

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6832347号  
(P6832347)

(45) 発行日 令和3年2月24日 (2021.2.24)

(24) 登録日 令和3年2月3日 (2021.2.3)

(51) Int. Cl.

F I

G 0 6 F 16/182 (2019.01)

G 0 6 F 16/182 1 0 0

請求項の数 11 (全 33 頁)

(21) 出願番号	特願2018-513855 (P2018-513855)	(73) 特許権者	502303739
(86) (22) 出願日	平成28年10月20日 (2016.10.20)		オラクル・インターナショナル・コーポレ イション
(65) 公表番号	特表2018-533130 (P2018-533130A)		アメリカ合衆国カリフォルニア州9406 5レッドウッド・シティー、オラクル・パ ークウェイ500
(43) 公表日	平成30年11月8日 (2018.11.8)		
(86) 国際出願番号	PCT/US2016/057924	(74) 代理人	110001195
(87) 国際公開番号	W02017/070354		特許業務法人深見特許事務所
(87) 国際公開日	平成29年4月27日 (2017.4.27)	(72) 発明者	ツッカラム、プラブ
審査請求日	令和1年6月27日 (2019.6.27)		アメリカ合衆国、94582 カリフォル ニア州、サン・ラモン、ハーロウ・レーン 、2525
(31) 優先権主張番号	62/244,452		
(32) 優先日	平成27年10月21日 (2015.10.21)		
(33) 優先権主張国・地域又は機関	米国 (US)		
(31) 優先権主張番号	15/095,766		
(32) 優先日	平成28年4月11日 (2016.4.11)		
(33) 優先権主張国・地域又は機関	米国 (US)		

最終頁に続く

(54) 【発明の名称】 分散型システムにおける多段階処理のためのイベント順序の保証

(57) 【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、  
イベントストリームの一部として受信される一連のイベントのうちの第1のイベントを  
判断するステップと、

イベントカウンタの値を前記第1のイベントのタイムスタンプで初期化するステップと

、  
前記イベントストリームのさらに他のイベントを処理するステップと、  
前記イベントストリームのフィルタリングされたイベントを識別するステップと、  
前記フィルタリングされたイベントのためのスキップビットを生成するステップとを備  
え、前記スキップビットは前記フィルタリングされたイベントのタイムスタンプを含み、

前記スキップビットを前記イベントストリームに挿入するステップと、  
前記イベントストリームの後続のイベントを受信するステップと、  
前記イベントストリームの順序が狂ったイベントを識別するステップと、  
前記イベントストリームにおける次のイベントが実際のイベントであるかスキップビ  
ットであるかに関係なく、前記後続のイベントの各々に関連付けられた前記タイムスタンプ  
の順序で後続のイベントを処理するステップとを備える、方法。

【請求項 2】

前記第1のイベントは、  
タイマを開始すること、

10

20

前記タイマが時間切れになるまで前記一連のイベントのうちの一組のイベントを受信すること、

前記一組のイベントを時系列順に再配列すること、および

前記第 1 のイベントを、最新のタイムスタンプを有する前記再配列された組のイベントとして識別すること、によって判断される、請求項 1 に記載の方法。

【請求項 3】

前記さらに他のイベントは、前記タイマが時間切れになる前にバッチ処理される、請求項 2 に記載の方法。

【請求項 4】

前記フィルタリングされたイベントは、上流の段階によってフィルタリングされて除去される、請求項 1 ~ 3 のいずれか 1 項に記載の方法。

10

【請求項 5】

前記実際のイベントは、前記イベントストリームに対応するイベントデータを備える、請求項 1 ~ 4 のいずれか 1 項に記載の方法。

【請求項 6】

プログラムコードを格納するコンピュータ読取可能プログラムであって、前記プログラムコードは、コンピューティングシステムのプロセッサによって実行されたときに、前記プロセッサに動作を実行させ、前記動作は、

一連のイベントのうち第 1 のイベントを判断することと、

イベントカウンタの値を前記第 1 のイベントのタイムスタンプで初期化することと、

20

さらに他のイベントを処理することと、

フィルタリングされたイベントを識別することと、

前記フィルタリングされたイベントのためのスキップビートを生成することとを備え、前記スキップビートは前記フィルタリングされたイベントのタイムスタンプを含み、

後続のイベントを受信することと、

順序が狂ったイベントを識別することと、

イベントストリームにおける次のイベントが実際のイベントであるかスキップビートであるかに関係なく、前記後続のイベントの各々に関連付けられた前記タイムスタンプの順序で後続のイベントを処理することとを備える、コンピュータ読取可能プログラム。

【請求項 7】

30

前記第 1 のイベントは、イベントストリームの一部として受信される、請求項 6 に記載のコンピュータ読取可能プログラム。

【請求項 8】

前記さらに他のイベントまたは前記フィルタリングされたイベントのうち少なくとも 1 つは、前記イベントストリームの一部として受信される、請求項 7 に記載のコンピュータ読取可能プログラム。

【請求項 9】

前記動作は、前記スキップビートを前記イベントストリームに挿入することをさらに備える、請求項 7 または 8 に記載のコンピュータ読取可能プログラム。

【請求項 10】

40

前記順序が狂ったイベントは廃棄される、請求項 6 ~ 9 のいずれか 1 項に記載のコンピュータ読取可能プログラム。

【請求項 11】

システムであって、

コンピュータによって実行可能な命令を格納するように構成されたメモリと、

前記メモリにアクセスして、前記コンピュータによって実行可能な命令を実行するように構成されたプロセッサとを備え、前記コンピュータによって実行可能な命令は、少なくとも、

一連のイベントのうち第 1 のイベントを判断する命令と、

イベントカウンタの値を前記第 1 のイベントのタイムスタンプで初期化する命令と、

50

さらに他のイベントを処理する命令と、  
フィルタリングされたイベントを識別する命令と、  
前記フィルタリングされたイベントのためのスキップビートを生成する命令とを備え、  
前記スキップビートは前記フィルタリングされたイベントのタイムスタンプを含み、  
後続のイベントを受信する命令と、  
順序が狂ったイベントを識別する命令と、  
イベントストリームにおける次のイベントが実際のイベントであるかスキップビートであるかに関係なく、前記後続のイベントの各々に関連付けられた前記タイムスタンプの順序で後続のイベントを処理する命令とを備える、システム。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

本願は、「分散型システムにおける多段階処理のためのイベント順序の保証 (GUARANTEEING THE EVENT ORDER FOR MULTI-STAGE PROCESSING IN DISTRIBUTED SYSTEMS)」と題される2016年4月11日出願された米国特許出願番号第15/095,766号からの優先権およびその利益を主張し、米国特許出願番号第15/095,766号は、「分散型システムにおける多段階処理のためのイベント順序の保証 (GUARANTEEING THE EVENT ORDER FOR MULTI-STAGE PROCESSING IN DISTRIBUTED SYSTEMS)」と題される2015年10月21日出願された米国仮出願番号第62/244,452号の非仮出願であり、  
米国特許法第119条(e)のもとでその利益および優先権を主張し、どちらも内容全体が全ての目的で引用によって本明細書に援用される。

【背景技術】

【0002】

背景

従来のデータベースシステムでは、データは、通常はテーブルの形態の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、構造化照会言語 (structured query language: SQL) などのデータ管理言語を使用して照会され操作される。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定義され実行されてもよい。したがって、SQLクエリは、データベースに格納された有限のデータセットに対して実行される。さらに、SQLクエリが実行される場合、それは有限のデータセットに対して一度実行されて、有限の静的な結果を生成する。したがって、格納された有限のデータセットに対して照会を行うようにデータベースが最適に準備される。

【発明の概要】

【発明が解決しようとする課題】

【0003】

しかし、多数の現代のアプリケーションおよびシステムは、有限のデータセットの代わりに、連続データまたはイベントストリームの形態のデータを生成する。このようなアプリケーションの例としては、センサデータアプリケーション、金融株式相場表示板、ネットワーク性能測定ツール (たとえば、ネットワーク監視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、自動車交通監視などが挙げられるが、それらに限定されるものではない。このようなアプリケーションは、データストリームを処理することができる新しいタイプのアプリケーションの必要性を生じさせてきた。たとえば、温度センサは、温度読取値を送信するように構成されてもよい。

【0004】

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の管理および処理は、強力な時間的焦点を有するデータ管理および照会機能を構築することを必要とする。連続的な無限のデータセットに対して長期間照会を行うことを含む異なる種類の照会機構が必要である。イベントストリーム処理に適合した製品一式を提供しているべ

10

20

30

40

50

ンダもいるが、これらの製品提供は、依然として、今日のイベント処理ニーズに対処するのに必要な処理柔軟性が欠如している。

【課題を解決するための手段】

【0005】

概要

本開示の以下の部分は、少なくとも主題を基本的に理解できるようにする目的で、本開示に記載されている1つ以上の革新的技術、実施形態および/または実施例の簡略化された概要を提供する。この概要は、いずれかの特定の実施形態または実施例の広範囲にわたる概略を提供しようとしているわけではない。また、この概要は、ある実施形態または実施例の重要な/不可欠な要素を識別することを意図したものではなく、または本開示の主題の範囲を図示することを意図したものでもない。したがって、この概要の1つの目的は、簡略化した形で本開示に記載されているいくつかの革新的技術、実施形態および/または実施例を、以下に記載するより詳細な説明に対する前置きとして提供することであり得る。

10

【0006】

本開示の主題の本質および等価物（ならびに、提供されるいずれの固有のまたは明確な利点および改良点も）のさらなる理解は、上記の段落に加えて、本開示の残りの部分、添付の図面および特許請求の範囲を参照することによって実現されるべきである。

【0007】

いくつかの例では、方法、システム、およびコンピュータ読取可能な媒体が提供され得る。上記方法、システムおよび/またはコンピュータ読取可能な媒体は、イベントストリームの一部として受信される一連のイベントのうちの第1のイベントを判断すること、上記第1のイベントのタイムスタンプを有するイベントカウンタの値を初期化すること、および/または、上記イベントストリームのさらに他のイベントを処理することを備え得る。さらに他のイベントは、上記第1のイベント以外の、上記一連のイベントのうちのいずれかのイベントを含み得る。いくつかの例は、上記方法、システムおよび/またはコンピュータ読取可能な媒体は、上記イベントストリームのフィルタリングされたイベントを識別すること、上記フィルタリングされたイベントのためのスキップビートを生成すること、および/または、上記スキップビートを上記イベントストリームに挿入することも備え得る。さらに、いくつかの例では、上記方法、システムおよび/またはコンピュータ読取可能な媒体は、上記イベントストリームの後続のイベントを受信すること、上記イベントストリームの順序が狂ったイベントを識別すること、および/または、上記イベントストリームにおける次のイベントが実際のイベントであるかスキップビートであるかに関係なく、上記後続のイベントの各々に関連付けられた上記タイムスタンプの順序で後続のイベントを処理することも備え得る。後続のイベントは、上記スキップビートの挿入後に上記イベントストリームの一部として受信される、上記一連のイベントのうちのいずれかのイベントを含み得る。いくつかの例では、上記方法、システムおよび/またはコンピュータ読取可能な媒体は、タイマを開始すること、上記タイマが時間切れになるまで上記一連のイベントのうちの一组のイベントを受信すること、上記一组のイベントを時系列順に再配列すること、および上記第1のイベントを、最新のタイムスタンプを有する上記再配列された組のイベントとして識別すること、によって上記第1のイベントを判断することも備え得る。さらに他のイベントは、上記タイマが時間切れになる前にバッチ処理され得る。上記フィルタリングされたイベントは、上流の段階によってフィルタリングにより除去され得る。上記実際のイベントは、上記イベントストリームに対応するイベントデータを備え得る。

20

30

40

【0008】

本開示に記載されている革新的技術、実施形態および/または実施例を合理的に説明し図示するために、1つ以上の添付の図面を参照し得る。1つ以上の添付の図面を説明するために使用されるさらなる詳細または例は、記載されている開示のいずれの範囲に対しても、ここに記載されている実施形態および/または実施例のいずれの範囲に対しても、ま

50

たは本開示に提示されている、現在のところ最良の形態であると理解されている革新的技術のいずれの範囲に対しても、限定として考えられるべきではない。

【図面の簡単な説明】

【0009】

【図1】本開示の実施形態を組み込むことができるシステムの簡略図である。

【図2】本開示の実施形態を実現するための詳細の簡略ブロック図である。

【図3】本開示の実施形態を実現するための方法のフローチャートである。

【図4】本開示の実施形態を実現するための方法のフローチャートである。

【図5】本開示の実施形態を実現するための方法のフローチャートである。

【図6】本明細書に記載されている例のうちのいくつかを実現するための、少なくとも1つの例に係る分散型システムの簡略図である。

【図7】本明細書に記載されている例のうちのいくつかに従って実施形態のシステムのコンポーネント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提供することができる、少なくとも1つの例に係るシステム環境のコンポーネントの簡略ブロック図である。

【図8】本明細書に記載されている例のうちのいくつかに従って本開示のさまざまな実施形態を実現することができる、少なくとも1つの例に係る例示的なコンピュータシステムを示す。

【発明を実施するための形態】

【0010】

詳細な説明

複合イベント処理プラットフォームでは、正確な結果を達成するために、アプリケーションタイムスタンプの順序でイベントを処理すべきである。イベント「a」がイベント「b」の前にアプリケーションによって生成された場合、イベント「a」は、「b」の前に処理されるべきである。たとえば、現在の温度が以前の温度よりも高いデバイスを識別するために、データをデバイスIDによってパーティション化して順番に処理すべきである。いくつかの例では、現在のデータと以前のデータとの間の相違は、それらのイベントのタイムスタンプによって定義される。データが単一の時点から取り込まれるシステムでは、イベントの順序付けされた処理を確実にすることは、比較的容易である。たとえば、イベントのパーティション番号を識別してもよく、処理のためのパーティションに特有の待ち行列上にイベントを配置してもよい。しかし、分散型システムでは、特にデータがクラスタ内の任意のノードから取り込まれる場合に、イベントの順序付けされた処理は非常に複雑になる。場合によっては、順序付けされた処理を確実にするためにいくつかの出来事が起こらなければならない。(1)第一に、正しいシーケンス番号またはタイムスタンプを有する開始カウンタが初期化されるべきである。(2)第二に、順序が狂った状態でイベントが到着したか否かが検出されるべきである。(3)第三に、バッファリングされたがまだ処理されていないイベントが再配列されるべきである。(4)最後に、順序が狂った古いイベントが到着するのをいつまでも待つことがないように、いつ次に進むかが判断されるべきである。これは多段階処理に特に当てはまるであろう。なぜなら、それらはパイプライン内の前の処理段階によってフィルタリングにより除去された可能性があるからである。ウォームアップおよびスキップビートの技術によって、本開示は、イベントを順番に処理するための例、およびいくつかの例では、たとえ処理が複数の段階を伴うとしてもイベントが発生順に処理されることを保証するための例を提供する。

【0011】

いくつかの例では、本開示の1つの特徴は、システム起動の冒頭におよびシステム起動時に、後続のイベントを明らかにするというものである。これは、ウォームアップ局面と呼ぶことができる。いくつかの例では、システムがまさに最初のイベントを受信するとクロックが開始され、クロックまたはタイマが時間切れになるまでイベントが収集される。タイマが時間切れになった後、イベントは再配列され、最高/最新のタイムスタンプを有するイベントを使用して、カウンタ「LAST\_SEEN\_TIMESTAMP」の値を初期化し、カウンタ「NEXT\_EXPECTED\_TIMESTAMP」は、LAST\_SEEN\_TIMESTAMP + 1 に設定される。バッチ処理さ

れたイベントが処理され、ウォームアップ局面が完了する。ウォームアップ局面後、上記の2つのカウンタを使用して順序付けが厳密に実施される。

【0012】

本開示の別の特徴は、フィルタリングされたイベントと同一のタイムスタンプを使用してスキップビートを作動させるというものである。多段階処理では、上流の段階によってイベントがフィルタリングにより除去される可能性がある。このような場合、フィルタリングされたイベントは到着することがないので、処理システム（たとえば、複合イベント処理（CEP）エンジン）は、いつまでも待つべきではなく、イベント順序付けを試みるべきである。上流の段階は、さまざまな方法でおよび/またはさまざまな理由でストリームのイベントをフィルタリングにより除去する可能性がある。いずれのイベントにおいても、イベントは、一旦フィルタリングにより除去されると、もはやストリームにおける一連のイベントの一部ではなくなってしまう。したがって、フィルタリングが行われた後は、パイプラインのどの段階にもイベントが到着することはないであろう（たとえば、イベントはCEPエンジンに到着することはできない）。このような場合に対処するために、フィルタリングされたアプリケーションと同一のタイムスタンプを有するスキップビートを生成することができ、システムは、当該スキップビートを全ての下流の段階に伝搬させることができる。次いで、実際のイベントの代わりに当該スキップビートを使用して、多段階処理を展開するアプリケーションにおける順序付けを保証する。上記のウォームアップ局面が完了すると、入来イベントは正確な順序で処理することができる（たとえば、後続のイベントは、当該後続のイベントの各々に関連付けられたタイムスタンプの順序で処理される）。このようにして、以下の論理を使用して、後続のイベントの順序付けされた処理が保証される。

【0013】

「NEXT\_EXPECTED\_TIMESTAMP」と同一のタイムスタンプを有する実際のイベントまたは「スキップビート」が到着すると、LAST\_SEEN\_TIMESTAMPは、NEXT\_EXPECTED\_TIMESTAMPの値を割り当てられ、NEXT\_EXPECTED\_TIMESTAMPはインクリメントされる。

【0014】

NEXT\_EXPECTED\_TIMESTAMPよりも高いタイムスタンプを有する「スキップビート」または「実際のイベント」が到着すると、当該イベントはバッファに追加され、予め構成された時間切れ間隔を有するタイマが開始され、以下の動作が実施される。

- ・タイマが実行し続けるにつれて、NEXT\_EXPECTED\_TIMESTAMPよりも高いタイムスタンプを有するより多くの入来イベントまたはスキップビートがバッファリングされる。
- ・NEXT\_EXPECTED\_TIMESTAMPと同一のタイムスタンプを有する入来イベントまたはスキップビートが到着すると、タイマがオフにされ、バッファ内のイベントが再配列されて処理される。スキップビートは伝搬し続けることに留意されたい。
- ・NEXT\_EXPECTED\_TIMESTAMPよりも低いタイムスタンプを有する入来イベントまたはスキップビートが到着すると、当該イベントまたはビートは帯域外であるとして廃棄される。やはり、ウォームアップ局面の長さを長くすることにより、このシナリオの可能性を減少させることができる。
- ・タイマが時間切れになると、スキップビートの伝搬を継続しながら、バッファリングされたイベントが再配列されて処理される。

【0015】

これらの技術により、カウンタの適切な初期化が可能であり、処理パイプライン内の上流の段階によってフィルタリングされ得るイベントが明らかにされる。

【0016】

いくつかの例では、クエリは、イベントが前のイベントに対して高いか低いかを判断するタスクを備え得る。たとえば、温度がこの1時間（または、その他の期間）で上昇したか否かである。しかし、この質問に正確に答えるためには、（たとえば、温度に関連する）イベントを、当該イベントが生成された順序と同一の順序で処理しなければならない。これは、単一プロセッサシステムまたはデータが単一の時点から取り込まれるシステムで

は、単に入来する通りにイベントを待ち行列に入れてそれらを順番に処理することによって、比較的容易にできる。しかし、分散型システムでは、イベントは、イベントプロセッサに到着する前に複数のノードを通過するため、順序が狂う可能性がある。

#### 【 0 0 1 7 】

記載するように、1つの解決策は、イベントプロセッサが正確な始点を有するように正確な値で開始カウンタを初期化するというものである。さもなければ、システムが起動するときに（たとえば、再始動または再起動後に）、イベントプロセッサが正確な値で始動することになることを想定できない。したがって、システムが起動したときに、ウォームアップ期間を使用して、どのイベントが適切な開始イベントであるかを判断することができる。次いで、この開始イベントを使用して、配列を開始させることができる。次いで、イベントの処理が開始すると、システムは、あるものがいつ順序が狂うかを検出できる必要がある。場合によっては、イベントが順序が狂っていると判断されると、イベントは再順序付けされなければならない。しかし、順序が狂ったイベントがいつか到着することを保証することは、必ずしも可能であるとは限らない。たとえば、場合によっては、イベントは、分散型システムにおける前のノードまたはプロセスによってフィルタリングにより除去されているかもしれない。

#### 【 0 0 1 8 】

いくつかの例では、イベントプロセッサが特定のイベントを待たなくてもいいことを示すために「スキップビート」（たとえば、システム中に伝搬されるダミーイベント）が使用される。たとえば、スキップビートは、イベントが到着することがないことを示してもよい。場合によっては、イベントが前のノードによってフィルタリングにより除去されている場合、適切なタイムスタンプ（たとえば、フィルタリングされたイベントに対応していたタイムスタンプと同一のタイムスタンプ）を有するストリームに「スキップビート」が挿入され得る。下流のプロセッサが当該「スキップビート」を受信すると、それは見つかからないイベント（たとえば、順序が狂っているように見えるイベントのうちの、プロセッサが待っているイベント）として処理され、次のイベントが検出されると、プロセッサは、次のイベントの処理に移ることになる。いくつかの例では、イベントは、前のノードにおいて処理される条件によって、フィルタリングにより除去され場合がある。たとえば、特定の閾値を上回る（たとえば、70度を超える、など）温度上昇をクエリが識別しているのみであれば、60度未満の温度に関連付けられたイベントは全てフィルタリングにより除去されてもよい。イベントがフィルタリングにより除去されると、フィルタリングされたイベントの代わりに、「スキップビート」がストリームに挿入され得る。イベントプロセッサは、当該「スキップビート」を識別すると、上記のイベントを待つことをやめる。「スキップビート」イベントは、タイムスタンプ情報のみを含む空イベントを作成することによって生成され得る。空イベントを生成する際に、タプルは、フィルタリングされたイベントのタイムスタンプを含み得る。

#### 【 0 0 1 9 】

イベントプロセッサは、どのイベントが上流でフィルタリングにより除去されたかが分からない可能性がある。そのため、イベントプロセッサは、「スキップビート」イベントを聞き取る場合、どのイベントを順番に処理しなくてもよいか（または、この場合、どのイベントも順番に処理しなくてもよいこと）が分かる。上記のように、「スキップビート」イベントは、本質的に、実際のイベントの代わりにダミーイベントである。そのため、本質的には、イベントプロセッサが正確な順序付けされたイベント処理を保証できるようにしたければ、システムは、少なくとも上記のウォームアップ期間および「スキップビート」の使用を利用しなければならない。また、イベントがいつ順序が狂うかを検出できることが重要であろう。

#### 【 0 0 2 0 】

いくつかの例では、分散型システムにおいて多段階処理のためのイベント順序を保証する機構が提供され得る。一般に、連続的なデータストリーム（イベントストリームとも称される）は、明確な終端を持たない、本来は連続的または無限であり得るデータまたはイ

10

20

30

40

50

ベントのストリームを含み得る。論理的には、イベントまたはデータストリームは、一連のデータ要素（イベントとも称される）であり得て、各データ要素は、関連付けられたタイムスタンプを有する。連続的なイベントストリームは、論理的には、一袋または一組の要素（ $s$  ,  $T$ ）として表現され得て、「 $s$ 」はデータ部分を表わし、「 $T$ 」は時間領域内にある。「 $s$ 」部分は、一般にタプルまたはイベントと称される。したがって、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。

#### 【0021】

いくつかの局面では、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、クロックタイムと同一であってもよい。しかし、他の例では、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定義されてもよく、クロックタイムに対応しているのではなくたとえば代わりにシーケンス番号によって表わされてもよい。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、番号、タイムスタンプ、または時間の概念を表わすその他の情報によって表わすことができる。入力イベントストリームを受信するシステムでは、イベントは、タイムスタンプが増加する順にシステムに到着する。同一のタイムスタンプを有する2つ以上のイベントがある場合もある。

#### 【0022】

いくつかの例では、イベントストリームにおけるイベントは、何らかの世俗的なイベントの発生（たとえば、いつ温度センサが値を新たな値に変更したか、いつ株式銘柄の価格が変化したか）を表わし得て、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俗的なイベントがいつ発生したかを示し得る。

#### 【0023】

イベントがイベントストリームを介して受信される場合、イベントに関連付けられた時間情報を使用して、当該イベントストリームにおけるイベントが、タイムスタンプ値が増加する順に到着することを保証し得る。これにより、イベントストリームにおける受信したイベントは、それらの関連付けられた時間情報に基づいて順序付けす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、後に生成されるイベントが前に生成されるイベントよりも遅いタイムスタンプを有するように、タイムスタンプは、減少しない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得る。別の例として、時間情報としてシーケンス番号が使用されている場合には、後に生成されるイベントに関連付けられたシーケンス番号は、前に生成されるイベントに関連付けられたシーケンス番号よりも大きくてもよい。いくつかの例では、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俗的なイベントが同時に発生する場合、複数のイベントが同一のタイムスタンプまたはシーケンス番号に関連付けられてもよい。同一のイベントストリームに属するイベントは、一般に、前のイベントが後のイベントよりも前に処理される状態で、関連付けられた時間情報によってイベントに対して課される順序で処理され得る。

#### 【0024】

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、ストリームを受信するシステムによって設定されてもよい。たとえば、特定の実施形態では、イベントストリームを受信するシステム上でハートビートが維持されてもよく、イベントに関連付けられた時間は、当該ハートビートによって測定される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つのイベントが同一の時間情報を有することが可能である。なお、タイムスタンプ順序付け要件は1つのイベントストリームに特有であるが、異なるストリームのイベントを任意にさしはさんでもよい。

#### 【0025】

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、当該スキーマは、時間情報と、一組の1つ以上の名前付き属性とを備える。特定のイベントストリームに属する全てのイベントは、当該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一致する。したがって、イベントストリーム（ $s$  ,  $T$ ）では、イベントストリームは、（ $\langle \text{time\_st}$

10

20

30

40

50



amp>, <attribute(s)>)としてスキーマ「S」を有し得て、<attributes>は、スキーマのデータ部分を表わし、1つ以上の属性を備え得る。たとえば、株式相場表示機イベントストリームのためのスキーマは、属性<stock symbol>および<stock price>を備えていてもよい。このようなストリームを介して受信した各イベントは、タイムスタンプと、2つの属性を有することになる。たとえば、株式相場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信し得る。

【 0 0 2 6 】

【 数 1 】

...

(<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>)

...

【 0 0 2 7 】

上記のストリームにおいて、ストリーム要素 (<timestamp\_N+1>, <ORCL,62>) では、イベントは、属性「stock\_symbol」および「stock\_value」を有する<ORCL,62>である。当該ストリーム要素に関連付けられたタイムスタンプは、「timestamp\_N+1」である。したがって、連続的なイベントストリームはイベントのフローであり、各イベントは同一の一連の属性を有する。

【 0 0 2 8 】

上記のように、ストリームは、CQLクエリが作用し得る原理データソースであり得る。ストリームSは、一袋の(「複数組の」とも称される)要素(s, T)であり得て、「s」はスキーマS内にあり、「T」は時間領域内にある。さらに、ストリーム要素は、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タプル・タイムスタンプ対であり得る。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り得る。場合によっては、同一のタイムスタンプを有する2つ以上のタプルがあってもよい。そして、入力ストリームのタプルは、タイムスタンプが増加する順にシステムに到着することを要求され得る。代替的に、関係(「時間変化関係」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」と混同すべきではない)は、時間領域からスキーマRのタプルの無限の集合へのマッピングであってもよい。いくつかの例では、関係は、タプルの順序付けされていない、時間変化的な集合(すなわち、瞬間的な関係)であってもよい。場合によっては、時間の各インスタンスにおいて、関係は、有界集合であってもよい。また、関係は、関係の変化する状態を取り込むための挿入、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もできる。ストリームと同様に、関係は、関係の各タプルが一致し得る固定のスキーマを有し得る。さらに、本明細書では、連続クエリは、一般に、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/または関係に対して照会される)データを処理することが可能であり得る。また、関係は、ストリームのデータを参照し得る。

【 0 0 2 9 】

いくつかの例では、ビジネスインテリジェンス(business intelligence: BI)が、特定の間隔で(たとえば、場合によっては毎日)ビジネス活動を駆動して最適化すること

10

20

30

40

50

を手助けし得る。このタイプのBI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(OI)と呼ばれる。いくつかの例では、オペレーショナルインテリジェンスは、BI間の回線およびビジネスアクティビティ監視(business activity monitoring: B A M)を曖昧にする。たとえば、BIは、履歴データの定期的なクエリに焦点を合わせてもよい。したがって、それは、後ろ向きの焦点を有し得る。しかし、BIは、オペレーショナルアプリケーションにも配置され得て、そのため、単なる戦略分析ツールからビジネス活動の前線に拡大することができる。したがって、BIシステムは、リアルタイムでイベントストリームを分析して総計を計算するようにも構成され得る。

#### 【0030】

10

さらに、いくつかの例では、OIは、可視性およびビジネス活動への洞察を提供することができる一種のリアルタイムの動的ビジネス解析である。OIは、BIもリアルタイムBIも大量の情報を理解することを手助けするという意味で、しばしばBIまたはリアルタイムBIに結び付けられて比較される。しかし、いくつかの基本的な違いがある。すなわち、OIは主にアクティビティ中心であるのに対して、BIは主にデータ中心である、というものである。さらに、OIは、事後の、報告ベースのパターン識別アプローチとして従来から使用されているBIとは異なって、展開中の状況(たとえば、トレンドおよびパターン)を検出して応答することにさらに適しているであろう。

#### 【0031】

いくつかの例では、ビジネスイベント分析および監視(business event analysis and monitoring: B E A M)システムは、流れている(in-flight)データを処理および/または受信するためにCQLエンジンを含み得る。たとえば、CQLエンジンは、入来リアルタイム情報に照会するか、またはそうでなければ入来リアルタイム情報を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(たとえば、BIまたはOI)であってもよい。CQLエンジンは、時間意味論を利用または理解して、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得る。場合によっては、CQLエンジンを利用することは、入来データに対して常に照会することを含み得る。

20

#### 【0032】

いくつかの局面では、CQLエンジンは、本格的な照会言語を含み得る。したがって、ユーザは、クエリの観点から計算を指定し得る。さらに、CQLエンジンは、照会言語特徴、オペレータ共有、豊かなパターンマッチング、豊かな言語構造などを利用してメモリを最適化するように設計され得る。さらに、いくつかの例では、CQLエンジンは、履歴データもストリーミングデータも処理し得る。たとえば、ユーザは、カリフォルニアセールズが特定の目標値を上回ると警告を送信するようにクエリを設定してもよい。したがって、いくつかの例では、当該警告は、少なくとも一部には履歴販売データおよび入来ライブ(すなわち、リアルタイム)販売データに基づき得る。

30

#### 【0033】

いくつかの例では、CQLエンジンまたは以下で説明する概念の他の特徴は、リアルタイムの態様で履歴文脈(すなわち、倉庫データ)と入来データとを組み合わせるように構成され得る。したがって、場合によっては、本開示は、データベース格納情報および流れている情報の境界について説明することができる。データベース格納情報も流れている情報もBIデータを含み得る。したがって、いくつかの例では、データベースは、BIサーバであってもよく、またはいずれかのタイプのデータベースであってもよい。さらに、いくつかの例では、本開示の特徴は、どのようにしてコードをプログラムするか、またはそうでなければどのようにしてコードを書込むかをユーザが知らなくても、上記の特徴を実現することを可能にし得る。言い換えれば、当該特徴は、非開発者が履歴データとリアルタイムデータとの組み合わせを実現することを可能にする機能豊富なユーザインターフェイス(UI)または他の態様で提供され得る。

40

#### 【0034】

上記および下記の技術は、多くの方法および多くの文脈で実現することができる。以

50

下でさらに詳細に説明するように、以下の図面を参照していくつかの例示的な実現例および文脈を提供する。しかし、以下の実現例および文脈は、多くのもののうちのいくつかに過ぎない。

#### 【 0 0 3 5 】

図 1 は、イベント順序を保証するための技術を実現することができ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 1 0 0 を示す。アーキテクチャ 1 0 0 では、1 人以上のユーザ 1 0 2（たとえば、アカウント所有者）は、ユーザコンピューティングデバイス 1 0 4（1）～（N）（まとめて「ユーザデバイス 1 0 4」）を利用して、1 つ以上のネットワーク 1 0 8 を介して 1 つ以上のサービスプロバイダコンピュータ 1 0 6 にアクセスし得る。また、いくつかの局面では、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 1 つ以上の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 / または 1 つ以上のデータベース 1 1 2 と通信し得る。たとえば、ユーザ 1 0 2 は、サービスプロバイダコンピュータ 1 0 6 を利用して、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 / またはデータベース 1 1 2 のデータにアクセスするか、またはそうでなければ当該データを管理してもよい（たとえば、1 1 0、1 1 2 のいずれかまたは両方に対して照会が行われてもよい）。データベース 1 1 2 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、いくつかの例では、ユーザ 1 0 2 の代わりに履歴データ、イベントデータ、関係、アーカイブされた関係などを管理し得る。さらに、データベース 1 1 2 は、ストリーミングデータソースコンピュータ 1 1 0 によって提供されるデータを受信するか、またはそうでなければ格納し得る。いくつかの例では、ユーザ 1 0 2 は、ユーザデバイス 1 0 4 を利用して、データ（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）に対するクエリ（「クエリステートメント」とも称される）または他の要求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 1 0 6 と対話し得る。次いで、このようなクエリまたは要求は、サービスプロバイダコンピュータ 1 0 6 によって実行されて、データベース 1 1 2 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 1 1 0 からの入来データを処理し得る。さらに、いくつかの例では、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 / またはデータベース 1 1 2 は、サービスプロバイダコンピュータ 1 0 6 に関連付けられた統合分散環境の一部であってもよい。

#### 【 0 0 3 6 】

いくつかの例では、ネットワーク 1 0 8 は、ケーブルネットワーク、インターネット、無線ネットワーク、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、他の私的ネットワークおよび / もしくは公共ネットワークなどの複数の異なるタイプのネットワークのいずれか 1 つまたは組み合わせを含み得る。示されている例は、ユーザ 1 0 2 がネットワーク 1 0 8 を介してサービスプロバイダコンピュータ 1 0 6 にアクセスすることを示しているが、記載されている技術は、ユーザ 1 0 2 が固定電話を介して、キオスクを介して、またはその他の態様で 1 つ以上のユーザデバイス 1 0 4 を介して 1 つ以上のサービスプロバイダコンピュータ 1 0 6 と対話する場合に等しく適用することができる。なお、また、記載されている技術は、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）および非クライアント / サーバ構成（たとえば、ローカルに格納されたアプリケーションなど）において適用することもできる。

#### 【 0 0 3 7 】

ユーザデバイス 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 などであるがそれらに限定されないいずれかのタイプのコンピューティングデバイスであってもよい。いくつかの例では、ユーザデバイス 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 と通信し得る。さらに、ユーザデバイス 1 0 4 は、処理すべきデータベース 1 1 2（または、他のデータストア）のデータを要求するための 1 つ以上のクエリまたはクエリステートメントを提供するようにも構成され得る。

## 【 0 0 3 8 】

また、いくつかの局面では、サービスプロバイダコンピュータ 1 0 6 は、サーバなどの、モバイルコンピューティングデバイス、デスクトップコンピューティングデバイス、シンクライアントコンピューティングデバイスおよび/またはクラウドコンピューティングデバイスなどであるがそれらに限定されないいずれかのタイプのコンピューティングデバイスであってもよい。いくつかの例では、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、ユーザデバイス 1 0 4 と通信し得る。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、恐らくクラスタ状態で配置された 1 つ以上のサーバを含み得る。これらのサーバは、本明細書に記載されている特徴を実行するか、またはそうでなければ提供するように構成され得て、本明細書に記載されている特徴は、アーカイブされた関係の管理、アーカイブされた関係に関連付けられた構成可能なデータウィンドウ、および/または、本明細書に記載されているアーカイブされた関係の管理に関連付けられた変更イベントの正確な集計を含むが、それらに限定されるものではない。さらに、いくつかの局面では、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/またはデータベース 1 1 2 を含む統合分散コンピューティング環境の一部として構成され得る。

10

## 【 0 0 3 9 】

1 つの例示的な構成では、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 3 6 と、1 つ以上の処理ユニット（または、プロセッサ）1 3 8 とを含み得る。プロセッサ 1 3 8 は、ハードウェア、コンピュータによって実行可能な命令、ファームウェア、またはそれらの組み合わせで適宜実現され得る。プロセッサ 1 3 8 のコンピュータによって実行可能な命令またはファームウェア実現例は、記載されているさまざまな機能を実行するために、任意の好適なプログラミング言語で書かれた、コンピュータによって実行可能なまたはマシンによって実行可能な命令を含み得る。

20

## 【 0 0 4 0 】

メモリ 1 3 6 は、プロセッサ 1 3 8 上でロード可能であり実行可能であるプログラム命令、および、これらのプログラムの実行中に生成されるデータを格納し得る。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 3 6 は、揮発性であってもよく（ランダムアクセスメモリ（random access memory : R A M ）など）、および/または、不揮発性であってもよい（リードオンリメモリ（read-only memory : R O M ））、フラッシュメモリなど）。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、リムーバブルストレージおよび/または非リムーバブルストレージを含み得るさらに他のストレージ 1 4 0 も含み得る。さらに他のストレージ 1 4 0 は、磁気ストレージ、光ディスクおよび/またはテープストレージを含み得るが、それらに限定されるものではない。ディスクドライブおよびそれらの関連付けられた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、およびコンピューティングデバイスのための他のデータの揮発性ストレージを提供し得る。いくつかの実現例では、メモリ 1 3 6 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 などの複数の異なるタイプのメモリを含み得る。

30

40

## 【 0 0 4 1 】

メモリ 1 3 6 、さらに他のストレージ 1 4 0 は、全て、リムーバブルなものも非リムーバブルなものも、コンピュータ読取可能な記憶媒体の例である。たとえば、コンピュータ読取可能な記憶媒体は、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または他のデータなどの情報の格納のためのいずれかの方法または技術で実現される揮発性または不揮発性の、リムーバブルまたは非リムーバブルな媒体を含み得る。メモリ 1 3 6 およびさらに他のストレージ 1 4 0 は全て、コンピュータ記憶媒体の例である。

## 【 0 0 4 2 】

サービスプロバイダコンピュータ 1 0 6 は、アイデンティティインターフェイスコンピ

50

ユータ 120 が、格納されたデータベース、別のコンピューティングデバイスまたはサーバ、ユーザ端末、および/または、ネットワーク 108 上の他のデバイスと通信することを可能にする通信接続 142 も含み得る。サービスプロバイダコンピュータ 106 は、キーボード、マウス、ペン、音声入力デバイス、タッチ入力デバイス、ディスプレイ、1つ以上のスピーカ、プリンタなどの入力/出力(I/O)デバイス 144 も含み得る。

#### 【0043】

メモリ 136 のコンテンツをさらに詳細に見てみると、メモリ 136 は、オペレーティングシステム 146 と、本明細書に開示されてい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て、本明細書に開示されている特徴は、少なくとも、ウォームアップモジュール 148、(たとえば、レスリー・ランポータル  
10  
アルゴリズムなどを利用した)順序狂い検出モジュール 150、および/または、スキップビートモジュール 152 を含む。ウォームアップモジュール 148 は、上記のウォームアップ期間を可能にするように構成され得る。順序狂い検出モジュール 150 は、イベントを受信したときに順序が狂ったイベントの検出を可能にするように構成され得る。そして、スキップビートモジュール 152 は、いつイベントがフィルタリングにより除去されたかを判断して、スキップビートを生成し、それらを適宜ストリームに挿入するように構成され得る。本明細書では、モジュールは、サービスの一部である、サーバまたはサーバのクラスタによって実行されるプログラミングモジュールを指し得る。この特定の文脈において、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  
20  
サーバのクラスタによって実行され得る。

#### 【0044】

図 2 は、本開示の実施形態を組み込むことができるイベント処理システム 200 の簡略化されたハイレベル図を示す。イベント処理システム 200 は、1つ以上のイベントソース(204, 206, 208)と、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たイベント処理サーバ(EPS) 202(CQサービス 202とも称される)と、1つ以上のイベントシンク(210, 212)とを備え得る。イベントソースは、EPS 202 によって受信されるイベントストリームを生成する。EPS 202 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信し得る。たとえば、図 2 に示されるように、EPS 202 は、イベントソース 204 から入力イベントストリーム 214 を受信し、イベントソース 206 から第2の入力イベントストリーム 216 を受  
30  
信し、イベントソース 208 から第3のイベントストリーム 218 を受信する。1つ以上のイベント処理アプリケーション(220, 222 および 224)が EPS 202 上にデプロイされて EPS 202 によって実行され得る。EPS 202 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を聞き取り、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1つ以上のイベントを選択する処理論理に基づいて、1つ以上のイベントストリームを介して受信したイベントを処理するように構成され得る。次いで、当該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態で1つ以上のイベントシンク(210, 212)に送信され得る。たとえば、図 2 では、EPS 202 は、出力イベントストリーム 226 をイベントシンク 210 に出力し、第2の出力イベントストリーム 228 をイベントシンク 212 に出力する。特定の  
40  
実施形態では、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、他のコンポーネントに対する変更を生じさせることなくこれらのコンポーネントのうちのいずれかを追加または除去することができるよう互いに分離される。

#### 【0045】

一実施形態では、EPS 202 は、共有サービスを有する、Equinox OSGiに基づくものなどの軽量 Java(登録商標)アプリケーションコンテナを備える Javaサーバとして実現され得る。いくつかの実施形態では、EPS 202 は、たとえば JRockit Real Time を使用してイベントを処理するための超高スループットおよびマイクロ秒待ち時間をサ  
50  
ポートし得る。また、EPS 202 は、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、オラクル CEP ビジュアライザおよびオラクル 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のエンドツーエンドＪａｖａイベント駆動アーキテクチャ（Event-Driven Architecture：ＥＤＡ）開発プラットフォーム）を提供し得る。

【００４６】

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を聞き取って、１つ以上の入力イベントストリームから１つ以上の注目すべきイベントを選択するための論理（たとえば、クエリ）を実行し、選択され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２は、１つのこのようなイベント処理アプリケーション２２０のためのドリルダウンを提供する。図２に示されるように、イベント処理アプリケーション２２０は、入力イベントストリーム２１８を聞き取って、入力イベントストリーム２１８から１つ以上の注目すべきイベントを選択するための論理を備えるクエリ２３０を実行し、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２２８を介してイベントシンク２１２に出力するように構成される。イベントソースの例としては、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、キャッシュなどが挙げられるが、それらに限定されるものではない。イベントシンクの例としては、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、キャッシュなどが挙げられるが、それらに限定されるものではない。

10

【００４７】

図２におけるイベント処理アプリケーション２２０は、１つの入力ストリームを聞き取って、選択されたイベントを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、これは限定的であるよう意図されるものではない。代替的な実施形態では、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聞き取って、監視されたストリームからイベントを選択し、選択されたイベントを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得る。同一のクエリを２つ以上のイベントシンクおよび異なるタイプのイベントシンクに関連付けることができる。

20

【００４８】

無限の性質のために、イベントストリームを介して受信されるデータの量は、一般に非常に多い。その結果、照会目的で全てのデータを格納またはアーカイブすることは、一般に非現実的であり、望ましくない。イベントストリームの処理は、受信した全てのイベントデータを格納する必要なしにイベントがＥＰＳ２０２によって受信されるときにリアルタイムでイベントの処理を必要とする。したがって、ＥＰＳ２０２は、受信した全てのイベントを格納する必要なしにイベントがＥＰＳ２０２によって受信されるときにイベントの処理を実行することを可能にする特別な照会機構を提供する。

30

【００４９】

イベント駆動アプリケーションは、ルール駆動であり、これらのルールは、入力ストリームを処理するために使用される連続クエリの形態で表現され得る。連続クエリは、クエリ処理の結果として、どのイベントを注目すべきイベントとして選択して出力すべきかを含む、受信したイベントに対して実行される処理を識別する命令（たとえば、ビジネス論理）を備え得る。連続クエリは、データストアに留まって、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に使用され得る。連続クエリは、一般に、フィルタリングおよび集約機能を実行して、入力イベントストリームから注目すべきイベントを発見して抽出する。その結果、出力イベントストリームにおける送信イベントの数は、一般に、イベントが選択され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はるかに少なくなる。

40

【００５０】

有限のデータセットに対して一度実行されるＳＱＬクエリとは異なって、ＥＰＳ２０２を有するアプリケーションによって特定のイベントストリームのために登録された連続クエリは、当該イベントストリームにおいてイベントが受信されるたびに実行され得る。連

50

続クエリの実行の一部として、E P S 2 0 2 は、連続クエリによって指定された命令に基づいて、受信したイベントを評価して、連続クエリの実行の結果として、1 つ以上のイベントを注目すべきイベントとして選択すべきか否かを判断して出力する。

【 0 0 5 1 】

連続クエリは、さまざまな言語を使用してプログラムされ得る。特定の実施形態では、連続クエリは、オラクル社によって提供される C Q L を使用して構成され、オラクルの複合イベント処理 ( C E P ) 製品提供によって使用され得る。オラクルの C Q L は、イベントストリームに対して実行可能なクエリ ( C Q L クエリと称される ) をプログラムするために使用できる宣言型言語である。特定の実施形態では、C Q L 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追加の構造を有する S Q L に基づく。

10

【 0 0 5 2 】

一実施形態では、イベント処理アプリケーションは、以下のコンポーネントタイプで構成され得る。

【 0 0 5 3 】

( 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接続する 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理解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によって照会可能な正規化された形態に変換する役割を担う。アダプタは、正規化されたイベントデータをチャ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得る。イベントアダプタは、さまざまなデータソースおよびシンクについて定義され得る。

20

【 0 0 5 4 】

( 2 ) イベント処理終点の役割を果たす 1 つ以上のチャネル。とりわけ、チャネルは、イベント処理エージェントがイベントデータに対して作用することができるまでイベントデータを待ち行列に入れる役割を担う。

【 0 0 5 5 】

( 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、正規化されたイベントデータをチャネルから消費し、クエリを使用してそれを処理して注目すべきイベントを選択し、選択された注目すべきイベントを出力チャネルに転送 ( または、コピー ) するように構成される。

【 0 0 5 6 】

( 4 ) 1 つ以上のビーンは、出力チャネルを聞き取るように構成され、出力チャネルへの新たなイベントの挿入によって作動される。いくつかの実施形態では、このユーザコードは、プレーンオールド 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、ウェブサービスおよびファイルライタなどの外部サービス一式を活用して、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。

30

【 0 0 5 7 】

( 5 ) イベントビーンは、出力チャネルを聞き取るように登録され得て、出力チャネルへの新たなイベントの挿入によって作動される。いくつかの実施形態では、このユーザコードは、ビーンがオラクル C E P によって管理できるようにオラクル C E P イベントビーン A P I を使用し得る。

40

【 0 0 5 8 】

一実施形態では、イベントアダプタは、イベントデータを入力チャネルに提供する。入力チャネルは、入力チャネルによって提供されるイベント上で動作する 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ネルに接続される。

【 0 0 5 9 】

いくつかの実施形態では、イベント処理アプリケーションのさまざまなコンポーネント、コンポーネントがどのように接続されるか、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記述するアセンブリファイルがイベント処理アプリケーションに対して提供

50

され得る。イベントの選択のために連続クエリまたはビジネス論理を指定するために別々のファイルが提供されてもよい。

【0060】

図2に示されるシステム200は、図2に示されているコンポーネント以外の他のコンポーネント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2に示される実施形態は、本開示の実施形態を組み込むことができるシステムの一例に過ぎない。いくつかの他の実施形態では、システム200は、図2に示されるよりも多くのコンポーネントもしくは少ないコンポーネントを有していてもよく、2つ以上のコンポーネントを組み合わせてもよく、またはコンポーネントの異なる構成もしくは配置を有していてもよい。システム200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、携帯電話またはデバイス）、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、またはその他のデータ処理システムを含むさまざまなタイプのシステムであってもよい。いくつかの他の実施形態では、システム200は、システム200の1つ以上のコンポーネントがクラウド内の1つ以上のネットワークにわたって分散される分散型システムとして構成されてもよい。

【0061】

図2に示されるコンポーネントのうちの1つ以上は、ソフトウェア、ハードウェア、またはそれらの組み合わせで実現されてもよい。いくつかの実施形態では、ソフトウェアは、メモリ内に格納されてもよく（たとえば、非一時的なコンピュータ読取可能な媒体）、メモリデバイス上に格納されてもよく、または何らかの他の物理メモリ上に格納されてもよく、1つ以上の処理ユニット（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、1つ以上のGPUなど）によって実行されてもよい。

【0062】

連続的な予定されたクエリのハイブリッド実行を実現するための例示的な方法およびシステムについて上記した。これらのシステムおよび方法のうちのいくつかまたは全ては、少なくとも部分的に、少なくとも上記の図1～図2に示されるようなアーキテクチャおよびプロセスによって実現されてもよいが、そうでなくてもよい。

【0063】

図3～図5は、それぞれ、本開示の実施形態に係る多段階処理のイベント順序を保証するための方法300、400および500のフローチャートである。図3～図5に示される方法300、400および500の実現例または処理は、コンピュータシステムまたは情報処理デバイスなどの論理マシンの中央処理装置（CPUまたはプロセッサ）によって実行されたときにはソフトウェア（たとえば、命令またはコードモジュール）によって実行されてもよく、電子デバイスまたは特定用途向け集積回路のハードウェアコンポーネントによって実行されてもよく、またはソフトウェアとハードウェア要素との組み合わせによって実行されてもよい。本明細書および以下に記載されているプロセスのうちのいずれかの各ステップは、所望のごとく、いかなる順序で実行されてもよく、省略（またはそうでなければスキップ）されてもよく、他のステップと置換されてもよく、および/または、繰返しもしくは再帰的に実行されてもよい。

【0064】

図3に示される方法300は、302から開始し得て、302において、少なくともタイマが時間切れになるまで起こるウォームアップ期間に少なくとも部分的に基づいてイベントストリームの第1のイベントを判断（または、識別）し得る。いくつかの例では、304において、第1のイベントのタイムスタンプでイベントカウンタを初期化し得る。306において、イベントストリームのさらに他のイベントを処理し得る。さらに他のイベントは、タイマが時間切れになる前にバッチ処理されているかもしれない。いくつかの例では、308において、フィルタリングされたイベントを識別し得る。フィルタリングされたイベントは、上流の段階によってフィルタリングにより除去されているかもしれない。310において、フィルタリングされたイベントのために、スキップビートを生成し、および/または、スキップビートをストリームに挿入し得る。312において、イベント



ストリームの後続のイベントを受信し得る。場合によっては、314において、順序が狂ったイベントを識別し得る。いくつかの例では、方法300は、後続のイベントを順番に処理することによって、および/または、後続のイベントが順番に処理されることを保証することによって、316において終了し得る。これは、各々の入来イベントが実際のイベントであるかスキップビートであるかに関係なくなされ得る。言い換えれば、スキップビートを受信されると、各イベントは順番に処理されることになる。いくつかの例では、スキップビートは、イベントデータを含まないで実際のイベントではない(たとえば、それは、CEPエンジンが、フィルタリングされたイベントを待つことなくストリームを処理し続けることを可能にするダミーイベントである)。

#### 【0065】

10

図4は、本開示の実施形態に係る多段階処理のイベント順序を保証するための方法400のフローチャートである。図4に示される方法400は、402から開始し得て、402において、タイマを開始し得る。いくつかの例では、404において、タイマが時間切れになるまでストリームのイベントを受信し得る。406において、入来イベントを時系列順に再配列(または、配列)し得る(たとえば、順序が狂っている場合には並べる)。方法400のウォームアップ期間は、第1のイベントを、少なくとも部分的に最高の(最古の、最新の、最も最近の)タイムスタンプを有するイベントに基づく第1のイベントとして識別する408において終了し得る。

#### 【0066】

20

図5は、本開示の実施形態に係る多段階処理のイベント順序を保証するための方法500のフローチャートである。図5に示される方法500は、502から開始し得て、502において、イベントを受信し、どのタイプのイベントを受信されたかおよびイベントのタイムスタンプを判断する。いくつかの例では、502において、次の予想されるタイムスタンプよりも高いタイムスタンプを有するスキップビートまたは実際のイベントが到着すると、当該イベントがバッファに追加され、タイマが開始される。タイマは、予め構成された時間切れ間隔を有し得る。504において、より多くの実際のイベントまたはスキップビートを受信される。それらが次の予想されるタイムスタンプよりも高いタイムスタンプを有する場合には、それらもバッファに追加され、タイマが動作し続ける。506において、次の予想されるタイムスタンプと同一のタイムスタンプを有する実際のイベントまたはスキップビートが到着すると、タイマがオフにされ得て、バッファ内のイベントが再配列および/または処理され得る。なお、スキップビートは適宜伝搬され続ける。508において、次の予想されるタイムスタンプよりも低いタイムスタンプを有する実際のイベントまたはスキップビートが到着すると、当該イベントは帯域外として廃棄され得る。いくつかの例では、方法500は、タイマが時間切れになるとバッファリングされたイベントを再配列および/または処理することによって510において終了し得る。また、方法500は、この間にスキップビートを所望のごとく伝搬し続け得る。

30

#### 【0067】

図6は、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600の簡略図を示す。示されている実施形態では、分散型システム600は、1つ以上のクライアントコンピューティングデバイス602, 604, 606および608を含み、それらは、1つ以上のネットワーク610を介してウェブブラウザ、所有権付きクライアント(たとえばオラクルフォームズ(Oracle Forms))などのクライアントアプリケーションを実行および動作させるように構成される。サーバ612は、リモートクライアントコンピューティングデバイス602, 604, 606および608とネットワーク610を介して通信可能に結合されてもよい。

40

#### 【0068】

さまざまな実施形態では、サーバ612は、システムのコンポーネントのうちの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。当該サービスまたはソフトウェアアプリケーションは、非仮想および仮想環境を含み得る。仮想環境は、仮想イベント、展示会、シミュレータ、教室

50

、ショッピングでのやりとり、企業、二次元または三次元（３Ｄ）表示、ページベースの論理環境などに使用されるものを含み得る。いくつかの実施形態では、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（ＳaaS）モデルの下で、クライアントコンピューティングデバイス６０２，６０４，６０６および／または６０８のユーザに対して提供されてもよい。クライアントコンピューティングデバイス６０２，６０４，６０６および／または６０８を動作させるユーザは、次いで、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６１２と対話して、これらのコンポーネントによって提供されるサービスを利用してもよい。

#### 【００６９】

図に示される構成では、システム６００のソフトウェアコンポーネント６１８，６２０および６２２は、サーバ６１２上で実現されるものとして示されている。他の実施形態では、システム６００のコンポーネントのうちの１つ以上および／またはこれらのコンポーネント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デバイス６０２，６０４，６０６および／または６０８のうちの１つ以上によって実現されてもよい。クライアントコンピューティングデバイスを動作させるユーザは、次いで、１つ以上のクライアントアプリケーションを利用して、これらのコンポーネントによって提供されるサービスを用いてもよい。これらのコンポーネントは、ハードウェア、ファームウェア、ソフトウェア、またはそれらの組み合わせで実現されてもよい。分散型システム６００とは異なってもよい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。図に示される実施形態は、したがって、実施形態のシステムを実現するための分散型システムの一例であり、限定的であるよう意図されるものではない。

#### 【００７０】

クライアントコンピューティングデバイス６０２，６０４，６０６および／または６０８は、携帯可能な手持ち式のデバイス（たとえば、iPhone（登録商標）、セルラー電話、iPad（登録商標）、コンピューティングタブレット、携帯情報端末（PDA））またはウェアラブルデバイス（たとえばGoogle Glass（登録商標）頭部装着型ディスプレイ）であってもよく、Microsoft Windows Mobile（登録商標）などのソフトウェア、および／もしくは、iOS、Windows Phone、Android、BlackBerry 8、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、電子メール、ショートメッセージサービス（short message service：SMS）、BlackBerry（登録商標）、または他のインイーブルにされた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デバイスは、汎用パーソナルコンピュータであってもよく、一例として、Microsoft Windows（登録商標）、Apple Macintosh（登録商標）および／またはLinux（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デバイスは、たとえばGoogle Chrome OSなどのさまざまなGNU/Linuxオペレーティングシステムを限定を伴うことなく含む、さまざまな市場で入手可能なUNIX（登録商標）またはUNIXのようなオペレーティングシステムのいずれかを実行するワークステーションコンピュータであり得る。代替的に、または加えて、クライアントコンピューティングデバイス６０２，６０４，６０６および６０８は、ネットワーク６１０を介して通信することができる、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化されるゲームシステム（たとえば、Kinect（登録商標）ジェスチャ入力デバイスを伴うかまたは伴わないMicrosoft Xboxゲームコンソール）および／または個人メッセージ伝達デバイスなどのその他の電子デバイスであってもよい。

#### 【００７１】

例示的な分散型システム６００は４つのクライアントコンピューティングデバイスとともに示されているが、任意の数のクライアントコンピューティングデバイスがサポートさ

10

20

30

40

50

れてもよい。センサを伴うデバイスなど、他のデバイスがサーバ612と対話してもよい。

【0072】

分散型システム600におけるネットワーク610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）、AppleTalkなどを限定を伴うことなく含む、さまざまな市場で入手可能な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、当業者が精通している任意のタイプのネットワークであってもよい。単に一例として、ネットワーク610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610は、ワイドエリア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610は、仮想ネットワークを含み得て、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（IEEE）602.11のプロトコル式、ブルートゥース（登録商標）、および/もしくはその他の無線プロトコルのうちのいずれかの下で動作するネットワーク）、ならびに/またはこれらのいずれかの組み合わせおよび/もしくは他のネットワークを含むが、それらに限定されるものではない。

【0073】

サーバ612は、1つ以上の汎用コンピュータ、専用の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、ラックマウント型サーバなどを含む）、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/もしくは組み合わせで構成されてもよい。サーバ612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、または仮想化を伴う他のコンピューティングアーキテクチャを含み得る。論理ストレージデバイスの1つ以上の柔軟なプールを仮想化してサーバのために仮想ストレージデバイスを維持することができる。仮想ネットワークを、サーバ612によって、ソフトウェア定義のネットワーク接続を用いて制御することができる。さまざまな実施形態において、サーバ612は、前述の開示に記載される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。たとえば、サーバ612は、本開示の実施形態に従って上記の処理を実行するためのサーバに対応してもよい。

【0074】

サーバ612は、上記のもののうちのいずれかを含むオペレーティングシステム、および任意の市場で入手可能なサーバオペレーティングシステムを実行してもよい。サーバ612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（コモンゲートウェイインターフェイス）サーバ、JAVAサーバ、データベースサーバなどを含むさまざまなさらに他の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も実行してもよい。例示的なデータベースサーバは、オラクル、マイクロソフト、サイベース、IBM（国際ビジネスマシズ）などから市場で入手可能なものを含むが、それらに限定されるものではない。

【0075】

いくつかの実現例では、サーバ612は、クライアントコンピューティングデバイス602, 604, 606および608のユーザから受信されるデータフィードおよび/またはイベント更新情報を解析および整理統合するための1つ以上のアプリケーションを含んでもよい。一例として、データフィードおよび/またはイベント更新情報は、センサデータアプリケーション、金融株式相場表示板、ネットワーク性能測定ツール（たとえば、ネットワーク監視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、自動車交通監視などに関連するリアルタイムのイベントを含んでもよい、1つ以上

の第三者情報源および連続データストリームから受信される、Twitter（登録商標）フィード、Facebook（登録商標）更新情報またはリアルタイムの更新情報を含んでもよいが、それらに限定されるものではない。サーバ612は、データフィードおよび/またはリアルタイムのイベントをクライアントコンピューティングデバイス602, 604, 606および608の1つ以上の表示デバイスを介して表示するための1つ以上のアプリケーションも含んでもよい。

#### 【0076】

分散型システム600は、1つ以上のデータベース614および616も含んでもよい。データベース614および616は、さまざまな位置にあってもよい。一例として、データベース614および616のうちの1つ以上は、サーバ612に局在する（および/またはサーバ612に常駐する）非一時的な記憶媒体にあってもよい。代替的に、データベース614および616は、サーバ612から遠隔にあり、ネットワークベースまたは専用の接続を介してサーバ612と通信してもよい。一組の実施形態では、データベース614および616は、記憶域ネットワーク（storage-area network：SAN）にあってもよい。同様に、サーバ612に帰する機能を実行するための任意の必要なファイルが、適宜、サーバ612上においてローカルに、および/または遠隔で格納されてもよい。一組の実施形態では、データベース614および616は、SQLフォーマットされたコマンドに応答してデータを格納、更新および検索取得するように適合される、オラクル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んでもよい。

#### 【0077】

図7は、本開示の実施形態に従って、実施形態のシステムの1つ以上のコンポーネント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てもよいシステム環境700の1つ以上のコンポーネントの簡略ブロック図である。示されている実施形態では、システム環境700は、1つ以上のクライアントコンピューティングデバイス704, 706および708を含み、1つ以上のクライアントコンピューティングデバイス704, 706および708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702と対話するようにユーザによって使用されてもよい。クライアントコンピューティングデバイスは、ウェブブラウザ、所有権付き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）、または何らかの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てもよく、当該クライアントアプリケーションは、クラウドインフラストラクチャシステム702と対話して、クラウドインフラストラクチャシステム702によって提供されるサービスを使用するようにクライアントコンピューティングデバイスのユーザによって使用されてもよい。

#### 【0078】

図に示されるクラウドインフラストラクチャシステム702は図示されるもの以外の他のコンポーネントを有してもよいことが理解されるべきである。さらに、図に示される実施形態は、本開示の実施形態を組み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では、クラウドインフラストラクチャシステム702は、図に示されるよりも多くのコンポーネントもしくは少ないコンポーネントを有していてもよく、2つ以上のコンポーネントを組み合わせてもよく、またはコンポーネントの異なる構成もしくは配置を有してもよい。

#### 【0079】

クライアントコンピューティングデバイス704, 706および708は、602, 604, 606および608に対して上記したものと同様のデバイスであってもよい。

#### 【0080】

例示的なシステム環境700が3つのクライアントコンピューティングデバイスとともに示されているが、任意の数のクライアントコンピューティングデバイスがサポートされてもよい。センサを伴うデバイスなどの他のデバイスがクラウドインフラストラクチャシステム702と対話してもよい。

#### 【0081】

10

20

30

40

50

ネットワーク 710 は、クライアント 704, 706 および 708 とクラウドインフラストラクチャシステム 702 との間におけるデータの通信および交換を容易にしてもよい。各ネットワークは、ネットワーク 610 に対して上記したものを含む、さまざまな市場で入手可能な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、当業者が精通している任意のタイプのネットワークであってもよい。

#### 【0082】

クラウドインフラストラクチャシステム 702 は、1 つ以上のコンピュータ、および / または、サーバ 612 に対して上記したものを含み得るサーバを備え得る。

#### 【0083】

特定の実施形態では、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、オンラインデータストレージおよびバックアップソリューション、ウェブベースの電子メールサービス、ホスト型オフィススイートおよび文書コラボレーションサービス、データベース処理、管理された技術サポートサービスなどの、オンデマンドでクラウドインフラストラクチャシステムのユーザに利用可能にされるサービスのホストを含んでもよい。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、動的にスケールしてそのユーザのニーズを満たすことができる。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ある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、クラウドサービスプロバイダのシステムからインターネットなどの通信ネットワークを介してユーザに利用可能にされる任意のサービスは、「クラウドサービス」と称される。典型的には、パブリックなクラウド環境においては、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のサーバおよびシステムとは異なる。たとえば、クラウドサービスプロバイダのシステムは、アプリケーションを运营管理してもよく、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、アプリケーションをオーダーし使用してもよい。

#### 【0084】

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、ストレージ、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、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他の態様で当該技術分野において公知であるような他のサービスに対する保護されたコンピュータネットワークアクセスを含んでもよい。たとえば、サービスは、クラウド上のリモートストレージに対するインターネットを介してのパスワード保護されたアクセスを含むことができる。別の例として、サービスは、ネットワーク接続された開発者による個人的な使用のために、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むことができる。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において运营管理される電子メールソフトウェアアプリケーションに対するアクセスを含むことができる。

#### 【0085】

特定の実施形態では、クラウドインフラストラクチャシステム 702 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で、信頼性があり、高可用性の、安全な態様で顧客に対して配送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を含んでもよい。そ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 (Oracle Public Cloud) である。

#### 【0086】

「ビッグデータ」は、多数のレベルにおいて、および異なるスケールで、インフラストラクチャシステムによって运营管理および / または操作され得る。大量のデータを視覚化し、傾向を検出し、および / またはその他の態様でデータと対話するように非常に大きなデータセットがアナリストおよび研究者によって格納および操作され得る。そのようなデータを表示するため、またはデータもしくはデータが表現するものに対する外部の力をシ

10

20

30

40

50

ミュレートするために、並列に連結された何十、何百または何千ものプロセッサがそのようなデータに対して作用することができる。これらのデータセットは、データベース状にもしくはさもなければ構造モデルに従って編成されたものなどの構造化データ、および/または、非構造化データ（たとえば、電子メール、画像、データプロブ（バイナリラージオブジェクト）、ウェブページ、複合イベント処理）を含み得る。より多くの（またはより少ない）計算リソースを比較的迅速に対象に集中させることができる実施形態の機能を活用することによって、クラウドインフラストラクチャシステムは、会社、政府機関、研究組織、私人、同じ考えを持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の要求に基づいて大きなデータセットに対してタスクを実行することにさらに利用できるであろう。

10

**【0087】**

さまざまな実施形態では、クラウドインフラストラクチャシステム702は、クラウドインフラストラクチャシステム702によって提供されるサービスに対する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し、管理し、および追跡するように適合されてもよい。クラウドインフラストラクチャシステム702は、クラウドサービスをさまざまなデプロイメントモデルを介して提供してもよい。たとえば、サービスは、クラウドインフラストラクチャシステム702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、サービスが一般大衆または異なる業界企業に対して利用可能にされるパブリッククラウドモデルの下で提供されてもよい。別の例として、サービスは、クラウドインフラストラクチャシステム702が単一の組織に対してのみ動作され、その組織内における1つ以上のエンティティに対してサービスを提供してもよいプライベートクラウドモデルの下で提供されてもよい。また、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702およびクラウドインフラストラクチャシステム702によって提供されるサービスが、関連するコミュニティにおける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下で提供されてもよい。また、クラウドサービスは、2つ以上の異なるモデルの組み合わせであるハイブリッドクラウドモデルの下で提供されてもよい。

20

**【0088】**

いくつかの実施形態では、クラウドインフラストラクチャシステム702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（SaaS）カテゴリ、プラットフォーム・アズ・ア・サービス（PaaS）カテゴリ、インフラストラクチャ・アズ・ア・サービス（IaaS）カテゴリ、またはハイブリッドサービスを含む他のサービスのカテゴリの下で提供される1つ以上のサービスを含んでもよい。顧客は、サブスクリプションオーダーを介して、クラウドインフラストラクチャシステム702によって提供される1つ以上のサービスをオーダーしてもよい。クラウドインフラストラクチャシステム702は、次いで、処理を実行して、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。

30

**【0089】**

いくつかの実施形態では、クラウドインフラストラクチャシステム702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んでもよいが、そ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、クラウドインフラストラクチャシステムによってSaaSプラットフォームを介して提供されてもよい。SaaSプラットフォームは、SaaSカテゴリに入るクラウドサービスを提供するように構成されてもよい。たとえば、SaaSプラットフォームは、一連のオンデマンドアプリケーションを統合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構築し配送する機能を提供してもよい。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てもよい。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、クラウドインフラストラクチャシステムにおいて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、別個の

40

50

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。その例としては、大きな組織に販売実績管理、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、それらに限定されるものではない。

【0090】

いくつかの実施形態では、プラットフォームサービスは、クラウドインフラストラクチャシステムによって PaaS プラットフォームを介して提供されてもよい。PaaS プラットフォームは、PaaS カテゴリに入るクラウドサービスを提供するように構成されてもよい。プラットフォームサービスの例としては、(オラクルなどの)組織が既存のアプリケーションを共有の共通のアーキテクチャにおいて整理統合することができるサービス、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、そ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てもよい。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提供される PaaS サービスを、別個のライセンスおよびサポートを購入する必要なく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(Oracle Java Cloud Service: JCS)、オラクル・データベース・クラウド・サービス(Oracle Database Cloud Service: DBCS)などが挙げられるが、それらに限定されるものではない。

【0091】

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使用することができ、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では、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んでもよい。一実施形態では、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、顧客にデータベース・アズ・ア・サービスをデータベースクラウドの形式で提供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てもよい。ミドルウェアクラウドサービスは、顧客が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のプラットフォームをクラウドインフラストラクチャシステムにおいて提供してもよく、Java クラウドサービスは、顧客が Java アプリケーションをデプロイするためのプラットフォームをクラウドインフラストラクチャシステムにおいて提供してもよい。

【0092】

さまざまな異なるインフラストラクチャサービスが IaaS プラットフォームによってクラウドインフラストラクチャシステムにおいて提供されてもよい。インフラストラクチャサービスは、ストレージ、ネットワーク、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に対する他の基礎的計算リソースなどの基本的な計算リソースの管理および制御を容易にする。

【0093】

特定の実施形態では、クラウドインフラストラクチャシステム 702 は、クラウドインフラストラクチャシステムの顧客に対してさまざまなサービスを提供するよう利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 730 も含んでもよい。一実施形態では、インフラストラクチャリソース 730 は、サーバ、ストレージ、ならびに 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ネットワーキングリソースなどの、ハードウェアの予め統合され最適化された組み合わせを含んでもよい。

【0094】

いくつかの実施形態では、クラウドインフラストラクチャシステム 702 におけるリソ

10

20

30

40

50

ースは、複数のユーザによって共有され、要求につき動的に再割り当てされてもよい。また、リソースは、ユーザに対してさまざまな時間ゾーンで割り当てられてもよい。たとえば、クラウドインフラストラクチャシステム 730 は、第 1 の時間ゾーンにおけるユーザの第 1 の組がクラウドインフラストラクチャシステムのリソースをある特定の時間の間利用することを可能にし、次いで、異なる時間ゾーンに位置するユーザの別の組に対する同じリソースの再割り当てを可能にし、それによって、リソースの利用を最大化してもよい。

#### 【0095】

特定の実施形態では、クラウドインフラストラクチャシステム 702 のさまざまなコンポーネントまたはモジュールによって共有されてクラウドインフラストラクチャシステム 702 によって提供されるサービスによって共有されるある数の内部共有サービス 732 が提供されてもよい。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、統合サービス、エンタープライズリポジトリサービス、エンタープライズ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、電子メールサービス、通知サービス、ファイル転送サービスなどを含んでもよいが、それらに限定されるものではない。

#### 【0096】

特定の実施形態では、クラウドインフラストラクチャシステム 702 は、クラウドインフラストラクチャシステムにおいてクラウドサービス（たとえば、SaaS、PaaS、および IaaS サービス）の包括的な管理を提供してもよい。一実施形態では、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 702 によって受信される顧客のサブスクリプションをプロビジョニングし、管理し、および追跡する機能などを含んでもよい。

#### 【0097】

一実施形態では、図に示されるように、クラウド管理機能は、オーダー管理モジュール 720、オーダーオーケストレーションモジュール 722、オーダープロビジョニングモジュール 724、オーダー管理および監視モジュール 726、ならびにアイデンティティ管理モジュール 728 などの 1 つ以上のモジュールによって提供されてもよい。これらのモジュールは、1 つ以上のコンピュータおよび / もしくはサーバを含んでもよく、またはそれらを用いて提供されてもよく、それらは、汎用コンピュータ、専用の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ってもよい。

#### 【0098】

例示的な動作 734 では、クライアントデバイス 704、706 または 708 などのクライアントデバイスを用いる顧客は、クラウドインフラストラクチャシステム 702 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 702 によって提供される 1 つ以上のサービスに対するサブスクリプションに対するオーダーを行なうことによって、クラウドインフラストラクチャシステム 702 と対話してもよい。特定の実施形態では、顧客は、クラウド UI 712、クラウド UI 714 および / またはクラウド UI 716 などのクラウドユーザインターフェイス (UI) にアクセスし、サブスクリプションオーダーをこれらの UI を介して行なってもよい。顧客がオーダーを行なう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 702 によって受信されるオーダー情報は、顧客を識別する情報、およびクラウドインフラストラクチャシステム 702 によって提供される、その顧客が契約する予定の 1 つ以上のサービスを含んでもよい。

#### 【0099】

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 712、714 および / または 716 を介して受信される。

#### 【0100】



動作 7 3 6 において、オーダーは、オーダーデータベース 7 1 8 に保存される。オーダーデータベース 7 1 8 は、クラウドインフラストラクチャシステム 7 0 2 によって動作されるいくつかのデータベースのうちの 1 つであり得、他のシステム要素と連携して動作され得る。

【 0 1 0 1 】

動作 7 3 8 において、オーダー情報は、オーダー管理モジュール 7 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 7 2 0 は、オーダーを検証すること、および検証次第そのオーダーを予約することなど、オーダーに関連する請求および課金機能を実行するように構成されてもよい。

【 0 1 0 2 】

動作 7 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 7 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 7 2 2 は、オーダー情報を利用して、顧客によってなされたオーダーに対して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ションし得る。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 7 2 2 は、オーダープロビジョニングモジュール 7 2 4 のサービスを使用してサブスクライブされたサービスをサポートするようにリソースのプロビジョニングをオーケストレーションしてもよい。

【 0 1 0 3 】

特定の実施形態では、オーダーオーケストレーションモジュール 7 2 2 は、各オーダーに関連付けられた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 7 4 2 において、新たなサブスクリプションに対する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 7 2 2 は、リソースを割り当てて、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リソースを構成するよう、オーダープロビジョニングモジュール 7 2 4 に対して要求を送信する。オーダープロビジョニングモジュール 7 2 4 は、顧客によってオーダーされたサービスに対するリソースの割り当てを可能にする。オーダープロビジョニングモジュール 7 2 4 は、クラウドインフラストラクチャシステム 7 0 2 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よう利用される物理的インプリメンテーション層との間にある抽象化レベルを提供する。これにより、サービスおよびリソースが実際にオンザフライでプロビジョニングされるか、サービスおよびリソースが予めプロビジョニングされて要求時にのみ割り当てられる / あてがわれるかなどの、インプリメンテーション詳細からオーダーオーケストレーションモジュール 7 2 2 を分離することができる。

【 0 1 0 4 】

動作 7 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 7 0 2 のオーダープロビジョニングモジュール 7 2 4 によってクライアントデバイス 7 0 4 , 7 0 6 および / または 7 0 8 上の顧客に送信されてもよい。

【 0 1 0 5 】

動作 7 4 6 において、顧客のサブスクリプションオーダーは、オーダー管理および監視モジュール 7 2 6 によって管理および追跡されてもよい。いくつかの例では、オーダー管理および監視モジュール 7 2 6 は、使用されるストレージの量、転送されるデータの量、ユーザの人数、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の量などの、サブスクリプションオーダーにおけるサービスの使用統計を収集するように構成されてもよい。

【 0 1 0 6 】

特定の実施形態では、クラウドインフラストラクチャシステム 7 0 0 は、アイデンティティ管理モジュール 7 2 8 を含んでもよい。アイデンティティ管理モジュール 7 2 8 は、クラウドインフラストラクチャシステム 7 0 0 におけるアクセス管理および承認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てもよい。いくつかの実施形

10

20

30

40

50

態では、アイデンティティ管理モジュール728は、クラウドインフラストラクチャシステム702によって提供されるサービスを利用することを望む顧客についての情報を制御してもよい。そのような情報は、そ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が承認されるかを記述する情報を含み得る。アイデンティティ管理モジュール728は、各顧客についての記述的情報ならびにどのように誰によってその記述的情報がアクセスおよび修正され得るかについての情報の管理も含んでもよい。

#### 【0107】

図8は、本開示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800を示す。システム800は、上記のコンピュータシステムのうちのいずれかを実現するよう用いられてもよい。図に示されるように、コンピュータシステム800は、多数の周辺サブシステムとバスサブシステム802を介して通信する処理ユニット804を含む。これらの周辺サブシステムは、処理加速ユニット806、I/Oサブシステム808、ストレージサブシステム818および通信サブシステム824を含んでもよい。ストレージサブシステム818は、有形のコンピュータ読取可能な記憶媒体822およびシステムメモリ810を含む。

#### 【0108】

バスサブシステム802は、コンピュータシステム800のさまざまなコンポーネントおよびサブシステムに意図される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802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实施例は、複数のバスを利用してもよい。バスサブシステム802は、さまざまなバスアーキテクチャのうちのいずれかを用いるメモリバスまたはメモリコントローラ、周辺バスおよび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のいずれかであってもよい。たとえば、そのようなアーキテクチャは、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、エンハンスド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IEEE P1386.1規格に従って製造される中二階バスとして実現され得る周辺コンポーネントインターコネクト（Peripheral Component Interconnect：PCI）バスを含んでもよい。

#### 【0109】

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804は、コンピュータシステム800の動作を制御する。1つ以上のプロセッサが処理ユニット804に含まれてもよい。これらのプロセッサは、シングルコアプロセッサを含んでもよく、またはマルチコアプロセッサを含んでもよい。特定の实施形態では、処理ユニット804は、シングルコアまたはマルチコアプロセッサが各処理ユニットに含まれる1つ以上の独立した処理ユニット832および/または834として実現されてもよい。他の实施形態では、処理ユニット804は、2つのデュアルコアプロセッサを単一のチップに統合す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。

#### 【0110】

さまざまな実施形態では、処理ユニット804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することができ、複数の同時に実行されるプログラムまたはプロセスを維持することができる。任意の所与の時点で、実行されるべきプログラムコードの一部または全ては、プロセッサ804、および/または、ストレージサブシステム818に常駐することができる。好適なプログラミングを介して、プロセッサ804は、上記のさまざまな機能を提供することができる。コンピュータシステム800は、デジタル信号プロセッサ（digital signal processor：DSP）、特殊目的プロセッサなどを含み得る処理加速ユニット806をさらに含んでもよい。

10

20

30

40

50

## 【 0 1 1 1 】

I / Oサブシステム 8 0 8 は、ユーザインターフェイス入力デバイスおよびユーザインターフェイス出力デバイスを含んでもよい。ユーザインターフェイス入力デバイスは、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デバイス、ディスプレイに組み込まれたタッチパッドまたはタッチスクリーン、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、音声コマンド認識システムを伴う音声入力デバイス、マイクロフォン、および他のタイプの入力デバイスを含んでもよい。ユーザインターフェイス入力デバイスは、たとえば、ジェスチャおよび話し言葉コマンドを用いて、ナチュラルユーザインターフェイスを介して、Microsoft X b o x（登録商標）3 6 0 ゲームコントローラなどの入力デバイスをユーザが制御して対話することを可能にするMicrosoft K i n e c t（登録商標）モーションセンサなどのモーション感知および／またはジェスチャ認識デバイスを含んでもよい。ユーザインターフェイス入力デバイスは、ユーザから目の動き（たとえば、写真を撮っている間および／またはメニュー選択を行なっている間の「まばたき」）を検出し、アイジェスチャを入力デバイス（たとえばGoogle G l a s s（登録商標））への入力として変換するGoogle G l a s s（登録商標）瞬き検出器などのアイジェスチャ認識デバイスも含んでもよい。また、ユーザインターフェイス入力デバイスは、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S i r i（登録商標）ナビゲータ）と対話することを可能にする音声認識感知デバイスを含んでもよい。

10

## 【 0 1 1 2 】

ユーザインターフェイス入力デバイスは、三次元（3 D）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、ならびにスピーカ、デジタルカメラ、デジタルカムコーダ、ポータブルメディアプレーヤ、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3 Dスキャナ、3 Dプリンタ、レーザレンジファインダ、および視線追跡デバイスなどの聴覚／視覚デバイスも含んでもよいが、そ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デバイスは、たとえば、コンピュータ断層撮影、磁気共鳴撮像、ポジションエミッショントモグラフィー、医療用超音波検査デバイスなどの医療用画像化入力デバイスを含んでもよい。ユーザインターフェイス入力デバイスは、たとえば、M I D Iキーボード、デジタル楽器などの音声入力デバイスも含んでもよい。

20

30

## 【 0 1 1 3 】

ユーザインターフェイス出力デバイスは、ディスプレイサブシステム、インジケータライト、または音声出力デバイスなどの非ビジュアルディスプレイなどを含んでもよい。ディスプレイサブシステムは、陰極線管（cathode ray tube：C R T）、液晶ディスプレイ（liquid crystal display：L C D）またはプラズマディスプレイを使うものなどのフラットパネルデバイス、投影デバイス、タッチスクリーンなどであってもよい。一般に、「出力デバイス」という語の使用は、コンピュータシステム 8 0 0 からユーザまたは他のコンピュータに情報を出力するための全ての考えられ得るタイプのデバイスおよび機構を含むよう意図される。たとえば、ユーザインターフェイス出力デバイスは、モニタ、プリンタ、スピーカ、ヘッドフォン、自動車ナビゲーションシステム、プロッタ、音声出力デバイスおよびモデムなどの、テキスト、グラフィックスおよび音声／映像情報を視覚的に伝えるさまざまな表示デバイスを含んでもよいが、それらに限定されるものではない。

40

## 【 0 1 1 4 】

コンピュータシステム 8 0 0 は、現在のところシステムメモリ 8 1 0 内に位置しているものとして示されているソフトウェア要素を備えるストレージサブシステム 8 1 8 を備えてもよい。システムメモリ 8 1 0 は、処理ユニット 8 0 4 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てもよい。

## 【 0 1 1 5 】

コンピュータシステム 8 0 0 の構成およびタイプによって、システムメモリ 8 1 0 は、

50

揮発性であってもよく（ランダムアクセスメモリ（RAM）など）、および／または、不揮発性であってもよい（リードオンリメモリ（ROM）、フラッシュメモリなど）。RAMは、一般に、処理ユニット804にすぐにアクセス可能であり、および／または、処理ユニット804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含む。いくつかの実現例では、システムメモリ810は、スタティックランダムアクセスメモリ（SRAM）またはダイナミックランダムアクセスメモリ（DRAM）などの複数の異なるタイプのメモリを含んでもよい。いくつかの実現例では、起動中などにコンピュータシステム800内の要素間における情報の転送を助ける基本的なルーティンを含むベーシックインプット／アウトプットシステム（basic input/output system : BIOS）は、一般に、ROMに格納されてもよい。一例として、限定を伴うことなく、システムメモリ810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）などを含んでもよいアプリケーションプログラム812、プログラムデータ814およびオペレーティングシステム816も示す。一例として、オペレーティングシステム816は、Microsoft Windows（登録商標）、Apple Macintosh（登録商標）および／もしくはLinuxオペレーティングシステム、さまざまな市場で入手可能なUNIX（登録商標）またはUNIXのようなオペレーティングシステム（さまざまなGNU/Linuxオペレーティングシステム、Google Chrome（登録商標）OSなどを含むがそれらに限定されない）、ならびに／または、iOS、Windows（登録商標）Phone、Android（登録商標）OS、BlackBerry（登録商標）OS、およびPalm（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のさまざまなバージョンを含んでもよい。

#### 【0116】

ストレージサブシステム818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も提供してもよい。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）は、ストレージサブシステム818に格納されてもよい。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804によって実行されてもよい。ストレージサブシステム818は、本開示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も提供してもよい。

#### 【0117】

ストレージサブシステム818は、コンピュータ読取可能な記憶媒体822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820も含んでもよい。システムメモリ810とともに、およびオプションとしてシステムメモリ810との組み合わせで、コンピュータ読取可能な記憶媒体822は、コンピュータ読取可能な情報を一時的および／またはより永久的に収容、格納、伝送および検索取得するための、遠隔の、ローカルな、固定された、および／またはリムーバブルなストレージデバイスに記憶媒体を加えたものを包括的に表わしてもよい。

#### 【0118】

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822は、記憶媒体および通信媒体を含む、当該技術分野において公知であるまたは使用されるいずれかの適切な媒体も含んでもよく、当該媒体は、情報の格納および／または伝送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、リムーバブルおよび非リムーバブルな媒体などであるが、それらに限定されるものではない。これは、RAM、ROM、電氣的に消去可能なプログラム可能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（DVD）、または他の光学式ストレージ、磁気カセット、磁気テープ、磁気ディスクストレージもしくは他の磁気ストレージデバイス、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非一時的なコンピュータ読取可能な記憶媒体を含ん

でもよい。指定される場合には、これは、データ信号、データ伝送、または所望の情報を伝送するために使用可能でありコンピューティングシステム 800 によってアクセス可能であるその他の媒体などの無形の一時的なコンピュータ読取可能な媒体も含んでもよい。

#### 【0119】

一例として、コンピュータ読取可能な記憶媒体 822 は、非リムーバブル不揮発性磁気媒体に対して読み書きするハードディスクドライブ、リムーバブル不揮発性磁気ディスクに対して読み書きする磁気ディスクドライブ、CD ROM、DVD およびブルーレイ（登録商標）ディスクなどの、リムーバブル不揮発性光ディスクに対して読み書きする光ディスクドライブ、または他の光学式媒体を含んでもよい。コンピュータ読取可能な記憶媒体 822 は、Zip（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んでもよいが、それらに限定されるものではない。コンピュータ読取可能な記憶媒体 822 は、フラッシュメモリベースのSSD、エンタープライズ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との組み合わせを使用するハイブリッドSSDも含んでもよい。ディスクドライブおよびそれらの関連付けられた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の揮発性ストレージをコンピュータシステム 800 に提供してもよい。

#### 【0120】

通信サブシステム 824 は、他のコンピュータシステムおよびネットワークに対するインターフェイスを提供する。通信サブシステム 824 は、他のシステムとコンピュータシステム 800 との間のデータの送受のためのインターフェイスとして働く。たとえば、通信サブシステム 824 は、コンピュータシステム 800 がインターネットを介して1つ以上のデバイスに接続することを可能にしてもよい。いくつかの実施形態では、通信サブシステム 824 は、（たとえば、セルラー電話技術、3G、4GもしくはEDGE（グローバル進化のための高速データレート）などの先進データネットワーク技術、WiFi（IEEE 802.11ファミリー規格、もしくは他のモバイル通信技術、またはそれらのいずれかの組み合わせを用いて）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）送受信機コンポーネント、グローバルポジショニングシステム（global positioning system：GPS）受信機コンポーネント、ならびに/または他のコンポーネントを含んでもよい。いくつかの実施形態では、通信サブシステム 824 は、無線インターフェイスに加えて、またはその代わりに、有線ネットワーク接続（たとえば、イーサネット）を提供することができる。

#### 【0121】

また、いくつかの実施形態では、通信サブシステム 824 は、コンピュータシステム 800 を使用し得る1人以上のユーザの代わりに、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826、イベントストリーム828、イベント更新情報830などの形式で入力通信を受信してもよい。

#### 【0122】

一例として、通信サブシステム 824 は、ソーシャルメディアネットワークおよび/またはTwitter（登録商標）フィード、Facebook（登録商標）更新情報、Rich Site Summary（RSS）フィードなどのウェブフィード、および/もしくは1つ以上の第三者情報源からのリアルタイム更新情報などの他の通信サービスのユーザからリアルタイムでデータフィード826を受信するように構成されてもよい。

#### 【0123】

さらに、また、通信サブシステム 824 は、連続データストリームの形式でデータを受

信するように構成されてもよく、当該連続データストリームは、明確な終端を持たない、本来は連続的または無限であり得るリアルタイムイベントのイベントストリーム 8 2 8 および / または イベント更新情報 8 3 0 を含んでもよい。連続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、金融株式相場表示板、ネットワーク性能測定ツール（たとえば、ネットワーク監視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、自動車交通監視などを挙げることができる。

【 0 1 2 4 】

また、通信サブシステム 8 2 4 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 8 2 6、イベントストリーム 8 2 8、イベント更新情報 8 3 0などを、コンピュータシステム 8 0 0 に結合される 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構成されてもよい。

10

【 0 1 2 5 】

コンピュータシステム 8 0 0 は、手持ち式の携帯デバイス（たとえば、i P h o n e（登録商標）携帯電話、i P a d（登録商標）コンピューティングタブレット、P D A）、ウェアラブルデバイス（たとえば、G o o g l e G l a s s（登録商標）頭部装着型ディスプレイ）、P C、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もののうちの 1 つであり得る。

【 0 1 2 6 】

20

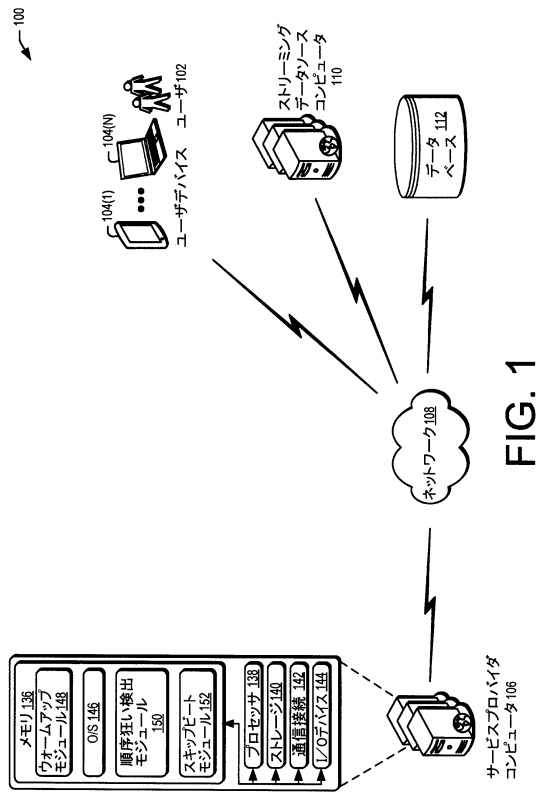
常に変化するコンピュータおよびネットワークの性質のため、図に示されるコンピュータシステム 8 0 0 の記載は、単に具体的な例として意図される。図に示されるシステムよりも多くのコンポーネントまたは少ないコンポーネントを有する多くの他の構成が可能である。たとえば、カスタマイズされたハードウェアも使用されてもよく、および / または、特定の要素が、ハードウェア、ファームウェア、ソフトウェア（タブレットを含む）、または組み合わせで実現されてもよい。さらに、ネットワーク入力 / 出力デバイスなどの他のコンピューティングデバイスへの接続が利用されてもよい。本明細書における開示および教示に基づいて、当業者は、さまざまな実施形態を実現するための他の態様および / または方法を理解するであろう。

【 0 1 2 7 】

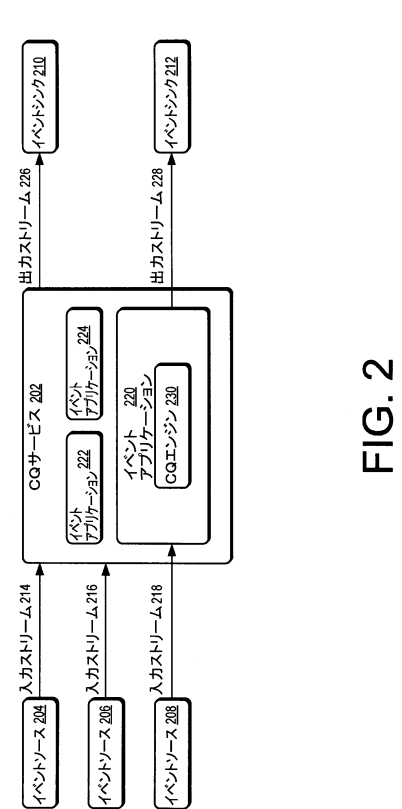
30

上記の明細書では、本開示の局面についてその具体的な実施形態を参照して説明しているが、本開示はそれに限定されるものではないということを当業者は認識するであろう。上記の開示のさまざまな特徴および局面は、個々にまたは一緒に用いられてもよい。また、さまざまな変形例、変更例、代替的な構造および等価物は、本開示の範囲内に包含される。変形例は、開示されている特徴のいずれかの関連の組み合わせを含む。さらに、実施形態は、明細書のさらに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超えて、さまざまな環境およびアプリケーションで利用することができる。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と見なされるべきである。

【図 1】



【図 2】



【図 3】

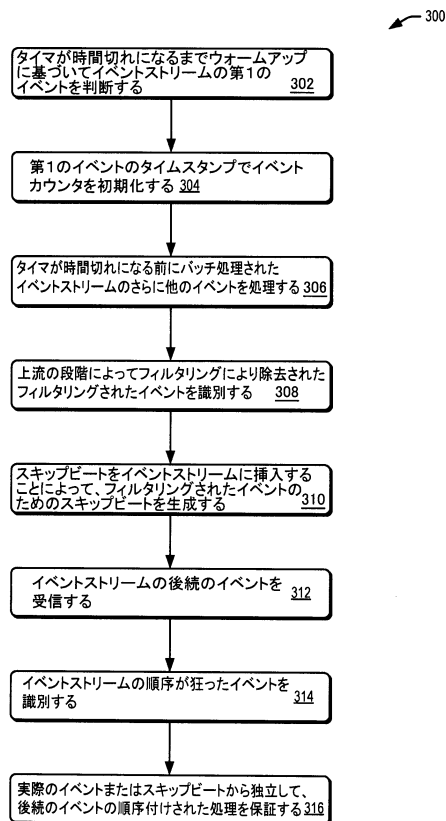


FIG. 3

【図 4】

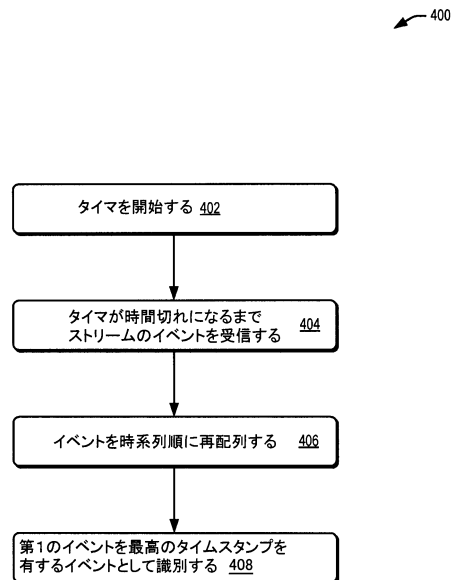


FIG. 4

【図 5】

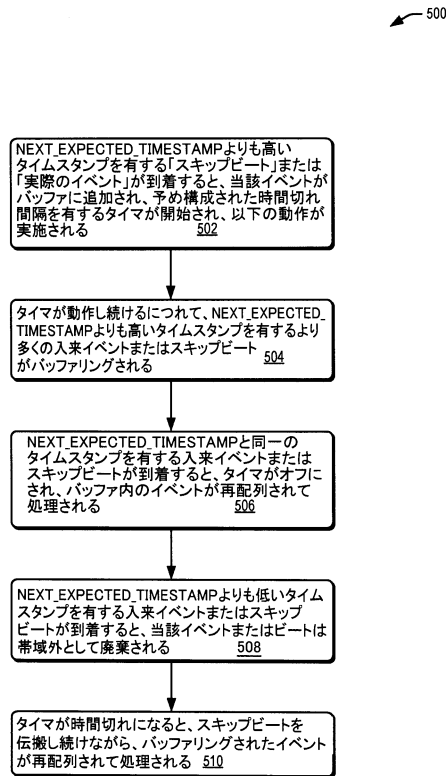


FIG. 5

【図 7】

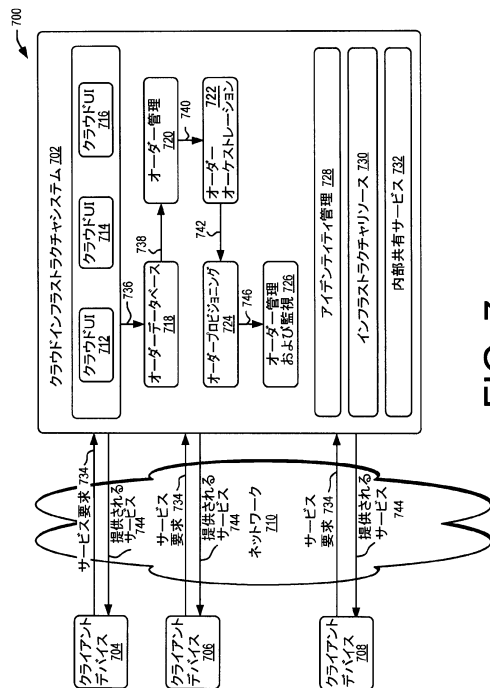


FIG. 7

【図 6】

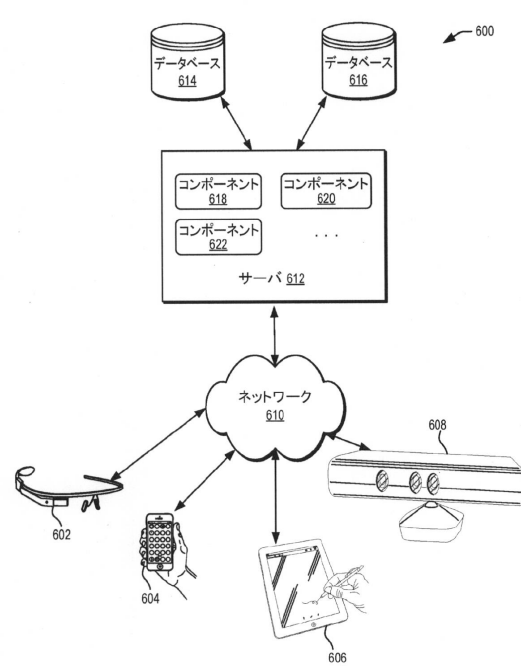


FIG. 6

【図 8】

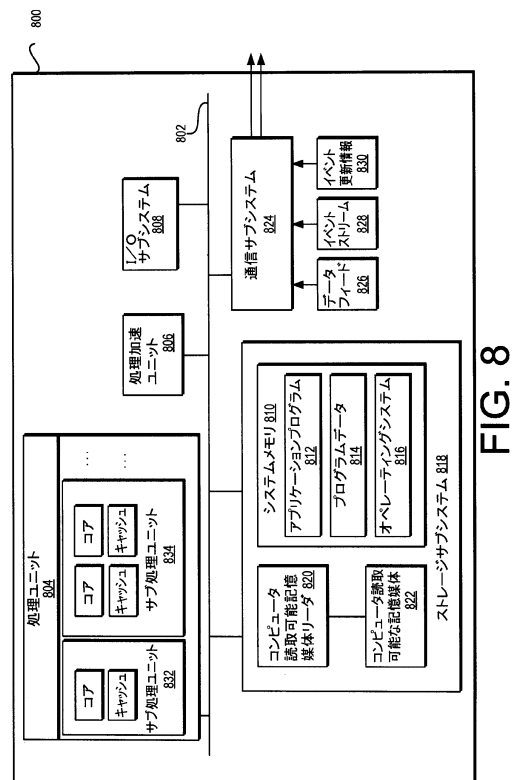


FIG. 8



---

フロントページの続き

- (72)発明者 ビシュノイ, サンディーブ  
アメリカ合衆国、9 4 0 0 2 カリフォルニア州、ベルモント、クレスト・ビュー・アベニュー、5  
1 3、ユニット・2 3 5
- (72)発明者 パーク, ホヨン  
アメリカ合衆国、9 5 1 2 9 カリフォルニア州、サン・ノゼ、オスナ・プレイス、1 1 9 6
- (72)発明者 デシュムク, ウンメシュ・アニル  
インド、4 4 0 0 2 5 マハーラーシュトラ、ナーグプル、ソネガアン、エイチ・ビィ・エステー  
ト・ニアー・シブ・マンディール、プラット・ナンバー・2 2 2 - エイ

審査官 甲斐 哲雄

- (56)参考文献 特開 2 0 1 4 - 1 9 9 5 8 1 ( J P , A )  
BRITO, Andrey et al. , Speculative Out-Of-Order Event Processing with Software Transact  
ion Memory , Proceedings of the second international conference on Distributed event-ba  
sed systems , 米国 , Association for Computing Machinery, New York , 2 0 0 8 年 7 月 , DE  
BS '08 , pp.265-275 , <https://doi.org/10.1145/1385989.1386023>

- (58)調査した分野(Int.Cl. , D B 名)  
G 0 6 F 1 6 / 0 0 - 1 6 / 9 5 8