

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号  
特許第7529673号  
(P7529673)

(45)発行日 令和6年8月6日(2024.8.6)

(24)登録日 令和6年7月29日(2024.7.29)

(51)国際特許分類 F I  
G 0 6 F 16/22 (2019.01) G 0 6 F 16/22

請求項の数 17 (全14頁)

|                   |                                  |          |                                                                |
|-------------------|----------------------------------|----------|----------------------------------------------------------------|
| (21)出願番号          | 特願2021-540318(P2021-540318)      | (73)特許権者 | 521304427                                                      |
| (86)(22)出願日       | 令和2年1月8日(2020.1.8)               |          | ログノベーションズ ホールディングス,<br>エルエルシー                                  |
| (65)公表番号          | 特表2022-518194(P2022-518194<br>A) |          | LOGNOVATIONS HOLDI<br>NGS, LLC                                 |
| (43)公表日           | 令和4年3月14日(2022.3.14)             |          | アメリカ合衆国 3 3 6 0 9 フロリダ州<br>タンパ, ダブリュ・パークランド プー<br>ルパード 3 2 3 4 |
| (86)国際出願番号        | PCT/US2020/012661                | (73)特許権者 | 521304438                                                      |
| (87)国際公開番号        | WO2020/146448                    |          | マケルヴィーン, クリストファー                                               |
| (87)国際公開日         | 令和2年7月16日(2020.7.16)             |          | MCELV EEN, Christop<br>her                                     |
| 審査請求日             | 令和4年10月24日(2022.10.24)           |          | アメリカ合衆国 3 3 6 0 9 フロリダ州<br>タンパ, サウス スターリング アベニュー<br>- 8 1 0    |
| (31)優先権主張番号       | 16/244,332                       |          |                                                                |
| (32)優先日           | 平成31年1月10日(2019.1.10)            |          |                                                                |
| (33)優先権主張国・地域又は機関 | 米国(US)                           |          |                                                                |

最終頁に続く

(54)【発明の名称】 コンテンツ不可知ファイルインデキシングの方法及びシステム

## (57)【特許請求の範囲】

## 【請求項1】

計算機システム上で実行されるバイナリデータファイルのコンテンツ不可知参照の為の情報処理方法であって、

前記計算機システムを使用して、入力シードを用いて表を事前生成するステップであって、表は所定の長さのビットの全ての順列を含むステップと、

前記計算機システムを使用して、前記バイナリデータファイルの長さを決定するステップであって、前記長さは、前記バイナリデータファイルのビット数を含むステップと、

前記計算機システムを使用して、前記バイナリデータファイルを部分ストリングにチャンクするステップであって、各部分ストリングは前記バイナリデータファイルの長さよりも小さい長さであるステップと、

前記計算機システムを使用して、前記バイナリデータファイルの各チャンクについて、そのチャンクが前記事前生成された表内にあるかどうかを判断し、そのチャンクが事前生成された表内にある場合には、そのチャンクに前記事前生成された表内のチャンクの位置のインデックスを関連付け、そのチャンクが前記事前生成された表内にはない場合には、チャンクされたバイナリデータを更に小さなチャンクに分割するステップと、

前記計算機システムを使用して、チャンクの数及び全チャンクの関連するインデックスを使用して前記バイナリデータファイルを示すステップと、を含む方法。

## 【請求項2】

前記チャンクの数及び全チャンクの関連するインデックスを使用して前記バイナリデー

10

20

タファイルを示すステップが、

前記バイナリデータファイルの代わりに、前記チャンクの数及び全ての関連するインデックスを記憶装置に永続化するステップを含む、請求項 1 に記載の方法。

【請求項 3】

前記チャンクの数及び全チャンクの関連するインデックスを使用して前記バイナリデータファイルを示すステップが、

前記バイナリデータファイルの代わりにチャンクの数及び全チャンクの関連するインデックスを送信するステップを含む、請求項 1 に記載の方法。

【請求項 4】

前記送信するステップは、前記チャンクの数及び全チャンクの関連するインデックスをネットワーク上で送信する、請求項 3 に記載の方法。 10

【請求項 5】

前記送信するステップは、前記チャンクの数及び全チャンクの関連するインデックスをバス上で送信する、請求項 3 に記載の方法。

【請求項 6】

前記チャンクの数及び全チャンクの関連するインデックスを使用して前記バイナリデータファイルを示すステップは、

各順序付けられたペアがチャンクレベル及び関連するインデックスを示す、順序付けられたペアのタプルを作成するステップを含む、請求項 1 に記載の方法。

【請求項 7】 20

前記チャンクの数及び全チャンクの関連するインデックスを使用して前記バイナリデータファイルを示すステップは、前記チャンクの数及び全チャンクの関連するインデックスを記憶装置上に永続化することを含む、請求項 1 に記載の方法。

【請求項 8】

前記記憶装置はディスクである、請求項 7 に記載の方法。

【請求項 9】

前記事前生成された表がハッシュ表である、請求項 1 に記載の方法。

【請求項 10】

前記事前生成された表が行列である、請求項 1 に記載の方法。

【請求項 11】 30

前記事前生成された表が揮発性メモリに永続化される、請求項 1 に記載の方法。

【請求項 12】

前記事前生成された表が不揮発性メモリに永続化される、請求項 1 に記載の方法。

【請求項 13】

前記バイナリデータファイルを部分ストリングにチャンクすることは、更に、前記バイナリデータを所定の長さのチャンクにチャンクすることを含む、請求項 1 に記載の方法。

【請求項 14】

前記所定の長さが 2 メガバイトである、請求項 13 に記載の方法。

【請求項 15】 40

前記所定の長さが 2 メガバイトよりも小さい、請求項 13 に記載の方法。

【請求項 16】

前記所定の長さが 2 メガバイトよりも大きい、請求項 13 に記載の方法。

【請求項 17】

前記バイナリデータファイルを部分ストリングにチャンクすることは、更に、前記バイナリデータファイルを、同じサイズの 2 つのチャンクに再帰的に分割することを含む請求項 1 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】 50

( 関連出願の相互参照 )

本出願は、2017年10月11日に出願された、「コンテンツ不可知ファイルインデキシングの方法及びシステム ( Method and System for Content Agnostic File Indexing ) 」という名称の特許出願第15 / 730 , 043号の一部継続出願であり、同出願の内容はあらゆる目的の為に本明細書に参照により完全に援用される。

【 0 0 0 2 】

( コンピュータプログラムリスト - シーケンスリスト )

以下のコンピュータプログラムリストを本明細書に添付して提出し、参照により援用する。夫々のファイルは参照により援用される。以下のコンピュータプログラムリストは、以下の形式である。 < バイト単位のサイズ > < 作成日 > < ファイル名 >。

【 0 0 0 3 】

3864 May 16 2018 squeeze-master-README-md.txt\*

【 0 0 0 4 】

83675 May 16 2018 squeeze-master-SqueezeReport-ipynb.txt\*

【 0 0 0 5 】

4293 May 16 2018 squeeze-master-demo\_app-py.txt\*

【 0 0 0 6 】

98 May 16 2018 squeeze-master-gitignore.txt\*

【 0 0 0 7 】

1383 May 16 2018 squeeze-master-requirements.txt\*

【 0 0 0 8 】

2490 May 16 2018 squeeze-master-rpc\_server-py.txt\*

【 0 0 0 9 】

239 May 16 2018 squeeze-master-scripts-buildprotos.txt\*

【 0 0 1 0 】

942 May 16 2018 squeeze-master-scripts-file\_test-py.txt\*

【 0 0 1 1 】

1391 May 16 2018 squeeze-master-scripts-generate\_key-py.txt\*

【 0 0 1 2 】

711 May 16 2018 squeeze-master-scripts-generate\_keyset-py.txt\*

【 0 0 1 3 】

629 May 16 2018 squeeze-master-scripts-keys\_from\_folder-py.txt\*

【 0 0 1 4 】

377 May 16 2018 squeeze-master-scripts-lzw\_test-py.txt\*

【 0 0 1 5 】

107 May 16 2018 squeeze-master-scripts-runserver.txt\*

【 0 0 1 6 】

3928 May 16 2018 squeeze-master-scripts-squeeze-bytes-report-py.txt\*

【 0 0 1 7 】

63 May 16 2018 squeeze-master-scripts-squeeze\_file-py.txt\*

【 0 0 1 8 】

1060 May 16 2018 squeeze-master-scripts-squeeze\_test-py.txt\*

【 0 0 1 9 】

947 May 16 2018 squeeze-master-scripts-string\_test-py.txt\*

【 0 0 2 0 】

222 May 16 2018 squeeze-master-scripts-test\_binary-py.txt\*

【 0 0 2 1 】

1799 May 16 2018 squeeze-master-scripts-test\_rpc-py.txt\*

【 0 0 2 2 】

10

20

30

40

50

|                                                                                                                                                        |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2736 May 16 2018 squeeze-master-scripts-time-squeeze-string-py.txt*                                                                                    |    |
| 【 0 0 2 3 】                                                                                                                                            |    |
| 211 May 16 2018 squeeze-master-scripts-time_keygen-py.txt*                                                                                             |    |
| 【 0 0 2 4 】                                                                                                                                            |    |
| 65 May 16 2018 squeeze-master-scripts-unsqueeze_file-py.txt*                                                                                           |    |
| 【 0 0 2 5 】                                                                                                                                            |    |
| 80 May 16 2018 squeeze-master-setup-py.txt*                                                                                                            |    |
| 【 0 0 2 6 】                                                                                                                                            |    |
| 10657 May 16 2018 squeeze-master-squeeze- _ init _ -py.txt*                                                                                            |    |
| 【 0 0 2 7 】                                                                                                                                            | 10 |
| 2783 May 16 2018 squeeze-master-squeeze-bitstring-py.txt*                                                                                              |    |
| 【 0 0 2 8 】                                                                                                                                            |    |
| 9191 May 16 2018 squeeze-master-squeeze-keys-py.txt*                                                                                                   |    |
| 【 0 0 2 9 】                                                                                                                                            |    |
| 613 May 16 2018 squeeze-master-squeeze-performance-csv.txt*                                                                                            |    |
| 【 0 0 3 0 】                                                                                                                                            |    |
| 22445 May 16 2018 squeeze-master-squeeze-squeeze_pb2-py.txt*                                                                                           |    |
| 【 0 0 3 1 】                                                                                                                                            |    |
| 2232 May 16 2018 squeeze-master-squeeze-squeeze_pb2_grpc-py.txt*                                                                                       |    |
| 【 0 0 3 2 】                                                                                                                                            | 20 |
| 3366 May 16 2018 squeeze-master-squeeze-proto.txt*                                                                                                     |    |
| 【 0 0 3 3 】                                                                                                                                            |    |
| 875 May 16 2018 squeeze-master-templates-layout-html.txt*                                                                                              |    |
| 【 0 0 3 4 】                                                                                                                                            |    |
| 816 May 16 2018 squeeze-master-templates-upload_form-html.txt*                                                                                         |    |
| 【 0 0 3 5 】                                                                                                                                            |    |
| 1513 May 16 2018 squeeze-master-templates-uploaded_file-html.txt*                                                                                      |    |
| 【 0 0 3 6 】                                                                                                                                            |    |
| 200 May 16 2018 squeezerpc-master-Makefile.txt*                                                                                                        |    |
| 【 0 0 3 7 】                                                                                                                                            | 30 |
| 1131 May 16 2018 squeezerpc-master-README-md.txt*                                                                                                      |    |
| 【 0 0 3 8 】                                                                                                                                            |    |
| 7 May 16 2018 squeezerpc-master-gitignore.txt*                                                                                                         |    |
| 【 0 0 3 9 】                                                                                                                                            |    |
| 8995 May 16 2018 squeezerpc-master-main-go.txt*                                                                                                        |    |
| 【 0 0 4 0 】                                                                                                                                            |    |
| 21292 May 16 2018 squeezerpc-master-squeeze-squeeze-pb-go.txt*                                                                                         |    |
| 【 0 0 4 1 】                                                                                                                                            |    |
| 3366 May 16 2018 squeezerpc-master-squeeze-proto.txt*                                                                                                  |    |
| 【 0 0 4 2 】                                                                                                                                            | 40 |
| 本開示は、コンテンツ不可知ファイル参照の為の方法に関するものである。本方法は、更に、コンテンツ不可知データ圧縮の為の方法に関するものである。                                                                                 |    |
| 【背景技術】                                                                                                                                                 |    |
| 【 0 0 4 3 】                                                                                                                                            |    |
| ファイル参照技術は、一般に、ファイル参照システムにおいてデータを効率的にインデキシングする為に、保存されているデータの種類に関する知識を必要とする。同様に、問題となっているデータに関する知識は、一般的に、送信、保存等の為にデータサイズを縮小する為の改良された圧縮アプローチを作成する際にも使用される。 |    |
| 【発明の概要】                                                                                                                                                |    |
| 【発明が解決しようとする課題】                                                                                                                                        | 50 |

## 【 0 0 4 4 】

業界では、保存及び/又は送信しなければならないデータ量を減らす為に、ファイル参照及びデータ圧縮技術を改善する必要性が存在する。

## 【課題を解決するための手段】

## 【 0 0 4 5 】

一実施形態によれば、本開示は、強化されたコンテンツ不可知ファイル参照システムを用いてコンピューティング技術を改善する方法を提供する。この方法は、コンピュータ自体の動作を改善するものである。

## 【 0 0 4 6 】

開示された方法は幾つかの重要な利点を有する。例えば、開示された方法は、任意のコンテンツタイプのファイル参照を可能にする。

10

## 【 0 0 4 7 】

開示された方法は、更に、データが永続化されるのとは対照的に、アクセス時に生成され得るので、永続化又は送信されなければならない情報又はデータの量を大幅に減少させることができる。

## 【 0 0 4 8 】

本開示の様々な実施形態は、これらの利点の何れも有していなくても、又はその一部或いは全てを有していてもよい。当業者には、本開示の他の技術的利点も容易に明らかになり得る。

## 【図面の簡単な説明】

20

## 【 0 0 4 9 】

本開示及びその利点をより完全に理解する為に、ここで、添付の図面と併せて以下の説明を参照する。

同じ参照符号は、図面の幾つかの図を通して同じ部品又はステップを参照する。

【図 1】本開示の一実施形態のステップを概説するフローチャートである。

【図 2】本開示の別の実施形態のステップを概説する別のフローチャートである。

【図 3】本開示の代替実施形態のステップを概説するフローチャートである。

## 【発明を実施するための形態】

## 【 0 0 5 0 】

本開示は、データのコンテンツ不可知インデキシングの為の方法に関するものである。本方法は、例えば、ファイル参照システムや圧縮システムとして等、コンピュータ特有の様々なニーズに使用され得る。

30

## 【 0 0 5 1 】

以下の開示では、例示としてバイナリデータの圧縮に関連して本発明を説明しているが、本教示は、「n - ary」データと呼ぶ方が相応しい、あらゆるタイプのデータで同様に機能する。例えば、本方法及びシステムは、量子ビット及びビットでも動作する。

## 【 0 0 5 2 】

本発明の一実施形態は、図 1 に描かれたフローチャートに記載されているような方法を含む。永続化又は送信されるバイナリデータ ( $n_i$ ) (例えば、データファイル) が分析され、ビット単位の長さ ( $l(n_i)$ ) が決定される。この情報を用いて、ステップ 106 で、本方法は、識別された長さのデータの全ての順列を計算する。例えば、入力データが次のような場合、

40

0 1

## 【 0 0 5 3 】

入力データは 2 ビットの長さとなる。ステップ 106 で、2 ビットの全ての順列が生成され、即ち次のようになる。

{ 0 0 } { 0 1 } { 1 0 } { 1 1 }

## 【 0 0 5 4 】

ステップ 108 で、本方法は、生成された順列における入力バイナリデータファイルのインデックス ( $n_f$ ) を決定する。上記の例では、返されるインデックス ( $n_f$ ) は「1

50

」となる。最後に、入力されたバイナリデータ（即ち「01」）を保存又は送信するのではなく、システムは代わりに長さ（2）とインデックス（1）を保存する。

【0055】

元の入力データをデコードする必要がある場合（例えば、元のバイナリデータをディスクから取得する要求や、ネットワークを介して送信されたデータの受信）、この方法は、入力として長さ（ $l(n_i)$ ）とインデックス（ $n_f$ ）のみを必要とする。上記の例では、長さ（2）とインデックス（1）が入力となる。図2に示すように、システムは、入力された長さの全ての順列を計算する。上記の例では、次のような順列が生成される。

{00} {01} {10} {11}

【0056】

システムは指定されたインデックス（上記の例では1）に移動し、順列を返す。再び上記の例を用いて、元のバイナリデータの「01」が返される。

【0057】

上記の方法は、例示の目的で、バイナリシステム（即ち、入力データがバイナリデータである）の観点から説明されている。本方法及びシステムは、 $n$ -aryシステムについても同様に動作する。上記のバイナリシステムは基本的にユークリッド平面で動作するが、 $n$ -aryデータではヒルベルト空間が概念的に同じ利点を提供する。この方法とプロセスは、以下のように $n$ -aryデータに対して一般化され得る。

$d^n = p(i)$

$(d^n)^n = p(f)$

$d$  = システムの次数

$n$  = システムの次数に応じた適切な  $n$ -ary 単位での長さ

$p(i)$  = 初期インデックス

$p(f)$  = 最終インデックス

【表1】

| システムの次数 (d) | 視覚表示  | 参照キー          | サーチパターン           |
|-------------|-------|---------------|-------------------|
| 1           | ストリング | $n/x$         | 左から右              |
| 2           | 平面    | $n/x/y$       | 左上から右下            |
| 3           | 3(倍)  | $n/x/y/z$     | 上後左から下前右          |
| D           | D(倍)  | $n/x/y/z/...$ | 上後左... 下前右<br>... |

【0058】

同じ入力ファイルで2つの代替的な順序付けされたシステムが与えられた場合、より高い順序を持つシステムが、より低い順序を持つ代替的なシステムと比較して、より高い  $n$ -ary 密度を有することに留意すべきである。

【0059】

本方法の一例は、以下のRubyコードスニペットに開示されている。以下のスニペットは、図1に開示されているような方法を示している。

10

20

30

40

50

## 【表 2】

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| <pre> class Input   require 'securerandom'    def create(k)     input_binary = SecureRandom.hex(k)   end    def clean(k)     input_string = create(k).unpack('B*').first.to_s   end    def build(n)     permutation = (0..2**n-1).map {  i  "%0#{n}b" % i }   end    def self.kmp_search(string, substring)     return nil if string.nil? or substring.nil?      pos = 2     cnd = 0     failure_table = [-1, 0]     while pos &lt; substring.length       if substring[pos - 1] == substring[cnd]         failure_table[pos] = cnd + 1         pos += 1         cnd += 1       elsif cnd &gt; 0         cnd = failure_table[cnd]       else         failure_table[pos] = 0         pos += 1       end     end      m = i = 0     while m + i &lt; string.length       if substring[i] == string[m + i]         i += 1         return m if i == substring.length       else         m = m + i - failure_table[i]         i = failure_table[i] if i &gt; 0       end     end     return nil   end    def kmp_search(substring)     Input.kmp_search(self, substring)   end end  init = Input.new  input = init.clean(1) depth = input.length generate = init.build(depth) steps = generate.join.to_s  step = Input.kmp_search("#{steps}", "#{input}")  p input p depth p step </pre> | <p>10</p> <p>20</p> <p>30</p> <p>40</p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|

## 【0060】

以下のスニペットは、入力長 ( $l(n_i)$ ) が 16、インデックス ( $n_f$ ) が 72, 629 の場合の、図 2 に開示されている方法を示している。

## 【表 3】

```

class Output

  def build(n)
    permutation = (0..2**n-1).map { |i| "%0#{n}b" % i }
  end
end

depth      = 16
step       = 72629

init       = Output.new
create     = init.build(depth)
interpret  = create.join.to_s
compute   = (depth + step) - 1

output     = interpret[step..compute].gsub(/\s+w+$/, '!...!')

p output

```

10

20

## 【0061】

好ましい実施形態では、入力バイトストリングは、入力バイトストリングの表現に対応するビットストリングに変換される。このビットストリングは、次に本明細書に記載された方法によって処理されるものである。

## 【0062】

代替的な実施形態では、データの長さに基づいて表を生成するのではなく、特定の長さのデータの全ての順列で表を事前生成してもよい。この事前生成された表は、不揮発性メモリ又は揮発性メモリの何れかのメモリに永続化されてもよい。上記の例では、所定の長さが2ビットの場合、事前生成された表には、次のような2ビットデータの全ての順列が含まれる。

30

```
{ 0 0 } { 0 1 } { 1 0 } { 1 1 }
```

## 【0063】

一実施形態では、この表は、以下のように対応するインデックスを付けた配列で格納されてもよい。

## 【表 4】

|      |      |      |      |
|------|------|------|------|
| {00} | {01} | {10} | {11} |
| 0    | 1    | 2    | 3    |

40

## 【0064】

この事前生成された表は、ディスクやRAM等に保存されてもよい。好ましくは、この事前生成された表は、ファイルサイズを縮小する（又はファイルをスクイーズする）計算機システムと、縮小されたファイルを拡張する（又はデータをスクイーズ解除する）計算機システムとで保存される。

## 【0065】

50

入力データを受け取ると、本方法はデータをより小さなサブセットに「チャンク」する。本明細書では、「チャンク」とは、データストリングを取得して大きなデータストリングのサブセットを構成する小さなデータストリングを作成することを意味する。全てのチャンクを合わせると元のデータストリングになる。例えば、入力データが

0 1 1 0 0 1 1 1 0 0 0 1

であれば、

【0 0 6 6】

以下の4ビットチャンクにチャンクされることになる。

0 0 1 0 0 1 1 1 0 0 0 1

【0 0 6 7】

次に、個々のチャンクは、事前生成された表と比較され、一致するものがあるかどうかを確認する。上記の例では、チャンクのサイズが4ビットの場合、表には全2ビットチャンクの順列を有する為、各チャンクは表に見つからないことになる。従って、各チャンクは再度チャンクされ、以下のようになる。

0 1 1 0 0 1 1 1 0 0 0 1

【0 0 6 8】

この方法は、特定のチャンクが事前生成された表の中に配置される時点まで、各チャンクに対して続けられる。その時点で、チャンクは夫々のインデックスと関連付けられ、好ましくは、チャンクレベル及び対応するインデックスを示す一連のタプルが生成される。上記の例では、システムが2回チャンクしたので、インデックスの関連付けは以下のようになる。

{ 2 , 1 } { 2 , 2 } { 2 , 1 } { 2 , 3 } { 2 , 0 } { 2 , 1 }

【0 0 6 9】

この例では、元の入力データ「0 1 1 0 0 1 1 1 0 0 0 1」は、最終的に2ビット長の6つのチャンクに分割された。図示のように、各チャンクはチャンクレベル(2)と、事前生成された表への対応するインデックスで表されている。

【0 0 7 0】

データは、任意の数の方法でチャンクされ得る。例えば、データは、上記の例のように、事前に決定されたサイズに基づいてチャンクされてもよい(ここで、事前に決定されたサイズは、例の目的の為に4ビットであった)。或いは、各データチャンクが事前生成された表の中で見つかるまで入力データを2つの別々のデータチャンクに再帰的にチャンクしてもよい。上記と同じ入力データを使用して、データを分割してチャンク化する方法では、次のような第1レベルのチャンクになる。

0 1 1 0 0 1 1 1 0 0 0 1

【0 0 7 1】

ここで、データセットが事前生成された表に見つからないので、再度チャンクされる。

0 1 1 0 0 1 1 1 0 0 0 1

【0 0 7 2】

ここでも、チャンクされたデータは事前生成された表では見つからないので、再度チャンクされなければならない。

0 0 1 0 0 1 1 1 0 0 0 1

【0 0 7 3】

注目すべきは、幾つかのセグメントが、事前生成された表のサイズよりも小さいデータにチャンクされていることである(即ち、セグメント「1」、「1」、「0」、「1」)。これらのセグメントは、事前生成された表と比較する為にパディングされることがある。一貫性が保たれていれば、数字はビッグエンディアン又はリトルエンディアンバイトオーダーで保存され得る。例えば、ビッグエンディアンバイトオーダーを使った場合、上記のチャンクデータは次のように表される。

0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0

【0 0 7 4】

10

20

30

40

50

その後、この方法は上記と同じように続く。

【0075】

データの全てのチャンクが、同じデータチャンクレベルで事前生成された表の中で見つかることは必要ではない。例えば、2ビットの組み合わせに関する上記の事前生成された表を使用して、もし入力データが以下であれば、

0 1 1 0 0 1 1 1 0 0

【0076】

このデータは本来、上記のように4ビットのシーケンスに分割してチャンクされ得る。

0 1 1 0 0 1 1 1 0 0

【0077】

上記と同様に、最初の2つの4ビットシーケンス（即ち「0110」と「0111」）は、事前生成された表に配置される為に、より小さなチャンクに再度チャンクされなければならない、その結果、以下のチャンクになる。

0 1 1 0 0 1 1 1 0 0

【0078】

また、上記のように、チャンクは以下のように自らのチャンクレベル及び対応するインデックスに関連付けられる。

{ 2 , 1 } { 2 , 2 } { 2 , 1 } { 2 , 3 } { 1 , 0 }

【0079】

上記の最後のタプルは、そのチャンクが2回目のチャンキングを必要としなかったため、チャンクレベルが1であることを示していることに留意されたい。

【0080】

入力データが一連のチャンクレベルとインデックスに還元されると、その一連のチャンクレベルとインデックスを使用して、元のデータを識別する。関連付けは、一連のタプルとして、個別のビットストリングとして、及びその他の方法で保存され得る。

【0081】

一連のチャンクレベルとインデックスに基づいてデータを再作成（又はスクイーズ解除）するには、プロセスは逆に動作する。この場合も、システムには同じ事前生成された表が必要である。チャンクレベルとインデックスの各タプルに対して、システムは事前生成された表を参照して、スクイーズされたチャンクをアンパックし、元のデータに戻す。

【0082】

この代替実施形態は、図3のフローチャートに示されている。まず、以下のような特定の長さのデータの全ての順列を含む事前生成された表がステップ302で作成される。上述したように、好ましくは、その表は何らかの方法で永続化される。次に、システムは、ステップ304で、スクイーズ対象入力データを受け取る。次にプロセスは、ステップ306及び308で、データの長さが「事前生成された表」に配置される長さになるまで、データをより小さなセグメントにチャンクする。上述したように、このプロセスは、入力データセットが何回チャンクされたかをシステムが知るできるように、チャンクレベルを維持する。その後、各チャンクはステップ310で事前生成された表に配置される。最後に、チャンク、そのチャンクレベル、及び事前生成された表内の夫々のインデックスが関連付けられ、その結果、ステップ312で、スクイーズされたデータが得られる。

【0083】

本開示を、特定の実施形態及び一般的に関連する方法の観点から説明してきたが、これらの実施形態及び方法の変更及び順列は、当業者には明らかであろう。従って、例示的な実施形態に関する上記の説明は、本開示を制約するものではない。本開示の精神及び範囲から逸脱することなく、他の変更、置換、及び改変も可能である。

〔付記1〕

バイナリデータファイルのコンテンツ不可知参照の為にコンピュータ実装方法であって、  
入力シードを用いて表を事前生成するステップであって、表は所定の長さのビットの全  
ての順列を含むステップと、

10

20

30

40

50

前記バイナリデータファイルの長さを決定するステップであって、前記長さは、前記バイナリデータファイルのビット数を含むステップと、

前記バイナリデータファイルを部分ストリングにチャンクするステップであって、各部分ストリングは前記バイナリデータファイルの長さよりも小さい長さであるステップと、

前記バイナリデータファイルの各チャンクについて、そのチャンクが前記事前生成された表内にあるかどうかを判断し、そのチャンクが事前生成された表内にある場合には、そのチャンクに前記事前生成された表内のチャンクの位置のインデックスを関連付け、そのチャンクが前記事前生成された表内にない場合には、チャンクされたバイナリデータを更に小さなチャンクに分割するステップと、

チャンクの数及び全チャンクの関連するインデックスを使用して前記バイナリデータファイルを示すステップと、を含む方法。

10

〔付記 2〕

前記チャンクの数及び全チャンクの関連するインデックスを使用して前記バイナリデータファイルを示すステップが、

前記バイナリデータファイルの代わりに、前記チャンクの数及び全ての関連するインデックスを記憶装置に永続化するステップを含む、付記 1 に記載の方法。

〔付記 3〕

前記チャンクの数及び全チャンクの関連するインデックスを使用して前記バイナリデータファイルを示すステップが、

前記データファイルの代わりに前記チャンクの数及び全チャンクの関連するインデックスを送信するステップを含む、付記 1 に記載の方法。

20

〔付記 4〕

前記送信するステップは、前記チャンクの数及び全チャンクの関連するインデックスをネットワーク上で送信する、付記 3 に記載の方法。

〔付記 5〕

前記送信するステップは、前記チャンク数及び全チャンクの関連するインデックスをバス上で送信する、付記 3 に記載の方法。

〔付記 6〕

前記チャンクの数及び全チャンクの関連するインデックスを使用して前記バイナリデータファイルを示すステップは、

30

各順序付けられたペアがチャンクレベル及び関連するインデックスを示す、順序付けられたペアのタプルを作成するステップを含む、付記 1 に記載の方法。

〔付記 7〕

前記チャンクの数及び全チャンクの関連するインデックスを使用して前記バイナリデータファイルを示すステップは、前記チャンクの数及び全チャンクの関連するインデックスを記憶装置上に永続化することを含む、付記 1 に記載の方法。

〔付記 8〕

前記記憶装置はディスクである、付記 7 に記載の方法。

〔付記 9〕

前記事前生成された表がハッシュ表である、付記 1 に記載の方法。

40

〔付記 10〕

前記事前生成された表が行列である、付記 1 に記載の方法。

〔付記 11〕

前記事前生成された表が揮発性メモリに永続化される、付記 1 に記載の方法。

〔付記 12〕

前記事前生成された表が不揮発性メモリに永続化される、付記 1 に記載の方法。

〔付記 13〕

前記バイナリデータファイルを部分ストリングにチャンクすることは、更に、

前記バイナリデータを所定の長さのチャンクにチャンクすることを含む、付記 1 に記載の方法。

50

〔付記 1 4〕

前記所定の長さが 2 メガバイトである、付記 1 3 に記載の方法。

〔付記 1 5〕

前記所定の長さが 2 メガバイトよりも小さい、付記 1 3 に記載の方法。

〔付記 1 6〕

前記所定の長さが 2 メガバイトよりも大きい、付記 1 3 に記載の方法。

〔付記 1 7〕

前記バイナリデータファイルを部分ストリングにチャンクすることは、更に、前記バイナリデータファイルを、同じサイズの 2 つのチャンクに再帰的に分割することを含む付記 1 に記載の方法。

〔付記 1 8〕

チャンクの数及び全チャンクの関連するインデックスに基づいてデータを取得する方法であって、

入力シードを使用して表を事前生成するステップであって、前記表は、所定の長さのビットの全ての順列を含み、前記事前生成された表を使用してチャンクの数及び関連するインデックスを生成するステップと、

各チャンクについて、そのチャンクに関連付けられたインデックスで表内にデータを配置するステップと、各チャンクに関連付けられたデータを返すステップと、を含む方法。

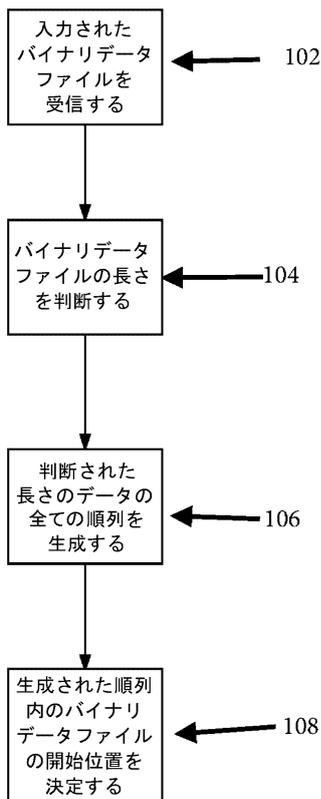
〔付記 1 9〕

各チャンクに関連するデータを返すステップは、

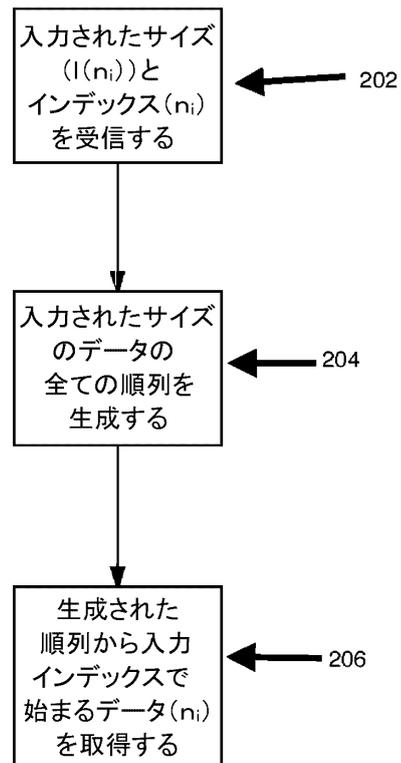
各チャンクに関連するデータを単一のビットストリームに連結することを含む、付記 1 8 に記載の方法。

【図面】

【図 1】



【図 2】



10

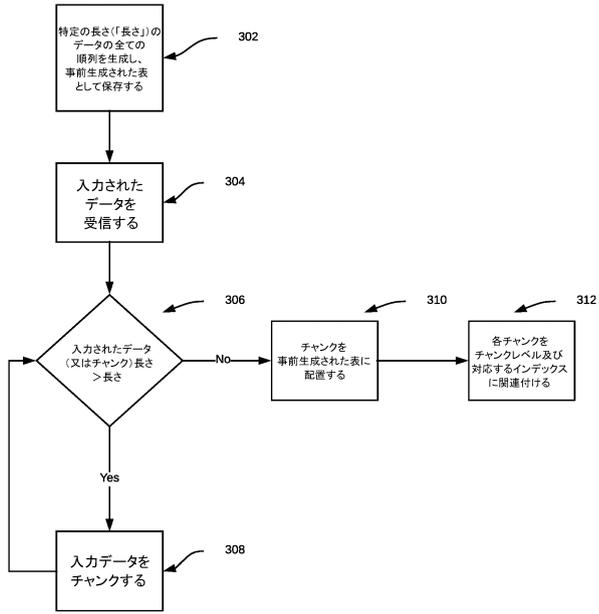
20

30

40

50

【図 3】



10

20

30

40

50

---

フロントページの続き

- (74)代理人 100107364  
弁理士 斉藤 達也
- (72)発明者 マケルヴィーン, クリストファー  
アメリカ合衆国 33609 フロリダ州 タンパ, サウス スターリング アベニュー 810
- 審査官 三橋 竜太郎
- (56)参考文献 米国特許出願公開第2012/0166448(US, A1)  
特表2007-508753(JP, A)  
米国特許出願公開第2015/0201043(US, A1)
- (58)調査した分野 (Int.Cl., DB名)  
G06F 16/00 - 16/958