは代わりに1を除去できる。さらに、これらの方法は、ビットマーカータブルもしくは頻度変換器の使用に対する代替手段として使用されているとして説明される。しかし、これらの方法を、ビットマーカータブルまたは頻度変換器と併せて、例えば、それらのプロトコルの出力であるマーカータブルを、それらが元のファイルの取得および再構成を可能にする方法で格納されている限り、切り捨てることにより、使用することは、本発明の範囲内である。

40

【0101】

ハッシュ値

【0102】

前述したある実施形態では、データを符号化するためにビットマーカータブルまたは

50

頻度変換器にアクセスする。別の実施形態では、チャンクレットまたはサブユニットが、暗号化ハッシュ関数値アルゴリズムに対する入力として機能する。

【0103】

従って、各入力に対して、入力よりサイズが小さい出力がある。例えば、Xバイトのサブユニット（Xバイトのフラグメント化されたユニットとも呼ばれ得る）がある場合、出力は、Xバイトの各フラグメント化されたユニットに対して生成されたハッシュ関数値である。ハッシュ関数値アルゴリズムの例は、MD5ハッシュ（メッセージダイジェストアルゴリズムとも呼ばれる）、MD4ハッシュおよびSHA-1を含むが、それらに制限されない。ハッシュ関数値アルゴリズムから出力される値は、チェックサムまたは和（sum）と呼ばれ得る。いくつかの実施形態では、和は、64、128、160または256ビットのサイズである。I/Oストリーム内のデータの非常に繰り返しが多い特質のために、矛盾する和、すなわち、同じであるが、異なるフラグメント化ユニットに対応する和、を生成する確率は、比較的低い。

【0104】

いくつかの実施形態では、各ハッシュ値は、複数の0および/または1から成り、好ましくは、同じ長さの、ビット列またはコードである。これらのハッシュ値は、非キャッシュ記録媒体上に格納するために使用され得る。

【0105】

一実施形態では、システムは、例えば、非キャッシュ記録媒体上にデータを格納するための方法を実行し得：（i）NバイトのI/Oストリームを受信すること、（ii）Nバイトを、Xバイトのフラグメント化ユニットにフラグメント化すること、（iii）暗号化ハッシュ関数値をXバイトの各フラグメント化ユニットと関連付けることであって、関連付けることが、相関ファイルにアクセスすることを含み、相関ファイルが、Yビットの格納されたハッシュ関数値を、複数の格納されたXバイトのシーケンスと関連付けて、（a）フラグメント化されたXバイトのシーケンスが相関ファイル内にある場合は、非キャッシュ記録媒体上に格納するために、Yビットの格納されたハッシュ関数値を使用し、また（b）フラグメント化されたXバイトのシーケンスが相関ファイル内にない場合は、Yビットの新しいハッシュ関数値を生成して、Xバイトのフラグメント化シーケンスとともに相関ファイル内に格納し、非キャッシュ記録媒体上に格納するために、新しいハッシュ関数値を使用する、暗号化ハッシュ関数値をXバイトの各フラグメント化ユニットと関連付けること、を含む。

【0106】

前述の方法は、NバイトのI/Oストリームのフラグメント化を必要とする。しかし、本方法は、暗号化ハッシュ関数値アルゴリズムをI/Oストリームに適用して、それをフラグメント化しないことにより適用できる。従って、代替方法は、（i）NバイトのI/Oストリームを受信すること、（ii）暗号化ハッシュ関数値をNバイトの各ユニットと関連付けることであって、前記関連付けることが、相関ファイルにアクセスすることを含み、相関ファイルが、Yビットの格納されたハッシュ関数値を、複数の格納されたNバイトのシーケンスの各々と関連付けて、（a）Nバイトのシーケンスが相関ファイル内にある場合は、非キャッシュ記録媒体上に格納するために、Yビットの格納されたハッシュ関数値を使用し、また（b）Nバイトのシーケンスが相関ファイル内にない場合は、Yビットの新しいハッシュ関数値を生成して、Nバイトとともに相関ファイル内に格納し、非キャッシュ記録媒体上に格納するために、新しいハッシュ関数値を使用する、暗号化ハッシュ関数値をNバイトの各ユニットと関連付けること、を含む。さらに、様々な方法が、例示目的で、NバイトまたはXバイトのセットに適用されるとして説明されているが、当業者は、本方法はファイルの全てのNバイトまたはXバイトに対して好ましくは適用され得ることを理解するであろう。

【0107】

加えて、前述の方法は、バイト列がまだ相関ファイル内にない場合に限り、アルゴリズムを適用する。別の実施形態では、本発明は、非キャッシュ記録媒体上にデータを格納す

10

20

30

40

50

るための方法を提供し：(i) NバイトのI/Oストリームを受信すること、(ii) 暗号化ハッシュ関数値アルゴリズムをNバイトの各ユニットに適用して、Nバイトの各ユニットに対して生成されたハッシュ関数値を形成すること、(iii) 相関ファイルにアクセスすることであって、相関ファイルが、Yビットの格納されたハッシュ関数値を、複数の格納されたNバイトのシーケンスの各々と相互に関連付けて、(a) Nバイトのユニットに対して生成されたハッシュ関数値が相関ファイル内にある場合は、非キャッシュ記録媒体上に格納するために、Yビットの格納されたハッシュ関数値を使用し、また(b) Nバイトのユニットに対して生成されたハッシュ関数値が相関ファイル内にない場合は、Yビットの生成されたハッシュ関数値をNバイトのフラグメント化ユニットとともに相関ファイル内に格納し、非キャッシュ記録媒体上に格納するために、生成されたハッシュ関数値を使用する、相関ファイルにアクセスすること、を含む。この実施形態は、Nバイトをフラグメント化してフラグメント化ユニットを形成し、暗号化ハッシュ関数値アルゴリズムを、Nバイトのユニットの代わりに、フラグメント化ユニットに適用するステップを追加するようにも修正され得る。

10

【0108】

前述したある実施形態では、各チャングレット、フラグメント化ユニット、またはサブユニットに対してチェックサムが取得された後、相関ファイルにアクセスする。これらの方法は、先入れ先出し方式(「FIFO」)プロトコルに従ってチェックサムを取得して、I/Oストリームが受信されている間、チェックサムが生成されている間、または全てのI/Oストリームが受信されて、フラグメント化され、ハッシュ関数値アルゴリズムを適用された後のいずれかに、相関ファイルのアクセスを開始し得る。

20

【0109】

前述のように、相関ファイルは、Yビットの異なる格納されたハッシュ関数値を、複数の格納されたXバイトのシーケンスの各々と関連付ける。それに応じて、生成されたハッシュ関数値があるこれらの状況では：(a) Xバイトのユニットに対して生成されたハッシュ関数値が相関ファイル内にある場合、本方法は、Yビットの格納されたハッシュ関数値を非キャッシュ記録媒体上に格納するために使用させ、また(b) Xバイトのユニットに対して生成されたハッシュ関数値が相関ファイル内にない場合、本方法は、Yビットの生成されたハッシュ関数値を、Xバイトのシーケンスとともに相関ファイル内に格納させて、非キャッシュ記録媒体上に格納するために、生成されたハッシュ関数値を使用する。

30

【0110】

矛盾解決モジュール

【0111】

ハッシュ値アルゴリズムの適用の結果として、矛盾するハッシュ値が生成されている確率は低い。それ故、いくつかの実施形態では、本明細書で説明する方法を、矛盾解決モジュールを使わずに採用することを選択し得る。しかし、様々な実施形態では、本発明は、矛盾が生じ得る状況に対処するために、矛盾解決モジュールを適用する。

【0112】

図6に示すように、システムは、Nバイトのセットのストリームを受信し得る(610)。システムは、次いで、各フラグメント化ユニットに対して、ハッシュ関数を適用して、ハッシュ関数値を生成する(630)ために、Nバイトを望ましいサイズのユニットにフラグメント化させ得る(620)。次に、システムは、各ハッシュ値が既に相関ファイル内にあるかどうかを問い合わせる(640)。

40

【0113】

これらの実施形態では、Xバイトのフラグメント化ユニットに対して、相関ファイル内のハッシュ値と同じハッシュ値が生成されるときにはいつでも起動される(650)矛盾解決モジュールがある。このモジュールは次いで、Xバイトのフラグメント化ユニットが、そのハッシュ値に対して既に格納されたXバイトの値と同じであるかどうかを問い合わせる。矛盾解決モジュールが、Xバイトのフラグメント化ユニットに対して、相関ファイル内に格納されたハッシュ関数値と同じ、ハッシュ関数値が生成されるが、Xバイトのフ

50

ラグメント化ユニットが、格納されたハッシュ関数値と関連付けられたXバイトと異なる
と判断する場合、相関ファイルを更新して、生成されたハッシュ値を、NCMに書き込む
だけでなく、その中に格納する必要がある(660)。例えば、いくつかの実施形態では
、以下で説明するように、モジュールは、格納されたハッシュ関数値および生成されたハ
ッシュ関数値と関連付けられた異なるZビットがあるようにさせる。

【0114】

生成されたハッシュ値が(存在する場合は、任意の関連付けられたZ値とともに)一意
である場合、それは、NCM上に格納するために使用され得、相関ファイルが関連付けを
含むように更新され得る(680)。さらに、ハッシュ値が既に相関ファイル内にあり、
受信されたNバイトが相関ファイル内のものと同じである場合、格納されたハッシュ値が
(存在する場合は、任意の関連付けられたZ値とともに)NCMに書き込まれて、システ
ムは相関ファイルを更新する必要がない(670)。従って、相関ファイルファイルがア
クセスされた後、新しく生成されたハッシュ値が既に相関ファイル内にある場合に限り、
矛盾モジュールがチェックとしてアクセスされ得る。矛盾モジュールは次いで、矛盾があ
るか、または受信したファイルからのチェックサムおよびフラグメント化ユニットの両方
が、既に相関ファイル内で互いに関連付けられているかを判断するであろう。

【0115】

前述の実施形態は、各フラグメントのビットのユニットが、相関ファイルにアクセスす
る前に、暗号化ハッシュ関数値アルゴリズムを適用される方法を記述する。しかし、代替
として、まず、サブユニットまたはチャンクレットの存在について相関ファイルを確認し
、フラグメント化ユニットがまだテーブル内にない場合に限り、暗号化ハッシュ関数値ア
ルゴリズムを適用できる。次いで、チャンクレットまたはサブユニットが既に相関ファ
イル内にある場合は、格納されたチェックサム値を使用する。サブユニットまたはチャン
クレットがファイル内にない場合には、ハッシュ値アルゴリズムを適用して、矛盾をチェ
ックするためにプロトコルに入るであろう。本当の矛盾がある、すなわち、同じハッシュ
値が異なるサブユニットまたはチャンクレットと関連付けられている場合には、この問題に
対処するように相関ファイルを更新するであろう。

【0116】

本当の矛盾の問題に対処するための一手段は、矛盾が存在しない全てのハッシュ関数値
に対して、Zビットが関連付けられることであり、Zビットは、例えば、8~16個のゼロ
の均一の長さである。限定されない例として、N=4096バイト(チャンクレットの
サイズ)、X=512バイト(サブユニットのサイズ)、Y=128または256ビット
(ハッシュ値サイズ)、およびZ=8または16ビット(ハッシュ値拡張サイズ)である
。本方法は、例えば、チェックサムが、以前に格納されたチェックサムと矛盾しない場合
、チェックサムの端部に8個のゼロを結合させ得る。矛盾を識別した際には(例えば、異
なるフラグメント化ユニットが同じチェックサムと関連付けられている)、最新のチェッ
クサムが異なるZ値を割り当てられ得る。従って、相関ファイル内に格納された際のZ値
が00000000の場合、第1の矛盾しているチェックサムに対するZ値は、0000
0001であり得、別の矛盾しているチェックサムがあるとすれば、00000010で
ある。さらに矛盾しているチェックサムがある場合、矛盾しているチェックサムが識別さ
れると、各矛盾しているチェックサムは次のZ値を割り当てられ得る。

【0117】

各フラグメント化ユニットに対してチェックサムが取得され、矛盾モジュール(存在す
る場合)の適用に続いて、複数のハッシュ関数値(および、いくつかの実施形態では、Z
値とともに)が、格納のために非キャッシュ記録媒体に書き込まれ得る。

【0118】

ハッシュ値およびビットマーカまたは頻度変換器の併用

【0119】

いくつかの実施形態では、ハッシュ値を格納する代わりに、本方法は、格納のための各
ハッシュ関数値を、ビットマーカテーブルの使用を通して変換(符号化とも呼ぶ)して

10

20

30

40

50

、格納のための各ハッシュ関数値に対してビットマーカ－を生成すること、ならびにそれぞれのハッシュ関数値（および、いくつかの実施形態では、Z 値とともに）に対する各ビットマーカ－を、非キャッシュ記録媒体内に格納することをさらに含む。ビットマーカ－テーブルを使用することにより、ハッシュ値を変換またはコード化し得る。従って、ハッシュ値は、ビットマーカ－テーブルをアクセスする際に入力サブユニットとして機能し得、関連付けられたマーカ－が出力として機能し得る。

【0120】

サブユニットは、長さ A によって定義され得、A で割ったハッシュ値の長さは整数である。ビット列のいずれかがそのビット数を有していない、例えば、1 つ以上のビット列が、変換されるように構成されているサブユニット内の数よりも少ないビット数しか有していない場合、システムは、ビット、例えば、ゼロを、全てのサブユニットが同じサイズになるまで、追加し得る。代替として、0 の追加が、ハッシュ値をサブユニットに分割する前に、ハッシュ値レベルで実行され得る。さらに、ハッシュ値（および、いくつかの実施形態では、Z 値と結合されたとおり）がサブユニットのサイズよりも小さい場合、それらは、サブユニットを形成するように結合されるか、または適切なサイズのサブユニットを形成するために結合されてフラグメント化され得る。

【0121】

ハッシュ値のコード化されたデータへの変換は、アルゴリズムまたはコンピュータプログラムによって実施され得る。アルゴリズムまたはコンピュータプログラムの実行は、例えば、ハッシュ関数値アルゴリズムの適用も制御する、マネージャによって制御され得る。コード化されたデータ（変換されたデータとも呼ばれ得る）も、2 値信号から成り、ファイルのハッシュ値に変換して戻すことが可能な方法で、コード化されて格納される。従って、情報を失うことなく復号を可能にする符号化プロセス中に、情報が保持される。

【0122】

符号化は、2 つの独立した目的に役立ち得る。第 1 に、格納のためのデータをビットマーカ－テーブルを通して変換することにより、セキュリティが向上する。コードを知っている人またはエンティティだけが、それを復号して文書を再構築することができる。第 2 に、コードが、文書に対応するハッシュ値よりも少ないビットを使用して作成される場合、変換されたデータおよびハッシュ値の両方の使用により、必要な記憶空間が少なくなつて、費用をさらに節約できる。

【0123】

いくつかの実施形態では、ビットマーカ－テーブルを使用するのではなく、本方法は、格納のための各ハッシュ関数値を頻度変換器の使用を通して符号化することをさらに含む。これらの方法では、格納のための各ハッシュ関数値に対して、変換されたビット列が生成され、同じサイズのハッシュ値に対して、出力される変換されたビット列は異なる長さである。頻度変換器は、複数の変換されたビット列の各々を、複数のハッシュ関数値の各々と関連付け得、頻度変換器内の複数の変換されたビット列は、読み戻された際に、変換されたビット列が適切なハッシュ値と相互に関連付けることができるように、十分に区別できる。

【0124】

ハッシュ値アルゴリズムの使用を、（1）ビットマーカ－テーブルの適用、または（2）頻度変換器の適用：のいずれかと組み合わせることにより、さらなるセキュリティおよび/または経費節減を取り入れることができる。加えて、前述のように、ハッシュ値は、ビットマーカ－テーブルまたは頻度変換器に対する入力である。しかし、順序が逆転され得、チャンクレットまたはサブユニットが、マーカ－を生成するためにビットマーカ－テーブルまたは頻度変換器を使用するプロトコルに入り得、それらのマーカ－は、矛盾解決モジュールを使用する方法を含むが、それらに限定されず、本明細書で説明する方法のいずれかに従い、ハッシュ関数アルゴリズムに対する入力として機能し得る。

【0125】

メディアータおよびマネージャ

【 0 1 2 6 】

ビットマーカータブル、頻度変換器、ハッシュ値相関ファイル、ならびに前述のファイルおよびプロトコルにアクセスして使用するためのプログラムのいずれかまたは各々が、メディアータ、マネージャまたは他の場所に格納され得る。前述のプロトコルまたはファイルのいずれかが1つ以上が、本発明の実施形態に関連して使用され、メディアータまたはマネージャ内に格納されない場合はいつでも、好ましくは、メディアータ、マネージャ、または両方に動作可能に結合される。ローカルまたはリモートに格納されているファイルおよびプログラムと通信するための方法およびシステムは、当業者に周知である。

【 0 1 2 7 】

様々な実施形態では、マネージャは、本発明の方法を制御する。例えば、I/O（ホストからの出力、システムへの入力）がホストから受信された後、マネージャは、I/Oストリームの受信をホストに対して確認させ得る。

10

【 0 1 2 8 】

ホストは、I/Oストリームのファイルを第1の記憶アドレスに格納されているとして記録し得る。しかし、変換されたビット列は、実際には、第2の記憶アドレスにおいて非キャッシュ記録媒体内に格納され得、第1の記憶アドレスは、第2の記憶アドレスと同じではない。さらに、第1の記憶アドレスは、ファイルを第1のファイルサイズを有するとして示し（または、その表示と関連付けられ）得、他方、第2の記憶アドレスは、第2のファイルサイズに対応し得、第1のファイルサイズは、第2のファイルサイズの、少なくとも2倍の大きさ、または少なくとも4倍の大きさである。

20

【 0 1 2 9 】

いくつかの実施形態では、第2の記憶アドレスは、メディアータ上に格納される。メディアータ内で、第2の記憶アドレスはビットフィールド内に格納され得る。本明細書で説明するメディアータは、ビットマーカータブル、頻度変換器およびハッシュ値アルゴリズムの1つ以上を通してデータを翻訳するために説明した方法と無関係に、またはそれに関連して、使用され得る。

【 0 1 3 0 】

いくつかの実施形態では、メディアータは、ハードウェア、ソフトウェア、またはその組合せであり：（i）トラックの第1のセットであって、ファイルシステム情報、ブート可能性情報、およびパーティション情報に対応する情報を含むか、または包含する、トラックの第1のセット、（ii）トラックの第2のセットであって、トラックの第1のセットのコピーに対応する情報を含むか、または包含する、トラックの第2のセット、（iii）トラックの第3のセットであって、ファイルシステム情報、ブート可能性情報、およびパーティション情報以外のメタデータに対応する情報を含むか、または包含する、トラックの第3のセット、ならびに（iv）トラックの第4のセットであって、ビットフィールドに対応する情報を含むか、または包含する、トラックの第4のセット、を含む。例として、メディアータは、L2キャッシュ内に常駐し得、ビットマーカータブル、頻度変換器またはハッシュ値テーブルのいずれかが1つ以上は、使用中、サーバーのRAM内に常駐し得、また、マーカータブルまたはハッシュ値が格納されているNCMと同じか、または異なり得る、ソリッドステートデバイスの永続記憶装置内にも存在し得る。更新される場合、更新は、RAM内のテーブルに対して、および永続的メモリに対しても行われる。

30

40

【 0 1 3 1 】

メディアータの使用に加えて、本発明の様々な実施形態は、マネージャの使用を必要とする。マネージャは、1つ以上のモジュールを含み、ローカルコンピュータ上、ネットワーク上、またはクラウド内に常駐し得る。マネージャは、ある情報自体の受信を調整するか、またはある情報自体を受信して、この情報をメディアータに転送するか、または、情報の受信を直接メディアータによって制御するように構成されている。従って、方法は、イニシエータからの情報がマネージャからメディアータに流れるか、またはマネージャだけが情報の流れをシステムの他の構成要素に、例えば、ホストもしくはNCMに向けるように、設計できる。

50

【0132】

本発明の様々な実施形態では、マネージャは、ハッシュ関数アルゴリズムまたは他のプロトコル、任意の矛盾モジュール（存在する場合）および任意の変換モジュール（存在する場合）を適用するか、または適用させ得、データの流れを直接、メディアータに向かわせ得る。マネージャはまた、メディアータの使用を通じた情報の格納、ならびに情報の取得および伝送も制御し得る。情報を格納する場合、LBA番号が識別され得、格納されるデータが、ボトルネックを回避するか、またはそれを低減するために、バッファに送信され得る。さらに、格納中、L1および/またはL2キャッシュが使用され得る。

【0133】

同様に、データを取得する際に、本方法は、データをNCMから呼び出して、バッファに充填し、チェックサム値を取得して、フラグメント化ユニットを再作成および再構成し得るか、またはフラグメント化されていない場合は、ホストが受信して再検討し得る形式に情報を形成するように直接再構築し得る。格納されたデータが変換されたデータである場合は、チェックサム値を取得する前に、データが復号され得る。これらのステップは、必要なプロトコルを実行する、マネージャを通して調整および制御され得る。

【0134】

いくつかの実施形態では、マネージャは、1つまたは複数のメディアータを制御し、それと通信し、かつ、その活動を調整し得る。各メディアータに対して、マネージャは、パラメータのセットを受信する（または、その受信を調整する）。これらのパラメータは、ファイルシステム情報、ブート可能性情報、およびパーティション分割情報の1つ、2つ、もしくは3つ全部を含むか、基本的にそれらから成るか、またはそれらを含み得る。マネージャは、この情報を、メディアータ上のトラックの第1のセット内に格納させ、それは、予約1またはR₁と呼ばれ得る。ファイルシステムは、予約ブロックがどのように使用されるかを指示する。例えば、NTFSを使用する場合、セクター1~2がMBR（マスターブートレコード）用であり得、セクター3が\$MFT用であり得る。任意選択で、これらのトラックが、トラックの第2のセットにコピーされ得、それは、予約2またはR₂と呼ばれ得る。

【0135】

マネージャは、前のパラグラフで説明したパラメータに加えて、メタデータも受信し得る。メタデータは、メディアータ上のトラックの第3のセット内に格納される。マネージャがパラメータおよびメタデータを受信する時、またはその後、非キャッシュ媒体上に格納するための1つ以上のファイルも受信し得る。各ファイルは、ファイル名とともに受信される。ファイル名は、ファイルを伝送するホストによって生成され、ホストのファイルシステムによって定義され得る。例えば、SANもしくはNASもしくはそれらの組合せであるか、またはその一部であり得る、マネージャは、ファイルをファイル名とともに受信すると、自動的に、格納のための本明細書で説明したステップを実行できる。

【0136】

本発明のシステムは、ハッシュ関数値アルゴリズムおよび変換のためのアルゴリズムが、メディアータ、またはマネージャ、またはメディアータもしくはマネージャに動作可能に結合されている他のハードウェアおよび/もしくはソフトウェア内のいずれかに格納される。アルゴリズムのいずれかまたは両方は、ファイル名もメディアータ内に格納させ得る。メディアータが物理的にどこに置かれるかに関して制限はない。しかし、好ましくは、メディアータは、ホストまたは、好ましくは、メディアータからリモートに配置されている、ホストと通信可能なコンピュータと通信するように構成されている。メディアータは、直接的または間接的に（例えば、マネージャを通じて）、記録媒体、例えば、データのコード化セットが格納される非キャッシュ媒体（任意選択で、メディアータ、任意のマネージャおよびホストからリモートである）と、通信するようにも構成される。前述のように、メディアータは、ファイル名を識別するユーザーが、コード化されたデータのセットを非キャッシュ記憶媒体から取得するのを可能にする。

【0137】

キャッシュ

【0138】

前述のように、いくつかの実施形態では、生データを受信すると、本発明の方法は、受信の確認を自動的にホストに返させ得る。あるQoS（サービス品質）プロトコルでは、データファイルがI/Oを通して受信されて、直ちにL1キャッシュに送信される。受信されると、確認がL1キャッシュからI/Oを通して返送される。L1キャッシュから、データファイルがL2キャッシュに送信され得、L2キャッシュは確認をL1キャッシュに返送する。L2キャッシュは、データファイルを長期格納のために非キャッシュ媒体（NCM）にも送信し得る。NCMは、同様に、確認をL2キャッシュに返送し得る。

【0139】

いくつかの実施形態では、メディアータは、L1キャッシュ内のヒープ（動的に割り当てられるメモリ）内に常駐し得るか、またはそれに動作可能に結合され得る。代替として、メディアータは、カード内に常駐し得るか、またはL2キャッシュの一部であり得るか、もしくはL2キャッシュに動作可能に結合され得る。

【0140】

当業者であれば分かるように、メディアータを、L2に対してL1内に置くという判断は、格納されたデータの使用の頻度などの要因によって影響を受けるであろう。従って、L1キャッシュは、システムまたはエンドユーザーによって頻繁に使用されるデータを格納するために使用され、他方、L2キャッシュは、幾分頻繁にアクセスされるデータに対して使用され得る。

【0141】

別のQoSプロトコルでは、I/Oを通して、データファイルがL1キャッシュによって受信される。データファイルは、L1キャッシュからL2キャッシュおよびNCMの両方に転送される。L2キャッシュおよびNCMの各々が、確認をL1キャッシュに送信する。L2キャッシュおよびNCMの一方または両方から確認を受信する前、または後のいずれかに、L1キャッシュはI/Oを通して確認を送信する。

【0142】

ファイルの相関関係

【0143】

本発明の様々な実施形態では、ホストは、第1の記憶アドレスにおいて格納される各ファイルを理解するであろう。第1の記憶アドレスは、ホストによって、セクターマップ内に格納されて、LUNに対応し得る。それは、ファイルに対応する、ユニット、セクターまたはブロックの開始および、暗黙的または明示的のいずれかで、終了も含み得る。第1の記憶アドレスは、そのファイルが記憶装置またはストレージエリアネットワーク内で置かれているとホストが信じる位置に対応する。ホストは、このファイルアドレスを使用して、その格納された文書またはファイルを追跡し、また、それらを取得する。第1の記憶アドレスは、仮想アドレス、すなわち、データが実際に格納されている位置に対応していない。

【0144】

当業者であれば認識するように、ホストがファイル記憶アドレスを生成して、それを本発明のシステムに従い、SCSIコマンドで送信し、任意選択で、セクターまたはLBA番号（複数可）を関連付ける、方法およびシステムが使用され得る。メディアータは、ファイル名、ホストがファイルの位置として考えるもの、およびホストから受信されたときのファイルの記憶サイズ、すなわち、生データおよび任意のヘッダーもしくはフッターデータを、データの実際の記憶アドレスである、第2の記憶アドレスと相互に関連付け得、それは、ファイルがそれから再作成できるが、例えば、ハッシュ関数値アルゴリズム、ビットマーカータブルおよび頻度変換器の1つ以上を通して、変換されている、情報を含み得る。代替として、メディアータはファイル名のみを格納し得、任意選択で、ファイルに対する第1の記憶アドレスを受信しない可能性がある。前述のように、記憶アドレスは、データの線形構成に基づいているので、それらは、第1のユーザー認識LBAのみを列

10

20

30

40

50

挙するか、または明示的に全ての位置を列挙することにより、格納された情報のサイズを暗黙的に含み得る。

【0145】

前のパラグラフでは、ホストは、それが第1の記憶アドレスであると信じるものを提供すると記述しているが、情報は、ホストがそれを通してメディアータと直接的もしくは間接的に通信するダクト、ホスト内もしくはホストに動作可能に結合されたモジュール、またはメディアータおよび/もしくはマネージャ内であるか、もしくはそれに動作可能に結合されたモジュールのいずれかである別のエンティティによって生成され得る。当業者であれば認識するように、記憶装置上のデータファイルの位置を識別する格納された情報は、ポインタと呼ばれ得る。ファイルに対して、ポインタは、ユーザーが、データが配置されていると信じるNCM上の位置よりもむしろ、異なる位置(ブロック)からのデータの取得を指示し得る。

10

【0146】

任意選択で、ファイルをファイルタイプと関連付け得る。ファイルタイプは、データの受信者に、それをオープンするためにどのオペレーティングシステムが使用されるべきかを識別するように指示する。いくつかの実施形態では、ファイルタイプとの関連付けは、イニシエータまたはクライアントまたはホストで行われる。

【0147】

予備トラック

【0148】

前述のように、メディアータは、トラックの第1の予備セット(R_1)およびトラックの第2の予備セット(R_2)を含み得る。いくつかの実施形態では、トラックの第2の予備セット(R_2)は、トラックの第1の予備セット(R_1)のコピーである。追加として、いくつかの実施形態では、トラックの第2の予備セット(R_2)を使用して、トラックの第1の予備セット(R_1)内のエラーをチェックし得る。

20

【0149】

R_1 は、ホストの開始に対する中心点として機能するように構成され得る。従って、ホストは、 R_1 に送信するパラメータを選択し得る。メディアータは、この情報をホストから直接に、またはマネージャを通して間接的に、受信し得る。 R_2 は、好ましくは、決してホストに公開されない。従って、メディアータ自身またはマネージャのみが情報を R_2 に格納できる。 R_1 および R_2 の各々は、例えば、16のセクターを含み得、ホスト修飾子などの実際のデータで充填され得る。慣例により、番号付けは、0から始まり得る。従って、 R_1 は、例えば、セクター(またはトラック)0~15を含み得、 R_2 は、セクター(またはトラック)16~31を含み得る。しかし、メディアータは、 R_1 および R_2 の各々が、16トラックの初期サイズを超えて拡張するのを可能にするように構成され得る。

30

【0150】

いくつかの実施形態では、 R_1 は、一意の予備セクター情報およびパーティション情報を含む。パーティション情報内には、ファイルシステム情報を格納し得る。

【0151】

限定されない例として、また、当業者であれば分かるように、ボリュームをNTFSファイルシステムでフォーマットする際に、\$MFT(マスターファイルテーブル)、\$Bitmap、\$LogFileおよびその他のものの、メタデータファイルを作成する。このメタデータは、NTFSボリューム上の全てのファイルおよびフォルダに関する情報を含む。NTFSボリューム上の第1の情報は、パーティションブートセクター(\$Bootメタデータファイル)であり得、セクター0に配置され得る。このファイルは、基本NTFSボリューム情報およびメインメタデータファイル\$MFTの位置を記述し得る。

40

【0152】

フォーマットプログラムは、\$Bootメタデータファイルに対して最初の16セクターを割り当てる。最初のセクターは、ブートストラップコードを有するブートセクターで

50

あり、次の15のセクターは、ブートセクターのIPL（初期プログラムローダー）である。

【0153】

R₁およびR₂のトラックに加えて、メディエータは追加のメタデータを格納し得る。このメタデータは、例えば、シンプロビジョニングストラテジの実行を可能にする情報に対応し、それは、装置が、実際に利用可能であるよりも多くの物理的資源を有しているように見えるのを可能にして、例えば、トラック32～39であり得る、R₂の後の8トラック内に含まれ得る。メタデータは、LUN QoS、VMおよびWORMなどの特徴も提供し得る。

【0154】

いくつかの実施形態では、システムはSANインデクサを含み得る。SANインデクサは、何がR₁およびR₂内にあるかをチェックして、その情報を抽出し得る。この情報は、例えば、テキスト検索により、容易に検索され得るデータベースに入れることができる。

【0155】

ビットフィールド

【0156】

メディエータは、ビットフィールドに対応する情報も含むか、または包含し得る。ビットフィールドは、データが記憶媒体内の物理的にどこに格納されているかを示す情報を含み、メタデータがトラック32～39に置かれている場合、ビットフィールドのセクター番号はトラック40から始まり得る。ホストのファイル名とデータの位置との間の相関関係が格納されているのはメディエータのビットフィールド内である。従って、それは、セクターマップを含むか、基本的にそれから成るか、またはそれから成り得る。メディエータのビットフィールド構成要素からのこの情報は、任意の装置上での実際の空間節約を判断するために使用され得る。例えば、節約される空間の割合 = 1 - [（実際に使用される空間） / （ホストによってマップされている空間）] である。いくつかの実施形態では、本発明の方法に従ったデータ変換によって節約される空間は、少なくとも50%、少なくとも60%、少なくとも70%、または少なくとも80%である。これらの節約は、ファイル当たりであるか、または装置上の全てのファイルにわたる平均値であり得る。

【0157】

事実上、好ましくは、メディエータは、ファイルデータが格納されているディスクまたは記録媒体上に置かれていない。加えて、好ましくは、メディエータは、対応するディスクまたは記録媒体の総メモリの約0.1～0.2%のみを必要とする。

【0158】

空間の節約から経済的価値を提供することに加えて、本発明の様々な実施形態は、データの整合性を保護しようとする際に、効率の向上に対してドアを開く。その結果、本発明の様々な実施形態は、データのバックアップに対する新規で非自明の技術を提供する。

【0159】

多くの実施形態では、格納されたファイルは、ホストから受信したとおりの生データファイルより小さいので、それに対して必要な記憶空間が少ない。従って、ファイルの再作成に必要なデータは、ホストが使用されていると認識し、かつ第1のアドレスが示唆するよりも小さい位置に格納できる。ファイルが格納されている実際の位置は、第2の記憶アドレスと呼ばれ得る。従って、各ファイルに対して、ファイルが格納されているとホストが信じる、第1の記憶アドレス、およびコード化ファイルが実際に格納されている、第2の記憶アドレスがある。

【0160】

1つ以上のブロックに対応し得る、所与のファイルに対して、第1の記憶アドレスおよび第2の記憶アドレスが、記憶装置内の同じブロックまたはブロックの1つ以上の重なり合うセットに配置されることが可能である。しかし、好ましくは、ファイルの、少なくとも1つ、少なくとも50%、少なくとも60%、少なくとも70%、少なくとも80%、

10

20

30

40

50

少なくとも90%、または100%に対して、第1の記憶アドレスおよび第2の記憶アドレス内に重なり合うブロックがない。加えて、たとえホストが、メディアータが認識するのと同じ記憶アドレスを認識しても、データがコード化される際に、ホストは、まずデータを復号することなく、ファイルを再作成できない。いくつかの実施形態では、ホストは、コードを認識せず、従って、格納されたデータを復号できない。

【0161】

いくつかの実施形態では、メディアータは、ファイルに対応するチャンクレット（複数可）を受信して、それらをL1またはL2キャッシュ内に一時的に格納し得る。L2キャッシュが存在する場合、L2キャッシュは、受信をホストに対して確認し得、任意選択で、第1の記憶アドレスをホストに提供するか、または確認し得る。当業者であれば認識するように、受信の確認および第1の記憶アドレスの伝送が、第2の記憶アドレスにおける格納の前に、かつ符号化が実行される場合は、符号化の前または後に、行われ得る。確認がいつ送信されるかに関わらず、ファイルの実際の記憶アドレスと仮想記憶アドレスの相関関係が、好ましくは、ビットフィールド内で追跡される。

10

【0162】

データのバックアップ

【0163】

ある実施形態では、データのバックアップを容易にするために2つのメディアータを使用し得る。例えば、第1のメディアータでは、第1のセクター内に格納されるデータファイルまたは第1の記録媒体上の第1のセクタークラスタをファイル名と相互に関連付け得る。前述のように、第1のメディアータは、ファイル名を識別するユーザーまたはエンティティがデータファイルを記録媒体から取得するのを可能にするように構成される。

20

【0164】

第2のメディアータを生成する、データ保護プロトコルが実行され得る。第2のメディアータは、時間T1における第1のメディアータの正確なコピーであろう。従って、T1において、第1のメディアータおよび第2のメディアータの両方は、第1の記録媒体上の同じセクターまたはセクタークラスタ（およびその中の位置）を指す。

【0165】

時間T1の後、例えばT2において、ホストは、所与のセクターまたはセクタークラスタ上の所与の位置内に格納されているファイルを更新しようとし得る。ホストは、第1の記憶アドレスに格納されたデータを変更しない。セクターまたはセクタークラスタ上の情報を上書きさせるのではなく、第1のメディアータは、新しい情報を、異なるセクターまたはセクタークラスタ内の位置に対応する第3の記憶アドレスに書き込ませて、ファイル名および第1の記憶アドレスをこの新しい記憶アドレスと相互に関連付け得る。

30

【0166】

このように、たとえ、情報が特定の記憶アドレスにおいて上書きされているとホストが信じて、第1のメディアータは、新しいセクターまたはセクタークラスタを指す。それに応じて、ホストは、そのアドレスをセクタークラスタに対して更新する必要がない。

【0167】

第2のメディアータは更新されずに、ファイル名を、そのファイルが格納された第1の位置と継続して相互に関連付ける。2つのメディアータのこの使用は、ファイルが、T1およびT2の両方において存在していたとおりに格納されていることを示すように、ホストに、そのファイルシステムを更新させる必要なく、データがT1において存在するとおりにそのデータのスナップショットを提供するのを可能にする。従って、スナップショットは、時間T1において格納されている全てのデータファイルをロックして、いずれも、それらの物理ファイルを通して削除または書き込みが行われるのを防ぐ。しかし、ホストがそれらのファイルを修正したい場合、実際には新しいファイルが格納されているときに、ホストは、それを行っていると思い込んで動作できる。この方法は、セクターおよびセクタークラスタと関連して記述される。しかし、それは、セクターまたはセクタークラスタ内に配置されていない非キャッシュ媒体でも動作するであろう。例えば、それらは、LU

40

50

N内のLBAによって編成され得る。

【0168】

上で示唆するように、この方法は、第1のメディアータ、第2のメディアータおよび非キャッシュ媒体を含むシステムによって実装され得る。第1のメディアータ、第2のメディアータおよび記録媒体の各々は、持続性媒体を含むか、基本的にそれから成るか、またはそれから成り得る、別個の装置上に格納され得るか、または別個の装置から形成され得る。前述のシステムは、同じ記録媒体の異なるセクターの使用を詳述するが、異なる記録媒体の異なるセクターに書き込むことによっても使用できる。加えて、システム内で、メディアータおよび記録媒体は、互いに、ならびに任意選択で、命令を格納する1つ以上のコンピュータまたはCPUに、動作可能に結合されて、それらに、それらの意図する機能を実行させ、ネットワークを経由して、1つ以上のホストへ1つ以上のポータルを通して通信させる。さらになお、この実施形態は2つのメディアータの使用と関連して説明されているが、2つの別個のメディアータではなく、同じメディアータの2つのセクションを使用するシステムを実装できる。

10

【0169】

システムは、ロックモジュールとともにさらに構成され得る。ロックモジュールは、ある時間において書き込まれている1つ以上のブロックにおける修正、上書き、または削除を防ぎ得る。ロックモジュールは、新しいブロックの書き込みおよび、ロックされていないそれらの新しいブロックの修正を可能にするように設計もされ得る。従って、ロックモジュールは、ホスト、ユーザーまたはシステム管理者が、ある時間において書き込まれているあるブロックを選択するか、またはある時間において書き込まれている全てのブロックが上書きされないように選択するのを可能にするように構成され得る。

20

【0170】

さらに、デフォルト設定により、ファイルの取得および修正、上書きまたは削除に対する全ての要求を第1のメディアータを通じて送信する選択モジュールがあり得る。選択モジュールは、ロック技術が適用された時間における1つ以上のファイルの古いバージョンであるとホストが信じ得るものの復旧を可能にするように構成もされ得る。任意選択で、選択モジュール全体へのアクセスが、権限を有する人、例えば、システム管理者に制限され得る。

【0171】

データをバックアップするための前述のシステムが、2つのメディアータのコンテキストで説明される。しかし、格納されたファイルの履歴またはファイルのバージョンを捕捉するために3つ以上のメディアータが使用できる。例えば、少なくとも3つ、少なくとも4つ、少なくとも5つ、または少なくとも10のメディアータなど、が使用され得る。追加として、ホストは、メディアータに、一定の間隔で（例えば、毎週、毎月、3か月ごともしくは毎年）または不規則な間隔で（例えば、要求に応じて）、スナップショットを取らせ得る。

30

【0172】

データをバックアップするための別の方法によれば、非キャッシュ媒体のクローンが作成され得る。この方法では、第1のメディアータ内で、複数のファイル名を、非キャッシュ記憶媒体上に格納されている複数のデータ位置と相互に関連付ける。第1のメディアータは、特定のファイル名を識別するユーザーが、特定のファイル名に対応する第1の非キャッシュ記憶媒体からデータファイルを取得するのを可能にするように構成されている。特定ファイルの一部または全体が、第1のセクターまたはセクタークラスタ内に格納され得る。

40

【0173】

複数のデータファイル（または第1の非キャッシュ記憶媒体の全てのデータファイル）のコピーを第2の非キャッシュ記憶媒体および第2のメディアータに対して作成し得る。第2のメディアータは、時間T1における第1のメディアータのコピーであり、第2の非キャッシュ記憶媒体に動作可能に結合されている。T1の後である、時間T2において、

50

システムは、第1の非キャッシュ記憶媒体上の前記第1のセクターまたはセクタークラスタ内に格納されているデータファイルに、修正を保存し得る。しかし、第2の非キャッシュ媒体上の対応する位置に対する変更は行われぬ。ユーザーがT2の後にファイルを要求すると、彼または彼女は、第1のメディアータを経由して、直前に格納されたファイルのバージョンを取得するであろう。しかし、システム管理者は、第2の非キャッシュ媒体上に格納されて、第2のメディアータを経由することにより取得し得た、前のバージョンにアクセスしていたであろう。

【0174】

この方法は、第1のメディアータ、第2のメディアータ、第1の非キャッシュ記憶媒体および第2の非キャッシュ記憶媒体を含む、システムによって実装され得る。第1のメディアータ、第2のメディアータならびにデータファイルを格納するための第1および第2の記録媒体の各々は、持続性媒体を含むか、基本的にそれから成るか、またはそれから成り得る、別個の装置上に格納され得る。追加として、第1のメディアータは、ホストから導出されるファイル名を、第1の記録媒体の第1のLUNと相互に関連付け、第2のメディアータは、同じファイル名を、第2の記録媒体上の第2のLUNと相互に関連付ける。いくつかの実施形態では、第1の非キャッシュ媒体内に格納されている、最新のファイルは、従来のファイルが第2の非キャッシュ媒体内に有するのと同じLUNを有する。

【0175】

データ取得

【0176】

本発明の方法のいずれかによれば、変換された形式で格納されている任意のデータは、それをホストに返す前に、取得されて復号されることが可能である。変換されたデータを取得すること、前述の参照テーブルまたは頻度変換器にアクセスすること、ならびに均一のビット列およびチャンクレットに変換して戻すことを可能にする1つ以上のアルゴリズムの使用を通じて、ファイルがホストのために再作成できる。限定されない例として、データが変換されて、1つのマーカーがどこで終わるかの指示（例えば、一意のビット列の使用または均一サイズのマーカーの使用を通じて）を含むフォーマットで格納されている可能性がある。

【0177】

格納されたとおりのデータの取得は、現在知られているかまたは知られるようになり、かつ当業者が、本発明と関連して利用されるとして理解する、プロセスおよび技術を通し得る。任意選択で、マネージャは、ファイルの格納および取得を調整する。いくつかの実施形態では、マーカーなどのNCMからのデータは、並列処理を通して取得される。

【0178】

一方法では、記録媒体にアクセスすることにより始まる。記録媒体は、複数のマーカーを順番に格納して、これらのマーカーからファイルを再作成できる。アクセスは、ファイルの取得を要求して、その要求をストレージエリアネットワークに伝送するホストによって、またはストレージエリアネットワークの管理者によって開始され得る。

【0179】

データが、メディアータ（存在する場合）によって識別される位置から記憶媒体から取得された後、データが変換されている場合は、複数のマーカー（または頻度変換器を通して変換されているデータ）を、I/OストリームまたはI/Oストリームのフラグメント化ユニットに変換して戻されるチャンクレットまたはハッシュ値を形成するために使用され得るビットに翻訳する。マーカーは、各マーカーがチャンクレットに対応するか、または各マーカーがサブユニットに対応して、複数のサブユニットが結合されてチャンクレットを形成し得るように、格納され得る。格納されたフォーマットでは、マーカーは、所望の文書またはファイルの再作成を可能にする方法で、チャンクレット（またはハッシュ値）内のビットの再作成およびチャンクレット（またはハッシュ値）の順序の再作成を可能にする順序で配置される。

【0180】

データが変換される場合は、マーカーをチャンクレットに翻訳するために、ビットマーカーテーブルまたは頻度変換器にアクセスし得る。ビットマーカーテーブルまたは頻度変換器内では、各一意のビット列またはファイル内の各一意のビット列と関連付けられている一意のマーカーがあり得る。テーブルが、テーブル 2 に類似したフォーマットで編成されている場合、翻訳後、各サブユニットおよびチャンクレットが同じサイズを有するために、ゼロが追加され得る。復号時に、ビットマーカーテーブルまたは頻度変換器を、コード化のためにこれを使用するのは逆に使用する。任意選択で、同じテーブルを使用して、入力および出力を逆転する代わりに、別個のテーブルを使用し得る。

【0181】

他の実施形態と同様に、各チャンクレットは N ビット長であり得、 N は 1 よりも大きい整数であり、また、各サブユニットは A ビット長であり、 A は整数である。マーカーをチャンクレットに翻訳するために、ビットマーカーテーブルまたは頻度変換器にアクセスし得る。

【0182】

チャンクレットが形成された後に、文書がそれから再構成できるバイナリデータに対応する出力を有する。任意選択で、ファイルをファイルタイプと関連付け得る。例えば、ホストは M I M E 変換器を追跡して、返す際にそれをファイルと再度関連付け得る。ファイルタイプは、データの受信者に、それをオープンするためにどのオペレーティングシステムを使用すべきかを知るように指示する。当業者であれば認識するように、ストレージエリアネットワークは、ファイルタイプを追跡する必要がなく、いくつかの実施形態では、行わない。

【0183】

チャンクレットが形成された後に、いくつかの実施形態では、ファイルがそれから再構成できるバイナリデータに対応する出力を有する。他の実施形態では、変換されたデータは、ハッシュ値に対応し、ハッシュ値はまず、I/O ストリームのフラグメント化ユニットまたは I/O ストリーム自体を作成するために使用される必要がある。

【0184】

切り捨てられたデータに関しては、ビットマーカーテーブルまたは頻度変換器を利用する必要がない場合でさえ、データを取得し得る。記録媒体にアクセスすることによりそれを行い得、記録媒体は、複数のデータユニットを複数の位置に格納し、各データユニットは、複数のビットを含み、データユニットの最大サイズは第 1 のビット数であり、少なくとも 1 つのデータユニットは、第 2 のビット数を含み、第 2 のビット数は第 1 のビット数よりも小さい。

【0185】

次に、データユニットを取得して、1 つ以上のビット、例えば、0 を、 N ビット長よりも少ない任意のデータユニットの端部に追加して、データユニットに対応するチャンクレットのセットを生成し得、各チャンクレットは、同じビット数を含み、データユニットの順番に対応する順番にチャンクレットのセットを含む出力を生成する。切り捨てられたデータがゼロを除去することにより形成された場合には、データを取得する際に、ゼロを追加して戻すであろう。加えて、格納されたデータユニットがチャンクレットのサブユニットであった場合、システムはまず、均一サイズのサブユニットを生成するために、切り捨てられたサブユニットにゼロを追加して戻し、次いでサブユニットを結合してチャンクレットを形成し得る。

【0186】

ルックアップテーブルが使用される場合には、それは、好ましくは、コンピューティング装置のメモリ内に格納される。いくつかの実施形態では、ルックアップテーブルは静的であり、マーカーは事前に決められている。従って、1 つ以上の異なる文書タイプの複数の文書を長期間にわたって格納している場合、同じテーブルが使用され得る。任意選択で、それは、ホストの位置に、またはストレージエリアネットワークの一部として、格納され得る。

【 0 1 8 7 】

ホストからのデータの受信

【 0 1 8 8 】

様々な実施形態をさらに詳しく説明するため、およびコンテキストを提供するために、以下で、使用し得る特定のハードウェアが参照されるが、ハードウェアは、組み合わせられて、本発明の方法を実装するためのシステムを形成する。ハードウェアは、前述のプロセスが自動化されて、1つ以上のデータファイルをホストから受信すると、1つ以上のプロセッサによって制御されるのを可能するように構成され得る。

【 0 1 8 9 】

いくつかの実施形態では、ホストは、第1の位置において任意の方法で、文書およびコンピュータファイルを生成し得る。文書は、ホストのオペレーティングシステムによって生成されて、ホストのファイルシステムによる格納のために編成される。ホストのオペレーティングシステムは、そのメモリ内にローカルに、ファイル名を格納し得る。本発明は、オペレーティングシステムのタイプまたはホストが使用するファイルシステムによって制限されない。限定されない例として、ホストは、以下のハードウェア構成要素：メモリ、記憶装置、入力装置、出力装置、グラフィックユーザーインタフェース、1つ以上の通信ポータル、および中央処理装置、の1つ以上を有する、ネットワーク内のコンピュータまたはコンピュータのセットを含み得る。

10

【 0 1 9 0 】

その第1の位置において、S A Pは、文書またはファイルと相互に関連付けるデータを格納するためのプロトコルを実行する。S A Pは、データをI / Oストリームまたは、例えば、サイズが4 Kのチャンクレットにフォーマットする。

20

【 0 1 9 1 】

データは、S A Nを経由して、1つ以上のモジュールを有するコンピュータもしくはネットワークに、またはデータを受信するように構成されているコンピュータもしくはコンピュータのセットに送信され得る。この受信コンピュータは、以下のハードウェア構成要素：メモリ、記憶装置、入力装置、出力装置、グラフィックユーザーインタフェース、中央処理装置、ならびに1つ以上のホストおよび1つ以上の記憶装置とローカルに、および/またはネットワークを経由して、情報の通信を可能にするように構成されている、1つ以上の通信ポータル、のうちの1つ以上を含み得る。

30

【 0 1 9 2 】

加えて、実行可能コンピュータコードを、ハードウェア、ソフトウェア、またはハードウェアとソフトウェアの組合せ上に格納する、コンピュータプログラム製品があり得る。コンピュータプログラム製品は、本発明の方法を実施するように構成されている1つ以上のモジュールに分割されるか、またはそれらと通信可能であり得、1つ以上の持続性媒体内に格納され得る。

【 0 1 9 3 】

例えば、レベル1 (L 1) キャッシュおよびレベル2 キャッシュ (L 2) があり得る。当業者であれば承知のとおり、キャッシュ技術の使用は、慣例的に、データ格納における効率の向上を可能にしてきた。本発明では、例として、データがS A Nを経由してキャッシュに送信され得、データは、ハッシュ関数アルゴリズムにアクセスする前、ビットマーカータブルを調べる前、頻度変換器を調べる前、およびビットを切り捨てる前、ならびに/またはハッシュ関数アルゴリズムを調べた後、ビットマーカータブルを調べた後、頻度変換器を調べた後、およびビットを切り捨てた後に、キャッシュに送信され得る。

40

【 0 1 9 4 】

セクターサイズが5 1 2 Bであると仮定して、サイズが4 Kである各チャンクレットに対して、ホストは、8セクターの記憶装置が使用されると予期するであろう。

【 0 1 9 5 】

システム

【 0 1 9 6 】

50

本発明の様々な実施形態は、データ記憶および取得システムを提供する。一実施形態では、システムは、非キャッシュデータ記憶媒体、メディアータおよびマネージャを含む。これらの要素間、および任意選択で、イニシエータとの通信は、有線、無線またはそれらの組合せのネットワークを経由し得る。

【0197】

非キャッシュデータ記憶媒体は、例えば、1つ以上のディスクまたはソリッドステートドライブを含むか、基本的にそれらから成るか、またはそれらから成り得る。使用時に、非キャッシュデータ記憶媒体は、本発明の方法の1つ以上に従って処理されていないファイルと比べて、少なくとも50%、少なくとも60%、少なくとも70%、または少なくとも80%の空間節約で、ファイルを格納し得る。

10

【0198】

メディアータは、トラックの第1のセット、トラックの第2のセット、トラックの第3のセット、およびトラックの第4のセット：のトラックの4つのセットを含むか、基本的にそれらから成るか、またはそれらから成り得る。メディアータは、好ましくは、持続性媒体上に格納されて、非キャッシュデータ記憶媒体からリモートに配置される。従って、メディアータおよび非キャッシュデータ記憶媒体は、好ましくは、同じ装置の一部ではない。

【0199】

システムは、マネージャも含み得る。マネージャは、受信の制御、格納および取得の処理、ならびにメディアータを通じたデータの伝送を提供し得る。それ故、マネージャは、好ましくは、ホストおよびメディアータに動作可能に結合され、任意選択で、非キャッシュデータ記憶媒体に動作可能に結合される。さらに、いくつかの実施形態では、マネージャは、メディアータ、非キャッシュ媒体およびホストの各々からリモートに配置される。

20

【0200】

マネージャは、以下の機能：(a) ファイルシステム情報、ブート可能性情報、およびパーティション情報の1つ以上を含むデータをメディアータのトラックの第1のセット内に格納すること、(b) メタデータをメディアータのトラックの第3のセット内に格納すること、(c) 1つ以上のファイルを非キャッシュ媒体上に格納することであって、1つ以上のファイルが非キャッシュ媒体上に、ファイルシステム情報、ブート可能性情報、およびパーティション情報のいずれもなしで保存される（従って、いくつかの実施形態では、生データのみが非キャッシュ媒体上にある）、1つ以上のファイルを非キャッシュ媒体上に格納すること、(d) メディアータのトラックの第4のセット内に、非キャッシュ媒体内の各ファイルの位置を格納すること、ならびに(e) 非キャッシュ媒体内の各ファイルの位置の、ファイルに対するホスト名との相関関係をメディアータ上に格納すること：の1つ以上を実施するように構成され得る。好ましくは、非キャッシュ媒体内の各ファイルの位置の、ファイルに対するホスト名との相関関係が、トラックの第4のセット内に格納され、それは、ビットフィールドに対応する。加えて、マネージャは、本発明の方法の1つ以上を実行することが可能な実行可能命令を含むコンピュータプログラム製品を含むか、またはそれに動作可能に結合され得る。

30

【0201】

本発明の様々なシステム内で、SANは、コンピュータプログラム製品内に格納された指示に従って、ビットマーカータブルまたは頻度変換器にアクセスし得る。これらの資源は、ビットマーカータブルをサブユニットの各々と相互に関連付けて、出力を生成する。ビットマーカータブルの、全部ではないが、ほとんどが、サブユニットよりもサイズが小さいので、出力は、ホストから受信された入力ファイルよりも小さいデータファイルである。従って、ホストから受信されたとおりのファイルがサイズRであり得、一方、SANによって保存されたとおりの実際のデータがSであり得、 $R > S$ である。好ましくは、RはSの少なくとも2倍の大きさであり、さらに好ましくは、RはSの少なくとも3倍の大きさである。

40

【0202】

50

SANは、出力ファイルを受け取って、それを持続性記憶媒体、例えば、非キャッシュ媒体内に格納する。好ましくは、SANは、格納されたとおりのファイルを、ホストから受信されたとおりのファイルと相互に関連付けて、ホストがファイルを取得できるようにする。

【0203】

さらに詳細な説明のために、本発明の方法を実装するためのシステムを示す、図1が参照され得る。システム10において、ホスト1は、メモリ4に動作可能に結合されているプロセッサ3を含む、ストレージエリアネットワーク6にファイルを送信する。任意選択で、ストレージエリアネットワークは、受信の確認をホストに返す。

【0204】

メモリ内に、チャンクレットを取得して、その中に含まれるデータをサブユニットに分割するように設計されているコンピュータプログラム製品が格納される。メモリは、参照テーブル5も含むか、またはそれに動作可能に結合され得る。テーブルは、ビットマーカもしくは頻度変換器またはサブユニットの1つ以上に対する関連ファイルを含み、コンピュータプログラム製品は、元のサブユニットの代わりに1つ以上のマーカを含む、新しいデータファイルを作成する。

【0205】

プロセッサは次に、ビットマーカを、非キャッシュ媒体などの、記録媒体（例えば、ディスク2であり得る）上に格納させる。いくつかの実施形態では、当初は、マーカの全てが同じサイズである。しかし、いくつかの実施形態では、それらを格納する前に、1つ以上、好ましくは、少なくとも25%、少なくとも50%、または少なくとも75%が、格納前に切り捨てられる。

【0206】

図2は、 R_1 40および R_2 50、ならびに、ビットフィールド60およびメタデータファイル30のための空間を含む、メタデータ70を有するシステム100を示す。メタデータの表現は、例示のみを目的としており、メタデータの構造またはその中の編成に制限を課さない。非キャッシュ媒体(NCM)20も示されている。非キャッシュ媒体は、メタデータの近くに示されているが、それらは別個の構造である。いくつかの実施形態では、メタデータはNCMから離れた装置に置かれ、本発明の装置の1つ以上が可搬式である。マーカまたは他の変換された生データがNCM上に格納される。

【0207】

図3は、別のシステム200を示す。このシステムでは、イニシエータ(I^n)270がチャンクレットをキャッシュマネージャ230に伝送し、キャッシュマネージャ230は、任意選択で、データファイルのコード化を準備して、それらをメタデータ210に伝送する。ホストの例は、Microsoft Windows（登録商標）Server and Desktop、Apple OS X、Linux（登録商標）RHEL、およびSUSE Oracle Solaris、IBM AIX、HP UXならびにVM EXSおよびESXiを実行するコンピュータまたはコンピュータネットワークを含むが、それらに制限されない。データファイルに対応する情報が、当初は、 R_1 240に送信されるが、それは、以前はイニシエータが定義したパラメータが投入されていた。メタデータ自体は、ビットマーカテーブルまたは頻度変換器（図示せず）の使用を通して情報を翻訳し得るか、またはリモート符号器（リモート変換器とも呼ばれ得、図示せず）と通信し得、また、メタデータは R_1 内ならびに R_2 250内に、ホストから受信されるファイル名のコピーを格納する。データが変換されて、もっと小さいサイズのファイルが作成された後、ビットフィールド260のセクターマップ内に、ファイルがディスク220内に格納される位置または複数の位置が記録される。コード化されたデータが位置285に格納され得る。データのコード化前またはデータのコード化の代わりに、ハッシュ値アルゴリズムがアクセスされ得、チェックサムが格納または変換のために使用され得る。

【0208】

図4は、図3のシステムの実施形態の変形形態であり、記憶装置のバックアップを提供する、別のシステム300を示す。このシステムでは、イニシエータ370が、チャンクレットをキャッシュマネージャ330に伝送し、キャッシュマネージャ330は情報を、データを含む第1のメディアエータ310に転送して、図3に対して送信された同じファイルを修正する。修正されたファイルの受信前またはその受信後のいずれかであるが、それを非キャッシュ媒体内に格納する前に、第2のメディアエータ380が第1のメディアエータ310から作成される。第2のメディアエータは、それが作成された時における第1のメディアエータの正確なコピーであり、ファイル名に対しては、その時、非キャッシュ媒体320内の同じセクター（またはセクタークラス）385を指す。

【0209】

10

第1の修正されたファイルが、第1のメディアエータのR₁ 340で受信される。第1のメディアエータは、この場合もやはり、ビットマーカータブルまたは頻度変換器（図示せず）の使用を通して情報を翻訳するか、またはリモート符号器と通信するかのいずれかであろう。メディアエータは、R₁内ならびにR₂ 350内に、ホストから受信されるファイル名のコピーを継続して格納する。データが変換されて、もっと小さいサイズのファイルが作成された後、第1のメディアエータのビットフィールド360のセクターマップ内に、ファイルがディスク320内に格納される位置が記録される。しかし、修正されたファイルは異なるセクター395内に格納される。従って、第1のメディアエータに対する変更は、第2のメディアエータに対しては行われまいであろう。

【0210】

20

ホストは、デフォルト設定により、第1のメディアエータと通信する。従って、ホストがファイルを記憶装置から呼び戻したい場合、第1のメディアエータが、そのデータをセクター395から呼び戻すであろう。ホストまたはシステム管理者がデータの以前のバージョンを取得したい場合には、ファイル名を、セクター385を目指す、第2のメディアエータに投入し得る。

【0211】

本発明の別の実施形態によれば、以前にメタデータを本発明のシステムに（例えば、オペレーティングシステム情報、ブート可能性情報、パーティション情報、文書タイプ情報QoS情報など）提供しているイニシエータは、文書に対応するビットをキャッシュマネージャに送信する。ビットは、例えば、チャンクレットに編成され得る。

30

【0212】

キャッシュマネージャは、情報をL1キャッシュに、L2キャッシュに、およびメディアエータに送信し得る。キャッシュマネージャは受信の確認もイニシエータに送信し得る。

【0213】

その上に格納された関連メタデータを既に有し得る、メディアエータは、文書に対応するビットを変換器に送信し、変換器は、ビットをコード化情報に変換する。変換器と関連付けられているか、またはその一部は、計算機でもあり得、それは、変換されたビットのサイズを判断する。変換は、例えば、ビットマーカータブルまたは頻度変換器の使用を通して得る。

【0214】

40

変換器は次いで、メディアエータに、変換されたファイルのサイズを示し、メディアエータは、変換されたファイルが非キャッシュ媒体内に格納される位置を判断し得る。このステップに続いて、メディアエータはその位置に格納させ得る。

【0215】

本明細書で説明する様々な実施形態の特徴のいずれも、特別の定めのない限り、開示する任意の他の実施形態に関連して説明する特徴と併用できる。従って、様々な、または特定の実施形態に関連して説明する特徴は、かかる排他性が明記されていないか、またはコンテキストから暗黙的でない限り、本明細書で説明する他の実施形態に関連して適切でないと解釈されるべきでない。

【0216】

50

例

【 0 2 1 7 】

例 1 : ビットマーカータブル (予測的)

【 0 2 1 8 】

参照ロケータテーブル内で、各一意のマーカが、一意のビット列に対応するとして識別される。テーブルは、テーブル格納用として一般に知られているか、または知られるようになり、かつコンピュータアルゴリズムが、各入力に対して割り当てられている出力を取得するのを可能にする、任意のフォーマットで格納され得る。

【 0 2 1 9 】

以下のテーブル I は、サブユニットが 8 ビット長である、ビットマーカータブルからの抜粋例を提供する。

【 0 2 2 0 】

【表 1】

テーブル 1

ビットマーカ (格納されているとおり)	サブユニット=8 ビット (入力)
0101	00000001
1011	00000010
1100	00000011
1000	00000100
1010	00000101
11111101	11111101

【 0 2 2 1 】

例として、テーブル 1 で識別されたサブユニットを使用して、入力が 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 である場合、出力は： 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 1 0 1 である。ビットマーカ出力がサブユニット入力よりも小さい場合、それは記憶媒体上で少ない空間しか占めず、それによりビットの格納に必要な記憶空間および時間の両方を節約する。

【 0 2 2 2 】

当業者であれば認識するように、テーブル 1 を生成するために抜粋されたものなどの所与のビットマーカータブルでは、ビットの全ての組合せが使用される場合、 2^N のエントリが必要であり、N は、サブユニット内のビット数に対応する。8 ビットがある場合には、 2^8 のエントリが必要とされる。サブユニット内に 16 ビットがある場合は、 2^{16} のエントリが必要であり、それは 65,536 のエントリに等しい。サブユニット内に 32 ビットある場合は、 2^{32} のエントリを必要とし、それは 4,294,967,296 のエントリに等しい。あるビット列がファイル内で使用されないことが分かっている場

合には、テーブルは最小のものから始まるマーカーを割り当て得る。

【 0 2 2 3 】

例 2：前処理されたサブユニットのためのビットマーカーテーブル（予測的）

【 0 2 2 4 】

サブユニットサイズが大きくなるにつれて、テーブルがより煩雑になるので、いくつかの実施形態では、テーブルは、サブユニット列の 1 つの端部から全てのゼロが欠落するように構成され得、テーブルにアクセスする前に、各サブユニットのその端部から全てのゼロが除去される。従って、テーブル 1 がそれから抜粋されるテーブルではなく、テーブル 2 が抜粋されるテーブルが構成され得る。

【 0 2 2 5 】

10

【表 2】

テーブル 2

ビットマーカー（出力）	前処理されたサブユニット
0101	00000001
1011	0000001
1100	00000011
1000	000001
1010	00000101
11111101	11111101

20

30

【 0 2 2 6 】

このように、第 2 行および第 4 行で、サブユニットが前処理された後、それらは 8 ビットより少ないビットを有した。しかし、ホストから受信した生データ内の実際のサブユニットは全て、8 ビットを有した。方法が実装されるシステムは、桁の欠如はゼロを暗示しており、全ての桁の欠如は、任意の切り捨てられたサブユニットの同じ端部においてであると理解するように設計できるので、少ない空間を占めて、一意のマーカーを一意のサブユニットに割り当てる能力を保持するテーブルを使用できる。従って、方法は、システムが 0 0 0 0 0 0 0 1（7 つのゼロと 1 つの 1）および 0 0 0 0 0 0 1（6 つのゼロと 1 つの 1）を異なるとして解釈するのを可能にする。

40

【 0 2 2 7 】

この方法を実装するために、各サブユニット（または、サブユニットが使用されていない場合には各チャクレット）が第 1 の端部および第 2 の端部を有すると考え得る。第 1 の端部は、ビット列の右側または左側のいずれかにでき、第 2 の端部はその反対側であろう。説明のために、第 1 の端部を左端の桁と、また、第 2 の端部を右端の桁と考え得る。この方法の下で、次いで、各チャクレットの各サブユニット内の 1 つ以上のビットを分析して、第 2 の端部にあるビットが値 0 を有するかを判断する。このステップは、前処理と呼ばれ得、前処理された後のサブユニットは、テーブル 2 の右側の列に現れる。第 2 の端部におけるビットが値 0 を有する場合、本方法は、第 2 の端部におけるビットおよび値

50

0を有して、そのビットとともに連続的なビット列を形成し得る全てのビットを除去し、それにより、元は第2の端部に0を有していた任意のサブユニットに対して、修正されたサブユニット(テーブル内の前処理されたサブユニット)を形成する。

【0228】

各サブユニットを再検討して第2の端部に0があるかどうかを判断し、そうである場合は0を除去して前処理されたサブユニットを形成するアルゴリズムを使用し得、前処理されたサブユニットは、サブユニットの第2の端部に隣接していた位置に、修正された第2の端部をもつ修正されたサブユニットとも呼ばれ得る。次に、アルゴリズムは、修正されたサブユニットを再検討して、その現在の修正された第2の端部において0があるかを判断し、そうである場合は0を除去して、さらに修正された第2の端部を形成する。この方法では、修正された第2の端部は、以前は、第2の端部におけるビットに隣接していた位置であろう。任意のさらに修正された第2の端部は、サブユニットの第2の端部から2つ以上離れた位置になっているであろう。従って、用語「修正された(revised)」は、短縮されたか、または切り捨てられた第2の端部を意味する。アルゴリズムは、この方法を、その第2の端部に1をもつ、短縮されたチャンクレットが生成されるまで、修正されたサブユニットに対して繰り返し得る。

10

【0229】

当業者であれば認識するように、前述の方法は、1が、修正された第2の端部またはさらに修正された第2の端部になるまで、ゼロを第2の端部から除去することにより、適用されていると説明されている。本方法は、0が、修正された第2の端部またはさらに修正された第2の端部になるまで、システムが1を第2の端部から除去するように、逆に設計され得る。加えて、本開示を用いて、当業者は、ビットを、第2の端部の代わりに、第1の端部から除去し、それらの修正されたサブユニットをビットマーカに変換するために作成されたテーブルを使用し得る。

20

【0230】

例3：頻度交換(予測的)

【0231】

経験的分析に基づき、特定のホストから受信された文書もしくは文書のセットのタイプ内、または、所与の時間フレーム(例えば、過去1年または過去2年)内に受信されている文書のセット内からの、各サブユニットの頻度を判断できる。サブユニットが番号順に編成されているテーブル1またはテーブル2に例示するようなテーブルに頼るのではなく、この情報を用いて、さらに小さいビットマーカが、ファイル内、文書のタイプ内、または特定のホストから受信したとおりの文書のセット内に、最も出現しそうであると予測されているサブユニットと関連付けられている頻度変換器に頼ることができる。従って、頻度変換器内で、マーカは複数の異なるサイズであり、さらに小さいサイズのマーカが、より高い頻度のサブユニットと相互に関連づけられる。

30

【0232】

【表 3】

テーブル 3：頻度変換器

ビットマーカ (出力)	頻度	サブユニット=8ビット (入力)
0101	16%	00000001
1000	15%	00000010
11011	10%	00000011
10011101	0.00001%	00000100
10111110	0.00001%	00000101
1100	15%	11111101

10

20

【0233】

テーブル 3 は、テーブル 1 と同じサブユニットを使用する頻度変換器からの抜粋例である。しかし、ビットマーカは順番に割り当てられておらず、代わりに、より大きなビットマーカが低頻度のサブユニットに割り当てられていることに気付くであろう。テーブルに示すように、サブユニット 0 0 0 0 0 0 1 1 に割り当てられているマーカは、サブユニット 0 0 0 0 0 0 0 1 に割り当てられているものよりも、25 パーセント大きく、サブユニット 1 1 1 1 1 1 0 1 に対しては、高い数値であるにも関わらず、それは、特定のホストから受信したファイルのタイプ内により頻繁に出現するので、より小さいビットマーカを受信する。従って、テーブル 1 を使用して、サブユニット 1 1 1 1 1 1 0 1 が 1 0 , 0 0 0 箇所に出現する場合、それは 1 1 1 , 1 1 1 , 0 1 0 , 0 0 0 ビットに対応するであろう。しかし、テーブル 3 を使用した場合には、同じ情報を格納する目的に対して使用するために、1 1 , 0 0 0 , 0 0 0 ビットが必要であろう。この方法には示していないが、サブユニットは、ゼロを一方の端部または他方から除去するために前処理でき、テーブルは相関する切り捨てられたサブユニットを含むように設計され得る。

30

【0234】

前述のように、頻度変換器は、1 つ以上のホストから受信しそうなデータを代表すると考えられるファイルのセットの分析に基づき生成できる。いくつかの実施形態では、情報を処理するアルゴリズムは、それ自身の品質管理を実行して、所与の期間からの文書に対するサブユニットの実際の頻度を、頻度変換器内のマーカの割当てに基づいているものと比較する。統計的分析を使用して、それは、次いで、将来の使用のために、どのようにマーカがサブユニットと関連付けられているかを割り当てる、新しいテーブルが作成されるべきかを判断し得る。当業者であれば認識するように、テーブル 3 は、頻度変換器の単純化した抜粋である。しかし、実際には、相関関係を得るために 16 進記数法を選択し得る。加えて、テーブルに基づく頻度の列挙は、読者の便宜のために含まれており、それらは、本発明の様々な実施形態によってアクセスされる際にテーブルに含まれる必要はない。

40

50

【 0 2 3 5 】

例 4 : メディエータ内での空間の割当て (予測的)

【 0 2 3 6 】

1 M B のサイズの仮説の記録媒体では、当業者は、セクターを次のようにマッピングし得る：

【 0 2 3 7 】

1 MB の記録媒体は、1,024,000 バイトを有し、それは 250 セクターに対応する (1,024,000 / 4096 = 250)。記録媒体のジオメトリは、次のように要約され得る：ボリューム = (c * h * s p t * s s)、ここで

【 0 2 3 8 】

$$c \text{ (シリンダー数)} = 7 ;$$

【 0 2 3 9 】

h (ヘッド数) = 2 5 5 ;

【 0 2 4 0 】

s p t (トラックあたりのセクター) = 63 ; および

【 0 2 4 1 】

ss (バイト単位でのセクターサイズ) = 4096 とする。

【 0 2 4 2 】

メディアータ内では、セクターは次のように割り当てられ得る：

【 0 2 4 3 】

【表 4】

テーブル 4

アドレス	実際の非キャッシュ媒体 LBA
0-15	メディアータ<<予約済み 1>> “ブートセクター 0” + 1 5
16-31	メディアータ位置<<予約済み 2>> Sys_Internal のみ
32-35	メディアータ_メタデータ
36	マップデータ “LBA-nnnnnnnnnnnnnnnn”
37	マップデータ “LBA-nnnnnnnnnnnnnnnn”
...	マップデータ “LBA-nnnnnnnnnnnnnnnn”
250	マップデータ “LBA-nnnnnnnnnnnnnnnn”

【 0 2 4 4 】

例 5 : 空間の節約

【 0 2 4 5 】

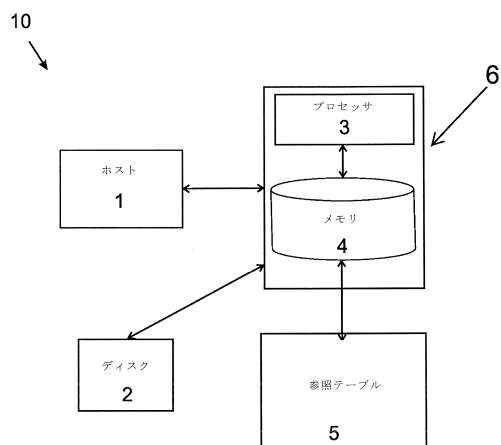
本発明のシステムは、4096バイトのI/Oストリーム内で4250万ブロックのデータを受信した。システムは、MD5ハッシュアルゴリズムを適用して、I/Oストリー

ム内の４２５０万ブロックに対応する７８０万ブロックのハッシュ値を生成する。

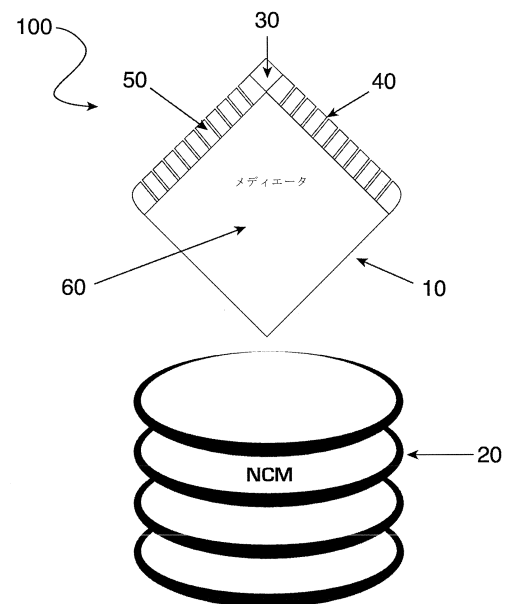
【０２４６】

これは、元の４２５０万ブロックを格納するために必要とされていた空間の１８．５％だけの使用に翻訳された。矛盾モジュールが適用されて、矛盾が存在しないことが検証された、すなわち、異なるブロックに対して、ハッシュ値の重複が生成されなかった。

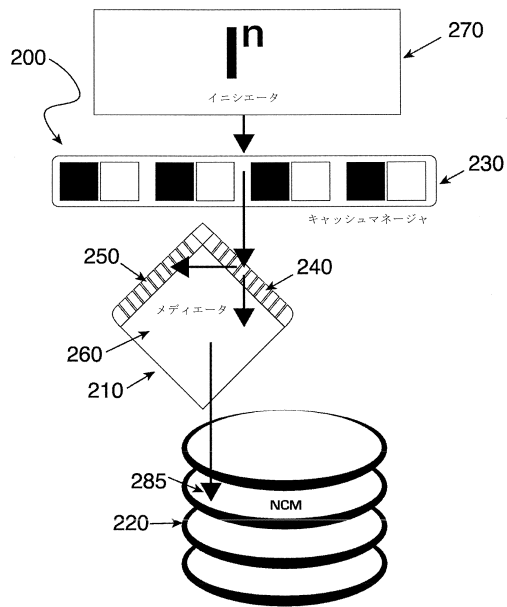
【図１】



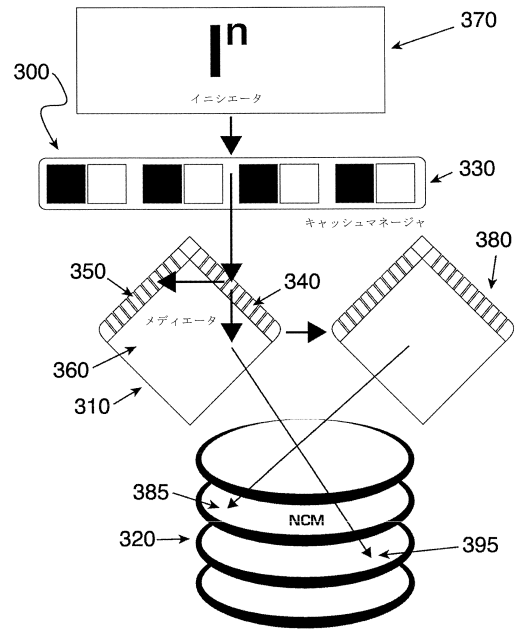
【図２】



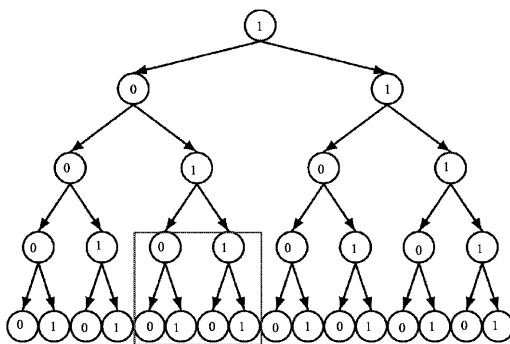
【図 3】



【図 4】



【図 5】



【図 6】

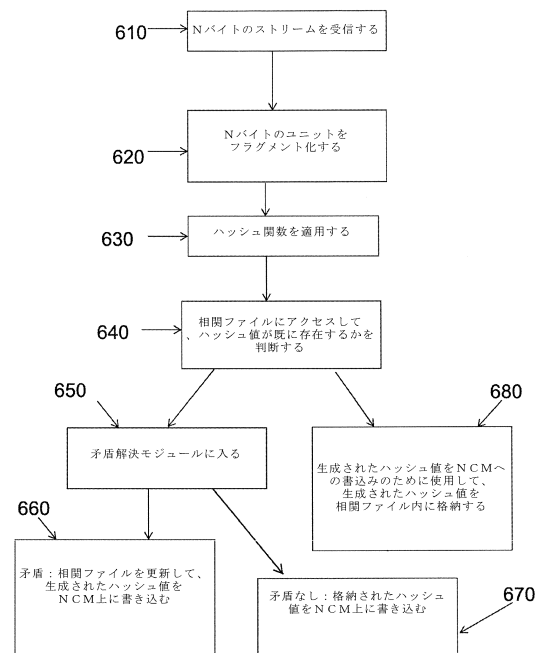


Figure 5

フロントページの続き

(31)優先権主張番号 13/908,239

(32)優先日 平成25年6月3日(2013.6.3)

(33)優先権主張国 米国(US)

(72)発明者 イグノミレロ, ブライアン

アメリカ合衆国, ニュージャージー州 07722, コルツ ネック, 40 マナー ロード

審査官 田上 隆一

(56)参考文献 特開2012-027587(JP, A)

米国特許出願公開第2009/0055593(US, A1)

国際公開第2012/117658(WO, A1)

(58)調査した分野(Int.Cl., DB名)

G06F 3/06