(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6903773号
(P6903773)
(45)発行日　令和3年7月14日(2021.7.14)　(24)登録日　令和3年6月25日(2021.6.25)

(51)Int.Cl.　　F I
G06F 17/10 (2006.01)　　G06F 17/10 Z
H03M 7/40 (2006.01)　　H03M 7/40

請求項の数 17　(全 30 頁)

(21)出願番号　特願2019-572780 (P2019-572780)
(86)(22)出願日　平成30年6月26日 (2018.6.26)
(65)公表番号　特表2020-527302 (P2020-527302A)
(43)公表日　令和2年9月3日 (2020.9.3)
(86)国際出願番号　PCT/US2018/039620
(87)国際公開番号　WO2019/005888
(87)国際公開日　平成31年1月3日 (2019.1.3)
審査請求日　令和2年3月2日 (2020.3.2)
(31)優先権主張番号　62/525,068
(32)優先日　平成29年6月26日 (2017.6.26)
(33)優先権主張国・地域又は機関
米国 (US)
(31)優先権主張番号　62/658,432
(32)優先日　平成30年4月16日 (2018.4.16)
(33)優先権主張国・地域又は機関
米国 (US)

(73)特許権者　519463075
タリン, スティーヴン
アメリカ合衆国・９４３０１・カリフォルニア州・パロアルト・リットン　アヴェニュ・３７９
(74)代理人　100098394
弁理士　山川　茂樹
(74)代理人　100064621
弁理士　山川　政樹
(72)発明者　タリン, スティーヴン
アメリカ合衆国・９４３０１・カリフォルニア州・パロアルト・リットン　アヴェニュ・３７９

審査官　漆原　孝治

最終頁に続く

(54)【発明の名称】大きなデータをより小さな表現に変換する、およびより小さな表現を当初の大きなデータに戻して再変換するためのシステムおよび方法

(57)【特許請求の範囲】
【請求項１】
標的データファイルを、数学的関数、および前記標的データファイルを表すデータ値からなるセットとして受信側に伝送する方法であって、
（ａ）前記標的データファイルの標的ビットパターンを識別するステップであって、前記標的データファイルは標的データファイルのビットパターンを含み、前記標的ビットパターンは、少なくとも前記標的データファイルのビットパターンの一部分を含む、ステップと、
（ｂ）第１の数学的関数を使用して、現在のビットパターンである第１のビットパターンを生成するように動作するステップと、
（ｃ）数学的関数セットに第１の数学的関数を含ませるステップと、
（ｄ）前記現在のビットパターンと前記標的ビットパターンを比較して、現在の誤差値である第１の誤差値を決定するステップと、
（ｅ）前記現在の誤差値はしきい値条件を満足するかどうかを判断するステップと、
（ｆ）前記現在の誤差値が前記しきい値条件を満足する場合、前記標的データファイルの代わりに、前記数学的関数セットおよび前記現在の誤差値を受信側に伝送するステップであって、前記現在の誤差値と組み合わせた前記数学的関数セットは、前記標的データファイルよりも小さなファイルサイズを有するステップと、
（ｇ）前記現在の誤差値が前記しきい値条件を満足しない場合、
（１）別の数学的関数を使用して、前記現在のビットパターンになる、調節されたビッ

トパターンを生成するように動作し、
（２）前記数学的関数セットに前記別の数学的関数を含ませ、
（３）前記現在のビットパターンと前記標的ビットパターンを比較して、前記現在の誤差値になる、調節された誤差値を生成し、
（４）前記ステップ（ｅ）に戻る
ステップと
を備える方法。
【請求項２】
所望の蓄積速度に基づき、異なる蓄積速度を有する方程式のグループから前記別の数学的関数を選択するステップをさらに備える、請求項１に記載の方法。
【請求項３】
前記現在の誤差値のサイズに基づき、前記別の数学的関数を選択するステップをさらに備える、請求項２に記載の方法。
【請求項４】
前記受信側は、少なくとも１つのオン・チップ・キャッシュを含み、前記現在の誤差値が前記しきい値条件を満たすかどうかを判断する前記ステップは、前記数学的関数のサイズおよび前記現在の誤差値が、前記少なくとも１つのオン・チップ・キャッシュの中に適合するに足るほどに十分に小さいことを確認するステップを含む、請求項１に記載の方法。
【請求項５】
前記現在の誤差値が前記しきい値条件を満たすかどうかを判断する前記ステップは、数学的関数を生成して、前記現在の誤差値を生成するステップを始めるべきかどうか決断するステップを含む、請求項１に記載の方法。
【請求項６】
前記（ｇ）は、
（ｉ）特定の数学的関数を使用して、前記調節された誤差値を生成するように動作するステップと、
（ｉｉ）前記特定の数学的関数は、含ませるためのしきい値要件を満たさないことを判断するステップと、
（ｉｉｉ）前記特定の数学的関数を捨て、前記数学的関数セットに含ませないステップと、
（ｉｖ）前記ステップ（１）に戻るステップと
をさらに備える、請求項１に記載の方法。
【請求項７】
請求項１に記載される方法によって伝送された標的ファイルを再生する方法であって、
受信システムによって、送信側から前記数学的関数および前記データ値からなるセットを受信するステップであって、前記数学的関数および前記データ値からなる前記セットは、前記数学的関数および前記データ値からなる前記セットよりも大きなファイルサイズを有する標的データファイルを代表するものとして生成されたステップと、
少なくとも１つのオン・チップ・キャッシュに、前記数学的関数および前記データ値からなる前記セットの少なくとも一部分を移送するステップと、
前記数学的関数および前記データ値からなる前記セットを使用して、前記受信システムにおいて前記標的ビットパターンを生成するステップと、
前記標的ビットパターンを使用して、前記標的データファイルの少なくとも一部分を生成するステップと、
前記標的データファイルの前記少なくとも一部分を処理するステップと
を備える方法。
【請求項８】
標的データファイルを、数学的関数、および前記標的データファイルを表すデータ値からなるセットとして受信側に伝送するためのシステムであって、

少なくとも１つのハードウェアプロセッサと、
コンピュータ命令を記憶するメモリであって、前記コンピュータ命令は、前記少なくとも１つのハードウェアプロセッサが実行するとき、前記システムに、
（ａ）前記標的データファイルの標的ビットパターンを識別させ、前記標的データファイルは標的データファイルビットパターンを含み、前記標的ビットパターンは、少なくとも前記標的データファイルビットパターンの一部分を含み、
（ｂ）第１の数学的関数を使用して、現在のビットパターンである第１のビットパターンを生成するように動作させ、
（ｃ）数学的関数セットに第１の数学的関数を含ませ、
（ｄ）前記現在のビットパターンと前記標的ビットパターンを比較して、現在の誤差値である第１の誤差値を決定させ、
（ｅ）前記現在の誤差値はしきい値条件を満足するかどうかを判断させ、
（ｆ）前記現在の誤差値が前記しきい値条件を満足する場合、前記標的データファイルの代わりに、前記現在の誤差値と組み合わせて前記標的データファイルよりも小さなファイルサイズを有する前記数学的関数セット、および前記現在の誤差値を受信側に伝送させ、
（ｇ）前記現在の誤差値が前記しきい値条件を満足しない場合、
（１）別の数学的関数を使用して、前記現在のビットパターンになる、調節されたビットパターンを生成するように動作させ、
（２）前記数学的関数セットに前記別の数学的関数を含ませ、
（３）前記現在のビットパターンと前記標的ビットパターンを比較して、前記現在の誤差値になる、調節された誤差値を生成させ、
（４）前記ステップ（ｅ）に戻らせる
ようにに構成されたメモリと
を備えるシステム。
【請求項９】
前記別の数学的関数は、所望の蓄積速度に基づき、異なる蓄積速度を有する方程式のグループから選択される、請求項８に記載のシステム。
【請求項１０】
前記別の数学的関数は、前記現在の誤差値のサイズに基づき選択される、請求項８に記載のシステム。
【請求項１１】
前記受信側は、少なくとも１つのオン・チップ・キャッシュを含み、前記しきい値条件は、前記数学的関数セットのサイズおよび前記現在の誤差値が、前記少なくとも１つのオン・チップ・キャッシュの中に適合するに足るほどに十分に小さいことを確認するステップを含む、請求項８に記載のシステム。
【請求項１２】
前記しきい値条件は、数学的関数を生成して、前記現在の誤差値を生成し始める決断を含む、請求項８に記載のシステム。
【請求項１３】
前記コンピュータ命令は、前記（ｇ）で前記システムに、
（ｉ）特定の数学的関数を使用して、前記調節されたビットパターンを生成するように動作させ、
（ｉｉ）前記特定の数学的関数は、含ませるためのしきい値要件を満たさないことを判断させ、
（ｉｉｉ）前記特定の数学的関数を捨てさせ、前記数学的関数セットに含ませず、
（ｉｖ）前記ステップ（１）に戻らせる
ようにさらに構成される、請求項８に記載のシステム。
【請求項１４】
前記標的データファイルは、ビデオファイルまたはオーディオファイルを含む、請求項

８に記載のシステム。
【請求項１５】
前記受信側は、前記生成された標的データファイルを消費するための消費機器である、請求項８に記載のシステム。
【請求項１６】
前記数学的関数および前記データ値からなる前記セットは、前記標的ビットパターンを無損失に生成するように動作する、請求項８に記載のシステム。
【請求項１７】
請求項８に記載のシステムによって伝送された前記標的ファイルを再生するための受信システムであって、
少なくとも１つのオン・チップ・キャッシュを有する少なくとも１つのハードウェアプロセッサと、
コンピュータ命令を記憶するメモリであって、前記コンピュータ命令は、前記少なくとも１つのハードウェアプロセッサが実行するとき、前記システムに、
前記数学的関数および前記データ値からなるセットよりも大きなファイルサイズを有する標的デーファイルを代表するものとして生成された前記数学的関数および前記データ値からなる前記セットを送信側から受信システムによって受信させ、
前記少なくとも１つのオン・チップ・キャッシュに、前記数学的関数および前記データ値からなる前記セットの少なくとも一部分を移送させ、
前記受信システムによって、前記数学的関数および前記データ値からなる前記セットを使用して、前記標的ビットパターンを生成させ、
前記標的ビットパターンを使用して、前記標的データファイルの少なくとも一部分を生成させ、
前記標的データファイルの前記少なくとも一部分を処理させる
ように構成されたメモリと
を備える、受信システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明の実施形態は、一般にデータ圧縮およびデータ復元の技術に関する。
【背景技術】
【０００２】
大規模データストアは、記憶するのに費用がかかり、通信回線を介して移送するのにかなりの時間がかかる。大規模データストアのサイズをより小さな表現に実質的に低減し、より小さな表現を当初の大規模データストアに戻すことができるシステムおよび方法を有することは役に立つ。
【発明の概要】
【課題を解決するための手段】
【０００３】
いくつかの実施形態では、本発明は、１つまたは複数のコアを伴う少なくとも１つのプロセッサと、方法を遂行するように少なくとも１つのプロセッサを構成するための命令を部分的に含むメモリとを備えるシステムを提供し、方法は、あるデータ長のデータセットを受信するステップと、データセットのビットパターンを決定するステップと、データ長に等しいセット長を有する、すべて０からすべて１までのあらゆる可能な異なるビットパターンを含むビットパターンの基準セットを生成するステップと、試験データに適用される第１のビットパターン生成関数を使用して第１の試験ビットパターンを決定するステップと、第１の試験ビットパターンの場所およびデータセットのビットパターンの場所を使用して、第１の試験ビットパターンとデータセットのビットパターンの間の距離を決定するステップであって、これらの距離は、ビットパターンの基準セットに等しいステップと、データセットのビットパターンの方向に第１の試験ビットパターンを反復して、第１の

試験ビットパターン生成関数を少なくとも１つの第２の試験パターン生成関数と組み合わせて、試験データに適用される第２の試験ビットパターンを決定するステップと、第２の試験ビットパターンの場所およびデータセットのビットパターンの場所を使用して、第２の試験ビットパターンとデータセットのビットパターンの間の距離を決定するステップであって、これらの場所は、ビットパターンの基準セットに等しいステップと、第２の試験ビットパターンとデータセットのビットパターンの間の距離がしきい値距離以内である場合、しきい値距離に基づき誤差関数を生成し、ビットパターン関数および誤差関数を提供して、データセットを再現し、ビットパターン関数は、第１の試験パターン生成関数、ならびに第１の試験パターン生成関数と少なくとも１つの第２の試験パターン生成関数の組合せの反復に等しいステップと、ビットパターン関数および誤差関数に基づきデータセットを回復するステップとを備える。

【０００４】

本技術のさまざまな実施形態のある種の特徴について、添付の特許請求の範囲に独自性と共に示す。本発明の原理を利用する例示的実施形態を示す以下の詳細な記述、および添付図面を参照することにより、本技術の特徴および利点をよりよく理解することができる。

【図面の簡単な説明】

【０００５】

【図１】いくつかの実施形態による、データのより大きな表現とデータのより小さな表現の間でデータを変換するためのシステムを描く構成図である。

【図２】いくつかの実施形態による、本明細書で記述するデータ変換操作の１つまたは複数を実装してもよいコンピュータシステムを例示する構成図である。

【図３（ａ）－（ｃ）】図３（ａ）、３（ｂ）、３（ｃ）は、データコンパイラ（Ｄａｔａ　Ｃｏｍｐｉｌｅｒ、ＤＣ）およびチューリング・デデキント機器（Ｔｕｒｉｎｇ　Ｄｅｄｅｋｉｎｄ　ｄｅｖｉｃｅ、ＴＤ）の可能な実装形態を示す。

【図４】ロード時にＤＣによるのと実行時にＴＤによるのと両方によって遂行される操作の、異なるが関係のあるＮのセットを例示するテンプレートをさらにまた提供する、すべてのＮに関する任意のサイズＮのビットパターンの組合せ複雑性の視覚的表現を両方とも提供するベクトルを示す。

【図５】ＨＣＳＳＦＨＩＬを例示し、ＨＣＳＳＦＨＩＬの記述名が表現するように、出力ビットパターンをＨＣＳＳＦＨＩＬとみなす特性を包含する、出力ビットパターンを作り出す生成関数をそれぞれ表す２つ以上のＬＯＯＰからＨＣＳＳＦＨＩＬが構成されることを示す。これらのループは、加算または任意の他の関数を表す可能性がある１つまたは複数の異なる操作により接続される。

【図６】連続して反復するＨＣＳＳＦＨＩＬがＤＣによりどのように作り出されるかを例示し、各ＨＣＳＳＦＨＩＬは、特定のＪ値を出力し、次いで、特定のＪ値は、次のＨＣＳＳＦＨＩＬを作り出すために改変され、次のＨＣＳＳＦＨＩＬは、次のＪ値を出力する。

【図７】仮想メモリアドレス、タイムスタンプ、および／もしくは他のメタデータに包含されるビットパターンから、ならびに／または仮想メモリアドレス、タイムスタンプ、および／もしくは他のメタデータに包含されるビットパターンの関数から、これらのパラメータの１つまたは複数自体を再計算する方法を介して、空間フットプリントを下げるために、および／またはＴＤが実際に記憶し、利用するパラメータの計算能力を高めるために、場合によってメタデータ項目の誤差項の表現を含むメタデータ項目のパラメータの一部またはすべてを変換するための仮想メモリアドレスおよび／またはタイムスタンプを含むがそれらに限定されないメタデータ項目をＤＣがどのように利用するかを例示する。

【図８】ＴＤが実行時に、似た動作をどのように実装するかを例示する。

【図９】入力ビットパターンｋから始めて、ＤＣがＴＤに引き渡すカーネルの形をとるＤＣの出力を作り出す一連の反復ステップをＤＣがどのように実行するかを段階的に示す、本明細書で記述する反復ステップを表す。

【発明を実施するための形態】

【０００６】
本発明は、集積回路の発明、および最初のマイクロプロセッサの出現までさかのぼり、ムーアの法則の下で４０世代以上の、指数関数的ハードウェア改善が性能に与える潜在的な好ましい影響を劇的に制限する役割を果たしてきた、表面上は手に負えない長年の問題に対する考え方を根本から変える一連の解決手段を提供する。これらの問題は、ネットワーク輻湊の問題、およびマルチ・コア・アーキテクチャのための有用なアプリケーションを見いだすことができないことと共に、ビッグデータの問題を含む。
【０００７】
ビッグデータの問題の中心にあるのは、従来、理解されているように、高速データプロセッサと大量のデータを記憶および移送する費用との間にある、長年にわたり絶えず悪化し、外見上は避けられない性能のミスマッチであり、大規模データストアを保存し、記憶し、取り出し、移送する費用は、ずっと以前から他の費用すべてを圧倒してきており、それにより、継続して指数関数的に増大するデータ処理速度は、データスループットとますます関係がなくなっている。
【０００８】
本発明が提供する、ビッグデータの問題に対する強力で根本的な画期的解決策は、必要なデータ再計算を十分に間に合うように遂行することができる十分迅速なプロセッサが出願する前に、以前は必要とされていたように大規模データストアを保存し、記憶し、再利用し続けるのではなく、はるかに速くファイルセーフな手法で、はるかに小さなカーネルから大きなビットパターンを迅速に再計算することができるようになる手法で、データ処理とデータ記憶の間の、これまで隠されていた基本的等価関係を本発明は利用するので、本発明の方法および装置と共に高速マイクロプロセッサを利用できることが、大規模データストアを保存し、記憶し、取り出し、移送し続ける必要をなくすという事実に由来する。
【０００９】
同様に、本発明が提供するネットワーク輻湊の改善は、再計算を介して同じ情報内容を再生するために、データをあちこちにほとんど動かす必要がないことに由来し、一方では、マルチ・コア・アーキテクチャにより提供される非常に重要な画期的有用性は、マルチ・コア・アーキテクチャをそのコアの数、速度、および能力の関数として利用するとき、複雑なデータ再計算をはるかにより迅速に遂行するマルチ・コア・アーキテクチャの能力に由来する。
【００１０】
本発明の方法と装置の両方が（以下で直接に記述するように）操作するデータの基本単位は、ビットパターンである。各ビットパターンの情報内容は、以下で同様に記述するように、チューリング・デデキント機器（本明細書で以後ＴＤと呼ぶ）に包含される仕組みを介して、高圧縮のカーネルの内容から再計算を介して情報内容の当初の形式でその後１回または複数回、再生するために、以下で記述するように、データコンパイラ（本明細書で以後ＤＣと呼ぶ）に包含される仕組みを介して、高圧縮のカーネルに空間的に低減されることを介して最初に変換される。
【００１１】
本発明の実施形態は、ビットパターンを操作し、ビットパターンの再計算を可能にし、一方では、ビットパターンを記憶する必要をなくす。そのような各ビットパターンは、ビット単位で測定可能な特有の長さを有し、その結果、ビットパターンが包含するビット数により測定したとき、任意のすべての特有の長さｎの任意のすべての特有のビットパターンｋは、そのデータの内容、目的、および／または有用性とは無関係に、本発明の方法および装置が操作するデータの基本単位を表し、それにより、最初に、ＤＣの中に入力された特有のｎビット長のビットパターンｋを特有のフェールセーフな手法で迅速に、無損失に再計算するために、次いでＴＤが実行することができる、はるかにより小さな空間フットプリントを占有する、パラメータおよびコードを含む必要なカーネルをＤＣが生み出すことが可能になり、それにより、前述の所望の結果が可能になる。

【００１２】
簡略化するために、本発明について説明するために、従来は０および１により表される伝統的な、相互に排他的な古典的なオンまたはオフの２進数字から構成されるようなビットパターンについて考えてみるが、本発明の方法および装置は、当業者が達成することができる（２進ではなく、簡単ではない、オンがオフではない）ｑビットおよびｅビットを利用する量子コンピューティングの分野を含むがそれに限定されない複数の可能な代替データ表現分野で本発明をどのようにして実装すべきかがわかる能力を用いてｑビットおよびｅビットを含むがそれに限定されない、他の必ずしもまだ完全に実用的というわけではないが潜在的にはるかにより効率的な基本データ表現単位で記憶されたデータに対して実装することについて、同様に適用可能である。

【００１３】
同様に、本発明の方法および装置は、ハードウェアで、ソフトウェアで、またはファームウェアおよび／もしくはマイクロコードおよび／もしくはそれらの任意の組合せなどの他の機構を介して実装することができる。さらに、ハードウェア、ソフトウェア、ファームウェア、および／またはマイクロコードなどの形をとる以下のメディア／機器／実装形態などのいくつかまたはいずれかの一部またはすべて、さまざまな可能な組合せの（上記の可能性のいずれかまたはすべての任意の可能な実行可能な組合せに基づく、任意のすべての可能なさまざまな混成実装形態を含む）本発明の複数の実施形態はまた、最小から広範囲にわたるまでの、人間による部分的介入、および人間による、またはコンピュータ化されていない他の可能な寄与まで、変化に富む可能な程度に適応させることができ、その結果、人間による、および／もしくはコンピュータ化されていない任意の可能な部分的寄与および／もしくは他の入力、または本明細書で記述するような、本発明の実施形態の本体および内容の少なくとも何らかの部分またはすべての、ハードウェア、ソフトウェア、ファームウェア、マイクロコード、および／もしくは他のコンピュータ化された実装方法および／もしくは装置の任意の可能な、実用的で実行可能な組合せで実装された任意の機器、装置、および／または方法への介入（または介入はまったくなし）は（任意の可能な、または潜在的な、人間による部分的介入を含む、または含まない）、本発明の実用的実施形態を構成する。

【００１４】
本発明の方法および装置を発見するのにかなり大きな実際的妨げとなるのは、真実である場合に本発明を達成できなくする、シャノン（Ｓｈａｎｎｏｎ）限界およびコルモゴロフ（Ｋｏｌｍｏｇｏｒｏｖ）複雑性理論（計数議論（ｃｏｕｎｔｉｎｇ　ａｒｇｕｍｅｎｔ）または鳩の巣原理と一般に呼ばれる簡略化された説明を含む）について長い間受け入れられてきた、以前は疑問の余地がなかったが証明可能に誤った解釈で明白な、現代の情報、データエントロピー、圧縮、および複雑性理論の基本的側面にある。

【００１５】
したがって、依然として頻繁に遭遇する解釈によれば、シャノン限界は、無損失データ圧縮の唯一の利用可能な手段として冗長性削減を達成する、外見上は絶対的で破られることのない、厳密に数学的に証明されたデータ圧縮限界を確立し、それにより、ランダム化されたビットパターンを本質的に圧縮できなくする。だか、本明細書では、この主張を無効にする、任意の多数の反例を特定する。当初、Ｔｈｅ　Ｂｅｌｌ　Ｓｙｓｔｅｍ　Ｔｅｃｈｎｉｃａｌ　Ｊｏｕｒｎａｌ、１９４８年７月および１０月、第２７巻、ｐ．３７９－４２３およびｐ．６２３－６５６で発行され、今では、ｈｔｔｐ：／／ｍａｔｈ．ｈａｖａｒｄ．ｅｄｕ／～ｃｔｍ／ｈｏｍｅ／ｔｅｘｔ／ｏｔｈｅｒｓ／ｓｈａｎｎｏｎ／ｅｎｔｒｏｐｙ／ｅｎｔｒｏｐｙ．ｐｄｆで、オンラインで入手できる、「Ａ　Ｍａｔｈｅｍａｔｉｃａｌ　Ｔｈｅｏｒｙ　ｏｆ　Ｃｏｍｍｕｎｉｃａｔｉｏｎ（通信の数学的理論）」と題する、（上記の誤った主張に合意する信奉者が、当初、現代の情報、データエントロピー、圧縮性、および通信の理論を確立し、その理論にしっかりとした基盤を置く原因になった）シャノンのまったく独創的な１９４８年の論文は、上述の誤った主張を直接に拒絶する反例の１つを具体的に参照することにより拒絶することを含み、ずっと以前か

らシャノンに原因があるとしてきた上述の誤った主張を明確に拒絶している。

【００１６】

シャノンのデータエントロピー理論は、マルコフ連鎖を利用する確率論的分析に完全に基づき、マルコフ理論の確率項は、公知の任意のビットパターンが、高圧縮のカーネルからビットパターンを再計算することを理論的に制限することのないように、当該のデータに関する知識を参照し、シャノンはこのことを、少なくともπの最初の２２兆以上の１０進数字まで数字がほぼ完全にランダム化されていることがその後、例証されている、πを計算する例によって例示している。

【００１７】

したがって、シャノンは、具体的には、自身のデータエントロピー理論の中核にある確率論的分析が、「通信の基本的問題は、他方の地点で選択されたメッセージを一方の地点で正確に、または近似的に再現するという問題である。しばしば、メッセージには意味がある。すなわち、メッセージは、ある種の物理的または概念的実体を伴う何らかのシステムを参照する、またはそれに従って相関させられる。通信のこれらの意味論的側面は、工学的問題に関係がない。重要な側面は、実際のメッセージが、可能なメッセージのセットから選択された１つであるということである。システムは、設計時点では実際に選ばれた１つがわからないので、実際に選ばれた、ただ１つのメッセージではなく、可能な各選択に対して動作するように設計されなければならない。」（ｈｔｔｐ：／／ｍａｔｈ．ｈａｖａｒｄ．ｅｄｕ／～ｃｔｍ／ｈｏｍｅ／ｔｅｘｔ／ｏｔｈｅｒｓ／ｓｈａｎｎｏｎ／ｅｎｔｒｏｐｙ／ｅｎｔｒｏｐｙ．ｐｄｆの１ページ）のように、自身の論文の第２節で、伝送されている特有のデータストリームについて完全にわからないという仮定に依存するという自身の仮定を具体的に明言している。

【００１８】

次いで、シャノンは、続けて、「送信元が１つの特定のメッセージだけを作り出すことができる場合、エントロピーはゼロであり、チャネルはまったく必要ない。たとえば、πの連続する数字を計算するようにセットアップされた計算機は、偶然の要素をまったく伴わない確定したシーケンスを作り出す。この確定したシーケンスを別の地点に『伝送する』ためにチャネルをまったく必要としない。第２の機械を構築して、別の地点で同じシーケンスを計算することができる。しかしながら、これは実用的ではない場合がある。そのような場合、送信元に関して有する統計的知識の一部またはすべてを無視することを選ぶことができる。」（ｈｔｔｐ：／／ｍａｔｈ．ｈａｖａｒｄ．ｅｄｕ／～ｃｔｍ／ｈｏｍｅ／ｔｅｘｔ／ｏｔｈｅｒｓ／ｓｈａｎｎｏｎ／ｅｎｔｒｏｐｙ／ｅｎｔｒｏｐｙ．ｐｄｆの１８ページ）のように、データの特有の内容が伝送時点で前もって既知である場合、そのような理論的制限がまったく適用できないことを証明する反例としてπを具体的に提供している。

【００１９】

さらに、シャノン限界が、無損失データ圧縮の唯一の利用可能な手段として冗長性削減を確立する、データ圧縮性に対して絶対的で破られることのない厳密に数学的に証明された限界を確立し、それにより、ランダム化されたすべてのビットパターンを圧縮不可能にすると主張する、そのような依然として頻繁に繰り返される誤った主張は、Ｐｅｔｅｒ　ＧｒｕｎｗａｌｄおよびＰａｕｌ　Ｖｉｔａｎｙｉ、「Ｓｈａｎｎｏｎ　Ｉｎｆｏｒｍａｔｉｏｎ　ａｎｄ　Ｋｏｌｍｏｇｏｒｏｖ　Ｃｏｍｐｌｅｘｉｔｙ（シャノン情報およびコルモゴロフ複雑性）」、２０１０年７月２２日の論文などの、シャノンの１９４８年の論文の、完全性に関連する前述の引用に十分な注意を払う現代の学術的記事により、少なくとも今ではかなり一般的にきっぱりと否定されており、この論文は、引き続き、たとえ「高いコルモゴロフ複雑性について関係があるメッセージを両方とも選ぶことができる」（２ページ）としても、「２つのメッセージだけを放出するあらゆる送信元について、シャノン情報は、最大で１ビットである」という含意を明示的に明言することにより、上記で提示した２つのシャノン引用のうち最初の方の全体を引用している。

【００２０】

さらに、上記で引用した言明は、それ自体で確かに十分に明快であり、明確であるが、一方では、これらの２人の著者は、「詳細な数学的理論が、１つだけの出版物から本質的に最終形態で芽を出すことはめったに起こらない。そのことは、当然のことながら、Ｃ．Ｅ．シャノンの論文『Ｔｈｅ　ｍａｔｈｅｍａｔｉｃａｌ　ｔｈｅｏｒｙ　ｏｆ　ｃｏｍｍｕｎｉｃａｔｉｏｎ（通信の数学的理論）』（原文のまま）の出現だけで適切に始まったシャノンの情報理論に当てはまった。この論文で、シャノンは、『エントロピー』と呼ぶ、情報の分布の尺度を提案した。分布Ｐのエントロピー Ｈ（Ｐ）は、『Ｐでの固有の不確実性』、または（実際には等価な）『Ｐの出力を観察したとき、どれだけ多くの情報が得られるか』を測定する。このことをより正確に言えば、ＸがＰに従って分布していることがわかっている観察者を想像してみる。次いで、観察者は、Ｘ＝ｘであることを観察する。Ｐのエントロピーは、観察者が出力ｘを観察する前の、出力ｘに関する観察者の不確実性を表す。次に、観察者を、Ｘの値を包含するメッセージを受け取る『受信側』と考える。この２重の観点から、エントロピーは、ランダム変数Ｘの、実現された出力ｘを受け取った後に観察者が得た平均的情報量を表す。」（８ページ）と明言することにより、はるかにより一般的かつ厳密に、ここに含まれる基本的問題を依然としてさらに明らかにし続けており、上記の明言について、２人の著者は、引き続いて、以下の言葉でエントロピーに関するシャノンの数学的定義を提供している。「Ｘを有限集合または可算集合とし、Ｘを分布Ｐ（Ｘ＝ｘ）＝ｐｘを伴いＸ内の値をとるランダム変数とする、次いで、ランダム変数Ｘの（シャノン）エントロピーは、Ｈ（Ｘ）＝Σｘ∈Ｘ　ｐｘ　ｌｏｇ１／ｐｘにより与えられる」（８ページ）、これについて、２人の著者は、「エントロピーは、この場合、ランダム変数から実数への機能上のマッピングとして定義される。多くの教科書では、エントロピーは、本質的にはランダム変数の分布から実数へのマッピングとして等価に定義される。したがって、定義により、Ｈ（Ｐ）：＝Ｈ（Ｘ）＝Σｘ∈Ｘ　ｐｘ　ｌｏｇ１／ｐｘである。」（８ページ）と明言することによりさらに明らかにしている。

【００２１】

しかしながら、コルモゴロフ複雑性について現在依然として受け入れられている理解に関する計数議論／鳩の巣原理のいくつかのバージョンにより表される核となる議論が、少なくとも４つの主要な誤った考えを示すという事実のために、コルモゴロフ複雑性について依然として事実上普遍的に受け入れられているが、高度に根源的で同様に欠点のある理解のこととなると、状況は、より複雑になっている。

【００２２】

したがって、圧縮された形態の任意のビットパターンを、長さＮビットのあらゆるビットパターンの中に２Ｎの可能性を必然的に包含する同じビットパターンの、当初の圧縮されていない形態に戻して一意にマッピングすることができる必要性が、１／２５６の可能性があるそのようなビットパターンだけを１バイトにより圧縮することができ、一方では、１／６５，５３６の可能性があるそのようなビットパターンだけを２バイトにより圧縮することができるなどという、一般にそのような疑問の余地のない自称の、表面上は明白な自明の確実性を伴って強く主張される要件を課すという主張にあるコルモゴロフ複雑性について一般に抱かれている核となる理解は、以下の理由のうち少なくとも４つすべてに関するそのようなきわめて厳格な制約を課すことがまったくできない。

【００２３】

第１に、Ｎビットを包含するビットパターンが、それ自体２Ｎの可能な一意の別個のまったく異なるビットパターンだけを識別するのに役立つことができるという議論は、Ｎビットを包含する２Ｎよりも多くの別個のビットパターンを一意に識別しようと試みるために、さらに別の基準を使用する必要があることを主張しようとするためだけにせいぜい利用される可能性がある。だが、これは、それにもかかわらず、そのようにさらに識別する特性が存在する可能性がどうあってもないということだけに基づき行われる、明確に論理的に是認されていない主張にはるかに及ばない。たとえば、それ自体２進情報内容の１ビットだけを表すことができる単一の一体型２択オン／オフ照明スイッチによりそれぞれ制御される３つの白熱電球を識別しようとする状況について考えてみる。鳩の巣原理／計数

議論の支持者により行われる明確で過度に単純な主張によれば、そのような装置は、２つのまったく異なる白熱電球状態、すなわち、オンおよびオフだけを識別することができる。だが、１つスイッチをオフにしたまま、１つのスイッチをすぐにオンにし、十分長く単に待った後に第３のスイッチをオンにすることにより、オフ、オンかつ冷たい、およびオンかつ熱いという３つのまったく異なる別個の白熱電球状態を識別するだけの十分な情報内容を提供する。同様に、ビットパターンだけに基づく２Ｎの可能な別個の情報内容の状態に加えて、周波数、信号強度などのような特性により通信回線を介して送信されるビットパターンを互いにさらに識別することができることは長い間公知であり、利用されてきており、同様に、本発明の実施形態は、これらのビットパターンを記憶する仮想メモリアドレス、およびこれらのビットパターンに対してある種の操作が遂行された時間を示すタイムスタンプなどのような因子を使用することによって、２つ以上のまったく異なるが同一ビット内容のビットパターン内部に包含される情報内容を識別する手段を提供する。

【００２４】

第２に、操作コードに対して異なるパーセンテージのデータをそれぞれ包含するＫのまったく異なる構成にＮビットを分割することができる条件の下で、そのような可能な結果すべての累積した総計数が２Ｎの別個のビットパターンを超えないことを期待する理由が明確にないという条件の下で、共同でとるこれらの可能な構成すべてが作り出すことができる別個のまったく異なるビットパターンの実際の累計が複製結果の数、および特有の操作が蓄積的かどうかのような因子により決定されるように、そのような各構成は、２Ｎの別個のビットパターンを包含することができる。

【００２５】

第３に、コルモゴロフ複雑性についての従来の理解は、圧縮された各ビットパターンを独立した自己完結型の構成単位として取り扱う。だが、非常に小さなカーネルから理論的に任意の大きな圧縮比を生み出す能力は、計数議論／鳩の巣原理が基づいている指数関数的関係としておおよそ直感的に見ることができる手法で、幾分かそれほど圧縮されていない非常に多数のビットパターンを生み出す能力と引き換えに、非常に圧縮された少数のビットパターンを「手放す」ことができるトレードオフを確立する多数の等価関係を生み出し、逆に走る、および／または反対の立場に転じる。したがって、たとえば、非常に少数のビットのべき乗を連続してとることにより、または大きなインデックス、および短い空間フットプリントを伴う少数のパラメータを用いて他のループを実行することを介して類似の比を達成することを介して、ｎのまったく異なる、たとえば１兆対１の圧縮比を生み出す能力は、任意の２つ以上のそのように非常に高圧縮のビットストリング間のすべての数字上の差、および／またはそれらの間の他の簡単な機能上の組合せが、ほぼ等しく高圧縮のビットストリングのはるかにより大きな新しいセットを作り出す状況ａ）、上記のａ）で記述したようなこの、または他の類似する操作を遂行して、各反復で高圧縮のビットパターンの別の組合せの爆発に再帰的に導くことができる状況ｂ）、ならびにたとえば１キロバイトで記述することができる任意の可能な関数の追加操作が、おおよそ１０億対１の圧縮比をそれぞれ明らかにする、上記のａ）およびｂ）で記述した操作が作り出す２＊＊８０００のまったく異なる倍数の総計を生み出す状況ｃ）を生み出す。

【００２６】

第４に、実用的圧縮機構を生み出すために、その数のコルモゴロフ複雑性がほぼ確実に２０バイトよりもはるかに少なくなる条件の下で、実際に遭遇する特有の別個のビットパターンが何であれ、一意に区別することができる機能上達成可能なマッピングを生み出すことが必要であり、それには、ちょうど２０バイトの組合せ複雑性を使い尽くすために、ビッグバン以来、宇宙の年齢の現在の推定値の２００万～３００万倍の間、休みなく走る、毎秒１０＊＊－２４の操作を遂行することができるコンピュータを必要とすることが留意される。

【００２７】

当初、プログラム記憶式コンピューティングの存在を確立する、いわゆるフォン・ノイマン・アーキテクチャが前兆となった、これまで隠されてきた双方向記憶計算等価関係は

、非常に速いマイクロプロセッサを利用して、データを計算することを介して完全に実を結び、それにより、上記で記述したビッグデータの問題が、非常に大規模データストアを記憶および移送する費用が高いという避けられない結果に由来するのではなく、むしろ、集積回路およびムーアの法則が出現する前の、コンピュータ科学の夜明けにさかのぼる、非常により原始的な機械の能力および制約に基づき設計された手法で、現代のコンピュータシステムでデータを表現し、処理し続けることに由来し、上記で示したように、情報、データエントロピー、圧縮性、および複雑性の理論の分野に非常に悪い影響を及ぼした、長い間一般に抱かれた誤った原理という、決定的に重要な複雑でわかりにくいものを作り出した。

【００２８】

これらの誤った制約（軌道に到達するために必要とされる燃料の重さが、脱出速度に到達するのを妨げるので、宇宙飛行は不可能であるという第二次世界大戦後、一般に抱かれた見解に類似し、この誤った制約は、多段ロケットにより解決された）の旧来の影響から解放され、十分速いプロセッサの存在は、以下に記述するように、比較的簡単な概念的手法で、ＤＣおよびＴＤの方法および装置を介してデータを記憶するのではなく、むしろ計算することを可能にする。

【００２９】

図１は、いくつかの実施形態によるデータ変換システム１００の例を描く構成図を示す。データ変換方式１００は、大規模データストア１０２、データ変換機器１０４、より小さな表現１０６、および宛先／送信元１０８を含む。より小さな表現への変換では、シーケンスは、大規模データストア１０２から宛先／送信元１０８に向けて流れる。大規模データストアに戻すデータ再変換では、シーケンスは、宛先／送信元１０８から大規模データストア１０２に流れる。いくつかの実施形態では、本明細書で記述するデータコンパイラ（ＤＣ）は、「変換機器」と呼ばれる場合があり、本明細書で記述するチューリング・デデキント機器（ＴＤ）は、「再変換機器」と呼ばれる場合がある。

【００３０】

大規模データストア１０２は、圧縮されていない、改変されていない形態のデータを表す。大規模データストア１０２は、未加工のデータを含んでもよい。いくつかの実施形態では、大規模データストア１０２は、自身に基づき意図された操作を行うためにコンピュータシステムのプロセッサにより処理されることになる。いくつかの実施形態では、大規模データストア１０２は、プロセッサが処理し、ユーザに提示される、文書、画像、写真、ビデオ、およびオーディオなどのようなユーザデータを含んでもよい。いくつかの実施形態では、大規模データストア１０２は、命令が指令する操作を遂行するためにプロセッサが実行すべき命令を含むプログラムファイルを含んでもよい。いくつかの実施形態では、大規模データストア１０２は、コンピューティングシステムが管理するハードウェア資源およびソフトウェア資源を管理するためのシステムデータを含んでもよい。大規模データストア１０２を、メモリに、永続記憶領域に、またはそれらの組合で記憶してもよい。大規模データストア１０２は、コンピューティングシステムにより、部分的に消費されてもよい。

【００３１】

データ変換機器１０４は、データをより小さな表現に変換するための機器（たとえば、ＤＣ）、および当初の大規模データストアに戻して再変換するための機器（たとえば、ＴＤ）を含む。データ変換が行われたとき、大規模データストア１０２は、より小さなデータ表現１０６に変換される。本開示によれば、本明細書に記述する変換技法は、より小さなデータ表現１０６を生成するために採用される。データ再変換を行うとき、より小さなデータ表現１０６は、これらの技法の逆を使用して大規模データストア１０２に戻して変換される。本開示によれば、特有のデータ変換アルゴリズムを使用して、変換技法に対応する、大きく拡張可能なデータ変換技法を行う。データを変換および再変換するための特有の技法について本明細書で記述する。

【００３２】

より小さなデータ表現１０６は、大規模データストアに戻して再現することができる小さなフットプリントでデータを表す。より小さなデータ表現は、データ変換機器１０４によってデータ変換された結果としての、変換されたデータ、およびデータ送信元１０８から得られる、より小さなデータ表現を含んでもよい。

【００３３】

いくつかの実施形態では、宛先１０８は、データ変換の結果として生成された、より小さなデータ表現１０６を受信するデータストア（たとえば、メモリ、永続記憶領域、取外し可能記憶領域、またはキャッシュ）を表す場合がある。いくつかの実施形態では、データ送信元１０８は、再変換されるべき、より小さなデータ表現１０６が得られるデータストア（たとえば、メモリ、永続記憶領域、またはキャッシュ）を表す場合がある。いくつかの実施形態では、宛先／送信元１０８は、コンピューティングシステムと一体化した内部記憶装置およびコンピューティングシステムに接続された取外し可能な外部記憶装置を含むローカル記憶装置を含んでもよい。内部記憶装置は、光学ドライブ、ハードディスク、ＳＳＤ（ｓｏｌｉｄ　ｓｔａｔｅ　ｄｒｉｖｅ）などを含んでもよい。取外し可能な外部記憶装置は、メモリカード（たとえば、ＳＤカード）、メモリスティック（たとえば、ＵＳＢメモリ）などを含んでもよい。いくつかの実施形態では、宛先／送信元１０８は、コンピュータネットワークを介してデータ変換を遂行する、ローカル・コンピューティング・システムに接続された外部コンピューティングシステムを含んでもよい。外部コンピューティングシステムは、クラウド・サーバ・システム、特有のエンティティ専用のサーバシステム、クライアント・コンピューティング・システムなどを含んでもよい。いくつかの実施形態では、宛先／送信元１０８は、より小さなデータ表現を伝送または受信することができるデータトランスポート層を含んでもよい。

【００３４】

図２は、本明細書で記述するいくつかの実施形態による、データ変換操作の１つまたは複数を実装してもよいコンピュータシステム２００を例示する構成図である。コンピュータシステム２００は、バス２０２、もしくはデータを伝達するための他の通信機構、データを処理するためにバス２０２に連結された１つまたは複数のハードウェアプロセッサ２０４を含む。１つまたは複数のハードウェアプロセッサ２０４は、たとえば１つもしくは複数の汎用マイクロプロセッサであっても、１つもしくは複数の専用マイクロプロセッサであってもよい。本明細書で特に言及するとき、プロセッサ２０４は、マルチ・コア・プロセッサであってもよい。１つまたは複数のハードウェアプロセッサ２０４は、以下で記述するデータ変換エンジン２１８を中で実行するように構成されてもよい。

【００３５】

コンピュータシステム２００はまた、一時的データ、および１つまたは複数のプロセッサ２０４が実行すべき命令を記憶するためにバス２０２に連結された、ランダム・アクセス・メモリ（ｒａｎｄｏｍ　ａｃｃｅｓｓ　ｍｅｍｏｒｙ、ＲＡＭ）、キャッシュ、および／または他の動的記憶装置などのメインメモリ２０６を含む。命令は、本開示で記述するデータ変換アルゴリズムに従って１つまたは複数のプロセッサ２０４に、（当初の）データをより小さなデータ表現に変換させ、より小さなデータ表現を大規模データストアに再変換させるように構成されたデータ変換エンジン２１８を含む。メインメモリ２０６はまた、再変換されたデータを１つまたは複数のプロセッサ２０４が処理することができるように、データ変換エンジン２１８を実行することにより生成された、再変換されたデータを一時的に記憶するために使用されてもよい。そのような命令は、１つまたは複数のプロセッサ２０４がアクセス可能な記憶媒体に記憶されたとき、コンピュータシステム２００を、命令で指定された操作を遂行するようにカスタマイズされた専用機械にする。

【００３６】

一般に、「エンジン」という単語は、ソフトウェアエンジン（たとえば、機械可読媒体上に具体化されたコード）またはハードウェアエンジンを構成する場合がある。「ハードウェアエンジン」は、ある種の操作を遂行することができる有形の構成単位であり、ある種の物理的手法で構成されても、配列されてもよい。さまざまな例示的実施形態では、１

つもしくは複数のコンピュータシステム（たとえば、独立型コンピュータシステム、クライアント・コンピュータ・システム、またはサーバ・コンピュータ・システム）、またはコンピュータシステムの１つもしくは複数のハードウェアエンジン（たとえば、プロセッサ、またはプロセッサのグループ）は、本明細書で記述するように、ある種の操作を遂行するように動作するハードウェアエンジンとしてソフトウェア（たとえば、アプリケーション、またはアプリケーション部分）により構成されてもよい。

【００３７】

コンピュータシステム２００は、１つまたは複数のプロセッサ２０４のための静的情報および命令を記憶するために、バス２０２に連結された読出し専用メモリ（ｒｅａｄ　ｏｎｌｙ　ｍｅｍｏｒｙ、ＲＯＭ）２０８または他の静的記憶装置をさらに含む。特有の実装に応じて、１つまたは複数のプロセッサ２０４がデータ変換操作を行うとき、データ変換エンジン２１８をメインメモリ２０６にロードしもよい。

【００３８】

データおよび命令を記憶するために、磁気ディスク、光ディスク、またはＵＳＢサムドライブ（フラッシュドライブ）などのような記憶装置２１０を提供し、バス２０２に連結する。記憶装置２１０は、大規模データストア２２０を含む。

【００３９】

コンピュータシステム２００は、コンピュータユーザに情報を表示するために、バス２０２を介して陰極線管（ｃａｔｈｏｄｅ　ｒａｙ　ｔｕｂｅ、ＣＲＴ）またはＬＣＤ表示装置（またはタッチスクリーン）などの１つまたは複数の出力機器２１２に連結されてもよい。英数字および他のキーを含む１つまたは複数の入力機器２１４は、１つまたは複数のプロセッサ２０４に情報およびコマンド選択を伝達するために、バス２０２に連結される。別のタイプのユーザ入力機器は、１つまたは複数のプロセッサ２０４に方向情報およびコマンド選択を伝達するための、および１つまたは複数の出力機器２１２上でカーソル移動を制御するための、マウス、トラックボール、またはカーソル方向キーなどのカーソルコントローラ２１６である。この入力機器２１４は、典型的には機器が平面内の位置を指定できるようにする第１の軸（たとえばｘ）および第２の軸（たとえばｙ）という２つの軸で２つの自由度を有する。いくつかの実施形態では、カーソル制御と同じ方向情報およびコマンド選択を、カーソルなしにタッチ画面上でタッチを受信することにより実装してもよい。

【００４０】

コンピュータシステム２００は、１つまたは複数のコンピューティング機器が実行する、実行可能なソフトウェアコードとして大容量記憶装置に記憶してもよい、ＧＵＩを実装するためのユーザ・インタフェース・モジュールを含んでもよい。このモジュールおよび他のモジュールは、例として、ソフトウェア構成要素、オブジェクト指向ソフトウェア構成要素、クラス構成要素、タスク構成要素などの構成要素、プロセス、関数、属性、手順、サブルーチン、プログラムコードのセグメント、ドライバ、ファームウェア、マイクロコード、回路、データ、データベース、データ構造、テーブル、アレイ、および変数を含んでもよい。

【００４１】

コンピュータシステム２００は、カスタマイズされたハード・ワイヤード・ロジック、１つもしくは複数のＡＳＩＣまたはＦＰＧＡ、ファームウェア、および／またはコンピュータシステムと組み合わせてコンピュータシステム２００を専用機械にする、またはプログラムするプログラムロジックを使用して、本明細書で記述する技法を実装してもよい。いくつかの実施形態によれば、本明細書の技法は、メインメモリ２０６に包含される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１つまたは複数のプロセッサ２０４が実行することに応答して、コンピュータシステム２００により遂行される。そのような命令は、記憶装置２１０などの別の記憶媒体からメインメモリ２０６の中に読み込まれてもよい。メインメモリ２０６に包含される命令のシーケンスを実行することにより、１つまたは複数のプロセッサ２０４に、本明細書で記述する処理ステップを遂行させる。代替実

施形態では、ソフトウェア命令の代わりに、またはそれと組み合わせて、配線接続された回路を使用してもよい。

【００４２】

コンピュータシステム２００はまた、バス２０２に連結された通信インタフェース２１８を含む。通信インタフェース２１８は、１つまたは複数のローカルネットワークに接続された１つまたは複数のネットワークリンクに連結する双方向データ通信を提供する。たとえば、通信インタフェース２１８は、サービス統合デジタル網（ｉｎｔｅｇｒａｔｅｄ　ｓｅｒｖｉｃｅｓ　ｄｉｇｉｔａｌ　ｎｅｔｗｏｒｋ、ＩＳＤＮ）カード、ケーブルモデム、衛星モデム、または対応するタイプの電話回線へのデータ通信接続を提供するモデムであってもよい。別の例として、通信インタフェース２１８は、互換性のあるローカル・エリア・ネットワーク（ｌｏｃａｌ　ａｒｅａ　ｎｅｔｗｏｒｋ、ＬＡＮ）（またはＷＡＮと通信するためのＷＡＮ構成要素）へのデータ通信接続を提供するためのＬＡＮカードであってもよい。さらにまた、無線リンクバイ会談_追加翻訳文を実装してもよい。任意のそのような実装形態では、通信インタフェース２１８は、さまざまなタイプの情報を表すデジタル・データ・ストリームを運ぶ電気信号、電磁信号、または光信号を送信し、受信する。

【００４３】

ネットワークリンクは、典型的には１つまたは複数のネットワークを通して他のデータ機器へのデータ通信を提供する。たとえば、ネットワークリンクは、ローカルネットワークを通して、インターネットサービス提供者（Ｉｎｔｅｒｎｅｔ　Ｓｅｒｖｉｃｅ　Ｐｒｏｖｉｄｅｒ、ＩＳＰ）が運営するホストコンピュータまたはデータ設備への接続を提供してもよい。ＩＳＰは、次に、今では一般に「インターネット」と呼ばれるワールド・ワイド・パケット・データ通信ネットワークを通してデータ通信サービスを提供する。ローカルネットワークおよびインターネットは両方とも、デジタル・データ・ストリームを運ぶ電気信号、電磁信号、または光信号を使用する。コンピュータシステム２００の間でデジタル信号を運ぶ、さまざまなネットワークを通る信号、およびネットワークリンク上で通信インタフェース２１８を通る信号は、伝送媒体の例示的形態である。

【００４４】

コンピュータシステム２００は、１つまたは複数のネットワーク、ネットワークリンク、および通信インタフェース２１８を通して、メッセージを送信し、データを、詳細にはより小さなデータ表現を受信することができる。インターネットの例では、サーバは、インターネット、ＩＳＰ、ローカルネットワーク、および通信インタフェース２１８を通して要求されたアプリケーションプログラム用コードを伝送してもよい。

【００４５】

先行する節で記述した処理、方法、アルゴリズムの各々を、コンピュータハードウェアを備える１つもしくは複数のコンピュータシステムまたはコンコンピュータプロセッサが実行するコードモジュールの中に具体化してもよい、またはそのコードモジュールにより完全に、または部分的に自動化してもよい。プロセッサおよびアルゴリズムを、アプリケーション特有回路の中に部分的に、または全体として実装してもよい。

【００４６】

データ変換機器１０４は、本明細書で記述するように、一般に行われるようにデータを変換し、再変換するように動作する。

【００４７】

「データコンパイラ」という用語は、最適化コンパイラが、人間にとってより容易に理解できるが機械にとってはるかにきわめてより非効率な形式から、機能上等価な、コンピュータにとってはるかにきわめてより効率的であるが、それにほぼ対応して人間にとってはるかにほとんど理解できない形式にコンピュータコードを変換する限り、啓発的であるのに対して、本明細書で記述するようなデータンパイラ（ＤＣ）は、（記憶し、あちこち動かすのに非常に費用がかかる）コンピュータにとってきわめて非効率なビットパターンの形をとるデータを、当初、データコンパイラの中に入力された当初のビットパターンを

迅速に、無損失に再計算する能力をＣＰＵ、グラフィックス、または他の処理エンティティに提供する、機械のオン・チップ・キャッシュの中に全部または大部分適合させることができるはるかにより小さなカーネルに変換することにより類似の機能性を遂行し、一方では、対応するチューリング・デデキント機器（ＴＤ）は、ＤＣが作り出したカーネルからから、この当初のビットパターンを再計算することを必然的に高速に、無損失に遂行する。とりわけ、ＤＣが遂行する処理を「ロード時」と考えることは有用であり、同様に、ＴＤが遂行する処理を「実行時」と考えることは有用である。

【００４８】

同様に、「チューリング・デデキント機器」という用語は、チューリング機械とデデキントの切断の両方を大いに連想させる、ＴＤが所有する重要な方法論的設計特徴を考慮することにより、有用に概念化することができる手法でこの再計算を遂行するためにＴＤが使用する方法に由来する。ビットパターンｋの組合せ複雑性に注目する手法で、ならびに左から右まで連続する各点がＫ０＋１の数値に等しいビットパターンＫ１を表すラインセグメントを考慮することにより、ロード時にＤＣが十分に効率的であるとこれまで判断してきたこの特定のビットパターンｋを正確に再現するように設計された、より汎用な方法の特有な例を介して、高速プロセッサ（たとえは、プロセッサ２０４）により、ＴＤが高圧縮のカーネルから特有のビットパターンｋを再計算する手法で、ｎビットを包含する特有のビットパターンｋを考慮することは有用であり、この場合、Ｋ０は、このラインセグメント上のすぐ左側にある点を表し、このラインセグメント上で最も左側の位置にある最初の点は、すべてオフビットを包含するｎビット長のビットパターンを表し、このラインセグメント上の最も右側の位置にある最後の点は、すべてオンビットを包含するｎビット長のビットパターンを表し（すなわちこの場合、各ラインセグメントは、正確に２＊＊Ｎの別個の点を包含する）、そこでは、以下で記述する方法に従ってＤＣによりこの目的のために両方とも前もって設計されていた、特有のカーネルを操作する特有の命令セットを介して、ならびに／またはコンピューティング機器により完全に自動化された方式で、および／もしくは代わりに、人間の介入による助けを借りて当業者が設計および／もしくは遂行することができる代替方法および／もしくは変形方法を介して、１つまたは複数の高速プロセッサが遂行する特有の再計算を通して、ＴＤが無損失に再現するように特に構成された特有のかなり大きなｎビット長の各ビットパターンｋは、このラインセグメント内のデデキントの切断を表す。

【００４９】

さらに、（上記で参照した、長年にわたり現在も依然として受け入れられている、コルモゴロフ複雑性についての証明可能に誤った理解を前提になおさら関連性があり、啓蒙的な）２＊＊Ｎ点を包含する各ラインセグメントを介して達成されるような、この処理に伴う組合せ複雑性に注目するのに加えて、特有なそのような各ビットパターンｋをこのラインセグメント内のデデキントの切断として考慮することを特に適切にさせている別の理由は、ほとんどすべての場合、ＴＤが１つまたは複数の実行時に遂行するために、ＤＣがロード時に前もって確立していたビットパターンを再計算する方法は、ｋを直接に、無損失に再計算できるようにするのではなく、むしろ、ｊとｋの間の差の絶対値が小さいほど、それだけ初期近似がよくなり、かつ誤差項が小さくなるという条件の下で、加算もしくは減算、またはｊをｋに変換する誤差項に対する何らかの他の簡単な関数を介して、その後ｋが導出されるｊを計算する高圧縮のカーネルから繰返しｋを無損失に再計算することができるように、ｋに十分近い何らかの他のビットパターンｊを計算することを可能にするという現実に由来し、この場合も、デデキントの切断の特性に注目する一方、これらの機器が、最も典型的な場合に、フーリエ変換などの古典的双方向変換を実装するために設計された機器が使用する方法と最も一般的水準で比較して共有する方法を示す。

【００５０】

同様に、ＴＤがｊおよび／またはｋを再計算する処理が、テープを片方向または前後に動かすチューリング機械の能力、すなわち、既知の速度で進める能力に対応する、単調であっても、単調ではなくてもよい蓄積処理を介する限り、ｊおよびｋなどのビットパター

ンを再計算する手法は、チューリング機械を連想させ、その結果、ｋを最も効率的に無損失に再現するＴＤの能力の鍵は、より小さな誤差項を伴う、より正確な近似を次第に導出するＤＣの能力にあり、近似のうち最も効率的なのは、最も典型的には、システムに既知の、ｊを正確に繰り返して再現するために必要とされる蓄積速度に基づき、実行時にＴＤにより実行され、ＴＤがビットパターンｊを作り出したときに蓄積を止める方法は、概念的には、チューリング機械が、停止させるべき正しい時間がわかっていることと相まって、正確で適切な所定の手法でテープを片方向または前後に動かすことに対応する。

【００５１】

ＴＤの本質的関数が、ロード時にＤＣがすでに確立していた不可欠な再計算を遂行するために必要な手順を実行時に実行するためにあるという前提で、上記の簡潔な説明は、本明細書で提示した実施形態を可能にする詳細な記述と共に、当業者がＴＤを実装することができるだけの十分以上の情報を提供する。

【００５２】

当初、ＤＣの中に入力されたビットパターンの、何らかの別個に識別されたサブセットまたは全体を再生するために、ＤＣがＴＤにカーネルとコードの両方を引き渡すカーネルに対してＴＤがコードを「単に」実行する限りでは、ＴＤおよびＤＣのうち概念的により簡単なＴＤと対比して、ＤＣは、受信するビットパターンｋごとに、カーネルとカーネルを操作するコードの両方を決定することを含む「困難な作業を遂行する」必要があり、当初、典型的にははるかにより大きなビットパターンｋを「そのままで」保存し、記憶し、転送し、取り出す伝統的な旧来の処理よりも非常に機械効率な手法で、ＴＤが、ＤＣがＴＤに引き渡した高圧縮のカーネルを当初のビットパターンｋに戻して変換することを可能にする、十分に小さな空間フットプリントを同様に示す、ＤＣが同様に提供する必要のあるコードと共に、ＤＣがＴＤに提供する必要がある、はるかにより小さなカーネルから、高速プロセッサを利用してｋを再計算するＴＤの能力に今では取って代わられた、ビットパターンｋを記憶する必要性をなくすというはっきりした目的で、ＤＣは、これらのカーネルおよびコードを設計し、生み出す必要がある。

【００５３】

「カーネル」という用語は、明示的または暗示的に、これら２つの別個の方法のうち用語「カーネル」が使用されている状況を介して、ありとあらゆるそのような具体的実例で明らかにされる限り、パラメータおよびパラメータを操作するコード、または代わりにコードが操作するパラメータだけを、この潜在的にいろいろに解釈できる２重の意味が混乱の潜在的原因または実際の原因に決してなることなく参照するために、「カーネル」という用語を合理的に使用することができることに留意されたい。

【００５４】

ＤＣの実際の動作に関してより本質的には、以下で理解されるように、２つの別個の方法を使用して、ＤＣを可能にする２つのまったく異なるタイプの実施形態を当業者が生み出すことを可能にする際に有用であることがそれぞれ証明されている，以下の３つの点について予備的に簡単に説明する。

【００５５】

第１に、重要なことには、上記で参照したように、適切なデータ完全性検査と共に、十分に高速なプロセッサが存在するという前提の下で、当業者がＤＣを作り出すことを可能にするために、別の性能が必要とされる。これらの性能は、ａ）直感的には、ある種の「呼び水」に対応する、圧縮されていない大きなビットパターンの、かなりの数ではあるが、それにもかかわらず比例してとても小さいサブセットを無損失に、正確に、迅速に再計算する性能（はるかにより小さな高圧縮のカーネルからループを介してそうすることを含む）、ならびにｂ）、その特有のビットパターンｋ、またはｋに十分近い何らかの他のビットパターンｊを、ｋに漸近的に接近する一連の連続するそのようなｊａを再計算することを可能にする手法で再計算することができ、一方では、カーネルと、この圧縮されたカーネルから当初のビットパターンを再計算するために必要な不可欠なコードの両方を、このコードおよびこのカーネルの十分にかなり大きなサブセット、またはより有益にはこの

コードおよびこのカーネルの全体がマイクロプロセッサのオン・チップ・キャッシュの内部に適合することを可能にするほどに十分に小さく保つように、ｎビットを包含する任意の特有のビットパターンｋを標的にすることができる能力から構成される。

【００５６】

第２に、ＤＣを実装するために目標ａ）とｂ）の両方を達成する、２つの完全に別個の独立した方法は、以下で示すように、これらの２つの別個の方法のいずれか一方を利用することが、本発明を可能にする実施形態を構成し、一方では、現在公知の本発明の好ましい実施形態が両方の方法を一緒に利用するという状況の下で存在する。

【００５７】

第３に、上記で直接に参照したように、不可欠の２重の機能性ａ）およびｂ）を達成するためにＤＣが別々に、または共同で利用するこれらの２つの別個の独立した方法は、従来受け入れられていたコルモゴロフ複雑性の誤った解釈が認識できなかった、以前に参照した４つのまったく異なる問題のうち２つに依存し、そこでは、以下で記述するＤＣを実装する第１の方法は、従来の過度に厳格なコルモゴロフ複雑性の解釈に対する第３の異論で一緒に参照した一連の等価関係の存在と共に、任意の大きな理論的圧縮性を生み出すことができることを直接利用し（これらの２重の性能は、前述の必要とされる２重の重要なａ）およびｂ）の機能性との、非常に近い、偶然ではない概念的対応を示す）、一方では、これらの２重の重要なａ）およびｂ）の機能性を独立して単独で、および／または共同で実装する第２の方法は、上記で提示した決定的に欠点のある旧来のコルモゴロフ複雑性の解釈を伴うこれら４つの上述の問題のうち第１の問題について参照したように、仮想メモリアドレスおよびタイムスタンプのような特性を介して、同一ビットパターンの情報内容を区別する方法を依然としてより効率的に利用する。

【００５８】

ＤＣを作り出すのに役立つ上記の予備的考察の助けを借りて、完全に機能できるＤＣを生み出すことを可能にする手法で、上記の２つの重要な２重の機能性ａ）およびｂ）を達成するために（本発明が利用するこれら２つの別個の方法によりＤＣを実装するために）採用される技法の考察に直接に目を向ける。

【００５９】

第１に、ＤＣの第１の実施形態の範囲内で、比例して少数のすべての可能なサイズｎのビットパターンであって、それにもかかわらず、プロセッサの速度、および不可欠な計算を遂行するために必要な時間だけにより、任意の大きな理論的圧縮比が実際上は制限される非常に小さなカーネルから得られる別個のまったく異なる任意の大きなビットパターンの比較的大きな数字のセットを表すサイズｎのビットパターンを生み出すことは、大きなインデックスが示すような数多くの反復を通して小さな空間フットプリントおよび数少ないコード行を包含するループを介して、当業者により容易に達成可能である。

【００６０】

その上、有用であるとみなされる場合、計算全体の速度、およびその後、依然としてより大きな任意のビットパターンをさらにより迅速に作り出す能力は、追加の並列化およびマルチ・コア・アーキテクチャを通してだけではなく、すでに非常に大きな圧縮比が少し低減するという代価も払って、任意のサイズのワード境界を越える桁上げを無視することによってもさらに改善することができる。機能上の依存性が低減することを介して並列化をさらにまた高めるのに役立つこの手順は、本計算の唯一の要件が、数学的精度ではなく、整合性のあるビットパターン再現性であるという事実により達成可能である。

【００６１】

多分、最も簡単なそのようなループは、任意の基数の浮動小数点数の任意の大きなべき乗をとることである。他のそのような有効なループは、潜在的に大きなｋに関して再帰的にｋ回、ＸをＮ乗することと、たとえば、複素平面ではなく実平面内で遂行されるマンデルブロー集合（Ｍａｎｄｅｌｂｒｏｔ　Ｓｅｔ）の生成関数に正確に類似するものを含むがそれに限定されない、ｋ回の反復ごとにループ内部で１つまたは複数のパラメータを増大させる、減少させる、または他の方法で機能上変換することを介して、ループ内部の内

容を周期的に改変することと、モジュロｍの任意の計算と、任意の平均と、加重平均と、２つ以上の生成関数の間の数値差、または２つ以上の生成関数の任意の組合せの他の簡単な関数と、当業者に認識可能および／または発見可能な数多くの他の変形形態と共に、他の簡単な論理的操作および／または他の数学的操作を遂行することに加えて、さまざまな可能なビットシフト操作、ビットパターン反転、異なるビット・パターン・セグメントの切断および／または接合および／または連結に加えて、定数を包含する付加項を含めることとを含むが、それらに限定されず、これらの可能性のすべては、かなり大きな数の非常に大きな高圧縮の別個の、それにもかかわらずビットパターンの総コルモゴロフ複雑性の非常に小さい部分を表すビットパターンの立脚点から始めて、鳩の巣原理／計数議論を反転することと直感的にはほぼ等価な、さらにかなり大きな誤差項または誤差関数を含めることにより、さらに拡張することができる。

【００６２】

混乱の可能性を回避するために、本明細書で参照するさまざまなパラメータｋは、当初、ＤＣの中に入力されたとき、ビットパターンｋとの、必要な機能上の、または他の関係をまったく示さないことを意図するものではない、またはそれらの関係をまったく有しないことを意図するものではないことに留意されたい。

【００６３】

本明細書で以後、上記で直接記述したように、典型的には高圧縮のそのような小さな空間フットプリントの高インデックスループの２つ以上のセット、または上記で直接列挙した基準を多かれ少なかれ理想的に近似する任意のそのような他の計算機構を、ＨＣＳＳＦＨＩＬ（ＨＩＣＫＳ－ＦＩＬＬ、またはＨＩＣＫＳＶＩＬＬＥまたはＬＯＮＧ　ＩＳＬＡＮＤと発音する）と呼び、一方では、本明細書で以後、直前の段落で記述した例を含むが必ずしもそれに限定されない、いくつかのタイプの高圧縮の小さな空間フットプリントの高インデックスループにより、多分最良に、かつ最も効果的に例示される、ＨＣＳＳＦＨＩＬ内部の別個の各計算単位を、これらの多かれ少なかれ理想的な高圧縮の小さな空間フットプリントの高インデックスループ計算単位の各々の内容全体が単一ループ内部、または２つ以上のネスティングされたループ内部に完全に包含されているかいないかに関係なく「ＬＯＯＰ（ループ）」と呼ぶ。

【００６４】

ＨＣＳＳＦＨＩＬは、定義により、任意の数の２つ以上のＬＯＯＰを包含することができるが、その一方で、本発明の好ましい実施形態では、ＨＣＳＳＦＨＩＬあたり、より多くのＬＯＯＰが、ＴＤ内の処理ステップがより多くなるという代価を払って、より効率的標的機構を提供する状況の下で、各ＨＣＳＳＦＨＩＬ内のＬＯＯＰの数は、実際的な最適化効率について考察することにより決定される。

【００６５】

したがって、２つ以上のループを接続して、複数のループを組み合わせてＨＣＳＳＦＨＩＬにするためにＤＣが遂行する操作の最も簡単な説明となる例を同様に提供するようなＨＣＳＳＦＨＩＬを形成する最も簡単な手段は、そのＨＣＳＳＦＨＩＬ内のループの各々の出力の数値を合計して、その特定のＨＣＳＳＦＨＩＬ内部のＬＯＯＰの各々が出力する別個のまったく異なる数値の合計に、関連する数値が等しい特有のＨＣＳＳＦＨＩＬに対応する結果のビットパターンを作り出すことから構成されるが、一方では、ＨＣＳＳＦＨＩＬを作り出すために複数のループを一緒に接続する他の手段を利用して、２つ以上のループを一緒に接続して、ＨＣＳＳＦＨＩＬを形成することができる。したがって、各ＨＣＳＳＦＨＩＬ内部のＬＯＯＰの特定の多数のＬＯＯＰの出力の内容に対して必ずしも加算関数ではないこれらの代替関数のうちこれらの１つまたは複数を実行する結果として、ＨＣＳＳＦＨＩＬが作り出すビットパターンの数値に数値が対応する、典型的にはかなりより長いビットパターンを作り出すために、２つ以上のＬＯＯＰが作り出す複数のビットパターンを一緒に組み合わせる、加算以外の代替関数は、複数のＬＯＯＰをＨＣＳＳＦＨＩＬに変えるためにＤＣが利用する接続機構を提供することができる。

【００６６】

同様に、２つ以上のＬＯＯＰを接続してＨＣＳＳＦＨＩＬを形成する方法に類似して、次の上位レベルで、ＤＣの対応するＴＤが作り出す出力を提供するためにその後再計算される入力をＤＣに供給する各ビットパターンを２つ以上のＨＣＳＳＦＨＩＬに分割することができるが、そうする必要はない。

【００６７】

その上、合計が、複数のＬＯＯＰを接続してＨＣＳＳＦＨＩＬを形成するために使用することができる最も簡単ではあるが唯一というわけではない可能にする機構を提供するのと同様に、これらのまったく異なる部分から、ロード時にＤＣの中に最初に入力された当初のビットパターンを無損失に再生するために、ビットパターンを２つ以上のＨＣＳＳＦＨＩＬに分割し、続いて、実行時に複数の別個のより小さな一部の再計算されたビットパターンをＴＤが再接続することを可能にする最も簡単な方法は、ＤＣの中に最初に入力された当初のビットパターン全体をＫのまったく異なるＮビット長ビットパターンに最初に分割することであり、次いで、Ｋのまったく異なるＮビット長ビットパターンの各々は、ＴＤによる実行時の連結を介して再結合するために、ロード時にＤＣにより確立されたように、別々に処理される。さらにまた、上記で論じたように、２つ以上のＬＯＯＰを組み合わせてＨＣＳＳＦＨＩＬを作り出すことに類似して、本発明の実施形態で、ビットパターンを２つ以上のＨＣＳＳＦＨＩＬに最初に分割するために利用される任意の関数を使用してもよく、次いで、その逆操作を採用して、実行時に結果として得られる出力を再結合して、当初のビットパターンを無損失に再現する。

【００６８】

本発明の２重の目標である圧縮性および標的設定の考察に戻ると、上記で参照したような（すでに指摘したように、基本的な鳩の巣原理／計数議論を逆に走らせるとき、有用には、直感的に理解することができる）長く受け入れられてきた、コルモゴロフ複雑性についての誤った理解で誤った考えをさらす、本明細書の４つの議論の３番目の議論ですでに前もって示したように、理論的に任意の非常に高い圧縮比を伴うより少数のビットパターンから、少しから幾分かまで、より小さい圧縮比を伴う、かなり多数のビットパターンまで、圧縮比を「平準化する」ことを可能にする、まったく異なるが、関係がある機能上の等価性の結果として、本来なら別個のまったく異なるビットパターンのセットの中からこの能力が自動的に確立する一連の等価関係と相まって、高速プロセッサを利用することにより、ＴＤについての本明細書の初期の記述だけではなく、小さな空間フットプリントを包含するカーネル内部の少数のパラメータを理論的に任意の大きな多数のビットパターンに無損失に展開する能力について前の説明でも示したように、チューリングおよびデデキントの方法についての本明細書の前の議論で直感的かつ本質的にすでに記述したように、上記のｂ）で示した不可欠の標的設定機能性に留意されたい。上記で記述したようなこれらの特有の原理および技法は、実際上は本発明の１つまたは複数の実施形態を達成するために以下で直接に提示する詳細に従って実装することができる。

【００６９】

ＤＣが、各反復で出力する最終結果の各ビットビットパターンｊａをＤＣの入力ビットパターンｋの値にますますより近くする目的で、ループおよびＨＣＳＳＦＨＩＬを連続して反復することによって処理するとき、ＨＣＳＳＦＨＩＬを順次生み出す処理の間、上記で定義するような複数のＬＯＯＰを包含するＨＣＳＳＦＨＩＬについて考えてみる、そして各ＨＣＳＳＦＨＩＬを備えるループの各々は、この処理の間、ＨＣＳＳＦＨＩＬの内部ループパラメータの全体が定数であるかどうか、またはループが遂行する反復回数が何らかの他の停止イベント、たとえば割込みにより決定されるかどうかの関数として、生成関数、または関連する値を伴う特有のビットパターンもしくは経時的に動的に変化するハイブリッド状態を表すと有用に考えることができることに留意し、ＨＣＳＳＦＨＩＬの内部ループパラメータの全体が定数である場合、そのループは、関連する数値を伴うビットパターンを出力する、または場合によっては、ＨＣＳＳＦＨＩＬのループインデックスを含むＨＣＳＳＦＨＩＬのパラメータの１つもしくは複数が変数であり（この場合そのループは、より有用には生成関数と考えることができる）、ループが遂行する反復回数が何らか

の他の停止イベントにより決定される場合、ループは、ＨＣＳＳＦＨＩＬが停止する前に継続している生成関数として機能し、ＨＣＳＳＦＨＩＬが停止した時点で、ＨＣＳＳＦＨＩＬの特有の関連する数値を伴う特有のビットパターンの一部であるＨＣＳＳＦＨＩＬを出力し、ＨＣＳＳＦＨＩＬに寄与する。

【００７０】

最も重要には、さらに、これらのループの各々を包含するＨＣＳＳＦＨＩＬに加えて、特有のＨＣＳＳＦＨＩＬの内部にある各ループは、ＤＣに既知の、または既知の１次の、および２次の、および場合によってはより高次の微分に関して表現可能な、各ループ自体の特有のＡＣＣＵＭＵＬＡＴＩＯＮ　ＲＡＴＥ（蓄積速度）を明らかにし、それに基づき、ＣＤは、ループがいつ停止するかを決定するインデックスを含むがそれに限定されない、ＤＣのループ内部の１つまたは複数のパラメータを常に変えることができ、そのインデックスに加えて、ＤＣは、特に選ばれたパラメータを伴う新しいループをいつでも追加することができること、ならびに／または、当初、ＤＣに入力されたビットパターンに対応する一意の関連する入力値と、当初、ＤＣに入力として提示されたビットパターンの関連する値をなおいっそうより近く近似するタスクが、上記で論じたように２つ以上のＨＣＳＳＦＨＩＬに前もって分割されていたかどうかの両方により、新しく設計された連続する「次の反復の」ＨＣＳＳＦＨＩＬそれぞれが出力する新しく改変されたビットパターンｊａ＋１が、そのＨＣＳＳＦＨＩＬが、最終的に決定されたときに最もよく近似しようと試みるビットパターンｋの一意の関連する値により決定されるように、ＤＣの標的に対するそのＨＣＳＳＦＨＩＬの直前の「先代」ビットパターンｊａよりも近い、関連する数値を包含することを保証するように常に設計することができる手法で、ＤＣは、そのＨＣＳＳＦＨＩＬがすでに包含する１つまたは複数のループを取り除くことができることに留意されたい。

【００７１】

次に、新しいビットパターンｊａの各々の関連する値が標的ビットパターンｋの関連する値をより近く近似することを確実にするために上記で直接示した手段を含むがそれらに限定されない、ＤＣの自由になる手段の完全性に加えて、ＤＣが上記の反復処理を通してさらにまたより効率的に進むことを可能にするために、当業者に公知のさまざまな他のツールを利用することができることに留意されたい。

【００７２】

ある種の逆操作をあらかじめ計算すること、たとえば、よりよい近似を作り出すために連続してべき乗をとるのではなく連続して根をとること、ならびにさまざまな整数関係検出技法、たとえば、さまざまなより高度でより高速な加重平均を伴うＰＳＬＱを含むがそれに限定されない、とりわけ実験数学で使用される一連のさまざまな統計的および／または複雑な計算最適化法を含む、さまざまな中間標的値をよりよく近似するように設計された計算、ならびに／または本質的に線形時間で素数を発見する研究の過程で具体的に述べられたような、Ｄａｎｉｅｌ　ｇ７Ｊ．　Ｂｅｒｎｓｔｅｉｎが具体的に述べた技法などの切捨て２分検索技法を含むがそれらに限定されないそのような追加の改善はすべて、上記で記述するように、ＤＣおよび／またはＴＤの操作をよりよく最適化するために当業者に明白な手法で使用されるとき、本発明の構成要素を成す部分を表すと考えられるべきである。

【００７３】

言及するに値する、本発明の特有のそのような一代替実施形態は、ビットパターンｋをＤＣに入力する前に、既存の公知の標準的圧縮法を使用して、そのような標準的データ圧縮を遂行するために必要な時間の代価として、以下の考えられる２重の利点を生み出すために、最初に、ビットパターンｋをよりよくランダム化する可能性を含む。ビットパターンｋがよりランダムなビットパターンを包含する利点は、前述の反復処理をＤＣが成功して終了する速度によいまたは悪い影響を及ぼす場合も、及ぼさない場合もある、ランダムではないビットパターンに関連する統計的異常のために、どんな２つの任意のビットパターンも、大きく異なる速度で、上記で記述する反復処理で進行する確率を大きく低減する

ことにより作り出される、考えられる分布出力の「平準化」が高まることを両方とも利用することを含む。当初、ＤＣに入力されるビットパターンｋを、最初にランダム化する２つ目の潜在的利点は、高度にランダム化された所与の入力ビットパターンｋと不可欠の等価関係を有するビットパターンの非常に小さなサブセットに適用される統計的異常をよい方に利用することにあり、この技法の潜在的利点は、極端な場合に最も明確に明らかであり、極端な場合では、当初のビットパターンｋが十分に小さい空間フットプリントを占有する、および／または当初のビットパターンｋをそのようなより小さいセグメントに十分有利に分割することができ、上記ですでに記述した反復処理に対する代替実装形態として以下ですぐに定義し、記述するような、以下の「ルーズベルト島」法を利用することを可能にする十分に高速な平均処理速度を達成することを可能にする、広範囲にわたるマルチ・コア・アーキテクチャまたはそのような他の方法を介して、おそらくは費用がかかる、大規模並列化を伴う大型サーバを利用することができる。

【００７４】

ビットパターンｋの内容が単にビットパターンｋのコルモゴロフ複雑性に基づき非常にアランダムに分布していることが既知である状況の下でビットパターンｋのサイズだけから、非常に高度の統計的確実性であらかじめ自明に、明白に知ることができる特有のビットパターンｋのあらゆる１対Ｎのランダム変換を統計的に得る状況の下で、ならびにさらに、客観的時間単位で測定したとき、プロセッサの数および平均速度とは無関係に、かつ正しいビットパターンをＤＣが「うまく当てる」まで、上記で指定したように、単にＮのランダム・ビット・パターン変換を遂行し、試験することにより、ＤＣに必要とされるすべての操作をＤＣが遂行することができるカーネルおよびコードを生み出すためにＤＣにより必要とされる必要な実行時間、機能するために（そのような客観的時間単位、たとえば、秒、年などで同様に測定したときに）無理のないと考えられる時間の量とは無関係に、実行時に機能するために無理のないと考えられる時間フレームの範囲内で、不可欠なＮの操作を遂行し、結果として得られる可能性のあるＮのビットパターンを試験するのに十分な処理能力を伴うコンピュータ上にＤＣを実装することができる状況の下で、ＤＣが、正しいビットパターンをうまく当てた時点で、実行時にこれらの操作をＴＤが実行することができるように、ＤＣは、今ではＴＤに利用可能な適切なコードおよびカーネルを単に転送する必要がある。

【００７５】

極端な「ルーズベルト島」条件、または前もって記述したような、はるかにより一般に期待される「正常な」状況の下でのＤＣの動作に関して、例外的な「ルーズベルト島」の場合だけではなく、その前に記述したはるかにより正常な一般的場合の両方で、上記で記述したような「直接当たり」が一般に仮定され、はるかにより正常な一般的場合では、ＴＤが正確に、無損失に計算することができるとＤＣが判断するビットパターンは、最も典型的にはＤＣに入力された標的ビットパターンとの正確な１対１の、ビット対ビットの対応を依然として示すのではなく、むしろ、ＤＣに入力されたビットパターンの関連する数値と、ＤＣが作り出し、ＴＤに提供する出力ビットパターンとの差が十分に小さいので、関連する「誤差項」として定義されるこの数値は、（ビットパターンとの関連性の低い数値という些細な結果として）十分に小さな空間フットプリントを伴う別個のまったく異なるビットパターンとして直接提供されることができる、または代わりに、ＤＣに既知のこの誤差項の何らかの関数は、ビットパターンを計算するために必要な関数とコードの両方が、それら自体十分に小さな空間フットプリントを有するという条件の下で、カーネルの一部として提供されることができる。満たされるべきこれらの要件を可能にする不可欠の前提条件は、上記で記述したような誤差項を効果的に符号化するために必要な関数を明らかにするために、当業者に公知の標準的方法をＤＣが採用する状況（たとえば、前もって参照した整数関係検出法などの実験的な数学的方法を利用するとき）の下でを含み、満足されると考えられ、同様に、当業者に公知の他のそのような方法は、他の公知の方法を利用して、カーネル内部の誤差項を表すために必要とされる空間フットプリントのサイズをさらに低減する手法で、関数を使用することによって直接または間接的に誤差項を表すた

めに、本発明を可能にする実施形態でＤＣにより使用されてもよい。そのような簡単な例を１つ提供するために、マルチ・コア・アーキテクチャでの数多くのコア、および／または機能上の依存性に関係がある、関連する問題をまったく被ることのないｎ重並列処理性能を提供する、きわめて効率的で迅速な並列化を提供する他の手段を、広範囲にわたり利用する大規模サーバ上でのＤＣの実装形態について考えてみる、そしてＤＣが正常に機能する過程で、この同じサーバを利用して、実行時にＴＤを実装すること、ならびにそのＴＤは、ＤＣがＴＤに提供する操作を実装する過程で、ＴＤのｎの利用可能な並列処理ノードの各々で、ｍの操作を迅速に実行し、１／ｍ＊ｎ×誤差項に等しい、より小さな小数部ｘおよび／またはその何らかの依然として小さな関数は、当初、実行時に無損失にＤＣの中に入力されたビットパターンをＴＤが再計算することを可能にするカーネルの中に記憶されることができることについてさらに考えてみる。

【００７６】

すでに具体的に述べたように、ロード時にＤＣが遂行し、実行時にＴＤが実行する、２重の不可欠の高い圧縮性および正確な標的設定機能性を満足させる（上記の説明で前もって「ａ）」および「ｂ）」として具体的に述べた）追加の一連の方法は、本発明の重要な側面を表す。コルモゴロフ複雑性に関して一般に受け入れられている４つの誤った考えのうち１番目の誤った考えを参照する上記の節で前もってついでに指摘したように、本明細書で直接記述するこれらの追加の方法の両方が、共通に共有する同一のビットパターンを示すという事実にもかかわらず、２つ以上の別個の情報内容をさらに区別する代替手段を利用するこれらの追加の方法は、本発明を可能にする好ましい実施形態がこれらの追加の方法の両方を利用するという状況を除き、これらの実施形態の各々を独立して実装することができるという状況の下で、本発明の代替実施形態を構成することができる。

【００７７】

その上、有用には当業者に理解可能な手法で、本説明で示したように、実験数学の分野から得られる技法と共に、さまざまな算術値計算に基づく統計的技法を含むがそれに限定されない多種多様の公知の技法は、本明細書の以下のどこかでさらに詳細に記述するように、当面の問題についてのＤＣの広範囲にわたる知識に基づき、より精密な標的設定技法の比較的大きなセットをＤＣが利用することを可能にする。たとえば、ＰＳＬＱ（２乗ベクトルの部分和およびＬＱ（ｌｏｗｅｒ－ｄｉａｇｏｎａｌ－ｏｒｔｈｏｇｏｎａｌ、下位－対角－直交）行列の因数分解の利用に由来する略語）およびすでに参照したような他の整数関係検出法のような技法の使用を含むこれらの技法は、カーネル内部の誤差項の空間フットプリントをさらに低下させるなどのタスクに有用に適用することができるだけではなく、何らかの標的ｋに、より効果的に接近するために個々のＬＯＯＰまたはＨＣＳＳＦＨＩＬに特有の一連のｊａ項を得ることについて本明細書で記述するように、複数の反復をこつこつと進める、より効率的経路を見つけ出すために使用することができる。そのような方法により明らかになるような、そのようなより深い知識に基づき、ＤＣに既知のこれらの特有の状況の下で、ＤＣは、より直接の経路を利用した場合よりも迅速にｋに接近する経路を見つけ出すために、２つ以上の連続するｊａが、一時的にｋからさらに遠く離れて移動してもよい１つまたは複数の反復を採用してもよい。この点に関連して、ｋにますます近く接近する一連の連続するｊａを常に見つけ出すＤＣの能力について本説明でこれまでに書いたとき、ＤＣの目標を達成するために決定的に重要で有用な操作を遂行するＤＣの能力に関する重要な含意を伴う重大な真の原理を参照していたが、その説明のどの部分についても、上記で記述したように、ｊとｋの差の絶対値を一時的に増大させる２つ以上の連続する反復を意図的に選ぶことにより、ＤＣを関与させる十分に小さい誤差項により距離を直接に表すことができる地点までｊとｋの間の距離を大きく低下させる、より最適化された経路をＤＣが利用するのを妨げると解釈すべきではないことに留意されたい。

【００７８】

そしてさらにより小さな空間フットプリントを達成する、典型的には多くの場合特に便宜主義的なさらに他の手段、ならびに／または再計算および／もしくは他の処理をさらに

スピードアップする手段、ならびに／または本明細書に記述するように、本発明の方法および／もしくは装置と併用して使用するときに当業者に明白であり、および／もしくはすでに公知の方法および／もしくは装置を実装する手段はまた、本発明の一部として考えられるべきである。そのような方法は、これらの部分の各々の間の距離の数値に関する完全な知識、および上記で記述したようにＤＣがどんなに最も有用であるとわかっても漸近的に直接修正される標的とすることができる、変わる蓄積速度もまたＤＣに既知である、既知の高圧縮可能な小さな空間フットプリントの既知の生成関数の大きな集合と共に、たとえば、当初、ＤＣに入力されたビットパターンの各サブセクションの知識も、関連する異なる数値を作り出すことができる、ビットパターンをスライシングおよびダイシングする異なる方法も、これらの複数の可能なスライシングおよびダイシングの操作の下でビットパターンの別個のまったく異なる知識の各々の、他のそのような明白に有用な論理的ビットごとの決定特性も含む、ビットパターンに利用可能な、ＤＣが有する大量の精密で正確な統計データを利用する、上記で具体的に述べた方法を含むがそれらにまったく限定されない。しかし、これは、ＤＣの自由に利用可能な、ＤＣが有するツールのタイプについて決して網羅的ではない。

【００７９】

たとえば、仮想メモリアドレス、タイムスタンプ、および／または他の基準を介してなど、本来ならビットごとに同一である２つ以上のビットパターンの間で元になる情報内容を識別するビットパターン以外の代替手段を通して本質的な情報を示す、本発明が利用する有用な方法の１つの使用法について考えてみる。以下で記述するこれらの追加の方法は、情報内容を表す、および／または本来なら同一の２つ以上のビットパターンのデータ表現機能性を区別するこれらの代替手段が、メタデータ、ならびに／または他のハウスキーピング、もしくは上記で参照し、以下でさらに詳細に短く記述する、継続的に機能するコンピュータの何らかの構成要素を成す部分、および／もしくは補助データ処理操作を形成する他のタスクで表される、それらにより計算される、および／またはそれらの内部に包含される、および／またはそれらにより「吐き出される」ように他の方法で見いだされる値の関連する数値および／または関数に対応する、ビットパターンの大規模リポジトリにアクセスすることによって達成される方法を含むがそれに限定されない。情報内容を表すそのような手段は、複数のポインタ、マッピング、および／または他の内容ラベル付け、内容参照および／または内容を識別する表示、および／または実質的に類似する、および／または同一の有用な機能上の目的に役立つ手順を介してしばしば表すことができるが、この手段の空間フットプリントは、多くの場合かなり異なる。たとえば、それ自体６４ビットのエンティティであるが、タイムスタンプおよび／または仮想メモリアドレスの順序を表す順番もしくはリスト内のそれらの位置により、またはこれらの指し示すエンティティまたは他のそのような機能上のエンティティを、多くの場合、たとえばより大きくより複雑な階層の内部で、異なるレベルで指し示される異なる項目の数の関数として異なる長さポインタにより表すことができる特有のデータ項目を一意に識別するために使用することができ、たとえばポインタおよび／またはマッピングの再帰的システムによることを含む何らかの他のポインタ機構により一意に識別することができる小さな一連のタイムスタンプおよび／または仮想メモリアドレスについて考えてみる。たとえば、６４ｋの異なる一意の仮想メモリアドレスの順序付きリストを指し示す、大きな一連の２５０のポインタを指し示す１２のポインタのリストについて考えてみる、それにより、４ビットポインタは、０と２＊＊６４の間でＤＣにより事前に決定されるような任意の値を表すことができる一意の数を指し示す８ビットポインタを指し示して、６４ｋ×６４ｋの階乗を選ぶことが可能になる。これらおよび／または他の類似の方法を利用して、カーネルの不可欠な空間フットプリントをさらに低減することはすべて、本発明の方法を可能にする実施形態をさらに構成するのに役立つ。同様に、たとえば異なる、相互に関係のある複雑なマッピングを有利に利用するなどの、そのような他の機会は、共有メモリシステム、および／もしくは異なるタイプの連想型または非連想型のオン・チップ・キャッシュを使用することに由来する異なる複雑な一意の組合せマッピングが継続的に機能する際に、ならびに／また

はビットパターンのそれぞれの位置により、ＴＤに一意に指定される６４ビットスタンプの比較的小さなリストの複数のかなり大きな３２ビットセグメントの間の差の何らかの関数、または比較的短い順序付きリスト内部のそれぞれの位置の何らかの関数を介して、大きなビットパターンを一意に識別するために機能上使用することができる大きなビットパターンの数値、または何らかの数を指定する際に現れ、これらの操作は、単一ループまたはネスティングされたループの内部で繰り返し、および場合によっては再帰的に遂行することができ、これらの操作すべては、当業者に公知の、または容易に発見可能な数多くの他の類似のおよび／または似た方法と共に、本発明の構成要素を成す側面を表す新規な方法をさらに構成する、カーネルの空間フットプリントをさらに低下させる手段をさらに提供する。

【００８０】

すでに具体的に述べたように、そのような追加の方法は、それらの方法が本発明のビットに関して同一ビットパターン特徴を示すという事実にもかかわらず、２つ以上のビットパターンに基づき区別する代替手段を利用する。すでに示したように、そのような追加の方法は、より大きな圧縮性とより効率的な標的設定の両方を提供し、上記で記述したうに、一連のｊがｋに対してより効率的に収束することを可能にする際にも、同様に本明細書の誤差項を効果的に表す際にも、これらの利点の両方を提供することができる。そしてこれらの方法は、すでに示したように、最も重要には、以下で直接定義するように、「ＤＩＤ」および「Ｃ」として有用にカテゴリ分類することができる２つの非常に異なるタイプのビットパターンを操作するとき、これらの方法の改善形態を提供し、この場合、「ＤＩＤ」は、かなり大きな不連続を包含するという代価を払って、非常に長い、多くの場合任意に大きなビットパターンの非常に大きな数（上記で記述した、本明細書の高圧縮の生成関数から出力される非常に大きな数など）を表し、その一方で、「Ｃ」は、その正常操作の過程で、すでに示した方法でコンピュータが数多くの任意のビットパターンを作り出し、一方では、たとえばタイムスタンプおよび仮想メモリアドレスを、不連続がまったくない任意の６４ビット値に「プリセット」することができ、これらの区別の重大性およびそれぞれ提供される別個の効率性能が当業者に明らかであるという事実により示されるように、かなり大きな範囲の内部に不連続がまったくない、どんな任意の値の、またはどんな任意の値のどんな任意の関数もとる、これらの不連続の能力に基づき、これらの不連続の影響をより効率的に取り扱うために有用な「充填文字」ビットパターンを表す。

【００８１】

他のメタデータの中でも、たとえばとりわけ値をもつメモリアドレスおよびタイムスタンプを、どの特有のデータ項目および操作に関連づける、または割り当てるかを決定する能力は、（本発明が可能にする、オン・チップ・キャッシュ内にすでに全体またはほぼ全体が常駐するカーネルからビットパターンを迅速に再計算するのとは対照的に）現代のコンピュータシステムで大きなビットパターンを移動させる、操作する、および利用することが、オペレーティングシステムの何らかの構成要素、および／もしくは他のシステム・ソフトウェア、またはいくつかの処理では、異なるデータがどの仮想メモリアドレスに最も非効率に記憶されるか、ならびにそのデータに対してどの時間に、およびどの時間シーケンスにデータ処理操作が最もよく遂行されるかを含む、コンピュータ操作のそのような側面をすべて最もよく最適化するように特に設計された、さらにまた下流にある、ハードウェアにより近く、よりアクセスできない他の何らかのソフトウェアにより典型的には遂行される、複雑で時間のかかるルーチンの存在を余儀なくさせる、十分に骨の折れる、非常に非効率な処理であるという現実により、以前は禁止されていた。これらの制約から解放されているのが、以下の主要な利点の少なくともすべてを提供する方法で、ＴＤが遂行するデータ再計算の一部として後で利用される特に選ばれた関連する値を伴う特有のビットパターンに仮想メモリアドレス、タイムスタンプ、およびシステムに既知の他のメタデータパラメータのようなものを割り当てることができるＤＣの能力である。

【００８２】

ａ）上記の例を含む、すでに説明した、ＴＤが、かなりより小さなフットプリントを用

いて著しくより迅速に自身の計算を遂行することを可能にする、当業者に自明の明白な圧縮性および標的設定の利点。ＤＣが一方のコンピュータ、たとえばサーバ上に常駐し、かつＴＤが別のコンピュータ、たとえばクライアント上に常駐する実装形態では、特に選ばれたネットワーク伝送の側面、たとえば、周波数、信号強度、伝送速度の変化などを利用して、そのような情報を符号化することにより当業者が知ることができる手法で、データ伝送処理の間、その類似する利点を提供することができる。

【００８３】

ｂ）本発明のこの側面は、ビットパターンの最終出力値が、仮想メモリアドレスおよび／またはタイムスタンプおよび／またはそれら関数などの因子に一部は依存することによる限り、ＤＣに既知の決定的な手法で、リアルタイム更新（従来のデータ圧縮には存在しない性能）を実装する能力を提供し、ＴＤに、新しく変更されたビットパターンを出力させる手法で、関連する仮想メモリアドレスおよび／またはタイムスタンプおよび／または他のメタデータリポジトリおよび／またはそれらの関数の値を変更することにより、実行中に大きなビットパターンの一定のセグメントに変更を迅速に加えることを達成することができる。この方法は、逆方向で（すなわち、記憶場所を改変して、データ値を変更することを介して）記憶と処理の間の基礎となる機能上の双方向等価性を利用するが、その一方で、たとえば仮想メモリアドレス割当てを変更してデータ値を更新する場合、ＤＣがタグ情報を改変することを介して、適切な変更を適応させているとき、どんなビットパターンも実際に、物理的に移動させる必要がまったくないことに留意されたい。

【００８４】

たとえば仮想メモリアドレス、タイムスタンプ、または他のメタデータを表す６４ビットエンティティのそのような変更は、インデックスする目的で利用することができ、任意の指定した地点から始まる、またはそこで止まるビットパターン全体の一部にアクセスする能力を提供する。そのような変更は、仮想メモリアドレスまたはタイムスタンプまたは何らかの他のメタデータリポジトリなどの、たとえば６４ビットエンティティでビットパターンのサブセットを利用して達成することができる、ならびに／または２つ以上の独立したインデックス基準を適合させる手法で、たとえばそのようにインデックス付けを遂行する際に有用である可能性がある、仮想メモリアドレスまたはタイムスタンプまたは何らかの他のメタデータリポジトリなどの何らかの組合せで、および／もしく位置で表すことができることに留意されたい。たとえば、そのような変更は、４時間半の長さの映画で、１秒ごとに区別するために１４ビット（任意に選ばれた任意の開始地点および止まる地点を指定するために使用することができる）を必要とすると考えてみる。したがって、この例のように、対応してサイズ設定された小数部に対応する開始地点を探索しようとする場合（小数部のサブセクションの各々の微分すべてがわかっているというＤＣとＴＤの両方の利点、および同様にランダム化という別の可能な利点に留意されたい）、開始から終了まで順次オン・チップ・キャッシュを横断するのではなく、キャッシュラインの仮想メモリアドレス内部の適切な関連する１４ビットセグメントを包含する仮想メモリアドレスに記憶されたキャッシュセグメントから始まるキャッシュ位置に直接に進む（検索のこれらのソートを促進するように設計された関連するキャッシングの枠組みを選ぶことにより，容易に活用することができるタスク）。その上、代わりに、微分すべて、および蓄積速度の平滑さの程度がわかっていることに基づき、そのようなインデックスなしに適切な開始地点を位置的に計算することができ、代わりに、キャッシュライン内部の何らかの特有の位置で特有の副場所を位置的に探索する能力を適合させるハイブリッド機構を利用することができる。

【００８５】

これらの目的のために使用されないとき、その位置情報を、情報をより効率的に符号化することを含む他の目的のために使用することができる。たとえば、固定サイズの、または少なくとも既知のサイズの構成単位を可換操作により処理している条件の下で、その計算の最終結果は、順序不変量になり、ＤＣが何らかの順序づけ原理（たとえば、そのバイナリ・ビット・パターン内部の適切な長さのそのサブセクションの関連する数値）に従っ

てユニットを置く順序を許可し、次いで、ｎ！の可能な置換順序づけのうちどれが実際に選ばれたかに基づき、他の情報を符号化する。

【００８６】

さらに、上記の定義は、ＤＣの中に入力された各ビットパターンが１つまたは複数のＨＣＳＳＦＨＩＬにより表され、かつ各ＨＣＳＳＦＨＩＬが２つ以上のＬＯＯＰにより表される場合について具体的に述べているが、その一方で、単一ループから入力ビットパターン全体を再計算することができる場合もまた、本発明の実施形態と考えられるべきであることに留意されたい。

【００８７】

図３（ａ）、図３（ｂ）、および図３（ｃ）は、ＤＣおよびＴＤの複数の可能な実装形態を示す。図３（ａ）は、ＤＣが（ロード時に）処理操作を確立した単一コンピュータ上のオン・チップ・キャッシュ内に主に、または全体として存在するＤＣおよびそれに対応するＴＤ、ならびに当初、ＤＣの中に入力されたビットパターンを迅速に、無損失に再現するためにＴＤが操作するカーネルを形成するデータパラメータを示す。図３（ｂ）は、一方の機械（たとえば、サーバ）上に常住するＤＣを用いて達成される同じ機能性を示し、一方では、カーネルを包含するＴＤは、まったく別の機械（たとえば、クライアント）のオン・チップ・キャッシュ内に主に、または全体として常駐する。図３（ｃ）は、ＤＣに入力を提供するビットパターンが、サーバと異なる機械から、クライアントであってもなくてもよい別の機械から発生するという点で図３（ｂ）と区別される。

【００８８】

図４は、ロード時にＤＣによっても、実行時にＴＤによっても遂行される操作のＮの、異なるが関係があるセットを例示するテンプレートをさらにまた提供する、すべてのＮに関する任意のサイズのＮビットパターンの組合せ複雑性の視覚的表現を両方とも提供するベクトルを示す。ベクトル自体は、サイズＮのビットパターンを表すことができるあらゆる可能なビット内容の概念上の表現を包含し、たとえば、ベクトルは、各要素の関連する数値が、その直前の先行するビットパターンの関連する数値＋１に等しい、すべて０から始まり、すべて１で終わる２＊＊８Ｎの要素を包含する。ｋとして指定されたこのベクトルに対応するラインセグメント上の何らかの場所で表された１つのそのようなビットパターンは、ロード時にＤＣに入力されており、同様に実行時にＴＤが出力する特定のビットパターンに正確に対応する特定のビットパターンを表す。ＤＣもＴＤも、典型的には、特定のＪａによりそれぞれ指定される、ビットパターンｋの数値を次第に近似するように設計された、一連の他の別個のサイズＮのビットパターンを作り出す。ＤＣが作り出す連続するＪ値は、典型的には、反復して改変され、その一方で、ＤＣが作り出すＪａの最終セットは、ＴＤが利用するＪａに対応する。

【００８９】

図５は、ＨＣＳＳＦＨＩＬを例示し、ＨＣＳＳＦＨＩＬの記述名により表現されるように、出力ビットパターンをＨＣＳＳＦＨＩＬとみなす特性を包含する出力ビットパターンを作り出す生成関数をそれぞれ表す２つ以上のＬＯＯＰからＨＣＳＳＦＨＩＬが構成されることを示す。これらのループは、加算または任意の他の関数を表すことができる１つまたは複数の異なる操作により接続される。

【００９０】

図６は、ＨＣＳＳＦＨＩＬを順次反復するステップが、ＤＣによりどのように作り出されるかを例示し、各ＨＣＳＳＦＨＩＬは、特定のＪ値を出力し、次いで、特定のＪ値は、次のＨＣＳＳＦＨＩＬを作り出すために変えられ、次のＨＣＳＳＦＨＩＬは、次のＪ値を出力する。

【００９１】

図７は、仮想メモリアドレス、タイムスタンプ、および／もしくは他のメタデータに包含されるビットパターンから、ならびに／または仮想メモリアドレス、タイムスタンプ、および／もしくは他のメタデータに包含されるビットパターンの関数から、これらのパラメータの１つまたは複数自体を再計算する方法を介して、空間フットプリントを低下させ

る、および／またはＴＤが実際に記憶し、利用するパラメータの計算能力を高めるために、メタデータ項目の誤差項の表現を多くの場合に含むメタデータ項目のパラメータの一部またはすべてを変換するために、仮想メモリアドレス、および／またはタイムスタンプを含むがそれらに限定されないメタデータ項目をＤＣがどのように利用するかを例示する。

【００９２】

図８は、ＴＤが、類似する動作を実行時にどのように実装するかを例示する。したがって、図８は、図８の出力が、図７の入力に正確に対応し、かつ図６の入力が、図７の出力に正確に対応するように、図７に例示する操作の逆を表す。

【００９３】

図９は、本明細書で記述する反復ステップを表し、入力ビットパターンｋから始めて、ＤＣがＴＤに引き渡すカーネルの形をとるＤＣの出力を作り出す一連の反復ステップをＤＣがどのように実行するかを段階的に示す。ステップ１およびステップ２は、たとえば、ＤＣの初期ｊ値を計算する最初の反復を最適化するためにＤＣが採用してもしなくてもよい従来の圧縮方法および手順を適用することによって、ビットパターンｋを初期にランダム化するステップを表す。次いで、ＤＣは、図９に示すさまざまなステップを通して反復して、当初のビットパターンｋをより近く近似する目的である一連の追加のｊ値を提供する。ＤＣは、各段階で、ＤＣの内部試験基準に従って、一連の追加のｊ値が十分に近いかどうかを試験し、その後、さらになおカーネルのフットプリントを下げることによることを含み、ＴＤが実装するカーネルを形成する操作および／またはデータの効率をさらに改善するために、図７に例示するステップをさらに実装してもしなくてもよい。

【図１】

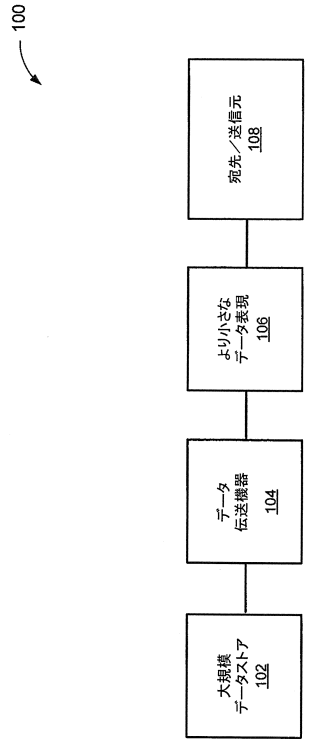


FIG. 1

【図２】

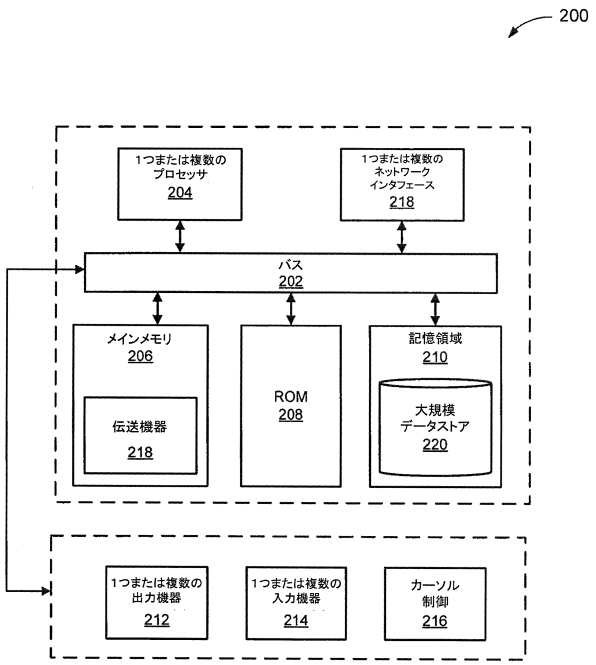


FIG. 2

【図３（ａ）－（ｃ）】

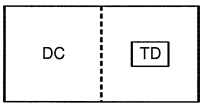


FIG. 3(a)

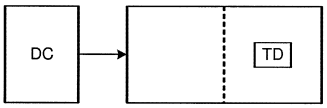


FIG. 3(b)

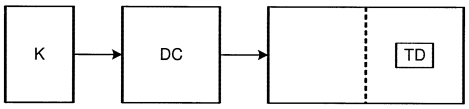


FIG. 3(c)

【図４】

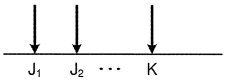


FIG. 4

【図５】

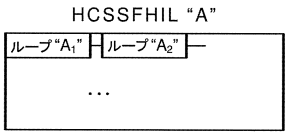


FIG. 5

【図６】

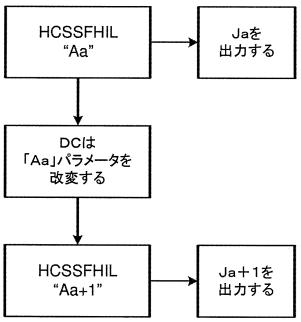


FIG. 6

【図７】

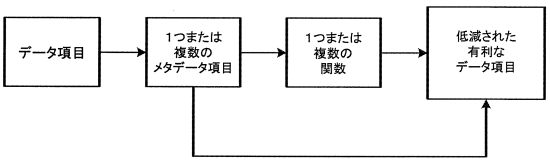


FIG. 7

【図８】

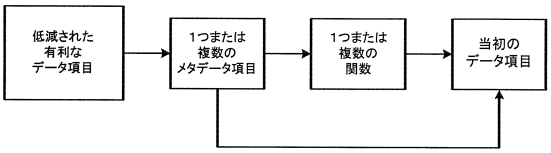


FIG. 8

【図９】

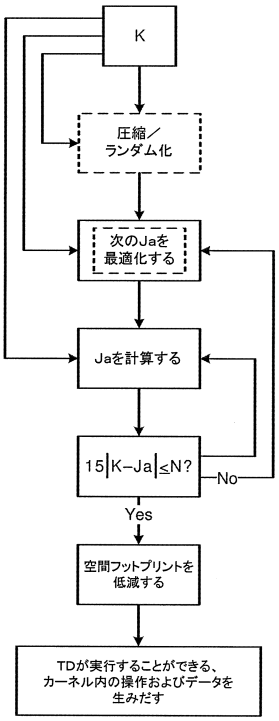

K
圧縮／
ランダム化
次のJaを
最適化する
Jaを計算する
15|K−Ja|≤N?
No
Yes
空間フットプリントを
低減する
TDが実行することができる、
カーネル内の操作およびデータを
生みだす

FIG. 9

フロントページの続き

早期審査対象出願

(56)参考文献　特開２００５－０５１４７９（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１７／１０
Ｈ０３Ｍ　７／４０