

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6577588号  
(P6577588)

(45) 発行日 令和1年9月18日 (2019.9.18)

(24) 登録日 令和1年8月30日 (2019.8.30)

(51) Int. Cl.

F I

G O 6 F 16/21 (2019.01)

G O 6 F 16/21

G O 6 F 16/24 (2019.01)

G O 6 F 16/24

G O 6 F 16/903 (2019.01)

G O 6 F 16/903

請求項の数 21 (全 28 頁)

(21) 出願番号 特願2017-533723 (P2017-533723)  
 (86) (22) 出願日 平成27年9月14日 (2015.9.14)  
 (65) 公表番号 特表2017-532702 (P2017-532702A)  
 (43) 公表日 平成29年11月2日 (2017.11.2)  
 (86) 国際出願番号 PCT/US2015/049876  
 (87) 国際公開番号 W02016/044114  
 (87) 国際公開日 平成28年3月24日 (2016.3.24)  
 審査請求日 平成30年9月10日 (2018.9.10)  
 (31) 優先権主張番号 14/486,638  
 (32) 優先日 平成26年9月15日 (2014.9.15)  
 (33) 優先権主張国・地域又は機関  
 米国 (US)

(73) 特許権者 314015767  
 マイクロソフト テクノロジー ライセン  
 シング、エルエルシー  
 アメリカ合衆国 ワシントン州 9805  
 2 レッドモンド ワン マイクロソフト  
 ウェイ  
 (74) 代理人 100140109  
 弁理士 小野 新次郎  
 (74) 代理人 100118902  
 弁理士 山本 修  
 (74) 代理人 100106208  
 弁理士 宮前 徹  
 (74) 代理人 100120112  
 弁理士 中西 基晴

最終頁に続く

(54) 【発明の名称】 向上したイベント処理のための構築されたデータストリーム

(57) 【特許請求の範囲】

【請求項 1】

向上したイベント処理のためのコンピューターが実施する方法であって、

向上したイベント処理のリクエストを受けるステップであって、前記向上したイベント  
 処理のリクエストは、データベースデータのソースからの入力データの指示を含む、ステ  
 ップと、

構築されたデータストリームのソースを使用して、入力データの前記指示における前記  
 入力データにアクセスするステップであって、前記構築されたデータストリームのソース  
 は、前記データベースデータのソースのデータベースデータの経時的なスナップショット  
 のコレクションへのアクセスを提供し、前記入力データは、前記データベースデータのソ  
 ースに対するデータストリームのシミュレーションプロキシとして機能する前記構築され  
 たデータストリームのソースを使用してアクセスされ、前記構築されたデータストリーム  
 は、スナップショットの前記コレクションから、前記構築されたデータストリームが、該  
 構築されたデータストリームのイベントに対応する時間を指示する時間要素を含むように  
 構築される、ステップと、

前記構築されたデータストリームを使用してアクセスされた前記入力データに基づき、  
 前記向上したイベント処理のリクエストに対する出力を生成するステップと  
 を含む方法。

【請求項 2】

請求項 1 に記載の方法であって、前記データベースデータのソースからの前記入力データ

10

20

は、前記入力データに対する時間要素と、前記データベースデータのソースがデータベースデータをリレーショナルデータベースのテーブルにおいて維持するという点で関連しない、方法。

【請求項 3】

請求項 1 に記載の方法であって、前記向上したイベント処理のリクエストにおける前記入力データは、前記構築されたデータストリームを介してアクセスされる、スナップショットの前記コレクションにおける選択されたスナップショットにおける選択された値及び選択された時間に対応する、方法。

【請求項 4】

請求項 1 に記載の方法であって、前記構築されたデータストリームの前記時間要素は、前記構築されたデータストリームのイベントが起きる時間から求められ、前記イベントのスナップショットの一部として前記データベースデータにおいて記憶される、方法。

10

【請求項 5】

請求項 1 に記載の方法であって、前記構築されたデータストリームは、経時変化する前記データベースデータを使用して反復可能な計算を提供し、前記構築されたデータストリームは、選択された値及び選択された時間の関数としてアクセスされる、方法。

【請求項 6】

請求項 1 に記載の方法であって、前記向上したイベント処理のリクエストは、別のデータストリームからの入力データの第 2 の指示を更に含む、方法。

【請求項 7】

20

請求項 6 に記載の方法であって、

前記構築されたデータストリームの第 1 の時間要素を前記別のデータストリームの第 2 の時間要素と関連させるステップと、

前記向上したイベント処理のリクエストに対する出力を、前記構築されたデータストリームの前記入力データ及び前記別のデータストリームの前記入力データに基づき生成するステップと

を更に含む方法。

【請求項 8】

請求項 1 に記載の方法であって、前記出力は、妥当性時間要素と関連し、前記妥当性時間要素は、前記構築されたデータストリームに基づく前記出力が前記データベースデータにおけるデータを正確に反映した時間を指示する、方法。

30

【請求項 9】

1 つまたは複数のプロセッサに、向上したイベント処理のための方法を実施させるコンピュータプログラムであって、前記方法は、

データベースデータのソースを識別するステップと、

前記データベースデータのソースのデータベースデータの少なくとも一部を、前記データベースデータの 1 つまたは複数のスナップショットを生成するためにサンプリングするステップと、

構築されたデータストリームを、前記データベースデータの前記 1 つまたは複数のスナップショットに基づき生成するステップであって、前記構築されたデータストリームは時間要素に関連する、ステップと、

40

前記構築されたデータストリームへの、向上したイベント処理のリクエストに基づくアクセスを提供するステップであって、前記向上したイベント処理のリクエストは、前記構築されたデータストリームを使用してアクセス可能な、前記データベースデータのソースのデータベースにおける入力データの指示に関連する、ステップと

を含む、コンピュータプログラム。

【請求項 10】

請求項 9 に記載のコンピュータプログラムであって、前記データベースデータをサンプリングするステップは定期的なサンプリングメカニズムに基づき、前記定期的なサンプリングメカニズムは、1 つまたは複数のスナップショットを、所定の時間間隔に基づき生成

50

する、コンピュータプログラム。

【請求項 1 1】

請求項 9 に記載のコンピュータプログラムであって、前記データベースデータをサンプリングするステップは差分サンプリングメカニズムに基づき、前記差分サンプリングメカニズムは、前記データベースデータに対する変更のみ含む 1 つまたは複数のスナップショットを生成する、コンピュータプログラム。

【請求項 1 2】

請求項 9 に記載のコンピュータプログラムであって、前記データベースデータをサンプリングするステップは通知メカニズムに基づき、前記通知メカニズムは、前記データベースデータに対する変更の指示のときにスナップショットが生成されるように、前記データベースデータに対する変更が生じたときを指示する、コンピュータプログラム。

10

【請求項 1 3】

請求項 1 2 に記載のコンピュータプログラムであって、前記通知メカニズムは計量機構を更に含み、前記計量機構は、サンプリングを実行するための通知をトリガする前の、変更の所定数を決定するかまたは所定期間を決定する、コンピュータプログラム。

【請求項 1 4】

請求項 1 2 に記載のコンピュータプログラムであって、前記通知メカニズムは管理時間要素に基づき、前記管理時間要素は、前記データベースデータの操作ベースの機能性に関連し、前記管理時間要素は、前記データベースに記録されない、コンピュータプログラム。

20

【請求項 1 5】

請求項 1 4 に記載のコンピュータプログラムであって、前記管理時間要素は、スナップショットに関連する時間を指示し、前記スナップショットは、前記構築されたデータストリームを、前記管理時間要素を用いて生成するために使用される、コンピュータプログラム。

【請求項 1 6】

向上したイベント処理を実施するためのコンピュータシステムであって、

データベースデータへのアクセスを提供するように構成されたデータベースデータのソースの構成要素と、

構築されたデータストリームのソースの構成要素であって、

30

データベースデータのソースを識別し、

前記データベースデータのソースのデータベースデータの少なくとも一部を、前記データベースデータの 1 つまたは複数のスナップショットを生成するためにサンプリングし、

構築されたデータストリームを、前記データベースデータの前記 1 つまたは複数のスナップショットに基づき生成し、前記構築されたデータストリームは、時間要素に関連し、

前記構築されたデータストリームへの、向上したイベント処理のリクエストに基づくアクセスを提供し、前記向上したイベント処理のリクエストは、前記構築されたデータストリームを使用してアクセス可能な、前記データベースデータのソースのデータベースにおける入力データの少なくとも指示に関連する

40

ように構成された構築されたデータストリームのソースの構成要素と、

別のデータストリームへのアクセスを提供する別のデータストリームのソースの構成要素と、

向上したイベント処理のサーバーであって、

前記向上したイベント処理のリクエストを受け、前記向上したイベント処理のリクエストは、前記データベースデータのソースからの入力データの第 1 の指示と、前記別のデータストリームのソースからの入力データの第 2 の指示とを含み、

入力データの前記第 1 の指示における前記入力データに、前記構築されたデータストリームのソースを使用してアクセスし、前記構築されたデータストリームのソースは、前

50

記データベースデータのソースのデータベースデータの、経時的なスナップショットのコレクションへのアクセスを提供し、前記入力データは、第1の時間要素を含む前記構築されたデータストリームを使用してアクセスされ、

入力データの前記第2の指示における入力データに、前記別のデータストリームのソースを使用してアクセスし、

前記構築されたデータストリームの前記第1の時間要素を、前記別のデータストリームの第2の時間要素と関連させ、

前記向上したイベント処理のリクエストに対する出力を、前記構築されたデータストリームの前記入力データと、前記別のデータストリームの前記入力データとに基づき生成する

10

ように構成された向上したイベント処理のサーバーとを含むコンピューターシステム。

【請求項17】

請求項16に記載のコンピューターシステムであって、前記向上したイベント処理のサーバーはロジックモジュールを含み、前記ロジックモジュールは、

出力を前記向上したイベント処理のリクエストに基づき生成するためのロジックを識別し、前記ロジックは、

最新のスナップショットロジックと、

補間ロジックと

のうちの少なくとも1つから選択され、

20

前記ロジックを、前記出力が前記ロジックに基づき生成されるように提供し、前記ロジックは、前記構築されたデータストリームと下位互換である

ように構成された、コンピューターシステム。

【請求項18】

請求項16に記載のコンピューターシステムであって、前記向上したイベント処理のサーバーは操作モジュールを含み、前記操作モジュールは、

前記向上したイベント処理のリクエストを、データベースにおける入力データの指示と共に受け、前記入力データは、時間要素に関連せず、

前記入力データに、構築されたデータストリームを使用してアクセスし、前記構築されたデータストリームは、時間要素に関連した前記入力データを含む

30

ように構成された、コンピューターシステム。

【請求項19】

請求項16に記載のコンピューターシステムであって、前記向上したイベント処理のサーバーは妥当性時間モジュールを含み、前記妥当性時間モジュールは、

妥当性時間要素を提供し、前記妥当性時間要素は、前記構築されたデータストリームに基づく前記出力が前記データベースデータにおけるデータを正確に反映した時間を指示する

ように構成された、コンピューターシステム。

【請求項20】

請求項19に記載のコンピューターシステムであって、前記妥当性時間要素は、向上したイベント処理のリクエストの処理を進めるように加速される、コンピューターシステム。

40

【請求項21】

請求項9から15のうちの何れか一項に記載のコンピュータープログラムを記憶した1つまたは複数のコンピューター可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

[0001] イベント処理は、情報のストリームを追跡して分析し、その情報から結論を導き出すことを言う。

【背景技術】

50

## 【 0 0 0 2 】

情報のストリームは、具体的にイベントについてのものである場合もある。また、イベント処理は、複数のデータソースからのデータを互いに組み合わせて、より複雑な状況を示唆するイベントまたはパターンを推論するイベント処理を含む。そのように、イベント処理によって、有意味なイベントを識別し、できるだけ迅速にそれらイベントに応じることができる。

## 【 発 明 の 概 要 】

## 【 0 0 0 3 】

[0002]この要約は、以下において詳細な記述でさらに述べる概念の選択を簡単化された形態で紹介するために設けられている。この要約は、主張する主題のキー特徴または基本的な特徴を識別するように意図されず、また主張する主題の範囲を決定する助けとして分離して使用されることも意図されない。

## 【 0 0 0 4 】

本明細書に述べる実施形態は、向上したイベント処理のための方法、システムおよびコンピュータ記憶媒体を対象とする。向上したイベント処理のリクエストを受け取り得、向上した処理のリクエストは、出力データを導き出すために使用されるデータベースデータのソースからの入力データの指標を含む。向上したイベント処理のリクエストに示された入力データは、データベースデータのソースではなく、構築されたデータストリームのソースを使用してアクセスされる。構築されたデータストリームデータのソースは、データベースデータのソースのデータベースデータの経時的なスナップショットのコレクションへのアクセスを提供する。構築されたデータストリームのソースからのデータは、選択された値および選択された時間の関数としてアクセスし得る。構築されたデータストリームデータは、時間要素を含む。構築されたデータストリームデータは、時間要素を含み、それは、データベースデータで利用できず、時間要素によって、構築されたデータストリームデータを使用して、データベースデータのための向上したイベント処理の実施が促進される。向上したイベント処理のリクエストに対する出力が、構築されたデータストリームデータを使用してアクセスされる入力データに基づき生成される。

## 【 0 0 0 5 】

向上したイベント処理は、データストリームのソースと構築されたデータストリームのソースを組み合わせるステップをさらに含んでもよく、構築されたデータストリームのソースは、データベースデータのソースのためのプロキシとして働く。向上したイベント処理のリクエストは、データベースデータのソースからの入力データの第1の指標とデータストリームのソースからの入力データの第2の指標との両方を含むことができる。この点で、入力データの第1の指標で示された入力データが、構築されたデータストリームのソースを介してアクセスすることができ、第2の指標で示された入力データが、データストリームのソースを介してアクセスすることができ、入力データは、構築されたデータストリームの第1の時間要素およびデータストリームの第2の時間要素を使用して互いに組み合わせて関連させ、向上したイベント処理のリクエストに対して導き出される出力を生成する。実施形態では、妥当性時間メカニズムが、向上したイベント処理のリクエストに対する矛盾しない出力を示すためのルーチンおよびプロシージャールを提供する。具体的には、妥当性時間要素は、1つまたは複数のデータソースに基づく出力が、その時間におけるデータソース中のそのデータを正確に反映している時間を示す。

## 【 0 0 0 6 】

[0003]以下で本発明について添付図面を参照して詳細に述べる。

## 【 図 面 の 簡 単 な 説 明 】

## 【 0 0 0 7 】

【 図 1 】 [0004]本明細書に述べる実施形態を実施する上で使用するのに適切な例示的な計算環境のブロック図である。

【 図 2 】 [0005]本明細書に述べる実施形態が用いられることが可能な例示的な動作環境のブロック図である。

10

20

30

40

50

【図3】[0006]本明細書に述べる実施形態による、向上したイベント処理を実施するための例示的な方法を示すフロー図である。

【図4】[0007]本明細書に述べる実施形態による、向上したイベント処理を実施するための例示的な方法を示すフロー図である。

【図5】[0008]本明細書に述べる実施形態による、向上したイベント処理を実施するための例示的な方法を示すフロー図である。

【発明を実施するための形態】

【0008】

[0009]本発明の実施形態の主題は、法定要件を満たすように、本明細書で具体的に述べる。しかし、記述自体は、本特許の範囲を限定する意図はない。むしろ、発明者等は、主張する主題が、他の現在または将来の技術と連動して、異なるステップ、または本文書に述べるステップと同様のステップの組み合わせを含む他の方法でも具現化できる可能性があるはずであると予期している。さらにまた、用いる方法の異なる要素を含意するために、本明細書では用語「ステップ」および/または「ブロック」を使用することがあるが、用語は、本明細書に述べる個々のステップの順序が明白に述べられていない限り、および明白に述べられているときを除き、様々なステップの間でいずれかの特定の順序を暗示するものとして解釈すべきでない。

【0009】

[0010]本開示のために、言葉「含む(including)」は、言葉「含む(comprising)」と同じ広い意味を持つ。さらに、「1つ(a)」および「1つ(an)」などの言葉は、それと反対に規定されていない限り、複数形、さらにまた単数形を含む。それゆえ、たとえば、1つまたは複数の特徴が存在するところでは、「1つの特徴」の要求が満たされている。また、用語「または(or)」は、接続、分離およびその両方を含む(それゆえ、aまたはbは、aまたはbのどちらか、aおよびb、の両方を含む)。

【0010】

[0011]以下の詳細な議論のために、実施形態は、データセンター中にインラックハードウェアのインベントリをセットアップすること、および例として、ブートストラッピングの前にハードウェアのインベントリをセットアップすることを参照して述べる。しかし、セットアップするワークフローは、様々な異なるタイプのハードウェアサービスのために、新しいインラックハードウェアのインベントリをセットアップするために使用されることがあり、データセンターのブートストラッピングは、単なる例示的なものである。述べる技法は、他の実装形態の文脈に拡張できる可能性があると考えられる。

【0011】

[0012]イベント処理は、情報のストリームを追跡して分析し、それらの情報から結論を導き出すことを言う。情報のストリームは、具体的にイベントについてのものである場合もある。また、イベント処理は、複数のデータソースからのデータを互いに組み合わせて、より複雑な状況を示唆するイベントまたはパターンを推論するというイベント処理を含む。そのように、イベント処理によって、有意義なイベント(たとえば好機や脅威)を識別し、できるだけ迅速にそれらイベントに応じることを容易にすることができる。イベントは、一般に、セールスリード、注文または顧客のサービス依頼の電話、新しいアイテム、テキストのメッセージ、ソーシャルメディアの投稿、株式市場フィード、トラフィックレポート、天気予報、モノのインターネットのデータ、または他の種類のデータを含み得る。

【0012】

[0013]イベント処理データソースは、データストリームおよびデータベース中のデジタル化データを含むことができる。例示的なシナリオでは、データストリームの処理は、識別された基準に基づき、イベントをフィルタで除去する、または集約するステップを含むことができる。他のタイプの処理を、異なるイベントに基づき実施することができる。また、イベント処理は、異なるデータストリームのソースからの複数の受信データのセット

を分析するステップに基づくこともできる。

【0013】

[0014]また、イベント処理データソースは、データベースとしてもよい。データベースは、データの系統的なコレクションである。データは、通常、処理要求をサポートするような形で現実の諸状況をモデル化するように系統化される。たとえば、誰が顧客であり、顧客がどこに住んでいるのか（たとえば顧客名、顧客住所）を知るのをサポートするように顧客トランザクションがモデル化される。データベース管理システムは、ユーザーと情報をやり取りするソフトウェアアプリケーションと、データをファイルに保存し、分析するための他のアプリケーションとの中にデータベーステーブル（たとえば、リレーショナルデータベーステーブル）を維持する。

10

【0014】

[0015]データベースデータは、具体的に、関連した、または分類されたグローバルデータのセットを含む基準データを含んでもよい。基準データは、データベーステーブル中に格納されるイベントと関連付けられるデータとしてもよい。基準データは、信頼できるパブリックドメインにおいて、またはプレミアム商用コンテンツプロバイダーから入手することができる。また、基準データは、他のデータフィールドによって使用される許容値のセットを含むことができる。

【0015】

[0016]データベーステーブルとの情報のやり取りは、データベーステーブルと本質的に関連するロジックを利用する宣言型フレームワークを使用して実施することができる。宣言型フレームワークは、制御フロー（すなわち個々のステートメント、命令または関数の呼び出しが実行される順番）の記述を最小にできる可能性がある。この点で、データベーステーブルとの情報のやり取りは、単にどのデータがデータベーステーブルから必要であるのかを示し、テーブルからのデータの獲得にどのように取り掛かるのかは示さないことによって容易にすることができる可能性がある。例として、リレーショナルデータベース中に保持されたデータを管理するために、SQL（構造化クエリー言語（Structured Query Language））を使用してもよい。

20

【0016】

[0017]基準データを含むデータベースデータは、ストリームイベントをモデル化しない。たとえば、データベースデータは、時間データ（たとえば、時間などの時間要素）として構成されず、それゆえデータストリームのイベントの時間要素に依存するデータストリーム操作は、データベースデータに基づいては実施することができない。そのように、データベースデータ、またはデータベースデータとデータストリームデータの組み合わせに基づくイベント処理操作は、データベースデータのソースのみと関連する、またはデータストリームのソースと組み合わせた、向上したイベント処理のリクエストのための異なるフレームワークを有利に利用することができる。データベースデータ、データストリームおよびその組み合わせの向上したイベント処理を定義するフレームワークがないと、データストリームに基づき実施することができる操作は、データベースデータに基づき実施することができない、それゆえ、これら2つのデータソースからイベントおよびパターンを推論するための能力が限定される。

30

40

【0017】

[0018]代替の解決策は、データベース中へのデータ入力に関するタイムスタンプを積極的に追跡するステップを含んでもよい。たとえば、データベース中への各データ入力に関する開始時間および終了時間である。しかし、そのような実装形態は、通常、格納することが必要な追加されるデータ量のために非常に高費用になり、ひいてはリレーショナルデータベースの効率を低下させる。さらに、代替解決策は、簡単なターンキー方式の解決策でなく、下位互換性を欠いている。というのは、データがデータベーステーブル中に取り込まれた状態では、データベースデータを使用するイベント処理のためのロジックが、データベースデータとともに働くようには構成されないからである。時々、データベーステーブルは、データベースデータを用いてイベント処理を実施するために、完全に移植する

50

必要がある場合がある。

【 0 0 1 8 】

[0019]したがって、本発明の実施形態は、構築されたデータストリームのソースに基づく、向上したイベント処理のための簡単で効率の良い方法およびシステムを提供する。向上したイベント処理のリクエストは、出力データを導き出すために、生成して伝達できる可能性がある。向上したイベント処理のリクエストは、データベースデータのソースからアクセスされるべき入力データを示す。向上したイベント処理は、構築されたデータストリームのソースを使用して入力データへのアクセスを提供する。構築されたデータストリームのソースは、構築されたデータストリームのソースからのデータに選択された値および選択された時間の関数としてアクセスすることができるように、経時的に取られるデータベースデータのスナップショットのコレクションへのアクセスを提供する。

10

【 0 0 1 9 】

[0020]入力データは、時間要素を有する構築されたデータストリームとしてアクセスされる。構築されたデータストリームデータは、時間要素を含み、時間要素は、データベースデータのソースでは入手できず、時間要素によって、構築されたデータストリームを使用したデータベースデータに対する向上したイベント処理の実施が促進される。そのように、データベースデータを使用して実施することができないはずであるデータストリーム操作が、その代わりに構築されたデータストリームデータを使用すると実施することができる。向上したイベント処理のリクエストに対する出力が、構築されたデータストリームデータに基づき生成される。構築されたデータストリームは、選択された値および選択された時間に基づきルックアップリクエストを実施するために使用することができる。この点で、構築されたデータストリームは、クエリーに基づきデータベースデータを取り出すために使用することができる。

20

【 0 0 2 0 】

[0021]向上したイベント処理は、少なくとも1つのデータストリームのソースと少なくとも1つの構築されたデータストリームのソースを組み合わせるステップをさらに含んでもよく、少なくとも1つの構築されたデータストリームのソースは、データベースデータのソースのためのプロキシとして働く。この点で、向上したイベント処理のリクエストは、データベースデータのソースからの入力データ（たとえば基準データ）の第1の指標とデータストリームのソースからの入力データ（たとえばイベントデータ）の第2の指標の両方を含むことができる。入力データの第1の指標で示された入力データは、構築されたデータストリームのソースを介してアクセスすることができ、第2の指標で示された入力データは、データストリームのソースを介してアクセスすることができる。入力データは、構築されたデータストリームデータの第1の時間要素およびデータストリームデータの第2の時間要素を使用して互いに組み合わせられて関連付けられ、それによって向上したイベント処理のリクエストに対する、導き出された出力が生成される。データストリームと構築されたデータストリームは、イベント処理操作を実施する上でより容易に関連させることができる。

30

【 0 0 2 1 】

[0022]構築されたデータストリームを用いるイベント処理は、再現性を保証するように構成される。再現性または繰り返し可能な計算は、同じデータソースを用いて、その後でイベント処理分析を再実行し、同じ結果が生成されることを言う。たとえば、第1の時間に生のデータストリームに基づき実行されたクエリーが、第1の時間の後の第2の時間に実行されたとき、全く同じ結果が生成される。構築されたデータストリームおよびデータストリームを用いると、たとえ構築されたデータストリームがそれから生成されるデータベースデータが変更されたとしても、両方のストリームからの入力データを組み合わせるクエリーが、同じ結果を生成する。動作時には、時間要素（たとえばアプリケーション時間）が、データベースデータと一致するように識別されて、構築されたデータストリームが生成される。構築されたデータストリームは、時間要素を有する構築されたデータストリームが、入力されたクエリーを実行するために、アクセスすることができるように、デ

40

50



ータベースデータのサンプリングされたスナップショットを含む。再現性があると、イベント処理は、同じ入力データに対して実行されるクエリーの間で比較するステップを含むことができる。クエリーは、クエリーの性質を評価して最適化し、向上したクエリーを生成するためにクエリーの代替バージョンを構築し、クエリーの変形の間で結果を比較し、またはより多くの情報を得るためにクエリーの代替バージョンをランさせるために、互いに比較してもよい。

【 0 0 2 2 】

[0023]向上したイベント処理のフレームワーク（「向上したフレームワーク」）が、構築されたデータストリーム、追加のモジュール／構成要素およびメカニズム（たとえばルーチンおよびプロシージャ）をさらに含み、それらは、構築されたデータストリームとしてのデータストリームおよびデータベースデータに基づきイベント処理を実施するためのサポートシステムを形成する。実施形態では、妥当性時間メカニズムが、向上したイベント処理のリクエストに対して矛盾しない出力を示すためのルーチンおよびプロシージャをもたらす。具体的には、妥当性時間要素は、1つまたは複数のデータソースに基づく出力がその時間におけるデータソース中のそのデータを正確に反映している時間を示す。この点で、本明細書に述べる実施形態は、時間状態クエリーの妥当性および性能を維持しながら、履歴クエリーを有利に可能にすることができる。

【 0 0 2 3 】

[0024]様々な用語が本記述全体を通じて使用される。様々な用語に関するより多くの詳細が本記述全体を通じて提示されているが、いくつかの用語の一般的な定義は、本明細書に開示するアイデアのより明確な理解をもたらすために以下に含まれている。

【 0 0 2 4 】

[0025]向上したイベント処理のリクエストは、時間要素と関連付けられていないが、その代わりに時間要素を有する構築されたデータストリームを使用してアクセスされる、少なくともデータベースデータを含む1つまたは複数の入力を含むリクエストを言う。また、向上したイベント処理は、出力を生成するために互いに組み合わせられるデータベースデータとデータストリームの両方に対する入力を含んでもよい。

【 0 0 2 5 】

[0026]データベースデータは、データの系統的なコレクションを言う。データベースデータは、具体的に、関連付けられた、または分類されたグローバルデータのセットを含む基準データを含んでもよい。データベースデータは、データベース中のデータと関連する時間要素を有さないことがある。時間要素は、本明細書で使用する時、データベーステーブル中への入力と関連するイベントが生じたときの特定の時間を言うことがある。

【 0 0 2 6 】

[0027]データストリームは、データストリーム中のデジタル化データを言う。イベント処理は、異なるデータストリームのソースからの複数の受信データのセットを分析することに基づくことができる。そのように、イベント処理によって、有意なイベント（たとえば好機や脅威）を識別し、できるだけ迅速にそれらイベントに応じることを促進することができる。イベントは、一般に、セールスリード、注文または顧客のサービス依頼の電話、新しいアイテム、テキストのメッセージ、ソーシャルメディアの投稿、株式市場のフィード、トラフィックレポート、天気予報、モノのインターネットのデータ、または他の種類のデータを含み得る。

【 0 0 2 7 】

[0028]構築されたデータストリームは、経時的なスナップショットのコレクションとしてデータベースデータから生成されるデータを言う。構築されたデータストリームは、構築されたデータストリームのソースが選択された値および選択された時間の関数としてアクセスされ得るように、時間要素と関連付けられる。構築されたデータストリームは、時間要素を備えるデータストリームとしてアクセスすることができ、また、クエリーに基づき、構築されたデータストリームからデータを取り出すルックアップメカニズムを使用してアクセスすることができる。

10

20

30

40

50

## 【 0 0 2 8 】

[0029]時間要素は、イベントが生じたときの特定の時間を指す。時間要素は、時間要素が保存されないデータベースに入力されたイベントと関連付けてもよい。時間要素は、データストリームイベントと関連付けてもよい。また、時間要素は、イベントが生じてデータベースデータのスナップショットの一部として保存されたとき、それから導き出される構築されたデータストリームイベントと関連付けてもよい。また、構築されたデータストリームは、構築されたデータストリームが、以下でより詳細に述べるように、双方向時間的結合をサポートするように、複数の時間要素と関連付けてもよい。

## 【 0 0 2 9 】

[0030]したがって、本明細書に述べる第1の実施形態では、向上したイベント処理のためのコンピューターによる実施方法が提示される。この方法は、向上したイベント処理のリクエストを受け取るステップを含む。向上したイベント処理のリクエストは、データベースデータのソースからの入力データの指標を含む。方法は、構築されたデータストリームのソースを使用して入力データの指標で示された入力データをアクセスするステップをさらに含み、構築されたデータストリームのソースは、経時的なスナップショットのコレクションへのアクセスを提供する。入力データは、時間要素を含む構築されたデータストリームのソースを使用してアクセスされる。また、方法は、構築されたデータストリームを使用してアクセスされる入力データに基づき、向上したイベント処理のリクエストに対する出力を生成するステップを含む。

## 【 0 0 3 0 】

[0031]本明細書に述べる第2の実施形態では、1つまたは複数のコンピューティングデバイスによって使用されたとき、その1つまたは複数のコンピューティングデバイスに向上したイベント処理プロセスのための方法を実施させるコンピューター使用可能命令を格納する、1つまたは複数のコンピューター記憶媒体が提供される。その方法は、データベースデータのソースを識別するステップを含む。方法は、データベースデータのソースのデータベースデータの少なくとも一部をサンプリングして、データベースデータの1つまたは複数のスナップショットを生成するステップをさらに含み。また、方法は、データベースデータの1つまたは複数のスナップショットに基づき、構築されたデータストリームを生成するステップを含む。構築されたデータストリームは、時間要素と関連付けられる。方法は、向上したイベント処理のリクエストに基づき、構築されたデータストリームへのアクセスを提供するステップをさらに含み、向上したイベント処理のリクエストは、構築されたデータストリームを使用してアクセス可能なデータベースデータのソースのデータベース中の入力データの指標と関連付けられる。

## 【 0 0 3 1 】

[0032]本明細書に述べる第3の実施形態では、向上したイベント処理を実施するためのコンピューターシステムが提供される。このシステムは、データベースデータへのアクセスを提供するように構成されたデータベースデータのソース構成要素を含む。システムは、データベースデータのソースを識別し、データベースデータのソースのデータベースデータの少なくとも一部をサンプリングして、データベースデータの1つまたは複数のスナップショットを生成し、データベースデータの1つまたは複数のスナップショットに基づき、構築されたデータストリームを生成し、構築されたデータストリームは時間要素と関連付けられ、かつ向上したイベント処理のリクエストに基づき、構築されたデータストリームへのアクセスを提供するように構成される構築されたデータストリームのソース構成要素を含む。向上したイベント処理のリクエストは、構築されたデータストリームを使用してアクセス可能なデータベースデータのソースのデータベース中の入力データの少なくとも指標と関連付けられる。また、システムは、データストリームへのアクセスを提供するように構成されたデータストリームのソースを含む。

## 【 0 0 3 2 】

[0033]システムはイベント処理サーバーを含み、このイベント処理サーバーは、向上したイベント処理のリクエストを受け取ることであって、向上したイベント処理のリクエス

トは、データベースデータのソースからの入力データの指標およびデータストリームからの入力データの第2の指標を含む、こと、構築されたデータストリームのソースを使用して入力データの指標で入力データにアクセスすることであって、構築されたデータストリームのソースは、データベースデータのソースのデータベースデータの経時的なスナップショットのコレクションへのアクセスを提供し、入力データは、第1の時間要素を含む構築されたデータストリームを使用してアクセスされる、こと、データストリームのソースを使用して入力データの第2の指標で入力データにアクセスし、構築されたデータストリームの第1の時間要素をデータストリームの第2の時間要素と関連させること、および、構築されたデータストリームの入力データおよびデータストリームの入力データに基づき、向上したイベント処理のリクエストに対する出力を生成することのために構成される。

10

#### 【0033】

[0034]本発明の実施形態の要旨を簡単に述べてきたが、本発明の様々な態様に関する全体的な状況を提示するために、本発明の実施形態が実施されることが可能な例示的な動作環境を以下に述べる。具体的に図1を最初に参照すると、本発明の実施形態を実施するための例示的な動作環境が示されており、全体がコンピューティングデバイス100として示されている。コンピューティングデバイス100は、適切な計算環境の一例に過ぎず、本発明の使用または機能性の範囲について、いかなる限定も示唆する意図はない。コンピューティングデバイス100は、例示された構成要素のいずれか1つ、またはその組み合わせに関して何らかの依存性または必要性も有していると解釈すべきでない。

#### 【0034】

20

[0035]本発明は、プログラムモジュールなどのコンピューター実行可能命令を含み、パーソナルデータアシスタントまたは他の携帯端末などのコンピューターまたは他のマシンによって実行されるコンピューターコードまたはマシン使用可能命令の全体的な文脈で述べる。全体的に、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プログラムモジュールは、特定のタスクを実施する、または特定の抽象データ型を実装するコードを言う。本発明は、携帯端末、家庭用電化製品、汎用コンピューター、より専門的なコンピューティングデバイスなどを含む様々なシステム構成で実施される可能性がある。また、本発明は、タスクが通信ネットワークを通じてリンクされる遠隔処理装置によって実施される分散型の計算環境中で実施される可能性がある。

#### 【0035】

30

[0036]図1を参照すると、コンピューティングデバイス100は、次の装置を直接または間接的に結合するバス110を含み、その装置は、メモリー112、1つまたは複数のプロセッサ114、1つまたは複数のプレゼンテーション構成要素116、入力/出力ポート118、入力/出力構成要素120、および例示の電源122である。バス110は、1つまたは複数のバス（アドレスバス、データバスまたはその組み合わせなど）である可能性のものを表す。図1の様々なブロックは、明確にするために、線を用いて示されているが、実際には、様々な構成要素の境界を示すことはそれほど明瞭ではなく、比喩的に言えば、線は、より正確にはグレーであいまいなものになるはずである。たとえば、表示装置などのプレゼンテーション構成要素は、I/O構成要素であると考えられることがある。また、プロセッサは、メモリーを有する。そのようなことは、この技術の性質であると認識され、図1の略図は、本発明の1つまたは複数の実施形態と関係して使用することができる例示的なコンピューティングデバイスを単に例示すると繰り返される。「ワークステーション」、「サーバー」、「ラップトップ」、「携帯端末」などのようなカテゴリーは区別されない。というのは、すべてが図1の範囲内に在り、「コンピューティングデバイス」を言うと考えられるからである。

40

#### 【0036】

[0037]コンピューティングデバイス100は、通常、様々なコンピューター可読媒体を含む。コンピューター可読媒体は、コンピューティングデバイス100がアクセスすることができる、いずれもの利用できる媒体とすることができ、揮発性および不揮発性の媒体と、取り外し可能なおよび取り外しできない媒体の両方を含む。例としてだけで、限定す

50

ることなく、コンピューター可読媒体は、コンピューター記憶媒体および通信媒体を含んでもよい。

【 0 0 3 7 】

[0038]コンピューター記憶媒体は、コンピューター可読命令、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で実現される揮発性および不揮発性、取り外し可能なおよび取り外しできない媒体を含む。コンピューター記憶媒体は、ただしこれらに限定されないが、RAM、ROM、EEPROM、フラッシュメモリーまたは他のメモリー技術、CD-ROM、デジタル多用途ディスク(DVD)または他の光ディスク記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置または他の磁気記憶装置、または所望の情報を格納するために使用することができる、コンピューティングデバイス100がアクセスすることができる他のいずれかの媒体を含む。コンピューター記憶媒体は、本質的に信号を考慮しない。

10

【 0 0 3 8 】

[0039]通信媒体は、通常、搬送波または他の移送メカニズムなどの変調されたデータ信号の形で、コンピューター可読命令、データ構造、プログラムモジュールまたは他のデータを具現化し、いずれかの情報配信媒体を含む。用語「変調されたデータ信号」は、信号中に情報をコード化するように設定された、または変化させた信号の特性の1つまたは複数を有する信号を言う。例としてだけで、限定せずに、通信媒体は、有線ネットワークまたは直接配線接続などの有線媒体、および音響、RF、赤外線などの無線媒体、および他の無線媒体を含む。また、上記事項のいずれかの組み合わせは、コンピューター可読媒体の範囲内に含めるべきである。

20

【 0 0 3 9 】

[0040]メモリー112は、揮発性および/または不揮発性メモリーの形態でコンピューター記憶媒体を含む。メモリーは、取り外し可能、取り外しできない、またはその組み合わせであってもよい、例示的なハードウェア装置は、半導体メモリー、ハードドライブ、光ディスクドライブなどを含む。コンピューティングデバイス100は、メモリー112またはI/O構成要素120などの様々な実在物からデータを読む、1つまたは複数のプロセッサを含む。プレゼンテーション構成要素(複数可)116は、データ表示をユーザーまたは他の装置に与える。例示的なプレゼンテーション構成要素は、表示装置、スピーカー、印刷構成要素、振動構成要素などを含む。

30

【 0 0 4 0 】

[0041]I/Oポート118によって、コンピューティングデバイス100が、I/O構成要素120を含む他の装置に論理的に結合されることが可能になり、それら装置のいくつかは、組み込んでよい。例示の構成要素は、マイクロフォン、ジョイスティック、ゲームパッド、衛星放送受信アンテナ、スキャナー、プリンター、無線装置などを含む。

【 0 0 4 1 】

[0042]ここで図2を見ると、本明細書に述べる実施形態を実施する上で使用するのに適切な例示的な向上したイベント処理システム200を描くブロック図が示されている。全体的に、システム200は、とりわけ向上したイベント処理を実施するために使用できる可能性がある動作環境を例示する。向上したイベント処理は、少なくともデータベースデータのソースからの入力データを示すイベント処理のリクエストと関連付けてよい。

40

【 0 0 4 2 】

[0043]図示しない構成要素のなかでも、システム200は、一般的に、データベースデータのソース構成要素220、構築されたデータストリームのソース構成要素230、データストリームのソース構成要素240、コンピューティングデバイス250および向上したイベント処理サーバー260を含んでもよく、すべてがネットワーク270を介して互いに通信状態にある。ネットワーク210は、限定せずに、1つまたは複数のローカルエリアネットワーク(LAN)および/またはワイドエリアネットワーク(WAN)を含んでもよい。そのようなネットワーク化された環境は、オフィス、企業規模のコンピューターネットワーク、イントラネットおよびインターネットでよく一般的である。

50

## 【 0 0 4 3 】

[0044]いくつかの実施形態では、例示された構成要素および/またはモジュールの1つまたは複数、独立型のアプリケーションとして実装してもよい。さらなる実施形態では、例示された構成要素および/またはモジュールの1つまたは複数、コンピューティングデバイスを介して、インターネットベースのサービスとして、および/または向上したイベント処理サーバー260内のモジュールとして実装してもよい。当業者は、図2に例示された構成要素およびモジュールが、性質および数で例示的なものであり、限定するものと解釈すべきでないことを理解されるはずである。本明細書に述べる機能性を達成するために、いずれもの数の構成要素および/またはモジュールを用いてもよい。たとえば、任意の数のコンピューティングデバイス、データベースデータのソース構成要素、構築されたデータストリームのソース構成要素、データストリームのソース構成要素、コンピューティングデバイス、向上したイベント処理サーバーおよびネットワークは、この実施形態の範囲内でシステム200中に採用してもよい。それぞれは、単一の装置/インターフェースまたは動作環境中で互いに協力する複数の装置/インターフェースを含んでもよい。たとえば、分散型環境中に配置された複数の装置および/またはモジュールは、本明細書に述べる向上したイベント処理サーバー260の機能性を共同でもたらしめてもよい。語句「アプリケーション」または「サービス」は、本明細書で使用する時、コンピューティングデバイスおよび/またはデータセンター中の複数のコンピューティングデバイスなどの複数のコンピューティングデバイス上でランする、またはその内の記憶場所にアクセスする、いずれかのソフトウェア、またはソフトウェアの部分を広く言うことがある。

10

20

## 【 0 0 4 4 】

[0045]本明細書に述べるこのおよび他の構成は、単に例として説明されていることを理解すべきである。他の構成および要素(たとえば、マシン、インターフェース、機能、順序、および/または機能のグループ分け)は、示されるものに加えて、またはその代わりに使用することができ、いくつかの要素を同時に省いてもよい。さらに、本明細書に述べる要素の多くは、別個の、または分散された構成要素として、または他の構成要素と関係して、およびいずれかの適切な組み合わせおよび記憶場所で実装してもよい機能的な実在物である。1つまたは複数の実在物によって果たされるような本明細書に述べる様々な機能は、ハードウェア、ファームウェアおよび/またはソフトウェアによって実施してもよい。たとえば、システム200を参照して以下で述べる機能を含む様々な機能は、メモリーに格納された命令を実行するプロセッサによって実施してもよい。

30

## 【 0 0 4 5 】

[0046]ここでシステム200中に含まれ得る様々な構成要素を見ると、データベースデータのソース構成要素220は、データベースデータを格納し、データベースデータへのアクセスを提供することに関与する。データベースデータのソース構成要素は、リレーショナルデータベーステーブルを有するデータベース中にデータを格納することができる。データベースデータは、データベース中のデータと関連する時間要素を有さないことがある。時間要素は、本明細書で使用する時、データベーステーブル中への入力に関連するイベントが生じたときの特定の時間を言うことがある。たとえば、顧客について、第1のトランザクションが、顧客の第1の住所をファイルに保存することである場合がある。顧客は、第2の時間に第2のトランザクションを実施することがあり、次いで第1の住所を顧客の新しい第2の住所に更新する場合がある。そのように、データベースは、時間要素に関係なく顧客住所を更新する。データベーステーブルは、前の入力または新しい入力と関連する時間に関係なく記録される、新しい情報に基づき経時的に変化するレコードを含む。この点で、データベースデータ構成要素に対するデータリクエスト(たとえば、クエリー)は、データが経時的に変化するので、繰り返し可能でない。繰り返し可能な計算とは、その後で同じデータのソースを用いてイベント処理分析を再実行し、同じ結果を生じることと言う。

40

## 【 0 0 4 6 】

[0047]さらに、データベースデータ構成要素は、時間要素を含まない操作をサポートす

50

ることができる。顧客住所情報を求めるクエリーは、クエリーが実行されたときの顧客の顧客住所情報をもたらす。特に第1のトランザクション時における、または第2のトランザクションの間の顧客住所情報を見出すための操作は、サポートされない。というのは、データベーステーブルは、トランザクションと関連する最新顧客住所情報を記録するだけであるからである。

#### 【0047】

[0048]例示の実施形態では、データベースデータのソース構成要素220は、管理時間要素をサポートするデータベースへのアクセスを提供することができる。管理時間要素は、データベースデータの操作ベースの機能性（たとえば監査、保守およびプロビジョニング）のために使用される時間要素（たとえば時間）を言う。管理時間要素を有するデータベースでは、管理時間要素は、データベーステーブル中に記録されるように意図されていない。しかし、管理時間要素は、1つまたは複数の外部構成要素にデータベースデータの変更を知らせる通知メカニズムの一部として使用してもよい。通知メカニズムは、計量機構が、データベースデータのサンプリングを実施するように通知を発するまでの変更の所定数を決定する、または所定期間を決定することができる計量機構をさらに実装してもよい。そのように、管理時間要素の意図する性質は、管理時間要素がつかの間のものであって記録されないことである可能性があるが、時間要素は、構築されたデータストリームを生成する上で使用することができる。構築されたデータストリームのソース構成要素は、データベースデータの変更の通知を受け取り、以下でより詳細に議論するように、差分スナップショットを生成するサンプリングを使用して構築されたデータストリームを生成するためのサブスクリプションメカニズムを有してもよいことが企図される。

#### 【0048】

[0049]データベースデータのソース構成要素220は、向上したイベント処理ロジックを識別する前に、イベント処理を単独で使用するか、またはデータストリームのソースと統合して使用して、データベースデータ構成要素中のデータを処理するように開発されている場合があることがさらに企図される。この点で、データベースデータのソース構成要素220は、向上したイベント処理のために選択されたロジックと特に関連するいずれもの特徴も有さない可能性がある。言い換えると、本明細書に述べる実施形態での向上したイベント処理ロジックは、以下でより詳細に議論するように、旧来のシステムとともに機能するように、または向上したイベント処理ロジックと無関係に開発されるデータベースと下位互換性があるように構成される。

#### 【0049】

[0050]構築されたデータストリームのソース構成要素230は、構築されたデータストリームを管理し、それへのアクセスを提供することに関与する。構築されたデータストリームは、データベースデータと関連する値および時間に基づきアクセスすることができるように生成される。この点で、構築されたデータストリームは、値および時間要素の関数である。構築されたデータストリームは、データの値および時間要素に基づき、データベースデータと関連付けられる。キーアトリビュートまたはアトリビュートのセットは、入力が、データストリーム中に構築されるデータベースデータに対して存在することを保証するために使用してもよいことが企図される。

#### 【0050】

[0051]構築されたデータストリームは、再現性を可能にするために、所与の期間、スナップショットを保存する。データストリームをデータベースデータと結合したとき、構築されたデータストリームのスナップショットは、データストリーム（たとえば生のデータストリーム）の時間要素（たとえばアプリケーション時間のタイムスタンプ）によって定義することができる。データベースデータは、リレーショナルテーブルを時間の概念によっては管理しないが、スナップショットを生成するためにサンプリングされるデータベースデータと関連する時間として管理時間要素（たとえば、データベースシステムのコミット時間）を用いて管理する。この点で、データベースデータのソースは、同期時間としてコミット時間を用いてスナップショットティングをサポートし、データベースデータは、イ

ベント処理システム中のストリーム配信データソースとしてアクセスすることができる。

【 0 0 5 1 】

[0052]実施形態では、データベースデータのソースは、スナップショットの変更が存在しない場合でも、向上したイベント処理システム側で自動的に時間を進める時間増加方式を用いて実装してもよい。また、この方式は、さらに、データベースデータのソースからの取り出しベースの更新を推進するために使用することができる。データベースデータのソースは、初期化モード（ランダウンと同様）を可能にし得て、そのとき任意選択のタイムスタンプを設定することができる。タイムスタンプは、このモードではいずれかの特定の順序である必要がないことがある。この操作は、データベースデータがこれまで使用される前に一度だけ実施することができる。

10

【 0 0 5 2 】

[0053]構築されたデータストリームは、基準データまたは `refData` として以下のコードで示され、構築されたデータストリームを使用する操作の実施を可能にするアプリケーションプログラミングインターフェース（API）をサポートしてもよい。APIは、ユーザーが操作を実施することができるCRUD（作成、読み出し、更新および削除（`create`, `read`, `update` and `delete`））インターフェースとしてもよい。以下のコードでの例示的なシナリオが関連する可能性があり、アプリケーションサービスは、準リアルタイム分析を実施するために、様々な装置からログエラー情報を集めるように構成することができる。各装置は、装置識別子（たとえば `MachineID`）によって識別してもよい。装置識別子は、対応する装置名および詳細（たとえば、`MachineName` および `MachineDetails`）にマッピングしてもよいことが企図される。装置識別子から対応する装置名へのマッピングは、複数の装置識別子の装置名へのマッピングのデータベーステーブル中にメタデータとして格納することができる。APIは、具体的には、キー値をテーブルに格納する、時間要素（たとえば、システム時間としてのトランザクション時間）を実装するスキーマをサポートしてもよい。以下の比較として、構築されたデータストリームの初期化には明確な時間を持つAPIが使用されてよい。

20

【 0 0 5 3 】

【表 1】

```
var refData = new ReferenceData<int, MachineDetails>(new Uri("http://..."));

refData.Add(1, new MachineDetails(machineId:1, machineName:"x", domain:"A"));
refData.Add(2, new MachineDetails(2, "y", "A"));
refData.Update(1, new MachineDetails(1, "x2", "A"));
refData.Delete(2);
```

30

```
var refData2 = new ReferenceData<int, MachineDetails, long>(new Uri("http://myRefData"));

refData2.Add(1, new MachineDetails(machineId: 1, machineName: "x", domain: "A"), 1000);
refData2.Add(2, new MachineDetails(2, "y", "A"), 5000);
refData2.Update(1, new MachineDetails(1, "x2", "A"), 3000);
refData2.Delete(2, 5000);
refData2.Add(2, new MachineDetails(2, "yz", "ABC"), 4000);
```

40

```
refData2.Add(2, new MachineDetails(2, "yz", "ABC"), 4000);
```

【 0 0 5 4 】

[0054]時間情報は、どの特定の順序でも読み出されず、データベースデータのソースに対する時間関連情報のコールは、時間を現在のシステム時間に設定することになる通常の

50

A P I コールと混ぜることができる。

【 0 0 5 5 】

[0055]構築されたデータストリームのA P I は、 $t_1$ と $t_2$ の間で変化が存在しないとき、時間 $t_2$ における値が $t_1$ における値になる階段関数を実装してもよい。A P I は、所与の時間の特定のキーの値を読むために使用することができる。

【 0 0 5 6 】

【表 2】

```
refData2.MoveTimeTo(4500);

MachineDetails value = refData2.ReadAt(key: 2, time: 3500);           // (2, "y", "A")
value = refData2.ReadAt(key: 2, time: 4200);                           // null
```

10

【 0 0 5 7 】

[0056]構築されたデータストリーム中の最後のアプリケーション時間の後で構築されたデータストリームの読み出しは禁止できることが企図される。というのは、そうすると再現不可能な結果に繋がるためである。アプリケーション時間の場合、時間の進行を強制して、アプリケーション時間を前進させるために、または初期化フェーズを閉じてシステム時間に切り替えるために、パンクチュエーションメカニズム (p u n c t u a t i o n m e c h a n i s m) を使用してもよい。システム時間の場合、本明細書の実施形態は、以下でより詳細に議論するように、現在時間以後の読み出しを禁止することができる。

20

【 0 0 5 8 】

[0057]また、A P I は、所与の時間におけるすべての値を読むために使用することができる。

【 0 0 5 9 】

【表 3】

```
Dictionary<int, MachineDetails> values = refData2.ReadAt(2500);       // {(1, (1, "x", "A")),
                                                                    // (2, (2, "y", "A"))}
```

【 0 0 6 0 】

[0058]その結果は、ローカルに知られているすべての値である。分割 / パーティション分割 ( s h a r d i n g / p a r t i t i o n i n g ) に依存して、これは、構築されたデータストリーム中で基準データのサブセットとしてもよい。

30

【 0 0 6 1 】

[0059]ストリーミングメカニズムは、構築されたデータストリームをストリーム配信するために実装することができる。A P I の基本読み出し機能 (たとえば、R e a d A T) は、構築されたデータストリームの時間要素を正確にはサポートしない恐れがある。というのは、いつ変化が生じたのかを知る必要があるからである。ストリーミングメカニズムは、対応するエッジおよび期間のイベントを生成するために、構築されたデータストリームを利用する。ストリーミングメカニズムは、変化毎にエッジイベントを生成する能力を含み、各変化は、前の値を有するエッジ終了イベントおよび新しい値を有するエッジ開始イベントをもたらすことになる。また、ストリーミングメカニズムは、構築されたデータストリームが、特定の時間から、すなわち時間の始めからでなく、ストリーム配信される必要があるときの初期状態を知る能力を含む。

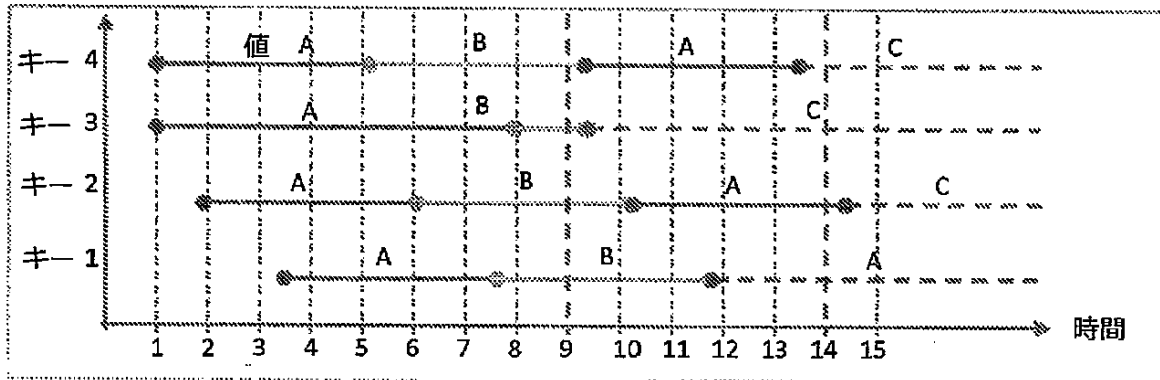
40

例として、

【 0 0 6 2 】



【表 4】



10

テーブル構築されたストリームデータ

## 【0063】

[0060] 9 から 14 までのこの構築済のストリームデータに対応する変化のストリーム配信は、

## 1) 時間 9 における初期「状態」

- ・タイムスタンプ：5 . 2 , キー：4 , 値：B
- ・タイムスタンプ：6 . 1 , キー：2 , 値：B
- ・タイムスタンプ：7 . 5 , キー：1 , 値：B
- ・タイムスタンプ：7 . 9 , キー：3 , 値：B

20

## 2) 9 と 14 の間の変化

- ・タイムスタンプ：9 . 2 , キー：4 , 値：A
- ・タイムスタンプ：9 . 3 , キー：3 , 値：C
- ・タイムスタンプ：10 . 1 , キー：2 , 値：A
- ・タイムスタンプ：11 . 8 , キー：1 , 値：A
- ・タイムスタンプ：13 . 5 , キー：4 , 値：C

をもたらすことになるはずである。

## 【0064】

30

[0061] しかし、この変化のストリームからエッジ / 期間イベントのストリームの生成は、前の値に対応するエッジ終了イベントを生成するために、各キー（最後のタイムスタンプおよび最後の値）の追跡を継続することが必要になるはずである。値のペア（旧および新）として変化の格納およびストリーム配信は、それをより容易にすることになるはずである。

## 【0065】

## 1) 時間 9 における初期「状態」

（変化イベント当たり 1 つのエッジ開始イベントを生成するために使用することができる）

- ・タイムスタンプ：5 . 2 , キー：4 , 値：B , 旧：ゼロ
- ・タイムスタンプ：6 . 1 , キー：2 , 値：B , 旧：ゼロ
- ・タイムスタンプ：7 . 5 , キー：1 , 値：B , 旧：ゼロ
- ・タイムスタンプ：7 . 9 , キー：3 , 値：B , 旧：ゼロ

40

## 2) 9 と 14 の間の変化

（変化イベント当たり 1 つのエッジ終了イベントおよび 1 つのエッジ開始イベントを生成するために使用することができる）

- ・タイムスタンプ：9 . 2 , キー：4 , 値：A , 旧：[ 値：B , から：5 . 2 ]
- ・タイムスタンプ：9 . 3 , キー：3 , 値：C , 旧：[ 値：B , since：7 .

9 ]

- ・タイムスタンプ：10 . 1 , キー：2 , 値：A , 旧：[ 値：B , since：6

50

. 1 ]

・タイムスタンプ：11.8，キー：1，値：A，旧：[ 値：B，since：7

. 5 ]

・タイムスタンプ：13.5，キー：4，値：C，旧：[ 値：A，since：9

. 2 ]

[0062]このストリームは、構築されたデータストリームに対応するエッジイベントを生成するのに十分である。これは、向上したイベント処理システムが使用することができるソースとして構築されたデータストリームを顕在化させるために、内部的に使用されることになる。

【0066】

10

【表5】

```
IObservable<Change<int, MachineDetails, long>> observable = refData.StreamChanges(fromTime:
0, toTime: 5000);

observable.ForEachAsync(change => Console.WriteLine(string.Format(
    "time: {0}, key: {1}, oldValue: {2}, newValue: {3}",
    change.Time, change.Key,
    change.OldValue!=null ? change.OldValue.Domain + "\\" + change.OldValue.MachineName :
    "null",
    change.NewValue!=null ? change.NewValue.Domain + "\\" + change.NewValue.MachineName :
    "null")
));
```

20

【0067】

[0063] t1 から t2、次いで t2 から t3 までの基準データおよびをストリーム配信するとき、t2 における初期状態は、無視することができることが企図される。それを要求しない方法があるか、またはその結果生じるエッジ開始イベントが重複しないようにする必要がある。

【0068】

[0064]構築されたストリームデータは、さらに、オペレーターが構築されたデータストリームをポイントイベントのストリーム（たとえばデータストリーム）と結合するのをサポートする可能性がある。例としてだけで、以下にコードを示す。

30

【0069】

【表6】

```
var queryDef = ReferenceData.From(refData)
    .TemporalJoin(EventStream.From(stream),
        (ref, str) => ref.MachineId == ref.Id,
        (ref, str) => new { Id = str.Id, State = str.State, Name = ref.MachineName } )
    .To(sink);

var query = ctx.StartQuery("q1", queryDef);
```

40

【0070】

[0065]結合のパラメーターは、1)ストリーム、2)結合述部、3)結果セクター（すなわちプロジェクション）である。結合述部は、ストリームおよび基準データが、machine\_\_idによって再編成済のとき、無視する、または“(ref, str) true”に置き換えてもよい。

【0071】

[0066]双方向時間的結合は、構築されたデータストリームを使用して実施することができることが企図される。たとえば、データベース中の基準データが、ストリーミングメカ

50

ニズムを使用して構築されたデータストリームとして使い果たされたとき、イベントの時間要素（たとえばタイムスタンプ）は、時間的結合に使用することができ、データベース中の基準データが、ルックアップメカニズムを使用して構築されたデータストリームとして使い果たされたとき、異なるタイムライン（たとえば、イベントのペイロード中の別の時間）が、結合条件として使用することができるはずである。

【 0 0 7 2 】

[0067] 構築されたデータストリーム構成要素 2 3 0 が、向上したイベント処理において利用されるデータベースデータを有する識別されたデータベースデータのソースを使用して、構築されたデータストリームを生成するように構成される。待たされていた向上したイベント処理のリクエストは、何のデータベースデータが構築されたデータストリームを生成する上で使用されるのかを事前に決定し、示すことができる可能性があることがさらに企図される。構築されたデータストリームを構築する際に使用されるデータベースデータは、データベースデータに対して出される、待たされていた向上したイベント処理のリクエストの要因であり得る。

10

【 0 0 7 3 】

[0068] 向上したイベント処理のリクエストは、構築されたデータストリームのソース構成要素 2 3 0 の構築されたデータストリームを使用して実際にアクセスされる、データベースデータ構成要素 2 2 0 中のデータを求めるリクエストを含む。構築されたデータストリームは、データストリーム特徴および機能性をシミュレーションしながら、データベースデータ構成要素に対するプロキシとして働くことができる。構築されたデータストリームは、向上したイベント処理システム中でデータ型として定義してもよいことが企図される。データ型は、構築されたデータストリームがどのように系統化されているのかを決定するスキーマとして定義してもよい。

20

【 0 0 7 4 】

[0069] 操作では、構築されたデータストリームは、データベースデータの個別の実例をサンプリングして、データベースデータのスナップショットを生成することをサポートするサンプリングメカニズムに基づき、生成してもよい。サンプリングは、全データベースについて、またはデータベースデータの識別される部分についてだけ実施してもよい。また、サンプリングは、向上したイベント処理のリクエストと関連する値がデータベースのスナップショット中に保存されるように、期待される向上したイベント処理のリクエストに特に基づき、実施してもよい。

30

【 0 0 7 5 】

[0070] 一実施形態では、構築されたデータストリームのソース構成要素 2 3 0 は、定期的なサンプリングメカニズムを実装してもよい。定期的なサンプリングメカニズムは、所定の、または指定された時間間隔でサンプリングの実施をサポートする。例として、データベースデータは、5 分ごとにサンプリングしてもよい。別の例示の実施形態では、構築されたデータストリームは、データベースデータの差分スナップショットを生成するためにサンプリングをサポートする差分サンプリングメカニズムに基づき、生成してもよい。差分スナップショットは、データベースデータの変化を示し、データベースデータの全スナップショットではない。異なるスナップショットを生成するための差分サンプリングは、データベースデータの全スナップショットを格納するコストを減少させるために実装できる可能性があることが企図される。例として、ゆっくりと変化するデータのセットは、差分スナップショットを生成するための差分サンプリングには理想的である可能性がある。というのは、変化はめったに起こらないからであり、そのように、差分スナップショットは、その後に生じることがあるいずれもの変化をファイルに保存する。構築されたデータストリームのスナップショットは、スナップショットを有する構築されたデータストリームに対するクエリーが同じ答えをもたらすという点で再現性をサポートするが、一方データベースデータに対するクエリーは、データベース中のデータが経時的に変化したとき、異なる答えをもたらす可能性がある。

40

【 0 0 7 6 】

50

[0071] 構築されたデータストリームのソース構成要素 230 は、データベースデータの変化の通知を受け取るサブスクリプションメカニズムを実装してもよい。変化の通知は、データベースデータの変化を変化の時間と関連付けるために管理時間要素を使用する、データベースデータのソース構成要素 220 に実装された通知メカニズムに基づかせてもよい。通知が受け取られたとき、構築されたデータストリームデータ構成要素は、データベースデータをサンプリングし、サンプリングされたデータベースデータを管理時間要素中の時間と関連付けることができる。サンプリングは、上記で議論したように、全サンプリングまたは差分サンプリングとしてもよい。この点で、サンプリングは、定期的なメカニズムに基づかないが、しかしむしろデータベースデータの変化の通知に基づく。さらに、向上したイベント処理のリクエストからの出力の正確さは、向上する、つまり構築されたデータストリームデータは、定期的なサンプリングメカニズムとは異なり、データベースデータの変化を正確に反映するからである。

10

#### 【0077】

[0072] 構築されたデータストリームは、それが、構築されたデータストリームと関連する、異なるタイプの向上したイベント処理ロジックをサポートすることができ、かつ下位互換でサポートすることができる点で有利に柔軟である。構築されたデータストリームは、データベースデータのスナップショットを生成するために、サンプリングを利用してもよい。サンプリングは、定期的なサンプリングメカニズム、差分サンプリングメカニズムに基づいてもよい。構築されたデータストリームは、データベースデータまたは構築されたデータストリームの履歴情報または構造を変える必要がなく、向上したイベント処理に再現性を与える。向上したイベント処理のリクエストに対する出力を生成するためのロジックは、本明細書に述べるように、異なるサンプリング方法を用いて、サンプリング方法を変化させてさえ、維持することができる。構築されたデータストリームに対するリクエストは、どちらにしてもデータベースデータのスナップショットの構築データストリームを用いて有利に機能する。

20

#### 【0078】

[0073] 引き続き図 2 を参照すると、データストリームのソース 240 は、データストリームを管理し、それへのアクセスを提供することに関与する。データストリームまたは情報のストリームは、具体的にイベントについてのものとしてもよい。データストリームデータは、データと関連するイベントが生じたときを示す時間要素と関連付けられる。データストリームは、リスナーアプリケーションを介して積極的にリッスンしてもよい。また、データストリームは、短期間（たとえば 5 分間）一時的に、または長期間（たとえば一年から数年）にわたって格納し、データが活発にストリーム配信されているかのように再生してもよい。例として、イベント処理のリクエストは、新しいストリーム配信データを 1 年前のストリーム配信データと比較し、その比較から結論を導き出すために、使用してもよい。この点で、データストリームは、追跡し分析して、出力を導き出すために、互いに組み合わせてもよい。さらに、データストリームは、データベースデータを用いる向上したイベント処理のリクエスト中に含めてもよいので、データストリームは、向上したイベント処理のリクエストに対する出力を生成するために、構築されたデータストリームと結合される。

30

40

#### 【0079】

[0074] コンピューティングデバイス 250 は、たとえば、図 1 を参照して上記に述べた、いずれかのタイプのコンピューティングデバイス 100 を含んでもよい。コンピューティングデバイス 250 は、本明細書に述べる種々の構成要素へのアクセスを提供することができる。具体的に、コンピューティングデバイス 250 は、さらに本明細書で述べるように、向上したイベント処理サーバーによって促進される 1 つまたは複数の操作を実施するために、向上したイベント処理サーバー 260 にアクセスしてもよい。

#### 【0080】

[0075] 向上したイベント処理サーバー 260 は、本明細書に述べるように、向上したイベント処理の管理に関与する。向上したイベント処理サーバー 260 は、選択された値お

50

よび時間の関数として、スナップショットのコレクションとしての構築されたデータストリームにアクセスする。向上したイベント処理サーバー 260 は、向上したイベント処理のリクエストを受け取り得る。向上したイベント処理のリクエストは、コンピューティングデバイス 250 を介して、または体系的なイベント処理を求めるリクエストを生成するようにプログラムされたリクエストサービスを使用して受け取ることができることが企図される。向上したイベント処理のリクエストは、向上したイベント処理のリクエストが 1 つまたは複数のデータベースデータのソースと関連付けられている、またはデータベースデータのソースおよびデータストリームのソースと関連付けられているのかどうかに基づき、処理することができる。この点で、向上したイベント処理サーバー 260 は、特定のロジックを使用してデータストリームとデータベースデータを関連させることができ、それによってデータに対して出力が導き出される。向上したイベント処理のリクエストは、データを分析して推論の結果を生成するために、追跡したデータの入力に基づき、出力をリクエストしてもよい。出力は、データストリームおよび構築されたストリームデータ中の両方のデータから導き出された結論である可能性がある。

10

【0081】

[0076]引き続き図 2 を参照すると、向上したイベント処理サーバー 260 は、タイミングモジュール 262、操作モジュール 264、ロジックモジュール 266 および妥当性時間モジュール 268 を含む数個のモジュールを含んでもよい。タイミングモジュールは、イベントを処理するためのタイミングメカニズムを実装することができる。具体的に、タイミングメカニズムは、イベントと関連する時間要素および終結要素を特に識別し、時間要素は、イベントが生じた時間を示し、終結要素は、イベントが利用できなくなったときを示す。この点で、タイミングメカニズムを使用するタイミングモジュール 262 は、受け取られた入力と関連するタイミングに基づき迅速に出力を生成するために、どれぐらい早くイベントが到着したのかどうかに基づきイベント処理をサポートしてもよい。

20

【0082】

[0077]操作モジュール 264 は、データストリーム、データベースデータ、構築されたデータストリームおよびそのいずれかの組み合わせに対して実行される可能性がある操作の管理に関与する。具体的に、イベント処理のリクエストは、データベースデータからの入力データの指標を含んでもよく、またはデータベースデータおよびデータストリームに対する入力データの指標を含む組み合わせリクエストを含んでもよい。構築されたデータストリームは、データベースデータのプロキシとして動作できる可能性があり、それで、異なるタイプの操作が、構築されたデータストリームおよびデータベースデータと関連するイベントを使用して実施される可能性がある。

30

【0083】

[0078]向上したイベント処理のリクエストは、データベースデータへの直接アクセスを介しては可能にならない時間要素を含むことができる。この点で、操作モジュール 264 は、向上したイベントデータ処理リクエストを発行するように宣言するクエリーメカニズムをサポートすることができる。宣言型クエリーメカニズムは、操作が機能すべきやり方を最小限にするかまたはなくし、操作が達成すべきものよりもっと多くをサポートする。たとえば、ユーザーは、構築されたデータストリームを使用してデータストリームをデータベーステーブルからの基準データと結合するために、2 つの従来のデータベーステーブルを結合するのと同じように、簡単な結合操作を入力してもよい。

40

【0084】

[0079]例として、異なる器具によって実施される化学分析に対応するイベントのストリームを仮定すると、

【0085】

【表 7】

## UseCase1\_stream

Assay_time	Chem_id	Instrument_id
8:10 am	1	1
8:20 am	1	2
8:30 am	2	1
8:40 am	3	2

10

20

【 0 0 8 6 】

【0080】所与のタイプのテスト（たとえばH1AC）がいくつ実施されたのかを知るためには、イベントに名前（mapping\_id - nameが基準データ中にある）を注釈することが必要である。したがって、次のスナップショットが、8:00amから9amまで有効であると仮定する。

【 0 0 8 7 】

【表 8】

Chem_Id	名前	説明
1	H1AC	糖尿病のためのヘモグロビンA1cテスト
2	Xyz	x y zテスト
3	Abc	a b cテスト

30

40

【 0 0 8 8 】

【0081】chem\_idを名前と置き換えるために、クエリーは、  
 1 SELECT System.Timestamp AS Assay\_time.stream, Chem\_id.map, Name AS Chem\_name,  
 2 stream.Instrument\_id  
 3 FROM Timestamp(VseCase1\_stream. Assay\_time) stream  
 4 JOIN UseCase1\_mapping map  
 5 ON stream.Chem\_id=map.Chem\_id  
 とすることができるはずである。

【 0 0 8 9 】

50

[0082] その結果得られた名前を有するストリームは、  
 【 0 0 9 0 】  
 【 表 9 】

Assay_time	Chem_id	Chem_name	Instrument_id
8:10 am	1	H1AC	1
8:20 am	1	H1AC	2
8:30 am	2	Xyz	1
8:40 am	3	Abc	2

【 0 0 9 1 】

になるはずである。

[0083] 有利にも、向上したイベント処理のリクエストの生成は、既存の宣言型クエリメカニズムと一致し、識別された入力から出力を生成するために、直観的方法が維持される。そのように、向上したイベント処理の特徴は、操作構成要素にクエリを提出する、または操作するユーザーに対して透過である。また、操作モジュール 2 6 4 は、以下でより詳細に議論するような妥当性時間メカニズムに少なくとも部分的に基づき、向上したイベント処理のリクエストを実行し得る。

【 0 0 9 2 】

[0084] ロジックモジュール 2 6 6 は、構築されたデータストリームの構造に基づき、向上したイベント処理の特定の方法を管理するのに関与する。上記で議論したように、構築されたデータストリームは、定期的なサンプリングまたは差分サンプリングのどちらかを使用してデータベースデータのスナップショットを生成するデータのサンプリングに基づいてもよい。サンプリング方法は、入力に基づき出力を生成するために、ロジックにロジックモジュール 2 6 6 を知らせる。また、特定のタイプのデータは、出力を生成する上で使用される可能性があるロジックに影響を与える。

【 0 0 9 3 】

[0085] 一実施形態では、ロジックモジュール 2 6 6 は、選択されたロジックを実装してもよく、それによって操作モジュール 2 6 4 を介して受け取られる操作が、選択されたロジックを使用して実行される。最新のスナップショットロジックは、向上したイベント処理のリクエストと関連する操作を実施するために、最新スナップショットを含むスナップショットのコレクションを利用する。この点で、データベースデータは、構築されたデータストリーム中の所与のスナップショットにまで使用できる状態であり、操作は、最新スナップショットまでのスナップショットのコレクションを使用して実施される。

【 0 0 9 4 】

[0086] 別の例示の実施形態では、ロジックモジュール 2 6 6 は、構築されたデータストリームの 2 つのスナップショットからの構築されたデータストリームの所与の値の値近似をサポートする補間ロジックを実装してもよい。補間ロジックは、特定のタイプの変数データ型のために中間値を生成することができるが、他のタイプにはできない（たとえば温度。しかし顧客住所にはできない）。操作では、サンプリングメカニズム（たとえば、定期的な、または差分）は、第 1 の時間（たとえば 5 : 0 0 p m）における第 1 の値（たとえば 7 0 ° F）および第 2 の時間（たとえば 5 : 0 5 p m）における第 2 の値（たとえば 7 3 ° F）を含むことがある。補間ロジックは、向上したイベント処理のリクエストが中間の時間（たとえば 5 : 0 3 p m）における温度を求めてクエリした場合、起こり得る

実際の値として中間値（たとえば70°Fと73°F）を近似するように構成される。補間メカニズムは、その結果得られる出力が補間に基づいたものであり、構築されたデータストリームからの実際の入力データでないことを明確に示し得ることが企図される。

【0095】

[0087]有利にも、ロジックモジュール266は、構築されたデータストリームが、ロジックのいずれかの変更をサポートするように下位互換である状態で、ロジック選択を変更するステップ（たとえば、最新スナップショットから補間ロジックに、またはその逆も同様）を実装してもよい。言い換えると、構築されたデータストリームの性質は、構築されたデータストリーム自体を再構築せずに、構築されたデータストリームから出力データを生成するロジックの変更をサポートする。例として、データストリームとして、またはルックアップメカニズムを使用してどちらかでアクセスされる基準データを有するデータベースデータは、明確に定義された時点における異なるロジックに基づきアクセスし得る。具体的には、基準データは、毎分取られる温度の当該の分における読みとすることができる。構築されたストリームデータは、取られた最後の温度の読み値の関数である（たとえば最新スナップショットロジック）。それに応じて、階段関数が生成される。異なる時点において（たとえば指定されたロジック変更時間）、ロジックが線形補間ロジックに変更されるイベントでは、構築されたデータストリームのクエリーは、指定されたロジック変更時間までの最新データスナップショットロジックと関連付けることができ、次いで指定されたロジック変更時間の後の値を求めるために、線形補間ロジックを使用することができる。そのようなステートメントは、これまで蓄積された基準データに影響しない、それゆえいずれもの再計算も必要でなく、それに応じて構築されたデータストリームは、下位互換である。

【0096】

[0088]引き続き図2を参照すると、妥当性時間モジュール268は、向上したイベント処理のリクエストに対する矛盾しない出力を示すことに関与する。具体的に、妥当性時間モジュール268は、構築されたデータストリームと関連する出力のための妥当性時間要素を示す妥当性時間メカニズムを実装する。妥当性時間メカニズムは、向上したイベント処理のリクエストに対する矛盾しない出力を示すためのルーチンおよびプロシージャータをもたらす。具体的に、妥当性時間要素は、1つまたは複数のデータソースに基づく出力が、実行された時間におけるデータソース中のデータを正確に反映している時間を示す。構築されたデータストリームの性質は、本明細書に議論する1つまたは複数のメカニズムが、構築されたデータストリームを更新するために使用される前のある一定時間にまでだけ、データベースデータを正確に反映している可能性があるというようなことである。データベース中の最新データを反映するという構築されたデータストリームの能力は、サンプリング、特に利用されるサンプリングメカニズムの機能である。そのように、リクエストは、構築されたデータストリームデータの最後のスナップショットの時間（たとえば有効時間要素）にまでだけ有効である。他の制約（たとえば待ち時間）によって、データストリームデータについてさえ、遅延が最新情報にまで及ぶ恐れがある。

【0097】

[0089]妥当性時間モジュール268は、操作モジュールが、リクエストの妥当性を認可する前に、すべてのイベントデータ（実際のデータストリームまたは構築されたデータストリーム）が到達するのを待つことができるように、操作のための妥当性時間を操作モジュールに伝達し得ると考えられる。妥当性モジュール268を使用して所与の妥当性時間が示されると、操作モジュール264は、その出力が妥当性時間要素と関連付けられないことになるはずである時間をリクエスト中に有するリクエストを、実行しないように防止し得ることがさらに企図される。実施形態では、操作モジュール264は、リクエストを実行し、出力をもたらすことができるが、リクエストが妥当性時間要素の時間にまでだけ有効であることが知られているので、出力をもっとも新しい妥当性時間要素と関連付けることができる。

【0098】

10

20

30

40

50



[0090]実施形態では、妥当性時間は、妥当性時間モジュール268を使用して加速し得る。イベント処理システムを取り巻く状況（たとえばシステム故障または待ち時間）が、構築されたデータストリームを更新するためのデータベースデータへのアクセスを遅延させる恐れがある。イベント処理を継続するために、妥当性時間モジュールは、データベースデータのスナップショットを生成するためのサンプリングが、妥当性時間要素をサポートするために実施されていないときでさえ、構築されたデータストリームイベントと関連する新しい妥当性時間要素を示し得る。そのように、妥当性時間モジュール268は、向上したイベント処理のリクエストを前に進めるために、構築されたデータストリームの正確さを犠牲にすることがある。

【0099】

10

[0091]ここで図3を参照すると、向上したイベント処理を実施するための例示的な方法300を例示するフロー図が示されている。用語「ステップ」および/または「ブロック」が、用いられる方法の異なる要素を含意するために、本明細書で使用されることがあるが、用語は、個別のステップの順序が明確に述べられていない限り、およびそのように述べられているときを除き、本明細書に開示する様々なステップの間のいずれもの特定の順序を暗示するものとして解釈すべきでない。

【0100】

[0092]ブロック310で、向上したイベント処理のリクエストが受け取られる。向上したイベント処理のリクエストは、データベースデータのソースからの入力データの指標を含む。データベースデータのソースからの入力データは、入力データに関する時間要素と関連付けられていない。ブロック320で、入力データが、構築されたデータストリームのソースを使用して入力データの指標でアクセスされる。構築されたデータストリームのソースは、経時的なスナップショットのコレクションへのアクセスを提供する。スナップショットは、データベースデータのソースのデータベースデータについてのものである。入力データは、時間要素を有する構築されたデータストリームを使用してアクセスされる。時間要素は、入力データと関連するイベントが生じたときの時間を示す。構築されたデータストリームの時間要素は、構築されたデータストリームのイベントが生じ、データベースデータ中にイベントのスナップショットの一部として格納された時間から決定される。ブロック330で、向上したイベント処理のリクエストに対する出力が、構築されたデータストリームを使用してアクセスされる入力データに基づき生成される。出力は、構築されたデータストリームに基づく出力が、データベースデータ中のデータを正確に反映している時間を示す妥当性時間要素と関連付けられる。

20

30

【0101】

[0093]ここで図4を見ると、向上したイベント処理を実施するための例示的な方法400を例示するフロー図が示されている。ブロック410で、向上したイベント処理のリクエストが受け取られる。向上したイベント処理のリクエストは、データベースデータのソースからの入力データの指標およびデータストリームのソースからの入力データの第2の指標を含む。ブロック420で、入力データが、構築されたデータストリームのソースを使用して入力データの指標でアクセスされる。構築されたデータストリームのソースは、データベースデータのソースのデータベースデータの経時的なスナップショットのコレクションへのアクセスを提供する。入力データは、第1の時間要素を含む構築されたデータストリームを使用してアクセスされる。

40

【0102】

[0094]ブロック430で、入力データは、データストリームのソースを使用して入力データの第2の指標でアクセスされる。ブロック440で、構築されたデータストリームの第1の時間要素が、データストリームの第2の時間要素と相関される。ブロック450で、構築されたデータストリームの入力データおよびデータストリームの入力データに基づく向上したイベント処理のリクエストに対する出力が生成される。

【0103】

[0095]ここで図5を見ると、向上したイベント処理を実施するための例示的な方法50

50

0を例示するフロー図が示されている。ブロック510で、データベースデータのソースが識別される。データベースデータのソースは、データベースデータと関連付けられる。データベースデータは、基準データを含む。ブロック520で、データベースデータのソースのデータベースデータの少なくとも一部が、データベースデータの1つまたは複数のスナップショットを生成するためにサンプリングされる。データベースデータのサンプリングは、所定の時間間隔に基づき1つまたは複数のスナップショットを生成する定期的なサンプリングメカニズムに、またはデータベースデータへの変更だけを含む1つまたは複数のスナップショットを生成する差分サンプリングメカニズムに基づく。

【0104】

[0096]サンプリングは、さらに、スナップショットがデータベースデータへの変更の指標に基づき生成されるように、データベースデータへの変更が生じたときを示す通知メカニズムに基づいてもよい。通知メカニズムは、計量機構を含むことができ、計量機構は、サンプリングを実施するように通知を発するまでの変更の所定数を決定する、または所定の期間を決定する。通知システムは、データベースデータの操作ベースの機能性と関連付けられる管理時間要素に基づかせることができ、管理時間要素は、データベース中に記録されない。管理時間要素は、時間要素を備える構築されたデータストリームを生成するために使用される、スナップショットと関連付けられる時間を示す。

【0105】

[0097]ブロック530で、構築されたデータストリームは、データベースデータの1つまたは複数のスナップショットに基づき、構築されたデータストリームは、時間要素と関連付けられる。ブロック540で、構築されたデータストリームへのアクセスが、向上したイベント処理のリクエストに基づき提供され、向上したイベント処理リクエストは、構築されたデータストリームを使用してアクセス可能なデータベースデータのソースのデータベース中の入力データの指標と関連付けられる。

【0106】

[0098]本明細書に提示された実施形態を特定の実施形態に関して述べてきたが、それら実施形態は、あらゆる点で、制限するよりむしろ例示するものと意図される。代替実施形態が、本発明が関連する当業者に、本発明の範囲から逸脱せずに明らかになる。

【0107】

[0099]前述から、すべての目標および目的を達成するようになされた1つの源泉中の本発明が、他の利点とともに上記に述べられており、それら利点は、明らかであって、その構造に固有であることが分かるはずである。

【0108】

[00100]いくつかの特徴および下位の組み合わせが役に立ち、それらは、他の特徴または下位の組み合わせに関係なく採用できる可能性があることを理解されたい。これは、特許請求の範囲によって予期され、その範囲内にある。

10

20

30

【図 1】

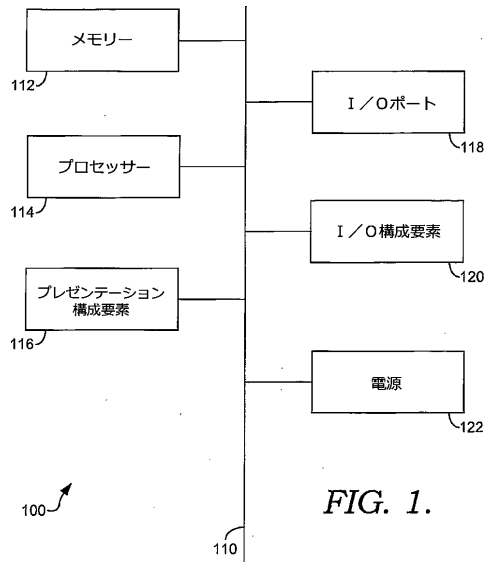


FIG. 1.

【図 2】

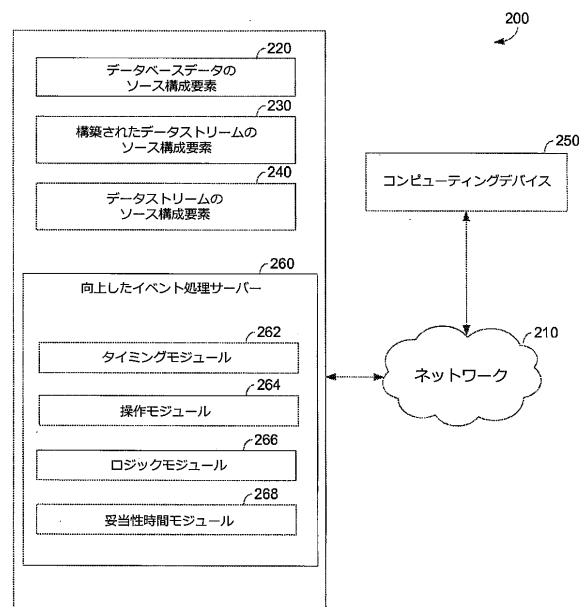


FIG. 2.

【図 3】

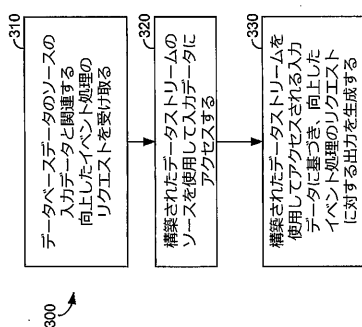


FIG. 3.

【図 4】

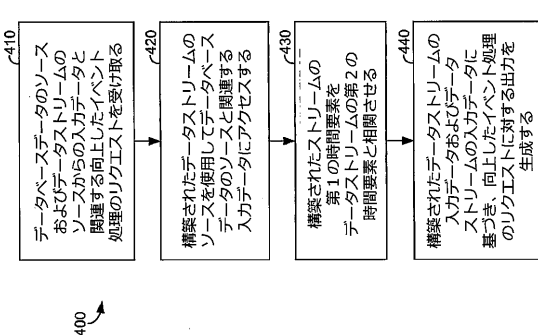


FIG. 4.

【図 5】

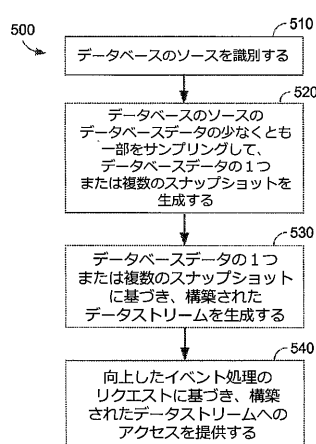


FIG. 5.

## フロントページの続き

(74)代理人 100153028

弁理士 上田 忠

(72)発明者 ノヴィク, レヴ

アメリカ合衆国 ワシントン州 98052-6399 レッドモンド ワン マイクロソフト  
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー

(72)発明者 ナノ, オリヴィエ

アメリカ合衆国 ワシントン州 98052-6399 レッドモンド ワン マイクロソフト  
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー

(72)発明者 ドス・サントス, イヴォ・ホセ・ガルシア

アメリカ合衆国 ワシントン州 98052-6399 レッドモンド ワン マイクロソフト  
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー

(72)発明者 ブッサード, ローレント

アメリカ合衆国 ワシントン州 98052-6399 レッドモンド ワン マイクロソフト  
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー

(72)発明者 ライズマン, アレックス

アメリカ合衆国 ワシントン州 98052-6399 レッドモンド ワン マイクロソフト  
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー

(72)発明者 バニク, ディバンジャン

アメリカ合衆国 ワシントン州 98052-6399 レッドモンド ワン マイクロソフト  
ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー

審査官 鹿野 博嗣

(56)参考文献 特開2011-221799(JP, A)

特開2001-337965(JP, A)

国際公開第2013/069073(WO, A1)

米国特許出願公開第2015/0363464(US, A1)

(58)調査した分野(Int.Cl., DB名)

G06F 16/21

G06F 16/24

G06F 16/903