

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2016-539427
(P2016-539427A)

(43) 公表日 平成28年12月15日(2016.12.15)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 17/30 (2006.01)	G06F 17/30 350C	
	G06F 17/30 415	

審査請求 未請求 予備審査請求 未請求 (全 52 頁)

(21) 出願番号 特願2016-536781 (P2016-536781)
 (86) (22) 出願日 平成26年12月4日 (2014.12.4)
 (85) 翻訳文提出日 平成28年6月10日 (2016.6.10)
 (86) 国際出願番号 PCT/US2014/068641
 (87) 国際公開番号 WO2015/085103
 (87) 国際公開日 平成27年6月11日 (2015.6.11)
 (31) 優先権主張番号 61/912,344
 (32) 優先日 平成25年12月5日 (2013.12.5)
 (33) 優先権主張国 米国 (US)
 (31) 優先権主張番号 14/559,550
 (32) 優先日 平成26年12月3日 (2014.12.3)
 (33) 優先権主張国 米国 (US)

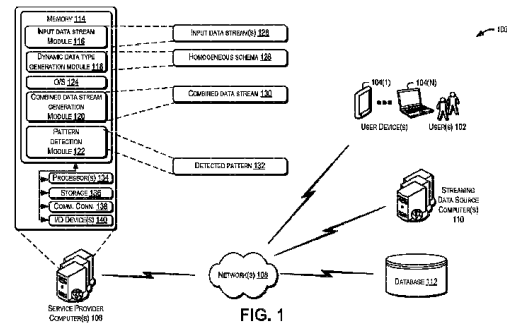
(71) 出願人 502303739
 オラクル・インターナショナル・コーポレーション
 アメリカ合衆国カリフォルニア州94065
 レッドウッド・シティー, オラクル・パークウェイ500
 (74) 代理人 110001195
 特許業務法人深見特許事務所
 (72) 発明者 カリ, プラタプ
 インド、560078 カルナータカ、バンガロール、ジェイ・ピー・ナガー・セカンド・フェイズ、トゥウェンティセカンド・メイン、テンス・クロス、899、ファースト・フロア

最終頁に続く

(54) 【発明の名称】 多重入力データストリームにわたるパターンマッチング

(57) 【要約】

1つ以上のアプリケーションに関する多重入力データストリームにわたってパターンを検出するための方法が開示される。当該方法は、多重入力データストリームを受取るステップと、入力データストリームの1つ以上の属性についての1つ以上のダイナミックデータタイプを生成するステップとを含む。いくつかの実施形態においては、当該方法は、入力データストリームを結合して、ダイナミックデータタイプに基づいて結合入力データストリームを生成するステップと、結合データストリームに関する連続クエリを処理してパターンを検出するステップとを含み得る。



【特許請求の範囲】**【請求項 1】**

コンピュータによって実行される方法であって、

コンピュータ実行可能命令を用いて構成されるコンピュータシステムによって、第 1 の入力データストリームおよび第 2 の入力データストリームを少なくとも含む複数の入力データストリームを受取るステップと、

前記コンピュータシステムによって、前記第 1 の入力データストリームについての第 1 のダイナミックデータタイプを生成するステップと、

前記コンピュータシステムによって、前記第 2 の入力データストリームについての第 2 のダイナミックデータタイプを生成するステップと、

前記コンピュータシステムによって、前記第 1 の入力データストリームと前記第 2 の入力データストリームとを結合して、前記第 1 のダイナミックデータタイプおよび前記第 2 のダイナミックデータタイプに少なくとも部分的に基づく結合データストリームを生成するステップと、

前記コンピュータシステムによって前記結合データストリームに関する連続クエリを処理して、パターンを検出するステップとを含む、コンピュータによって実行される方法。

【請求項 2】

前記第 1 のダイナミックデータタイプを生成するステップは、

前記第 1 の入力データストリームの第 1 の属性を前記第 2 の入力データストリーム中に存在しないものとして識別するステップと、

前記第 1 の属性についての前記第 1 のダイナミックデータタイプを生成するステップとを含む、請求項 1 に記載の、コンピュータによって実行される方法。

【請求項 3】

前記第 1 のダイナミックデータタイプは、前記第 1 の入力データストリームの前記第 1 の属性に対応する第 1 のデータ値を格納するように構成される、請求項 2 に記載のコンピュータによって実行される方法。

【請求項 4】

前記第 2 のダイナミックデータタイプを生成するステップは、

前記第 2 の入力データストリームの第 2 の属性を前記第 1 の入力データストリーム中に存在しないものとして識別するステップと、

前記第 2 の属性についての前記第 2 のダイナミックデータタイプを生成するステップとを含み、前記第 2 のダイナミックデータタイプは、前記第 2 の入力データストリームの前記第 2 の属性に対応する第 2 のデータ値を格納するように構成される、請求項 1 ~ 3 のいずれか 1 項に記載のコンピュータによって実行される方法。

【請求項 5】

共通属性を識別するステップをさらに含み、前記共通属性は、前記第 1 の入力データストリームに存在するものとして、かつ前記第 2 の入力データストリームに存在するものとして識別され、さらに、

同種スキーマを生成するステップを含み、前記同種スキーマは、前記第 1 の入力データストリームおよび前記第 2 の入力データストリームの 1 つ以上の属性の表現を含み、前記表現は、少なくとも、前記共通属性、前記第 1 のダイナミックデータタイプおよび前記第 2 のダイナミックデータタイプを含む、請求項 1 ~ 4 のいずれか 1 項に記載のコンピュータによって実行される方法。

【請求項 6】

前記同種スキーマは、ストリーム名識別子属性、前記第 1 の入力データストリームに関連付けられた第 1 のタイムスタンプ属性、または前記第 2 の入力データストリームに関連付けられた第 2 のタイムスタンプ属性のうち少なくとも 1 つを含む、請求項 5 に記載のコンピュータによって実行される方法。

【請求項 7】

前記第 1 の入力データストリームと前記第 2 の入力データストリームとを結合するステ

10

20

30

40

50

ップはさらに、

前記同種スキーマによって識別される前記第 1 の入力データストリームからタブルの第 1 のセットを選択するステップと、

前記同種スキーマによって識別される前記第 2 の入力データストリームからタブルの第 2 のセットを選択するステップと、

前記タブルの第 1 のセットと前記タブルの第 2 のセットとに関するサブクエリを処理して、前記結合データストリームを生成するステップとを含む、請求項 5 または 6 に記載のコンピュータによって実行される方法。

【請求項 8】

前記パターンは、前記結合データストリームの分析に少なくとも部分的に基づいて検出され、前記パターンは、前記第 1 の入力データストリームにおける第 1 のイベントと、後続する前記第 2 の入力データストリームにおける第 2 のイベントとを識別する、請求項 1 ~ 7 のいずれか 1 項に記載のコンピュータによって実行される方法。

10

【請求項 9】

システムであって、

複数の命令を格納するメモリと、

前記メモリにアクセスするように構成されたプロセッサとを含み、前記プロセッサはさらに、少なくとも、

第 1 の入力データストリームおよび第 2 の入力データストリームを識別する連続クエリを受取り、

20

前記第 1 の入力データストリームの第 1 の属性についての第 1 のダイナミックデータタイプを識別し、

前記第 2 の入力データストリームの第 2 の属性についての第 2 のダイナミックデータタイプを識別し、

前記第 1 の入力データストリームにおいて識別された前記第 1 のダイナミックデータタイプと、前記第 2 の入力データストリームにおいて識別された前記第 2 のダイナミックデータタイプとに少なくとも部分的に基づく結合データストリームを生成し、

前記結合データストリームに関する前記連続クエリを実行してパターンを検出するための、前記複数の命令を実行するように構成される、システム。

【請求項 10】

30

前記少なくとも 1 つのプロセッサは、前記第 1 の入力データストリームの前記第 1 の属性を、前記第 2 の入力データストリーム中に存在しないものとして識別するための前記コンピュータ実行可能命令を実行するように構成される、請求項 9 に記載のシステム。

【請求項 11】

前記少なくとも 1 つのプロセッサは、前記第 2 の入力データストリームの前記第 2 の属性を、前記第 1 の入力データストリーム中に存在しないものとして識別するための前記コンピュータ実行可能命令を実行するように構成される、請求項 9 または 10 に記載のシステム。

【請求項 12】

40

前記少なくとも 1 つのプロセッサはさらに、

前記第 1 の入力データストリームおよび前記第 2 の入力データストリーム中に存在するものとして識別された共通属性を識別し、

同種スキーマを生成するための、前記コンピュータ実行可能命令を実行するように構成され、前記同種スキーマは、少なくとも、前記共通属性、前記第 1 の属性、前記第 1 のダイナミックデータタイプ、前記第 2 の属性、および前記第 2 のダイナミックデータタイプの表現を含む、請求項 9 ~ 11 のいずれか 1 項に記載のシステム。

【請求項 13】

前記同種スキーマはさらに、ストリーム名識別子属性、前記第 1 の入力データストリームに関連付けられた第 1 のタイムスタンプ属性、または前記第 2 の入力データストリームに関連付けられた第 2 のタイムスタンプ属性のうち少なくとも 1 つを含む、請求項 12 に

50

記載のシステム。

【請求項 14】

前記少なくとも 1 つのプロセッサはさらに、結合データストリームを生成するための前記コンピュータ実行可能命令を実行するように構成され、前記結合データストリームは、前記同種スキーマによって識別される前記第 1 の入力データストリームからタブルの第 1 のセットを選択し、前記同種スキーマによって識別される前記第 2 の入力データストリームからタブルの第 2 のセットを選択し、前記タブルの第 1 のセットおよび前記タブルの第 2 のセットに関するサブクエリを処理して前記結合データストリームを生成するための、命令を実行することによって生成される、請求項 13 に記載のシステム。

10

【請求項 15】

前記パターンは、前記結合データストリームの分析に少なくとも部分的に基づいて検出され、前記パターンは、前記第 1 の入力データストリームにおける第 1 のイベントと、後続する前記第 2 の入力データストリームにおける第 2 のイベントとを識別する、請求項 9 ~ 14 のいずれか 1 項に記載のシステム。

【請求項 16】

1 つ以上のプロセッサによって実行可能なコンピュータ実行可能命令を格納する 1 つ以上の非一時的なコンピュータ読取可能媒体であって、前記コンピュータ実行可能命令は、前記 1 つ以上のプロセッサに、第 1 の入力データストリームおよび第 2 の入力データストリームを少なくとも含む複数の入力データストリームを受取らせる命令と、前記 1 つ以上のプロセッサに、前記第 1 の入力データストリームについての第 1 のダイナミックデータタイプを生成させる命令と、前記 1 つ以上のプロセッサに、前記第 2 の入力データストリームについての第 2 のダイナミックデータタイプを生成させる命令と、前記 1 つ以上のプロセッサに、前記第 1 の入力データストリームと前記第 2 の入力データストリームとを結合させて、前記第 1 のダイナミックデータタイプおよび前記第 2 のダイナミックデータタイプに少なくとも部分的に基づく結合データストリームを生成させる命令と、

20

前記 1 つ以上のプロセッサに、前記結合データストリームに関する連続クエリを処理させてパターンを検出させる命令とを含む、コンピュータ読取可能媒体。

30

【請求項 17】

前記第 1 のダイナミックデータタイプを生成させる命令はさらに、前記第 1 の入力データストリームの第 1 の属性を前記第 2 の入力データストリームに存在しないものとして識別する命令と、前記第 1 の属性についての前記第 1 のダイナミックデータタイプを生成する命令とを含み、前記第 1 のダイナミックデータタイプは、前記第 1 の入力データストリームの前記第 1 の属性に対応する第 1 のデータ値を格納するように構成される、請求項 16 に記載のコンピュータ読取可能媒体。

【請求項 18】

前記第 2 のダイナミックデータタイプを生成させる命令はさらに、前記 1 つ以上のプロセッサに、前記第 2 の入力データストリームの第 2 の属性を前記第 1 の入力データストリーム中に存在しないものとして識別させる命令と、前記第 2 の属性についての前記第 2 のダイナミックデータタイプを生成させる命令とを含み、前記第 2 のダイナミックデータタイプは、前記第 2 の入力データストリームの前記第 2 の属性に対応する第 2 のデータ値を格納するように構成される、請求項 16 または 17 に記載のコンピュータ読取可能媒体。

40

【請求項 19】

前記コンピュータ実行可能命令はさらに、

50

前記1つ以上のプロセッサに共通属性を識別させる命令を含み、前記共通属性は、前記第1の入力データストリームに存在するものとして、かつ前記第2の入力データストリームに存在するものとして識別され、前記コンピュータ実行可能命令はさらに、

前記1つ以上のプロセッサに同種スキーマを生成させる命令を含み、前記同種スキーマは、前記第1の入力データストリームおよび前記第2の入力データストリームの1つ以上の属性の表現を含み、前記表現は、少なくとも、前記共通属性、前記第1のダイナミックデータタイプおよび前記第2のダイナミックデータタイプを含み、前記コンピュータ実行可能命令はさらに、

前記1つ以上のプロセッサに、前記同種スキーマに少なくとも部分的に基づいて前記結合データストリームを生成させる命令を含む、請求項16～18のいずれか1項に記載のコンピュータ読取可能媒体。

10

【請求項20】

前記コンピュータ実行可能命令はさらに、前記1つ以上のプロセッサに、前記結合データストリームの分析に少なくとも部分的に基づいて前記パターンを検出させる命令を含み、前記パターンは、前記第1の入力データストリームにおける第1のイベントと、後続する前記第2の入力データストリームにおける第2のイベントとを識別する、請求項16～19のいずれか1項に記載のコンピュータ読取可能媒体。

【発明の詳細な説明】

【技術分野】

【0001】

20

関連出願の相互参照

本願は、2014年12月3日に提出され、「多重入力データストリームにわたるパターンマッチング(PATTERN MATCHING ACROSS MULTIPLE INPUT DATA STREAMS)」と題された米国特許出願第14/559,550号と、2013年12月5日に提出され、「多重入力データストリームにわたるパターンマッチング(PATTERN MATCHING ACROSS MULTIPLE INPUT DATA STREAMS)」と題された米国仮特許出願第61/912,344号とに対する優先権を主張するものであって、各々の内容全体があらゆる目的のために引用により援用されている。

【背景技術】

【0002】

30

背景

従来のデータベースシステムにおいては、データが通常は表の形式である1つ以上のデータベースに格納される。格納されたデータは、次いで、構造化クエリ言語(structured query language: SQL)などのデータ管理言語を用いて照会および処理される。たとえば、SQLクエリは、データベースに格納されたデータから関連データを識別するように規定および実行されてもよい。このため、SQLクエリは、データベースに格納された有限集合データ上で実行される。さらに、SQLクエリが実行される場合、このSQLクエリは、有限データ集合に対して1回だけ実行され、有限の静的な結果を生成する。このため、データベースは、有限の格納データ集合に対してクエリを実行するために最適に実装される。

40

【0003】

しかしながら、いくつかの現在のアプリケーションおよびシステムは、有限のデータ集合ではなく、連続的なデータまたはイベントストリームの形式でデータを生成する。このようなアプリケーションの例は、センサデータアプリケーション、金融ティッカ、ネットワーク性能測定ツール(たとえばネットワークモニタリングおよびトラフィック管理アプリケーション)、クリックストリーム解析ツール、自動車交通モニタリングなどを含むがこれらに限定されない。このようなアプリケーションのために、データストリームを処理することのできる新しい種類のアプリケーションが必要となってきた。たとえば、温度センサは、温度測定値を連続的に発信するように構成されてもよい。

【0004】

50

これらのタイプのイベントストリームベースのアプリケーションのためのデータを管理および処理するには、一時的に極集中的にデータの管理および照会能力を構築する必要がある。連続的な無限のセットのデータに対する長期間の照会を含む、さまざまな種類の照会メカニズムが必要とされる。ベンダーの中には、現在、イベントストリーム処理用に意図された製品 1 式を提供しているものもあるが、これらの提供される製品には、依然として、現代のイベント処理のニーズに対処するのに必要な処理融通性が欠けている。

【発明の概要】

【課題を解決するための手段】

【0005】

概要

いくつかの実施形態においては、1つ以上のアプリケーションに関する多重入力データストリームにわたるパターンを検出するための技術（たとえば、1つ以上のプロセッサによって実行可能なコードまたは命令を格納する方法、システム、非一時的なコンピュータ読取可能媒体）が提供される。

【0006】

一実施形態に従うと、1つ以上のアプリケーションに関する多重入力データストリームにわたってパターンを検出するための方法が開示される。当該方法は、第1の入力データストリームおよび第2の入力データストリームを含む複数の入力データストリームを受取るステップを含む。当該方法はさらに、第1の入力データストリームについての第1のダイナミックデータタイプを生成するステップと、第2の入力データストリームについての第2のダイナミックデータタイプを生成するステップとを含む。いくつかの実施形態においては、第1のダイナミックデータタイプは、第1の入力データストリームの第1の属性を、第2のデータストリームには存在しないものとして識別することによって生成されてもよい。次いで、第1のダイナミックデータタイプが、第1の属性のために生成される。第2のダイナミックデータタイプは、第2の入力データストリームの第2の属性を、第1のデータストリームには存在しないものとして識別することによって生成されてもよい。次いで、第2のダイナミックデータタイプが第2の属性のために生成される。一実施形態においては、第1のダイナミックデータタイプは、第1の入力データストリームの第1の属性に対応する第1のデータ値を格納するように構成されてもよく、第2のダイナミックデータタイプは、第2の入力データストリームの第2の属性に対応する第2のデータ値を格納するように構成されてもよい。

【0007】

いくつかの実施形態においては、当該方法は、第1の入力データストリームと第2の入力データストリームとを結合して、第1のダイナミックデータタイプおよび第2のダイナミックデータタイプに基づく結合データストリーム（combined data stream）を生成するステップを含み得る。一実施形態においては、当該方法はさらに、結合データストリームに対する連続クエリを処理してパターンを検出するステップを含み得る。いくつかの実施形態においては、「パターン」は、第1のデータストリームにおける第1のイベントの発生と、これに後続する第2のデータストリームにおける別のイベントの発生とを構成し得る。

【0008】

別の実施形態に従うと、1つ以上のアプリケーションに関する多重入力データストリームにわたってパターンを検出するためのシステムが開示される。当該システムは、複数の命令を格納するためのメモリと、メモリにアクセスするように構成されたプロセッサとを含む。一実施形態においては、プロセッサは、第1の入力データストリームおよび第2の入力データストリームを識別する連続クエリを受取るようにとの命令を実行するように構成される。プロセッサは、第1の入力データストリームの第1の属性についての第1のダイナミックデータタイプと、第2の入力データストリームの第2の属性についての第2のダイナミックデータタイプとを識別するようにとの命令を実行するようにさらに構成される。いくつかの実施形態においては、プロセッサは、第1のダイナミックデータタイプお

10

20

30

40

50

よび第2のダイナミックデータタイプに基づいて結合データストリームを生成し、結合データストリームに対して連続クエリを実行してパターンを検出するように構成される。

【0009】

いくつかの実施形態に従うと、1つ以上のプロセッサによって実行可能なコンピュータ実行可能命令を格納する非一時的なコンピュータ読取可能媒体が開示される。コンピュータ実行可能命令は、1つ以上のプロセッサに、第1の入力データストリームおよび第2の入力データストリームを少なくとも含む複数の入力データストリームを受取らせるための命令を含む。コンピュータ実行可能命令はさらに、1つ以上のプロセッサに、第1の入力データストリームについての第1のダイナミックデータタイプおよび第2の入力データストリームについての第2のダイナミックデータタイプを生成させるための命令を含む。いくつかの実施形態においては、コンピュータ実行可能命令は、第1の入力データストリームと第2の入力データストリームとを結合して、第1のダイナミックデータタイプおよび第2のダイナミックデータタイプに基づいて結合データストリームを生成し、結合データストリームに対する連続クエリを処理して、パターンを検出するための命令を含む。

10

【0010】

いくつかの実施形態に従うと、第1の入力データストリームおよび第2の入力データストリームを少なくとも含む複数の入力データストリームを受取るステップと、第1の入力データストリームについての第1のダイナミックデータタイプを生成するステップと、第2の入力データストリームについての第2のダイナミックデータタイプを生成するステップと、第1の入力データストリームと第2の入力データストリームとを結合して、第1のダイナミックデータタイプおよび第2のダイナミックデータタイプに少なくとも部分的に基づき結合データストリームを生成するステップと、結合データストリームに対する連続クエリを処理してパターンを検出するステップとを含む。

20

【0011】

いくつかの実施形態においては、第1のダイナミックデータタイプを生成するステップはさらに、第1の入力データストリームの第1の属性を、第2のデータストリームには存在しないものとして識別するステップと、第1の属性についての第1のダイナミックデータタイプを生成するステップとを含む。第1のダイナミックデータタイプは、第1の入力データストリームの第1の属性に対応する第1のデータ値を格納するように構成される。

【0012】

いくつかの実施形態においては、第2のダイナミックデータタイプを生成するステップはさらに、第2の入力データストリームの第2の属性を、第1のデータストリームには存在しないものとして識別するステップと、第2の属性についての第2のダイナミックデータタイプを生成するステップとを含む。第2のダイナミックデータタイプは、第2の入力データストリームの第2の属性に対応する第2のデータ値を格納するように構成される。

30

【0013】

いくつかの実施形態においては、当該方法はさらに、共通属性を識別するステップを含む。共通属性は、第1の入力データストリームに存在し、かつ第2の入力データストリームに存在するものとして識別される。当該方法はさらに、同種スキーマを生成するステップを含む。同種スキーマは、第1の入力データストリームおよび第2の入力データストリームの1つ以上の属性の表現を含み、当該表現は、少なくとも、共通属性、第1のダイナミックデータタイプおよび第2のダイナミックデータタイプを含む。当該方法はさらに、同種スキーマに少なくとも部分的に基づき結合データストリームを生成するステップを含む。

40

【0014】

いくつかの実施形態においては、当該方法はさらに、結合データストリームの分析に少なくとも部分的に基づきパターンを検出するステップを含む。パターンは、第1の入力データストリームにおける第1のイベントと、後続する第2の入力データストリームにおける第2のイベントとを識別する。

【0015】

50

いくつかの実施形態に従うと、サービスプロバイダデバイスが提供される。サービスプロバイダデバイスは、第1の入力データストリームおよび第2の入力データストリームを少なくとも含む複数の入力データストリームを受取るように構成された入力データストリーム受取ユニットと、第1の入力データストリームについての第1のダイナミックデータタイプを生成するように構成された第1のダイナミックデータタイプ生成ユニットと、第2の入力データストリームについての第2のダイナミックデータタイプを生成するように構成された第2のダイナミックデータタイプ生成ユニットと、第1の入力データストリームと第2の入力データストリームとを結合して、第1のダイナミックデータタイプおよび第2のダイナミックデータタイプに少なくとも部分的に基づく結合データストリームを生成するように構成された結合データストリーム生成ユニットと、結合データストリームに対する連続クエリを処理してパターンを検出するように構成されたパターン検出ユニットとを備える。

10

【0016】

いくつかの実施形態においては、第1のダイナミックデータタイプ生成ユニットはさらに、第1の入力データストリームの第1の属性を第2のデータストリームに存在しないものとして識別し、第1の属性についての第1のダイナミックデータタイプを生成するように構成される。

【0017】

いくつかの実施形態においては、第1のダイナミックデータタイプは、第1の入力データストリームの第1の属性に対応する第1のデータ値を格納するように構成される。

20

【0018】

いくつかの実施形態においては、第2のダイナミックデータタイプ生成ユニットはさらに、第2の入力データストリームの第2の属性を第1のデータストリームには存在しないものとして識別し、第2の属性についての第2のダイナミックデータタイプを生成するように構成される。第2のダイナミックデータタイプは、第2の入力データストリームの第2の属性に対応する第2のデータ値を格納するように構成される。

【0019】

いくつかの実施形態においては、サービスプロバイダデバイスはさらに、共通属性識別ユニットを含み得る。共通属性識別ユニットは、共通属性を識別するように構成される。共通属性は、第1の入力データストリームに存在し、かつ第2の入力データストリームに存在するものとして識別される。サービスプロバイダデバイスはさらに、同種スキーマを生成するように構成された同種スキーマ生成ユニットを含む。同種スキーマは、第1の入力データストリームおよび第2の入力データストリームの1つ以上の属性の表現を含む。当該表現は、少なくとも共通属性、第1のダイナミックデータタイプおよび第2のダイナミックデータタイプを含む。

30

【0020】

いくつかの実施形態においては、同種スキーマは、ストリーム名識別子属性、第1の入力データストリームに関連付けられた第1のタイムスタンプ属性、または第2の入力データストリームに関連付けられた第2のタイムスタンプ属性のうち少なくとも1つを含む。

【0021】

いくつかの実施形態においては、結合データストリーム生成ユニットはさらに、同種スキーマによって識別される第1の入力データストリームからタブルの第1のセットを選択し、同種スキーマによって識別される第2の入力データストリームからタブルの第2のセットを選択し、タブルの第1のセットおよびタブルの第2のセットに対するサブクエリを処理して結合データストリームを生成するように構成されてもよい。

40

【0022】

いくつかの実施形態においては、パターンは、結合データストリームの分析に少なくとも部分的に基づいて検出されてもよい。パターンは、第1の入力データストリームにおける第1のイベントと、後続する第2の入力データストリームにおける第2のイベントとを識別する。

50

【 0 0 2 3 】

図面の簡単な説明

添付の図面に関連付けて詳細な説明を行う。図においては、参照番号の左端の数字は、その参照番号が最初に現われる図番号を示している。さまざまな図において同じ参照番号を用いる場合、類似または同一のアイテムを示している。

【図面の簡単な説明】

【 0 0 2 4 】

【図 1】多重入力データストリームにわたってパターンを識別するための技術が実現され得る簡略化された例示的なシステムまたはアーキテクチャを示す図である。

【図 2】多重入力データストリームにわたるパターンの検出についての特徴が記載され得る簡略化されたブロック図を示す。

【図 3】本開示の一実施形態に従って、多重入力データストリームを識別する C Q L クエリを用いてパターン認識を実行する具体例を示す図である。

【図 4】本開示の別の実施形態に従って、多重入力データストリームを識別する C Q L クエリを用いてパターン認識を実行する具体例を示す図である。

【図 5】本開示の一実施形態に従って、多重入力データストリームにわたってパターンを検出するためのプロセスを示す高レベルフローチャートである。

【図 6】本開示の一実施形態に従って、同種スキーマを生成するためのプロセスを示す高レベルフローチャートである。

【図 7】本開示の実施形態を援用し得るイベント処理システムを示す簡略化された高レベル図である。

【図 8】実施形態のうちの 1 つを実現するための分散型システムを示す簡略図である。

【図 9】本開示の実施形態に従って、具体例としてのシステムの 1 つ以上の構成要素によって提供されるサービスがクラウドサービスとして提供され得るシステム環境の 1 つ以上の構成要素を示す簡略化されたブロック図である。

【図 10】本発明のさまざまな実施形態が実現され得る例示的なコンピュータシステムを示す図である。

【図 11】本発明のさまざまな実施形態が実現され得る例示的なサービスプロバイダデバイスを示す簡略化されたブロック図である。

【発明を実施するための形態】

【 0 0 2 5 】

詳細な説明

以下の説明においては、さまざまな実施形態が記載される。それら実施形態を十分に理解できるようにするために具体的な構成および詳細が説明のために記載される。しかしながら、これらの実施形態が具体的な詳細なしで実施され得ることも当業者には明らかになるだろう。さらに、周知の特徴は、記載されている実施形態が不明瞭にならないようにするために省略または簡素化される可能性がある。

【 0 0 2 6 】

いくつかのアプリケーションにおいては、データは、有限の格納されたデータ集合ではなく、連続的な無限のデータストリームの形を採る可能性がある。このようなデータストリームの例には、金融分野における株価表示機、ネットワーク監視およびトラフィック管理における性能測定、ウェブトラッキングおよびパーソナル化におけるログレコードまたはクリックストリーム、センサアプリケーションからのデータ供給、ファイアウォールベースのセキュリティにおけるネットワークパケットおよびメッセージ、電気通信における通話記録、などが含まれ得る。これらの連続的な性質があるために、これらのデータストリームは、典型的には、従来の 1 回限りの S Q L クエリではなく、連続クエリを用いて照会され得る。

【 0 0 2 7 】

概して、連続データストリーム（イベントストリームとも称される）は、本質的に明確な終わりが無い連続的または無限であり得るデータまたはイベントのストリームを含み得

10

20

30

40

50

る。論理的には、イベントまたはデータストリームは、一連のデータエレメント（またイベントとも称される）であってもよく、各々のデータエレメントは関連付けられたタイムスタンプを有し得る。連続的なイベントストリームは、論理的には、1塊または1セットのエレメント（ s , T ）として表わされてもよい。この場合、「 s 」はデータ部分を表わし、「 T 」は時間ドメインにある。「 s 」部分は概してタプルまたはイベントと称される。このため、イベントストリームは、一連のタイムスタンプ付きタプルまたはイベントであり得る。

【0028】

いくつかの局面においては、ストリームにおけるイベントに関連付けられたタイムスタンプはクロックタイムと等しい可能性がある。しかしながら、他の例においては、イベントストリームにおけるイベントに関連付けられた時間は、アプリケーションドメインによって規定される可能性があり、クロックタイムに対応しない可能性があるが、たとえば、代わりに一連の数字によって表わされる可能性もある。したがって、イベントストリームにおけるイベントに関連付けられた時間情報は、数、タイムスタンプ、または時間の概念を表わす他の情報によって表わされる可能性がある。入力イベントストリームを受取るシステムの場合、イベントは、タイムスタンプが増加する順序でシステムに到達する。同じタイムスタンプを備えた2つ以上のイベントが存在する可能性もある。

10

【0029】

いくつかの例においては、イベントストリームにおけるイベントは、（たとえば、温度センサが値を新しい値に変更したとき、株式銘柄の価格が変化するときなどに）いくつかの世界的なイベントが発生したことを表わしてもよく、イベントに関連付けられた時間情報は、データストリームイベントによって表わされる世界的なイベントがいつ発生したかを示してもよい。

20

【0030】

イベントストリームを介してイベントが受取られた場合、イベントに関連付けられた時間情報を用いて、タイムスタンプ値が増大する順序で、イベントストリームにおけるイベントが到達することを確実にしてもよい。これにより、イベントストリームにおいて受取られたイベントを、それらの関連付けられた時間情報に基づいて順序付けることが可能になり得る。この順序付けを可能にするために、タイムスタンプは、後に生成されるイベントが、以前に生成されたイベントよりも後のタイムスタンプを有するように非減少の態様で、イベントストリームにおけるイベントに関連付けられてもよい。別の例として、順序番号が時間情報として用いられている場合、後に生成されたイベントに関連付けられた順序番号が、以前に生成されたイベントに関連付けられた順序番号より大きくなる可能性がある。いくつかの例においては、たとえば、データストリームイベントによって表わされる世界的なイベントが同時に発生したときに、複数のイベントが同じタイムスタンプまたは順序番号に関連付けられてもよい。同じイベントストリームに属するイベントは、概して、関連付けられた時間情報によってイベントに課された順序で処理される可能性があり、先行のイベントが後のイベントよりも前に処理される。

30

【0031】

イベントストリームにおけるイベントに関連付けられた時間情報（たとえばタイムスタンプ）は、ストリームのソースによって設定されてもよく、または、代替的には、ストリームを受取るシステムによって設定されてもよい。たとえば、いくつかの実施形態においては、イベントストリームを受取るシステム上で心臓の鼓動が維持されてもよく、イベントに関連付けられた時間は、心臓の鼓動によって測定されるようなシステムへのイベントの到着時間に基づいてもよい。イベントストリームにおける2つのイベントは同じ時間情報を有することができる。なお、タイムスタンプの順序付けが1つのイベントストリームにとって特有であるが、さまざまなストリームのイベントが任意にインターリーブされることにも留意されたい。

40

【0032】

イベントストリームは、関連付けられたスキーマ「 S 」を有してもよい。スキーマは、

50

時間情報と1つ以上の名前付き属性のセットとを含み得る。特定のイベントストリームに属するすべてのイベントは、その特定のイベントストリームに関連付けられたスキーマに合致する。したがって、イベントストリーム (s , T) の場合、当該イベントストリームは、 (<time_stamp> , <attribute(s)>) としてスキーマ「 S 」を有し得る。この場合、 <attributes> はスキーマのデータ部分を表わし、1つ以上の属性を含み得る。たとえば、株価表示機のイベントストリームについてのスキーマは、属性 <stock symbol> および <stock price> を含み得る。このようなストリームを介して受取られたイベントは各々、タイムスタンプおよび2つの属性を有することとなる。たとえば、株価表示機イベントストリームは以下のイベントおよび関連付けられたタイムスタンプを受取ってもよい：

...

```
( <timestamp_N> , <NVDA,4> )
( <timestamp_N+1> , <ORCL,62> )
( <timestamp_N+2> , <PCAR,38> )
( <timestamp_N+3> , <SPOT,53> )
( <timestamp_N+4> , <PDCO,44> )
( <timestamp_N+5> , <PTEN,50> )
```

...

上述のストリームにおいては、ストリームエレメント (<timestamp_N+1> , <ORCL,62>) の場合、イベントは、属性「stock_symbol」および「stock_value」を備えた <ORCL,62> である。ストリームエレメントに関連付けられたタイムスタンプは「timestamp_N+1」である。このため、連続的なイベントストリームはイベントのフローとなり、各々のイベントは同じ系列の属性を有する。

【 0 0 3 3 】

上述のように、ストリームは、CQLクエリが作用し得る主要なデータソースであってもよい。ストリーム S は、 (「マルチセット」とも称される) 一塊のエレメント (s , T) であってもよい。この場合、「 s 」は S のスキーマにあり、「 T 」は時間ドメインにある。付加的には、ストリームエレメントはタプルのタイムスタンプ対であってもよく、これは、一連のタイムスタンプ付きタプル挿入として表わすことができる。言い換えれば、ストリームは、一連のタイムスタンプ付きタプルであってもよい。場合によっては、同じタイムスタンプを備えた2つ以上のタプルがあってもよい。さらに、入力データストリームのタプルは、タイムスタンプが増大する順序でシステムに到達するように要求されてもよい。代替的には、 (「時間可変関係」とも称され、リレーショナルデータベースからのデータを含み得る「リレーショナルデータ」と混同されるべきではない) 関係は、時間ドメインから、スキーマ R のタプルからなる無制限の塊までのマッピングであってもよい。いくつかの例においては、関係は、無秩序な時間依存型タプルの塊 (すなわち瞬間的な関係) であってもよい。場合によっては、時間の各々のインスタンスにおいては、関係は有限のセットであってもよい。これはまた、関係の変更状態を把握するために、挿入、削除および/または更新を含み得る一連のタイムスタンプ付きタプルとして表わすことができる。ストリームと同様に、関係は、関係の各タプルが準拠し得る一定のスキーマを有し得る。さらに、連続クエリは、この明細書中において用いられる場合、概して、ストリームおよび/または関係の (すなわち、ストリームおよび/または関係について照会された) データを処理することができる可能性がある。付加的には、関係は、ストリームのデータを参照してもよい。

【 0 0 3 4 】

いくつかの例においては、ビジネスインテリジェンス (business intelligence : B I) は、特定の間隔で (たとえば、場合によって毎日) ビジネス活動を促進および最適化するのに役立ち得る。このタイプの B I は、通常、オペレーショナルビジネスインテリジェンス、リアルタイムビジネスインテリジェンスまたはオペレーショナルインテリジェンス (operational intelligence : O I) と称される。オペレーショナルインテリジェンスは、いくつかの例においては、B I とビジネス活動監視 (business activity monitoring :

10

20

30

40

50

B A M) との間の境界を曖昧にする。たとえば、B I は履歴データの周期的な問合せに焦点が合わされる可能性があり、そのため、後方重視の焦点を有する可能性がある。しかしながら、B I はまた、オペレーショナルアプリケーションに分類されてもよく、したがって、単なる戦略上の分析ツールからビジネスオペレーションにおける最前線にまで拡大する可能性がある。そのため、B I システムはまた、イベントストリームを分析して、総計をリアルタイムで計算するように構成されてもよい。

【 0 0 3 5 】

いくつかの例においては、連続的なクエリ言語サービス (C Q (continuous query) サービス) は、連続クエリを処理してリアルタイムアラートを使用可能にするために B I 解析サーバを拡張するように構成されてもよい。C Q サービスによって、いくつかの局面においては、B I 解析サーバおよび C Q L エンジンとが統合される可能性がある。単なる一例として、B I 解析サーバは、C Q サービスに連続クエリを委ねてもよく、C Q サービスはまた、C Q L エンジンのための論理データベース (database : D B) ゲートウェイとして機能してもよい。このようにして、C Q L エンジンは、その解析能力およびセマンティックモデリングのために B I 解析サーバを活用可能であり得る。

10

【 0 0 3 6 】

いくつかの例においては、C Q サービスは、特に以下の機能を提供し得る：

- C Q L エンジンゲートウェイとしての、B I 解析サーバのための遠隔操作サービス；
- イベントソース / シンクアダプタ；
- C Q L 拡張に加えて論理 S Q L からデータ定義言語 (data definition language : D D L) を生成；
- すべてのタイプの連続クエリおよびインプリメンテーション選択のために一体化モデルを提供；
- メタデータの維持、再開可能性のサポート；ならびに、
- 高可用性およびスケーラビリティのサポート。

20

【 0 0 3 7 】

付加的には、いくつかの例においては、O I は、可視性および洞察力をビジネスオペレーションに与えることができるリアルタイムのダイナミックなビジネス解析の一形式である。O I は、しばしば、B I もしくはリアルタイム B I にリンクされるか、または B I もしくはリアルタイム B I と比較される。というのも、これら B I およびリアルタイム B I がともに両方が大量の情報を把握するのに役立つからである。但し、いくつかの基本的な違いがある。すなわち、O I は主としてアクティビティ中心であり得るのに対して、B I は主としてデータ中心であり得る。加えて、従来から、パターンを識別するために事後ベースかつレポートベースのアプローチとして用いられる可能性のある B I とは異なり、O I は、開発状況 (たとえば、トレンドおよびパターン) を検出してこれに対応するのにより適切であり得る。

30

【 0 0 3 8 】

いくつかの例においては、ビジネスイベント分析および監視 (business event analysis and monitoring : B E A M) システムが、機内データの処理および / または受取りのために C Q L エンジンを含んでもよい。たとえば、C Q L エンジンは、受信したリアルタイム情報 (たとえば B I または O I) を照会または処理するように構成されたインメモリ・リアルタイムイベント処理エンジンであってもよい。C Q L エンジンは、時間的なセマンティックを利用または理解し、データのウィンドウの定義を処理することを可能にするように構成されてもよい。C Q L エンジンを利用する場合、場合によっては、受信データ上でクエリを常に行うことが必要になるかもしれない。

40

【 0 0 3 9 】

いくつかの局面においては、C Q L エンジンは十分に発達したクエリ言語を含んでもよい。そのため、ユーザは照会の観点から演算を指定してもよい。加えて、C Q L エンジンは、クエリ言語特徴、オペレータ共有、豊富なパターンマッチング、豊富な言語構築などを利用して、メモリを最適化するように設計されてもよい。加えて、いくつかの例におい

50

ては、CQLエンジンは履歴データおよびストリーミングデータの両方を処理してもよい。たとえば、カリフォルニアでの売上高が一定の目標を超えた場合、ユーザはアラートを送信するようにクエリを設定することができる。このため、いくつかの例においては、アラートは、履歴上の売上データならびに受信した実際の（すなわち、リアルタイムの）売上データに少なくとも部分的に基づいている可能性がある。

【0040】

いくつかの例においては、CQLエンジンまたは以下に記載される概念の他の特徴が、履歴コンテキスト（すなわち倉庫データ）を受信データとリアルタイムで結合するように構成されてもよい。これにより、場合によっては、本開示では、データベースに格納された情報および機内情報の範囲が説明され得る。データベースに格納された情報および機内情報とともにBIデータを含み得る。このため、データベースは、いくつかの例においては、BIサーバであってもよく、または、如何なるタイプのデータベースであってもよい。さらに、いくつかの例においては、本開示の特徴は、ユーザがコードをプログラムするかまたは書込む方法を知らなくても、上述の特徴の実施を可能にし得る。言い換えれば、これらの特徴は、特徴を豊富に含むユーザインターフェイス（user interface：UI）において提供されてもよく、または、非デベロッパがリアルタイムデータと履歴データとの結合を実現することを可能にする他の態様で提供されてもよい。

10

【0041】

いくつかの実施形態においては、連続データストリームにおいて受取られたイベントは、データストリームにおける特定のパターンの発生を検出するために、実行時間中に処理されてもよい。「パターン」は、連続データストリームにおいて、各々がいくつかの条件を満たす一連の連続したイベントまたはタプルを構成してもよい。一例として、株の取引高の変化などのイベントが発生すると、結果として、株式の価格の変化などの別のイベントが発生し、金融分野に関する「株のティック」イベントを受取る連続データストリームにおいて或る「パターン」を構成し得る。

20

【0042】

多重連続データストリームの文脈においては、「パターン」は、第1のデータストリームにおける第1のイベントの発生、後続する第2のデータストリームにおける別のイベントの発生などを構成し得る。一例として、第1の一連のイベントによって促進される第1のビジネスプロセスについて検討する。この場合、これらイベントはレンタカーの予約アプリケーションに関するものである。同様に、第2の一連のイベントによって促進される第2のビジネスプロセスを検討する。この場合、イベントは飛行機の予約アプリケーションに関するものである。加えて、第1の一連のイベントが第1の連続入力データストリームを介して受取られ、第2の一連のイベントが第2の連続入力データストリームを介して受取られると想定する。このような状況においては、第1の連続入力データストリームにおけるイベント（たとえばユーザによる自動車の予約）が到着し、その後、第2の連続入力データストリームにおける別のイベント（たとえば、ユーザへの飛行チケットの発行）が到着したときに、パターンマッチを出力することが所望され得る。

30

【0043】

一実施形態においては、多重連続入力データストリームにわたるパターンマッチングは、受信した入力データストリームに連続クエリ（たとえばCQLクエリ）を適用することによって実行されてもよい。ある1つのアプローチにおいては、中間ストリームに対応するビューを規定する結果とのパターンマッチングが実行されるべき関連入力データストリームのうち、すべてのUNIONまたはすべてのサブセットを最初に行うことによって、多重連続入力データストリームにわたるパターンマッチングが実行されてもよい。次いで、マッチングされるべきパターンはこの単一の中間ストリームにわたって特定することができる。一実施形態においては、CQL言語構築MATCH_RECOGNIZE文節は、多重入力データストリームを識別するCQLクエリにおいてパターン認識を実行するために用いられてもよい。次いで、パターンを、ビューに含まれるすべてのストリームにマッチングさせてもよい。

40

50

【 0 0 4 4 】

一例として、以下のCQLクエリ「Q1」を検討する。Q1は、第1の入力データストリームS1および第2の入力データストリームS2にわたってマッチングされるべきパターンを指定する連続クエリである。

【 0 0 4 5 】

```

Q1:
SELECT *
FROM
  (SELECT clientId, p1, -1L as p2 from S1 UNION ALL SELECT clientId, -1L as p1,
  p2 from S1) AS S MATCH_RECOGNIZE
  (
    PARTITION BY clientId
    MEASURES
      A.p1 as p1,
    B.P2 AS P2
    PATTERN (A b)
    DEFINE
      A as (P1 !=-1L)
      b as (P2 !=-1L)
  )

```

10

20

加えて、ストリームS1が第1のスキーマによって規定されると想定する：S1 (int p1, int clientId)。さらに、ストリームS2が第2のスキーマによって規定されると想定する：S2 (int p2, int clientId)。この場合、p1、p2およびclientIdは、ストリームS1およびストリームS2の1つ以上の属性に対応する。

【 0 0 4 6 】

クエリQ1は、ストリームS1およびストリームS2のUNIONを指定するFROM分節を含む。UNION ALLクエリにより、ストリームS1およびストリームS2の結果セットを結合するために、CQL UNION ALLクエリ内のCQL SELECTステートメントは各々、典型的には、類似のデータタイプをもつ結果セットにおいて同数のフィールドを有することが必要とされる。上述のクエリQ1の例においては、ストリームS1のスキーマはストリームS2のスキーマとは異なる。1つのアプローチにおいては、ストリームS1およびストリームS2は、ストリームS1およびストリームS2のスキーマ同士を正規化することによって結合され得る。

30

【 0 0 4 7 】

一例においては、スキーマの正規化は、ストリームS1およびストリームS2のスキーマの各々に付加的なカラムを追加することによって実行されてもよい。上述のクエリQ1の例においては、カラムは、ストリームS1およびストリームS2のスキーマの各々に追加され、「-1L」などのハードコード化された値でポピュレートされてスキーマを正規化する。しかしながら、各々のストリームにハードコード化されたデータ値を導入することは、エラーが起こり易い手動プロセスである。なぜなら、異なるスキーマで多重入力データストリームを識別するCQLクエリが処理されるたびに、ハードコード化された値を入力しなければならないからである。

40

【 0 0 4 8 】

本開示の一実施形態においては、1つ以上の関与する入力データストリームを表わす同種スキーマが生成される。いくつかの実施形態においては、同種スキーマは、関与する入力データストリームの1つ以上の属性についての1つ以上のダイナミックデータタイプを作成することによって生成されてもよい。ダイナミックデータタイプおよび同種スキーマを生成することにより、UNION ALLサブクエリ内の各々のSELECTステートメントが、同数のフィールドおよび類似するデータタイプを含むことが可能となり、結果

50

として、ストリーム S 1 およびストリーム S 2 の結果セットが SELECT ステートメントを用いて結合され得る。加えて、関与する入力データストリームの属性についてのダイナミックデータタイプの生成はリアルタイムで実行され、CQL クエリが処理されるたびに、ハードコード化された値を関与する入力データストリームに導入する必要がなくなる。一例においては、以下に詳細に説明されるように、ダイナミックデータタイプは、入力データストリームの 1 つ以上の属性について識別された複合データタイプを参照する可能性もある。

【 0 0 4 9 】

いくつかの実施形態においては、次いで、入力データストリームが結合されて、同種スキーマに基づいて結合データストリームが生成されてもよい。次いで、入力データストリームにわたるパターンを検出するために、連続クエリが結合データストリームにわたって処理されてもよい。一実施形態においては、多重入力データストリームを識別する CQL クエリでパターン認識を実行するために、CQL 言語構築 MATCH__RECOGNIZE 分節が用いられてもよい。多重入力データストリームを識別する CQL クエリにおけるパターン認識を実行するために同種スキーマおよびダイナミックデータタイプが生成および利用され得る態様の付加的な詳細が、以下の図 1 ~ 図 4 において詳細に説明される。

10

【 0 0 5 0 】

MATCH__RECOGNIZE 分節を用いると、ユーザは、相関変数と称される識別子を用いることによって、受信したイベントの属性に関する条件を規定し、パターンマッチングについての条件を特定することができる。上述のとおり、入力データストリームにおける一連の連続イベントまたはタプルは、各々がいくつかの条件を満たすものであって、パターンを構成している。パターン認識機能は、ユーザが、受信したイベントまたはタプルの属性についての条件を規定することと、相関変数と称される文字列名を用いることによってこれらの条件を識別することとを可能にする。

20

【 0 0 5 1 】

上述のクエリ Q 1 においては、「A」および「B」は相関変数である。マッチングさせるべきパターンは、これらの相関変数に対する正規表現として指定され、有効な一致として認識されるべきさまざまな受信イベントによって条件が満たされるべき順番または順序を決定する。これらの条件を満たす入力データストリームにおける一連の連続イベントがパターンを構成する。一実施形態においては、MATCH__RECOGNIZE クエリの出力はストリームである。上述のクエリ Q 1 においては、MATCH__RECOGNIZE 分節はまた、いくつかのサブ文節を含む。

30

【 0 0 5 2 】

DEFINE サブ文節は、各々の相関変数についてのブール条件を指定する。これは、如何なる論理式または演算式として指定されてもよく、条件と一致するイベントの属性に対して如何なる単一行関数または集約関数を適用してもよい。入力データストリームを介して新しいイベントを受取ると、その時点で関連している相関変数の条件が評価される。イベントは、それがその規定条件を満たしている場合には相関変数と一致したものとみなされる。特定の入力値は 0、1 またはそれ以上の相関変数と一致し得る。入力イベントを受取った際に評価されるべき関連条件は、PATTERN 分節正規表現によって管理される処理ロジックと、それより前の入力を処理した後に達するパターン認識プロセスにおける状態とによって決定される。条件は、MATCH__RECOGNIZE 分節が適用されているストリームを評価するストリームまたはビューのスキーマの属性のうちのいずれを指してもよい。PATTERN 分節における相関変数は、DEFINE 分節において指定される必要はない。このような相関変数についてのデフォルトは常に真である叙述部となる。このような相関変数はすべてのイベントと一致する。

40

【 0 0 5 3 】

PARTITION BY サブ文節はストリーム属性を指定するが、このストリーム属性に基づいて、MATCH__RECOGNIZE 分節がその結果を分割するはずである。PARTITION BY 分節がない場合、すべてのストリーム属性は同じ区画に属する

50

。PARTITION BY分節がパターンマッチングと共に存在する場合、入力データストリームは、論理的には、区画リストにおいて記載される属性に基づいて分割され、パターンマッチングが区画内においてなされる。上述のクエリQ1においては、(ストリームS1およびストリームS2に共通の属性である)「clientId」属性が、PARTITION BY分節において指定され、これにより、MATCH_RECOGNIZE分節がその結果を分割し得る。

【0054】

MEASURESサブ文節は、指定されたパターンとうまく一致するイベントの1つ以上の属性値をエクスポートし(たとえば、SELECT分節に含めることを可能にし)、さらに、それらの属性値について表現を指定することを可能にする。この分節を用いることにより、DEFINE分節における条件(相関変数)と一致するイベントストリームにおけるイベントの属性についての表現を規定し、これらの表現を別名で記載し、これにより、このMATCH_RECOGNIZE条件が一部をなす主クエリのSELECT分節においてこれらの表現を好適に用いることができるようにし得る。イベントストリームの属性は、直接的または相関変数を介して参照され得る。

10

【0055】

PATTERNサブ文節は、1つ以上の相関変数に関する正規表現として一致させるべきパターンを指定する。受信イベントは与えられた順序で(左から右へと)これらの条件と一致するはずである。正規表現は、以下のような相関変数およびパターン修飾子から構成され得る:

20

* : 0回以上

+ : 1回以上

? : 0回または1回など。

【0056】

いくつかの実施形態においては、上述の1文字パターン修飾子は最大または「グリーディ(greedy)」であり、これらは、それらが適用される正規表現のうち可能な限り多くのインスタンスと一致するよう企図されるだろう。パターン修飾子はまた2文字であって、最小または「リラクタント(reluctant)」であり、それらが適用される正規表現のうち可能な限り少数のインスタンスと一致するよう企図されるだろう。2文字の修飾子の例は、以下を含むが、それらに限定されない。

30

【0057】

*? : 0回以上

+? : 1回以上

?? : 0回または1回。

【0058】

パターンマッチングの例として、以下のパターンを検討する:

パターン(AB* C)

このパターン分節は、以下の条件が連続的な受信入力イベントによって満たされたときにパターンマッチが認識されるだろうことを意味する:

状態1: 1つのイベントタプルが、相関変数Aを規定する条件と一致し、その後、

40

状態2: ゼロ以上のタプルが相関変数Bと一致し、その後、

状態3: 1つのタプルが相関変数Cと一致する。

【0059】

状態1、状態2および状態3を含む状態は、パターン(AB* C)についてのさまざまな状態を表わし得る。状態3はパターンについての最終状態である。パターンマッチが特定の状態であり、同じ特定の状態のままであり得るか、または、次のイベントのために特定の状態から次の状態まで移行し得る場合、これは、バイndingが大きくなる可能性があることを示唆している可能性がある。パターンは、バイndingが最終状態である場合に一致したとみなされてもよい。状態2である間、相関変数BおよびCの両方と一致するタプルまたはイベントが到達する場合(これによって、これら相関変数BおよびCの

50

両方の規定条件が満たされるので)、Bについての修飾子*がグリーディであれば、そのタプルはCではなくBと一致したとみなされる可能性がある。したがって、グリーディの特性があるために、BはCよりも優先され、より多くのBが一致する可能性がある。パターン表現が $AB^*C?$ である場合、Bに対してレイジーまたはリラクタントな修飾子が用いられ、さらに、BおよびCの両方と一致するタプルはCとしか一致しないものとして処理される可能性がある。このため、この例においては、CはBよりも優先される可能性があり、一致させるBの数がより少なくなる可能性がある。

【0060】

上述のクエリQ1においては、パターン(AB)は以下の場合に一致する：

状態1：1つのイベントタプルが相関変数Aを規定する条件と一致し、その後、

状態2(最終状態)：1つのタプルが相関変数Bと一致する。

10

【0061】

状態1および状態2などの状態は、パターン(AB)についてのさまざまな実現可能な状態を表わし、状態2はパターンについての最終状態である。

【0062】

上述および以下の技術は、いくつかの方法で、いくつかの文脈において実現され得る。いくつかの実現例および文脈が、添付の図面に関連付けて以下においてより詳細に提供される。しかしながら、以下の実現例および文脈は多数あるうちの数例に過ぎない。

【0063】

図1は、多重入力データストリームにわたってパターンマッチングを実行するための技術が実現され得る簡略化された例示的なシステムまたはアーキテクチャ100を示す。アーキテクチャ100においては、1つ以上のユーザ102(たとえば口座名義人)はユーザコンピューティング装置104(1)~140(N)(集合的に「ユーザ装置104」)を利用して、1つ以上のネットワーク108を介して1つ以上のサービスプロバイダコンピュータ106にアクセスし得る。いくつかの局面においては、サービスプロバイダコンピュータ106はまた、ネットワーク108を介して1つ以上のストリーミングデータソースコンピュータ110および/または1つ以上のデータベース112と通信し得る。たとえば、ユーザ102はサービスプロバイダコンピュータ106を利用して、ストリーミングデータソースコンピュータ110および/またはデータベース112のデータにアクセスするかまたは当該データを管理し得る(たとえば、クエリは110および112のいずれかまたは両方に対して実行され得る)。データベース112はリレーショナルデータベース、SQLサーバなどであってもよく、いくつかの例においては、ユーザ102の代わりに履歴データ、イベントデータ、リレーション、アーカイブされたリレーションなどを管理してもよい。加えて、データベース112は、ストリーミングデータソースコンピュータ110によって提供されるデータを受取るかまたは格納し得る。いくつかの例においては、ユーザ102はユーザ装置104を利用して、クエリ(「クエリステートメント」とも称される)またはデータについての他の要求(たとえば履歴イベントデータ、ストリーミングイベントデータなど)を提供することによってサービスプロバイダコンピュータ106と対話し得る。次いで、このようなクエリまたは要求がサービスプロバイダコンピュータ106によって実行されて、データベース112のデータおよび/またはストリーミングデータソースコンピュータ110からの受信データが処理され得る。さらに、いくつかの例においては、ストリーミングデータソースコンピュータ110および/またはデータベース112は、サービスプロバイダコンピュータ106に関連付けられた一体型の分散環境の一部であってもよい。

20

30

40

【0064】

いくつかの例においては、ネットワーク108は、ケーブルネットワーク、インターネット、ワイヤレスネットワーク、セルラーネットワーク、イントラネットシステム、ならびに/または、他のプライベートネットワークおよび/もしくはパブリックネットワークなどの複数のさまざまなタイプのネットワークのうちのいずれかまたはそれらの組合せを含み得る。図示される例は、ネットワーク108を介してサービスプロバイダコンピュ

50

タ 1 0 6 にアクセスするユーザ 1 0 2 を表わしており、記載された技術は、ユーザ 1 0 2 が、地上通信電話によって 1 つ以上のユーザ装置 1 0 4 を介して、キオスク経由で、または他の方法で、1 つ以上のサービスプロバイダコンピュータ 1 0 6 と対話する場合にも同様に適用され得る。なお、記載された技術が、他のクライアント/サーバ構成（たとえばセットトップボックスなど）において、ならびに非クライアント/サーバ構成（たとえば局所的に格納されたアプリケーションなど）においても同様に適用され得ることに留意されたい。

【 0 0 6 5 】

ユーザ装置 1 0 4 は、携帯電話、スマートフォン、携帯情報端末（personal digital assistant：PDA）、ラップトップコンピュータ、デスクトップコンピュータ、シンクライアントデバイス、タブレットPCなどの如何なるタイプのコンピューティング装置であってもよいが、これらに限定されない。いくつかの例においては、ユーザ装置 1 0 4 は、ネットワーク 1 0 8 を介して、または他のネットワーク接続を介して、サービスプロバイダコンピュータ 1 0 6 と通信してもよい。さらに、ユーザ装置 1 0 4 はまた、データベース 1 1 2（または他のデータストア）のデータの処理を要求するための 1 つ以上のクエリまたはクエリステートメントを供給するように構成されてもよい。

10

【 0 0 6 6 】

いくつかの局面においては、サービスプロバイダコンピュータ 1 0 6 はまた、サーバなどのモバイル、デスクトップ、シンクライアントおよび/もしくはクラウドコンピューティング装置などの如何なるタイプのコンピューティング装置などであってもよいが、これらに限定されない。いくつかの例においては、サービスプロバイダコンピュータ 1 0 6 は、ネットワーク 1 0 8 を介して、または他のネットワーク接続を介してユーザ装置 1 0 4 と通信してもよい。サービスプロバイダコンピュータ 1 0 6 は、サーバファームとして、または互いに関連付けられない個々のサーバとして、場合によってはクラスタに配置される 1 つ以上のサーバを含み得る。これらのサーバは、この明細書中に記載されるような、CQLリレーションの管理、入力リレーションの生成、入力リレーションに関連付けられた構成可能なウィンドウオペレータ、および出力リレーションの生成を含むがこれらに限定されない、この明細書中に記載される特徴を実行またはホストするように構成されてもよい。加えて、いくつかの局面においては、サービスプロバイダコンピュータ 1 0 6 は、ストリーミングデータソースコンピュータ 1 1 0 および/もしくはデータベース 1 1 2 を含む一体型分散コンピューティング環境の一部として構成されてもよい。

20

30

【 0 0 6 7 】

具体的な一構成においては、サービスプロバイダコンピュータ 1 0 6 は、少なくとも 1 つのメモリ 1 1 4 および 1 つ以上の処理ユニット（またはプロセッサ）1 3 4 を含み得る。プロセッサ 1 3 4 は、ハードウェア、コンピュータ実行可能命令、ファームウェアまたはこれらの組合せで、適宜実現され得る。プロセッサ 1 3 4 のコンピュータ実行可能命令またはファームウェア実現例は、記載されたさまざまな機能を実行するために好適なプログラミング言語で書込まれたコンピュータ実行可能命令またはマシン実行可能命令を含み得る。

【 0 0 6 8 】

メモリ 1 1 4 は、これらのプログラムの実行中に生成されるデータに加えて、プロセッサ 1 3 4 上でロード可能かつ実行可能なプログラム命令を格納してもよい。サービスプロバイダコンピュータ 1 0 6 の構成およびタイプに応じて、メモリ 1 1 4 は、揮発性（ランダムアクセスメモリ（random access memory：RAM）など）であってもよく、および/もしくは不揮発性（読取専用メモリ（read-only memory：ROM）、フラッシュメモリなど）であってもよい。サービスプロバイダコンピュータ 1 0 6 またはサーバはまた、リムーバブルストレージおよび/もしくは非リムーバブルストレージを含み得る追加の記憶装置 1 3 6 を含んでもよい。追加の記憶装置 1 3 6 は、磁気記憶装置、光ディスクおよび/もしくはテープ記憶装置を含み得るが、これらに限定されない。ディスクドライブおよびそれらの関連するコンピュータ読取可能媒体は、コンピューティング装置のためにコンピ

40

50

ユーザ読取可能命令、データ構造、プログラムモジュールおよび他のデータを不揮発的に格納し得る。いくつかの実現例においては、メモリ 114 は、スタティックランダムアクセスメモリ (static random access memory : S R A M)、ダイナミックランダムアクセスメモリ (dynamic random access memory : D R A M) または R O M などの複数のさまざまなタイプのメモリを含み得る。

【 0069 】

メモリ 114 および追加の記憶装置 136 はともに、取外し可能かつ取外し不可能であって、コンピュータ読取可能記憶媒体のあらゆる例となる。たとえば、コンピュータ読取可能記憶媒体は、コンピュータ読取可能命令、データ構造、プログラムモジュールまたは他のデータなどの情報を格納するためのいずれかの方法または技術において実現される揮発性または不揮発性の、取外し可能であるかまたは取外し不可能である媒体を含み得る。メモリ 114 および追加の記憶装置 136 はコンピュータ記憶媒体のあらゆる例となる。

10

【 0070 】

サービスプロバイダコンピュータ 106 はまた、サービスプロバイダコンピュータ 106 が、ネットワーク 108 上で、格納されたデータベース、別のコンピューティング装置もしくはサーバ、ユーザ端末および/または他の装置と通信することを可能にする通信接続部 138 を含んでもよい。サービスプロバイダコンピュータ 106 はまた、キーボード、マウス、ペン、音声入力装置、タッチ入力装置、ディスプレイ、1つ以上のスピーカ、プリンタなどの入出力 (input/output : I / O) 装置 140 を含んでもよい。

【 0071 】

メモリ 114 のコンテンツをより詳細に参照すると、メモリ 114 は、オペレーティングシステム 124 と、少なくとも入力データストリームモジュール 116、ダイナミックデータタイプ生成モジュール 118、結合データストリーム生成モジュール 120 およびパターン検知モジュール 122 を含む、この明細書中に開示される特徴を実現するための1つ以上のアプリケーションプログラムまたはサービスとを含み得る。この明細書中において用いられるように、モジュールは、サービスの一部でありサーバまたはサーバのクラスタによって実行されるプログラミングモジュールを指し得る。この特定のコンテキストにおいては、モジュールは、サービスプロバイダコンピュータ 106 の一部であるサーバまたはサーバのクラスタによって実行されてもよい。他の実施形態においては、モジュールは、サービスプロバイダコンピュータ 106 の一部であり得る C Q L エンジンおよび/もしくは (図 2 に示される) C Q サービス 200 によって実行されてもよい。さまざまな実施形態においては、以下に詳細に説明されるように、モジュール 116、118、120 および 122 は、複数の入力データストリームにわたるパターンの検出を可能にする機能を実行するように構成されてもよい。これらのモジュールは、ハードウェアで、ソフトウェアで、またはそれらの組合せで実現され得る。図 1 に示されるさまざまなモジュールは、説明を目的としたものであり、本発明の実施形態の範囲を限定するように意図されたものではない。代替的な実施形態に含まれ得るモジュールは、図 1 に示されるよりも多くてもまたは少なくともよい。

20

30

【 0072 】

一実施形態においては、入力データストリームモジュール 116 は、1つ以上の入力データストリーム 126 を受取るか、識別するか、生成するかまたは提供するように構成され得る。一例においては、入力データストリーム 126 は、1つ以上のアプリケーションに関するデータまたはイベントからなる受信連続データストリームを含み得る。いくつかの実施形態においては、入力データストリーム 126 は、データソースコンピュータ 110 および/もしくはデータベース 112 から受取られてもよい。

40

【 0073 】

いくつかの例においては、入力データストリーム 126 は、1つ以上のアプリケーションに関する一連のタイムスタンプ付きタプルまたはデータ記録を含み得る。たとえば、入力データストリーム 126 におけるデータ記録は各々、以下のスキーマ (< time_stamp > , < attribute(s) >) によって表わされ得るイベントストリームエントリを含んでもよい

50

。 <attributes> はスキーマのデータ部分を表わし、1つ以上の属性を含み得る。

【0074】

一例として、入力データストリーム126が第1のデータストリームおよび第2のデータストリームを含むものとする。この例のために、さらに、第1の入力データストリームが第1のアプリケーションに関するイベントの第1のセット（たとえばレンタカー予約アプリケーション）を含むものとし、第2の入力データストリームが第2のアプリケーションに関するイベントの第2のセット（たとえば飛行機予約アプリケーション）を含むものとする。加えて、第1の入力データストリームが第1のスキーマS1：(c1, 整数, c2, 整数)によって規定されるものとし、第2の入力データストリームが第2のスキーマS2：(c1, 整数, c3, 整数)によって規定されるものとする。

10

【0075】

一実施形態においては、第1の入力データストリームおよび第2の入力データストリームを分析してパターンを検出してもよい。この場合、「パターン」は、第1のデータストリームにおけるイベントの発生と、後続する第2のデータストリームにおける別のイベントの発生とを構成し得る。一例においては、パターンマッチングは、受信した連続データストリームに連続クエリ（たとえばCQLクエリ）を適用することによって実行されてもよい。図2は、多重入力データストリームに対してCQLクエリを適用することによってパターンマッチングを実行するためのCQLエンジンおよび/もしくはCQサービス200のうち少なくとも1つの実現例を記載する。上述の説明は第1のデータストリームおよび第2のデータストリームにわたってパターンを検出することに関連しているが、開示された技術が、少なくともいくつかの実施形態においては、多重連続入力データストリームにわたるパターンの検出に適用され得ることが認識されるはずである。

20

【0076】

いくつかの実施形態においては、入力データストリームの分析は、最初に、ダイナミックデータタイプ生成モジュール118によって実行されてもよい。一実施形態においては、ダイナミックデータタイプ生成モジュール118は、入力データストリームモジュール116から第1の入力データストリームおよび第2の入力データストリームを受取り、入力データストリームにおける1つ以上の属性を識別するように構成されてもよい。いくつかの例においては、ダイナミックデータタイプ生成モジュール118は、第1のデータストリームの第1の属性および第2のデータストリームの第2の属性を識別するように構成されてもよい。一例においては、第1の属性は、第2のデータストリームには存在しない属性として識別されてもよく、第2の属性は、第1のデータストリームには存在しない属性として識別されてもよい。

30

【0077】

上に説明されたデータストリームS1およびデータストリームS2の例に従って、一実施形態においては、ダイナミックデータタイプ生成モジュール118は、第1の属性「c2」を、第2のデータストリームS2には存在しない第1のデータストリームS1における属性として識別してもよい。同様に、ダイナミックデータタイプ生成モジュール118は、第2の属性「c3」を、第1のデータストリームS1には存在しない第2のデータストリームS2における属性として識別してもよい。上述の説明は、ストリームS1およびストリームS2において識別された第1の属性および第2の属性に関するものであるが、他の実施形態においては、ダイナミックデータタイプ生成モジュール118が、ストリームS1およびストリームS2からの付加的な属性を識別するように構成され得ることが認識されるはずである。

40

【0078】

いくつかの実施形態においては、ダイナミックデータタイプ生成モジュール118はさらに、第1の属性についての第1のダイナミックデータタイプおよび第2の属性についての第2のダイナミックデータタイプを生成するように構成されてもよい。一例においては、第1のダイナミックデータタイプは第1のデータストリームの第1の属性に対応する第1のデータ値を格納するように構成され、第2のダイナミックデータタイプは第2のデー

50

タストリームの第2の属性に対応する第2のデータ値を格納するように構成される。

【0079】

いくつかの実施形態においては、ダイナミックデータタイプは、この明細書中において「フィールド」と称される1つ以上のメンバタイプを含む複合データタイプとして実現されてもよい。一例においては、メンバフィールドは名前および関連付けられたタイプを含んでもよい。フィールドタイプは、たとえばCQL拡張可能タイプなどの如何なるCQL固有のタイプであってもよい。いくつかの例においては、サービス（たとえば、図2に示されるCQLエンジンおよび/もしくはCQサービス200）は、ダイナミックデータタイプがそのパブリックフィールド（たとえばフィールド名、フィールドのデータタイプ）を指定することによって作成および規定され得る1つ以上のアプリケーションプログラミングインターフェイス（API：application programming interface）を提供することによって、ダイナミックデータタイプについてのCQL拡張可能タイプを実現するように構成されてもよい。

10

【0080】

いくつかの実施形態においては、ダイナミックデータタイプ生成モジュール118は、第1のデータストリームおよび第2のデータストリームの両方に存在する属性として識別され得る共通属性を識別するように構成されてもよい。一実施形態においては、ダイナミックデータタイプ生成モジュール118はさらに、第1の入力データストリームおよび第2の入力データストリームを表わすために同種スキーマ128を生成するように構成されてもよい。一例においては、同種スキーマ128は、共通属性ならびに第1のダイナミックデータタイプおよび第2のダイナミックデータタイプを含む第1の入力データストリームおよび第2の入力データストリームの1つ以上の属性の表現を含み得る。

20

【0081】

上述の第1の入力データストリームS1および第2の入力データストリームS2の例に従って、一実施形態においては、ストリームS1およびストリームS2についての同種スキーマSが以下のように生成され得る：

同種スキーマS：(\$streamName char, c1 int, S1 Typ1@DynamicTypeCartridge, S2 Typ2@DynamicTypeCartridge)

ここで、「\$streamName」は、同種スキーマにおけるデータタイプチャーを有する属性を指す。「\$streamName」属性は、入力データストリームから受信された入力タプル（イベント）の入力データストリームソースを識別する。イベント「e」について、「\$streamName」属性は、「e」がイベントであるストリームエイリアスの名前を格納してもよい。これにより、「\$streamName」属性は、現在の入力イベント（タプル）が属する特定の入力データストリームの識別を可能にする。「c1」は、同種スキーマにおける共通属性を指す。この場合、共通属性は、関与する入力データストリームすべてにおいて同じデータタイプおよび同じ名前を有する属性として識別される。たとえば、共通属性「c1」は、入力データストリームS1および入力データストリームS2に関する第1のアプリケーションおよび第2のアプリケーションにアクセスするユーザを識別するユーザ属性を指してもよい。いくつかの例においては、共通属性を含む同種スキーマは、PARTITION by分節の規格で、それ自体がさまざまな入力データストリームからのイベントを含み得るサブストリームを作成することを可能にし得る。

30

40

【0082】

加えて、「Typ1@DynamicTypeCartridge」は、第1のデータストリームの第1の属性に対応する第1のデータ値を格納するように構成された第1のダイナミックデータタイプを指す。一例においては、「Typ1@DynamicTypeCartridge」は、第1の入力データストリームS1における第1の属性「c2」に対応するデータ値を格納するために用いられてもよい。一例として、第1の入力データストリームS1における第1の属性「c2」は、第1のアプリケーション（たとえばレンタカー予約アプリケーション）にアクセスするユーザから得られる自動車予約識別番号を格納する自動車予約属性を指し得る。

【0083】

50

同様に、「Typ2@DynamicTypeCartridge」は、第2のデータストリームの第2の属性に対応する第2のデータ値を格納するように構成された第2のダイナミックデータタイプを指す。一例においては、「Typ2@DynamicTypeCartridge」は、第2の入力データストリームS2における第2の属性「c3」に対応するデータ値を格納するために用いられてもよい。一例として、第2の入力データストリームS2における第2の属性「c3」は、第2のアプリケーション（たとえば飛行機予約アプリケーション）にアクセスするユーザのために飛行機予約識別番号を格納する飛行予約属性を指し得る。

【0084】

上述の例の場合、タイプ「Typ1@DynamicTypeCartridge」のS1についての1つの属性（たとえば「c2」）、およびタイプ「Typ2@DynamicTypeCartridge」のS2についての1つの属性（たとえば「c3」）が存在することが確認され得る。「Typ1@DynamicTypeCartridge」は、同じデータタイプのストリームS1の各属性ごとに1つのパブリックフィールドを含み得る。同様に、「Typ2@DynamicTypeCartridge」は、ストリームS2の各属性ごとに1つのパブリックフィールドを含み得る。Typ1およびTyp2が内部タイプであって、エンドユーザにとって可視可能ではないかもしれないことに留意されたい。検討された例に関して、Typ1はパブリックフィールド「c1」および「c2」を含んでもよく、Typ2はパブリックフィールド「c1」および「c3」を含んでもよい。

【0085】

一実施形態においては、このように生成された同種スキーマに基づいて、結合データストリーム生成モジュール120は結合データストリームを生成するように構成され得る。一例においては、結合データストリームは、第1の入力データストリームおよび第2の入力データストリームのUNIONを実行することによって生成されてもよい。この場合、第1の入力データストリームおよび第2の入力データストリームは、同種スキーマSによって表現される。一例においては、CQL UNION ALLサブクエリは、同種スキーマに基づいてストリームS1およびストリームS2の結果セットを結合するために第1のデータストリームS1および第2のデータストリームS2に適用されてもよい。同種スキーマを生成することにより、UNION ALLサブクエリ内の各々のCQL SELECTステートメントが、結果セットにおける同数のフィールドおよび類似のデータタイプを有することが可能となる。したがって、ストリームS1およびストリームS2の結果セットは、同種スキーマに基づいてUNION ALLサブクエリにおけるSELECTステートメントを用いて結合されてもよい。UNION ALLサブクエリを用いて第1のデータストリームと第2のデータストリームとを結合して結合データストリームを生成し得る態様を、図4において説明する。

【0086】

いくつかの実施形態においては、パターン検出モジュール122は、さらに、結合データストリームに基づいてパターン132を検出するように構成されてもよい。一実施形態においては、パターン検出モジュールは、結合データストリームにMATCH_RECOGNIZE分節を適用することによってパターンマッチを出力するように構成されてもよい。一例として、検出されたパターンは、第1の入力データストリームS1における第1のイベント（たとえばユーザによる自動車予約）の到着と、後続する第2の入力データストリームS2における第2のイベント（たとえばユーザへの飛行機チケット発行）の到着とを含み得る。多重入力データストリームを識別するパターン認識をCQLクエリで実行するようにパターン検出モジュールが構成され得る態様を、図3において説明する。

【0087】

図2は、多重入力データストリームにわたってパターンを検出するための特徴が記載され得る簡略化されたブロック図を示す。図示のとおり、図2は、多重入力データストリームを管理するためのCQLエンジンおよび/またはCQサービス200の少なくとも1つの実現例を示す。いくつかの実施形態においては、図1に示されるモジュール116、118、120および122は、CQLエンジンおよび/もしくはCQサービス200によって実行され得る。CQLエンジンおよび/もしくはCQサービス200は、最初に、入

10

20

30

40

50

カソース 204 から情報を受取ってもよい。一例においては、入力ソース 204 は、1つ以上のアプリケーションに関するデータまたはイベントのストリームを含む連続入力データストリームを受取るデータソースコンピュータ 110 を含み得る。いくつかの実施形態においては、第 1 の入力データストリーム S1 206 および第 2 の入力データストリーム S2 208 を含むクエリ（たとえば CQL クエリ）が識別されるかまたは受取られると、CQL エンジン 200 は、ストリーム S1 およびストリーム S2 にわたってパターンを検出するためにクエリをパズし得る。一実施形態においては、CQL エンジンおよび/または CQ サービス 200 は、第 1 の入力データストリームおよび第 2 の入力データストリームの UNION を最初に行うことによって CQL クエリを実行してもよい。この場合、第 1 の入力データストリームおよび第 2 の入力データストリームは同種スキーマ S によって表現される。一実施形態においては、同種スキーマ S は、図 1 において説明されるようなダイナミックデータタイプ生成モジュール 118 によって生成されてもよい。

10

【0088】

次いで、入力データストリーム S1 および入力データストリーム S2 が結合されて、同種スキーマに基づく結合データストリームが生成され得る。一実施形態においては、結合データストリームは結合データストリームモジュール 120 によって生成されてもよい。次いで、パターン検出モジュール 122 が、結合データストリームに対する連続クエリ（たとえば CQL クエリ）を処理して、入力データストリームにわたってパターンを検出し得る。一実施形態においては、パターン検出モジュールは、結合データストリームに対して MATCH__RECOGNIZE 分節を適用することによって、パターンマッチを出力するように構成されてもよい。いくつかの例においては、CQL エンジンおよび/または CQ サービス 200 は、次いで、たとえば図 1 に示されるデータベース 112 におけるような出力宛先 212 において検出されたパターンを格納し得る。

20

【0089】

図 3 は、本開示の一実施形態に従った、多重入力データストリームを識別する CQL クエリを用いてパターン認識を実行する具体例である。一例においては、クエリ 300 は、第 1 の入力データストリーム S1 および第 2 の入力データストリーム S2 にわたって一致させるべきパターンを指定する連続的な（たとえば CQL）クエリである。図 3 に示される例においては、ストリーム S1 およびストリーム S2 は括弧に囲まれてカンマで分離されたリストにおいて指定され、MATCH__RECOGNIZE 分節への入力として与えられる。結果として、クエリ 300 の FROM 分節は以下の式 (S1, S2) MATCH__RECOGNIZE を用いて、好都合に表現され得る。いくつかの例においては、リストにおける各々のストリームは、ベースストリームとして、ストリームに対するビューの評価として、またはストリームに対するサブクエリの評価として識別されてもよい。

30

【0090】

上述のようなカンマで分離されたリストを用いてストリーム S1 およびストリーム S2 が指定される場合、関与するストリームの各々は別個のエイリアスで識別される必要があるかもしれない。そのため、たとえば、MATCH__RECOGNIZE 分節が {S, S} に対して適用される場合、FROM 分節は (S, S AS S1) MATCH__RECOGNIZE) として指定される必要があるかもしれない。すなわち、任意の AS 分節は、第 2 のストリーム S を別名で記載して別個のエイリアス名を確実にするために導入される必要があるかもしれない。なお、明確な AS 分節が指定されない場合、他の例においてストリーム名がエイリアス名とみなされる可能性があることに留意されたい。

40

【0091】

いくつかの実施形態においては、カンマで分離されたリストとしてのストリーム S1 およびストリーム S2 の表現は、入力データストリーム S1 および入力データストリーム S2 の UNION と評価され得る。この場合、ストリーム S1 およびストリーム S2 は同種スキーマによって規定される。一実施形態においては、図 4 において詳細に説明されるように、結合データストリームは、UNION ALL サブクエリの実行に基づいて生成されてもよく、結合データストリームは MATCH__RECOGNIZE 分節への入力とし

50

て供給されてもよい。次いで、MATCH__RECOGNIZEが結合データストリームにおけるパターンを検出するために利用されてもよい。

【0092】

上述のように、MATCH__RECOGNIZE分節を用いると、ユーザは、受信したイベントの属性に関する条件を規定し、相関変数と称される識別子を用いることによってパターンマッチングについての条件を識別することができる。パターン認識機能は、ユーザが、受信したイベントまたはタブルの属性に関する条件を規定し、相関変数と称される文字列名を用いることによってこれらの条件を識別することを可能にする。

【0093】

クエリ300においては、「A」は相関変数である。一致させるべきパターンは、この相関変数に関する正規表現として指定されており、条件が有効な一致と認識されるべきさまざまな受信イベントによって満たされるべきである順番または順序を決定する。一例においては、これらの条件を満たす入力データストリームS1および入力データストリームS2における連続イベントの順番がパターンを構成する。一実施形態においては、MATCH__RECOGNIZEクエリの出力はストリームである。

【0094】

図4は、本開示の別の実施形態に従った、多重入力データストリームを識別するCQLクエリを用いてパターン認識を実行する具体例である。一実施形態においては、図4に示されるクエリ400は、クエリ300のFROM分節の内部表現を示す。一例においては、FROM分節は、ストリームS1およびストリームS2のUNION ALLサブクエリとして表現されてもよく、この場合、ストリームS1およびストリームS2は同種スキーマによって表現される。

【0095】

図4に示される例に関して、ストリームS1およびストリームS2を表わす同種スキーマは以下のように生成されてもよい：

同種スキーマS : (\$streamName char, c1 int, ELEMENT_TIME int, S1 Typ1@DynamicTypeCartridge, S2 Typ2@DynamicTypeCartridge)

ここで、「Typ1@DynamicTypeCartridge」は、ストリームS2中に存在しないストリームS1の属性のために生成された第1のダイナミックデータタイプを表わす。同様に、「Typ2@DynamicTypeCartridge」は、ストリームS1中に存在しないストリームS2の属性のために生成された第2のダイナミックデータタイプを表わす。図4に示される例に関して、「Typ1@DynamicTypeCartridge」は、ストリームS2中に存在しないストリームS1における第1の属性「c2」についての第1のダイナミックデータタイプを表わし、「Typ2@DynamicTypeCartridge」は、ストリームS1中に存在しないストリームS2における第2の属性「c3」についての第2のダイナミックデータタイプを表わす。

【0096】

一実施形態においては、上述のダイナミックデータタイプおよび同種スキーマを生成することによって、ストリームS1およびストリームS2の個別ストリーム特有の属性に以下のようにアクセスすることが可能となり得る。実行時には、ストリームS1から受取られたイベントに関して、「Typ1@DynamicTypeCartridge」ダイナミックデータタイプは、表現「Typ1@DynamicTypeCartridge(c1, c2)AS S1」を用いて、S2に存在しないストリームS1からの第1の属性「c2」を表わすように構成されてもよく、「Typ2@DynamicTypeCartridge」ダイナミックデータタイプは、式「Typ2@DynamicTypeCartridge() AS S2」を用いて、S1に存在しないストリームS2からの第2の属性「c3」を表わすように構成されてもよい。一例においては、「Typ2@DynamicTypeCartridge」は、受取られたイベントについてのストリームS2の第2の属性「c3」のNULL値を格納してもよい。

【0097】

同様に、ストリームS2から受取られたイベントに関して、「Typ1@DynamicTypeCartridge」ダイナミックデータタイプは、S2に存在しないストリームS1からの第1の属性(「c2」)を「Typ1@DynamicTypeCartridge()AS S1」として表わし、属性「c2」につ

10

20

30

40

50

いての NULL 値を格納するように構成されてもよい。加えて、「Typ2@DynamicTypeCartridge」は、表現「Typ2@DynamicTypeCartridge(c1, c3) AS S2」を用いて、S1 に存在しないストリーム S2 からの第 2 の属性「c3」を表わすように構成されてもよい。一実施形態においては、上述のダイナミックデータタイプ表現を用いて、属性「c2」が CQL クエリにおいて S1 . c2 としてアクセスされてもよく、属性「c3」は CQL クエリにおいて S2 . c3 としてアクセスされてもよい。

【0098】

加えて、図 4 に示される例においては、属性「c1」および「ELEMENT__TIME」(タイムスタンプ値)は共通属性として識別されてもよい。いくつかの例においては、共通属性「c1」は A . c1 または A . S1 . c1 として同種スキーマを用いてアクセスされてもよい。いくつかの例においては、PARTITION BY 分節も共通属性にアクセスするために用いられてもよい。

10

【0099】

上述の同種スキーマおよびダイナミックデータタイプの生成により、UNION ALL クエリ内の各々の CQL SELECT ステートメントが結果セットにおける同数のフィールドおよび類似のデータタイプを有することが可能になることが認識され得る。加えて、ダイナミックデータタイプは、特定のストリームに属さない属性に「NULL」値を割当てることができる。一実施形態においては、次いで、ストリーム S1 およびストリーム S2 の結果セットは、CQL UNION ALL サブクエリを適用してストリーム S1 およびストリーム S2 の結果セットを結合することによって、同種スキーマに基づいて結合データストリームを生成するように結合されてもよい。結合データストリームが生成され得る態様が、以下の非限定的な例を用いて説明され得る。

20

【0100】

一例として、第 1 の入力データストリーム S1 が以下のイベントおよび関連するタイムスタンプを受取るものとする。

【0101】

...
 (<timestamp 1000>, <C1 1, C2 5>)
 (<timestamp 2000>, <C1 1,C2 10>)
 (<timestamp 3000>, <C1 1 C2 15>)

30

同様に、第 2 の入力データストリーム S2 が以下のイベントおよび関連するタイムスタンプを受取るものとする。

【0102】

...
 (<timestamp 1000>, <C1 1, C3 5>)
 (<timestamp 2000>, <C1 1,C3 10>)
 (<timestamp 3000>, <C1 1 C3 15>)

40

上述のストリーム S1 およびストリーム S2 の結果セットを結合して、以下に示されるような結合データストリームを生成してもよい。一例においては、結合データストリームは、上述のような同種スキーマに基づいて、UNION ALL サブクエリにおける SELECT ステートメントを用いて結合され得るストリーム S1 およびストリーム S2 の結果セットを含む。一例においては、結合データストリームは、以下の同種スキーマ(\$streamName, c1, S1 . c2, S2, c3)によって識別され、以下のイベントおよび関連するタイムスタンプを含んでもよい。

【0103】

...
 (<timestamp 1000>, <S1, C1 1, C2 5, NULL>)
 (<timestamp 1000>, <S2, C1 1, C2 NULL, C3 5>)

50

(<timestamp 2000>, <S1, C1 1, C2 10, C3 NULL>)
 (<timestamp 2000>, <S2, C1 1, C2 NULL, C3 10>)
 (<timestamp 3000>, <S1, C1 1, C2 1, C3 NULL>)
 (<timestamp 3000>, <S2, C1 1, C2 NULL, C3 15>)

...

図5および図6は、この明細書中に記載される、多重入力データストリームにわたるパターンの検出を実現するためのそれぞれのプロセス500および600を示す例示的なフロー図である。これらのプロセス500および600は論理的なフロー図として示され、それらの各々の動作は、ハードウェア、コンピュータ命令またはそれらの組合せで実現することができる一連の動作を表わしている。コンピュータ命令のコンテキストにおいては、動作は、1つ以上のコンピュータ読取可能記憶媒体上に格納されたコンピュータ実行可能命令を表わす。これらコンピュータ実行可能命令は、1つ以上のプロセッサによって実行されたときに列挙された動作を実行する。一般に、コンピュータ実行可能命令は、特定の機能を実行するかまたは特定のデータタイプを実現するルーチン、プログラム、オブジェクト、構成要素、データ構造などを含む。動作が記載されている順序は、限定するものとして解釈されるよう意図されるものではなく、記載される動作の番号はいずれも如何なる順序で組合せされてもよく、および/または並行してプロセスを実現してもよい。

10

【0104】

加えて、プロセスのうちいくつか、いずれかまたはすべてが、実行可能命令を用いて構成された1つ以上のコンピュータシステムの制御下で実行されてもよく、1つ以上のプロセッサ上で、ハードウェアによって、またはそれらの組合せによって、集成的に実行されるコード（たとえば、実行可能命令、1つ以上のコンピュータプログラムまたは1つ以上のアプリケーション）として実現されてもよい。上述のように、コードは、たとえば、1つ以上のプロセッサによって実行可能な複数の命令を含むコンピュータプログラムの形式で、コンピュータ読取可能記憶媒体に格納されてもよい。コンピュータ読取可能記憶媒体は非一時的であってもよい。

20

【0105】

図5は、本開示の一実施形態に従って、多重入力データストリームにわたってパターンを検出するためのプロセスを示す高レベルフローチャートである。いくつかの例においては、少なくとも図1（など）に示される（たとえば、少なくとも入力データストリームモジュール116、ダイナミックデータタイプ生成モジュール118、結合データストリーム生成モジュール120、またはパターン検出モジュール122を利用する）1つ以上のサービスプロバイダコンピュータ106は、図5のプロセス500を実行し得る。プロセス500は、第1の入力データストリームおよび第2の入力データストリームを識別するステップおよび/または受取るステップを含め、ステップ502から開始されてもよい。

30

【0106】

504において、プロセス500は、第1の入力データストリームおよび第2の入力データストリームを表現するために同種スキーマを生成するステップを含み得る。同種スキーマが生成され得るプロセスが図6において詳細に説明される。

【0107】

506において、プロセス500は、第1の入力データストリームと第2の入力データストリームとを結合して、同種スキーマに基づく結合データストリームを生成するステップを含み得る。一実施形態においては、結合データストリームを生成するステップは、第1の入力データストリームから（第1のCQL SELECTステートメントを介して）タブルの第1のセットを選択するステップと、第2の入力ストリームから（第1のCQL SELECTステートメントを介して）タブルの第2のセットを選択するステップと、タブルの第1のセットおよびタブルの第2のセットにわたってサブクエリ（たとえば、UNION ALLサブクエリ）を処理して、結合データストリームを生成するステップとを含み得る。上述のとおり、同種スキーマを生成することにより、UNION ALLサブクエリ内の各々のCQL SELECTステートメントが、結果セットにおける同数の

40

50

フィールドおよび類似のデータタイプを有することが可能になる。したがって、ストリーム S 1 およびストリーム S 2 の結果セットは、結合データストリームを生成するために、同種スキーマに基づいて UNION ALL サブクエリにおける SELECT ステートメントを用いて結合されてもよい。

【 0 1 0 8 】

5 0 8 において、プロセス 5 0 0 は、結合データストリームにわたって連続クエリ（たとえば C Q L クエリ）を処理してパターンを識別するステップを含み得る。一実施形態においては、UNION ALL サブクエリの実行に基づいて生成された結合データストリームは、MATCH__RECOGNIZE 分節への入力として供給されてもよい。図 3 において説明されたように、MATCH__RECOGNIZE 分節を用いて、結合データストリームにわたって C Q L クエリを処理して、結合データストリームにおけるパターンを検出してもよい。

10

【 0 1 0 9 】

図 6 は、本開示の一実施形態に従って、同種スキーマを生成するためのプロセスを示す高レベルのフローチャートである。一実施形態においては、プロセス 6 0 0 は、図 5 において説明されたプロセス 5 0 4 の実行をより詳細に説明する。

【 0 1 1 0 】

6 0 2 において、プロセス 6 0 0 は、第 1 の入力データストリームの第 1 の属性を識別するステップを含み得る。

【 0 1 1 1 】

6 0 4 において、プロセスは、第 2 の入力データストリームの第 2 の属性を識別するステップを含み得る。一実施形態においては、第 1 の属性は、第 2 の入力データストリーム中に存在しない属性として識別されてもよく、第 2 の属性は、第 1 の入力データストリーム中に存在しない属性として識別されてもよい。

20

【 0 1 1 2 】

6 0 6 において、プロセス 6 0 0 は、第 1 の入力データストリームの第 1 の属性を表現するために第 1 のダイナミックデータタイプを生成するステップを含み得る。6 0 8 において、プロセス 6 0 0 は、第 2 の入力データストリームの第 2 の属性を表現するために第 2 のダイナミックデータタイプを生成するステップを含み得る。

【 0 1 1 3 】

6 1 0 において、プロセス 6 0 0 は共通属性を識別するステップを含み得る。一実施形態においては、共通属性は、第 1 の入力データストリームおよび第 2 の入力データストリーム中に存在する属性として識別されてもよい。

30

【 0 1 1 4 】

6 1 2 において、プロセス 6 0 0 は、同種スキーマを生成するステップを含み得る。一実施形態においては、図 2 に詳細に示されるように、同種スキーマは、第 1 の入力データストリームおよび第 2 の入力データストリームの 1 つ以上の属性の表現を含み得る。この表現は、少なくとも、共通属性、第 1 のダイナミックデータタイプおよび第 2 のダイナミックデータタイプを含む。

【 0 1 1 5 】

図 7 は、本開示の実施形態を援用し得るイベント処理システム 7 0 0 を示す簡略化された高レベル図である。イベント処理方式 7 0 0 は、1 つ以上のイベントソース（7 0 4、7 0 6、7 0 8）、イベントストリームを処理するための環境を提供するように構成されるイベント処理サーバ（event processing server : E P S）7 0 2、ならびに 1 つ以上のイベントシンク（7 1 0 および 7 1 2）を含み得る。イベントソースは、E P S 7 0 2 によって受取られるイベントストリームを生成する。E P S 7 0 2 は、1 つ以上のイベントソースから 1 つ以上のイベントストリームを受取ってもよい。たとえば、図 7 に示されるように、E P S 7 0 2 は、イベントソース 7 0 4 から入力イベントストリーム 7 1 4 を受取り、イベントソース 7 0 6 から第 2 の入力イベントストリーム 7 1 6 を受取り、イベントソース 7 0 8 から第 3 のイベント 7 1 8 を受取る。1 つ以上のイベント処理アプリケ

40

50

ーション(720、722および724)は、EPS702上でデプロイされ、EPS702によって実行されてもよい。EPS702によって実行されるイベント処理アプリケーションは、1つ以上の入力イベントストリームに従い、1つ以上のイベントを注目すべきイベントとして入力イベントストリームから選択する処理ロジックに基づいて1つ以上のイベントストリームを介して受取られたイベントを処理するように構成され得る。次いで、注目すべきイベントは、1つ以上の出力イベントストリームの形式で1つ以上のイベントシンク(710, 712)に送信されてもよい。たとえば、図7においては、EPS702は、イベントストリーム726をイベントシンク710に出力し、第2の出力イベントストリーム728をイベントシンク712に出力する。いくつかの実施形態においては、イベントソース、イベント処理アプリケーションおよびイベントシンクは、互いの構成要素に変化をもたらすことなく、これらの構成要素のうちのいずれかを追加するかまたは削除することができるように分離される。

10

20

30

40

50

【0116】

一実施形態において、EPS702を、サービスを共有するEquinox OSGiをベースとするもののような軽量Java(登録商標)アプリケーションコンテナを含むJavaサーバとして実現してもよい。いくつかの実施形態において、EPS702は、たとえばJRocket Real Timeを用いてイベント処理の超高スループットおよびマイクロ秒レイテンシをサポートし得る。EPS702はまた、イベント処理アプリケーションを開発するためのツール(たとえばOracle CEP VisualizerおよびOracle CEP IDE)を含む開発プラットフォーム(たとえば完全なリアルタイムエンドツーエンドJavaイベントドリブンアーキテクチャ(Event-Driven Architecture)(EDA)開発プラットフォーム)を提供し得る。

【0117】

イベント処理アプリケーションは、1つ以上の入力イベントストリームに従い、この1つ以上の入力イベントストリームから注目すべき1つ以上のイベントを選択するためにロジック(たとえばクエリ)を実行し、選択した注目すべきイベントを1つ以上の出力イベントストリームを介して1つ以上のイベントソースに出力するように構成される。図7は、このようなイベント処理アプリケーション720のためのドリルダウンを提供する。図7に示されるように、イベント処理アプリケーション720は、入力イベントストリーム718に従い、入力イベントストリーム718から1つ以上の注目すべきイベントを選択するためのロジックを含むクエリを、CQLエンジン/CQサービス200を介して実行し、選択された注目すべきイベントを出力イベントストリーム728を介してイベントシンク712に出力するように構成される。イベントソースの例は、アダプタ(たとえばJMS、HTTP、およびファイル)、チャンネル、プロセッサ、テーブル、キャッシュ等を含むがこれらに限定されない。イベントシンクの例は、アダプタ(たとえばJMS、HTTP、およびファイル)、チャンネル、プロセッサ、キャッシュなどを含むがこれらに限定されない。

【0118】

図7のイベント処理アプリケーション720は、1つの入力ストリームをリッスンし選択したイベントを1つの出力ストリームを介して出力するものとして示されているが、これは限定を意図したものでない。これに代わる実施形態においては、イベント処理アプリケーションは、1つ以上のイベントソースから受信した複数の入力ストリームをリッスンし、モニタリングされたストリームからイベントを選択し、選択したイベントを1つ以上の出力イベントストリームを介して1つ以上のイベントシンクに出力するように構成されてもよい。同一のクエリを2つ以上のイベントシンクに関連付け異なる種類のイベントシンクに関連付けてもよい。

【0119】

その性質上無制限なので、1つのイベントストリームを介して受信されるデータの量は通常非常に多い。結果として、クエリを目的としてすべてのデータを格納またはアーカイブすることは、通常実用的でなく望ましくない。イベントストリームを処理するには、イベントをEPS702が受信したときに、受信したイベントデータをすべて格納すること

なく、イベントをリアルタイムで処理することが必要である。したがって、EPS702は、EPS702がイベントを受信したときに、受信したイベントをすべて格納することなく、イベントを処理できるようにする、特別なクエリ機構を提供する。

【0120】

イベントドリブンのアプリケーションはルールドリブンであり、これらルールは、入力ストリームを処理するのに使用される連続クエリの形態で表わしてもよい。連続クエリは、クエリ処理の結果として注目すべきイベントおよび出力としてどんなイベントが選択されるべきであるかを含め、受信したイベントに対して実行すべき処理を特定する命令（たとえばビジネスロジック）を含み得る。連続クエリは、データ記憶装置まで持続させて、イベントの入力ストリームの処理およびイベントの出力ストリームの生成のために使用してもよい。連続クエリは、典型的には、発見すべきフィルタリングおよび集約関数を実行し入力イベントストリームから注目すべきイベントを抽出してもよい。結果として、出力イベントストリームにおけるアウトバウンドのイベントの数は通常、イベントを選択する元になる入力イベントストリームにおけるイベントの数よりも遥かに少ない。

10

【0121】

有限データ集合に対して一度実行されるSQLクエリと異なり、特定のイベントストリームに対してアプリケーションによりEPS702に登録された連続クエリは、このイベントストリーム内のイベントが受信されるたびに実行されてもよい。連続クエリ実行の一部として、EPS702は、受信したイベントを、連続クエリが指定する命令に基づいて評価して、1つ以上のイベントが注目すべきイベントとして選択されるべきであるかどうかを判断し、連続クエリ実行の結果を出力する。

20

【0122】

連続クエリは異なる言語を用いてプログラムしてもよい。ある実施形態において、連続クエリは、Oracle社が提供するCQLを用いて構成されOracle社の複合イベント処理（Complex Events Processing）（CEP）という製品によって使用されてもよい。Oracle社のCQLは、イベントストリームに対して実行することができるクエリ（CQLクエリと呼ばれる）をプログラムするのに使用できる宣言型言語である。ある実施形態において、CQLは、ストリーミングイベントデータの処理をサポートする構成が追加されたSQLに基づく。

【0123】

一実施形態において、イベント処理アプリケーションは、以下の種類の構成要素で構成されてもよい。

30

【0124】

（1）入力および出力ストリームならびにリレーションソースおよびシンクに直接インターフェイスする1つ以上のアダプタ。アダプタは、入力および出力ストリームプロトコルを解釈するように構成され、イベントデータを、アプリケーションプロセッサがクエリできる標準化された形態に変換する役割を担う。アダプタは、標準化されたイベントデータを、チャンネルまたは出力ストリームおよびリレーションシンクに転送してもよい。イベントアダプタを、さまざまなデータソースおよびシンクに対して定めてもよい。

【0125】

（2）イベント処理のエンドポイントとして作用する1つ以上のチャンネル。特に、チャンネルは、イベント処理エージェントがイベントデータに対して機能を発揮できるようになるまでイベントデータをキューに保管する役割を担う。

40

【0126】

（3）1つ以上のアプリケーションプロセッサ（またはイベント処理エージェント）は、チャンネルからの標準化されたイベントデータを消費（consume）し、これをクエリを用いて処理することにより、注目すべきイベントを選択し、選択した注目すべきイベントを出力チャンネルに転送（またはコピー）するように構成される。

【0127】

（4）1つ以上のビーン（bean）は、出力チャンネルをリッスンするように構成され、新

50

たなイベントが出力チャンネルに挿入されたことによってトリガされる。いくつかの実施形態において、このユーザコードは、昔からある単なる J a v a オブジェクト (plain-old-Java-object) (P O J O) である。ユーザアプリケーションは、 J M S、ウェブサービス、ファイルライター等の一組の外部サービスを利用して、生成されたイベントを外部イベントシンクに転送することができる。

【 0 1 2 8 】

(5) イベントビーンは、出力チャンネルをリスンするように登録されてもよく、出力チャンネルに新しいイベントを挿入することによってトリガされる。いくつかの実施形態においては、このユーザコードはオラクル C E P イベントビーン A P I を用いて、このビーンをオラクル C E P によって管理することができるようにしてもよい。

10

【 0 1 2 9 】

一実施形態においては、イベントアダプタは、イベントデータを入力チャンネルに与える。入力チャンネルは、入力チャンネルが与えるイベントに対して機能する 1 つ以上の C Q L クエリに関連付けられた C Q L プロセッサに接続される。 C Q L プロセッサは、クエリ結果が書込まれる出力チャンネルに接続される。

【 0 1 3 0 】

いくつかの実施形態において、イベント処理アプリケーションに対し、イベント処理アプリケーションのさまざまな構成要素を説明しどのようにして構成要素同士が接続されるかを説明しアプリケーションによって処理されるイベントタイプを説明するアセンブリファイルを与えてもよい。イベントを選択するための連続クエリまたはビジネスロジックを指定するために別々のファイルを与えてもよい。

20

【 0 1 3 1 】

図 7 に示されるシステム 7 0 0 が図 7 に示される構成要素以外の構成要素を有し得ることが理解されるはずである。さらに、図 7 に示される実施形態は、本開示の実施形態を組み込み得るシステムの一例に過ぎない。他のいくつかの実施形態においては、システム 7 0 0 が有する構成要素の数は図 7 に示されるものよりも多くても少なくてもよく、 2 つ以上の構成要素を組合せてもよく、または、構成要素の構成または配置が異なってもよい。システム 7 0 0 は、パーソナルコンピュータ、ポータブルデバイス (たとえば携帯電話または装置)、ワークステーション、ネットワークコンピュータ、メインフレーム、キオスク、サーバ、またはその他何等かのデータ処理システムを含むさまざまな種類のものであってもよい。他のいくつかの実施形態において、システム 7 0 0 を、システム 7 0 0 の 1 つ以上の構成要素がクラウド内の 1 つ以上のネットワークに分散している分散システムとして構成してもよい。

30

【 0 1 3 2 】

図 7 に示される構成要素のうち 1 つ以上は、ソフトウェア、ハードウェア、またはその組合せにおいて実現し得る。いくつかの実施形態において、ソフトウェアは、メモリ (たとえば非一時的なコンピュータ読取可能な媒体) に、メモリ素子に、または何等かのその他の物理メモリに格納されてもよく、 1 つ以上の処理部 (たとえば 1 つ以上のプロセッサ、 1 つ以上のプロセッサコア、 2 つ以上の G P U など) によって実行されてもよい。

【 0 1 3 3 】

図のいくつかに示されるシステムは、さまざまな構成で設けられてもよい。いくつかの実施形態においては、システムは、当該システムの 1 つ以上の構成要素がクラウドコンピューティングシステムにおける 1 つ以上のネットワークを介して分散されている分散型システムとして構成されてもよい。

40

【 0 1 3 4 】

図 8 は、いくつかの実施形態のうち 1 つの実施形態を実現するための分散型システム 8 0 0 の簡略図を示す。図示される実施形態においては、分散型システム 8 0 0 は、 1 つ以上のネットワーク 8 1 0 を介して、ウェブブラウザ、プロプライエタリクライアント (たとえばオラクルフォーム) などのクライアントアプリケーションを実行して動作させるように構成される 1 つ以上のクライアントコンピューティング装置 8 0 2、 8 0 4、 8 0

50

6 および 808 を含む。サーバ 812 は、ネットワーク 810 を介してリモートクライアントコンピューティング装置 802、804、806 および 808 と通信可能に結合されてもよい。

【0135】

さまざまな実施形態においては、サーバ 812 は、システムの構成要素のうち 1 つ以上によって提供される 1 つ以上のサービスまたはソフトウェアアプリケーションを実行するように適合されてもよい。サービスまたはソフトウェアアプリケーションは非仮想環境および仮想環境を含み得る。仮想環境は、2次元または三次元(3D)表現、ページベースの論理的環境などであるとなかろうと、仮想イベント、トレードショー、シミュレータ、クラスルーム、購買商品取引および企業活動のために用いられるものを含み得る。いくつかの実施形態においては、これらのサービスは、ウェブベースのサービスもしくはクラウドサービスとして、またはソフトウェア・アズ・ア・サービス(Software as a Service: SaaS)モデルのもとで、クライアントコンピューティング装置 802、804、806 および / または 808 のユーザに供給されてもよい。そして、クライアントコンピューティング装置 802、804、806 および / または 808 を動作させるユーザは、1 つ以上のクライアントアプリケーションを利用して、サーバ 812 と相互作用して、これらの構成要素によって提供されるサービスを利用し得る。

10

【0136】

図に示されている構成では、システム 800 のソフトウェアコンポーネント 818、820 および 822 は、サーバ 812 上に実装されるように示されている。また、他の実施形態においては、システム 800 の構成要素のうち 1 つ以上および / またはこれらの構成要素によって提供されるサービスは、クライアントコンピューティング装置 802、804、806 および / または 808 のうちの 1 つ以上によって実現されてもよい。その場合、クライアントコンピューティング装置を動作させるユーザは、1 つ以上のクライアントアプリケーションを利用して、これらの構成要素によって提供されるサービスを使用し得る。これらの構成要素は、ハードウェア、ファームウェア、ソフトウェア、またはそれらの組合せで実現されてもよい。分散システム 800 とは異なり得るさまざまな異なるシステム構成が可能であることが理解されるべきである。したがって、図に示されている実施形態は、実施形態のシステムを実現するための分散システムの一例であり、限定的であるよう意図したものではない。

20

30

【0137】

クライアントコンピューティング装置 802、804、806 および / または 808 は、手持ち式携帯機器(たとえば iPhone(登録商標)、携帯電話、iPad(登録商標)、計算タブレット、パーソナルデジタルアシスタント(personal digital assistant: PDA)またはウェアラブル装置(たとえば Google Glass(登録商標)ヘッドマウントディスプレイ)であってもよく、当該装置は、マイクロソフト・ウィンドウズ・モバイル(登録商標)などのソフトウェアを実行し、および / または、iOS、ウィンドウズ・フォン、アンドロイド、ブラックベリー 10、Palm OS などのさまざまなモバイルオペレーティングシステムを実行し、インターネット、eメール、ショート・メッセージ・サービス(short message service: SMS)、ブラックベリー(登録商標)、または使用可能な他の通信プロトコルである。クライアントコンピューティング装置は、汎用パーソナルコンピュータであってもよく、当該汎用パーソナルコンピュータは、一例として、マイクロソフト・ウィンドウズ(登録商標)、アップル・マッキントッシュ(登録商標)および / または リナックス(登録商標)オペレーティングシステムのさまざまなバージョンを実行するパーソナルコンピュータおよび / または ラップトップコンピュータを含む。クライアントコンピューティング装置は、ワークステーションコンピュータであってもよく、当該ワークステーションコンピュータは、たとえば Google Chrome OS などのさまざまな GNU / リナックスオペレーティングシステムを含むがこれらに限定されるものではないさまざまな市販の UNIX(登録商標)または UNIX ライクオペレーティングシステムのうちのいずれかを実行する。代替的に、またはさらに、クライアントコ

40

50

ンピューティング装置 802, 804, 806 および 808 は、シン・クライアントコンピュータ、インターネットにより可能なゲームシステム（たとえばキネクト（登録商標）ジェスチャ入力装置を備えるかまたは備えないマイクロソフト X ボックスゲーム機）、および/または、ネットワーク 810 を介して通信が可能なパーソナルメッセージング装置などのその他の電子装置であってもよい。

【0138】

例示的な分散システム 800 は、4 個のクライアントコンピューティング装置を有するように示されているが、任意の数のクライアントコンピューティング装置がサポートされてもよい。センサを有する装置などの他の装置が、サーバ 812 と相互作用してもよい。

【0139】

分散システム 800 におけるネットワーク 810 は、さまざまな市販のプロトコルのうちのいずれかを用いてデータ通信をサポートすることができる、当業者になじみのある任意のタイプのネットワークであってもよく、当該プロトコルは、TCP/IP（伝送制御プロトコル/インターネットプロトコル）、SNA（システムネットワークアーキテクチャ）、IPX（インターネットパケット交換）、アップルトークなどを含むが、これらに限定されるものではない。単に一例として、ネットワーク 810 は、イーサネット（登録商標）、トークンリングなどに基づくものなどのローカルエリアネットワーク（LAN）であってもよい。ネットワーク 810 は、広域ネットワークおよびインターネットであってもよい。ネットワーク 810 は、仮想ネットワークを含んでいてもよく、当該仮想ネットワークは、仮想プライベートネットワーク（virtual private network: VPN）、イントラネット、エクストラネット、公衆交換電話網（public switched telephone network: PSTN）、赤外線ネットワーク、無線ネットワーク（たとえば米国電気電子学会（Institute of Electrical and Electronics: IEEE）802.11 の一連のプロトコル、ブルートゥース（登録商標）および/またはその他の無線プロトコルのうちのいずれかのもとで動作するネットワーク）、および/またはこれらの任意の組合せ、および/または他のネットワークを含むが、これらに限定されるものではない。

【0140】

サーバ 812 は、1 つ以上の汎用コンピュータ、専用サーバコンピュータ（一例として、PC（パーソナルコンピュータ）サーバ、UNIX（登録商標）サーバ、ミッドレンジサーバ、メインフレームコンピュータ、ラックマウント式サーバなどを含む）、サーバファーム、サーバクラスタ、またはその他の適切な構成および/または組合せで構成され得る。サーバ 812 は、仮想オペレーティングシステムを実行する 1 つ以上の仮想マシン、または仮想化を含む他のコンピューティングアーキテクチャを含み得る。論理記憶装置の 1 つ以上のフレキシブルプールは、サーバのための仮想記憶デバイスを維持するように仮想化することができる。仮想ネットワークは、ソフトウェア定義型ネットワークングを用いて、サーバ 812 によって制御することができる。さまざまな実施形態においては、サーバ 812 は、上記の開示に記載されている 1 つ以上のサービスまたはソフトウェアアプリケーションを実行するように適合され得る。たとえば、サーバ 812 は、本開示の実施形態に係る上記の処理を実行するためのサーバに対応してもよい。

【0141】

サーバ 812 は、上記のものの中のいずれか、および、任意の市販のサーバオペレーティングシステムを含むオペレーティングシステムを実行し得る。また、サーバ 812 は、HTTP（ハイパーテキスト転送プロトコル）サーバ、FTP（ファイル転送プロトコル）サーバ、CGI（共通ゲートウェイインターフェース）サーバ、JAVA（登録商標）サーバ、データベースサーバなどを含むさまざまなさらなるサーバアプリケーションおよび/または中間層アプリケーションのうちのいずれかを実行し得る。例示的なデータベースサーバは、オラクル社、マイクロソフト社、サイベース社、IBM 社（International Business Machines）などから市販されているものを含むが、これらに限定されるものではない。

【0142】

サーバ 812 は、上記のものの中のいずれか、および、任意の市販のサーバオペレーティングシステムを含むオペレーティングシステムを実行し得る。また、サーバ 812 は、HTTP（ハイパーテキスト転送プロトコル）サーバ、FTP（ファイル転送プロトコル）サーバ、CGI（共通ゲートウェイインターフェース）サーバ、JAVA（登録商標）サーバ、データベースサーバなどを含むさまざまなさらなるサーバアプリケーションおよび/または中間層アプリケーションのうちのいずれかを実行し得る。例示的なデータベースサーバは、オラクル社、マイクロソフト社、サイベース社、IBM 社（International Business Machines）などから市販されているものを含むが、これらに限定されるものではない。

いくつかの実現例では、サーバ 812 は、クライアントコンピューティング装置 802、804、806 および 808 のユーザから受信されたデータフィードおよび/またはイベント更新を分析および統合するための 1 つ以上のアプリケーションを含み得る。一例として、データフィードおよび/またはイベント更新は、1 つ以上の第三者情報源および連続的なデータストリームから受信されるツイッター（登録商標）フィード、フェイスブック（登録商標）更新またはリアルタイム更新を含み得るが、これらに限定されるものではなく、センサデータアプリケーション、金融ティッカ、ネットワーク性能測定ツール（たとえばネットワークモニタリングおよびトラフィック管理アプリケーション）、クリックストリーム分析ツール、自動車交通モニタリングなどに関連するリアルタイムイベントを含み得る。また、サーバ 812 は、クライアントコンピューティング装置 802、804、806 および 808 の 1 つ以上の表示装置を介してデータフィードおよび/またはリアルタイムイベントを表示するための 1 つ以上のアプリケーションを含み得る。

10

20

30

40

50

【0143】

また、分散システム 800 は、1 つ以上のデータベース 814 および 816 を含み得る。データベース 814 および 816 は、さまざまな場所に存在し得る。一例として、データベース 814 および 816 の 1 つ以上は、サーバ 812 にローカルな（および/または存在する）非一時的な記憶媒体に存在していてもよい。代替的に、データベース 814 および 816 は、サーバ 812 から遠く離れていて、ネットワークベースまたは専用の接続を介してサーバ 812 と通信してもよい。一組の実施形態においては、データベース 814 および 816 は、記憶領域ネットワーク（storage-area network：SAN）に存在していてもよい。同様に、サーバ 812 に起因する機能を実行するための任意の必要なファイルが、サーバ 812 上にローカルに、および/または、リモートで適宜格納されていてもよい。一組の実施形態においては、データベース 814 および 816 は、SQL フォーマットコマンドに回答してデータを格納、更新および検索するように適合された、オラクル社によって提供されるデータベースなどのリレーショナルデータベースを含み得る。

【0144】

図 9 は、本開示の実施形態に係る、実施形態のシステムの 1 つ以上の構成要素によって提供されるサービスをクラウドサービスとして供給することができるシステム環境 900 の 1 つ以上の構成要素の簡略化されたブロック図である。示されている実施形態においては、システム環境 900 は、クラウドサービスを提供するクラウドインフラストラクチャシステム 902 と相互作用するようにユーザによって使用され得る 1 つ以上のクライアントコンピューティング装置 904、906 および 908 を含む。クライアントコンピューティング装置は、クラウドインフラストラクチャシステム 902 によって提供されるサービスを使用するためにクラウドインフラストラクチャシステム 902 と相互作用するようにクライアントコンピューティング装置のユーザによって使用され得る、ウェブブラウザ、専有のクライアントアプリケーション（たとえばオラクルフォームズ）または他のアプリケーションなどのクライアントアプリケーションを動作させるように構成され得る。

【0145】

図に示されているクラウドインフラストラクチャシステム 902 は、示されている構成要素とは他の構成要素を有していてもよいということが理解されるべきである。さらに、図に示されている実施形態は、本発明の実施形態を組込むことができるクラウドインフラストラクチャシステムの一例に過ぎない。いくつかの他の実施形態においては、クラウドインフラストラクチャシステム 902 は、図に示されているものよりも多いまたは少ない数の構成要素を有していてもよく、2 つ以上の構成要素を組合せてもよく、または構成要素の異なる構成または配置を有していてもよい。

【0146】

クライアントコンピューティング装置 904、906 および 908 は、802、804、806 および 808 について上記したものと類似の装置であってもよい。

【0147】

例示的なシステム環境 900 は 3 個のクライアントコンピューティング装置を有するよ

うに示されているが、任意の数のクライアントコンピューティング装置がサポートされてもよい。センサを有する装置などの他の装置が、クラウドインフラストラクチャシステム 902 と相互作用してもよい。

【0148】

ネットワーク 910 は、クライアント 904 , 906 および 908 とクラウドインフラストラクチャシステム 902 との間のデータの通信およびやりとりを容易にし得る。各々のネットワークは、ネットワーク 810 について上記したものを含むさまざまな市販のプロトコルのうちのいずれかを用いてデータ通信をサポートすることができる、当業者になじみのある任意のタイプのネットワークであってもよい。

【0149】

クラウドインフラストラクチャシステム 902 は、サーバ 812 について上記したものを含み得る 1 つ以上のコンピュータおよび / またはサーバを備え得る。

【0150】

特定の実施形態においては、クラウドインフラストラクチャシステムによって提供されるサービスは、オンラインデータ記憶およびバックアップソリューション、ウェブベースの e メールサービス、ホスト型オフィススイートおよびドキュメントコラボレーションサービス、データベース処理、管理技術サポートサービスなどの、クラウドインフラストラクチャシステムのユーザがオンデマンドで利用可能な多数のサービスを含み得る。クラウドインフラストラクチャシステムによって提供されるサービスは、そのユーザのニーズを満たすように動的にスケール可能である。クラウドインフラストラクチャシステムによって提供されるサービスの具体的なインスタンス化は、本明細書では「サービスインスタンス」と称される。一般に、インターネットなどの通信ネットワークを介してクラウドサービスプロバイダのシステムからユーザが利用可能な任意のサービスは、「クラウドサービス」と称される。通常、パブリッククラウド環境では、クラウドサービスプロバイダのシステムを構成するサーバおよびシステムは、顧客自身のオンプレミスサーバおよびシステムとは異なっている。たとえば、クラウドサービスプロバイダのシステムがアプリケーションをホストしてもよく、ユーザは、インターネットなどの通信ネットワークを介してオンデマンドで当該アプリケーションを注文および使用してもよい。

【0151】

いくつかの例では、コンピュータネットワーククラウドインフラストラクチャにおけるサービスは、記憶装置、ホスト型データベース、ホスト型ウェブサーバ、ソフトウェアアプリケーションへの保護されたコンピュータネットワークアクセス、またはクラウドベンダによってユーザに提供されるかもしくはそうでなければ当該技術分野において公知の他のサービスを含み得る。たとえば、サービスは、インターネットを介したクラウド上のリモート記憶装置へのパスワードによって保護されたアクセスを含み得る。別の例として、サービスは、ネットワーク化された開発者による私的使用のためのウェブサービスベースのホスト型リレーショナルデータベースおよびスクリプト言語ミドルウェアエンジンを含み得る。別の例として、サービスは、クラウドベンダのウェブサイト上でホストされる e メールソフトウェアアプリケーションへのアクセスを含み得る。

【0152】

特定の実施形態においては、クラウドインフラストラクチャシステム 902 は、セルフサービスの、サブスクリプションベースの、弾力的にスケラブルな、信頼性のある、高可用性の、安全な態様で顧客に配信される一連のアプリケーション、ミドルウェアおよびデータベースサービス提供品を含み得る。このようなクラウドインフラストラクチャシステムの一例は、本譲受人によって提供されるオラクルパブリッククラウドである。

【0153】

「ビッグデータ (Big data)」は、多くのレベルで、かつさまざまなスケールでインフラストラクチャシステムによってホストおよび / または処理されてもよい。大量のデータを視覚化し、トレンドを検出し、および / または、データと相互作用させるために、分析者および研究者は極めて大きいデータ集合を格納し処理することができる。平行にリンク

10

20

30

40

50

された何十、何百または何千ものプロセッサがこのようなデータに対して作用可能であり、これにより、このようなデータを表示し得るか、または、データに対する外力をシミュレートし得るかもしくはそれが表しているものをシミュレートし得る。これらのデータ集合は、データベースにおいて編制されたデータ、もしくは構造化モデルに従ったデータ、および/または、非体系的なデータ（たとえば電子メール、画像、データプロブ（バイナリ大型オブジェクト）、ウェブページ、複雑なイベント処理）などの構造化されたデータを必要とする可能性がある。目標物に対してより多くの（またはより少数の）コンピューティングリソースを比較的迅速に集中させるために実施形態の能力を強化することにより、ビジネス、政府関係機関、研究組織、私人、同じ目的をもった個人もしくは組織のグループ、または他のエンティティから、要求に基づいて大量のデータ集合上でタスクを実行するために、クラウドインフラストラクチャシステムより良好に利用可能となる。

10

【0154】

さまざまな実施形態においては、クラウドインフラストラクチャシステム902は、クラウドインフラストラクチャシステム902によって供給されるサービスへの顧客のサブスクリプションを自動的にプロビジョニング、管理および追跡するように適合され得る。クラウドインフラストラクチャシステム902は、さまざまなデプロイメントモデルを介してクラウドサービスを提供し得る。たとえば、クラウドインフラストラクチャシステム902が、（たとえばオラクル社によって所有される）クラウドサービスを販売する組織によって所有され、一般大衆またはさまざまな産業企業がサービスを利用できるパブリッククラウドモデルのもとでサービスが提供されてもよい。別の例として、クラウドインフラストラクチャシステム902が単一の組織のためだけに運営され、当該組織内の1つ以上のエンティティにサービスを提供し得るプライベートクラウドモデルのもとでサービスが提供されてもよい。また、クラウドインフラストラクチャシステム902およびクラウドインフラストラクチャシステム902によって提供されるサービスが、関連のコミュニティ内のいくつかの組織によって共有されるコミュニティクラウドモデルのもとでクラウドサービスが提供されてもよい。また、2つ以上の異なるモデルの組合せであるハイブリッドクラウドモデルのもとでクラウドサービスが提供されてもよい。

20

【0155】

いくつかの実施形態においては、クラウドインフラストラクチャシステム902によって提供されるサービスは、ソフトウェア・アズ・ア・サービス（Software as a Service : SaaS）カテゴリ、プラットフォーム・アズ・ア・サービス（Platform as a Service : PaaS）カテゴリ、インフラストラクチャ・アズ・ア・サービス（Infrastructure as a Service : IaaS）カテゴリ、またはハイブリッドサービスを含むサービスの他のカテゴリのもとで提供される1つ以上のサービスを含み得る。顧客は、サブスクリプションオーダーによって、クラウドインフラストラクチャシステム902によって提供される1つ以上のサービスを注文し得る。次いで、クラウドインフラストラクチャシステム902は、顧客のサブスクリプションオーダーにおけるサービスを提供するために処理を実行する。

30

【0156】

いくつかの実施形態においては、クラウドインフラストラクチャシステム902によって提供されるサービスは、アプリケーションサービス、プラットフォームサービスおよびインフラストラクチャサービスを含み得るが、これらに限定されるものではない。いくつかの例では、アプリケーションサービスは、SaaSプラットフォームを介してクラウドインフラストラクチャシステムによって提供されてもよい。SaaSプラットフォームは、SaaSカテゴリに分類されるクラウドサービスを提供するように構成され得る。たとえば、SaaSプラットフォームは、一体化された開発およびデプロイメントプラットフォーム上で一連のオンデマンドアプリケーションを構築および配信するための機能を提供し得る。SaaSプラットフォームは、SaaSサービスを提供するための基本的なソフトウェアおよびインフラストラクチャを管理および制御し得る。SaaSプラットフォームによって提供されるサービスを利用することによって、顧客は、クラウドインフラ

40

50

ラクチャシステムで実行されるアプリケーションを利用することができる。顧客は、顧客が別々のライセンスおよびサポートを購入する必要なく、アプリケーションサービスを取得することができる。さまざまな異なる SaaS サービスが提供されてもよい。例としては、大規模組織のための販売実績管理、企業統合およびビジネスの柔軟性のためのソリューションを提供するサービスが挙げられるが、これらに限定されるものではない。

【0157】

いくつかの実施形態においては、プラットフォームサービスは、PaaS プラットフォームを介してクラウドインフラストラクチャシステムによって提供されてもよい。PaaS プラットフォームは、PaaS カテゴリに分類されるクラウドサービスを提供するように構成され得る。プラットフォームサービスの例としては、組織（オラクル社など）が既存のアプリケーションを共有の共通アーキテクチャ上で統合することを可能にするサービス、および、プラットフォームによって提供される共有のサービスを活用する新たなアプリケーションを構築する機能を挙げることができるが、これらに限定されるものではない。PaaS プラットフォームは、PaaS サービスを提供するための基本的なソフトウェアおよびインフラストラクチャを管理および制御し得る。顧客は、顧客が別々のライセンスおよびサポートを購入する必要なく、クラウドインフラストラクチャシステムによって提供される PaaS サービスを取得することができる。プラットフォームサービスの例としては、オラクル Java クラウドサービス（Java Cloud Service：JCS）、オラクルデータベースクラウドサービス（Database Cloud Service：DBCS）などが挙げられるが、これらに限定されるものではない。

10

20

【0158】

PaaS プラットフォームによって提供されるサービスを利用することによって、顧客は、クラウドインフラストラクチャシステムによってサポートされるプログラミング言語およびツールを利用ことができ、デプロイされたサービスを制御することもできる。いくつかの実施形態においては、クラウドインフラストラクチャシステムによって提供されるプラットフォームサービスは、データベースクラウドサービス、ミドルウェアクラウドサービス（たとえばオラクルフュージョンミドルウェアサービス）および Java クラウドサービスを含み得る。一実施形態においては、データベースクラウドサービスは、組織がデータベースリソースをプールしてデータベースクラウドの形態でデータベース・アズ・ア・サービスを顧客に供給することを可能にする共有のサービスデプロイメントモデルをサポートし得る。ミドルウェアクラウドサービスは、クラウドインフラストラクチャシステムにおいてさまざまなビジネスアプリケーションを開発およびデプロイするために顧客にプラットフォームを提供し得て、Java クラウドサービスは、クラウドインフラストラクチャシステムにおいて Java アプリケーションをデプロイするために顧客にプラットフォームを提供し得る。

30

【0159】

さまざまな異なるインフラストラクチャサービスは、クラウドインフラストラクチャシステムにおける IaaS プラットフォームによって提供されてもよい。インフラストラクチャサービスは、記憶装置、ネットワークなどの基本的な計算リソース、ならびに、SaaS プラットフォームおよび PaaS プラットフォームによって提供されるサービスを利用する顧客のための他の基礎的な計算リソースの管理および制御を容易にする。

40

【0160】

また、特定の実施形態においては、クラウドインフラストラクチャシステム 902 は、クラウドインフラストラクチャシステムの顧客にさまざまなサービスを提供するために使用されるリソースを提供するためのインフラストラクチャリソース 930 を含み得る。一実施形態においては、インフラストラクチャリソース 930 は、PaaS プラットフォームおよび SaaS プラットフォームによって提供されるサービスを実行するための、サーバ、記憶装置およびネットワークリソースなどのハードウェアの事前に一体化された最適な組合せを含み得る。

【0161】

50

いくつかの実施形態においては、クラウドインフラストラクチャシステム 902 におけるリソースは、複数のユーザによって共有され、デマンドごとに動的に再割り振りされる。また、リソースは、異なる時間帯にユーザに割り振られ得る。たとえば、クラウドインフラストラクチャシステム 930 は、第 1 の時間帯におけるユーザの第 1 の組が規定の時間にわたってクラウドインフラストラクチャシステムのリソースを利用することを可能にし、次いで、異なる時間帯に位置するユーザの別の組への同一のリソースの再割り振りを可能にし、それによってリソースの利用を最大化することができる。

【0162】

特定の実施形態においては、クラウドインフラストラクチャシステム 902 の異なる構成要素またはモジュール、および、クラウドインフラストラクチャシステム 902 によって提供されるサービス、によって共有されるいくつかの内部共有サービス 932 が提供され得る。これらの内部共有サービスは、セキュリティおよびアイデンティティサービス、インテグレーションサービス、企業リポジトリサービス、企業マネージャサービス、ウイルススキャンおよびホワイトリストサービス、高可用性・バックアップおよび回復サービス、クラウドサポートを可能にするためのサービス、eメールサービス、通知サービス、ファイル転送サービスなどを含み得るが、これらに限定されるものではない。

10

【0163】

特定の実施形態においては、クラウドインフラストラクチャシステム 902 は、クラウドインフラストラクチャシステムにおけるクラウドサービス（たとえば SaaS、PaaS および IaaS サービス）の包括的管理を提供し得る。一実施形態においては、クラウド管理機能は、クラウドインフラストラクチャシステム 902 によって受信された顧客のサブスクリプションをプロビジョニング、管理および追跡などするための機能を含み得る。

20

【0164】

一実施形態においては、図に示されているように、クラウド管理機能は、オーダー管理モジュール 920、オーダーオーケストレーションモジュール 922、オーダープロビジョニングモジュール 924、オーダー管理および監視モジュール 926、ならびにアイデンティティ管理モジュール 928 などの 1 つ以上のモジュールによって提供され得る。これらのモジュールは、汎用コンピュータ、専用サーバコンピュータ、サーバファーム、サーバクラスタ、またはその他の適切な構成および/もしくは組み合わせであり得る 1 つ以上のコンピュータおよび/またはサーバを含み得るか、またはそれらを用いて提供され得る。

30

【0165】

例示的な動作 934 において、顧客は、クラウドインフラストラクチャシステム 902 によって提供される 1 つ以上のサービスを要求し、クラウドインフラストラクチャシステム 902 によって供給される 1 つ以上のサービスのサブスクリプションについてオーダーを行うことによって、クライアント装置 904、906 または 908 などのクライアント装置を用いてクラウドインフラストラクチャシステム 902 と相互作用し得る。特定の実施形態においては、顧客は、クラウドユーザインターフェース (User Interface: UI)、すなわちクラウド UI 912、クラウド UI 914 および/またはクラウド UI 916 にアクセスして、これらの UI を介してサブスクリプションオーダーを行い得る。顧客がオーダーを行ったことに応答してクラウドインフラストラクチャシステム 902 によって受信されたオーダー情報は、顧客および顧客がサブスクライブする予定のクラウドインフラストラクチャシステム 902 によって供給される 1 つ以上のサービスを特定する情報を含み得る。

40

【0166】

オーダーが顧客によって行われた後、オーダー情報は、クラウド UI 912、914 および/または 916 を介して受信される。

【0167】

動作 936 において、オーダーは、オーダーデータベース 918 に格納される。オーダ

50

ーデータベース 9 1 8 は、クラウドインフラストラクチャシステム 9 1 8 によって動作され、かつ、他のシステム要素と連携して動作されるいくつかのデータベースのうちの 1 つであってもよい。

【 0 1 6 8 】

動作 9 3 8 において、オーダー情報は、オーダー管理モジュール 9 2 0 に転送される。いくつかの例では、オーダー管理モジュール 9 2 0 は、オーダーの確認および確認時のオーダーの予約などのオーダーに関連するBillingおよびアカウントing機能を実行するように構成され得る。

【 0 1 6 9 】

動作 9 4 0 において、オーダーに関する情報は、オーダーオーケストレーションモジュール 9 2 2 に通信される。オーダーオーケストレーションモジュール 9 2 2 は、顧客によって行われたオーダーのためのサービスおよびリソースのプロビジョニングをオーケストレートするためにオーダー情報を利用し得る。いくつかの例では、オーダーオーケストレーションモジュール 9 2 2 は、オーダープロビジョニングモジュール 9 2 4 のサービスを用いてサブスクライブされたサービスをサポートするためにリソースのプロビジョニングをオーケストレートし得る。

10

【 0 1 7 0 】

特定の実施形態においては、オーダーオーケストレーションモジュール 9 2 2 は、各々のオーダーに関連付けられるビジネスプロセスの管理を可能にし、ビジネス論理を適用してオーダーがプロビジョニングに進むべきか否かを判断する。動作 9 4 2 において、新たなサブスクリプションについてのオーダーを受信すると、オーダーオーケストレーションモジュール 9 2 2 は、リソースを割り振って当該サブスクリプションオーダーを満たすのに必要とされるそれらのリソースを構成するための要求をオーダープロビジョニングモジュール 9 2 4 に送る。オーダープロビジョニングモジュール 9 2 4 は、顧客によってオーダーされたサービスのためのリソースの割り振りを可能にする。オーダープロビジョニングモジュール 9 2 4 は、クラウドインフラストラクチャシステム 9 0 0 によって提供されるクラウドサービスと、要求されたサービスを提供するためのリソースをプロビジョニングするために使用される物理的実装層との間にあるレベルの抽象化を提供する。したがって、オーダーオーケストレーションモジュール 9 2 2 は、サービスおよびリソースが実際に実行中にプロビジョニングされるか、事前にプロビジョニングされて要求があったときに割振られる / 割当てられるのみであるかなどの実装の詳細から分離させることができる。

20

30

【 0 1 7 1 】

動作 9 4 4 において、サービスおよびリソースがプロビジョニングされると、提供されたサービスの通知が、クラウドインフラストラクチャシステム 9 0 2 のオーダープロビジョニングモジュール 9 2 4 によってクライアント装置 9 0 4 , 9 0 6 および / または 9 0 8 上の顧客に送られ得る。

【 0 1 7 2 】

動作 9 4 6 において、顧客のサブスクリプションオーダーが、オーダー管理および監視モジュール 9 2 6 によって管理および追跡され得る。いくつかの例では、オーダー管理および監視モジュール 9 2 6 は、使用される記憶量、転送されるデータ量、ユーザの数、ならびにシステムアップ時間およびシステムダウン時間などのサブスクリプションオーダーにおけるサービスについての使用統計を収集するように構成され得る。

40

【 0 1 7 3 】

特定の実施形態においては、クラウドインフラストラクチャシステム 9 0 0 は、アイデンティティ管理モジュール 9 2 8 を含み得る。アイデンティティ管理モジュール 9 2 8 は、クラウドインフラストラクチャシステム 9 0 0 におけるアクセス管理および認可サービスなどのアイデンティティサービスを提供するように構成され得る。いくつかの実施形態においては、アイデンティティ管理モジュール 9 2 8 は、クラウドインフラストラクチャシステム 9 0 2 によって提供されるサービスを利用したい顧客についての情報を制御し得

50

る。このような情報は、このような顧客のアイデンティティを認証する情報、および、それらの顧客がさまざまなシステムリソース（たとえばファイル、ディレクトリ、アプリケーション、通信ポート、メモリセグメントなど）に対してどのアクションを実行することを認可されるかを記載する情報を含み得る。また、アイデンティティ管理モジュール 928 は、各々の顧客についての説明的情報、ならびに、どのようにしておよび誰によって当該説明的情報がアクセスおよび変更され得るかについての情報の管理を含み得る。

【0174】

図 10 は、本発明のさまざまな実施形態を実現することができる例示的なコンピュータシステム 1000 を示す。システム 1000 は、上記のコンピュータシステムのうちいずれかを実現するために使用され得る。図に示されているように、コンピュータシステム 1000 は、バスサブシステム 1002 を介していくつかの周辺サブシステムと通信する処理ユニット 1004 を含む。これらの周辺サブシステムは、処理加速ユニット 1006 と、I/O サブシステム 1008 と、記憶サブシステム 1018 と、通信サブシステム 1024 とを含み得る。記憶サブシステム 1018 は、有形のコンピュータ読取可能な記憶媒体 1022 と、システムメモリ 1010 とを含む。

10

【0175】

バスサブシステム 1002 は、コンピュータシステム 1000 のさまざまな構成要素およびサブシステムに、意図されたように互いに通信させるための機構を提供する。バスサブシステム 1002 は、単一のバスとして概略的に示されているが、バスサブシステムの代替的な実施形態は、複数のバスを利用してもよい。バスサブシステム 1002 は、メモリバスまたはメモリコントローラ、周辺バス、およびさまざまなバスアーキテクチャのうちいずれかを使用するローカルバスを含むいくつかのタイプのバス構造のうちいずれかであってもよい。たとえば、このようなアーキテクチャは、IEEE P1386.1 標準に合わせて製造されたメザニンバスとして実現可能な、業界標準アーキテクチャ（Industry Standard Architecture：ISA）バス、マイクロチャネルアーキテクチャ（Micro Channel Architecture：MCA）バス、拡張 ISA（Enhanced ISA：EISA）バス、ビデオ・エレクトロニクス・スタンダーズ・アソシエーション（Video Electronics Standards Association：VESA）ローカルバスおよび周辺機器相互接続（Peripheral Component Interconnect：PCI）バスを含み得る。

20

【0176】

1 つ以上の集積回路（たとえば従来のマイクロプロセッサまたはマイクロコントローラ）として実現可能な処理ユニット 1004 は、コンピュータシステム 1000 の動作を制御する。処理ユニット 1004 には、1 つ以上のプロセッサが含まれ得る。これらのプロセッサは、単一コアまたはマルチコアプロセッサを含み得る。特定の実施形態においては、処理ユニット 1004 は、各々の処理ユニットに含まれる単一またはマルチコアプロセッサを有する 1 つ以上の独立した処理ユニット 1032 および / または 1034 として実現されてもよい。また、他の実施形態においては、処理ユニット 1004 は、2 つのデュアルコアプロセッサを単一のチップに組み入れることによって形成されるクアッドコア処理ユニットとして実現されてもよい。

30

【0177】

さまざまな実施形態においては、処理ユニット 1004 は、プログラムコードに応答してさまざまなプログラムを実行し得て、同時に実行される複数のプログラムまたはプロセスを維持し得る。任意の所与の時点において、実行されるべきプログラムコードのうちいくつかまたは全ては、プロセッサ 1004 および / または記憶サブシステム 1018 に存在し得る。好適なプログラミングを通じて、プロセッサ 1004 は、上記のさまざまな機能を提供し得る。また、コンピュータシステム 1000 は、デジタル信号プロセッサ（digital signal processor：DSP）、特殊用途プロセッサなどを含み得る処理加速ユニット 1006 を含み得る。

40

【0178】

I/O サブシステム 1008 は、ユーザインターフェイス入力装置と、ユーザインター

50

フェイス出力装置とを含み得る。ユーザインターフェイス入力装置は、キーボード、マウスまたはトラックボールなどのポインティング装置、タッチパッドまたはタッチスクリーンを含み得て、これらは、音声コマンド認識システム、マイクロホンおよび他のタイプの入力装置とともに、ディスプレイ、スクロールホイール、クリックホイール、ダイヤル、ボタン、スイッチ、キーパッド、オーディオ入力装置に組込まれている。ユーザインターフェイス入力装置は、たとえば、ジェスチャおよび話されたコマンドを用いてナチュラルユーザインターフェイスを介してユーザがマイクロソフトXbox（登録商標）360ゲームコントローラなどの入力装置を制御して入力装置と相互作用することを可能にするマイクロソフトキネクト（登録商標）モーションセンサなどのモーション検知および/またはジェスチャ認識装置を含み得る。また、ユーザインターフェイス入力装置は、ユーザから眼球運動（たとえば撮影および/またはメニュー選択を行っている間の「まばたき」）を検出して、当該眼球ジェスチャを入力装置への入力として変換するグーグルグラス（登録商標）まばたき検出器などの眼球ジェスチャ認識装置を含み得る。また、ユーザインターフェイス入力装置は、ユーザが音声コマンドを介して音声認識システム（たとえばSiri（登録商標）ナビゲータ）と相互作用することを可能にする音声認識検知装置を含み得る。

10

20

30

40

50

【0179】

また、ユーザインターフェイス入力装置は、三次元（3D）マウス、ジョイスティックまたはポインティングスティック、ゲームパッドおよびグラフィックタブレット、およびスピーカなどのオーディオ/ビジュアル装置、デジタルカメラ、デジタルカムコーダ、携帯型メディアプレーヤ、ウェブカム、画像スキャナ、指紋スキャナ、バーコードリーダ3Dスキャナ、3Dプリンタ、レーザレンジファインダ、および視線検出装置を含み得るが、これらに限定されるものではない。また、ユーザインターフェイス入力装置は、たとえば、コンピュータ断層撮影、磁気共鳴画像化、位置発光断層撮影、医療用超音波検査装置などの医療用画像化入力装置を含み得る。また、ユーザインターフェイス入力装置は、たとえばMIDIキーボード、デジタル楽器などのオーディオ入力装置を含み得る。

【0180】

ユーザインターフェイス出力装置は、ディスプレイサブシステム、表示灯、またはオーディオ出力装置などの非視覚的ディスプレイなどを含み得る。ディスプレイサブシステムは、陰極線管（cathode ray tube：CRT）、液晶ディスプレイ（liquid crystal display：LCD）またはプラズマディスプレイを使用するものなどのフラットパネルディスプレイ、投影装置、タッチスクリーンなどであってもよい。一般に、「出力装置」という用語の使用は、コンピュータシステム1000からの情報をユーザまたは他のコンピュータに出力するための全ての可能なタイプの装置および機構を含むよう意図されている。たとえば、ユーザインターフェイス出力装置は、モニタ、プリンタ、スピーカ、ヘッドホン、自動車のナビゲーションシステム、プロッタ、音声出力装置およびモデムなどの、テキスト、グラフィックスおよびオーディオ/ビデオ情報を視覚的に伝えるさまざまな表示装置を含み得るが、これらに限定されるものではない。

【0181】

コンピュータシステム1000は、現在のところシステムメモリ1010内に位置しているように示されているソフトウェア要素を備える記憶サブシステム1018を備え得る。システムメモリ1010は、処理ユニット1004上でロード可能および実行可能なプログラム命令と、これらのプログラムの実行中に生成されるデータとを格納し得る。

【0182】

コンピュータシステム1000の構成およびタイプに応じて、システムメモリ1010は、揮発性（ランダムアクセスメモリ（random access memory：RAM）など）であってもよく、および/または、不揮発性（リードオンリメモリ（read-only memory：ROM）、フラッシュメモリなど）であってもよい。一般に、RAMは、処理ユニット1004がすぐにアクセス可能なデータおよび/またはプログラムモジュール、および/または、処理ユニット1004によって現在動作および実行されているデータおよび/またはプログ

ラムモジュールを収容する。いくつかの実現例では、システムメモリ1010は、スタティックランダムアクセスメモリ（static random access memory：SRAM）またはダイナミックランダムアクセスメモリ（dynamic random access memory：DRAM）などの複数の異なるタイプのメモリを含み得る。いくつかの実現例では、始動中などにコンピュータシステム1000内の要素間で情報を転送することを助ける基本ルーチンを含む基本入力/出力システム（basic input/output system：BIOS）が、一般にROMに格納され得る。一例としておよび非限定的に、システムメモリ1010は、クライアントアプリケーション、ウェブブラウザ、中間層アプリケーション、リレーショナルデータベース管理システム（relational database management system：RDBMS）などを含み得るアプリケーションプログラム1012、プログラムデータ1014およびオペレーティングシステム1016も示す。一例として、オペレーティングシステム1016は、マイクロソフトウィンドウズ（登録商標）、アップルマッキントッシュ（登録商標）および/もしくはリナックスオペレーティングシステムのさまざまなバージョン、さまざまな市販のUNIX（登録商標）もしくはUNIXライクオペレーティングシステム（さまざまなGNU/リナックスオペレーティングシステム、Google Chrome（登録商標）OSなどを含むが、これらに限定されるものではない）、ならびに/または、iOS、ウィンドウズ（登録商標）フォン、アンドロイド（登録商標）OS、ブラックベリー（登録商標）1005およびパーム（登録商標）OSオペレーティングシステムなどのモバイルオペレーティングシステムを含み得る。

10

20

【0183】

また、記憶サブシステム1018は、いくつかの実施形態の機能を提供する基本的なプログラミングおよびデータ構造を格納するための有形のコンピュータ読取可能な記憶媒体を提供し得る。プロセッサによって実行されたときに上記の機能を提供するソフトウェア（プログラム、コードモジュール、命令）が記憶サブシステム1018に格納され得る。これらのソフトウェアモジュールまたは命令は、処理ユニット1004によって実行され得る。また、記憶サブシステム1018は、本発明に従って使用されるデータを格納するためのリポジトリを提供し得る。

【0184】

また、記憶サブシステム1000は、コンピュータ読取可能な記憶媒体1022にさらに接続可能なコンピュータ読取可能な記憶媒体リーダ1020を含み得る。ともおよび任意には、システムメモリ1010と組合せて、コンピュータ読取可能な記憶媒体1022は、コンピュータ読取可能な情報を一時的および/または永久に収容、格納、送信および検索するための記憶媒体に加えて、リモートの、ローカルの、固定されたおよび/または取外し可能な記憶装置を包括的に表わし得る。

30

【0185】

また、コードまたはコードの一部を含むコンピュータ読取可能な記憶媒体1022は、当該技術分野において公知のまたは使用される任意の適切な媒体を含み得て、当該媒体は、情報の格納および/または送信のための任意の方法または技術において実現される揮発性および不揮発性の、取外し可能および取外し不可能な媒体などであるが、これらに限定されるものではない記憶媒体および通信媒体を含む。これは、RAM、ROM、電子的消去・プログラム可能ROM（electronically erasable programmable ROM：EEPROM）、フラッシュメモリもしくは他のメモリ技術、CD-ROM、デジタル多用途ディスク（digital versatile disk：DVD）、または他の光学式記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置もしくは他の磁気記憶装置、または他の有形のコンピュータ読取可能な媒体などの有形の一時的なコンピュータ読取可能な記憶媒体を含み得る。また、これは、データ信号、データ送信などの無形の一時的なコンピュータ読取可能な媒体、または、所望の情報を送信するために使用可能でありかつ計算システム1100によってアクセス可能なその他の媒体を含み得る。

40

【0186】

一例として、コンピュータ読取可能な記憶媒体1022は、取外し不可能な不揮発性磁

50

気媒体から読取るまたは当該媒体に書込むハードディスクドライブ、取外し可能な不揮発性磁気ディスクから読取るまたは当該ディスクに書込む磁気ディスクドライブ、ならびに、CD ROM、DVDおよびブルーレイ（登録商標）ディスクまたは他の光学式媒体などの取外し可能な不揮発性光学ディスクから読取るまたは当該ディスクに書込む光学式ディスクドライブを含み得る。コンピュータ読取可能な記憶媒体1022は、ジップ（登録商標）ドライブ、フラッシュメモリカード、ユニバーサルシリアルバス（universal serial bus：USB）フラッシュドライブ、セキュアデジタル（secure digital：SD）カード、DVDディスク、デジタルビデオテープなどを含み得るが、これらに限定されるものではない。また、コンピュータ読取可能な記憶媒体1022は、フラッシュメモリベースのSSD、企業向けフラッシュドライブ、ソリッドステートROMなどの不揮発性メモリに基づくソリッドステートドライブ（solid-state drive：SSD）、ソリッドステートRAM、ダイナミックRAM、スタティックRAMなどの揮発性メモリに基づくSSD、DRAMベースのSSD、磁気抵抗RAM（magnetoresistive RAM：MRAM）SSD、およびDRAMとフラッシュメモリベースのSSDとの組合せを使用するハイブリッドSSDを含み得る。ディスクドライブおよびそれらの関連のコンピュータ読取可能な媒体は、コンピュータ読取可能な命令、データ構造、プログラムモジュールおよび他のデータをコンピュータシステム1000に提供し得る。

【0187】

通信サブシステム1024は、他のコンピュータシステムおよびネットワークとのインターフェイスを提供する。通信サブシステム1024は、他のシステムからデータを受信したり、コンピュータシステム1000から他のシステムにデータを送信するためのインターフェイスの役割を果たす。たとえば、通信サブシステム1024は、コンピュータシステム1000がインターネットを介して1つ以上の装置に接続することを可能にし得る。いくつかの実施形態においては、通信サブシステム1024は、（たとえば3G、4GまたはEDGE（enhanced data rates for global evolution）などの携帯電話技術、高度データネットワーク技術を用いて）無線音声および/またはデータネットワークにアクセスするための無線周波数（radio frequency：RF）トランシーバコンポーネント、WiFi（IEEE 802.11ファミリ標準または他のモバイル通信技術またはそれらの任意の組合せ）、全地球測位システム（global positioning system：GPS）レシーバコンポーネント、および/または、他のコンポーネントを含み得る。いくつかの実施形態においては、通信サブシステム1024は、無線インターフェイスに加えて、または無線インターフェイスの代わりに、有線ネットワーク接続（例えばイーサネット）を提供し得る。

【0188】

また、いくつかの実施形態においては、通信サブシステム1024は、コンピュータシステム1000を使用し得る1人以上のユーザを代表して、構造化されたおよび/または構造化されていないデータフィード1026、イベントストリーム1028、イベント更新1030などの形態で入力通信を受信し得る。

【0189】

一例として、通信サブシステム1024は、ツイッター（登録商標）フィード、フェースブック（登録商標）更新、リッチ・サイト・サマリ（Rich Site Summary：RSS）フィードなどのウェブフィードなどのデータフィード1026をリアルタイムでソーシャルメディアネットワークおよび/または他の通信サービスのユーザから受信し、および/または、1つ以上の第三者情報源からリアルタイム更新を受信するように構成され得る。

【0190】

また、通信サブシステム1024は、連続的なデータストリームの形態でデータを受信するように構成され得て、当該データは、連続的である場合もあれば本質的に明確な端部をもたない状態で境界がない場合もあるリアルタイムイベントのイベントストリーム1028および/またはイベント更新1030を含み得る。連続的なデータを生成するアプリケーションの例としては、たとえばセンサデータアプリケーション、金融ティッカ、ネッ

10

20

30

40

50

トワーク性能測定ツール（たとえばネットワークモニタリングおよびトラフィック管理アプリケーション）、クリックストリーム分析ツール、自動車交通モニタリングなどを含み得る。

【0191】

また、通信サブシステム1024は、構造化されたおよび/または構造化されていないデータフィールド1026、イベントストリーム1028、イベント更新1030などを、コンピュータシステム1000に結合された1つ以上のストリーミングデータソースコンピュータと通信し得る1つ以上のデータベースに出力するように構成され得る。

【0192】

コンピュータシステム1000は、手持ち式携帯機器（たとえばiPhone（登録商標）携帯電話、iPad（登録商標）計算タブレット、PDA）、ウェアラブル装置（たとえばGoogleグラス（登録商標）ヘッドマウントディスプレイ）、PC、ワークステーション、メインフレーム、キオスク、サーバラックまたはその他のデータ処理システムを含むさまざまなタイプのうちの1つであってもよい。

10

【0193】

コンピュータおよびネットワークの絶え間なく変化し続ける性質のために、図に示されているコンピュータシステム1000の説明は、特定の例として意図されているに過ぎない。図に示されているシステムよりも多くのまたは少ない数の構成要素を有する多くの他の構成が可能である。たとえば、ハードウェア、ファームウェア、（タブレットを含む）ソフトウェア、または組合せにおいて、カスタマイズされたハードウェアも使用されてもよく、および/または、特定の要素が実装されてもよい。さらに、ネットワーク入力/出力装置などの他の計算装置への接続が利用されてもよい。本明細書で提供される開示および教示に基づいて、当業者は、さまざまな実施形態を実現するための他の手段および/または方法を理解するであろう。言い換えれば、いくつかの実施形態においては、コンピュータシステム1000の構成は、全体として、第1の入力データストリームおよび第2の入力データストリームを少なくとも含む複数の入力データストリームを受取るための手段と；第1の入力データストリームについての第1のダイナミックデータタイプを生成するための手段と；第2の入力データストリームについての第2のダイナミックデータタイプを生成するための手段と；第1の入力データストリームと第2の入力データストリームとを結合して、第1のダイナミックデータタイプおよび第2のダイナミックデータタイプに少なくとも部分的に基づく結合データストリームを生成するための手段と；結合データストリームに関する連続クエリをコンピュータシステムによって処理してパターンを検出するための手段とを含み得る。

20

30

【0194】

図11は、本発明のさまざまな実施形態が実現され得る例示的なサービスプロバイダデバイスの簡略化されたブロック図を示す。サービスプロバイダデバイス1100のブロックは、本発明の原理を実行するために、ハードウェア、ソフトウェアまたはハードウェアとソフトウェアとの組合せで実現されてもよい。図11に記載されるブロックが、上述のとおり本発明の原理を実現するために、サブブロックに組合わされ得るかまたは分割され得ることが当業者によって理解される。したがって、この明細書中における記載は、この明細書中に記載される機能ブロックのいずれかの実現可能な組合せまたは分割またはさらなる定義をサポートし得る。

40

【0195】

図11に示されるように、サービスプロバイダデバイス1100は、入力データストリーム受取ユニット1101、第1のダイナミックデータタイプ生成ユニット1103、第2のダイナミックデータタイプ生成ユニット1105、結合データストリーム生成ユニット1107、およびパターン検出ユニット1109を含み得る。入力データストリーム受取ユニット1101は、第1の入力データストリームおよび第2の入力データストリームを少なくとも含む複数の入力データストリームを受取るように構成されてもよい。第1のダイナミックデータタイプ生成ユニット1103は、第1の入力データストリームについ

50

ての第1のダイナミックデータタイプを生成するように構成されてもよい。第2のダイナミックデータタイプ生成ユニット1105は、第2の入力データストリームについての第2のダイナミックデータタイプを生成するように構成されてもよい。結合データストリーム生成ユニット1107は、第1の入力データストリームと第2の入力データストリームとを結合して、第1のダイナミックデータタイプおよび第2のダイナミックデータタイプに少なくとも部分的に基づく結合データストリームを生成するように構成されてもよい。パターン検出ユニット1109は、結合データストリームに関する連続クエリを処理して、パターンを検出するように構成されてもよい。一実施形態においては、各々のユニットは、コンピュータプログラム命令を読み出すことによって、対応するプロセスを実行するプロセッサとして実現されてもよい。

10

【0196】

一実施形態においては、第1のダイナミックデータタイプ生成ユニット1103はさらに、第1の入力データストリームの第1の属性を、第2のデータストリーム中に存在しないものとして識別し、第1の属性についての第1のダイナミックデータタイプを生成するように構成されてもよい。

【0197】

一実施形態においては、第1のダイナミックデータタイプは、第1の入力データストリームの第1の属性に対応する第1のデータ値を格納するように構成されてもよい。

【0198】

一実施形態においては、第2のダイナミックデータタイプ生成ユニット1105はさらに、第2の入力データストリームの第2の属性を、第1のデータストリーム中に存在しないものとして識別し、第2の属性についての第2のダイナミックデータタイプを生成するように構成されてもよい。第2のダイナミックデータタイプは、第2の入力データストリームの第2の属性に対応する第2のデータ値を格納するように構成され得る。

20

【0199】

一実施形態においては、サービスプロバイダデバイス1100はさらに、共通属性識別ユニット1111および同種スキーマ生成ユニット1113を含み得る。共通属性識別ユニット1111は、共通属性を識別するように構成されてもよい。共通属性は、第1の入力データストリーム中に存在するものとして、かつ、第2の入力データストリーム中に存在するものとして識別され得る。同種スキーマ生成ユニット1113は、同種スキーマを生成するように構成されてもよい。同種スキーマは、第1の入力データストリームおよび第2の入力データストリームの1つ以上の属性の表現を含む。当該表現は、少なくとも共通属性、第1のダイナミックデータタイプおよび第2のダイナミックデータタイプを含む。

30

【0200】

一実施形態においては、同種スキーマは、ストリーム名識別子属性、第1の入力データストリームに関連付けられた第1のタイムスタンプ属性、または第2の入力データストリームに関連付けられた第2のタイムスタンプ属性のうち少なくとも1つを含み得る。

【0201】

一実施形態においては、結合データストリーム生成ユニット1107はさらに、同種スキーマによって識別される第1の入力データストリームからタブルの第1のセットを選択し、同種スキーマによって識別される第2の入力データストリームからタブルの第2のセットを選択し、結合データストリームを生成するためにタブルの第1のセットおよびタブルの第2のセットに関してサブクエリを処理するように構成されてもよい。

40

【0202】

一実施形態においては、パターンは、結合データストリームの分析に少なくとも部分的に基づいて検出されてもよい。パターンは、第1の入力データストリームにおける第1のイベントと、後続する第2の入力データストリームにおける第2のイベントとを識別する。

【0203】

50

図 1 1 に記載されるブロックは、上述のとおり本発明の原理を実現するためにサブブロックに組合せされ得るかまたは分割され得る。たとえば、第 1 の動的データタイプ生成ユニット 1 1 0 3 および第 2 の動的データタイプ生成ユニット 1 1 0 5 は、動的データタイプ生成ユニットとして一体化されてもよい。一体型動的データタイプ生成ユニットはまた、共通属性識別ユニット 1 1 1 1 および同種スキーマ生成ユニット 1 1 1 3 を組込むことができる。

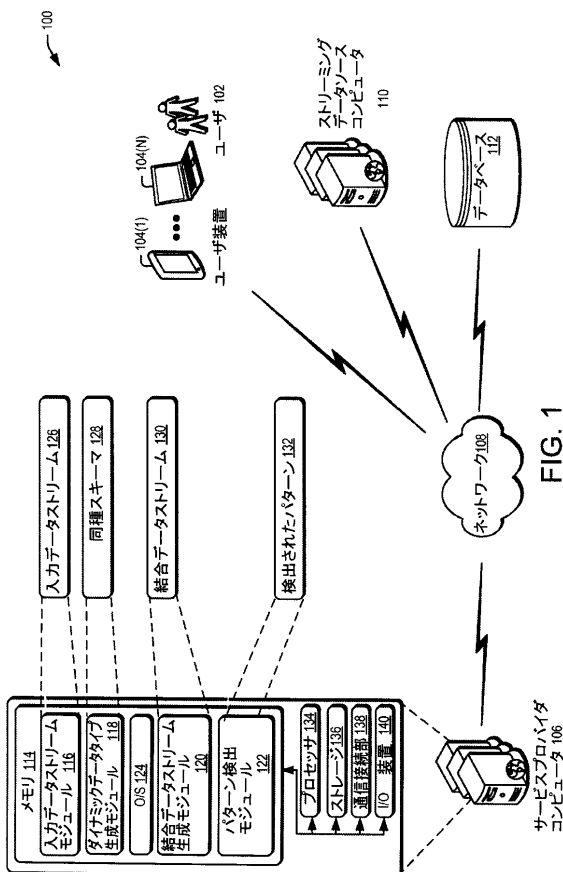
【 0 2 0 4 】

当業者であれば、サービスプロバイダデバイス 1 1 0 0 が、先の例示的な実施形態において記載されたサービスプロバイダコンピュータ 1 0 6 の例示的な実現例であり得ることを理解することができる。当業者であれば、また、サービスプロバイダデバイス 1 1 0 0 が、本発明の原則に従った動作またはその組合せのいずれをも実行するように必要に応じて変更され得ることを理解することができる。

【 0 2 0 5 】

上記の明細書では、本発明の局面は、その具体的な実施形態を参照して記載されているが、本発明はこれに限定されるものではないことを当業者は認識するであろう。上記の発明のさまざまな特徴および局面は、個々にまたは一緒に使用されてもよい。さらに、実施形態は、明細書のより広い精神および範囲から逸脱することなく、本明細書に記載されているものを越えたどのような環境およびアプリケーションでも利用可能である。したがって、明細書および図面は、限定的ではなく例示的であるものとみなされるべきである。

【 図 1 】



【 図 2 】

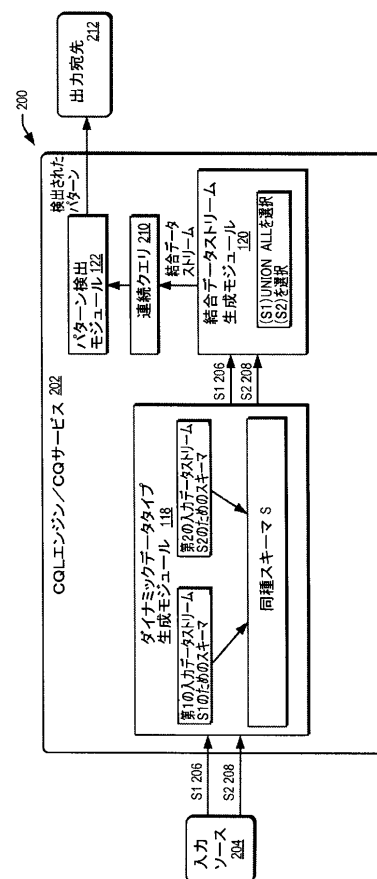


FIG. 2

【 図 3 】

```

クエリ 300
SELECT T.*
FROM (S1, S2)
MATCH_RECOGNIZE
(
  MEASURES
  $streamName as name
  A.c1 as Ac1,
  A.S1.c2 as Ac2,
  A.S2.c3 as Ac3,
  PATTERN (A)
  DEFINE
  A AS (1 = 1)
) as T

```

FIG. 3

【 図 4 】

```

クエリ 400
SELECT T.*
FROM
(
  (select "S1" as $streamName, c1, ELEMENT_TIME,
  Type1@DynamicTypeCartridge(c1, c2) AS S1, Type2@DynamicTypeCartridge(
  AS S2 from S1
  UNION ALL
  select "S2" as $streamName, c1, ELEMENT_TIME, Type1@DynamicTypeCartridge(
  ) as S1, Type2@DynamicTypeCartridge(c1, c3) AS S2 from S2) AS _SQ_ALIAS
  MATCH_RECOGNIZE
  (
    MEASURES
    $streamName as name
    A.c1 as Ac1,
    A.S1.c2 as Ac2,
    A.S2.c3 as Ac3,
    PATTERN (A)
    DEFINE
    A AS (1 = 1)
  ) as T

```

FIG. 4

【 図 5 】

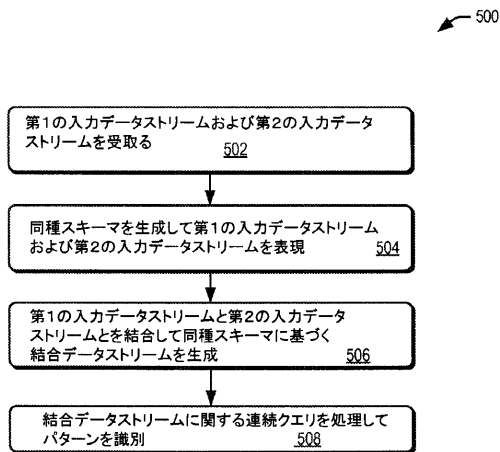


FIG. 5

【 図 6 】

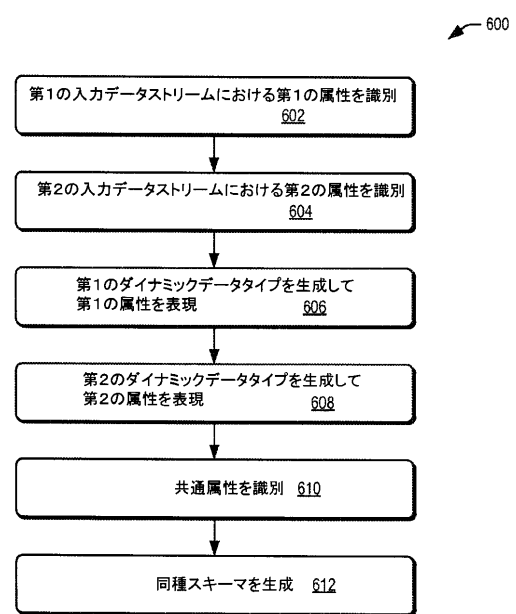


FIG. 6

【 図 7 】

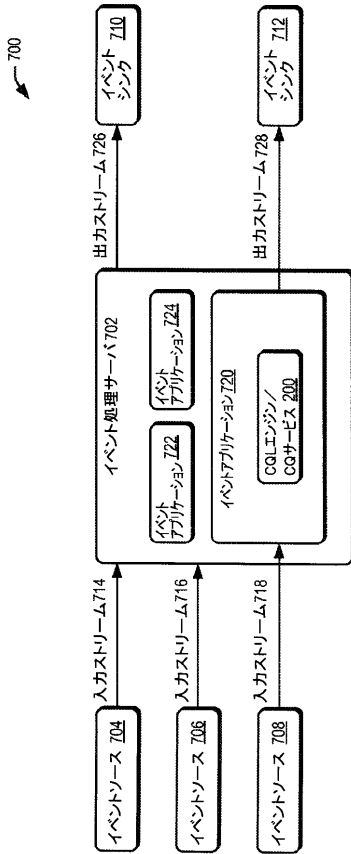


FIG. 7

【 図 8 】

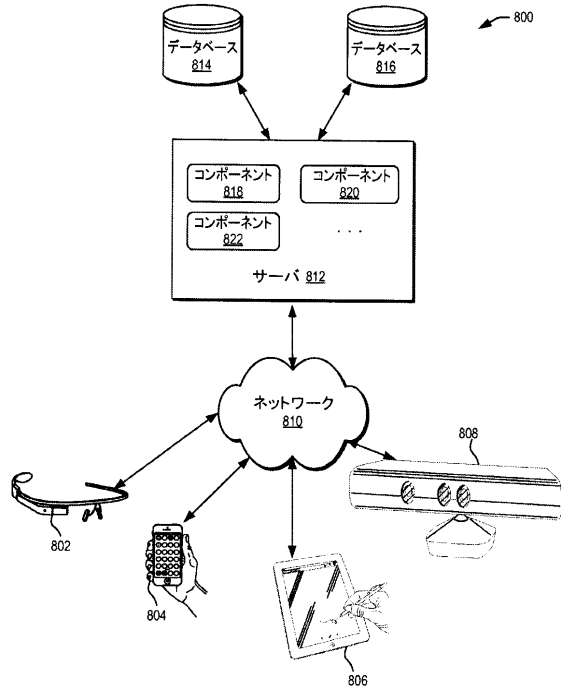


FIG. 8

【 図 9 】

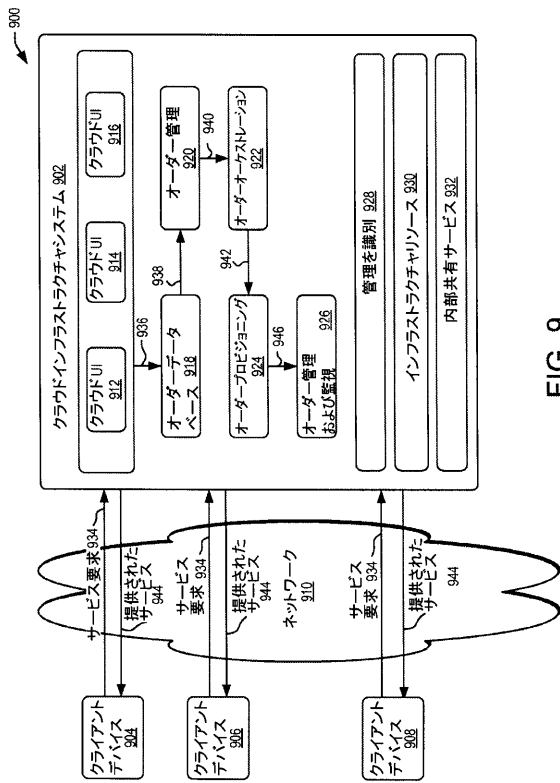


FIG. 9

【 図 10 】

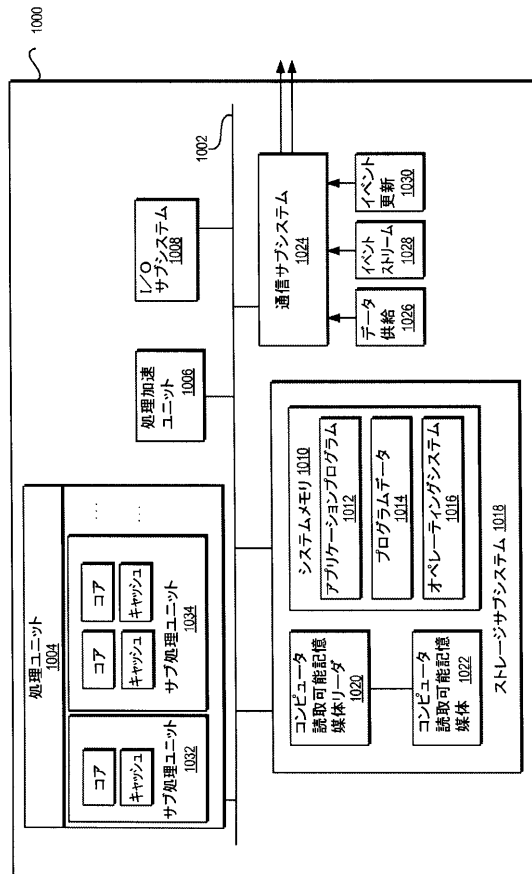


FIG. 10

【 図 1 1 】

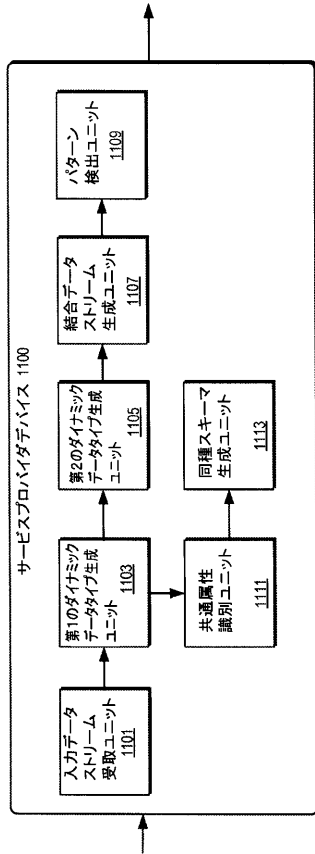


FIG. 11

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No PCT/US2014/068641

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F17/30 ADD.		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal, WPI Data		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012/324453 A1 (CHANDRAMOULI BADRISH [US] ET AL) 20 December 2012 (2012-12-20) paragraph [0001] - paragraph [0002] paragraph [0025] paragraph [0032] - paragraph [0039] paragraph [0054]	1-20
A	----- CHUCK CRANOR ET AL: "Gigascope", PROCEEDINGS OF THE 2003 ACM SIGMOD INTERNATIONAL CONFERENCE ON ON MANAGEMENT OF DATA , SIGMOD '03, 9 June 2003 (2003-06-09), - 9 June 2003 (2003-06-09), page 647, XP055170688, New York, New York, USA DOI: 10.1145/872836.872838 the whole document ----- -/--	1-20
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
19 February 2015		26/02/2015
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer
		de Castro Palomares

2

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2014/068641

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>ARVIND ARASU ET AL: "The CQL continuous query language: semantic foundations and query execution", THE VLDB JOURNAL ; THE INTERNATIONAL JOURNAL ON VERY LARGE DATA BASES, SPRINGER, BERLIN, DE, vol. 15, no. 2, 22 July 2005 (2005-07-22), pages 121-142, XP019431176, ISSN: 0949-877X the whole document</p> <p style="text-align: center;">-----</p>	1-20
A	<p>DANIEL J. ABADI ET AL: "Aurora: a new model and architecture for data stream management", THE VLDB JOURNAL THE INTERNATIONAL JOURNAL ON VERY LARGE DATA BASES, vol. 12, no. 2, 16 July 2003 (2003-07-16), pages 120-139, XP055009044, ISSN: 1066-8888, DOI: 10.1007/s00778-003-0095-z page 120 - page 122 page 130</p> <p style="text-align: center;">-----</p>	1-20

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2014/068641

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012324453 A1	20-12-2012	CA 2838966 A1	20-12-2012
		CN 103620584 A	05-03-2014
		EP 2721511 A2	23-04-2014
		JP 2014524175 A	18-09-2014
		KR 20140038462 A	28-03-2014
		US 2012324453 A1	20-12-2012
		WO 2012174023 A2	20-12-2012

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US

(72)発明者 スリニバサン, アナンド
インド、560076 カルナータカ、バンガロール、パナーガッタ・ロード、エヌ・エス・パ
リヤ、マントリ・エレガンス、イー・303

(72)発明者 ビシュノイ, サンディーブ
インド、125055 ハリヤーナ、シルサ、バーナラ・ロード、コロニー、29-エム・アイ・
ティ・シイ