

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6258975号  
(P6258975)

(45) 発行日 平成30年1月10日(2018.1.10)

(24) 登録日 平成29年12月15日(2017.12.15)

(51) Int.Cl.

F 1

<b>G06F 17/30</b>	<b>(2006.01)</b>	G06F	17/30	4 1 5
<b>G06F 9/50</b>	<b>(2006.01)</b>	G06F	17/30	1 1 O C
<b>G06F 13/00</b>	<b>(2006.01)</b>	G06F	9/46	4 6 5 E
		G06F	13/00	5 2 O B

請求項の数 13 (全 19 頁)

(21) 出願番号 特願2015-556052 (P2015-556052)  
 (86) (22) 出願日 平成26年1月17日 (2014.1.17)  
 (65) 公表番号 特表2016-515228 (P2016-515228A)  
 (43) 公表日 平成28年5月26日 (2016.5.26)  
 (86) 國際出願番号 PCT/US2014/012120  
 (87) 國際公開番号 WO2014/120487  
 (87) 國際公開日 平成26年8月7日 (2014.8.7)  
 審査請求日 平成29年1月16日 (2017.1.16)  
 (31) 優先権主張番号 13/756,340  
 (32) 優先日 平成25年1月31日 (2013.1.31)  
 (33) 優先権主張国 米国(US)

早期審査対象出願

(73) 特許権者 508178054  
 フェイスブック、インク。  
 アメリカ合衆国 カリフォルニア 940  
 25, メンロー パーク, ウィロー ロー  
 ド 1601  
 (74) 代理人 100105957  
 弁理士 恩田 誠  
 (74) 代理人 100068755  
 弁理士 恩田 博宣  
 (74) 代理人 100142907  
 弁理士 本田 淳  
 (72) 発明者 ラッシュ、サミュエル  
 アメリカ合衆国 94025 カリフォル  
 ニア州 メンロー パーク ウィロー ロ  
 ード 1601

最終頁に続く

(54) 【発明の名称】低レイテンシデータアクセス用のデータストリーム分割

## (57) 【特許請求の範囲】

## 【請求項 1】

複数のフロントエンドサーバにおいて、リアルタイムのユーザ活動に基づくログデータを生成すること、

前記ログデータを少なくとも1つの集約サーバに送信すること、

前記ログデータを前記集約サーバにおいて集約すること、

前記少なくとも1つの集約サーバにおいて前記ログデータを複数のログデータストリームに分割すること、

前記少なくとも1つの集約サーバによって前記ログデータストリームを少なくとも1つのバックエンドサーバに並列に供給すること、

を備え、前記ログデータは複数のログデータエントリを含み、各ログデータエントリがアプリケーション識別及びカテゴリフィールドを含み、

前記ログデータの各エントリについて、前記集約サーバによって、前記複数のログデータストリームの合計数であるパケット合計数を法として、前記アプリケーション識別及び前記カテゴリフィールドのハッシュ関数によってパケット番号を計算すること、

前記集約サーバによって、前記パケット番号によって識別されたログデータストリームに前記ログデータのエントリを割り当てること、

を備える方法。

## 【請求項 2】

前記分割するステップは、

10

20

前記ログデータが複数のログデータストリームに均等に分配されるように前記ログデータをランダムに分割することを含む、請求項1に記載の方法。

**【請求項3】**

集約された前記ログデータを処理前に一時的に記憶するための集約クラスタ内の中間記憶エリアを表すデータステージングエリアにおいて前記ログデータをステージングすることを更に備える請求項1又は2に記載の方法。

**【請求項4】**

前記ログデータをどのように前記複数のログデータストリームに分割するかに関する命令を前記バックエンドサーバから受信することを更に備える請求項1～3のうちの何れか一項に記載の方法。 10

**【請求項5】**

前記カテゴリフィールドが、前記ログデータエントリの意図された宛先の高レベルの記述を含む、請求項1に記載の方法。

**【請求項6】**

前記アプリケーション識別が、前記ログデータエントリを処理するためのデータ消費アプリケーションを識別する、請求項1～5のうちの何れか一項に記載の方法。

**【請求項7】**

前記パケット合計数が、前記ログデータストリームを受信するために利用可能なバックエンドサーバの数及び各バックエンドサーバが扱うことができる接続部の数によって決定される、請求項1に記載の方法。 20

**【請求項8】**

前記バックエンドサーバが前記ログデータストリームを受信して処理する場合に、前記バックエンドサーバに等しく負荷がかけられる、請求項7に記載の方法。

**【請求項9】**

前記パケット合計数が、少なくとも1つのバックエンドサーバで動作するデータ消費アプリケーションによって指示される、請求項1に記載の方法。

**【請求項10】**

前記ログデータのエントリの接頭部を検査して、エントリが割り当てられる前記ログデータストリームを決定することを更に備える請求項1～9のうちの何れか一項に記載の方法。 30

**【請求項11】**

前記少なくとも1つの集約サーバのうちの1つによって、前記ログデータをデータウェアハウスに送信すること、

前記データウェアハウスが、処理したログデータに基づいてデータ照会に応答することができるよう、前記データウェアハウスにおいて前記ログデータを処理すること、  
を更に備える請求項1～10のうちの何れか一項に記載の方法。

**【請求項12】**

実行された際に請求項1～11のうちの何れか一項に記載の方法を行うように動作可能なソフトウェアを具現化する1つ又は複数のコンピュータ可読非一時的記憶媒体。

**【請求項13】**

1つ又は複数のプロセッサと、

前記プロセッサに結合され、前記プロセッサによって実行可能な命令を含むメモリと、  
を備え、前記プロセッサは、請求項1～11のうちの何れか一項に記載の方法を行うため  
の命令を実行する際に動作可能である、システム。

**【発明の詳細な説明】**

**【技術分野】**

**【0001】**

本発明は、概して、データ取り込み及び処理システムに関し、特に、データを多数のデータストリームに分割することが可能なデータ取り込み及び処理システムに関する。

**【背景技術】**

10

20

30

40

50

**【 0 0 0 2 】**

コンピュータ及びネットワーク技術の発達は、大量のデータ記憶を必要とするアプリケーションをもたらした。例えば、何千万ものユーザが、ウェブページを作成し、ソーシャルメディアウェブサイトに画像及びテキストをアップロードすることができる。その結果、ソーシャルメディアウェブサイトは、毎日大量のデータを蓄積し、従って、データを記憶し処理するために非常にスケーラブルなシステムを必要とする可能性がある。かかる大量のデータ記憶を容易にするための様々なツールが存在する。

**【 0 0 0 3 】**

これらのソーシャルメディアウェブサイトのフロントエンドクラスタは、ユーザ活動を監視し、且つソーシャルメディアユーザの活動に基づいてログデータを作成する。フロントエンドクラスタは、集中型記憶フィラー又はデータウェアハウスにログデータを送信する。集中型記憶フィラー又はデータウェアハウスは、受信したログデータを編成し、データ処理アプリケーションからの要求に応える。大量のログデータを収容するために、大規模データウェアハウスが、ログデータを記憶し、且つデータ処理アプリケーションからのデータ集約的な照会に応えるために、一般に用いられる。10

**【 0 0 0 4 】**

アプリケーションが、何千ものコンピュータ（ノードとも呼ばれる）のクラスタ及びペタバイトのデータと対話できるようにすることによって、大規模なデータ集約型分散アプリケーションを支援するフレームワークが存在する。例えば、ハドウープ（Hadoop）と呼ばれるフレームワークが、ハドウープクラスタにおけるデータノード（子ノードとも呼ばれる）間で大量のデータ量を分配するためにハドウープ分散ファイルシステム（HDFS : Hadoop Distributed File System）と呼ばれる、分配され且つスケーラブルでポータブルなファイルシステムを利用する。データノードの停電又はネットワーク障害（スイッチ故障を含む）の悪影響を低減するために、HDFSにおけるデータは、典型的には、異なるデータノードに複写される。20

**【 0 0 0 5 】**

オープンソースデータウェアハウスシステムであるハイブ（Hive）は、ハドウープクラスタ上を走るために開発されている。ハイブは、ハイブQLと呼ばれる、スクリプト照会言語（SQL : scripted query language）のような宣言型言語で表現されたデータ照会を支援する。次に、ハイブシステムは、ハイブQLで表現された照会を、有向非巡回グラフの数学的形式において、ハドウープクラスタ上で実行できるマップリデュースジョブにコンパイルする。ハイブQL言語は、基本型、アレイ及びマップなどのコレクション、並びに入れ子にした型合成を含むテーブルを支援する型システムを含む。更に、ハイブシステムは、データ探索及び照会の最適化に有用な、スキーム及び統計を含むハイブメタストアと呼ばれるシステムカタログを含む。30

**【 0 0 0 6 】**

ハドウープクラスタと結合されると、ハイブシステムは、ソーシャルネットワーキングシステム用の大量のデータを記憶し分析することができる。例えば、ハイブシステムは、ユーザがソーシャルネットワーキングシステムを追い続けるという話をランク付けするために、ユーザ間の接続程度を分析することができる。ハイブシステムは、アプリケーション開発者、ページ管理者、及び広告主が、開発及びビジネスの決定を行うのを助けるために、ソーシャルネットワーキングシステムのサービスがどのように利用されているかに対する洞察力を得るように活動ログを分析することができる。ハイブシステムは、ソーシャルネットワーキングシステムのユーザに示される広告を最適化するために、複雑なデータマイニングプログラムを実行することができる。ハイブシステム、更に、ソーシャルネットワーキングシステムのスパム及び乱用を識別するために、使用ログを分析することができる。40

**【 0 0 0 7 】**

ハイブシステムは、プログラミング能力のない人々がハイブクエリをオーサリングして実行するための、複雑なデータパイプラインをオーサリングし、デバッグし、スケジ50

ユーリングするための、且つハイブシステム、並びにMySQL及びオラクルなどの他の関係データベースに記憶されたデータに基づいて報告を作成するためのウェブベースのツールを含む。

【発明の概要】

【発明が解決しようとする課題】

【0008】

しかしながら、フロントエンドクラスタは、リアルタイムではなく周期的に、取り込んだログデータを集中型データウェアハウスに送信する。更に、データウェアハウスが、これらのログデータに対するデータ照会に応答できる前に、データウェアハウスが、受信したログデータを編成するには時間がかかる。従って、データウェアハウスにおけるログデータは、ログデータが取り込まれてから或る期間後に利用できるだけである。その期間は、1時間又は1日にさえなる可能性がある。データ処理及び消費アプリケーションは、ログデータにアクセスできるまでに著しい待ち時間（レイテンシ）がかかるだけである。

【0009】

更に、集中型データウェアハウスは、連続的にログデータを受信するために、フロントエンドサーバとの接続を維持する必要がある。現在のソーシャルネットワークにおいて、フロントエンドサーバの数は、数千又はそれを超える可能性がある。データウェアハウスは、接続を維持する著しい負担を担う。接続を維持するかかる負担は、データウェアハウスの全体的性能に影響する。

【課題を解決するための手段】

【0010】

本明細書で導入される技術は、低レイテンシでのログデータアクセス及び処理の利点を提供する。特に、本明細書で導入される技術は、大量のログデータをリアルタイムで生成し、且つログデータを集約クラスタに転送するフロントエンドクラスタを含む。集約クラスタは、異なるフロントエンドサーバ及びクラスタからの着信ログデータストリームを集約するように設計される。集約クラスタは、更に、データストリームが受信アプリケーションに並列に送信されるように、ログデータを複数のデータストリームに分割する。一実施形態において、集約クラスタは、ログデータが、分割データストリームに均等に分配されることを保証するように、ログデータをランダムに分割する。別の実施形態において、分割データストリームを受信するアプリケーションは、どのようにログデータを分割するかを決定する。

【0011】

本明細書で導入される技術によれば、データストリームを分割するための方法が提供される。方法は、リアルタイムのユーザ活動に基づくログデータを生成すること、ログデータを集約サーバに送信すること、ログデータを集約サーバにおいて集約すること、ログデータを複数のログデータストリームに分割すること、およびログデータストリームを少なくとも1つのバックエンドサーバに並列に供給することを含む。

【0012】

本明細書で導入される技術によれば、データストリームを分割するためのコンピュータ実行システムもまた提供される。コンピュータ実行システムは、複数のフロントエンドサーバ及び少なくとも1つの集約サーバを含む。フロントエンドサーバは、リアルタイムのユーザ活動に基づくログデータを生成するように構成される。集約サーバは、フロントエンドサーバの少なくとも幾つかから受信されたログデータを集約するように構成される。集約サーバは、ネットワークを介して、フロントエンドサーバの少なくとも幾つかと接続される。集約サーバは、ログデータをステージングするように構成されたデータステージングエリアを含む。集約サーバは、1つ又は複数のバックエンドサーバがログデータストリームを並列に検索できるように、ログデータを複数のログデータストリームに分割するように更に構成される。

【0013】

本明細書で導入される技術によれば、ログデータをステージングするための集約サーバ

10

20

30

40

50

もまた提供される。集約サーバは、プロセッサ、ネットワークインターフェース、データ記憶装置及びメモリを含む。ネットワークインターフェースはプロセッサに結合され、このネットワークインターフェースを介して、集約サーバは複数のフロントエンドサーバと通信することができる。フロントエンドサーバは、リアルタイムのユーザ活動に基づくログデータを生成する。データ記憶装置は、データステージングエリアを含む。メモリは、プロセッサによって実行された場合に以下を含む処理、すなわち、フロントエンドサーバから、フロントエンドサーバによって生成されたリアルタイムのユーザ活動に基づくログデータを受信すること、ログデータを集約すること、データステージングエリアにおいてログデータをステージングすること、1つ又は複数のバックエンドサーバがログデータストリームを並列に検索できるようにログデータを複数のログデータストリームに分割することを含む処理を集約サーバに実行させる命令を記憶する。

#### 【0014】

本発明の実施形態において、特に、ソーシャルネットワーキングシステム若しくはその一部の、第三者システムの、又はクライアントシステムの無許可の使用を防ぐために、ソーシャルネットワーキングシステムの、第三者システムの、又はクライアントシステムのユーザを認証するための方法が提供される。

#### 【0015】

本発明の実施形態において、方法は、  
複数のフロントエンドサーバにおいて、リアルタイムのユーザ活動に基づくログデータを生成すること、  
ログデータを集約サーバに送信すること、  
集約サーバにおいてログデータを集約すること、  
ログデータを複数のログデータストリームに分割すること、  
ログデータストリームを少なくとも1つのバックエンドサーバに並列に供給すること、  
を含む。

#### 【0016】

上記分割するステップは、  
ログデータが複数のログデータストリームに均等に分配されるようにログデータをランダムに分割することを含むことができる。

#### 【0017】

上記方法は、  
データステージングエリアにおいてログデータをステージングすることを更に含むことができる。

#### 【0018】

上記方法は、  
ログデータをどのように複数のログデータストリームに分割するかに関する命令を、  
バックエンドサーバから受信することを更に含むことができる。

#### 【0019】

ログデータは複数のログデータエントリを含むことができ、各ログデータエントリはアプリケーション識別及びカテゴリフィールドを含むことができる。

上記方法は、  
ログデータの各エントリ用に、パケットの合計数を法 (modulo) として、アプリケーション識別及びカテゴリフィールドのハッシュ関数によってパケット番号を計算することであって、パケットの合計数が複数のログデータストリームの合計数であることと、  
ログデータのそのエントリを、パケット番号によって識別されたログデータストリームに割り当てること、  
を更に含むことができる。

#### 【0020】

カテゴリフィールドは、ログデータエントリの意図された宛先の高レベルの記述を含むことができる。

10

20

30

40

50

アプリケーション識別は、ログデータエントリを処理するためのデータ消費アプリケーションを識別することができる。

【0021】

ログデータは、複数のログデータエントリを含むことができる。

上記方法は、

各ログデータエントリ用に、1からパケットの合計数までの整数をランダムに生成することであって、パケットの合計数が複数のログデータストリームの合計数であることと、

そのログデータエントリを、パケット番号によって識別されたログデータストリームに割り当てる、

を更に含むことができる。

10

【0022】

パケットの合計数は、ログデータストリームを受信するために利用可能なバックエンドサーバの数、及び各バックエンドサーバが扱うことができる接続部の数によって決定することができる。

【0023】

パケットの合計数は、少なくとも1つのバックエンドサーバ上を走るデータ消費アプリケーションによって指示することができる。

バックエンドサーバがログデータストリームを受信して処理する場合に、バックエンドサーバには等しく負荷をかけることができる。

【0024】

20

上記方法は、

ログデータのエントリの接頭部を検査して、エントリが割り当てられるログデータストリームを決定することを更に含むことができる。

【0025】

上記方法は、

ログデータをデータウェアハウスに送信すること、

データウェアハウスが、処理したログデータに基づいてデータ照会に応答することができるよう、データウェアハウスにおいてログデータを処理すること、  
を更に含むことができる。

【0026】

30

本発明の実施形態において、コンピュータ実行システムは、

リアルタイムのユーザ活動に基づいてログデータを生成するように構成された複数のフロントエンドサーバと、

フロントエンドサーバの少なくとも幾つかから受信されたログデータを集約するように構成された少なくとも1つの集約サーバであって、ネットワークを介してフロントエンドサーバの少なくとも幾つかと接続される集約サーバと、

を含み、

集約サーバは、ログデータをステージングするように構成されたデータステージングエリアを含み、集約サーバは、1つ又は複数のバックエンドサーバが、ログデータストリームを並列に検索できるように、ログデータを複数のログデータストリームに分割するように構成される。

40

【0027】

ログデータは複数のログデータエントリを含むことができ、各ログデータエントリはアプリケーション識別及びカテゴリフィールドを含むことができ、集約サーバは更に、

ログデータの各エントリ用に、パケットの合計数を法として、アプリケーション識別及びカテゴリフィールドのハッシュ関数によってパケット番号を計算し、ここで、パケットの合計数は複数のログデータストリームの合計数であり、

ログデータのそのエントリを、パケット番号によって識別されたログデータストリームに割り当てる、

ように構成することができる。

50

**【0028】**

ログデータは、複数のログデータエントリを含むことができ、集約サーバは更に、各ログデータエントリ用に、1からパケットの合計数までの整数をランダムに生成し、ここで、パケットの合計数は複数のログデータストリームの合計数であり、そのログデータエントリを、パケット番号によって識別されたログデータストリームに割り当てる、  
ように構成することができる。

**【0029】**

パケットの合計数は、1つ又は複数のバックエンドサーバ上を走るデータ消費アプリケーションデータによって指示することができ、パケットの合計数は、ログデータストリームを受信するために利用可能なバックエンドサーバの数、及び各バックエンドサーバが扱うことができる接続部の数によって決定することができる。10

**【0030】**

本発明の実施形態において、集約サーバは、  
プロセッサと、  
プロセッサに結合され、集約サーバが複数のフロントエンドサーバと通信することを可能にするネットワークインターフェースと、  
データステージングエリアを含むデータ記憶装置と、  
プロセッサによって実行されることで集約サーバが以下を含む処理を行うことを可能にする命令を格納するメモリと、  
を備え、当該処理が、20

フロントエンドサーバからログデータを受信することであって、フロントエンドサーバがリアルタイムのユーザ活動に基づいてログデータを生成することと、

ログデータを集約すること、  
データステージングエリアにおいてログデータをステージングすること、  
1つ又は複数のバックエンドサーバがログデータストリームを並列に検索できるよう  
に、ログデータを複数のログデータストリームに分割すること、  
を含む。

**【0031】**

ログデータは複数のログデータエントリを含むことができ、各ログデータエントリはアプリケーション識別及びカテゴリフィールドを含むことができ、上記処理は、  
ログデータの各エントリ用に、パケットの合計数を法として、アプリケーション識別及びカテゴリフィールドのハッシュ関数によってパケット番号を計算することであって、パケットの合計数が複数のログデータストリームの合計数であることと、

ログデータのそのエントリを、パケット番号によって識別されたログデータストリームに割り当てる事と、  
を更に含むことができる。

**【0032】**

ログデータは複数のログデータエントリを含むことができ、上記処理は、  
各ログデータエントリ用に、1からパケットの合計数までの整数をランダムに生成することであって、パケットの合計数が複数のログデータストリームの合計数であることと、  
そのログデータエントリを、パケット番号によって識別されたログデータストリームに割り当てる事と、  
を更に含むことができる。

**【0033】**

本発明の更なる実施形態において、1つ又は複数のコンピュータ可読非一時的記憶媒体が、本発明による方法又は上記の実施形態のいずれかを実施するように実行された場合に動作可能なソフトウェアを具現化する。

**【0034】**

本発明の更なる実施形態において、システムは、1つ又は複数のプロセッサと、プロセ50

ツサによって実行可能な命令を含む、プロセッサに結合されたメモリであって、プロセッサが、本発明による方法又は上記の実施形態のいずれかを実施する命令を実行する場合に動作可能であるメモリと、を含む。

#### 【0035】

本明細書で導入される他の様子は、添付の図面及び以下に続く詳細な説明から明らかになる。本発明の上記した目的や他の目的、機構及び特徴は、全てが本明細書の一部を形成する、添付の特許請求の範囲及び図面と共に以下の詳細な説明の検討から、当業者には一層明らかになる。

#### 【図面の簡単な説明】

#### 【0036】

10

【図1】オンラインデータ収集システムの例を示す。

【図2】ログデータストリームを分割できる集約クラスタの例を示す。

【図3】ログデータエントリの例示的な構造を示す。

【図4】集約サーバの多層を有するオンラインデータ収集システムの別の例を示す。

【図5】ログデータを集約しステージングするための例示的な処理を示す。

【図6】ログデータをステージングし分割するための例示的な処理を示す。

【図7】本明細書で説明される任意のノード又はサーバを表し得るコンピュータサーバのアーキテクチャの例を示す高レベルブロック図である。

#### 【発明を実施するための形態】

#### 【0037】

20

本明細書において、「一実施形態」等への言及は、特定の機構、機能、又は特徴が、本発明の少なくとも一実施形態に含まれることを意味する。この記載におけるかかる句の出現は、全てが必ずしも同じ実施形態を指すわけでもなく、それらが必ずしも相互に排他的であることでもない。

#### 【0038】

図1は、オンラインデータ収集システムの例を示す。オンラインデータ収集システム100は、複数のフロントエンドクラスタ110を含む。各フロントエンドクラスタ110は、多数の相互接続されたフロントエンドサーバ112を含む。一実施形態において、フロントエンドクラスタ110は、5000～30000のフロントエンドサーバ112を含むことができる。フロントエンドクラスタは、ウェブトラフィックを扱い、且つオンラインデータ収集システム100によって監視されたユーザ活動に基づいて、リアルタイムでログデータを作成する。一実施形態において、フロントエンドクラスタ110は、更に、HTTPサービスを含む、システム100のユーザにユーザインターフェースを提供する責任を負う。

30

#### 【0039】

オンラインデータ収集システム100は、ログデータを集約するための、即ち、効率的なネットワークデータ伝送用にログデータを収集及びバッファリングするための責任を負う複数の集約クラスタ120を更に含む。フロントエンドクラスタ110のそれぞれは、集約クラスタ120が利用可能であるかどうかを決定するために、集約クラスタ120の少なくとも1つに要求を送信する。集約クラスタ120が利用可能な場合に、フロントエンドクラスタ110は、取り込んだログデータを集約クラスタ120へリアルタイムで流す。集約クラスタ120のそれぞれは、多数のサーバからストリーミングログデータを受信し、ログデータを集約する。一実施形態において、ログデータの集約は、ファイル及びディレクトリを組み合わせる観点で、ログデータを再編成することを含む。集約されたログデータファイルは、着信ログデータファイルと必ずしも一対一のマッピングを有しない。集約クラスタ120は、非常に多数のサーバノードに対してスケーリングするように、且つネットワーク及びノード障害に対して堅牢なように設計される。一実施形態において、集約クラスタ120は、集約されたログデータを圧縮することを含む、集約されたログデータに対する追加処理を実行する。別の実施形態において、集約クラスタ120は、集約されたログデータに対して追加処理を実行しない。

40

50

**【 0 0 4 0 】**

次に、集約クラスタ120は、ハドゥープクラスタに対して実現されるハイブデータウェアハウスなどのデータウェアハウス130に、集約されたログデータを周期的にコピーする。幾つかの実施形態において、ハイブデータウェアハウスは、ペタバイトのデータを記憶することができる。他の幾つかの実施形態において、集約クラスタ120は、集約されたログデータを1つ又は複数のNFSフィラーに送信する。

**【 0 0 4 1 】**

幾つかの実施形態において、集約クラスタ120は、2レベル構造として実現することができる。集約クラスタの1つは、親集約クラスタと呼ばれる。集約クラスタの残りは、レベル2集約クラスタである。親集約クラスタは、フロントエンドクラスタ110からログデータを受信し、且つ受信したログデータをレベル2集約クラスタに分配する責任を負う。バックエンドサーバ上を走るデータ消費アプリケーションは、レベル2集約クラスタからログデータをリアルタイムで検索する。集約クラスタのこの2レベル構造は、バックエンドサーバが集約クラスタからログデータを引き出すための大きな帯域幅を提供する。他の幾つかの実施形態において、バックエンドサーバ上を走るデータ消費アプリケーションは、親集約クラスタから直接にリアルタイムでログデータを検索する。

10

**【 0 0 4 2 】**

ひとたびデータウェアハウス130が、関連するログデータを受信し処理すると、バックエンドサーバ140は、オンラインデータ分析用に照会要求をデータウェアハウス130に送信することができる。より高速なログデータアクセスのために、集約クラスタ120のそれぞれは、データステージングエリア122を含む。データステージングエリア122は、集約されたログデータを、処理の前に一時的に記憶するための集約クラスタ120内の中間記憶エリア(パーキングとも呼ばれる)である。一実施形態において、データステージングエリア122は、集約クラスタ120のサーバにわたって拡張することができる。バックエンドサーバ140は、リアルタイム又はほぼリアルタイムでのログデータ供給のために、データステージングエリア122に直接アクセスすることができる。データステージングエリア122に記憶されたログデータは、所定の期間後に削除されても良い。例えば、一実施形態において、ログデータは、ログデータが取り込まれた3日後に削除される。一実施形態において、バックエンドサーバ140は、オンラインデータ収集システム100の一部である。別の実施形態において、バックエンドサーバ140は、オンラインデータ収集システム100とは別個である。

20

**【 0 0 4 3 】**

幾つかの実施形態において、フロントエンドクラスタ110のサーバ112は、レベル1プラスステージングエリア115を含む。例えば、フロントエンドクラスタ110におけるフロントエンドサーバ112は、その対応する集約クラスタ120が利用不可能であることを通知される。現在利用不可能な集約クラスタ120にログデータを流す代わりに、フロントエンドサーバ112は、レベル1プラスステージングエリア115にログデータを一時的に記憶する。レベル1プラスステージングエリア115は、バックエンドサーバによって直接アクセスすることができる。換言すれば、バックエンドサーバ140は、集約クラスタ120にもデータウェアハウス130にもアクセスする必要ななしに、レベル1プラスステージングエリア115からログデータを直接流すことができる。他の幾つかの実施形態において、フロントエンドクラスタ110は、レベル1プラスステージングエリアがあってもなくても、システム内の任意のレベルからサーバによって直接アクセスできるレベル0ステージングエリアを更に含むことができる。

30

**【 0 0 4 4 】**

一実施形態において、オンラインデータ収集システムは、データを多数の優先層に分割することによって、ログデータに優先順位をつけることができる。例えば、トップ層のログデータは、データウェアハウス及びバックエンドサーバに送信される、より高い優先権を有する。トップ層のログデータはまた、データがステージングエリアにパーキングしている場合に、より長い保持時間を有することができる。より低い層のログデータは、トッ

40

50

ログデータより短い保持期間後に、より早く削除される。幾つかの実施形態において、優先権を設定する命令は、ポリシーシステムで実行することができる。データ消費アプリケーションが提示された場合に、それが必要とするデータカテゴリ用に、優先層と同様に帯域幅に関してポリシーを指定することができる。ポリシーは、更に、要求されたログデータが、ステージングエリアから、アプリケーションが走るバックエンドサーバへとリアルタイムで直接送信されるべきかどうかということ、要求されたログデータが、データウェアハウスから送信されるべきかどうかということ、又は要求されたログデータが、レベル1プラスステージングエリア、又は集約サーバにおけるステージングエリア上でステージングされるべきかどうかということを更に指定することができる。

#### 【0045】

ログデータの処理を加速するために、集約クラスタは、ログデータストリームが、バックエンドサーバ上を走るデータ消費アプリケーションに並列に送信されるように、ログデータを多数のログデータストリームに分割することができる。図2は、ログデータストリームを分割できる集約クラスタの例を示す。集約クラスタ210は、分割ログデータストリーム212(バケット(bucket)とも呼ばれる)に均等に且つランダムに分配されるログデータを保証するために、ログデータのエントリを検査することによってログデータを分割する。図3は、ログデータエントリの例示的な構造を示す。ログデータのエントリ300は、アプリケーションID310を含み、アプリケーションID310は、そのログデータエントリを消費し処理しようとしているデータ消費アプリケーションを識別する。一実施形態において、アプリケーションID310は、開発者IDである。エントリ300は、更に、カテゴリ320を含む。カテゴリ320は、メッセージの意図した宛先の高レベルの記述である。エントリ300は、更に、メッセージ330を含む。メッセージ330は、ログデータエントリの実際の内容を含む。

#### 【0046】

一実施形態において、集約サーバは、アプリケーションID及びカテゴリを読み取り、次に、アプリケーションID及びカテゴリのハッシュ関数を計算する。例えば、ハッシュ関数は、32ビット又は128ビットのハッシュ値を生じるMurmurHash3ハッシュ関数とすることができます。集約サーバは、更に、バケットの合計数によるハッシュ関数の値を法(modulo)とする。結果は、ログデータエントリが割り当てられるバケット番号である。換言すれば、ログデータエントリは、そのバケット番号を有するログデータストリームに含まれることになる。ハッシュ関数は、エントリがバケット(即ち、分割ログデータストリーム)間で均等に且つランダムに分配されるように、ログデータのエントリを入れ替える。エントリへのバケット番号の割り当ては、エントリが生成されるときにも又はエントリのアプリケーションIDとも相関を有しない。幾つかの実施形態において、バケット番号は、アプリケーションIDに加えてログデータエントリのデータフィールドの関数とすることができます。例えば、ハッシュ関数は、バケット番号を決定するためのハッシュ値を生成するために、ログデータエントリのアプリケーションID及び別のデータフィールドを用いることができる。

#### 【0047】

バケットの合計数は、様々な方法で決定することができる。一実施形態において、バックエンドサーバの合計数は、各バックエンドサーバが扱うことができる推定合計データ量及びデータ帯域幅によって決定される。バケットの数は、システムが、より多くのバックエンドサーバを含むようにスケールアップするために、バックエンドサーバの数より大きな数として決定される。例えば、バケットの数が1024である場合に、それは、システムが、ストリーム分割において変更を行う必要なしに、1024のバックエンドサーバまでスケールアップできることを意味する。複数のバックエンドサーバが、分割データストリームを並列にリアルタイムで処理するために利用され得る。ストリームが均等に分割されるので、バックエンドサーバは、等しく負荷をかけられる。別の実施形態において、データ消費アプリケーションを提示する場合に、ログデータストリームを並列処理するためのバケット数を含むデータ消費アプリケーション用のポリシーを指定することができる

10

20

30

40

50

。

#### 【 0 0 4 8 】

別の実施形態において、各ログデータエントリに割り当てられるバケット番号は、乱数によって決定される。集約サーバは、バケットの合計数の範囲内でランダム整数を生成し、その生成された整数を、割り当てられたバケット番号としてログデータエントリに割り当てる。

#### 【 0 0 4 9 】

更に別の実施形態において、データ消費アプリケーションは、バケット番号を割り当てるための関数又は方法を指定することができる。例えば、データ消費アプリケーションは、エントリの或る ID に基づいてバケット番号を割り当てることができる。データ消費アプリケーションは、エントリのユーザ ID の第 1 の範囲を第 1 のバケット番号に割り当てることができ、エントリのユーザ ID の第 2 の範囲を第 2 のバケット番号に割り当てることができる等である。10

#### 【 0 0 5 0 】

一実施形態において、オンラインデータ収集システムは、図 1 の集約サーバに似た中間サーバにおける 1 を超える層を有することができる。図 4 は、集約サーバの多層を有するオンラインデータ収集システムの別の例を示す。オンラインデータ収集システム 4 0 0 は、複数のフロントエンドクラスタ 4 1 0 を含む。各フロントエンドクラスタ 4 1 0 は、多数の相互接続されたフロントエンドサーバ 4 1 2 を含む。フロントエンドクラスタ 4 1 0 は、オンラインデータ収集システム 1 0 0 によって監視されたユーザ活動に基づいてリアルタイムでログデータを生成する。一実施形態において、フロントエンドクラスタ 1 1 0 におけるサーバ 4 1 2 は、バックエンドサーバ 4 5 0 によって直接アクセスされ得るレベル 1 プラスステージングエリア 4 1 5 を更に含むことができる。20

#### 【 0 0 5 1 】

オンラインデータ収集システム 4 0 0 は、フロントエンドクラスタ 4 1 0 から送信されたログデータを集約する責任を負う複数の第 1 のレベルの集約クラスタ 4 2 0 を更に含む。更に、オンラインデータ収集システム 4 0 0 は、第 1 のレベルの集約クラスタ 4 2 0 から送信されたログデータを集約する責任を負う 1 つ又は複数の第 2 のレベルの集約クラスタ 4 3 0 を含む。次に、第 2 のレベルの集約クラスタ 4 3 0 は、集約されたログデータを、ハドウープクラスタ上に実現されたハイブデータウェアハウスなどのデータウェアハウス 4 4 0 に周期的にコピーする。30

#### 【 0 0 5 2 】

第 1 のレベルの集約クラスタ 4 2 0 は、第 1 のレベルのデータステージングエリア 4 2 2 を含み、第 2 のレベルの集約クラスタ 4 3 0 は、第 2 のレベルのデータステージングエリア 4 3 2 を含む。バックエンドサーバ 4 5 0 は、第 1 のレベルのデータステージングエリア 4 2 2 又は第 2 のレベルのデータステージングエリア 4 3 2 からログデータを直接検索することができる。一実施形態において、バックエンドサーバ 4 5 0 は、それが、ネットワークトポロジに基づいて第 1 のレベルのステージングエリア又は第 2 のレベルのステージングエリアからログデータを検索するかどうかを決定することができる。例えば、ネットワークトポロジの点で、バックエンドサーバ 4 5 0 が、第 1 のレベルの集約クラスタ 4 2 0 より第 2 のレベルの集約クラスタ 4 3 0 に一層近い場合に、バックエンドサーバ 4 5 0 は、第 2 のレベルのデータステージングエリア 4 3 2 からログデータを検索することを決定する。40

#### 【 0 0 5 3 】

例えば、一例において、ログデータは、ユーザの活動データであり、データ消費アプリケーションは、ニュースフィードアプリケーションである。或る期間後にデータウェアハウス上で利用可能なオフラインログデータを待つのではなく、ニュースフィードアプリケーションは、どちらの集約クラスタが、ネットワークトポロジにおいてニュースフィードアプリケーションのバックエンドサーバに一層近いかに依存して、第 1 又は第 2 のレベルのステージングエリアから、関連するユーザ活動のログデータをリアルタイムで検索する50

ように要求することができる。ニュースフィードアプリケーションは、非常に低いレイテンシーでユーザ活動のログデータを引き出し、且つできるだけ早くログデータを処理しニュースフィード結果を格納することができる。

#### 【0054】

一実施形態において、多重レベルの集約クラスタを備えたオンラインデータ収集システムが、レベル1プラスステージング能力を備えたフロントエンドクラスタを含む。集約クラスタが利用不可能な場合に、フロントエンドクラスタは、バックエンドサーバログデータを供給し続けるために、自らのレベル1プラスステージング能力を用いることができる。別の実施形態において、オンラインデータ収集システムは、データステージング能力を備えた2を超えるレベルの集約クラスタを含むことさえできる。

10

#### 【0055】

多重レベルのステージング能力を有するかかるオンラインデータ収集システムは、ログデータ集約の各ステージにおいてリアルタイムでログデータを供給することができる。システムは、ログデータが中央データリポジトリ（例えばデータウェアハウス）からまだ入手可能でない場合に、各ステージにおけるデータ消費照会に応じて、低レイテンシのデータ供給を提供することができる。バックエンドサーバは、ネットワークトポロジの点から、ステージログデータを検索するために、より近いクラスタを選択することができる。バックエンドサーバのデータストリームは、ネットワークにおいて、より少数のノード及びシステム通って移動する。従って、送信中にログデータを失うか又はログデータを破損する公算は、より少ない。従って、多重レベルのステージングシステムは、ログデータ送信の耐久性及び信頼性を向上させる。

20

#### 【0056】

図5は、ログデータを集約してステージングするための例示的な処理を示す。ステップ510において、フロントエンドサーバは、リアルタイムのユーザ活動に基づいてログデータを生成する。一実施形態において、フロントエンドサーバは、更に、ウェブコンテンツをユーザに提供する。一実施形態において、フロントエンドサーバは、ログデータを複数の優先層に分割し、それら優先層のうちのトップ層のログデータだけを集約サーバに送信する。

#### 【0057】

ステップ514において、フロントエンドサーバは、集約サーバが利用可能かどうかをチェックする。集約サーバが利用不可能な場合に、フロントエンドサーバは、ステップ516において、複数のフロントエンドサーバの少なくとも1つにおけるレベル1プラスステージングエリアにおいてログデータをステージングする。ステップ518において、フロントエンドサーバは、バックエンドサーバが要求した場合に、レベル1プラスステージングエリアからバックエンドサーバにログデータをリアルタイムで供給する。集約サーバが利用可能な場合に、ステップ520において、フロントエンドサーバは、ログデータを集約サーバに送信する。

30

#### 【0058】

ステップ530において、集約サーバは、ログデータを集約する。一実施形態において、集約は、集約サーバにおいてログデータのストリームを組み合わせることによって、ログデータを再編成することを含む。別の実施形態において、集約サーバは、更に、集約サーバにおいてログデータを圧縮する。

40

#### 【0059】

ステップ540において、集約サーバは、バックエンドサーバがログデータにリアルタイムでアクセスできるように、ログデータをステージングする。一実施形態において、1つのバックエンドサーバが、ログデータ用のデータステージング位置に関する命令を集約サーバ及びフロントエンドサーバに送信する。次に、集約サーバ及びフロントエンドサーバは、命令に従ってログデータをステージングするかどうかを決定する。ステップ550において、集約サーバは、集約サーバから直接バックエンドサーバにログデータをリアルタイムで供給する。

50

**【0060】**

ステップ560において、集約サーバは、ログデータをデータウェアハウスに送信する。ステップ570において、データウェアハウスは、それが、処理されたログデータに基づいてデータ照会に応答できるように、ログデータを処理する。

**【0061】**

ステップ580において、集約サーバは、所定の期間後に、集約サーバからログデータを削除する。フロントエンドサーバが、レベル1プラスステージングエリアにおいてログデータをステージングする場合に、フロントエンドサーバもまた、所定の期間後に、レベル1プラスステージングエリアからログデータを削除する。

**【0062】**

10

図6は、ログデータをステージング及び分割するための例示的な処理を示す。ステップ610において、フロントエンドサーバは、リアルタイムのユーザ活動に基づいてログデータを生成する。一実施形態において、フロントエンドサーバは、更に、ウェブコンテンツをユーザに提供する。

**【0063】**

ステップ614において、フロントエンドサーバは、集約サーバが利用可能かどうかをチェックする。集約サーバが利用不可能な場合に、フロントエンドサーバは、ステップ616において、複数のフロントエンドサーバの少なくとも1つにおけるレベル1プラスステージングエリアにおいてログデータをステージングする。

**【0064】**

20

ステップ618において、フロントエンドサーバは、バックエンドサーバが要求した場合に、レベル1プラスステージングエリアからバックエンドサーバにリアルタイムでログデータを供給する。集約サーバが利用可能な場合に、ステップ620において、フロントエンドサーバは、ログデータを集約サーバに送信する。

**【0065】**

ステップ630において、集約サーバは、ログデータを集約する。一実施形態において、集約は、集約サーバにおいてログデータのストリームを組み合わせることによって、ログデータを再編成することを含む。別の実施形態において、集約サーバは、更に、集約サーバにおいてログデータを圧縮する。

**【0066】**

30

ステップ640において、集約サーバは、バックエンドサーバがログデータにリアルタイムでアクセスできるように、ログデータをステージングする。

一実施形態において、ログデータは、複数のログデータエントリを含む。各ログデータエントリは、アプリケーション識別及びカテゴリフィールドを含む。カテゴリフィールドは、ログデータエントリの意図された宛先の高レベルの記述を含む。アプリケーション識別は、ログデータエントリを処理するためのデータ消費アプリケーションを識別する。集約サーバは、ログデータのエントリの接頭部を検査して、アプリケーション識別及びカテゴリフィールドを認識する。

**【0067】**

一実施形態ではステップ650において、ログデータの各エントリ用に、集約サーバは、パケットの合計数を法として、アプリケーション識別及びカテゴリフィールドのハッシュ関数によってパケット番号を計算する。パケットの合計数は、複数のログデータストリームの合計数である。一実施形態において、パケットの合計数は、ログデータストリームを受信するために利用可能なバックエンドサーバの数、及び各バックエンドサーバが扱うことができる接続部の数によって決定される。別の実施形態において、パケットの合計数は、少なくとも1つのバックエンドサーバ上を走るデータ消費アプリケーションによって指示される。その目的は、バックエンドサーバが、ログデータストリームを受信し処理する場合に、バックエンドサーバが等しく負荷をかけられるということである。ステップ652において、集約サーバは、ログデータのそのエントリを、パケット番号によって識別されたログデータストリームに割り当てる。

40

50

**【 0 0 6 8 】**

別の実施形態において、各ログデータエントリ用に、集約サーバは、1からパケットの合計数までの整数をランダムに生成する。次に、集約サーバは、そのログデータエントリを、パケット番号によって識別されたログデータストリームに割り当てる。

**【 0 0 6 9 】**

ステップ654において、集約サーバは、ログデータを複数のログデータストリームに分割する。一実施形態において、集約サーバは、ログデータをどのように複数のログデータストリームに分割するかに関する命令をバックエンドサーバから受信する。一実施形態において、フロントエンドサーバはまた、集約サーバがログデータを分割するのと似た方法で、ログデータを複数のログデータストリームに分割することができる。ステップ660において、集約サーバは、ログデータストリームを1つ又は複数のバックエンドサーバに並列に供給する。10

**【 0 0 7 0 】**

ステップ670において、集約サーバは、ログデータをデータウェアハウスに送信する。ステップ680において、データウェアハウスは、それが、処理されたログデータに基づいてデータ照会に応答できるように、ログデータを処理する。

**【 0 0 7 1 】**

ステップ690において、集約サーバは、所定の期間後に、ログデータを集約サーバから削除する。フロントエンドサーバが、レベル1プラスステージングエリアにおいてログデータをステージングする場合に、フロントエンドサーバもまた、所定の期間後に、レベル1プラスステージングからログデータを削除する。20

**【 0 0 7 2 】**

一実施形態において、バックエンドサーバは、それらが分割ログデータストリームを受信した後で、分割ログデータストリームを統合する。分割前のログデータが複数のストリームに属すると仮定すると、ストリームのそれぞれは、バックエンドサーバの1つに割り当てられる。バックエンドサーバのいずれかが、分割ログデータストリームのメッセージを集約サーバから受信した場合に、それは、メッセージが属するストリーム用にどのバックエンドサーバが割り当てられるかを決定する。次に、バックエンドサーバは、割り当てられたバックエンドサーバ(オーナとも呼ばれる)にメッセージを転送する。一実施形態において、バックエンドサーバは、1つのカテゴリ用のログデータをキューに統合する。キューは、様々な方法で実現することができる。例えば、キューは、最後のファイルだけが潜在的なオープンファイルであるディレクトリにおけるファイルと類似した命名規則において実現することができる。キューを実現する別の方法は、2層システムを利用することができる。30

**【 0 0 7 3 】**

幾つかの実施形態において、ストリーム対サーバ割り当て情報を記録するために、バックエンドサーバは、割り当て情報用の分配マップを維持する。分配マップは、割り当て情報に変更がある場合、例えば、サーバが利用不可能な場合、又は新しいストリームが追加された場合に、更新される。バックエンドサーバは、低レイテンシでこれらの変更の通知を得る。40

**【 0 0 7 4 】**

このストリーム対サーバの割り当て機構は、ストリームとバックエンドサーバとの間の関係を管理するための自動で動的な方法を提供する。バックエンドサーバの1つが故障した場合に、残りのバックエンドサーバは、故障したバックエンドサーバに割り当てられたストリームのオーナーシップを再分配することができる。幾つかのバックエンドサーバから他のバックエンドサーバへと幾つかのストリームのオーナーシップを移すことによって、この機構は、バックエンドサーバの仕事量のバランスを保つことができる。

**【 0 0 7 5 】**

図7は、サーバ700のアーキテクチャの例を示す高レベルブロック図であり、サーバ700は、フロントエンドサーバ、集約サーバ、データウェアハウスサーバ、及びバック

エンドサーバのいずれを表しても良い。サーバ700は、相互接続部730に結合された1つ又は複数のプロセッサ710及びメモリ720を含む。図7に示されている相互接続部730は、適切なブリッジ、アダプタ又はコントローラによって接続された任意の1つ若しくは複数の別個の物理バス、ポイントツーポイント接続部、又は両方を表す抽象概念である。従って、相互接続部730は、例えば、システムバス、周辺構成要素相互接続(PCIE)バス若しくはPCIEエクスプレスバス、ハイパーテラントスポート若しくは業界標準アーキテクチャ(ISA)バス、小型コンピュータシステムインターフェース(SCSI)バス、ユニバーサルシリアルバス(USB)、IIC(I2C)バス、又は「FireWire」(登録商標)とも呼ばれる電気電子技術者協会(IEEE)標準1394バスを含んでも良い。

10

#### 【0076】

プロセッサ710は、サーバ700の中央処理装置(CPU)であり、従って、サーバ700の全体的動作を制御する。或る実施形態において、プロセッサ710は、これを、メモリ720に記憶されたソフトウェア又はファームウェアを実行することによって達成する。プロセッサ710は、1つ又は複数のプログラマブル汎用若しくは特定目的マイクロプロセッサ、デジタル信号プロセッサ(DSP)、プログラマブルコントローラ、特定用途向け集積回路(AVIC)、プログラマブル論理装置(PLD)、トラステッドプラットホームモジュール(TPM)等、又はかかる装置の組み合わせを含んでも含まなくても良い。

20

#### 【0077】

メモリ720は、サーバ700のメインメモリであるか、又はそれを含む。メモリ720は、ランダムアクセスメモリ(RAM)、読み取り専用メモリ(ROM)、フラッシュメモリ等、又はかかる装置の組み合わせの任意の形態を表す。使用において、メモリ720は、とりわけ、サーバ700のオペレーティングシステムの少なくとも一部を具体化するコード770を含んでも良い。コード770はまた、本明細書で開示される技術を実行するための命令を含んでも良い。

#### 【0078】

相互接続部730を通してプロセッサ710に同様に接続されるのは、ネットワークアダプタ740及び記憶アダプタ750である。ネットワークアダプタ740は、他のフロントエンドサーバ、統合サーバ、データウェアハウスサーバ、又はバックエンドサーバなどの装置と、ネットワークを通じて通信する能力をサーバ700に提供し、且つ例えればインターネット(登録商標)アダプタ又はファイバチャンネルアダプタであっても良い。幾つかの実施形態において、サーバは、データ記憶クラスタ内及び外の通信に別々に対処するために、1を超えるネットワークアダプタを用いても良い。記憶アダプタ750は、サーバ700が永続記憶装置にアクセスできるようにし、且つ例えればファイバチャンネルアダプタ又はSCSIアダプタであっても良い。

30

#### 【0079】

メモリ720に記憶されたコード770は、以下で説明される動作を実行するようにプロセッサ710をプログラムするためのソフトウェア及び/又はファームウェアとして実現されても良い。或る実施形態において、かかるソフトウェア又はファームウェアは、最初は、それをシステムからサーバ700を通して(例えればネットワークアダプタ740を介して)ダウンロードすることによって、サーバ700に提供されても良い。

40

#### 【0080】

本明細書で導入される技術は、例えば、ソフトウェア及び/又しくはファームウェアでプログラムされたプログラマブル回路(例えば1つ若しくは複数のマイクロプロセッサ)によってか、完全に特定目的ハードワイヤード回路においてか、又はかかる形態の組み合わせにおいて実行することができる。特定目的ハードワイヤード回路は、例えば、1つ又は複数の特定用途向け集積回路(AVIC)、プログラマブル論理装置(PLD)、フィールドプログラマブルゲートアレイ(FPGA)等の形態であっても良い。

#### 【0081】

50

本明細書で導入される技術の実行において使用するためのソフトウェア又はファームウェアは、機械可読記憶媒体上に記憶されても良く、且つ1つ又は複数の汎用又は特定目的プログラマブルマイクロプロセッサによって実行されても良い。「機械可読記憶媒体」は、その用語が本明細書で用いられているように、機械（機械は、例えば、コンピュータ、ネットワーク装置、携帯電話、携帯情報端末（PDA）、製造用工具、1つ又は複数のプロセッサを備えた任意の装置等であっても良い）によってアクセス可能な形態で情報を記憶できる任意の機構を含む。例えば、機械アクセス可能な記憶媒体は、記録可能な／記録不能な媒体（例えば、読み取り専用メモリ（ROM）、ランダムアクセスメモリ（RAM）、磁気ディスク記憶媒体、光記憶媒体、フラッシュメモリ装置等）などを含む。

## 【0082】

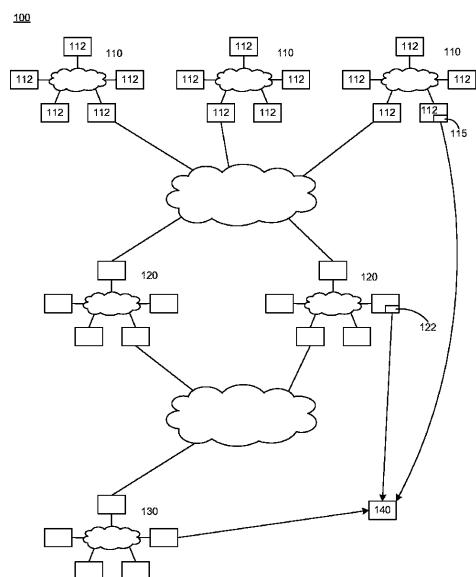
10

用語「論理」は、本明細書で用いられているように、例えば、特定のソフトウェア及び／又はファームウェアでプログラムされたプログラマブル回路、特定目的のハードウェイード回路、又はそれらの組み合わせを含むことができる。

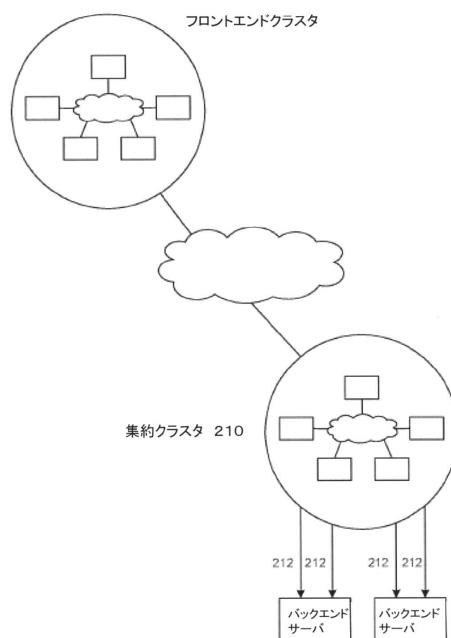
## 【0083】

上記の例に加えて、本発明の様々な他の変形形態及び変更形態が、本発明から逸脱せずに行われ得る。従って、上記の開示は、限定と見なされるべきではなく、添付の特許請求の範囲が、本発明の真の趣旨及び全範囲を包含すると解釈されるべきである。

【図1】



【図2】



【図3】



【図4】

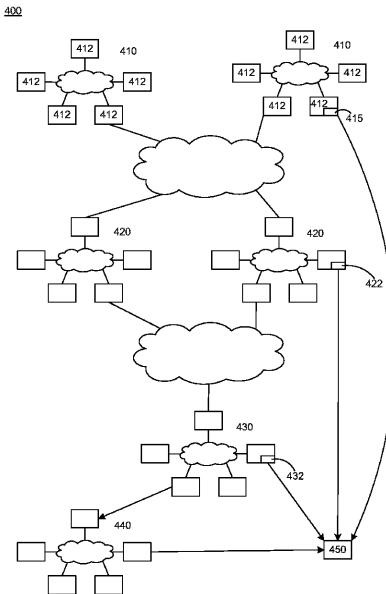
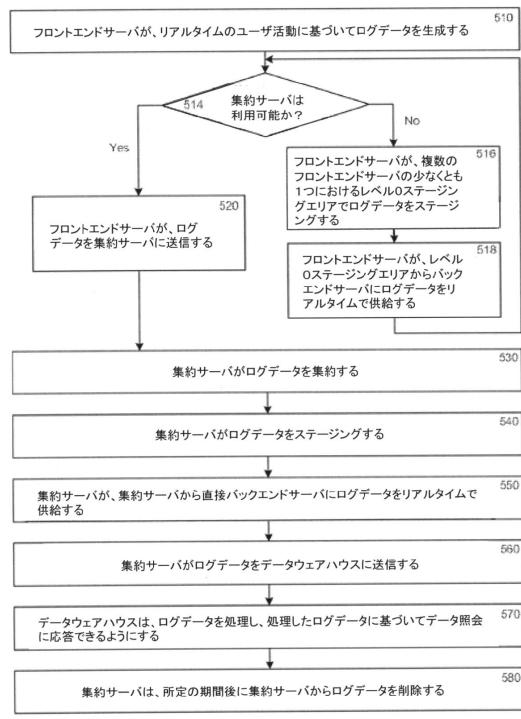
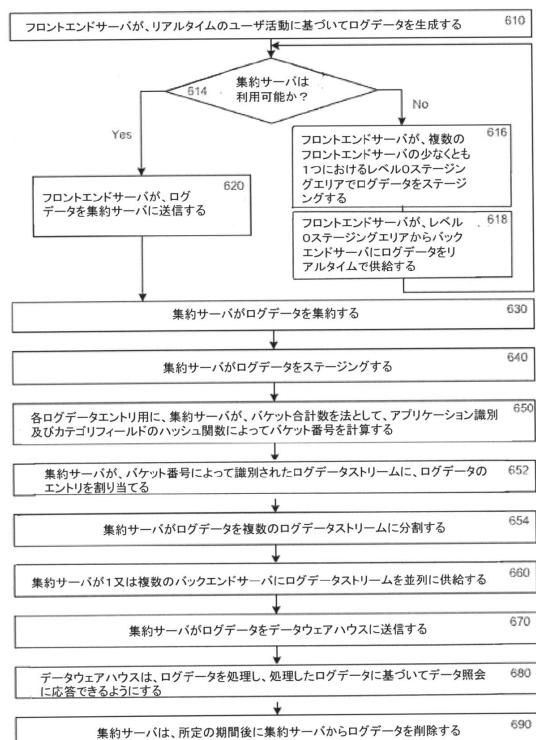


FIG. 4

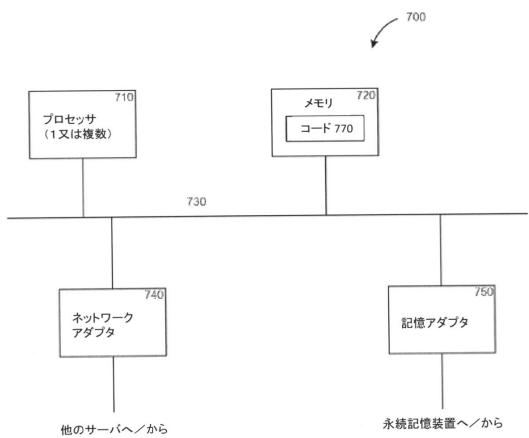
【図5】



【図6】



【図7】



---

フロントページの続き

(72)発明者 ボルタクール、ドゥルバジヨーティ  
アメリカ合衆国 94025 カリフォルニア州 メンロー パーク ウィロー ロード 160  
1

(72)発明者 シャオ、チェン  
アメリカ合衆国 94025 カリフォルニア州 メンロー パーク ウィロー ロード 160  
1

(72)発明者 ファン、エリック  
アメリカ合衆国 94025 カリフォルニア州 メンロー パーク ウィロー ロード 160  
1

審査官 田中 秀樹

(56)参考文献 特開2008-305352(JP,A)  
特開2011-39820(JP,A)  
特開2008-204206(JP,A)  
米国特許出願公開第2011/0246826(US,A1)  
米国特許第6978458(US,B1)  
米国特許出願公開第2008/0256079(US,A1)

(58)調査した分野(Int.Cl., DB名)

G 06 F 17 / 30  
G 06 F 13 / 00  
G 06 F 12 / 00  
G 06 F 9 / 46