(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7700128号
(P7700128)

(45)発行日　令和7年6月30日(2025.6.30)

(24)登録日　令和7年6月20日(2025.6.20)

(51)国際特許分類
Ｇ０６Ｆ　16/174(2019.01)

ＦＩ
Ｇ０６Ｆ　16/174

請求項の数　13　（全38頁）

(21)出願番号　特願2022-537814(P2022-537814)
(86)(22)出願日　令和2年12月17日(2020.12.17)
(65)公表番号　特表2023-510134(P2023-510134A)
(43)公表日　令和5年3月13日(2023.3.13)
(86)国際出願番号　PCT/US2020/065711
(87)国際公開番号　WO2021/127245
(87)国際公開日　令和3年6月24日(2021.6.24)
審査請求日　令和5年12月14日(2023.12.14)
(31)優先権主張番号　16/718,686
(32)優先日　令和1年12月18日(2019.12.18)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　16/718,703
(32)優先日　令和1年12月18日(2019.12.18)
最終頁に続く

(73)特許権者　522242030
グラニカ・コンピューティング・インコーポレイテッド
アメリカ合衆国、カリフォルニア・９４０４３、マウンテン・ビュー、モフェット・ブールバード・２７５
(74)代理人　100114188
弁理士　小野 誠
(74)代理人　100119253
弁理士　金山 賢教
(74)代理人　100124855
弁理士　坪倉 道明
(74)代理人　100129713
弁理士　重森 一輝
(74)代理人　100137213
最終頁に続く

(54)【発明の名称】　スケッチ計算のためのシステムおよび方法

(57)【特許請求の範囲】
【請求項１】
システムが実行するデータ削減のための方法であって、
（ａ）１つまたは複数のクライアントアプリケーションから１つまたは複数の入力データストリームを受信するステップと、
（ｂ）前記１つまたは複数の入力データストリームから少なくとも第１のセグメントおよび第２のセグメントを生成するステップであって、前記第１のセグメントは第１の複数のチャンクを含み、前記第２のセグメントは第２の複数のチャンクを含む、ステップと、
（ｃ）（ｉ）前記第１のセグメントの第１のスケッチおよび（ｉｉ）前記第２のセグメントの第２のスケッチを計算するステップであって、前記第１のスケッチは、前記第１のセグメントを表すかまたは前記第１のセグメントに固有の第１の特徴のセットを含み、前記第１の特徴のセットは、前記第１の複数のチャンクから選択されたチャンクの第１のサブセットを使用して計算され、前記第２のスケッチは、前記第２のセグメントを表すかまたは前記第２のセグメントに固有の第２の特徴のセットを含み、前記第２の特徴のセットは、前記第２の複数のチャンクから選択されたチャンクの第２のサブセットを使用して計算され、前記第１の特徴のセットは前記第１のセグメントに対応し、前記第２の特徴のセットは前記第２のセグメントに対応する、ステップと、
（ｄ）前記第１のスケッチおよび前記第２のスケッチを処理して、前記第２のセグメントが前記第１のセグメントと類似しているかどうかを示す類似度指標を生成するステップと、

（ｅ）（ｄ）に続いて、（１）前記類似度指標が類似度閾値以上である場合に前記第１のセグメントに対する前記第２のセグメントに対して差分演算を実行し、前記第１のセグメントと前記第２のセグメントとの間で一致するフィンガープリントを有するチャンクの群を決定し、当該チャンクの群に対するシングルポインタを生成して格納し、（２）前記類似度指標が前記類似度閾値未満である場合に前記差分演算を実行せずに前記第１のセグメントおよび前記第２のセグメントをデータベースに格納するステップと
を含む、方法。
【請求項２】
前記ステップ（ｅ）の差分演算が、（ｉ）前記第１のセグメントの前記第１の複数のチャンクの参照ハッシュセットを生成するステップと、（ｉｉ）前記参照ハッシュセットをメモリテーブルに格納するステップと、を含む、請求項１に記載の方法。
【請求項３】
前記参照ハッシュセットが弱いハッシュを含む、請求項２に記載の方法。
【請求項４】
前記差分演算が、（ｉｉｉ）前記第２のセグメントの前記第２の複数のチャンクについて第２のハッシュセットを生成するステップと、（ｉｖ）前記第２のハッシュセットを前記参照ハッシュセットと順序順で比較して、一致があるかどうかを判定するステップと、をさらに含む、請求項２に記載の方法。
【請求項５】
前記差分演算が、一連のシーケンシャルチャンクが前記参照ハッシュセットから一致を見つけるハッシュを有すること、及び、前記一連のシーケンシャルチャンクに後続する後続チャンクが前記参照ハッシュセットから一致を見つけられないことを判定すると、前記第２の複数のチャンクのサブセットから集合的に前記一連のシーケンシャルチャンクを参照するシングルポインタを生成して格納するステップをさらに含む、請求項４に記載の方法。
【請求項６】
前記シングルポインタは、縮小されたポインタのセットを含むスパースインデックスを生成するために部分的に使用される、請求項５に記載の方法。
【請求項７】
前記チャンクの第１のサブセットは、前記第１の複数のチャンクの１０％未満である、請求項１に記載の方法。
【請求項８】
前記チャンクの第１のサブセットは、前記第１の複数のチャンクの１％未満である、請求項１に記載の方法。
【請求項９】
前記類似度閾値が少なくとも約５０％である、請求項１に記載の方法。
【請求項１０】
前記チャンクの第２のサブセットは、前記第２の複数のチャンクの１０％未満である、請求項１に記載の方法。
【請求項１１】
前記チャンクの第２のサブセットは、前記第２の複数のチャンクの１％未満である、請求項１に記載の方法。
【請求項１２】
前記第１の複数のチャンクおよび前記第２の複数のチャンクが可変長である、請求項１に記載の方法。
【請求項１３】
前記類似度指標が類似度スコアである、請求項１に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】

関連出願の相互参照

【０００２】

本出願は、２０１９年１２月１８日に出願された米国特許出願第１６／７１８，６８６号、２０１９年１２月１８日に出願された米国特許出願第１６／７１８，７０３号、および２０１９年１２月１８日に出願された米国特許出願第１６／７１８，７１４号の優先権を主張し、これらの各々は、参照により本明細書に完全に組み込まれる。

【背景技術】

【０００３】

クラウドストレージシステムは、クライアントアプリケーション、例えばエンタープライズアプリケーションからの大量のデータを格納することができる。多くの場合、入力データのかなりの部分が重複している可能性がある。データを格納して処理するために、大量のメモリ、記憶空間、および処理能力が必要とされる場合がある。場合によっては、例えば重複排除または圧縮技術を使用して、データを格納前に削減することができる。しかしながら、小さなチャンクおよび１：１のチャンク比較を使用する大規模なデータ削減は、（書き込みプロセスと読み取りプロセスの両方に計算負荷を加えることによって）著しいメモリ空間を消費し、性能を低下させる生成された大きなインデックステーブルのために、技術的に困難または非実用的であることが示されていた。その結果、従来の重複排除技術は、一般に、数百テラバイトまたはペタバイト規模、特にエンタープライズアプリケーションによって生成された大量のデータのデータ削減に対処することができない。

【発明の概要】

【発明が解決しようとする課題】

【０００４】

データ生成における指数関数的スケーリングにより、本明細書では、計算性能を維持または改善しながら、大規模にデータ削減を効率的に処理することができる方法およびシステムの必要性が認識されている。データ削減は、計算リソースおよびストレージに関する運用コストの削減につながる可能性がある。

【０００５】

本開示は、高スループット（例えば、１日当たり少なくともテラバイトのレベル、または１日当たり１００テラバイト、または１日当たり数テラバイトなど）で大規模で（例えば、少なくともペタバイト規模で）データを最適に削減するように構成されたシステムおよび方法を提供する。本明細書のシステムおよび方法は、さまざまなクライアントアプリケーション、例えばエンタープライズアプリケーションによって生成されたデータに適用され得る。本明細書で使用される場合、「データ」という用語は、任意のタイプのデータ、例えば、構造化データ、非構造化データ、時系列データ、関係データなどを指すことができる。「エンタープライズアプリケーション」という用語は、エンタープライズアーキテクチャを使用して開発され、ビジネスなどの企業環境または政府で動作するように設計された大規模ソフトウェアシステムプラットフォームを指すことができる。本開示のいくつかの実施形態をエンタープライズアプリケーションに関して説明したが、本明細書のいくつかの実施形態を、非エンタープライズアプリケーションまたは他の小規模アプリケーションに適用可能にする、または適合させることができることを理解されたい。

【課題を解決するための手段】

【０００６】

一態様では、本開示は、スケッチ計算のための方法を提供し、方法は、（ａ）１つまたは複数のクライアントアプリケーションから入力データストリームを受信するステップと、（ｂ）入力データストリームから少なくとも１つのセグメントを生成するステップであって、少なくとも１つのセグメントは複数のチャンクを含む、ステップと、（ｃ）少なくとも１つのセグメントのスケッチを計算するステップであって、スケッチは、特徴のセットが少なくとも１つのセグメントに対応するように、少なくとも１つのセグメントを表すまたは少なくとも１つのセグメントに固有の特徴のセットを含み、スケッチは、（ｉ）複数のチャンクのフルインデックスを生成することなく、または（ｉｉ）少なくとも１つの

他の入力データストリームをフルインデックスと比較することなく、１つまたは複数のクライアントアプリケーションから受信した少なくとも１つの他の入力データストリームのインライン重複排除に使用可能である、ステップと、を含む。

【０００７】

別の態様では、本開示は、スケッチ計算のための方法を提供し、方法は、（ａ）１つまたは複数のクライアントアプリケーションから入力データストリームを受信するステップと、（ｂ）入力データストリームから少なくとも１つのセグメントを生成するステップであって、少なくとも１つのセグメントは複数のチャンクを含む、ステップと、（ｃ）少なくとも１つのセグメントのスケッチを計算するステップであって、スケッチは、特徴のセットが少なくとも１つのセグメントに対応するように、少なくとも１つのセグメントを表すまたは少なくとも１つのセグメントに固有の特徴のセットを含み、スケッチは、（ｉ）複数のチャンクのフルインデックスを生成することなく、または（ｉｉ）少なくとも１つの他の入力データストリームをフルインデックスと比較することなく、１つまたは複数のクライアントアプリケーションから受信した少なくとも１つの他の入力データストリームのインライン重複排除に使用可能である、ステップと、を含む。

【０００８】

別の態様では、本開示は、スケッチ計算のための方法を提供し、方法は、（ａ）１つまたは複数のクライアントアプリケーションから入力データストリームを受信するステップと、（ｂ）入力データストリームから少なくとも１つのセグメントを生成するステップであって、少なくとも１つのセグメントは複数のチャンクを含む、ステップと、（ｃ）少なくとも１つのセグメントのスケッチを計算するステップであって、スケッチは、特徴のセットが少なくとも１つのセグメントに対応するように、少なくとも１つのセグメントを表すまたは少なくとも１つのセグメントに固有の特徴のセットを含み、スケッチは、（ｉ）複数のチャンクのフルインデックスを生成することなく、または（ｉｉ）少なくとも１つの他の入力データストリームをフルインデックスと比較することなく、１つまたは複数のクライアントアプリケーションから受信した少なくとも１つの他の入力データストリームのインライン重複排除に使用可能である、ステップと、を含む。

【０００９】

別の態様では、本開示は、スケッチ計算を使用するインラインデータ重複排除のためのコンピュータ実装方法を提供し、方法は、コンピュータネットワークを介して１つまたは複数のクライアントアプリケーションから第１の入力データストリームを受信するステップと、少なくとも１つのコンピュータプロセッサを使用して、第１の入力データストリームから第１のセグメントを生成するステップであって、第１のセグメントは、ハッシュ関数を第１の入力データストリームに適用して、複数のチャンクを生成するための第１の入力データストリーム内の１つまたは複数のブレークを識別し、第１のセグメントを形成するためにターゲットセグメントサイズ範囲に基づいて複数のチャンクを組み立てることによって生成される、ステップと、第１のセグメントのスケッチを計算するステップであって、スケッチは、第１のセグメントを表すか、または第１のセグメントに固有の特徴のセットを含む、ステップと、第１のスケッチと、第２の入力データストリームから生成された第２のセグメントの第２のスケッチとの間の少なくとも類似度を判定することによって１つまたは複数のクライアントアプリケーションから受信された第２の入力データストリームのインラインデータ重複排除のために第１のスケッチを使用するステップと、を含む。

【００１０】

別の態様では、本開示は、スケッチ計算を使用するインラインデータ重複排除のためのコンピュータ実装方法を提供し、方法は、コンピュータネットワークを介して１つまたは複数のクライアントアプリケーションから第１の入力データストリームを受信するステップと、少なくとも１つのコンピュータプロセッサを使用して、第１の入力データストリームから第１のセグメントを生成するステップであって、第１のセグメントは、第１の入力データストリーム内の１つまたは複数のブレークに少なくとも部分的に基づいて複数のチャンクを組み立てることによって生成される、ステップと、第１のセグメントのスケッチ

を計算するステップであって、スケッチは、第１のセグメントを表すか、または第１のセグメントに固有の特徴のセットを含む、ステップと、第１のスケッチと、第２の入力データストリームから生成された第２のセグメントの第２のスケッチとの間の少なくとも類似度を判定することによって１つまたは複数のクライアントアプリケーションから受信された第２の入力データストリームのインラインデータ重複排除のために第１のスケッチを使用するステップと、を含む。

【００１１】

別の態様では、本開示は、データ処理のための方法を提供し、方法は、１つまたは複数のクライアントアプリケーションから１つまたは複数の入力データストリームを受信するステップと、前記１つまたは複数の入力データストリームから少なくとも第１のセグメントおよび第２のセグメントを生成するステップであって、前記第１のセグメントは第１のチャンクのセットを含み、前記第２のセグメントは第２のチャンクのセットを含む、ステップと、（ｉ）前記第１のチャンクのセットの第１のフィンガープリントのセットと、（ｉｉ）前記第２のチャンクのセットの第２のフィンガープリントのセットと、を計算するステップであって、第１のフィンガープリントのセットまたは第２のフィンガープリントのセットは、１つまたは複数のハッシングアルゴリズムを使用して生成された複数のハッシュを含む、ステップと、類似度スコアを生成するために前記第１のフィンガープリントのセットを前記第２のフィンガープリントのセットと比較するステップと、前記類似度スコアが類似度閾値以上である場合、差分演算を実行することによって前記第１のチャンクのセットおよび前記第２のチャンクのセットを処理するステップであって、前記差分演算には、少なくとも（ｉ）前記第１のチャンクのセットに基づいて参照ハッシュセットを生成し、前記第２のチャンクのセットに基づいて第２のハッシュのセットを生成するステップ、（ｉｉ）前記第２のハッシュのセットを前記参照ハッシュセットと比較するステップ、および（ｉｉ）前記参照ハッシュセットにおいて一致を見つけるハッシュを有する前記第２のチャンクのセットのサブセットを参照するシングルポインタを生成して格納するステップが含まれる、ステップと、を含む方法。

【００１２】

別の態様では、本開示は、データ削減のための方法を提供し、方法は、（ａ）１つまたは複数のクライアントアプリケーションから１つまたは複数の入力データストリームを受信するステップと、（ｂ）前記１つまたは複数の入力データストリームから少なくとも第１のセグメントおよび第２のセグメントを生成するステップであって、前記第１のセグメントは第１の複数のチャンクを含み、前記第２のセグメントは第２の複数のチャンクを含む、ステップと、（ｃ）（ｉ）前記第１のセグメントの第１のスケッチおよび（ｉｉ）前記第２のセグメントの第２のスケッチを計算するステップであって、前記第１のスケッチは、前記第１のセグメントを表すかまたは前記第１のセグメントに固有の第１の特徴のセットを含み、前記第２のスケッチは、前記第２のセグメントを表すかまたは前記第２のセグメントに固有の第２の特徴のセットを含み、前記第１の特徴のセットは前記第１のセグメントに対応し、前記第２の特徴のセットは前記第２のセグメントに対応する、ステップと、（ｄ）前記第１のスケッチおよび前記第２のスケッチを処理して、前記第２のセグメントが前記第１のセグメントと類似しているかどうかを示す類似度指標を生成するステップと、（ｅ）前記類似度指標に少なくとも部分的に基づいて、異なるハッシング強度を有する複数のハッシュ関数からハッシュ関数を選択するステップと、（ｆ）（ｅ）に続いて、（１）（ｅ）で選択された前記ハッシュ関数を使用して、前記類似度指標が類似度閾値以上である場合に前記第１のセグメントに対する前記第２のセグメントに対して差分演算を実行するステップ、または（２）前記類似度指標が前記類似度閾値未満である場合に前記差分演算を実行せずに前記第１のセグメントおよび前記第２のセグメントをデータベースに格納するステップと、を含む。

【００１３】

いくつかの実施形態では、特徴のセットは、少なくとも１つのセグメントを別のセグメントから一意に識別または区別するために使用可能な特徴の最小数を含み得る。いくつか

の実施形態では、特徴の最小数は、約３個の特徴から約１５個の特徴の範囲であってもよい。いくつかの実施形態では、特徴の最小数は、１５個以下の特徴を含み得る。

【００１４】

いくつかの実施形態では、少なくとも１つのセグメントは、少なくとも約１メガバイト（ＭＢ）のサイズを有し得る。いくつかの実施形態では、少なくとも１つのセグメントは、約１メガバイト（ＭＢ）～約４ＭＢの範囲のサイズを有し得る。

【００１５】

いくつかの実施形態では、複数のチャンクは、少なくとも約１００個のチャンクを含み得る。いくつかの実施形態では、複数のチャンクは、少なくとも約１０００個のチャンクを含み得る。

【００１６】

いくつかの実施形態では、複数のチャンクは可変長であってもよい。本方法のいくつかの実施形態では、ステップ（ｂ）は、入力データストリームから複数のセグメントを生成するステップであって、複数のセグメントは少なくとも１つのセグメントを含む、ステップをさらに含んでもよい。いくつかの実施形態では、複数のセグメントのうちのセグメントは、約１メガバイト（ＭＢ）～約４ＭＢの範囲の異なるサイズを有する。いくつかの実施形態では、複数のセグメントのうちのセグメントは、約１メガバイト（ＭＢ）～約４ＭＢの範囲内でほぼ同じサイズを有し得る。

【００１７】

本方法のいくつかの実施形態では、ステップ（ｂ）は、複数のチャンクの各チャンクについてフィンガープリントを生成するステップをさらに含んでもよい。いくつかの実施形態では、フィンガープリントは、１つまたは複数のハッシングアルゴリズムを使用して生成され得る。いくつかの実施形態では、フィンガープリントは、１つまたは複数の非ハッシングアルゴリズムを使用して生成され得る。いくつかの実施形態では、特徴のセットは、複数のチャンクから選択されたチャンクのサブセットに関連付けられ得る。

【００１８】

いくつかの実施形態では、特徴のセットは、チャンクのサブセットのためのフィンガープリントのセットを含み得る。いくつかの実施形態では、フィンガープリントのセットは、チャンクのサブセットのための複数のチャンクハッシュを含み得る。いくつかの実施形態では、チャンクのサブセットは、複数のチャンクの約１０％未満であってもよい。いくつかの実施形態では、チャンクのサブセットは、複数のチャンクの約１％未満であってもよい。いくつかの実施形態では、チャンクのサブセットは、約３個のチャンクから約１５個のチャンクを含み得る。

【００１９】

いくつかの実施形態では、チャンクのサブセットは、複数のチャンクについて生成された複数のハッシュに対する１つまたは複数のフィッティングアルゴリズムを使用して、複数のチャンクから選択され得る。いくつかの実施形態では、１つまたは複数のフィッティングアルゴリズムは、２つ以上の異なるハッシュ関数の各ハッシュ関数について最小ハッシュを決定するために使用され得る。いくつかの実施形態では、複数のハッシュは、２つ以上の異なるハッシュ関数を使用して生成され得る。いくつかの実施形態では、２つ以上の異なるハッシュ関数は、セキュアハッシュアルゴリズム０（ＳＨＡ－０）、セキュアハッシュアルゴリズム１（ＳＨＡ－１）、セキュアハッシュアルゴリズム２（ＳＨＡ－２）、およびセキュアハッシュアルゴリズム３（ＳＨＡ－３）からなる群から選択され得る。

【００２０】

いくつかの実施形態では、特徴のセットの各特徴は、２つ以上の異なるハッシュ関数の各ハッシュ関数について最小ハッシュを含み得る。いくつかの実施形態では、特徴のセットは、２つ以上の異なるハッシュ関数の最小ハッシュのベクトルを含み得る。いくつかの実施形態では、特徴のセットは、ベクトルを含む特徴の線形結合として提供され得る。

【００２１】

別の態様では、本開示は、データ処理のための方法を提供し、方法は、（ａ）１つまた

は複数のクライアントアプリケーションから１つまたは複数の入力データストリームを受信するステップと、（ｂ）１つまたは複数の入力データストリームから少なくとも第１のセグメントおよび第２のセグメントを生成するステップであって、第１のセグメントは第１のチャンクのセットを含み、第２のセグメントは第２のチャンクのセットを含む、ステップと、（ｃ）（ｉ）第１の複数のチャンクの第１のフィンガープリントのセットおよび（ｉｉ）第２の複数のチャンクの第２のフィンガープリントのセットを計算するステップと、（ｄ）第１のチャンクのセットおよび第２のチャンクのセットが類似度閾値を満たすと判定するために、第１のフィンガープリントのセットおよび第２のフィンガープリントのセットを処理するステップと、（ｅ）第１のセグメントと第２のセグメントとの間の１つまたは複数の差分を決定するために、第１のチャンクのセットおよび第２のチャンクのセットを処理するステップと、を含む。

【００２２】

いくつかの実施形態では、第１のセグメントおよび第２のセグメントは、少なくとも類似度閾値に基づいて類似していると判定され得る。

【００２３】

いくつかの実施形態では、類似度閾値は、少なくとも約５０％であってもよい。いくつかの実施形態では、類似度閾値は、第１のチャンクのセットと第２のチャンクのセットとの間の重複度を示してもよい。

【００２４】

いくつかの実施形態では、第２のセグメントは、第１のセグメントとほぼ同じサイズであってもよい。いくつかの実施形態では、第２のセグメントは、第１のセグメントとは実質的に異なるサイズであってもよい。

【００２５】

いくつかの実施形態では、第１のセグメントおよび第２のセグメントはそれぞれ、約１メガバイト（ＭＢ）～約４ＭＢの範囲のサイズを有し得る。

【００２６】

いくつかの実施形態では、第１のチャンクのセットおよび第２のチャンクのセットは、異なる数のチャンクを有し得る。

【００２７】

他の実施形態では、第１のチャンクのセットおよび第２のチャンクのセットは、同じ数のチャンクを有し得る。

【００２８】

いくつかの実施形態では、第１のチャンクのセットおよび第２のチャンクのセットはそれぞれ、少なくとも約１００個のチャンクを含んでもよい。いくつかの実施形態では、第１のチャンクのセットおよび第２のチャンクのセットはそれぞれ、少なくとも約１０００個のチャンクを含んでもよい。

【００２９】

いくつかの実施形態では、第１のチャンクのセットおよび第２のチャンクのセットは、可変長であってもよい。

【００３０】

いくつかの実施形態では、第１のフィンガープリントのセットは、第１のチャンクのセットから選択された第１のチャンクのサブセットに関連付けられてもよく、第２のフィンガープリントのセットは、第２のチャンクのセットから選択された第２のチャンクのサブセットに関連付けられてもよい。いくつかの実施形態では、第１のフィンガープリントのセットは、第１のチャンクのサブセットのための第１の複数のチャンクハッシュを含んでもよく、第２のフィンガープリントのセットは、第２のチャンクのサブセットのための第２の複数のチャンクハッシュを含んでもよい。いくつかの実施形態では、第１のチャンクのサブセットは、第１のチャンクのセットの約１０％未満であってもよい。いくつかの他の実施形態では、第１のチャンクのサブセットは、第１のチャンクのセットの約１％未満であってもよい。いくつかのさらなる実施形態では、第２のチャンクのサブセットは、第

２のチャンクのセットの約１０％未満であってもよい。いくつかの実施形態では、第２のチャンクのサブセットは、第２のチャンクのセットの約１％未満であってもよい。

【００３１】

いくつかの実施形態では、第１のチャンクのサブセットおよび第２のチャンクのサブセットは、同じ数のチャンクを有し得る。他の実施形態では、第１のチャンクのサブセットおよび第２のチャンクのサブセットは、異なる数のチャンクを有し得る。いくつかの実施形態では、第１のチャンクのサブセットおよび第２のチャンクのサブセットはそれぞれ、約３個のチャンクから約１５個のチャンクを含み得る。

【００３２】

いくつかの実施形態では、第１および第２のチャンクのサブセットは、第１および第２のチャンクのセットに対して生成された複数のハッシュに対する１つまたは複数のフィッティングアルゴリズムを使用して、第１および第２のチャンクのセットから選択され得る。いくつかの実施形態では、１つまたは複数のフィッティングアルゴリズムは、最小ハッシュ関数を含み得る。

【００３３】

いくつかの実施形態では、第１のフィンガープリントのセットおよび第２のフィンガープリントのセットは、１つまたは複数のハッシングアルゴリズムを使用して生成され得る。いくつかの実施形態では、１つまたは複数のハッシングアルゴリズムは、セキュアハッシュアルゴリズム０（ＳＨＡ－０）、セキュアハッシュアルゴリズム１（ＳＨＡ－１）、セキュアハッシュアルゴリズム２（ＳＨＡ－２）、およびセキュアハッシュアルゴリズム３（ＳＨＡ－３）からなる群から選択され得る。いくつかの実施形態では、第１のフィンガープリントのセットおよび第２のフィンガープリントのセットは、群から選択された２つ以上の異なるハッシングアルゴリズムを使用して生成され得る。

【００３４】

いくつかの他の実施形態では、第１のフィンガープリントのセットおよび第２のフィンガープリントのセットは、１つまたは複数の非ハッシングアルゴリズムを使用して生成され得る。

【００３５】

さらなる態様では、本開示は、データ削減のための方法を提供し、方法は、（ａ）１つまたは複数のクライアントアプリケーションから１つまたは複数の入力データストリームを受信するステップと、（ｂ）１つまたは複数の入力データストリームから少なくとも第１のセグメントおよび第２のセグメントを生成するステップであって、第１のセグメントは第１の複数のチャンクを含み、第２のセグメントは第２の複数のチャンクを含む、ステップと、（ｃ）（ｉ）第１のセグメントの第１のスケッチおよび（ｉｉ）第２のセグメントの第２のスケッチを計算するステップであって、第１のスケッチは、第１のセグメントを表すかまたは第１のセグメントに固有の第１の特徴のセットを含み、第２のスケッチは、第２のセグメントを表すかまたは第２のセグメントに固有の第２の特徴のセットを含み、第１の特徴のセットは第１のセグメントに対応し、第２の特徴のセットは第２のセグメントに対応する、ステップと、（ｄ）第１のスケッチおよび第２のスケッチを処理して、第２のセグメントが第１のセグメントと類似しているかどうかを示す類似度指標を生成するステップと、（ｅ）（ｄ）に続いて、（１）類似度指標が類似度閾値以上である場合に第１のセグメントに対する第２のセグメントに対して差分演算を実行するステップ、または（２）類似度指標が類似度閾値未満である場合に差分演算を実行せずに第１のセグメントおよび第２のセグメントをデータベースに格納するステップと、を含む。

【００３６】

いくつかの実施形態では、（ｅ）の差分演算は、（ｉ）第１のセグメントの第１の複数のチャンクの参照ハッシュセットを生成するステップと、（ｉｉ）参照ハッシュセットをメモリテーブルに格納するステップと、を含み得る。いくつかの実施形態では、参照ハッシュセットは、弱いハッシュを含み得る。いくつかの実施形態では、参照ハッシュセットは、少なくとも１ギガバイト（ＧＢ）規模のスループットを有するハッシング関数を使用

して生成され得る。いくつかの実施形態では、差分演算は、（ｉｉｉ）順次ローリングベースで、第２のセグメントの第２の複数のチャンクのうちのチャンクについてハッシュを生成するステップと、（ｉｖ）ハッシュを参照ハッシュセットと比較して、一致があるかどうかを判定するステップと、をさらに含んでもよい。

【００３７】

いくつかの実施形態では、差分演算は、（ｖ）ハッシュおよび１つまたは複数の他のハッシュが参照ハッシュセットから一致を見つける限り、第２の複数のチャンクのうちの１つまたは複数の後続のチャンクについて１つまたは複数の他のハッシュを生成し続けるステップをさらに含んでもよい。

【００３８】

いくつかの実施形態では、差分演算は、（ｖｉ）後続チャンクのハッシュが参照ハッシュセットから一致を見つけられないことを検出すると、チャンクおよび１つまたは複数の後続のチャンクを参照するシングルポインタを生成して格納するステップをさらに含んでもよい。

【００３９】

いくつかの実施形態では、ハッシュは弱いハッシュであってもよい。

【００４０】

いくつかの実施形態では、１つまたは複数の他のハッシュは、弱いハッシュを含み得る。いくつかの実施形態では、ハッシュおよび１つまたは複数の他のハッシュは、弱いハッシュを含み得る。

【００４１】

いくつかの実施形態では、第２の複数のうちの次のチャンクについて別のハッシュを生成し、別のハッシュを参照ハッシュセットと比較して一致があるかどうかを判定する前に、ハッシュを参照ハッシュセットと比較して一致があるかどうかを判定し得る。

【００４２】

いくつかの実施形態では、１つまたは複数の入力データストリームが１つまたは複数のクライアントアプリケーションから受信されると、差分演算がインラインで実行されてもよい。

【００４３】

いくつかの実施形態では、差分演算は、第１のセグメントおよび第２のセグメントを複数の均質なフラグメントに縮小し得る。いくつかの実施形態では、方法は、１つまたは複数のクラウドオブジェクトデータストアに複数の均質なフラグメントを格納するステップをさらに含んでもよい。いくつかの実施形態では、方法は、複数の均一フラグメントを第１のセグメントおよび第２のセグメントにマッピングするインデックスを生成するステップをさらに含んでもよい。いくつかの実施形態では、方法は、１つまたは複数のクライアントアプリケーションから送信された読み出し要求を受信するステップであって、読み出し要求は、第１のセグメントまたは第２のセグメントのうちの少なくとも１つを含むオブジェクトに対するものであり得る、ステップと、読み出し要求に応答してオブジェクトを生成するために、（１）１つまたは複数のクラウドオブジェクトデータストアに格納された複数の均質なフラグメント、および（２）インデックス、を少なくとも部分的に使用して、第１のセグメントまたは第２のセグメントを再構成するステップと、をさらに含んでもよい。いくつかの実施形態では、方法は、生成されたオブジェクトを、読み出し要求を送信した１つまたは複数のクライアントアプリケーションに提供するステップをさらに含んでもよい。

【００４４】

本方法のいくつかの実施形態では、ステップ（ｄ）の処理は、１つまたは複数の特徴が第１のセットと第２のセットの両方で共通であるかどうかを判定するために、第２の特徴のセットを第１の特徴のセットと比較するステップを含み得る。

【００４５】

いくつかの実施形態では、第２のセグメントは、（ｉ）類似度指標が類似度閾値以上で

ある場合に第１のセグメントと類似していると判定され得る、または（ｉｉ）類似度指標が類似度閾値未満である場合に第１のセグメントと類似していないと判定され得る。いくつかの実施形態では、類似度閾値は、少なくとも約５０％であってもよい。

【００４６】

いくつかの実施形態では、類似度指標は、第１のセグメントと第２のセグメントとの間の重複度を示してもよい。いくつかの実施形態では、１つまたは複数の特徴は、第１のセットおよび第２のセットにおいて類似または同一であってもよい。

【００４７】

いくつかの実施形態では、第２のセグメントは、第１のセグメントとほぼ同じサイズであってもよい。

【００４８】

いくつかの他の実施形態では、第２のセグメントは、第１のセグメントとは実質的に異なるサイズであってもよい。

【００４９】

いくつかの実施形態では、第１のセグメントおよび第２のセグメントはそれぞれ、約１メガバイト（ＭＢ）～約４ＭＢの範囲のサイズを有し得る。

【００５０】

いくつかの実施形態では、第１のセットおよび第２のセットはそれぞれ、約３個の特徴～約１５個の特徴を含み得る。

【００５１】

いくつかの実施形態では、第１の複数のチャンクおよび第２の複数のチャンクはそれぞれ、少なくとも約１００個のチャンクを含み得る。

【００５２】

いくつかの実施形態では、第１の複数のチャンクおよび第２の複数のチャンクは、可変長であってもよい。

【００５３】

本方法のいくつかの実施形態では、ステップ（ｅ）は、類似度指標が類似度閾値未満である場合に、第１のスケッチおよび第２のスケッチをデータベースに格納するステップをさらに含んでもよい。

【００５４】

いくつかの実施形態では、類似度指標は類似度スコアであってもよい。

【００５５】

いくつかの実施形態では、第１のセグメントは、複数のチャンクの長さの合計がターゲットセグメントサイズ範囲内にあると判定することによって生成され、ターゲットセグメントサイズ範囲は１メガバイト（ＭＢ）～１６ＭＢである。

【００５６】

いくつかの実施形態では、特徴のセットは、複数のチャンクから選択された複数のチャンクのサブセットに関連付けられる。

【００５７】

いくつかの実施形態では、特徴のセットは、複数のチャンクのサブセットのためのフィンガープリントのセットを含む。

【００５８】

いくつかの実施形態では、フィンガープリントのセットは、複数のチャンクのサブセットのための複数のチャンクハッシュを含む。

【００５９】

いくつかの実施形態では、複数のチャンクのサブセットは、複数のチャンクの１０％未満を含む。

【００６０】

いくつかの実施形態では、複数のチャンクのサブセットは、３個のチャンクから１５個のチャンクを含む。

【００６１】
いくつかの実施形態では、第２のセグメントは、ターゲットセグメントサイズ範囲に基づいてチャンクのセットを組み立てることによって形成され、チャンクのセットは、第２の入力データストリームから生成される。
【００６２】
いくつかの実施形態では、第２のセグメントのサイズは、第１のセグメントのサイズとは異なる。
【００６３】
いくつかの実施形態では、第２のセグメントのサイズと第１のセグメントのサイズは同じである。
【００６４】
いくつかの実施形態では、第１のセグメントまたは第２のセグメントは、少なくとも１メガバイト（ＭＢ）のサイズを有する。
【００６５】
いくつかの実施形態では、第１のセグメントのサイズは、複数のチャンクの長さの合計に依存する。
【００６６】
いくつかの実施形態では、第１のセグメントは、複数のチャンクの長さの合計がターゲットセグメントサイズ範囲内にあると判定することによって生成される。
【００６７】
いくつかの実施形態では、複数のハッシュは、セキュアハッシュアルゴリズム０（ＳＨＡ－０）、セキュアハッシュアルゴリズム１（ＳＨＡ－１）、セキュアハッシュアルゴリズム２（ＳＨＡ－２）、およびセキュアハッシュアルゴリズム３（ＳＨＡ－３）からなる群から選択される２つ以上の異なるハッシュ関数を使用して生成される。
【００６８】
いくつかの実施形態では、複数のチャンクを組み立てることは、ハッシュ関数を第１の入力データストリームに適用して第１の入力データストリーム内の１つまたは複数のブレークを識別することによって、第１の入力データストリームから複数のチャンクを生成することと、ターゲットセグメントサイズ範囲に基づいて複数のチャンクのサブセットの複数の長さを合計することと、を含む。
【００６９】
いくつかの実施形態では、第２のセグメントは、第２の入力データストリームから生成されたチャンクのセットを組み立てることによって生成される。
【００７０】
いくつかの実施形態では、第１のフィンガープリントのセットは、第１のチャンクのセットから選択された第１のチャンクのサブセットに関連付けられた複数のハッシュに基づいて計算され、第２のフィンガープリントのセットは、第２のチャンクのセットから選択された第２のチャンクのサブセットに関連付けられた複数のハッシュに基づいて計算される。
【００７１】
いくつかの実施形態では、チャンクのサブセットはシーケンシャルチャンクである。
【００７２】
いくつかの実施形態では、チャンクのサブセットへの後続チャンクが、参照ハッシュセット内で一致を見つけられないハッシュを有すると判定すると、シングルポインタが生成され格納される。
【００７３】
いくつかの実施形態では、第２のハッシュセットは弱いハッシュである。
【００７４】
いくつかの実施形態では、類似度指標が第１のスコアを有する場合、第１のハッシュ関数が選択され、類似度指標が第１のスコアよりも低い第２のスコアを有する場合、第２の

ハッシュ関数が選択される。
【００７５】
いくつかの実施形態では、第２のハッシュ関数は、第１のハッシュ関数よりも高いハッシング強度を有する。
【００７６】
いくつかの実施形態では、選択されたハッシュ関数の強度は、類似度指標のスコアと逆に変化する。
【００７７】
本開示の別の態様は、１つまたは複数のコンピュータプロセッサによる実行時に、上記または本明細書の他の場所に記載の方法のいずれかを実施する機械実行可能コードを含む非一時的なコンピュータ可読媒体を提供する。
【００７８】
本開示の別の態様は、１つまたは複数のコンピュータプロセッサおよびそれに結合されたコンピュータメモリを備えるシステムを提供する。コンピュータメモリは、１つまたは複数のコンピュータプロセッサによる実行時に、上記または本明細書の他の場所に記載の方法のいずれかを実施する機械実行可能コードを含む。
【００７９】
本開示の追加の態様および利点は、本開示の例示的な実施形態のみが示され、説明される以下の詳細な説明から、当業者には容易に明らかとされよう。理解されるように、本開示は、他の異なる実施形態が可能であり、そのいくつかの詳細は、すべて本開示から逸脱することなく、さまざまな明白な点で修正が可能である。したがって、図面および説明は、本質的に例示と見なされるべきであり、限定と見なされるべきではない。
参照による援用
【００８０】
本明細書で言及されるすべての刊行物、特許、および特許出願は、個々の刊行物、特許、または特許出願が参照により組み込まれることが具体的かつ個別に示された場合と同じ程度に、参照により本明細書に組み込まれる。参照により組み込まれる刊行物および特許または特許出願が本明細書に含まれる開示と相反する限り、本明細書は、そのような相反する材料に取って代わるおよび／または優先することを意図している。
【００８１】
本発明の新規の特徴は、添付の特許請求の範囲に詳細に記載されている。本発明の特徴および利点のより良い理解は、本発明の原理が利用される例示的な実施形態を説明する以下の詳細な説明、および添付の図面（本明細書では「ｆｉｇｕｒｅ」および「ＦＩＧ．」も参照））を参照することによって得られる。
【図面の簡単な説明】
【００８２】
【図１】本開示のいくつかの実施形態による、データを格納する前にデータを重複排除するためのモジュールを示す図である。
【００８３】
【図２】本開示のいくつかの実施形態による、データストリームをデータブロックおよびチャンクのセットに事前にチャンキングすることを示す図である。
【００８４】
【図３】本開示のいくつかの実施形態による、データストリームをセグメント化するためのフロー図である。
【００８５】
【図４】本開示のいくつかの実施形態による、データストリームを、可変サイズのセグメントおよびチャンクのセットへ事前にチャンキングすることを示す図である。
【００８６】
【図５】本開示のいくつかの実施形態による、可変サイズのセグメントを決定するためのフロー図である。

【００８７】
【図６Ａ】本開示のいくつかの実施形態による、スケッチを計算するためのフロー図である。
【図６Ｂ】本開示のいくつかの実施形態による、スケッチを計算するための要素を示す図である。
【００８８】
【図７Ａ】本開示のいくつかの実施形態による、スケッチを比較するためのフロー図である。
【図７Ｂ】本開示のいくつかの実施形態による、スケッチを比較するための別のフロー図である。
【００８９】
【図８Ａ】本開示のいくつかの実施形態による、差分演算を実行するためのフロー図である。
【図８Ｂ】本開示のいくつかの実施形態による、差分演算を実行するための別のフロー図である。
【図８Ｃ】本開示のいくつかの実施形態による、差分演算を実行するためのさらに別のフロー図である。
【図８Ｄ】本開示のいくつかの実施形態による、差分演算を実行するためのさらに別のフロー図である。
【図８Ｅ】本開示のいくつかの実施形態による、差分演算を実行するためのさらに別のフロー図である。
【００９０】
【図９】本開示のいくつかの実施形態による、データ再構成のためのフロー図である。
【００９１】
【図１０】本開示のいくつかの実施形態による、データ削減システムのためのエコシステムを示す図である。
【００９２】
【図１１】本開示のいくつかの実施形態による、データ削減システムのためのモジュールを示す図である。
【００９３】
【図１２】本明細書で提供される方法を実施するようにプログラムまたは他の方法で構成されたコンピュータシステムを示す図である。
【発明を実施するための形態】
【００９４】
本発明のさまざまな実施形態が本明細書に示され、説明されてきたが、そのような実施形態が例としてのみ提供されることは当業者には明らかであろう。当業者は、本発明から逸脱することなく、数多くの変形、変更、および置換を想到し得る。本明細書に記載の本発明の実施形態のさまざまな代替案が使用され得ることを理解されたい。
【００９５】
「少なくとも」、「より大きい」、または「以上」という用語が一連の２つ以上の数値の最初の数値の前にある場合、「少なくとも」、「より大きい」、または「以上」という用語は常に、その一連の数値の各数値に適用される。例えば、１、２、または３以上は、１以上、２以上、または３以上と同等である。
【００９６】
「わずかに」、「未満」、または「以下」という用語が一連の２つ以上の数値の最初の数値の前にある場合、「わずかに」、「未満」、「以下」という用語は常に、その一連の数値の各数値に適用される。例えば、３、２、または１以下は、３以下、２以下、または１以下と同等である。
【００９７】
本明細書で使用される「リアルタイム」という用語は、一般に、第２のイベントまたは

動作の発生に対する第１のイベントまたは動作の同時または実質的に同時の発生を指す。リアルタイムの動作またはイベントは、少なくとも別のイベントまたは動作に対して１秒未満、１０分の１秒未満、１００分の１秒未満、ミリ秒未満、またはそれ未満の応答時間内に実行され得る。リアルタイム動作は、１つまたは複数のコンピュータプロセッサによって実行され得る。

【００９８】

本明細書で使用される「スケッチ」という用語は、一般に、少なくとも１つのデータセグメントなどの少なくとも１つのデータユニットのフィンガープリントを指す。スケッチは、ファイルまたはオブジェクトの（１または複数の）データセグメントを記述または特徴付けるために使用されてもよい。スケッチは、（１または複数の）データセグメントを記述または特徴付けるために使用することができる特徴のセットを含んでもよい。

【００９９】

本明細書で使用される「弱いハッシングアルゴリズム」などの用語は、一般に、ハッシュ化データチャンクの衝突の総数を減らすことを犠牲にして、単位時間当たりにハッシングされるデータチャンクの数を最大化するハッシングアルゴリズムを指す。衝突は、ハッシングアルゴリズムが異なるデータチャンクに対して同じハッシュ値を生成するときに発生し得る。

【０１００】

本明細書で使用される「強いハッシングアルゴリズム、」などの用語は、一般に、単位時間当たりにハッシングされるハッシュ化データチャンクの数を最大化することを犠牲にして、ハッシュ化データチャンクの衝突の総数を最小化するハッシングアルゴリズムを指す。衝突は、ハッシングアルゴリズムが異なるデータチャンクに対して同じハッシュ値を生成するときに発生し得る。

概要

【０１０１】

データ削減は、データを格納するのに必要な容量を削減するプロセスであり得る。本明細書に記載のデータ削減システムは、他の利点の中でも、ストレージ効率を高め、処理／計算速度性能を向上させ、コストを削減することができる。

【０１０２】

データストレージシステムのデータ削減を処理する従来の方法は、一般に、データをチャンクにセグメント化し、データのチャンクについてフィンガープリント（例えば、ハッシュ）を生成し、フィンガープリントをメモリ内テーブルに格納することに依存している。フィンガープリントが計算された後、メモリ内でルックアップを実行して、チャンクを新しいチャンクと比較することができる。メモリ内でフィンガープリントが一致すると、新しいチャンクが一意であると見なされる場合、新しいチャンクを格納し得る。新しいチャンクが同じであると考えられる場合、最初に格納されたチャンクを指すポインタが格納され得る。このプロセスは、各フィンガープリントを格納するためにかなりの記憶空間を必要とする。さらに、このプロセスは、各チャンクの各フィンガープリントを生成し、次いで各チャンクに対応する各フィンガープリントを異なるチャンクの別のフィンガープリントと比較するために、かなりの処理および計算時間を必要とする。各チャンクのフィンガープリントの計算および処理は、時間がかかり、計算コストが高くなる可能性がある。

【０１０３】

本明細書に記載のデータ削減システムおよび方法は、少なくとも従来のデータ重複排除技術の欠点に対処することができる。例えば、個々のデータチャンクのすべての間でフィンガープリント（ハッシュ）の直接比較を行う代わりに、本明細書で提供されるデータ削減システムおよび方法は、スケッチを使用して、ファイルまたはオブジェクト内のデータの大きなセグメントを記述または特徴付け、スケッチを比較して、２つ以上のセグメントが同種である（例えば、類似である）かどうかを判定することができる。スケッチは、大きなセグメントデータを記述または特徴付けるために使用することができる特徴のセットを含み得る。２つ以上のセグメントのスケッチが実質的に類似していると判定された場合

、その後、より細かいレベル（例えば、特徴レベル、チャンクレベルなど）でスケッチを差分することができ、その時点でフィンガープリント比較を実行することができる。個々のチャンクのポインタの代わりに、フィンガープリントが一致するチャンクの群に対してシングルポインタが生成されてもよい。２つ以上のセグメントのスケッチが実質的に異なると判定された場合、セグメントおよびそれらの特徴のセットがデータベースに格納され得、チャンクレベルでの差分を必要とせず、他の同様のセグメントの重複排除のための計算リソースを節約することができる。

【０１０４】

本明細書のいくつかの実施形態によるデータ削減システム１０４０は、例えば、図１０に示すように、エコシステム内に存在し得る。エコシステムは、１つまたは複数のクライアントアプリケーション１０１０と、１つまたは複数の記憶モジュール１０２０および１０３０とを備えてもよい。

【０１０５】

本明細書に記載のデータ削減システム１０４０は、１つまたは複数のモジュールを備えてもよい。図１１に示すように、モジュールは、スケッチ計算モジュール６００、スケッチ比較モジュール７００、差分演算モジュール８００、データ再構成モジュール９００、データチャンキングモジュール１００、データセグメント化モジュール３００、可変セグメントサイジングモジュール５００、またはそれらのさまざまな組み合わせなどを含み得る。各モジュールの機能は、一般に、以下のように説明され得る。

【０１０６】

スケッチ計算モジュールは、１つまたは複数の入力データストリームから生成された１つまたは複数のデータセグメントについて、１つまたは複数のスケッチを計算するように構成され得る。スケッチがセグメントについて生成された後、スケッチ比較モジュールは、新しいセグメントのスケッチが前のセグメントのスケッチと実質的に類似しているか否かを判定することができる。２つのスケッチが実質的に類似していないと判定された場合、新しいセグメントはデータベースに格納され得る。２つのスケッチが実質的に類似していると判定された場合、差分演算モジュールを利用して、セグメント間でチャンクを比較し、セグメントが両方のセグメントに共通の１つまたは複数の重複チャンクを有するかどうかを判定する。差分モジュールは、スパースインデックス配列を生成し、重複チャンクのブロックのポインタを格納するように構成され得る。重複チャンクのブロックは、同種フラグメントとしてデータベースに格納されてもよい。クライアントアプリケーションから読み出し要求を受信すると、データ再構成モジュールは、スパースインデックス配列および差分モジュールによって生成された均質なフラグメントを使用して、（クライアントアプリケーションから要求された）要求されたオブジェクトを再構成することができる。データチャンキングモジュール、データセグメント化モジュール、または可変セグメントサイジングモジュールを使用したデータの分割に関する追加の態様は、本明細書の他の箇所でさらに詳細に説明される。

Ｉ．スケッチ計算

【０１０７】

一態様では、スケッチ計算のための方法が提供される。スケッチは、予め指定されたクエリのセットおよびデータベースへの更新をサポートするデータ構造であってもよい。スケッチは、セグメント全体のすべての情報を格納する場合と比較して、メモリ空間の消費が少なくなる可能性がある。スケッチは、セグメントのフィンガープリントであってもよい。スケッチ計算は、メモリ要件を低減し、データ書き込みおよび読み出しプロセスの速度を高速化するために使用され得る。スケッチ計算は、少なくとも１つのセグメントから特徴のセットを生成することを含み得る。特徴のセットは、それらの特徴（またはそれらの特徴のサブセット）を使用して類似のセグメントを識別することができるように、セグメント（例えば、フィンガープリントとして）の近似識別子として使用され得る。本明細書の他の箇所に記載されているように、ハッシングアルゴリズム（例えば、ハッシング関数）および／または他のアルゴリズム（例えば、非ハッシングアルゴリズム）を使用して

特徴のセットを決定することによって、スケッチを計算することができる。スケッチは、１つまたは複数のデータセグメントを表してもよい。いくつかの実施形態では、スケッチはメタデータ値であってもよい。スケッチは、１つまたは複数の入力データストリームから他のセグメントに関連付けられた一致または類似のスケッチを見つけるために利用されてもよい。スケッチは、以前に処理されたセグメントに関連する一致または類似のスケッチを見つけるために利用されてもよい。

【０１０８】

スケッチは、スケッチ計算モジュール６００を使用して計算され得る。スケッチ計算モジュールおよびスケッチ計算の一例が、図６Ａおよび図６Ｂを用いて示される。図６Ｂに示すように、入力データストリーム（６１０）がスケッチ計算に使用され得る。本方法は、１つまたはは複数のクライアントアプリケーションから入力データストリームを受信するステップ（ステップ６０１）を含み得る。入力データストリームは、経時的に利用可能にされた一連のデータを含み得る。入力データストリームは、送信されるプロセスにある情報を送信または受信するために使用される一連のデジタル符号化コヒーレント信号（例えば、データのパケット、データパケット、ネットワークパケットなど）であってもよい。入力データストリームは、データ、データパケット、ファイル、オブジェクトなどを含み得る。入力データストリームは、抽出された情報のセットを含み得る。入力データストリームは生データ（例えば、未処理データ、非構造化データなど）を含み得る。入力データストリームは、構造化データを含み得る。入力データストリームは、例えば、ネットワークトラフィック、グラフストリーム、クライアントアプリケーションデータストリーム、またはマルチメディアストリームなどであってもよい。入力データストリームは、少なくとも１つのセグメントを含んでもよい。

【０１０９】

クライアントアプリケーションは、ワークステーションまたはパーソナルコンピュータ上で実行するように構成されたアプリケーションであってもよい。ワークステーションまたはパーソナルコンピュータは、ネットワーク内にあってもよい。クライアントアプリケーションは、エンタープライズアプリケーションを含み得る。いくつかの実施形態では、エンタープライズアプリケーションは、企業環境で動作するように設計された大規模ソフトウェアシステムプラットフォームであってもよい。エンタープライズアプリケーションは、組織内で使用される他のアプリケーションと、または他のアプリケーションなしでインタフェースまたは統合するように設計されてもよい。エンタープライズアプリケーションは、個々のユーザではなく組織のニーズを満たすために使用されるコンピュータソフトウェアであってもよい。そのような組織は、例えば、ビジネス、政府などを含み得る。エンタープライズアプリケーションは、（コンピュータベースの）情報システムの不可欠な部分であり得る。エンタープライズアプリケーションは、例えば、データ管理、ビジネスインテリジェンス、ビジネスプロセス管理、知識管理、顧客関係管理、データベース、エンタープライズリソース計画、エンタープライズ資産管理、低コード開発プラットフォーム、サプライチェーン管理、製品データ管理、製品ライフサイクル管理、ネットワーキングおよび情報セキュリティ、オンラインショッピング、オンライン決済処理、インタラクティブ製品カタログ、自動課金システム、セキュリティ、ビジネスプロセス管理、エンタープライズコンテンツ管理、ＩＴサービス管理、顧客関係管理、エンタープライズリソース計画、ビジネスインテリジェンス、プロジェクト管理、協調、人的リソース管理、製造、労働衛生および安全、エンタープライズアプリケーション統合、情報ストレージまたはエンタープライズフォーム自動化などを支援し得る。エンタープライズアプリケーションの複雑さは、特殊な機能および特定の知識を必要とする場合がある。

【０１１０】

図６Ｂに示すように、本方法は、入力データストリームから少なくとも１つのセグメントを生成するステップ（ステップ６０２）を含み得る。本方法は、本明細書の他の箇所に記載されているように、少なくとも１つのセグメントのスケッチを計算するステップをさらに含んでもよい。図６Ｂに示すように、入力データストリーム６１０内の１つまたは複

数のセグメント６２０～６２２を生成することができる（ステップ６０２）。いくつかの実施形態では、複数のセグメントを入力データストリームから生成することができる。複数のセグメントは、少なくとも約１、５、１０、１５、２５、１００、１０００、１００００またはそれ以上のセグメントを含み得る。複数のセグメントは、少なくとも約１キロバイト（ＫＢ）、１０ＫＢ、１００ＫＢ、５００ＫＢ、１メガバイト（ＭＢ）、２ＭＢ、３ＭＢ、４ＭＢ、５ＭＢ、６ＭＢ、７ＭＢ、８ＭＢ、９ＭＢ、１０ＭＢ、またはそれ以上のサイズを有し得る。複数のセグメントは、最大で約１０ＭＢ、９ＭＢ、８ＭＢ、７ＭＢ、６ＭＢ、４ＭＢ、３ＭＢ、２ＭＢ、１ＭＢ、５００ＫＢ、１００ＫＢ、１０ＫＢ、またはそれ以下のサイズを有し得る。複数のセグメントは、約１００ＫＢ～１０ＭＢ、５００ＫＢ～５ＭＢ、または１ＭＢ～４ＭＢのサイズを有し得る。いくつかの実施形態では、複数のセグメントの各々は、約１ＭＢ～約４ＭＢの範囲のサイズを有し得る。いくつかの実施形態では、複数のセグメントは、約１ＭＢ～約４ＭＢの範囲の異なるサイズを有し得る。

【０１１１】

本明細書の他の箇所で説明するように、入力データストリームからセグメントを生成することができる。各セグメントは、複数のチャンクを含み得る。図６Ｂに示すように、セグメントは、複数のチャンク６３０に変換されてもよい。セグメントは、例えば１０００個のチャンクに変換されてもよい。チャンクはデータを含み得る。チャンクは、情報のフラグメントであってもよい。チャンクは、情報のユニットであってもよい。チャンクは、ヘッダを含み得る。ヘッダは、チャンクのパラメータを示し得る。パラメータは、例えば、チャンクのタイプ、コメント、サイズなどを含み得る。セグメントを取得し、１つまたは複数のチャンクを生成するプロセスは、チャンキングと呼ばれ得る。チャンキングは、データセグメント内のデータを（例えば、入力データストリームからの）連続データのいくつかのセクション（例えば、チャンク）に分割することを含み得る。例えば、中央処理装置（ＣＰＵ）のオーバーヘッドを低減する、またはレイテンシを低減するために、チャンキングを使用することができる。いくつかの実施形態では、セグメントは、少なくとも約５、１０、１５、２５、１００、１０００、１００００またはそれ以上のチャンクを含み得る。セグメントは、約２から１００００個のチャンク、１０から１０００個のチャンク、または２５から１００個のチャンクを含み得る。いくつかの実施形態では、セグメントは、少なくとも約１０００個のチャンクを含み得る。複数のチャンクは、同じ長さであってもよいし、可変長であってもよい。複数のチャンクは、同じデータサイズであってもよい。複数のチャンクは、異なるデータサイズであってもよい。複数のチャンクは、少なくとも約０．１キロバイト（ＫＢ）、０．５ＫＢ、１ＫＢ、２ＫＢ、３ＫＢ、４ＫＢ、５ＫＢ、６ＫＢ、７ＫＢ、８ＫＢ、９ＫＢ、１０ＫＢ、５０ＫＢ以上のサイズを有し得る。複数のチャンクのサイズは、０．１ＫＢ～１０ＫＢ、０．５ＫＢ～７ＫＢ、または１ＫＢ～４ＫＢの範囲であってもよい。いくつかの実施形態では、複数のチャンクは、４ＫＢ～１６ＫＢの範囲のサイズを有し得る。

【０１１２】

本方法は、ステップ６０４を使用して複数のチャンクの各チャンクについてフィンガープリントを生成するステップをさらに含み得、図６Ｂにさらに示される。フィンガープリントは、特定のデータチャンクを識別するために利用され得る。いくつかの実施形態では、フィンガープリントは、１つまたは複数のハッシングアルゴリズムを使用して生成され得る。フィンガープリントは、１つまたは複数のハッシングアルゴリズムによって生成された１つまたは複数のハッシュ値を含み得る。図６Ｂに示すように、複数のハッシュ値６４０（例えば１０００個のハッシュ値）を生成するために、チャンク（例えば、１０００個のチャンク）に対してハッシングアルゴリズムが実行され得る。各ハッシュ値は、各それぞれのチャンクに関連付けられたフィンガープリントであってもよい。いくつかの実施形態では、特定のチャンクに対して２つ以上のハッシュ値が計算され得る。

【０１１３】

本明細書に記載のハッシングアルゴリズム（例えば、ハッシュ（ハッシング）関数）は、任意のサイズのデータを固定サイズ値にマッピングするために使用できる任意の方法を

含み得る。ハッシュ関数によって返される値は、ハッシュ値、ハッシュコード、ダイジェスト、またはハッシュと呼ばれる場合がある。これらの値は、ハッシュテーブルと呼ばれる固定サイズテーブルにインデックスを付けるために使用され得る。場合によっては、暗号化グレードのハッシュ関数が、フィンガープリントを生成するために使用され得る。暗号化グレードのハッシュ関数は、鍵付きであっても、鍵なしであってもよく、またはそれらの組み合わせを使用してもよい。ハッシュ関数は、ＳＨＡ－０、ＳＨＡ－１、ＳＨＡ－２、ＳＨＡ－３、ＳＨＡ－２２４、ＳＨＡ－２５６、ＳＨＡ－３８４、ＳＨＡ－５１２、ＳＨＡ－５１２／２２４、ＳＨＡ－５１２／２５６、ＳＨＡ３－２２４、ＳＨＡ３－２５６、ＳＨＡ３－３８４、ＳＨＡ３－５１２、ＳＨＡＫＥ１２８、ＳＨＡＫＥ２５６、ＢＬＡＫＥ－２５６、ＢＬＡＫＥ－５１２、ＢＬＡＫＥ２ｓ、ＢＬＡＫＥ２ｂ、ＢＬＡＫＥ２Ｘ、ＥＣＯＨ、ＦＳＢ、ＧＯＳＴ、Ｇｒｏｓｔｌ、ＨＡＳ－１６０、ＨＡＶＡＬ、ＪＨ、ＬＳＨ、ＭＤ２、ＭＤ４、ＭＤ５、ＭＤ６、ＲａｄｉｏＧａｔｕｎ、ＲＩＰＥＭＤ、ＲＩＰＥＭＤ－１２８、ＲＩＰＥＭＤ－１６０、ＲＩＰＥＭＤ－３２０、Ｓｋｅｉｎ、Ｓｎｅｆｒｕ、Ｓｐｅｃｔｒａｌ Ｈａｓｈ、Ｓｔｒｅｅｂｏｇ、ＳＷＩＦＦＴ、Ｔｉｇｅｒ、Ｗｈｉｒｌｐｏｏｌ、ＨＭＡＣ、ＫＭＣＡ、Ｏｎｅ－ｋｅｙ ＭＡＣ、ＰＭＡＣ、Ｐｏｌｙ１３０５－ＡＥＳ、ＳｉｐＨａｓｈ、ＵＭＡＣ、ＶＭＡＣ、Ｐｅａｒｓｏｎ ｈａｓｈｉｎｇ、Ｐａｕｌ Ｈｓｉｅｈ’ｓ ＳｕｐｅｒＦａｓｔＨａｓｈ、Ｂｕｚｈａｓｈ、Ｆｏｗｌｅｒ－Ｎｏｌｌ－Ｖｏハッシュ関数、Ｊｅｎｋｉｎｓハッシュ関数、Ｂｅｒｎｓｔｅｉｎ ｈａｓｈ ｄｊｂ２、ＰＪＷ ｈａｓｈ、ＭｕｒｍｕｒＨａｓｈ、Ｆａｓｔ－Ｈａｓｈ、ＳｐｏｏｋｙＨａｓｈ、ＣｉｔｙＨａｓｈ、ＦａｒｍＨａｓｈ、ＭｅｔｒｏＨａｓｈ、ｎｕｍｂｅｒ ｈａｓｈ、ｘｘＨａｓｈ、ｔ１ｈａ、ｃｋｓｕｍ（Ｕｎｉｘ）、ＣＲＣ－１６、ＣＲＣ－３２、Ｒａｂｉｎフィンガープリント、集計ハッシング、一般的な一方向ハッシュ関数、およびＺｏｂｒｉｓｔハッシングで構成される群から選択されてもよい。追加的または代替的に、フィンガープリントはまた、１つまたは複数の非ハッシングアルゴリズムを使用して生成されてもよい。

【０１１４】

本方法は、セグメントの複数の特徴を生成するステップをさらに含んでもよい。セグメントのスケッチは、セグメントを表す、またはセグメントに固有の特徴（例えば、特性）のセットを含み得る。特徴生成または抽出は、セグメントを記述するために必要なリソースの量を削減することができる。特徴は、セグメントからの最も関連性の高い情報を記述し得る。セグメントの特徴は、チャンクに小さな変形が導入されても変化しない場合がある。特徴は、チャンクのセット全体を使用する代わりに縮小表現（例えば、特徴のスケッチ比較）を使用して所望のタスク（例えば、チャンク比較）を実行できるように、セグメントからの関連情報を記述し得る。特徴は、例えば、セグメント内のチャンクに関連付けられた特定の項目を含み得る。項目は、例えば、１つまたは複数のハッシングアルゴリズムによって生成されたハッシュ値を含み得る。項目は、例えば、整数（例えば、ＩＤ番号、ハッシュ値）、データ型、ファイル拡張子などを含み得る。

【０１１５】

セグメントのスケッチにおける特徴のセットは、セグメントを一意に識別または区別するために使用可能な特徴の最小数を含み得る。特徴のセットは、少なくとも約１、２、３、４、５、１０、１５、２５、１００、１００またはそれ以上の特徴を含んでもよい。特徴のセットは、最大で約１００、１００、２５、１５、１０、５、４、３、２、またはそれより少ない特徴を含んでもよい。特徴のセットは、約１から１００、２から２５、３から１５、または５から１０の特徴を含んでもよい。いくつかの実施形態では、セグメントのスケッチにおける特徴のセットは、約３～約１５の特徴の範囲であってもよい。特徴のセットは、特徴の線形結合を含んでもよい。特徴のセットは、セグメントを近似するために使用されてもよい。

【０１１６】

いくつかの実施形態では、特徴のセットは、複数のチャンクから選択されたチャンクのサブセットに関連付けられ得る。特徴のセットは、チャンクのサブセットのためのフィン

ガープリントのセットを含み得る。フィンガープリントのセットは、チャンクのサブセットのためのチャンクハッシュを含み得る。チャンクのサブセットは、複数のチャンクの約１％、５％、１０％、１５％、２０％、２５％、３０％、４０％、５０％未満であってもよい。チャンクのサブセットは、複数のチャンクの約１％～５０％、５％～４０％、または１０％～２５％であってもよい。チャンクのサブセットは、少なくとも約１、２、３、４、５、６、７、８、９、１０、１１、１２、１３、１４、１５、１６、１７、１８、１９、２０、５０、１００、またはそれ以上のチャンクを含み得る。チャンクのサブセットは、複数のチャンクについて生成された複数のハッシュに対する１つまたは複数のフィッティングアルゴリズムを使用して、複数のチャンクから選択され得る。いくつかの実施形態では、複数のハッシュは、２つ以上の異なるハッシュ関数を使用して生成され得る。２つ以上の異なるハッシュ関数は、本明細書の他の箇所で説明されるようなハッシュ関数のいずれかを含み得る。

【０１１７】

いくつかの実施形態では、チャンクのセット全体またはチャンクのサブセットの最小ハッシュを決定するために、１つまたは複数のフィッティングアルゴリズムが使用され得る（ステップ６０５）。特徴は、チャンクのセット全体またはチャンクのサブセットの最小ハッシュ値に対応し得る。図６Ｂに示すように、フィッティングアルゴリズムは、特定のチャンクに対応する１０００個のハッシュ値（例えば、フィンガープリント；６５０）の最小ハッシュ値を取得するために使用され得る。１つまたは複数のフィッティングアルゴリズムは、各チャンクのハッシュ値が生成されるときの最小ハッシュ値を計算するために使用され得る。１つまたは複数のフィッティングアルゴリズムは、チャンクのセット全体のすべてのハッシュ値が生成された後の最小ハッシュ値を計算するために使用され得る。チャンクのセットの最小ハッシュ値は、１０００個のチャンクを表現または記述するために使用することができる特徴（F_0，６５０）であってもよい。いくつかの実施形態では、１つまたは複数のフィッティングアルゴリズムを使用して、２つ以上の異なるハッシュ関数の各関数について最小ハッシュを決定し得る。場合によっては、特徴のセットの各特徴は、２つ以上の異なるハッシュ関数の各ハッシュ関数について最小ハッシュを含み得る。特徴のセットは、２つ以上の異なるハッシュ関数の最小ハッシュのベクトルを含み得る。

【０１１８】

いくつかの実施形態では、チャンクのセット全体またはチャンクのサブセットの最大ハッシュを決定するために、１つまたは複数のフィッティングアルゴリズムが使用され得る。特徴は、チャンクのセット全体またはチャンクのサブセットの最大ハッシュ値に対応し得る。場合によっては、最大ハッシュ値は、追加的または代替的に、特徴のセットを決定するための最小ハッシュ値と共に使用されてもよい。場合によっては、特徴ベクトルは、最小ハッシュ値から生成された１つまたは複数の特徴および／または最大ハッシュ値から生成された１つまたは複数の特徴を含み得る。場合によっては、特徴は、１つまたは複数のチャンクに対する１つまたは複数のハッシングアルゴリズムによって生成された１つまたは複数のハッシュの線形結合であってもよい。

【０１１９】

例えば図６Ｂに示すように、１つまたは複数のハッシングアルゴリズムを使用して、１つまたは複数の特徴（例えば、F_0、F_1、F_2、．．．、F_i）を生成し得る。例えば、１０００個のＳＨＡ－２ハッシュ値を生成するために、ＳＨＡ－２ハッシングアルゴリズムが使用され得る。ＳＨＡ－２ハッシュ値のセットにおける最小ハッシュ値は、第１の特徴（F_0）を生成するために使用され得る。次いで、ＭＤ２ハッシングアルゴリズムを使用して、１０００個のＭＤ２ハッシュ関数値を生成し得る。ＭＤ２ハッシュ値のセットにおける最小ハッシュ値は、第２の特徴（F_1）を生成するために使用され得る。特徴は、同時に生成されてもよい。特徴は、データベースに、または本明細書の他の箇所に記載されているように格納されてもよい。チャンクの個々のフィンガープリント（例えば、すべてのハッシュ値）の代わりに特徴を格納することにより、データストレージおよびメモリ要件の大きさを低減することができる。例えば、１０００個のデータチャンクを含むセグ

メントの場合、１０個の特徴は、セグメント全体（またはチャンクのセット全体）を表すことができる。１０００個の個々のデータチャンクに対して１０００個のハッシュ値を格納する代わりに、本明細書に記載のシステムは、１０個の特徴を格納するだけでよく、したがって、メモリストレージを３桁削減することができる。場合によっては、１つまたは複数の特徴は、チャンクのセット内の１つまたは複数の特定のチャンクに関連付けられてもよい。例えば、特定のチャンク（例えば、チャンク１，１）は、セット内の最小ハッシュ値であるハッシュ値を有し得る。

【０１２０】

図６Ｂに示すように、特徴を組み合わせて特徴ベクトル６７０を生成し得る。特徴を使用して、スケッチ６８０を生成し得る（ステップ６０６）。スケッチ６８０は、特徴のセットを含むことができる。スケッチは、１つまたは複数の特徴ベクトルを含むことができる。スケッチは、本明細書の他の箇所に記載されているように、１つまたは複数の他のスケッチと比較することができる。場合によっては、スケッチは、例えば、時空間スケッチングアルゴリズム、Ｃｏｕｎｔスケッチ、Ｃｏｕｎｔ－ｍｉｎスケッチ、コンサバティブアップデートスケッチ、Ｃｏｕｎｔ－Ｍｉｎ－Ｌｏｇスケッチ、Ｓｌｉｍ－Ｆａｔスケッチ、またはＷｅｉｇｈｔ－Ｍｅｄｉａｎ　Ｓｋｅｔｃｈなどを使用して計算されてもよい。いくつかの実施形態では、スケッチは、類似度ハッシングアルゴリズムまたは同様の機能を使用して生成されてもよい。

【０１２１】

スケッチは、１つまたは複数のクライアントアプリケーションから受信した入力データストリームからの少なくとも１つの他のセグメントのインライン重複排除に使用可能であり得る。インライン重複排除にスケッチを使用することにより、大量のデータ（例えば、ペタバイト規模で）の削減を達成することができる。スケッチは、複数のチャンクのフルインデックスを必要とせずに、インライン重複排除のために使用可能であり得る。スケッチは、少なくとも１つの他の入力データストリーム内のすべてのチャンクをフルインデックスにルックアップする必要なく、インライン重複排除に使用可能であり得る。

ＩＩ．スケッチ比較

【０１２２】

スケッチ比較は、例えば図７Ａに示すように、スケッチ比較モジュール７００を使用して実行され得る。本方法は、少なくとも１つの他の入力データストリームから少なくとも１つの他のセグメントを生成するステップ（ステップ７０１）をさらに含んでもよい。図７Ｂは、第１のセグメント７２０および第２のセグメント７２５を生成するために使用できる第１の入力データストリーム７１０および第２の入力データストリーム７１５を示す。場合によっては、第１のセグメント７２０および第２のセグメント７２５は、同じデータ入力ストリームから生成されてもよい。第２のセグメント７２５は、本明細書の他の箇所に記載された方法を使用して生成され得る。本方法は、第２のセグメントのスケッチを計算するステップ（ステップ７０２）をさらに含んでもよい。図７Ｂは、第１のセグメント７３０のスケッチと第２のセグメント７４５のスケッチとの比較（ステップ７０３）を示している。第２のセグメントのスケッチは、本明細書の他の箇所で説明するように計算され得る。第２のセグメントのスケッチは、第２のセグメントを表すまたは第２のセグメントに固有の特徴の別のセット（例えば、特性）を含んでもよい。図７Ｂは、第１のセグメントのスケッチ７３０が特徴のセット７４０を含んでもよく、また、第２のセグメントのスケッチ７３５が特徴のセット７４５を含んでもよいことを示している。上記の特徴は、本明細書の他の箇所で説明されるように生成され得る。

【０１２３】

本方法は、第２のセグメントが第１のセグメントと確率的に類似しているかどうかを判定するために、類似度スコアに少なくとも部分的に基づいて、第１のスケッチおよび第２のスケッチを処理するステップをさらに含んでもよい（ステップ７０４）。処理するステップは、１つまたは複数の特徴が両方のセットに共通であるかどうかを判定するために、第１の特徴のセットを第２の特徴のセットと比較するステップを含み得る。図７Ｂに示す

ように、第１のセグメントのスケッチ７３０および第２のセグメントのスケッチ７３５は、両方のセット７６４にある特徴および／または両方のセット（７６２，７６６）によって共有され得ない特徴を決定するために比較され得る（７５０）。各スケッチは、異なる数の特徴を含むことができる。例えば、スケッチ７３０は１０個の特徴を含んでもよく、一方、スケッチ７３５は６個の特徴を含んでもよい。各スケッチは、同じ数の特徴を含むことができる。例えば、スケッチ７３０は１０個の特徴を含んでもよく、一方、スケッチ７３５は１０個の特徴を含んでもよい。

【０１２４】

類似度スコアが類似度閾値以上である場合、第２のセグメントは、第１のセグメントと確率的に類似していると判定され得る。類似度スコアが類似度閾値未満である場合、第２のセグメントは、第１のセグメントと確率的に類似していないと判定され得る。図７Ｂに示すように、類似度スコアが閾値以上である場合（７７０）、第１のセグメントと第２のセグメントとは類似していると考えてよい。場合によっては、第１のセグメントと第２のセグメントとが類似していると考えられる場合、本方法は、差分演算を実行するステップをさらに含んでもよい。差分演算は、本明細書の他の箇所に記載されている通りであってもよい。類似度スコアが閾値以下である場合（７７５）、第１のセグメントと第２のセグメントは類似していないと考えてよい。場合によっては、第１のセグメントと第２のセグメントとが類似していないと考えられ得る場合、第２のセグメントの１つまたは複数のチャンクをデータベースに格納してもよい（ステップ７０５）。類似度閾値は、少なくとも約５％、１０％、１５％、２０％、２５％、３０％、３５％、４０％、５０％、５５％、６０％、６５％、７０％、７５％、８０％、８５％、９０％、９５％、９９％、またはそれ以上であってもよい。類似度スコア閾値は、約５％～９９％、１０％～９０％、２０％～８０％、３０％～７０％、または４０％～５０％であってもよい。いくつかの実施形態では、類似度閾値は、少なくとも約５０％であってもよい。

【０１２５】

類似度スコアは、第１のセグメントと第２のセグメントとの間の重複度を示してもよい。場合によっては、例えば、第１のセグメントが１０個の特徴を有し、第２のセグメントが８個の特徴を有し、両方のセットで６個の特徴が一致することが判明した場合、類似度スコアは５０％（例えば、６個の一致する特徴／１２個の固有の特徴）であってもよい。場合によっては、類似度スコアは、特定のセグメントに対して計算されてもよい。例えば、第１のセグメントが１０個の特徴を有し、第２のセグメントが８個の特徴を有し、両方のセットで６個の特徴が一致することが判明した場合、類似度スコアは、それぞれ６／１０（すなわち、６０％）または６／８（すなわち、７５％）であってもよい。場合によっては、第１のセグメントのスケッチの特徴の数と第２のセグメントのスケッチの特徴の数は同じであってもよい。場合によっては、第１のセグメントのスケッチの特徴の数と第２のセグメントのスケッチの特徴の数は異なっていてもよい。

【０１２６】

類似度スコアは、第１のセグメントと第２のセグメントとの間の一致する特徴の数を示してもよい。図７Ｂに示すように、一致する特徴７６４は、両方のスケッチに見られ得る。１つまたは複数の特徴は、両方のセットに類似または同一であってもよい。１つまたは複数の特徴は、両方のセットによって共有されなくてもよく、または両方のセットに共通でなくてもよい。１つまたは複数の特徴は、第１のセグメントと第２のセグメントとの間の一致する特徴と一致しない特徴との組み合わせであってもよい。第２のセグメントは、第１のセグメントと同じサイズであってもよい。第２のセグメントは、第１のセグメントと異なるサイズであってもよい。第１のセグメントおよび第２のセグメントの各々は、約１メガバイト（ＭＢ）～約４ＭＢの範囲のサイズを有し得る。第１のセグメントおよび第２のセグメントの各々は、本明細書の他の箇所で説明されるような範囲のサイズを有し得る。

ＩＩＩ．差分演算

【０１２７】

データ差分演算は、差分演算モジュール８００を使用して実行されてもよく、例えば図８Ａに示されている。本方法は、第１のセグメントおよび第２のセグメントの特徴をデータベースに格納するステップをさらに含んでもよい。図８Ｂに示すように、セグメントのスケッチ（例えば、セグメントの特徴のセット，８１０）と、そのスケッチに対応するチャンクとは、データベース（８２０，８４０）に格納され得る。本方法は、類似度スコアが類似度閾値未満である場合、第２のセグメントおよびその特徴のセット（例えば、スケッチ，８３０）をデータベースに格納するステップをさらに含んでもよい。例えば、スケッチと別のスケッチとの間の類似度スコアが１５％であり、閾値が４０％に設定されている場合、第２のセグメントおよびその特徴のセットをデータベースに格納することができる。図８Ｂに示すように、２つのスケッチが比較され（例えば、８１０対８３０）、類似度スコアが閾値を下回るとき、そのセグメントに対応するスケッチおよびチャンクの特徴は、データベース８４０に格納され得る。場合によっては、データベースは同じであってもよい。場合によっては、データベースは異なっていてもよい。

【０１２８】

本方法は、類似度スコアが類似度閾値以上である場合に、第２のセグメントに対して第１のセグメントとの差分演算を実行するステップをさらに含んでもよい。例えば、あるスケッチと別のスケッチとの間の類似度スコアが６５％であり、閾値が４０％に設定されている場合、第１のセグメントのチャンクと第２のセグメントのチャンクとは差分され得る。図８Ｃに示すように、第１のセグメントおよび第２のセグメントの類似度スコアが閾値以上である場合、両方のセグメントの個々のチャンクを比較することができる（例えば、８５０対８６０）。差分演算は、第１のセグメントの複数のチャンクについて参照ハッシュセット（８７０）を生成するステップを含み得る（ステップ８０１）。第１のセグメントの複数のチャンクのハッシュは、本明細書の他の箇所に記載されているような１つまたは複数のハッシングアルゴリズムを使用して生成され得る（ステップ８０２）。複数のチャンクのハッシュは、予め生成されたハッシュであってもよい。本方法は、参照ハッシュセットをメモリテーブルに格納するステップを含み得る。

【０１２９】

参照ハッシュセットは、弱いハッシュおよび／または強いハッシュを含み得る。ハッシュの強度は、ハッシングアルゴリズムに依存し得る。弱いハッシングアルゴリズムは、１つまたは複数の弱いハッシュを生成し得る。強いハッシングアルゴリズムは、１つまたは複数の強いハッシュを生成し得る。弱ハッシュおよび／または強ハッシュは、本明細書の他の箇所に記載されている１つまたは複数のハッシングアルゴリズムを使用して生成され得る。弱いハッシングアルゴリズムは、弱い衝突耐性を有するハッシングアルゴリズムであってもよい。弱い衝突耐性は、衝突を見つけることができない確率が無視できないことを示し得る。強いハッシングアルゴリズムは、強い衝突耐性を有するハッシングアルゴリズムであってもよい。強い衝突耐性は、衝突を見つけることができない確率が無視できることを示し得る。強いハッシングアルゴリズムは、同じハッシュ値にマッピングする入力を見つけることを困難にする可能性がある。弱いハッシングアルゴリズムは、強いハッシングアルゴリズムよりも、同じハッシュ値にマッピングする入力を見つけることを容易にし得る。弱いハッシングアルゴリズムは、強いハッシュ関数よりもハッシュ値（例えば、同じハッシュ値への鍵のマッピング）をクラスタ化する可能性が高い場合がある。強いハッシュ関数は、ハッシュ値の均一な分布を有し得る。

【０１３０】

ハッシングアルゴリズムの強度（例えば、弱から強）は、勾配スケールであってもよい。ハッシングアルゴリズムの強度は、ハッシングアルゴリズムを使用する時間スケール、ハッシングアルゴリズムの複雑さ、ハッシングアルゴリズムの実装、中央処理装置のベンチマーク、またはバイト当たりのサイクルなどに依存し得る。ハッシングアルゴリズムの強度は、１つまたは複数の統計的テストを使用して決定され得る。１つまたは複数の統計的テストは、例えば、ハッシュ関数がランダム関数と容易に区別できるかどうかを測定し得る。テストは、例えば、ハッシュ関数がアバランシェ効果を表示するかどうかを判定す

ることであってもよい。アバランシェ効果は、入力キーの任意の単一ビット変更が出力の平均で半分のビットに影響を与える効果であり得る。

【０１３１】

弱いハッシングアルゴリズムは、単位時間当たりにハッシングされるデータチャンクの数を最大化し得る。弱いハッシングアルゴリズムは、ハッシュ化データチャンクの衝突の総数を減らすことを犠牲にして、単位時間当たりにハッシングされるデータチャンクの数を最大化し得る。衝突は、ハッシングアルゴリズムが異なるデータチャンクに対して同じハッシュ値を生成する場合であってもよい。強いハッシングアルゴリズムは、ハッシュ化データチャンクの衝突の総数を最小限に抑え得る。強いハッシングアルゴリズムは、単位時間当たりにハッシングされるハッシュ化データチャンクの数を最大化することを犠牲にして、ハッシュ化データチャンクの衝突の総数を最小化し得る。衝突は、ハッシングアルゴリズムが異なるデータチャンクに対して同じハッシュ値を生成する場合であってもよい。

【０１３２】

参照ハッシュセットは、少なくとも１ギガバイト規模のスループットを有する高スループットハッシング関数を使用して生成され得る。場合によっては、高スループットハッシング関数は弱いハッシングアルゴリズムであってもよい。ハッシングアルゴリズムは、本明細書の他の箇所に記載されているようなハッシングアルゴリズムであってもよい。２つのスケッチ／セグメント間の類似度は、高スループットハッシング関数に使用されるハッシングアルゴリズムの強度を決定し得る。例えば、２つのスケッチ／セグメントが特定の類似度スコアを有する場合、別のハッシングアルゴリズムを超える特定のハッシングアルゴリズムが選択され得る。例えば、２つのセグメントの２つのスケッチ間で類似度スコアが高いと考えられる場合、弱いハッシング関数が使用され得る。スケッチ比較が類似度を定量化する際の一次近似であり得るので、弱いハッシング関数が使用され得る（例えば、スケッチは、２つのセグメントが類似していると判定することを支援し、その結果、弱いハッシュが使用され得る）。逆に、２つのセグメント間の特徴のセットの類似度スコアが低い場合、強いハッシング関数が使用され得る。場合によっては、２つのスケッチが低い類似度スコアを有する場合、ハッシング関数を使用しなくてもよい。

【０１３３】

いくつかの実施形態では、類似度スコアが例えば７０％～９０％の範囲である場合、第１のハッシングアルゴリズムが使用されてもよい。場合によっては、類似度スコアが例えば９０％より大きい場合、本方法は、第１のハッシュ関数とは異なる第２のハッシング関数を使用してもよい。類似度スコアが例えば７０％未満であるが５０％より大きい場合、本方法は第３のハッシング関数を使用してもよい。類似度が例えば５０％未満である場合、本方法は、ハッシングアルゴリズムを使用せず、代わりに２つのスケッチ（例えば、特徴）および２つのセグメントをデータベースに格納してもよい。場合によっては、２つのセグメントのスケッチを比較した後に２つのセグメントが実質的に類似していないと考えられる場合、２つのセグメントを区別する利点はわずかであり得る。

【０１３４】

場合によっては、ハッシングスループット能力を最大化するためにハッシング関数を支援するために、さまざまな１つまたは複数のパラメータが変更されてもよい。例えば、パラメータは、ハッシュ値を生成するのに必要なクロックサイクル数を減少させ、ハッシュ値メモリフットプリント、またはデータワードサイズなどを調整するために使用され得る。ハッシュは、反復的に計算され得る。ハッシュは、ハッシングアルゴリズムに与えられるバイトサイズを調整することによって反復的に計算され得る。バイトサイズは、少なくとも約１バイト、２バイト、４バイト、８バイト、１６バイト、３２バイト、６４バイト、またはそれ以上であってもよい。バイトサイズは、最大で約６４バイト、３２バイト、１６バイト、８バイト、４バイト、２バイト、１バイト、またはそれ以下であってもよい。バイトサイズは、約１バイトから６４バイト、１バイトから１６バイト、または１バイトから４バイトであってもよい。

【０１３５】

いくつかの実施形態では、高スループットハッシュ値生成のためのハッシングアルゴリズムの性能は、スループットのデータサイズ（例えば、ギガバイト）に依存し得る。ハッシングアルゴリズムの性能は、ハッシュ値生成のためのデータのスループットの速度（例えば、毎秒ギガバイト）に依存し得る。ハッシングアルゴリズムの性能は、ハッシングアルゴリズムの強度に依存し得る。例えば、弱いハッシング関数が一般に強いハッシング関数よりも速くハッシュ値を生成するので、より迅速なハッシュ値生成が望まれる場合、弱いハッシングアルゴリズムは性能の向上をもたらし得る。

【０１３６】

いくつかの実施形態では、差分演算は、順次ローリングベースで、第２のセグメントの複数のチャンクのうちのチャンクについてハッシュを生成するステップをさらに含んでもよい。例えば、１０００個のチャンクのハッシュ値は、１０００個のハッシュ値が生成されるまで、または１０００個のハッシュ値のサブセットが生成されるまで、第１のチャンクの第１のハッシュ値を生成し、その後、第２のチャンクの第２のハッシュ値を生成することによって生成され得る。図８Ｄに示すように、第２のセグメントにおけるチャンクのハッシュ値（$ＨＣ_{1}$，８６１）は、初期時間（例えば、ｔ１）において計算され得る。第２のセグメント内のチャンクの第２のハッシュ値（$ＨＣ_{2}$，８６２）は、ｔ１（例えば、ｔ２）の後の時間に計算され得る。チャンクは、互いに隣接していてもよい。あるいは、チャンクは連続したチャンクである必要はない（例えば、８６１対８６３）。場合によっては、第２のセグメントの複数のチャンクのうちの次のチャンクについて別のハッシュを生成する前に、ハッシュを参照ハッシュセットと比較して、一致があるかどうかを判定してもよい。場合によっては、第２のセグメントの複数のチャンクのすべてのハッシュを同時に生成してもよい。

【０１３７】

差分演算は、ハッシュを参照ハッシュセットと比較して、一致があるかどうかを判定するステップ（ステップ８０３）をさらに含んでもよい。差分演算は、ハッシュおよび１つまたは複数の他のハッシュが参照ハッシュセットから一致を見つける限り、複数のチャンクのうちの１つまたは複数の後続のチャンクについて１つまたは複数の他のハッシュを生成し続けるステップ（ステップ８０２～８０４）をさらに含んでもよい。差分演算は、後続チャンクのハッシュが参照ハッシュセットから一致を見つけられないことを検出すると、チャンクおよび１つまたは複数の後続のチャンクを参照するシングルポインタを生成して格納するステップ（ステップ８０５）をさらに含んでもよい。ハッシュは、本明細書の他の箇所に記載されているような弱いハッシュであってもよい。

【０１３８】

図８Ｅに示すように、第２のセグメント８６０は、連続した順序であってもよい複数のチャンクを含んでもよい。第１のハッシュ値８７１は、参照ハッシュ値８８０と比較されてもよい。参照ハッシュは、後のセグメントのハッシングの前に任意のセグメントから生成されたハッシュであってもよい。参照ハッシュは、第１のセグメントからのハッシュであってもよい。参照ハッシュは、データベースに格納されたハッシュであってもよい。参照ハッシュ値は、第１のハッシュ値と同等であってもよい。場合によっては、この時点でポインタを生成する代わりに、順次チャンクを検査してもよい。シーケンシャルチャンク（例えば、８７２）が参照ハッシュ値と同じハッシュ値を有する場合、本方法は、不一致が生じるまで（例えば、ハッシュ値は参照ハッシュ値と一致しない，８７４）、各チャンクのハッシュ値（例えば、８７１から８７４）をチェックし続けるステップを含んでもよい。このとき、後続の各チャンク（８７１から８７３）を参照してポインタを格納してもよい。シーケンシャルチャンク分析に続いてポインタを格納すると、格納する必要のあるポインタが少なくなるため、メモリ使用量の減少に加えて、アクセスする必要のあるポインタが少なくなり、計算速度の向上につながる可能性がある。

【０１３９】

１つまたは複数の入力データストリームが１つまたは複数のクライアントアプリケーションから受信されると、差分演算がインラインで実行されてもよい。いくつかの代替実施

形態では、差分演算はオフラインで実行されてもよい。例えば、差分演算は、１つまたは複数のセグメントがデータベースに格納された後にオフラインで実行されてもよい。差分演算は、第１のセグメントおよび第２のセグメントを複数の均質なフラグメントに縮小するために使用されてもよい。複数の均質なフラグメントは、１または複数のクラウドオブジェクトストアに格納されてもよい。差分演算は、縮小されたポインタのセットを含むスパースインデックスを生成するために使用されてもよい。縮小されたポインタのセットは、一連のシーケンシャルチャンクを参照するシングルポインタを含んでもよい。均質なフラグメントを使用すると、差分演算後に格納する必要があるチャンクが少なくなるため、メモリ記憶要件を低減することができる。

ＩＶ．データ再構成

【０１４０】

均質なフラグメントからのデータの再構成は、例えば図９Ａに示すように、データ再構成モジュール９００を使用して実行されてもよい。本方法は、１つまたは複数のクライアントアプリケーションから読み出し要求を受信するステップ（ステップ９１０）をさらに含んでもよい。読み出し要求は、第１のセグメントおよび／または第２のセグメントを含むオブジェクトに対するものであってもよい。本方法は、読み出し要求に応答してオブジェクトを生成するために、複数の均質なフラグメントおよびスパースインデックスを少なくとも部分的に使用して第１および／または第２のセグメントを再構成するステップ（ステップ９２０）をさらに含んでもよい。均質なフラグメントは、本明細書の他の箇所に記載されているように、１つまたは複数のデータチャンクを含んでもよい。

【０１４１】

本方法は、再構成されたオブジェクトを１つまたは複数のクライアントアプリケーションに提供するステップ（ステップ９３０）をさらに含んでもよい。読み出し要求は、スパースアレイインデックスを利用して、オブジェクトを迅速に再構成または再編成してもよい。スパースインデックスアレイは、クライアントアプリケーションによって要求されたオブジェクトを再構成するために各均質なフラグメントを指し示してもよい。データ再構成モジュールは、個々のチャンクのすべてとは対照的に、スパースインデックスおよび均質なフラグメント（例えば、チャンクの集合）を使用してオブジェクトを再構成することができるので、処理時間および計算能力を節約し得る。

Ｖ．データチャンキング

【０１４２】

本明細書に記載の入力データストリームは、可変サイズのセグメントにセグメント化され得る。データストリームのセグメントは、データストリームを、セグメントのうちの１つに組み立てられ得るチャンクのセットに事前にチャンキングすることによって決定され得る。各セグメントは、余分なスペースを浪費することなく重複排除され得る。重複排除チャンクアルゴリズムは、整数のチャンクを含み得るセグメントを生成するために使用され得る。例えば、データストリームのスライディングウィンドウ分析を使用して、４キロバイト（ｋＢ）～１６ｋＢのチャンクをサポートするためにデータストリーム内の自然なブレークを見つけることによってチャンクを識別し得る。この例では、自然なブレークは、１６バイト領域のハッシュを計算し、そのハッシュがパターン内の最後の１３ビットをゼロとして有するパターンを有するかどうかを判定することによって生成され得る。チャンクは、ターゲット範囲（例えば、１メガバイト（ＭＢ）～８ＭＢ、２ＭＢ～１６ＭＢ、または他の何らかの範囲）内のセグメントにさらに組み立てられ得る。

【０１４３】

図１は、データを格納する前にデータを重複排除するために使用され得るデータチャンキングモジュール１００のブロック図である。図１において、データチャンキングモジュール１００は、重複排除されたチャンク（例えば、チャンク１０４Ａ、１０４Ｃ、および１０４Ｄ）を格納するために使用され得るデータストレージ１１０を含み得る。データストレージ１１０は、データ（例えば、ハードディスクドライブ、ソールドステートドライブ、メモリ、光学ドライブ、テープドライブ、および／またはデータを格納することがで

きる別のタイプのシステムを含むストレージシステム；分散ストレージシステム；クラウドストレージシステム；および／または別のタイプのストレージシステム）を重複排除および／または格納することができる任意のタイプのデータストレージシステムであってもよい。データストレージシステムは、物理的または仮想的なデータストレージシステムであってもよい。

【０１４４】

データストリーム１０８を重複排除するために、データチャンキングモジュール１００は、データストリーム１０８を、データブロックのセット１０２Ａ～１０２Ｃに分割し得る。例えば、３つのデータブロック１０２Ａ～１０２Ｂが示されており、代替の実施形態では、より多いまたはより少ないデータブロック１０２Ａ～１０２Ｃが存在することができる。データブロックのサイズを、１ＭＢ～１６ＭＢの範囲（例えば、１ＭＢ～８ＭＢの範囲、２ＭＢ～１６ＭＢの範囲、または他の何らかの範囲）とすることができ、データブロックのサイズは、より大きくてもよいし、より小さくてもよい。場合によっては、データブロックは均等に分割されてもよく、各データブロック１０２Ａ～１０２Ｃは同じ固定サイズを有してもよい。

【０１４５】

重複排除コンポーネント１０６は、各データブロックをより小さいチャンク１０４Ａ～１０４Ｅに分割することによってデータブロック１０２Ａ～１０２Ｃを重複排除し、チャンク１０４Ａ～１０４Ｅの各々がデータストレージ１１０に現在格納され得るか否かを判定し得る。例えば、チャンク１０４Ａ～１０４Ｅの各々について、システム１１０は、そのチャンク１０４Ａ～１０４Ｅのフィンガープリントを計算し得る。この実施形態では、フィンガープリントは、各チャンク１０４Ａ～１０４Ｅを一意に識別するために使用されるメカニズムであってもよい。セキュアハッシングアルゴリズム（ＳＨＡ）（例えば、ＳＨＡ－１、ＳＨＡ－２５６など、および／または別のタイプの暗号化ハッシュ関数）の１つとして、フィンガープリントを暗号化ハッシュ関数とすることができる。チャンク１０４Ａ～１０４Ｅの各々のフィンガープリントは、（フィンガープリント計算においてデータ衝突がないと仮定して）チャンク１０４Ａ～１０４Ｅを一意に識別し得る。フィンガープリントは、チャンク１０４Ａ～１０４Ｅのうちの１つがデータストレージ１１０に現在格納されているかどうかを判定するために使用され得る。システム１１０は、チャンクフィンガープリントをデータベースに格納し得る。格納され得る各チャンク１０４Ａ～１０４Ｅについて、データチャンキングモジュール１００は、チャンク（例えば、チャンク１０４Ａ）についてのフィンガープリントを計算し得、そのフィンガープリントがフィンガープリントデータベースに存在するかどうかを判定し得る。新たに計算されたフィンガープリントがデータベース内にない場合、システム１００は、対応するチャンクを格納し得る。チャンクフィンガープリントがデータベース内のフィンガープリントのうちの１つと一致する場合、このチャンクのコピーは、データストレージ１１０に現在格納され得る。この場合、システム１００は、チャンクを格納しなくてもよい。代わりに、システムは、データストレージ内のこのチャンクへの参照の数のカウントをインクリメントし、そのチャンクへの参照を格納し得る。参照カウントは、そのチャンクをデータストレージ１１０からいつ削除することができるかを判定するために使用され得る。図１に示すように、チャンク１０４Ａ、１０４Ｃ、および１０４Ｄは現在データストレージ１１０に格納されているため、システム１００はデータブロック１０２Ａ用のチャンク１０４Ｂおよび１０４Ｅを格納し得る。データブロック１０４Ａ、１０４Ｃ、および１０４Ｄは、データストレージシステム１１０に既に格納されている可能性があるためである。結果として、システム１００は、それらのチャンクを格納しなくてもよい。場合によっては、データストレージシステムは、データチャンキングモジュール１００の外部に存在してもよい。

【０１４６】

図１で説明したように、データチャンキングモジュール１００は、各データブロックをより小さなチャンクに分割し得、チャンクレベルで重複排除分析を実行し得る。データチャンキングモジュール１００は、データブロックを等しいサイズのチャンクに分割し得る

。しかしながら、これは、データストリーム１０２内の可変サイズのオブジェクトが無作為にランダムチャンクに分割される可能性があるため、重複データの不十分な判定につながる可能性がある。あるいは、データチャンキングモジュール１００は、データストリーム内の異なるオブジェクトを見つけるために、データブロックをより自然なブレークで可変サイズのチャンクに分割し得る。これにより、データストリーム内の重複チャンクを見つける可能性を高めることができる。場合によっては、しかしながら、システムが固定サイズのデータブロックから可変サイズのチャンクを決定する場合、余分なデータチャンクを有する可能性があるため、問題が発生する場合がある。

【０１４７】

図２は、データストリーム２００をデータブロックおよびチャンクのセットに事前にチャンキングするブロック図である。図２では、データストリーム２００は、固定サイズのデータブロック２０２Ａ～２０２Ｃ（例えば、１ＭＢ）に分割され得る。上記のように、データストリーム２００に対して３つより多いまたは少ないデータブロック２０２Ａ～２０２Ｃが存在し得る。データブロック２０２Ａの場合、システムは、データブロック２０２Ａをチャンク２０４Ａ～２０４Ｅに分割し得る。チャンクは可変サイズ（例えば４ｋＢ～１６ｋＢ）であり得るため、チャンク定義に適合しない余分なデータのパーティションが存在する可能性がある。システムは、データブロック２０２０Ａをチャンク２０４Ａ～２０４Ｅにチャンキングし得、システムによって使用されている分割アルゴリズムに適合しない余分なデータのチャンク２０６が存在し得る。例えば、システムは、スライディングウィンドウを使用して、データブロック２０２Ａ～２０２Ｃ内の１６バイトのスライディングウィンドウを調べることによって４～１６ｋＢサイズのチャンク２０４Ａ～２０４Ｅを見つけることができ、パターン内の最後の１３ビットをゼロとして有するパターンを探すことができる。しかしながら、これは、上記のパターン内に適合しない余分なチャンク２０６を残す可能性がある。この結果、別のチャンクが余分なチャンク２０６のフィンガープリントと同じフィンガープリントを有する可能性は低いため、このデータは無駄になる可能性がある。余分なチャンク２０６は、重複を有する可能性が低い別個のチャンクとして格納される。これは、チャンクサイズが小さい場合にはあまり問題にならないかもしれないが、チャンクサイズ（および場合によってはデータブロックサイズ）が大きくなるにつれて、データの浪費の可能性が増大する可能性がある。

【０１４８】

いくつかの実施形態では、これに対する回避策は、余分なチャンク２０６を含むチャンクについて次のデータブロックの先頭を調べることであり得る。例えば、余分なチャンク２０６は、次のデータブロックであるデータブロック２０２Ｂの先頭と併せて分析されてもよい。次のデータブロックを調べることにより、事前チャンキングプロセスを直列化し得、これにより、重複排除プロセス全体の並列化を抑制し得る。

【０１４９】

いくつかの実施形態では、固定サイズのデータブロックを有する代わりに、システムは、データブロックを複数のチャンクにチャンキングするために使用されるものと同じまたは同様の基準を使用して、データストリームを可変サイズのセグメントに事前にチャンキングし得る。システムは、重複排除動作のためにデータをチャンキングするために使用されるのと同じまたは類似の基準を使用して、チャンクについてデータストリームを分析し得る。システムがセグメントの範囲（例えば、１ＭＢ～８ＭＢ、２ＭＢ～１６ＭＢ、または他の何らかの範囲）内にあるデータの量を含むのに十分なチャンクを有し得ると、システムはこのセグメントを複製し得る。この事前チャンキングを実行することにより、システムは、上記の図２に記載されたように、余分なチャンクを有することなくチャンキングされ得るセグメントを生成し得る。これにより、無駄が削減され、並列化が増加する可能性がある。

ＶＩ．データセグメント化

【０１５０】

図３は、データストリームをセグメント化するように構成されたセグメント化モジュー

ル３００のフロー図である。図３において、セグメント化モジュール３００は、ブロック３０２においてデータストリームを受信することによって開始し得る。データストリームは、ファイル、または、重複排除することができる別のタイプのオブジェクトであってもよい。ブロック３０２において、セグメント化モジュール３００は、チャンクのセグメントを作成するためにデータストリームを事前にチャンキングし得る。セグメントは、余分なチャンクを持たない複数のチャンクを含み得る。場合によっては、セグメント化モジュール３００は、データブロックをチャンキングするための重複排除プロセスと同じまたは類似の基準を使用してデータストリームを事前にチャンキングし得る。事前チャンキングは、以下の図４および図５でさらに説明され得る。セグメント化モジュール３００は、ブロック３０６においてセグメントを使用してデータストリームを重複排除し得る。重複排除は、セグメントが重複排除のための余分なチャンクを有しないため、順次または並列に行われ得る。セグメント化モジュール３００は、セグメントの各々をチャンキングし得、これらのチャンクに対して重複排除を実行し得る。例えば、各チャンクについて、セグメント化モジュール３００は、各チャンクについてフィンガープリントを計算し得、このフィンガープリントを使用して、このチャンクがデータストレージに現在格納されているかどうかを判定し得る。セグメント化モジュール３００は、チャンクフィンガープリントをデータベースに格納し得る。格納される各チャンクについて、セグメント化モジュール３００は、チャンクのフィンガープリントを計算し、そのフィンガープリントがフィンガープリントデータベースに存在するかどうかを判定し得る。新たに計算されたフィンガープリントがデータベース内にない場合、セグメント化モジュール３００は、対応するチャンクを格納し得る。チャンクフィンガープリントがデータベース内のフィンガープリントのうちの１つと一致する場合、このチャンクのコピーは、データストレージに格納され得る。ブロック３０８において、このプロセスは、重複排除データストリームを格納し得る。重複排除されたデータストリームは、データストレージに現在格納されていない固有のチャンクを含み得る。セグメント化モジュール３００は、データストリームが書き込まれているときに重複排除されたデータストリームを格納し得るか、または初期格納（例えば、バックグラウンドで重複排除）の後に行うことができる。

【０１５１】

上述したように、セグメント化モジュール３００は、データストリームを、重複排除プロセスの準備ができているセグメントのセットに事前にチャンキングし得る。図４は、データストリーム４００を可変サイズのデータブロックおよびチャンクのセットに事前にチャンキングするブロック図である。図４では、データストリーム４００は、可変サイズのセグメントに事前にチャンキングされ得る。各セグメントは、チャンクの総数であってもよい（例えば、上記の図２に示すように、余分なチャンクがない）。例えば、セグメント４０２Ｂまたは４０２Ｃよりも小さいセグメント４０２Ａは、チャンク４０４Ａ～４０４Ｅを含み得る。場合によっては、チャンク４０４Ａ～４０４Ｅは可変サイズであってもよく、セグメント４０２Ａの一部である余分なチャンクがない。セグメント４０２Ａは、セグメント４０２Ｂまたは４０２Ｃよりも小さいセグメントとして示されてもよく、場合によっては、セグメント４０２Ａは、必ずしも他のデータセグメントよりも小さくなくてもよい（例えば、１つ、いくつか、またはすべてのセグメントより大きくてもよく、別のセグメントと同じサイズであってもよい）。

ＶＩＩ．可変セグメントサイジング

【０１５２】

図５は、重複排除のための可変サイズのセグメントを決定し得る可変セグメントサイジングモジュール５００のフロー図である。図５において、可変セグメントサイジングモジュール５００は、ブロック５０２においてターゲットセグメント情報を受信することによって開始し得る。ターゲットセグメント情報は、可変サイズのセグメントを決定するために使用することができるバイトの範囲を有し得る。例えば、ターゲットセグメント範囲は、１ＭＢ～８ＭＢ、２ＭＧ～１６ＭＢ、または他の何らかの範囲であり得る。ブロック５０４において、ステップ５００は、データストリームを受信し得る。データストリームは

、データストレージに格納することができるファイルまたは別のオブジェクトであってもよい。

【０１５３】

可変セグメントサイジングモジュール５００は、ブロック５０６においてデータストリームの先頭からのオフセットを計算し得る。可変セグメントサイジングモジュール５００は、チャンクを見つけるために使用され得る４ｋＢ～１６ｋＢの範囲内にあり得るオフセットを計算し得る。例えば、可変セグメントサイジングモジュール５００は、データストリームの先頭から４ｋＢ～１６ｋＢの範囲内とすることができるオフセットをランダムに計算し得る。ブロック５０８において、分析のための領域を選択し得る。可変セグメントサイジングモジュール５００は、データストリームに自然なブレークがあるかどうかを判定するために１６バイト領域を選択し得る。ステップ５００は、ブロック５１０において領域ハッシュを計算し得る。可変セグメントサイジングモジュール５００は、ローリングハッシュを使用して領域ハッシュを計算し得る（例えば、Ｒａｂｉｎ－Ｋａｒｐ、Ｒａｂｉｎフィンガープリント、Ｃｙｃｌｉｃフィンガープリント、Ａｄｄｌｅｒローリングハッシュ、および／または何らかの他のタイプのローリングハッシュ）。可変セグメントサイジングモジュール５００は、本明細書の他の箇所に記載されているようなハッシュ関数アルゴリズムを使用し得る。可変セグメントサイジングモジュール５００は、データストリームに自然なブレークがあるかどうかを判定する方法としてこのハッシュを計算し得る。ブロック５１２において、可変セグメントサイジングモジュール５００は、チャンクが見つかったかどうかを判定し得る。可変セグメントサイジングモジュール５００は、１６バイト領域について計算されたハッシュが、ハッシュの最後のビットのうちの少なくとも１３ビットがゼロであると判定することによって、チャンクが存在するか否かを判定し得る。可変セグメントサイジングモジュール５００は、チャンクが見つかったかどうかを判定するために異なる基準を使用し得る（例えば、異なるゼロの数、異なるパターンなど）。チャンクが見つかった場合、実行はブロック５１４に進み得る。チャンクが見つからない場合、実行はブロック５０８に進み得、そこで分析のためにデータストリーム内のウィンドウを進めることによって新しい領域が選択され得る。

【０１５４】

ブロック５１４において、可変セグメントサイジングモジュール５００は、セグメントが見つかったか否かを判定し得る。可変セグメントサイジングモジュール５００は、識別されたセグメントの現在の部分ではないものについて可変セグメントサイジングモジュール５００によって決定され得るようなチャンクの長さを合計することによって、セグメントが見つかったと判定し得る。これらの長さの合計がターゲットセグメントサイズ範囲内にある場合、可変セグメントサイジングモジュール５００は、新しいセグメントが見つかったと判定し得、実行はブロック５１６に進み得る。セグメントが見つからない場合、実行はブロック５０８に進み得、新しいチャンクの分析のためにデータストリーム内のウィンドウを進めることによって新しい領域が選択され得る。ブロック５１８において、可変セグメントサイジングモジュール５００は、重複排除のためにセグメントをマークし得る。可変セグメントサイジングモジュール５００は、このセグメントを重複排除のためにマークし得、セグメントは、後で重複排除され得る。

コンピュータシステム

【０１５５】

本開示は、本開示の方法を実施するようにプログラムされたコンピュータシステムを提供する。図１２は、入力データストリームを取り込み、入力データストリームから１つまたは複数のセグメントを生成し、１つまたは複数のセグメントの１つまたは複数のチャンクのハッシュ値を生成し、１つまたは複数のハッシュ値から特徴を生成し、１つまたは複数のセグメントのスケッチを計算し、１つまたは複数の入力データストリームからの１つまたは複数のセグメントの１つまたは複数のスケッチを比較し、１つまたは複数のセグメントを差分し、１つまたは複数のチャンクをデータベースに格納し、データ重複を削減し、１つまたは複数の読み出し要求からデータを再構成するようにプログラムされた、また

は他の方法で構成されたコンピュータシステム１２０１を示す。コンピュータシステム１２０１は、例えば、１つまたは複数のチャンクのハッシュ値を生成するためのハッシングアルゴリズムが、スケッチ計算およびスケッチ比較のための異なる特徴を取得するように調整され得るなど、本開示のスケッチ計算、スケッチ比較、セグメント差分、およびデータ再構成のさまざまな態様を調整することができる。コンピュータシステム１２０１は、ユーザの電子デバイス、または電子デバイスに対して遠隔に配置されたコンピュータシステムとすることができる。電子デバイスは、モバイル電子デバイスとすることができる。

【０１５６】

コンピュータシステム１２０１は、中央処理装置（ＣＰＵ、本明細書では「プロセッサ」および「コンピュータプロセッサ」でもある）１２０５を含み、これは、シングルコアまたはマルチコアプロセッサ、あるいは並列処理のための複数のプロセッサであり得る。コンピュータシステム１２０１はまた、メモリまたはメモリロケーション１２１０（例えば、ランダムアクセスメモリ、読み取り専用メモリ、フラッシュメモリ）、電子記憶ユニット１２１５（例えば、ハードディスク）、１つまたは複数の他のシステムと通信するための通信インタフェース１２２０（例えば、ネットワークアダプタ）、ならびにキャッシュ、他のメモリ、データストレージおよび／または電子ディスプレイアダプタなどの周辺機器１２２５を含む。メモリ１２１０、記憶ユニット１２１５、インタフェース１２２０、および周辺機器１２２５は、マザーボードなどの通信バス（実線）を介してＣＰＵ１２０５と通信している。記憶ユニット１２１５は、データを格納するためのデータ記憶ユニット（またはデータリポジトリ）であり得る。コンピュータシステム１２０１は、通信インタフェース１２２０の助けを借りて、コンピュータネットワーク（「ネットワーク」）１２３０に動作可能に結合され得る。ネットワーク１２３０は、インターネット、インターネットおよび／またはエクストラネット、あるいはインターネットと通信しているイントラネットおよび／またはエクストラネットであり得る。ネットワーク１２３０は、場合によっては、電気通信および／またはデータネットワークである。ネットワーク１２３０は、クラウドコンピューティングなどの分散コンピューティングを可能にすることができる１つまたは複数のコンピュータサーバを含むことができる。ネットワーク１２３０は、場合によっては、コンピュータシステム１２０１の助けを借りて、ピアツーピアネットワークを実装することができ、これにより、コンピュータシステム１２０１に結合されたデバイスがクライアントまたはサーバとして動作することが可能になる。

【０１５７】

ＣＰＵ１２０５は、プログラムまたはソフトウェアで具体化することができる一連の機械可読命令を実行することができる。命令は、メモリ１２１０などのメモリロケーションに格納され得る。命令は、ＣＰＵ１２０５を対象とすることができ、これは、その後、本開示の方法を実装するようにＣＰＵ１２０５をプログラムまたは他の方法で構成することができる。ＣＰＵ１２０５によって実行される動作の例には、フェッチ、デコード、実行、およびライトバックが含まれ得る。

【０１５８】

ＣＰＵ１２０５は、集積回路などの回路の一部であり得る。システム１２０１の他の１つまたは複数のコンポーネントを回路に含めることができる。場合によっては、回路は特定用途向け集積回路（ＡＳＩＣ）である。

【０１５９】

記憶ユニット１２１５は、ドライバ、ライブラリ、および保存されたプログラムなどのファイルを格納することができる。記憶ユニット１２１５は、ユーザデータ、例えば、ユーザプリファレンスおよびユーザプログラムを格納することができる。コンピュータシステム１２０１は、場合によっては、イントラネットまたはインターネットを介してコンピュータシステム１２０１と通信しているリモートサーバ上に配置されるなど、コンピュータシステム１２０１の外部にある１つまたは複数の追加のデータ記憶ユニットを含むことができる。

【０１６０】

コンピュータシステム１２０１は、ネットワーク１２３０を介して１つまたは複数のリモートコンピュータシステムと通信することができる。例えば、コンピュータシステム１２０１は、ユーザのリモートコンピュータシステムと通信することができる。リモートコンピュータシステムの例には、パーソナルコンピュータ（例えば、ポータブルＰＣ）、スレートまたはタブレットＰＣ（例えば、Ａｐｐｌｅ（登録商標）ｉＰａｄ、Ｓａｍｓｕｎｇ（登録商標）Ｇａｌａｘｙ　Ｔａｂ）、電話、スマートフォン（例えば、Ａｐｐｌｅ（登録商標）ｉＰｈｏｎｅ、Ａｎｄｒｏｉｄ対応デバイス、Ｂｌａｃｋｂｅｒｒｙ（登録商標））、またはパーソナルデジタルアシスタントが含まれる。ユーザは、ネットワーク１２３０を介してコンピュータシステム１２０１にアクセスすることができる。

【０１６１】

本明細書に記載の方法は、例えば、メモリ１２１０または電子記憶ユニット１２１５などのコンピュータシステム１２０１の電子記憶ロケーションに格納された機械（例えば、コンピュータプロセッサ）実行可能コードによって実施され得る。機械実行可能コードまたは機械可読コードは、ソフトウェアの形で提供できる。使用中、コードはプロセッサ１２０５によって実行され得る。場合によっては、コードは、記憶ユニット１２１５から取り出され、プロセッサ１２０５による容易なアクセスのためにメモリ１２１０に格納され得る。状況によっては、電子記憶ユニット１２１５を排除することができ、機械実行可能命令がメモリ１２１０に格納される。

【０１６２】

コードを、事前にコンパイルして、コードを実行するように適合されたプロセッサを有する機械で使用するように構成することも、実行時にコンパイルすることもできる。コードは、選択できるプログラミング言語で提供されて、事前にコンパイルされる方法または実行時にコンパイルされる方法でコードを実行できる。

【０１６３】

コンピュータシステム１２０１など、本明細書で提供されるシステムおよび方法の態様は、プログラミングで具体化され得る。技術のさまざまな態様は、通常は、機械可読媒体のタイプで実行または具体化される機械（またはプロセッサ）実行可能コードおよび／または関連データの形式での「製品」または「製造品」と考えることができる。機械実行可能コードを、メモリ（例えば、読み取り専用メモリ、ランダムアクセスメモリ、フラッシュメモリ）またはハードディスクなどの電子記憶ユニットに格納することができる。「記憶」タイプの媒体は、コンピュータ、プロセッサなどの有形メモリのいずれかもしくはすべて、またはソフトウェアプログラミングのためにいつでも非一時的な記憶を提供し得るさまざまな半導体メモリ、テープドライブ、ディスクドライブなどのそれらの関連モジュールを含むことができる。ソフトウェアのすべてまたは一部は、インターネットまたはその他のさまざまな電気通信ネットワークを通じて通信される場合がある。そのような通信は、例えば、あるコンピュータまたはプロセッサから別のコンピュータへの、例えば、管理サーバまたはホストコンピュータからアプリケーションサーバのコンピュータプラットフォームへのソフトウェアのロードを可能にし得る。したがって、ソフトウェア要素を保持し得る別のタイプの媒体には、ローカルデバイス間の物理インタフェース、有線および光の固定電話ネットワーク、およびさまざまなエアリンクで使用されるような、光、電気、および電磁波が含まれる。有線または無線リンク、光リンクなど、そのような波を運ぶ物理的要素も、ソフトウェアを保持した媒体と見なすことができる。本明細書で使用される場合、非一時的で有形の「記憶」媒体に限定されない限り、コンピュータまたは機械の「可読媒体」などの用語は、実行のためにプロセッサに命令を提供することに関与する任意の媒体を指す。

【０１６４】

したがって、コンピュータ実行可能コードなどの機械可読媒体は、有形記憶媒体、搬送波媒体、または物理的伝送媒体を含むがこれらに限定されない多くの形態をとることができる。不揮発性記憶媒体は、例えば、図面に示されるデータベースなどを実装するために使用され得るような、任意の（１または複数の）コンピュータなどの任意の記憶デバイス

などの、光ディスクまたは磁気ディスクを含む。揮発性記憶媒体は、そのようなコンピュータプラットフォームのメインメモリなどの動的メモリを含む。有形の伝送媒体には、コンピュータシステム内のバスを構成するワイヤを備える同軸ケーブル；銅線および光ファイバが含まれる。搬送波伝送媒体は、電気信号または電磁信号、あるいは無線周波数（ＲＦ）および赤外線（ＩＲ）データ通信中に生成されるような音響波または光波の形態をとることができる。したがって、コンピュータ可読媒体の一般的な形式には、例えば、フロッピーディスク、フレキシブルディスク、ハードディスク、磁気テープ、その他の磁気媒体、ＣＤ－ＲＯＭ、ＤＶＤもしくはＤＶＤ－ＲＯＭ、その他の光学媒体、パンチカード紙テープ、穴のパターンを有するその他の物理的な記憶媒体、ＲＡＭ、ＲＯＭ、ＰＲＯＭとＥＰＲＯＭ、ＦＬＡＳＨ－ＥＰＲＯＭ、その他のメモリチップもしくはカートリッジ、データもしくは命令を伝送する搬送波、そのような搬送波を伝送するケーブルもしくはリンク、またはコンピュータがプログラミングコードやデータを読み取ることができるその他の媒体が含まれる。これらの形式のコンピュータ可読媒体の多くは、実行のためにプロセッサに１つまたは複数の命令の１つまたは複数のシーケンスを運ぶことに関与し得る。

【０１６５】

コンピュータシステム１２０１は、例えば、スケッチングのための複数のチャンクの特徴決定のためのハッシングアルゴリズムを提供するためのユーザインタフェース（ＵＩ）１２４０を備える電子ディスプレイ１２３５を含むか、またはそれと通信することができる。ＵＩの例には、グラフィカルユーザインタフェース（ＧＵＩ）およびウェブベースのユーザインタフェースが含まれるが、これらに限定されない。

【０１６６】

本開示の方法およびシステムを、１つまたは複数のアルゴリズムによって実装することができる。アルゴリズムを、中央処理装置１２０５による実行時にソフトウェアを介して実施することができる。アルゴリズムは、例えば、複数のチャンクのハッシュ値のセットから最小ハッシュ値を生成することができる。

【０１６７】

本発明の好ましい実施形態を本明細書に示し、説明してきたが、そのような実施形態が例としてのみ提供されていることは当業者には明らかであろう。本発明が、明細書内に提供される特定の例によって限定されることは意図されていない。本発明を前述の明細書を参照して説明してきたが、本明細書の実施形態の説明および例示は、限定的な意味で解釈されることを意味するものではない。当業者は、本発明から逸脱することなく、数多くの変形、変更、および置換を想到し得る。さらに、本発明のすべての態様は、さまざまな条件および変数に依存する、本明細書に記載された特定の描写、構成または相対的比率に限定されないことを理解されたい。本明細書に記載された本発明の実施形態に対するさまざまな代替案が、本発明を実施する際に使用されてもよいことを理解されたい。したがって、本発明は、そのような代替、修正、変形、または同等物も網羅することが企図されている。以下の特許請求の範囲が本発明の範囲を定義し、これらの特許請求の範囲内の方法および構造ならびにそれらの均等物がそれによってカバーされることが意図されている。

【図面】

【図１】

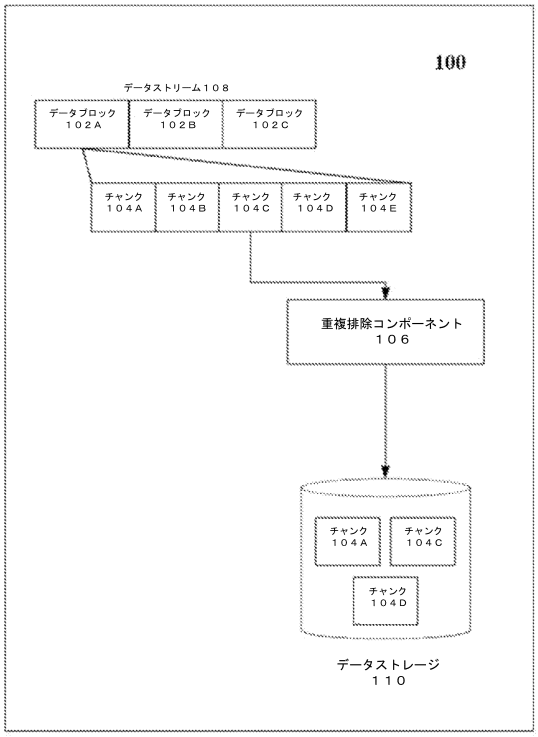

FIG. 1

【図２】

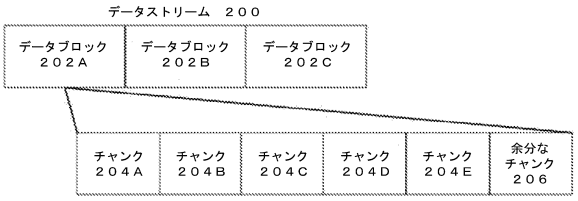

FIG. 2

【図３】

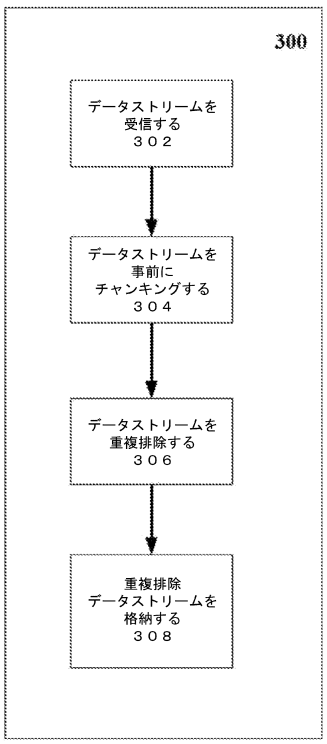

FIG. 3

【図４】

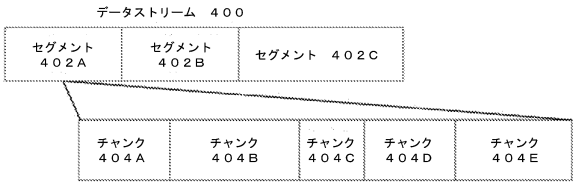

FIG. 4

【図５】

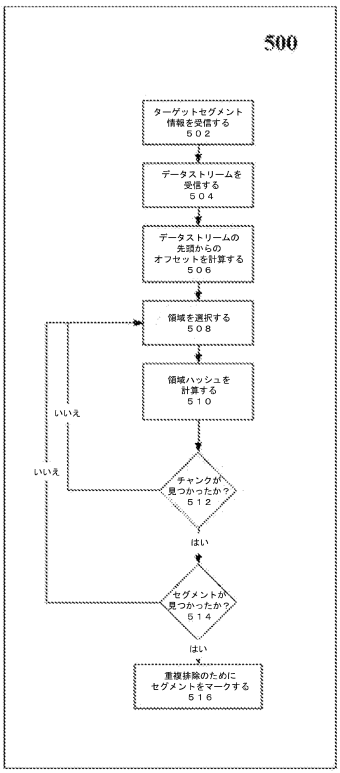

FIG. 5

【図６Ａ】

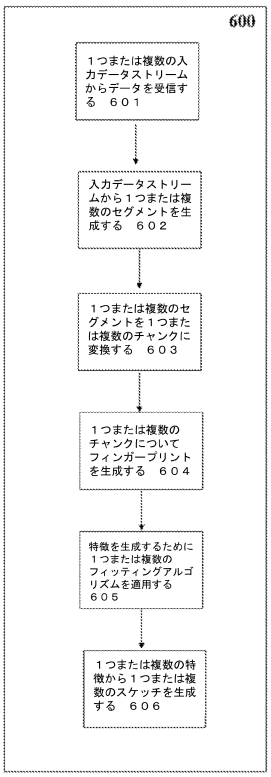

FIG. 6A

【図６Ｂ】

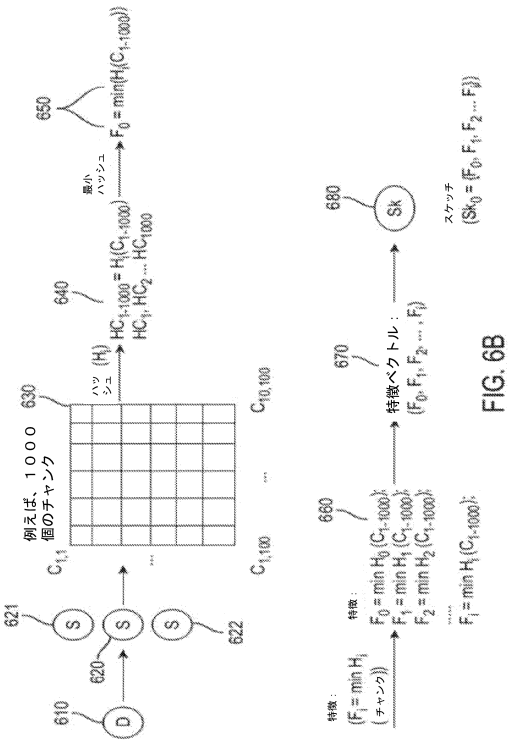

FIG. 6B

【図７Ａ】

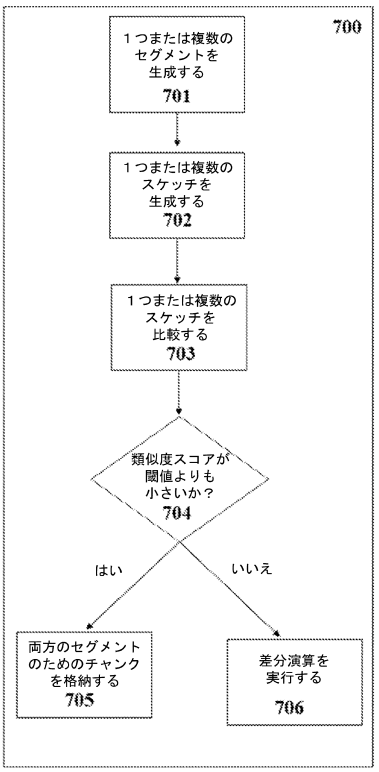

FIG. 7A

【図７Ｂ】

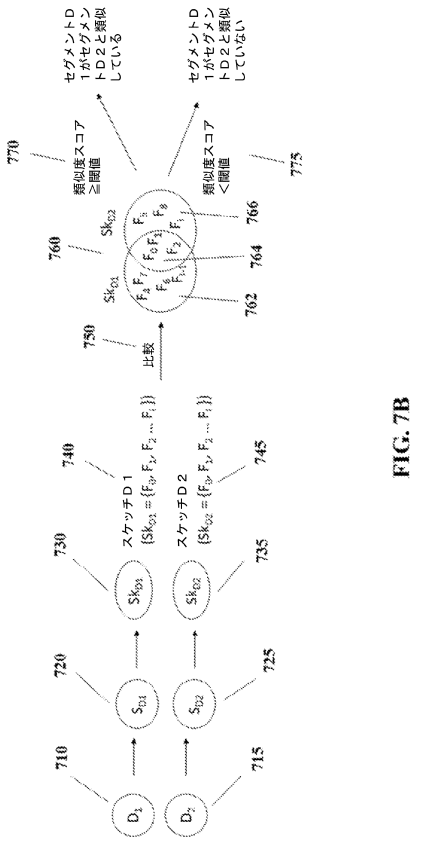

FIG. 7B

【図８Ａ】

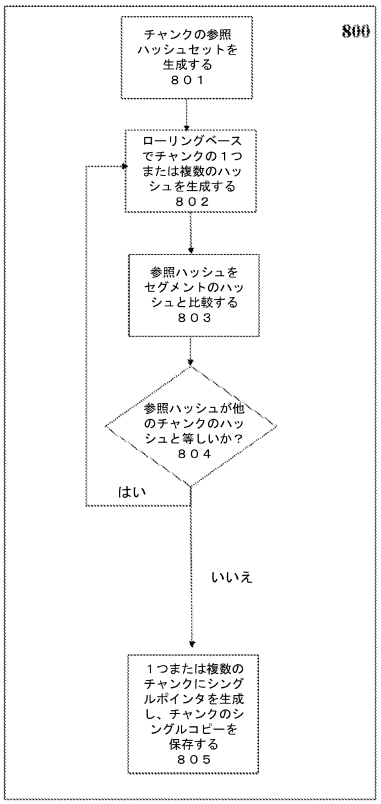

FIG. 8A

【図８Ｂ】

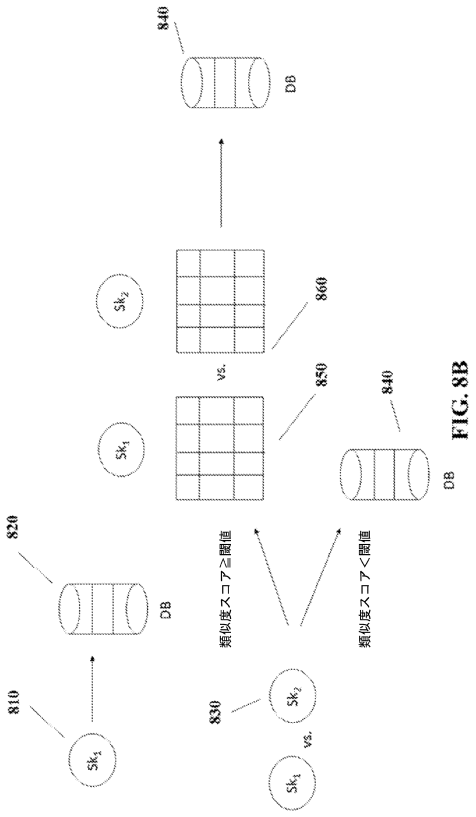

FIG. 8B

【図８Ｃ】

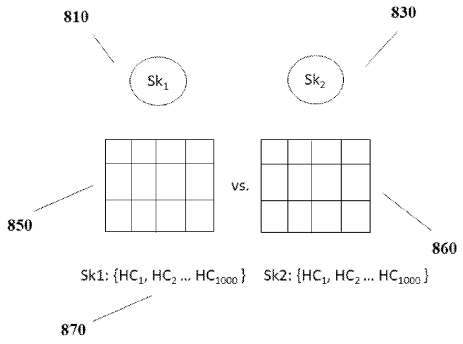

FIG. 8C

【図８Ｄ】

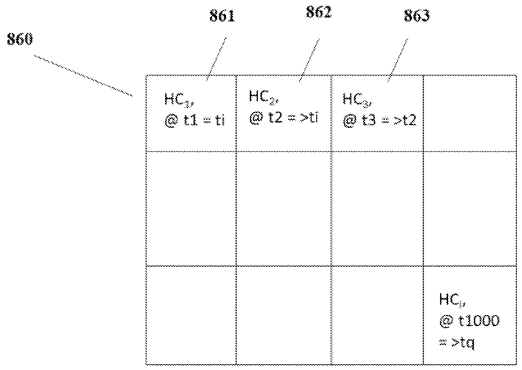

FIG. 8D

【図８Ｅ】

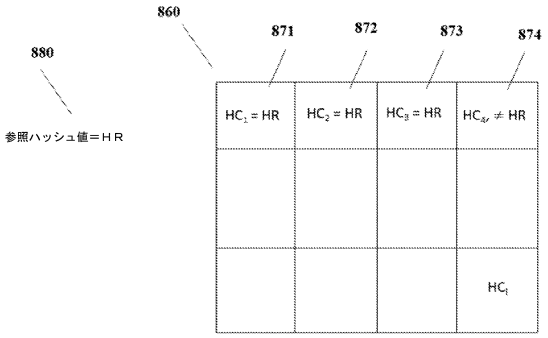

FIG. 8E

【図９】

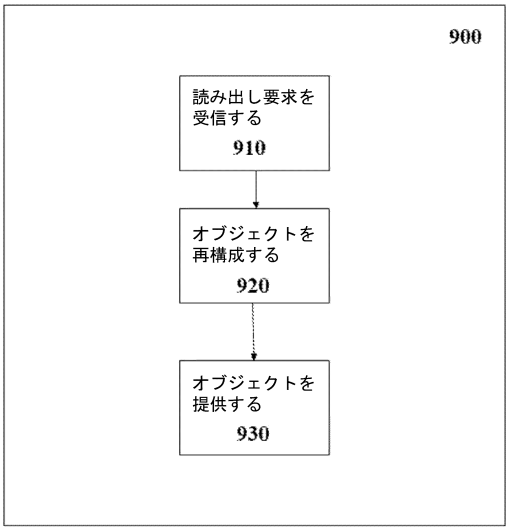

FIG. 9

【図１０】

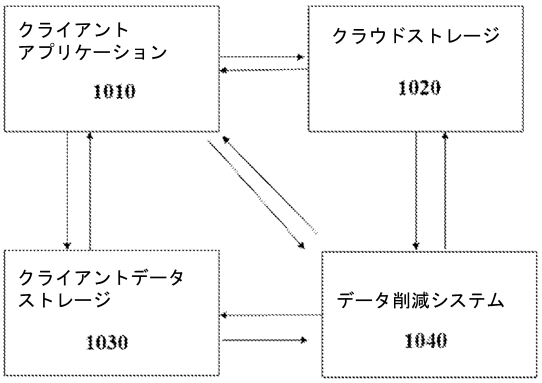

FIG. 10

【図１１】



FIG. 11

【図１２】

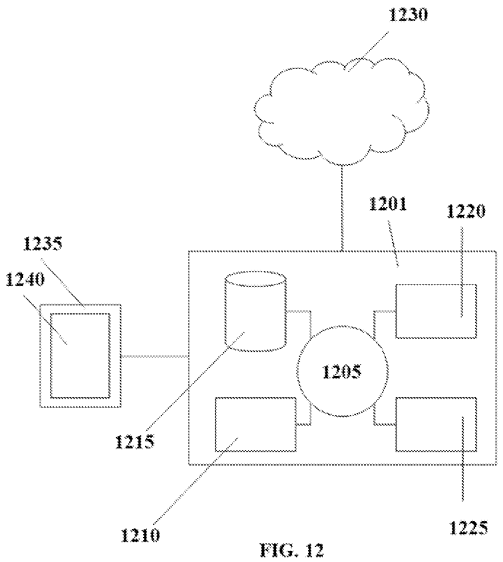

FIG. 12

フロントページの続き

(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号 16/718,714
(32)優先日 令和1年12月18日(2019.12.18)
(33)優先権主張国・地域又は機関
米国(US)
弁理士 安藤 健司
(74)代理人 100143823
弁理士 市川 英彦
(74)代理人 100183519
弁理士 櫻田 芳恵
(74)代理人 100196483
弁理士 川嵜 洋祐
(74)代理人 100160749
弁理士 飯野 陽一
(74)代理人 100160255
弁理士 市川 祐輔
(74)代理人 100182132
弁理士 河野 隆
(74)代理人 100172683
弁理士 綾 聡平
(74)代理人 100219265
弁理士 鈴木 崇大
(74)代理人 100146318
弁理士 岩瀬 吉和
(74)代理人 100127812
弁理士 城山 康文
(72)発明者 ポナラ，サンソッシュ・ラフール
アメリカ合衆国、カリフォルニア・９４０４３、マウンテン・ビュー、モフェット・ブールバード・２７５、エヌデータ・インコーポレイテッド気付
(72)発明者 ヴァイシュ，タラン
アメリカ合衆国、カリフォルニア・９４０４３、マウンテン・ビュー、モフェット・ブールバード・２７５、エヌデータ・インコーポレイテッド気付
審査官 早川 学
(56)参考文献 米国特許出願公開第２０１２／０３２３８６０（ＵＳ，Ａ１）
米国特許出願公開第２０１２／０１６６４４８（ＵＳ，Ａ１）
米国特許出願公開第２０１２／０２３３１３５（ＵＳ，Ａ１）
米国特許出願公開第２０１４／０１１４９３４（ＵＳ，Ａ１）
米国特許出願公開第２０１７／０１２３６７７（ＵＳ，Ａ１）
特開２０１７－１４２６８３（ＪＰ，Ａ）
米国特許出願公開第２０１５／０２９３８１７（ＵＳ，Ａ１）
米国特許出願公開第２０１２／０１４３７１５（ＵＳ，Ａ１）
米国特許出願公開第２０１７／０３０８５５４（ＵＳ，Ａ１）
米国特許出願公開第２０１１／００９９３５１（ＵＳ，Ａ１）
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １６／００－１６／９５８