

(19) 日本国特許庁 (JP)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2016-513300

(P2016-513300A)

(43) 公表日 平成28年5月12日 (2016.5.12)

(51) Int.Cl.	F I	テーマコード (参考)
GO 6 F 17/30 (2006.01)	GO 6 F 17/30 4 1 5	5 B O 6 1
GO 6 F 13/36 (2006.01)	GO 6 F 13/36 3 1 O E	

審査請求 未請求 予備審査請求 有 (全 53 頁)

(21) 出願番号	特願2015-552781 (P2015-552781)	(71) 出願人	502303739
(86) (22) 出願日	平成26年1月9日 (2014.1.9)		オラクル・インターナショナル・コーポレ イション
(85) 翻訳文提出日	平成27年9月4日 (2015.9.4)		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
(86) 国際出願番号	PCT/US2014/010920		
(87) 国際公開番号	W02014/113273	(74) 代理人	110001195
(87) 国際公開日	平成26年7月24日 (2014.7.24)		特許業務法人深見特許事務所
(31) 優先権主張番号	61/752, 850	(72) 発明者	ビシュノイ, サンディープ
(32) 優先日	平成25年1月15日 (2013.1.15)		アメリカ合衆国、94065 カリフォル ニア州、レッドウッド・シティー、オラク ル・パークウェイ、500
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	13/838, 259		
(32) 優先日	平成25年3月15日 (2013.3.15)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 連続データストリームに対する継続時間可変ウィンドウ

(57) 【要約】

データのストリームを処理するための、改善された技術が提供される。1つのイベントが時間ベースのウィンドウ内に留まる資格がある継続時間は、同一のイベントストリームを介して受信した異なるイベントごとに異なり得る。ある実施形態において、1つの入力イベントが時間ベースのウィンドウ内にある継続時間は、このイベントの1つ以上の属性の1つ以上の値の関数である。異なるイベントは異なる1つ以上の属性値を有することができるので、異なるイベントがそのイベントストリームのための時間ベースのウィンドウ内にある時間の量は異なることがある。1つのイベントが時間ベースのウィンドウ内にある時間の量は、このイベントの1つ以上の属性によって制御することができる。

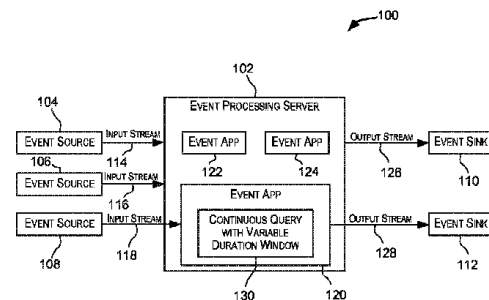


FIG. 1

【特許請求の範囲】**【請求項 1】**

方法であって、

イベントストリームを介して受信した第 1 のイベントの第 1 の継続時間を計算装置によって決定することを含み、前記第 1 の継続時間は、前記イベントストリームに対して指定された時間ベースのウィンドウ内に前記第 1 のイベントが留まる資格がある継続時間を示し、

前記イベントストリームを介して受信した第 2 のイベントの第 2 の継続時間を前記計算装置によって決定することを含み、前記第 2 の継続時間は、前記時間ベースのウィンドウ内に前記第 2 のイベントが留まる資格がある継続時間を示し、前記第 2 の継続時間は前記第 1 の継続時間と異なる、方法。

10

【請求項 2】

前記第 1 の時間イベントが前記時間ベースのウィンドウ内に前記第 1 の継続時間留まった後に、前記第 1 のイベントを前記時間ベースのウィンドウから前記計算装置によって除去することと、

前記第 2 の時間イベントが前記時間ベースのウィンドウ内に前記第 2 の継続時間留まった後に、前記第 2 のイベントを前記時間ベースのウィンドウから前記計算装置によって除去することとをさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記第 1 のイベントは関連付けられた第 1 の時間を有し、

前記第 2 のイベントは関連付けられた第 2 の時間を有し、

前記第 1 の時間は前記第 2 の時間と同一である、請求項 1 または 2 に記載の方法。

20

【請求項 4】

前記第 1 のイベントは関連付けられた第 1 の時間を有し、

前記第 2 のイベントは関連付けられた第 2 の時間を有し、

前記第 1 の時間は前記第 2 の時間と異なる、請求項 1 または 2 に記載の方法。

【請求項 5】

前記第 1 の継続時間を決定することは、前記第 1 のイベントの 1 つ以上の属性の 1 つ以上の値に基づいて前記第 1 の継続時間を計算することを含み、

前記第 2 の継続時間を決定することは、前記第 2 のイベントの 1 つ以上の属性の 1 つ以上の値に基づいて前記第 2 の継続時間を計算することを含む、請求項 1 ~ 4 のいずれか一項に記載の方法。

30

【請求項 6】

前記第 1 のイベントの第 1 の属性の値に基づいて前記第 1 のイベントのための第 1 の区画を決定することと、

前記第 2 のイベントの第 1 の属性の値に基づいて前記第 2 のイベントが前記第 1 の区画に属すると決定することとをさらに含む、

前記時間ベースのウィンドウは前記第 1 の区画のためのものである、請求項 1 ~ 5 のいずれか一項に記載の方法。

【請求項 7】

前記第 1 のイベントに関連付けられた第 1 の時間と前記第 1 の継続時間とに基づいて前記第 1 のイベントの第 1 の満了時間を前記計算装置によって決定することと、

前記第 2 のイベントに関連付けられた第 2 の時間と前記第 2 の継続時間とに基づいて前記第 2 のイベントの第 2 の満了時間を前記計算装置によって決定することと、

前記第 1 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 1 のイベントを除去することと、

前記第 2 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 2 のイベントを除去することとをさらに含む、請求項 1 ~ 6 のいずれか一項に記載の方法。

40

【請求項 8】

計算装置であって、

50

メモリと、

一組の処理部とを備え、前記一組の処理部のうちの１つ以上の処理部は、

イベントストリームを介して受信した第１のイベントの第１の継続時間を決定するように適合され、前記第１の継続時間は、前記イベントストリームに対して指定された時間ベースのウィンドウ内に前記第１のイベントが留まる資格がある継続時間を示し、

前記イベントストリームを介して受信した第２のイベントの第２の継続時間を決定するように適合され、前記第２の継続時間は、前記時間ベースのウィンドウ内に前記第２のイベントが留まる資格がある継続時間を示し、前記第２の継続時間は前記第１の継続時間と異なる、計算装置。

【請求項 ９】

10

前記一組の処理部のうちの１つ以上の処理部は、

前記第１の時間イベントが前記時間ベースのウィンドウ内に前記第１の継続時間留まった後に、前記第１のイベントを前記時間ベースのウィンドウから除去するように適合され、

前記第２の時間イベントが前記時間ベースのウィンドウ内に前記第２の継続時間留まった後に、前記第２のイベントを前記時間ベースのウィンドウから除去するように適合される、請求項 ８ に記載の計算装置。

【請求項 １０】

前記第１のイベントは関連付けられた第１の時間を有し、

前記第２のイベントは関連付けられた第２の時間を有し、

20

前記第１の時間は前記第２の時間と同一である、請求項 ８ または ９ に記載の計算装置。

【請求項 １１】

前記第１のイベントは関連付けられた第１の時間を有し、

前記第２のイベントは関連付けられた第２の時間を有し、

前記第１の時間は前記第２の時間と異なる、請求項 ８ または ９ に記載の計算装置。

【請求項 １２】

前記一組の処理部のうちの１つ以上の処理部は、

前記第１のイベントの１つ以上の属性の１つ以上の値に基づいて前記第１の継続時間を決定するように適合され、

前記第２のイベントの１つ以上の属性の１つ以上の値に基づいて前記第２の継続時間を決定するように適合される、請求項 ８ ～ １１ のいずれか一項に記載の計算装置。

30

【請求項 １３】

前記一組の処理部のうちの１つ以上の処理部は、

前記第１のイベントの第１の属性の値に基づいて前記第１のイベントのための第１の区画を決定するように適合され、

前記第２のイベントの第１の属性の値に基づいて前記第２のイベントが前記第１の区画に属すると決定するように適合され、

前記時間ベースのウィンドウは前記第１の区画のためのものである、請求項 ８ ～ １２ のいずれか一項に記載の計算装置。

【請求項 １４】

40

前記一組の処理部のうちの１つ以上の処理部は、

前記第１のイベントに関連付けられた第１の時間と前記第１の継続時間とに基づいて前記第１のイベントの第１の満了時間を決定するように適合され、

前記第２のイベントに関連付けられた第２の時間と前記第２の継続時間とに基づいて前記第２のイベントの第２の満了時間を決定するように適合され、

前記第１の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第１のイベントを除去するように適合され、

前記第２の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第２のイベントを除去するように適合される、請求項 ８ ～ １３ のいずれか一項に記載の計算装置。

50

【請求項 15】

1つ以上の処理部によって実行可能な複数の命令を格納するコンピュータ読取可能なメモリであって、前記複数の命令は、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、イベントストリームを介して受信した第1のイベントの第1の継続時間を決定させる命令を含み、前記第1の継続時間は、前記イベントストリームに対して指定された時間ベースのウィンドウ内に前記第1のイベントが留まる資格がある継続時間を示し、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、前記イベントストリームを介して受信した第2のイベントの第2の継続時間を決定させる命令を含み、前記第2の継続時間は、前記時間ベースのウィンドウ内に前記第2のイベントが留まる資格がある継続時間を示し、前記第2の継続時間は前記第1の継続時間と異なる、コンピュータ読取可能なメモリ。

10

【請求項 16】

前記複数の命令は、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、前記第1の時間イベントが前記時間ベースのウィンドウ内に前記第1の継続時間留まった後に、前記第1のイベントを前記時間ベースのウィンドウから除去させる命令と、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、前記第2の時間イベントが前記時間ベースのウィンドウ内に前記第2の継続時間留まった後に、前記第2のイベントを前記時間ベースのウィンドウから除去させる命令とを含む、請求項15に記載のコンピュータ読取可能なメモリ。

20

【請求項 17】

前記第1のイベントは関連付けられた第1の時間を有し、

前記第2のイベントは関連付けられた第2の時間を有し、

前記第1の時間は前記第2の時間と同一である、請求項15または16に記載のコンピュータ読取可能なメモリ。

【請求項 18】

前記第1のイベントは関連付けられた第1の時間を有し、

前記第2のイベントは関連付けられた第2の時間を有し、

前記第1の時間は前記第2の時間と異なる、請求項15または16に記載のコンピュータ読取可能なメモリ。

30

【請求項 19】

前記複数の命令は、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、前記第1のイベントの1つ以上の属性の1つ以上の値に基づいて前記第1の継続時間を決定させる命令と、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、前記第2のイベントの1つ以上の属性の1つ以上の値に基づいて前記第2の継続時間を決定させる命令とを含む、請求項15～18のいずれか一項に記載のコンピュータ読取可能なメモリ。

【請求項 20】

前記複数の命令は、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、前記第1のイベントの第1の属性の値に基づいて前記第1のイベントのための第1の区画を決定させる命令と、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、前記第2のイベントの第1の属性の値に基づいて前記第2のイベントが前記第1の区画に属すると決定させる命令とを含み、

前記時間ベースのウィンドウは前記第1の区画のためのものである、請求項15～19のいずれか一項に記載のコンピュータ読取可能なメモリ。

40

【請求項 21】

前記複数の命令は、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、前記第1のイベントに関連

50

付けられた第 1 の時間と前記第 1 の継続時間とに基づいて前記第 1 のイベントの第 1 の満了時間を決定させる命令と、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 2 のイベントに関連付けられた第 2 の時間と前記第 2 の継続時間とに基づいて前記第 2 のイベントの第 2 の満了時間を決定させる命令と、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 1 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 1 のイベントを除去させる命令と、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 2 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 2 のイベントを除去させる命令を含む、請求項 15 ~ 20 のいずれか一項に記載のコンピュータ読取可能なメモリ。

【請求項 22】

システムであって、

イベントストリームを介して受信した第 1 のイベントの第 1 の継続時間を決定するための手段を備え、前記第 1 の継続時間は、前記イベントストリームに対して指定された時間ベースのウィンドウ内に前記第 1 のイベントが留まる資格がある継続時間を示し、

前記イベントストリームを介して受信した第 2 のイベントの第 2 の継続時間を決定するための手段を備え、前記第 2 の継続時間は、前記時間ベースのウィンドウ内に前記第 2 のイベントが留まる資格がある継続時間を示し、前記第 2 の継続時間は前記第 1 の継続時間と異なる、システム。

【請求項 23】

前記第 1 の時間イベントが前記時間ベースのウィンドウ内に前記第 1 の継続時間留まった後に、前記第 1 のイベントを前記時間ベースのウィンドウから除去するための手段と、

前記第 2 の時間イベントが前記時間ベースのウィンドウ内に前記第 2 の継続時間留まった後に、前記第 2 のイベントを前記時間ベースのウィンドウから除去するための手段とをさらに備える、請求項 22 に記載のシステム。

【請求項 24】

前記第 1 のイベントは関連付けられた第 1 の時間を有し、

前記第 2 のイベントは関連付けられた第 2 の時間を有し、

前記第 1 の時間は前記第 2 の時間と同一である、請求項 22 または 23 に記載のシステム。

【請求項 25】

前記第 1 のイベントは関連付けられた第 1 の時間を有し、

前記第 2 のイベントは関連付けられた第 2 の時間を有し、

前記第 1 の時間は前記第 2 の時間と異なる、請求項 22 または 23 に記載のシステム。

【請求項 26】

前記第 1 の継続時間を決定するための手段は、前記第 1 のイベントの 1 つ以上の属性の 1 つ以上の値に基づいて前記第 1 の継続時間を計算するための手段を含み、

前記第 2 の継続時間を決定するための手段は、前記第 2 のイベントの 1 つ以上の属性の 1 つ以上の値に基づいて前記第 2 の継続時間を計算するための手段を含む、請求項 22 ~ 25 のいずれか一項に記載のシステム。

【請求項 27】

前記第 1 のイベントの第 1 の属性の値に基づいて前記第 1 のイベントのための第 1 の区画を決定するための手段と、

前記第 2 のイベントの第 1 の属性の値に基づいて前記第 2 のイベントが前記第 1 の区画に属すると決定するための手段とをさらに備え、

前記時間ベースのウィンドウは前記第 1 の区画のためのものである、請求項 22 ~ 26 のいずれか一項に記載のシステム。

【請求項 28】

前記第 1 のイベントに関連付けられた第 1 の時間と前記第 1 の継続時間とに基づいて前

10

20

30

40

50

記第 1 のイベントの第 1 の満了時間を決定するための手段と、

前記第 2 のイベントに関連付けられた第 2 の時間と前記第 2 の継続時間とに基づいて前記第 2 のイベントの第 2 の満了時間を決定するための手段と、

前記第 1 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 1 のイベントを除去するための手段と、

前記第 2 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 2 のイベントを除去するための手段とをさらに備える、請求項 22 ~ 27 のいずれか一項に記載のシステム。

【発明の詳細な説明】

【技術分野】

10

【0001】

関連出願の相互参照

本願は、2013年1月15日に出願され「VARIABLE DURATION WINDOWS ON CONTINUOUS DATA STREAMS (連続データストリームに対する継続時間可変ウィンドウ)」と題された米国仮出願第61/752,850号に基づく利益および優先権を主張する。本願はまた、米国仮出願第61/752,850号に基づく優先権を主張する2013年3月15日に出願され「VARIABLE DURATION WINDOWS ON CONTINUOUS DATA STREAMS」と題された米国非仮出願第13/838,259号に基づく利益および優先権を主張する。出願第61/752,850号および第13/838,259号の内容全体をすべての目的のために本明細書に引用により援用する。

20

【背景技術】

【0002】

背景

開示されている実施形態は、概してデータ処理システムに関し、より具体的には継続時間可変ウィンドウを用いて連続イベントストリームを処理するための技術に関する。

【0003】

従来のデータベースシステムにおいて、データは、通常はテーブルの形態で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その後、SQL等のデータ管理言語を用いてクエリされ操作される。たとえば、SQLクエリは、データベースに格納されているデータのうちの関連データを識別するために定義し実行することができる。したがって、SQLクエリは、データベースに格納されている有限集合のデータに対して実行される。さらに、SQLクエリが実行されるとき、SQLは有限のデータ集合に対して一度実行されて有限の静的結果をもたらす。このように、データベースは、格納されている有限のデータ集合に対してクエリが実行されるよう最適に作成されている。

30

【0004】

しかしながら、数多くの現代のアプリケーションおよびシステムは、データを、有限のデータ集合ではなく連続データまたはイベントストリームの形態で生成する。このようなアプリケーションの例は、センサデータアプリケーション、財務表示機、ネットワークパフォーマンス測定ツール(たとえば、ネットワークモニタリングおよびトラフィック管理アプリケーション)、クリックストリーム分析ツール、自動車交通モニタリング等を含むが、これらに限定されない。たとえば、温度センサを、温度測定値を発信するように構成することができる。このようなアプリケーションから、データストリームを処理することが可能な新たな種類のアプリケーションの必要性が生じた。

40

【0005】

イベントストリームに基づくこれらの種類のアプリケーションの場合のデータの管理および処理は、時間を強く重視してデータ管理およびクエリ機能を構築することを含む。連続する無限のデータ集合に対する長時間のクエリ実行を含む、異なる種類のクエリ機構が必要である。現在イベントストリーム処理に適した製品一式を提供している販売業者はあるが、これら製品が提供するものは依然として、現代のイベント処理のニーズに対処するのに必要な処理の柔軟性に欠けている。

50

【発明の概要】

【課題を解決するための手段】

【0006】

簡単な概要

本発明のある実施形態は、一般的にはデータ処理システムに関し、より具体的には継続時間可変ウィンドウを用いて連続イベントストリームを処理するための技術に関する。この技術は、たとえば、イベントデータストリームに対して機能する、コンピュータによって実施される方法、イベントデータストリームを処理する計算機、システム、または装置、1つ以上の処理部によって実行可能な複数の命令を格納するコンピュータ読取可能なメモリなどを用いて実施される技術を含み得る。

10

【0007】

ある実施形態において、1つのイベントが時間ベースのウィンドウ内に留まる資格がある継続時間は、同一のイベントストリームを介して受信した異なるイベントごとに異なり得る。たとえば、このイベントストリームを介して受信した第1の入力イベントの場合にこの第1のイベントが時間ベースの範囲ウィンドウ内に留まる資格がある継続時間が「X」で、同一のイベントストリームを介して受信した第2のイベントの場合にこの第2のイベントが時間ベースのウィンドウ内に留まる資格がある継続時間が「Y」で、「X」が「Y」と同じでないことがある。第1および第2のイベントは、関連付けられたタイムスタンプが同一または異なる場合がある。本発明の実施形態はしたがって、1つのイベントが時間ベースのウィンドウ内にある時間の量を制御できるようにする。

20

【0008】

ある実施形態において、入力イベントが時間ベースのウィンドウ内にある継続時間は、このイベントの1つ以上の属性の1つ以上の値の関数である。イベントの種類が異なれば1つ以上の属性の値は異なり得るので、異なるイベントが時間ベースのウィンドウ内にある時間の量は異なり得る。したがって、1つのイベントが時間ベースのウィンドウ内にある時間の量は、このイベントの1つ以上の属性を介して制御することができる。

【0009】

一実施形態において、複数のイベントがイベントストリームを介して計算装置によって受信されてもよい。たとえば、第1のイベントおよび第2のイベントがイベントストリームを介して受信されてもよい。第1のイベントに対して第1の継続時間を決定してもよく、第1の継続時間は、このイベントストリームに対して指定された時間ベースのウィンドウ内に第1のイベントが留まる資格がある継続時間を示す。第2のイベントに対して第2の継続時間を決定してもよく、第2の継続時間は、時間ベースのウィンドウ内に第2のイベントが留まる資格がある継続時間を示し、第2の継続時間は第1の継続時間と異なる。

30

【0010】

ある実施形態において、第1および第2のイベントを、そのイベントストリームに対して指定された時間ベースのウィンドウに追加してもよい。第1のイベントは、時間ベースのウィンドウ内に第1の継続時間留まった後に、時間ベースのウィンドウから除去または削除される。第2のイベントは、時間ベースのウィンドウ内に第2の継続時間留まった後に、時間ベースのウィンドウから除去または削除される。このように、第1のイベントが時間ベースのウィンドウ内にある継続時間は、第2のイベントがこの時間ベースのウィンドウ内にある継続時間と異なり得る。

40

【0011】

上記第1および第2のイベントの例の実施形態において、第1のイベントは関連付けられた第1の時間を有していてもよく、第2のイベントは関連付けられた第2の時間を有していてもよい。一例において、第1の時間と第2の時間は同一であってもよい。第2の例において、第1の時間は第2の時間と異なってもよい。

【0012】

ある実施形態において、1つのイベントが時間ベースのウィンドウ内にある資格がある継続時間は、このイベントの1つ以上の属性の値に基づいていてもよい。

50

【 0 0 1 3 】

ある実施形態において、行 時間範囲ウィンドウは、異なる区画であってもよい。1つのイベントが属する区画は、第1のイベントの属性の値に基づいていてもよい。1つのイベントが、1つの区画のための行 時間範囲ウィンドウ内にある資格がある時間の量は、このイベントの1つ以上の属性の1つ以上の値に基づいていてもよい。したがって、特定の区画に属するイベントについては、異なるイベントは、この特定の区画のための行 時間範囲ウィンドウ内に異なる継続時間留まる資格を有し得る。

【 0 0 1 4 】

ある実施形態において、第1のイベントに関連付けられた第1の時間と第1の継続時間とに基づいて、第1のイベントの第1の満了時間を計算してもよい。第2のイベントに関連付けられた第2の時間と第2の継続時間とに基づいて、第2のイベントの第2の満了時間を計算してもよい。第1のイベントは、第1の満了時間の到来時または到来後に、時間ベースのウィンドウから除去または削除される。同様に、第2のイベントは、第2の満了時間の到来時または到来後に、時間ベースのウィンドウから除去または削除されてもよい。

10

【 0 0 1 5 】

本開示の一局面に従い、計算装置が提供される。この計算装置は、計算装置がイベントストリームを介して受信した第1のイベントの第1の継続時間を決定するように構成された第1の決定部を含み、第1の継続時間は、イベントストリームに対して指定された時間ベースのウィンドウ内に第1のイベントが留まる資格がある継続時間を示す。計算装置は、計算装置がイベントストリームを介して受信した第2のイベントの第2の継続時間を決定するように構成された第2の決定部を含んでいてもよく、第2の継続時間は、時間ベースのウィンドウ内に第2のイベントが留まる資格がある継続時間を示す。第2の継続時間は第1の継続時間と異なってもよい。

20

【 0 0 1 6 】

ある実施形態において、計算装置はさらに、第1の時間イベントが時間ベースのウィンドウ内に第1の継続時間留まった後に第1のイベントを時間ベースのウィンドウから除去するように構成された第1の除去部と、第2の時間イベントが時間ベースのウィンドウ内に第2の継続時間留まった後に第2のイベントを時間ベースのウィンドウから除去するように構成された第2の除去部とを含む。

30

【 0 0 1 7 】

ある実施形態において、第1のイベントは関連付けられた第1の時間を有し、第2のイベントは関連付けられた第2の時間を有し、第1の時間は第2の時間と同一である。

【 0 0 1 8 】

ある実施形態において、第1のイベントは関連付けられた第1の時間を有し、第2のイベントは関連付けられた第2の時間を有し、第1の時間は第2の時間と異なる。

【 0 0 1 9 】

ある実施形態において、第1の継続時間を決定することは、第1のイベントの1つ以上の属性の1つ以上の値に基づいて第1の継続時間を計算することを含む。さらに、第2の継続時間を決定することは、第2のイベントの1つ以上の属性の1つ以上の値に基づいて第2の継続時間を計算することを含む。

40

【 0 0 2 0 】

ある実施形態において、第1の決定部は、第1のイベントの第1の属性の値に基づいて、第1のイベントのための第1の区画を決定するように構成される。第2の決定部は、第2のイベントの第1の属性の値に基づいて、第2のイベントは第1の区画に属すると決定するように構成される。時間ベースのウィンドウは第1の区画のためのものである。

【 0 0 2 1 】

ある実施形態において、第1の決定部は、第1のイベントに関連付けられた第1の時間と第1の継続時間とに基づいて第1のイベントの第1の満了時間を決定するように構成される。第2の決定部は第2のイベントに関連付けられた第2の時間と第2の継続時間とに

50

基づいて第2のイベントの第2の満了時間を決定するように構成される。第1の除去部は、第1の満了時間の到来時または到来後に時間ベースのウィンドウから第1のイベントを除去するように構成される。第2の除去部は、第2の満了時間の到来時または到来後に時間ベースのウィンドウから第2のイベントを除去するように構成される。

【0022】

以下の詳細な説明は、添付の図面とともに、本発明の実施形態の性質の一層の理解をもたらすであろう。

【図面の簡単な説明】

【0023】

【図1】本発明の実施形態を組み込み得るイベント処理システムの簡略化された高レベルの図を示す。

10

【図2】本発明の実施形態に従い、継続時間可変ウィンドウを管理するための方法を示す簡略化されたフローチャートを示す。

【図3】本発明の実施形態に従い、継続時間可変の時間ベースウィンドウ処理を実現するために使用し得るモジュールおよびデータ構造を示す。

【図4】上記のように優先度キューを用いて継続時間可変の時間ベースウィンドウを実現する実施形態において時間ベースウィンドウオペレータが実行し得る処理を示す簡略化されたフローチャートを示す。

【図5】本発明の実施形態に従い、区画のための継続時間可変ウィンドウを管理するための方法を示す簡略化されたフローチャートを示す。

20

【図6】本発明の実施形態に従い、優先度キューを用いて区画のための継続時間可変ウィンドウを扱うために時間ベースウィンドウオペレータが実行し得る処理を示す簡略化されたフローチャートを示す。

【図7】本発明の実施形態に従い使用し得るシステム環境の構成要素を示す簡略化されたブロック図である。

【図8】本発明のある実施形態に従い使用し得るコンピュータシステムの簡略化されたブロック図である。

【図9】本発明のある実施形態に従い使用し得る計算装置の簡略化されたブロック図である。

【発明を実施するための形態】

30

【0024】

詳細な説明

以下の記載では、説明を目的として具体的な詳細事項が本発明の実施形態の十分な理解をもたらすべく述べられている。しかしながら、さまざまな実施形態がこういった具体的な詳細事項なしで実施し得ることは明らかであろう。図面および説明は限定を意図したものではない。

【0025】

データのストリームを処理するための改善された技術を提供する、ある実施形態について説明する。1つのイベントが時間ベースのウィンドウ内に留まる資格がある継続時間は、同一イベントストリームを介して受信した異なるイベントごとに異なり得る。ある実施形態において、入力イベントが時間ベースのウィンドウ内にある継続時間は、このイベントの1つ以上の属性の1つ以上の値の関数である。イベントが異なれば1つ以上の属性値は異なり得るので、異なるイベントがそのイベントストリーム用の時間ベースのウィンドウ内にある時間の量は異なり得る。1つのイベントが時間ベースのウィンドウ内にある時間の量は、このイベントの1つ以上の属性を介して制御することができる。

40

【0026】

連続データストリーム（イベントストリームまたはイベントデータストリームとも呼ばれる）は、明確な終わりがなく、本質的に連続するまたは無限であるデータまたはイベントのストリームである。論理的に、イベントまたはデータストリームは、一連のデータ要素（イベントとも呼ばれる）であり、各データ要素は関連付けられたタイムスタンプを有

50

する。連続イベントストリームは、論理的には要素 (s, T) のバッグ (bag) またはセットで表わすことができ、「s」はデータ部分を表わし「T」は時間ドメインを表わす。「s」部分は一般的にタプルまたはイベントと呼ばれる。イベントストリームはしたがってタイムスタンプされた一連のタプルまたはイベントである。

【0027】

いくつかの実施形態において、あるストリーム内のイベントに関連付けられるタイムスタンプは、クロック時間に匹敵し得る。しかしながら、他の実施形態において、あるイベントストリーム内のイベントに関連付けられる時間は、アプリケーションドメインによって定義してもよく、クロック時間に対応しなくてもよいが、たとえばその代わりに連続番号によって表わしてもよい。したがって、あるイベントストリーム内のイベントに関連付けられる時間情報は、数字、タイムスタンプ、または時間の連続の概念を表わす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す順序で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がある。

【0028】

いくつかの実施形態において、あるイベントストリーム内のイベントは、世の中の何らかのイベントの発生（たとえば、温度センサの値が新たな値に変化したとき、株式銘柄の価格が変化したとき）を表わしてもよく、このイベントに関連付けられた時間情報が、データストリームイベントによって表わされる上記世の中のイベントが発生した時間を示してもよい。

【0029】

イベントストリームを介して受信するイベントについては、イベントに関連付けられた時間情報を用いて、イベントストリーム内のイベントがタイムスタンプの値が増す順序で到着することを保証する。これにより、受信するイベントストリーム内のイベントを、それぞれのイベントに関連付けられた時間情報に基づいて並べることができる。このように並べることができるようにするためには、後で生成されたイベントが、先に生成されたイベントよりも後のタイムスタンプを有するように、タイムスタンプを、イベントストリーム内のイベントと、タイムスタンプが減少しないやり方で関連付ければよい。別の例として、連続番号を時間情報として用いるのであれば、後に生成されたイベントに関連付けられる連続番号は、先に生成されたイベントに関連付けられる連続番号よりも大きい。同一のイベントストリームに属するイベントは通常、関連付けられた時間情報によってこれらイベントに課された順序で処理され、先のイベントは後のイベントよりも前に処理される。いくつかの実施形態において、たとえば、データストリームイベントで表わされる世の中のイベントが同時に発生するときには、複数のイベントが同一のタイムスタンプまたは連続番号に関連付けられることがある。これらの状況において、イベントは受信された順序で処理される。

【0030】

イベントストリーム内のイベントに関連付けられる時間情報（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、その代わりに、この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、ある実施形態において、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 (heartbeat) を管理してもよく、イベントに関連付けられる時間は、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いてもよい。あるイベントストリーム内の2つのイベントが同一の時間情報を有することが起こり得る。タイムスタンプの順序条件は、1つのイベントストリームに特有のものであるが、異なるストリームのイベントを、任意で交互配置してもよいことに注意しなければならない。

【0031】

イベントストリームには、関連付けられたスキーマ「S」があり、スキーマは、時間情報と、名前付きの1つ以上の属性からなる一組の属性とを含む。特定のイベントストリームに属するすべてのイベントは、この特定のイベントストリームに関連付けられたスキーマ

10

20

30

40

50

マに従う。したがって、イベントストリーム (s , T) の場合、このイベントストリームは、スキーマ「 S 」を、 (< time_stamp > , < attribute(s) >) として有し得る。 < attributes > は、スキーマのデータ部分を表わし 1 つ以上の属性を含み得る。たとえば、株式相場表示機のイベントストリームについてのスキーマは、 < 株式銘柄 > および < 株価 > という属性を含み得る。このようなストリームを介して受ける各イベントは、 1 つのタイムスタンプと上記 2 つの属性とを有する。たとえば、株式相場表示機の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けることができる。

【 0 0 3 2 】

...

(< timestamp_N > , < NVDA , 4 >)

(< timestamp_N+1 > , < ORCL , 62 >)

(< timestamp_N+2 > , < PCAR , 38 >)

(< timestamp_N+3 > , < SPOT , 53 >)

(< timestamp_N+4 > , < PDCO , 44 >)

(< timestamp_N+5 > , < PTEN , 50 >)

...

上記ストリームにおけるストリーム要素 (< timestamp_N+1 > , < ORCL , 62 >) の場合、「 stock_symbol 」 (株式銘柄) および「 stock_value 」 (株価) という属性に対するイベント値はそれぞれ O R C L および 6 2 である。この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。このように、連続イベントストリームは、イベントのフローであり、各イベントは同じ一連の属性を有する。

【 0 0 3 3 】

図 1 は、本発明の実施形態を組み込み得るイベント処理システム 1 0 0 の簡略化された高レベルの図を示す。イベント処理システム 1 0 0 は、 1 つ以上のイベントソース (1 0 4 , 1 0 6 , 1 0 8) 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 (event processing server) (E P S) 1 0 2 と、 1 つ以上のイベントシンク (1 1 0 , 1 1 2) とを含み得る。イベントソースは、 E P S 1 0 2 が受信するイベントストリームを生成する。 E P S 1 0 2 は、 1 つ以上のイベントストリームを 1 つ以上のイベントソースから受信し得る。たとえば、図 1 に示されるように、 E P S 1 0 2 は、入力イベントストリーム 1 1 4 をイベントソース 1 0 4 から受信し、第 2 の入力イベントストリーム 1 1 6 をイベントソース 1 0 6 から受信し、第 3 のイベントストリーム 1 1 8 をイベントソース 1 0 8 から受信する。 1 つ以上のイベント処理アプリケーション (1 2 0 , 1 2 2 , および 1 2 4) を E P S 1 0 2 に導入し E P S 1 0 2 によって実行してもよい。 E P S 1 0 2 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リッスンし、 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、入力イベントストリームから 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る処理ロジックに基づいて処理するように構成されてもよい。注目すべきイベントは次に、 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 (1 1 0 , 1 1 2) に送信してもよい。たとえば、図 1 において、 E P S 1 0 2 は、出力イベントストリーム 1 2 6 をイベントシンク 1 1 0 に出力し、第 2 の出力イベントストリーム 1 2 8 をイベントシンク 1 1 2 に出力する。ある実施形態において、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、これら構成要素のうちのいずれも、他の構成要素の変更を生じさせることなく追加または除去できるよう、互いに分離されている。

【 0 0 3 4 】

一実施形態において、 E P S 1 0 2 を、サービスを共有する Equinox OSGi をベースとするもののような軽量 J a v a (登録商標) アプリケーションコンテナを含む J a v a サーバとして実現してもよい。いくつかの実施形態において、 E P S 1 0 2 は、たとえば JRoc kit Real Time を用いてイベント処理の超高スループットおよびマイクロ秒レイテンシをサポートし得る。 E P S 1 0 2 はまた、イベント処理アプリケーションを開発するための

ツール（たとえばOracle CEP VisualizerおよびOracle CEP IDE）を含む開発プラットフォーム（たとえば完全なリアルタイムエンドツーエンドJavaイベント駆動型アーキテクチャ（Event-Driven Architecture）（EDA）開発プラットフォーム）を提供し得る。

【0035】

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リッスンし、この1つ以上の入力イベントストリームから注目すべき1つ以上のイベントを選択するためにロジック（たとえばクエリ）を実行し、選択した注目すべきイベントを1つ以上の出力イベントストリームを介して1つ以上のイベントソースに出力するように構成される。図1は、1つのこのようなイベント処理アプリケーション120についてのドリルダウンを提供する。図1に示されるように、イベント処理アプリケーション120は、入力イベントストリーム118をリッスンし、入力イベントストリーム118から注目すべき1つ以上のイベントを選択するためにロジックを含むクエリ130を実行し、選択した注目すべきイベントを出力イベントストリーム128を介してイベントシンク112に出力するように構成される。イベントソースの例は、アダプタ（たとえばJMS、HTTP、およびファイル）、チャンネル、プロセッサ、テーブル、キャッシュ等を含むがこれらに限定されない。イベントシンクの例は、アダプタ（たとえばJMS、HTTP、およびファイル）、チャンネル、プロセッサ、キャッシュなどを含むがこれらに限定されない。

10

【0036】

図1のイベント処理アプリケーション120は、1つの入力ストリームをリッスンし選択したイベントを1つの出力ストリームを介して出力するものとして示されているが、これは限定を意図したものではない。これに代わる実施形態では、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リッスンし、モニタリングされたストリームからイベントを選択し、選択したイベントを1つ以上の出力イベントストリームを介して1つ以上のイベントシンクに出力するように構成されてもよい。同一のクエリを2つ以上のイベントシンクに関連付け異なる種類のイベントシンクに関連付けてもよい。

20

【0037】

その性質上無制限なので、1つのイベントストリームを介して受信されるデータの量は通常非常に多い。結果として、クエリを目的としてすべてのデータを格納またはアーカイブすることは、通常実用的でなく望ましくない。イベントストリームを処理するには、イベントをEPS102が受信したときに、受信したイベントデータをすべて格納することなく、イベントをリアルタイムで処理することが必要である。したがって、EPS102は、EPS102がイベントを受信したときに、受信したイベントをすべて格納することなく、イベントを処理できるようにする、特別なクエリ機構を提供する。

30

【0038】

イベント駆動型のアプリケーションはルール駆動型であり、これらルールは、入力ストリームを処理するのに使用される連続クエリの形態で表わしてもよい。連続クエリは、クエリ処理の結果としてどのイベントを注目すべきイベントとして選択し出力すべきかということを含む、受信したイベントに対して実行すべき処理を特定する命令（たとえばビジネスロジック）を含み得る。連続クエリは、データ記憶装置まで持続させて、イベントの入力ストリームの処理およびイベントの出力ストリームの生成のために使用してもよい。連続クエリは、発見すべきフィルタリングおよび集約関数を指定し入力イベントストリームから注目すべきイベントを抽出してもよい。結果として、出力イベントストリームにおけるアウトバウンドのイベントの数は通常、イベントを選択する元になる入力イベントストリームにおけるイベントの数よりも遥かに少ない。

40

【0039】

有限データ集合に対して一度実行されるSQLクエリと異なり、特定のイベントストリームに対してアプリケーションによりEPS102に登録された連続クエリは、このイベントストリーム内のイベントが受信される度に実行されてもよい。連続クエリ実行の一部

50

として、E P S 1 0 2 は、受信したイベントを、連続クエリが指定する命令に基づいて評価することにより、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決定し、連続クエリ実行の結果として出力する。

【 0 0 4 0 】

連続クエリは異なる言語を用いてプログラムしてもよい。ある実施形態において、連続クエリは、Oracle社が提供する連続クエリ言語 (Continuous Query Language) (C Q L) を用いて構成されOracle社の複合イベント処理 (Complex Events Processing) (C E P) という製品によって使用されてもよい。Oracle社のC Q Lは、イベントストリームに対して実行することができるクエリ (C Q L クエリと呼ばれる) をプログラムするのに使用できる宣言型言語である。ある実施形態において、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構成が追加されたS Q Lに基づく。

10

【 0 0 4 1 】

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

【 0 0 4 2 】

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

20

【 0 0 4 3 】

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

【 0 0 4 4 】

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

30

【 0 0 4 5 】

(4) 1 つ以上のビーン (bean) は、出力チャンネルをリッスンするように構成または登録されてもよく、新たなイベントが出力チャンネルに挿入されたことによってトリガされてもよい。いくつかの実施形態において、このユーザコードは、昔からある単なるJ a v aオブジェクト (plain-old-Java-object) (P O J O) であってもよく、または、ユーザコードがO r a c l e C E PのイベントビーンA P Iを使用してビーンをO r a c l e C E Pによって管理できるようにしてもよい。ユーザアプリケーションは、J M S、ウェブサービス、ファイルライター等の一組の外部サービスを利用して、生成されたイベントを外部イベントシンクに転送することができる。

40

【 0 0 4 6 】

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

【 0 0 4 7 】

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

50

【 0 0 4 8 】

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

10

【 0 0 4 9 】

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

【 0 0 5 0 】

イベントおよびアプリケーションの例

（ 1 ）市場変動に基づいた株式取引を自動化するための金融アルゴリズム取引アプリケーション。クエリの例：いずれかの 2 0 秒ウィンドウ内で株式 B の上昇が 2 % を上回り株式 A はそうではない場合、自動的に株式 A を購入する。

20

【 0 0 5 1 】

（ 2 ）運輸。イベント間のパターンを検出することによって不正行為を発見するための安全および不正検出アプリケーション。クエリの例：都市地下鉄システムにアクセスするために 1 枚の I D カードが 5 秒未満の間に 2 度使用された場合、ピギーバックについてセキュリティ警告を発する。

【 0 0 5 2 】

（ 3 ）偽陽性（false positive）警告を減じるためのエネルギーおよび電気通信警告関連アプリケーション。クエリの例：いずれかの 5 秒ウィンドウ内で 1 5 の警告を受けたが 3 0 秒以内に検出された同様の警告が 5 未満の場合、何もしない。

30

【 0 0 5 3 】

（ 4 ）患者のバイタルサインをモニタリングし特定のイベントが発生した場合に何らかのタスクを実行するためのヘルスケア患者モニタリングアプリケーション。クエリの例：薬の変更に続いて患者の血圧がいずれかの 1 0 秒ウィンドウ内でこの患者の最大許容血圧の 2 0 % 以内上昇した場合、最も近い看護師に通知する。

【 0 0 5 4 】

ウィンドウベースの処理

別の例として、イベント処理アプリケーションは、ある時間範囲のデータまたはイベントを照合し処理するように構成されてもよい。たとえば、E P S 1 0 2 は、「温度（temp）」属性および関連付けられた時間を含むイベントからなる温度イベントストリームを受信してもよい。「温度」データは、関連付けられた時間における温度を示してもよい。前の 1 0 秒の平均温度を得るために、C Q L クエリをこのようなイベントストリーム用に構築してもよい。クエリは次のように指定してもよい。

40

【 0 0 5 5 】

```
Select avg (temp) from temp [range 10]
```

上記クエリは、「1 0」という範囲（range）における「温度（temp）」の平均（avg）を発見する。この例における範囲「1 0」は 1 0 秒を表わす。他の実施形態において、この範囲を、分、日、またはそれ以外の指定された時間単位等、他の時間単位

50

を用いて指定してもよい。このCQLクエリは、EPS 102によって実行されると、前の10秒で受信したタプルを評価し平均温度を見出す。「満了」タプル、すなわち示されている範囲よりも前に（すなわち10秒ウィンドウ外で）受信したタプルは、クエリによって無視される。したがって、CQLクエリは、受信したイベントすべてを捕えて格納する必要はなく、最新の10秒ウィンドウ内のイベントを処理するだけでよい。

【0056】

イベントストリームは無限の可能性のあるタプルストリームを表わすので、タプルを受信したときに処理するために、CQLクエリ（または一般的には連続クエリ）は、CQLクエリの各実行時に処理するのに使用される受信イベントのサブセットを特定する。CQLクエリはこれをウィンドウを指定することによって行なう。この場合のウィンドウは、CQLクエリを実行するときに処理目的で使用する受信イベントのサブセット（すなわち一部）を定める。ウィンドウは、イベントストリームからのゼロ以上のイベントからなる有限のサブセットである。

【0057】

オラクルイベント処理のCQLのセマンティクスでは、ウィンドウは、ストリーム・リレーションオペレータとして定義され、すべての瞬間において、リレーションは、順序付けられていない、時間によって変化する、イベントの有限の集合（またはバッグ）である。一実施形態において、ストリーム・リレーションオペレータは、ストリームに対してスライディングウィンドウを提供し、ウィンドウ仕様を用いて定められる。いくつかの実施形態において、ウィンドウは、

- （１）時間ベース 特定の継続時間に基づくもの、
- （２）タプルベースまたはイベントベース イベントの数に基づくもの、または、
- （３）区画ベース 各区画で時間ベースまたはタプルベースのものであってもよい。

【0058】

時間ベースのウィンドウ

イベントストリームを処理するために、時間ベースのウィンドウは、このウィンドウに含まれるイベントを制御する時間パラメータを用いて定められる。いくつかの実施形態において、時間パラメータは、このウィンドウの時間範囲を指定してもよい。たとえば、CQLクエリは、入力ストリームSに対する時間範囲ベースのウィンドウを次のように指定してもよい。

【0059】

S [range T]

上記CQLコードは、継続時間「T」をパラメータとして取る、ストリームに対する時間ベースの範囲ウィンドウを作成し、出力リレーションを生成する。継続時間Tは、分、秒、日等の何等かの時間単位で指定されればよく、一実施形態では、単位が指定されなければデフォルトを秒にする。このように、「範囲」というキーワードを用いて指定された時間ベースのウィンドウは、ストリームSに対し、時間間隔「T」をパラメータとして取り出力リレーション「R(t)」を生成する時間ベースのスライディングウィンドウを作成する。時間「t」において、出力リレーションR(t)は、入力イベントストリームSからの、「t-T」と「t」の間の関連するタイムスタンプを有するすべてのイベントを含み、「T」はこのウィンドウのサイズすなわち継続時間である。

【0060】

時間ベースの範囲ウィンドウの過去の実装例では、CQLにおいて時間ウィンドウの継続時間を指定するために使用される時間パラメータ「T」は、一定または固定されていた。したがって、以前の実装例では、各イベントが時間ベースのウィンドウ内にある時間の量は、イベント自体に応じて異なるものではない。すべてのイベントは等しく扱われ、すべてのイベントは時間ベースのウィンドウ内に同じ継続時間存在する。たとえば、先に説明した温度平均のCQLクエリの例では、範囲は「[range 10]」として指定され、「T」の値は10秒を表わす固定値「10」に設定される。この範囲ウィンドウは固定されており変動しないという性質のものであるので、時間「t」において、ウィンドウ

は最新の10秒(すなわち「t」と「t-10」の間)で受信したイベントのリレーションセットを評価する。ウィンドウ自体は「t」でスライドするが、ウィンドウの継続時間は常に10秒に固定されることに注意しなければならない。継続時間が固定されているので、以前は、時間ベースのウィンドウ内のイベントは常に、この時間ウィンドウ内の同じ継続時間後に満了した。

【0061】

固定されたまたは一定の時間ベースの範囲ウィンドウに加えて、本発明のある実施形態は、時間ベースのウィンドウの継続時間を、同一のイベントストリームを介して受信した異なるイベントごとに変えることができる。このような実施形態では、スライドする時間ベースの範囲ウィンドウの継続時間を定めるために用いられる継続時間パラメータ「T」の値を、異なるイベントごとに異なるようにする、したがって変えることができる。このため、特定のイベントストリームに対してEPS102に登録されたクエリにとって、このイベントストリームを介して受信したイベントの継続時間は、異なるイベントごとに異なり得る。たとえば、図1に示されるように、イベントアプリケーション120は、可変の時間ベースのウィンドウを指定するCQLコードを有する連続クエリ130を含み得る。

10

【0062】

たとえば、イベントストリームを介して受信した第1の入力イベントについて、第1のイベントが時間ベースの範囲ウィンドウ内に留まる資格がある継続時間は「X」で、同じイベントストリームを介して受信した第2のイベントについて、第2のイベントが時間ベースのウィンドウ内に留まる資格がある継続時間は「Y」で、「X」が「Y」と同一でないという場合がある。第1および第2のイベントは、同じ関連タイムスタンプを有し得る、または、異なるタイムスタンプを有し得る。本発明の実施形態はしたがって、イベントが時間ベースのウィンドウ内にある時間の量を制御できるようにする。

20

【0063】

イベントに対して継続時間可変範囲パラメータTの値を設定する方法としてはさまざまな方法がある。いくつかの実施形態において、継続時間パラメータTは、ユーザによって構成可能なものであってもよい。ある実施形態において、Tの値(すなわち時間ベースの範囲ウィンドウの継続時間)は、イベント自体の関数である。たとえば、受信したイベントに対するTの値は、受信したイベントの1つ以上の属性の値に基づいて設定してもよい。たとえば、関連するスキーマが2つの属性<attr_1, attr_2>を有するイベントストリームを想定する。一実施形態において、このイベントストリームを対象とするCQLクエリにおいて、イベントが時間ベースの範囲ウィンドウ内にいることが可能な時間を、属性「attr_2」の値に設定されるようにプログラムしてもよい。たとえば、イベントストリームが下記イベントを受信するとする。

30

【0064】

...

(<timestamp_N>,<attr_1=100,attr_2=5>)

(<timestamp_N+1>,<attr_1=100,attr_2=2>)

(<timestamp_N+2>,<attr_1=200,attr_2=2>)

(<timestamp_N+3>,<attr_1=600,attr_2=4>)

40

...

timestamp_Nを有するイベントを受信したとき、このイベントが時間ベースの範囲ウィンドウ内に留まることができる継続時間は、5時間単位と計算される。なぜなら、タブルのattr_2の値が5であるからである。timestamp_N+1を有するイベントの場合、このイベントが時間ベースの範囲ウィンドウ内に留まることができる継続時間は、2時間単位と計算され、timestamp_N+2を有するイベントの場合、このイベントが時間ベースの範囲ウィンドウ内に留まることができる継続時間は、2時間単位と計算され、timestamp_N+3を有するイベントの場合、このイベントが時間ベースの範囲ウィンドウ内に留まることができる継続時間は、4時間単位と計算

50

され、以降同様に続く。このようにして、1つのイベントがそのイベントストリームに対する時間ベースのウィンドウ内に留まる資格がある継続時間は、イベントごとに異なり得る。上記例において、継続時間は受信したイベントの属性「`attr__2`」に基づいて変化する。

【0065】

他のいくつかの実施形態において、時間パラメータ `T` は、数式 (mathematical expression) として次のように表わしてもよい。

【0066】

`S [RANGE <mathematical_expression>]`

ある実施形態において、この数式は、イベントストリームスキーマの1つ以上の属性に基づくものであってもよい。たとえば、スキーマ属性が `< attr__1, attr__2 >` であるイベントストリームについて、`< mathematical_expression >` は、`< attr__2 + 4 >`、`< attr__1 + attr__2 >` 等として定めてもよい。これに代わる実施形態において、イベントベースであってもなくてもよい他の関数または条件を用いて、連続クエリの時間ベースの範囲ウィンドウの継続時間を設定し変化させてもよい。

【0067】

別の例として、ストリームを次のように定義とする。

`CREATE STREAM S (DATA INTEGER, RANGE_COLUMN BIGINT)`

この定義によると、各ストリームイベントは、2つの属性 (カラムとも呼ばれる)、すなわち、整数データ値を取る第1の属性「`DATA`」と、`bigint` (長い) 値を取る第2の属性「`RANGE_COLUMN`」を含む。次に、この「`RANGE_COLUMN`」属性を用いて、時間ベースの範囲ウィンドウの継続時間を指定してもよい。

【0068】

継続時間可変の時間ベース範囲ウィンドウを指定する `CQL` クエリは、ストリーム `S` に対し、次のように指定してもよい。

【0069】

`CREATE QUERY Q1 AS SELECT * FROM S [RANGE <range_expression>]`

これにより、時間ベースの範囲ウィンドウに対して連続クエリ `Q1` が作成され、この場合、時間ベースのウィンドウの継続時間が「`range_expression`」という式として指定される。

【0070】

一実施形態において、`range_expression` は、ストリーム `S` からの1つ以上の属性に基づいていてもよい。たとえば次の通りである。

【0071】

`CREATE QUERY Q1 AS SELECT * FROM S [RANGE RANGE_COLUMN]`

ここで、受信したイベントの属性 `RANGE_COLUMN` の値は、時間ベースの範囲ウィンドウの継続時間を指定する。この場合、`< range_expression >` の値は、`S` というイベントストリームを介して受信した各入力イベントに対し、受信した入力イベントの属性 `RANGE_COLUMN` の値を用いて計算される。時間「`t`」で受信した入力タプルは、時間ベースのウィンドウに挿入され、この時間ベースのウィンドウ内に、`< range_expression >` の値に従う継続時間留まった後、満了するかまたはウィンドウから除去される。受信イベントに対して計算される満了時間はしたがって、`(t + range_expression_value)` であり、「`range_expression_value`」は、イベントを受信したときに、受信したイベントの `RANGE_COLUMN` 属性の値を用いて `< range_expression >` の値を求めることによって計算される値である。属性 `RANGE_COLUMN` の値はイベントごとに異なり得るので、時間ベースのウィンドウの継続時間もイベントごとに異なり得る。

【0072】

10

20

30

40

50

以下の例は、イベント受信時の継続時間可変の時間ベース範囲ウィンドウの効果を示す。この例について、入力イベントストリーム S は、3つの属性 c 1、c 2、および c 3として (c 1 integer , c 2 bigint , c 3 double) を有するスキーマ (< timestamp > , < attributes >) を有すると想定する。連続 C Q L クエリをストリーム S に対して以下のように指定してもよい。

【 0 0 7 3 】

SELECT * FROM S [range c2]

このクエリによると、受信した各入力について、時間ベースのウィンドウの継続時間は属性 c 2 の値に基づく。

【 0 0 7 4 】

以下の表 A は、タブルの入カストリームと、入力イベントストリーム S を介してイベントを受信するときのさまざまな時点における時間ベースの範囲ウィンドウの内容とを示す。

【 0 0 7 5 】

【表 1】

入カイベントストリーム スキーマ:(タイムスタンプ, c1, c2, c3)	ウィンドウ出力内容 スキーマ:(タイムスタンプ, c1, c2, c3)
(1000, 10, 1, 10.0)	$+(1000, 10, 1, 10.0)^{2s}$
(2000, 20, 2, 20.0)	$-(1000, 10, 1, 10.0)^{4s}, +(2000, 20, 2, 20.0)^{4s}$
(3000, 30, 3, 30.0)	$(2000, 20, 2, 20.0)^{4s}, +(3000, 30, 3, 30.0)^{6s}$
(4000, 40, 4, 40.0)	$-(2000, 20, 2, 20.0)^{4s}, (3000, 30, 3, 30.0)^{6s},$ $+(4000, 40, 4, 40.0)^{8s}$
(5000, 50, 5, 50.0)	$(3000, 30, 3, 30.0)^{6s}, (4000, 40, 4, 40.0)^{8s},$ $+(5000, 50, 5, 50.0)^{10s}$
(6000, 60, 6, 60.0)	$-(3000, 30, 3, 30.0)^{6s}, (4000, 40, 4, 40.0)^{8s},$ $(5000, 50, 5, 50.0)^{10s}, +(6000, 60, 6, 60.0)^{12s}$
(7000, 70, 7, 70.0)	$(4000, 40, 4, 40.0)^{8s}, (5000, 50, 5, 50.0)^{10s},$ $(6000, 60, 6, 60.0)^{12s}, +(7000, 70, 7, 70.0)^{14s}$
(8000, 80, 8, 80.0)	$-(4000, 40, 4, 40.0)^{8s}, (5000, 50, 5, 50.0)^{10s},$ $(6000, 60, 6, 60.0)^{12s}, (7000, 70, 7, 70.0)^{14s},$ $+(8000, 80, 8, 80.0)^{16s}$
(9000, 90, 9, 90.0)	$(5000, 50, 5, 50.0)^{10s}, (6000, 60, 6, 60.0)^{12s},$ $(7000, 70, 7, 70.0)^{14s}, (8000, 80, 8, 80.0)^{16s},$ $+(9000, 90, 9, 90.0)^{18s}$
(10000, 100, 10, 100.0)	$-(5000, 50, 5, 50.0)^{10s}, (6000, 60, 6, 60.0)^{12s},$ $(7000, 70, 7, 70.0)^{14s}, (8000, 80, 8, 80.0)^{16s},$ $(9000, 90, 9, 90.0)^{18s}, +(10000, 100, 10, 100.0)^{20s}$
(11000, 110, 11, 110.0)	$(6000, 60, 6, 60.0)^{12s}, (7000, 70, 7, 70.0)^{14s},$ $(8000, 80, 8, 80.0)^{16s}, (9000, 90, 9, 90.0)^{18s},$ $(10000, 100, 10, 100.0)^{20s}, +(11000, 110, 11, 110.0)^{22s}$
(12000, 120, 12, 120.0)	$-(6000, 60, 6, 60.0)^{12s}, (7000, 70, 7, 70.0)^{14s},$ $(8000, 80, 8, 80.0)^{16s}, (9000, 90, 9, 90.0)^{18s},$ $(10000, 100, 10, 100.0)^{20s}, (11000, 110, 11, 110.0)^{22s},$ $+(12000, 120, 12, 120.0)^{24s}$

表A:継続時間可変の時間ベース範囲ウィンドウ処理

【0076】

上記表Aにおいて、左の列はスキーマ(タイムスタンプ, c1, c2, c3)を有するストリームSを介して受信されたイベントのストリームを示す。タイムスタンプはミリ秒(msec)で示され、1秒=1000msecである。タイムスタンプは、たとえば、イベントの受信時の時間またはイベントの発生時の時間であってもよい。表Aの右の

10

20

30

40

50

列は、さまざまな時点（ミリ秒（`m s e c s`）で示され、1秒 = 1000 `m s e c s`である）における、時間ベースの範囲ウィンドウの内容（すなわちイベント）を示す。イベントの前の「+」表記は、このイベントが時間ベースのウィンドウに追加されたことを示すために使用される。イベントの前の「-」表記は、このイベントが、このイベントが満了したために、時間ベースのウィンドウから削除されたことを示すために使用される。

【0077】

一実施形態において、イベントが受信されると、このイベントの満了時間が計算される。イベントに対して計算される満了時間は、このイベントが満了し時間ベースのウィンドウから削除されることになる時間を示す。その後このイベントは時間ベースのウィンドウに追加される。また、時間ベースのウィンドウが検査されて、このウィンドウ内のいずれかのイベントが満了しておりウィンドウから取除かれる必要があるか否か判断される。

10

【0078】

時間「`t`」に到着し時間範囲または継続時間が「`T`」であるイベントについて、このイベントの満了時間は（`t + T`）である。以下のクエリの場合、範囲の値は、受信したイベントの属性 `c 2` の値に基づく。

【0079】

```
SELECT * FROM S [range c2]
```

したがって、関連付けられたタイムスタンプが「`t`」である受信イベントについて、このイベントの満了時間は（`t + c 2` の値）である。上記表 A において、各受信イベントの満了時間（`expiration time`）は、このイベントの添え字として示されている。

20

【0080】

【数 1】

（たとえば（タイムスタンプ，`c 1`，`c 2`，`c 3`）`Expiration_time`）

【0081】

ある実施形態において、範囲句によって具体的に指定されない限り、範囲の時間単位のデフォルトは秒であり、したがって、`S [range 1]` は、`S [range 1 秒]` に等しい。

【0082】

時間ベースのウィンドウ（すなわち時間ベースのウィンドウによって出力されるリレーションセット）は、1000 ミリ秒の時点でイベントが到着する前、すなわち関連付けられたタイムスタンプが1000 ミリ秒であるイベントが到着する前は、空である。表 A に示されるように、イベント（1000，10，1，10.0）が最初に受信される。このイベントの満了時間は2秒（イベントのタイムスタンプ + `c 2` の値、すなわち $1 + 1 = 2$ 秒）と計算される。次に、このイベントは時間ベースのウィンドウに追加される（「+」で示される）。イベント（1000，10，1，10.0）は、時間ベースのウィンドウに1秒間留まり2秒の標識の時点でウィンドウから削除されることになる。この時点で、時間ベースの範囲ウィンドウは、1つのイベント（1000，10，1，10.0）を含む。

30

40

【0083】

次にイベント（2000，20，2，20.0）が受信される。一実施形態において、2000 ミリ秒というタイムスタンプは、このイベントが受信された時間を表わしてもよい。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断される。イベント（1000，10，1，10.0）は満了しており時間ベースのウィンドウから削除される（「-」で示される）と判断される。イベント（2000，20，2，20.0）の満了時間は4秒（イベントのタイムスタンプ + `c 2` の値、すなわち、 $2 + 2 = 4$ 秒）と計算される。次に、このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に2秒間留まり4秒の標識の時点でウィンドウから削除されることになっている。この時点で、時間ベースの範囲ウィ

50

ンドウは、1つのイベント(2000, 20, 2, 20.0)を含む。

【0084】

次にイベント(3000, 30, 3, 30.0)が受信される。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断され、満了しているイベントはないと判断される。イベント(3000, 30, 3, 30.0)の満了時間は6秒(イベントのタイムスタンプ+c2の値、すなわち、3+3=6秒)と計算される。このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に3秒間留まり6秒の標識の時点でウィンドウから削除されることになっている。この時点で、時間ベースの範囲ウィンドウは、2つのイベント(2000, 20, 2, 20.0)および新たに追加されたイベント(3000, 30, 3, 30.0)を含む。

10

【0085】

次にイベント(4000, 40, 4, 40.0)が受信される。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断される。イベント(2000, 20, 2, 20.0)は満了しており時間ベースのウィンドウから削除される(「-」で示される)と判断される。イベント(4000, 40, 4, 40.0)の満了時間は8秒(イベントのタイムスタンプ+c2の値、すなわち、4+4=8秒)と計算される。次に、このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に4秒間留まり8秒の標識の時点でウィンドウから削除されることになっている。この時点で、時間ベースの範囲ウィンドウは、2つのイベント(3000, 30, 3, 30.0)および新たに追加されたイベント(4000, 40, 4, 40.0)を含む。

20

【0086】

次にイベント(5000, 50, 5, 50.0)が受信される。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断され、満了しているイベントはないと判断される。イベント(5000, 50, 5, 50.0)の満了時間は10秒(イベントのタイムスタンプ+c2の値、すなわち、5+5=10秒)と計算される。次に、このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に5秒間留まり10秒の標識の時点でウィンドウから削除されることになっている。この時点で、時間ベースの範囲ウィンドウは、3つのイベント(3000, 30, 3, 30.0)、(4000, 40, 4, 40.0)、および新たに追加されたイベント(5000, 50, 5, 50.0)を含む。

30

【0087】

次にイベント(6000, 60, 6, 60.0)が受信される。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断され、イベント(3000, 30, 3, 30.0)は満了しており時間ベースのウィンドウから削除される(「-」で示される)と判断される。イベント(6000, 60, 6, 60.0)の満了時間は12秒(イベントのタイムスタンプ+c2の値、すなわち、6+6=12秒)と計算される。次に、このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に6秒間留まり12秒の標識の時点でウィンドウから削除されることになっている。この時点で、時間ベースの範囲ウィンドウは、3つのイベント(4000, 40, 4, 40.0)、(5000, 50, 5, 50.0)および新たに追加されたイベント(6000, 60, 6, 60.0)を含む。

40

【0088】

次にイベント(7000, 70, 7, 70.0)が受信される。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断され、満了しているイベントはないと判断される。イベント(7000, 70, 7, 70.0)の満了時間は14秒(イベントの到着時間+c2の値、すなわち、7+7=14秒)と計算される。次に、このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に7秒間留まり14秒の標識の時点でウィンドウから削除される

50

ことになっている。この時点で、時間ベースの範囲ウィンドウは、4つのイベント(4000, 40, 4, 40.0)、(5000, 50, 5, 50.0)、(6000, 60, 6, 60.0)、および新たに追加されたイベント(7000, 70, 7, 70.0)を含む。

【0089】

次にイベント(8000, 80, 8, 80.0)が受信される。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断され、イベント(4000, 40, 4, 40.0)は満了しており時間ベースのウィンドウから削除される(「-」で示される)と判断される。イベント(8000, 80, 8, 80.0)の満了時間は16秒(イベントのタイムスタンプ+c2の値、すなわち、8+8=16秒)と計算される。次に、このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に8秒間留まり16秒の標識の時点でウィンドウから削除されることになっている。この時点で、時間ベースの範囲ウィンドウは、4つのイベント(5000, 50, 5, 50.0)、(6000, 60, 6, 60.0)、(7000, 70, 7, 70.0)、および新たに追加されたイベント(8000, 80, 8, 80.0)を含む。

10

【0090】

次にイベント(9000, 90, 9, 90.0)が受信される。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断され、満了しているイベントはないと判断される。イベント(9000, 90, 9, 90.0)の満了時間は18秒(イベントの到着時間+c2の値、すなわち、9+9=18秒)と計算される。このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に9秒間留まり18秒の標識の時点でウィンドウから削除されることになっている。この時点で、時間ベースの範囲ウィンドウは、5つのイベント(5000, 50, 5, 50.0)、(6000, 60, 6, 60.0)、(7000, 70, 7, 70.0)、(8000, 80, 8, 80.0)、および新たに追加されたイベント(9000, 90, 9, 90.0)を含む。

20

【0091】

次にイベント(10000, 100, 10, 100.0)が受信される。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断され、イベント(5000, 50, 5, 50.0)は満了しており時間ベースのウィンドウから削除される(「-」で示される)と判断される。イベント(10000, 100, 10, 100.0)の満了時間は20秒(イベントのタイムスタンプ+c2の値、すなわち、10+10=20秒)と計算される。このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に10秒間留まり20秒の標識の時点でウィンドウから削除されることになっている。この時点で、時間ベースの範囲ウィンドウは、5つのイベント(6000, 60, 6, 60.0)、(7000, 70, 7, 70.0)、(8000, 80, 8, 80.0)、(9000, 90, 9, 90.0)、および新たに追加されたイベント(10000, 100, 10, 100.0)を含む。

30

【0092】

次にイベント(11000, 110, 11, 110.0)が受信される。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断され、満了しているイベントはないと判断される。イベント(11000, 110, 11, 110.0)の満了時間は22秒(イベントのタイムスタンプ+c2の値、すなわち、11+11=22秒)と計算される。このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に11秒間留まり22秒の標識の時点でウィンドウから削除されることになっている。この時点で、時間ベースの範囲ウィンドウは、6つのイベント(6000, 60, 6, 60.0)、(7000, 70, 7, 70.0)、(8000, 80, 8, 80.0)、(9000, 90, 9, 90.0)、(10000, 100, 10, 100.0)、および新たに追加されたイベント(11000, 110, 11, 110.0)を含む。

40

50

1 0 , 1 1 , 1 1 0 . 0) を含む。

【 0 0 9 3 】

次にイベント (1 2 0 0 0 , 1 2 0 , 1 2 , 1 2 . 0) が受信される。時間ベースのウィンドウの内容が検査されて、このウィンドウ内のいずれかのイベントが満了しているか否か判断され、イベント (6 0 0 0 , 6 0 , 6 , 6 0 . 0) は満了しており時間ベースのウィンドウから削除される (「 - 」 で示される) と判断される。イベント (1 2 0 0 0 , 1 2 0 , 1 2 , 1 2 0 . 0) の満了時間は 2 4 秒 (イベントの到着時間 + c 2 の値、すなわち、 $12 + 12 = 24$ 秒) と計算される。次に、このイベントは時間ベースのウィンドウに追加される。このイベントは、ウィンドウ内に 1 2 秒間留まり 2 4 秒の標識の時点でウィンドウから削除されることになっている。この時点で、時間ベースの範囲ウィンドウは、6 つのイベント (7 0 0 0 , 7 0 , 7 , 7 0 . 0) 、 (8 0 0 0 , 8 0 , 8 , 8 0 . 0) 、 (9 0 0 0 , 9 0 , 9 , 9 0 . 0) 、 (1 0 0 0 0 , 1 0 0 , 1 0 , 1 0 0 . 0) 、 (1 1 0 0 0 , 1 1 0 , 1 1 , 1 1 0 . 0) 、および新たに追加されたイベント (1 2 0 0 0 , 1 2 0 , 1 2 , 1 2 0 . 0) を含む。

10

【 0 0 9 4 】

表 A に関して先に述べたように、入力イベントの受信時に実行される処理は、入力イベントの満了時間を決定することと、入力イベントを時間ベースのウィンドウに追加することと、ウィンドウを検査することにより、満了したイベントを同定し削除することを含む。イベントが時間ベースのウィンドウ内にある継続時間は、入力イベントの属性の関数であってもよい。この属性の値は、イベントが異なれば異なり得るので、イベントが時間ベースのウィンドウ内にある継続時間は、イベントごとに異なり得る。したがって、イベントが時間ベースのウィンドウ内に留まる資格がある継続時間は、イベントが異なれば異なり得る。

20

【 0 0 9 5 】

図 2 は、本発明の実施形態に従い、継続時間可変ウィンドウを管理するための方法を示す簡略化されたフローチャート 2 0 0 を示す。図 2 に示される処理は、1 つ以上のプロセッサによって実行されるソフトウェア (たとえばコード、命令、プログラム) で実施されてもよく、ハードウェアで実施されてもよく、またはその組み合わせにおいて実施されてもよい。ソフトウェアは、メモリに (たとえばメモリ素子に、非一時的なコンピュータ読取可能な記録媒体に) 格納されていてもよい。図 2 に示される特定の一連の処理ステップは、限定を意図したものではない。

30

【 0 0 9 6 】

ある実施形態において、図 2 に示される処理は、イベントストリームを介して入力イベントが受信される度に、連続クエリ実行の一部として実行されてもよい。2 0 2 で、イベントストリーム内の入力イベント (タプル) が受信されてもよい。このイベントは、関連付けられた時間情報 (たとえばタイムスタンプ) を有していてもよい。いくつかの例において、あるイベントに関連付けられた時間は、このイベントの受信時の時間を示すものであってもよい。

【 0 0 9 7 】

2 0 4 で、受信されたイベントについて、このイベントが時間ベースのウィンドウ内で保持されることになる継続時間が計算される。2 0 4 の一部として、C Q L コードによって時間ベースのウィンドウに対して指定された範囲パラメータが決定され継続時間を決定するのに使用される。先に説明したように、いくつかの実施形態において、連続クエリは、時間ベースのウィンドウの継続時間を、受信された入力イベントの 1 つ以上の属性の値の関数として定めてもよい。このようなシナリオにおいて、2 0 4 で継続時間を決定することは、2 0 2 で受信されたイベントの 1 つ以上の属性の値を決定することと、これら値の関数として継続時間を計算することとを含み得る。継続時間を計算するために使用される上記 1 つ以上の属性の値は、異なるイベントごとに異なり得るので、この継続時間は個々のイベントごとに異なる可能性がある。

40

【 0 0 9 8 】

50

たとえば、ある実施形態において、この継続時間は、イベントの特定の属性の値に設定されてもよい。この特定の属性の値は、204で決定されてもよく、入力イベントに対するウィンドウ継続時間は、決定された値に設定される。他のいくつかの実施形態において、連続クエリは、ウィンドウの継続時間を、1つ以上の属性値に基づいて、算術式として定めてもよい。このようなシナリオにおいて、204で、受信されたイベントの1つ以上の属性値が決定されてもよく、その後、決定された値を用いて算術式の値が求められ、それにより、受信イベントの継続時間が計算されてもよい。

【0099】

206で、受信されたイベントについて、204で決定された継続時間に基づいて満了時間が計算される。一実施形態において、

イベントの満了時間 = $t + T$

であり、「 t 」は202で受信されたイベントに関連付けられた時間（たとえばイベントの到着時間）であり、「 T 」は204で決定された継続時間である。

【0100】

208で、206で計算された満了時間は、202で受信されたイベントに関連付けられてもよい。たとえば、満了時間情報を格納しておいて、EPS102がイベントが与えられたときにこのイベントに対して計算される満了時間を容易に求めることができるようにしてもよい。

【0101】

210で、時間ベースのウィンドウが検査されて、満了したイベント（すなわち以前に受信したイベント）を含むか否か判断される。満了していると識別されたイベントについては、このイベントは210で時間ベースのウィンドウから削除または除去される。一実施形態において、時間ベースのウィンドウ内のイベントは、このイベントについて計算された満了時間が現在の時間と等しいかまたはそれよりも前であれば、満了しているとみなされる。上記のように、あるイベントの満了時間は $(t + T)$ と計算される。したがって、現在の時間 (t_p) が $(t + T)$ と同一かまたはそれよりも後であれば、イベントは満了しているとみなされる。満了したと判断されたイベントは次に、時間ベースのウィンドウから削除または除去される。

【0102】

212で、202で受信されたイベントが、時間ベースのウィンドウに挿入または追加される。214で、この処理はその後、イベントストリーム内の次のイベントを待つ。次のイベントが受信されると、処理は202から214まで繰り返される。

【0103】

図2に関して先に説明したように、受信された各イベントについて継続時間と満了時間が計算され、イベントはその後時間ベースのウィンドウに挿入される。さらに、時間ベースのウィンドウ内にあるイベントが検査されて、満了したイベントが識別され、満了したイベントはその後時間ベースのウィンドウから削除または除去される。このようにして、あるイベントが時間ベースのウィンドウ内に留まることになる継続時間を、イベントごとに計算し、この継続時間は、イベントの1つ以上の属性値に基づいていてもよい。イベントについて、継続時間に基づいて計算される満了時間は、イベントが満了し時間ベースのウィンドウから削除または除去されることになる時間を表わす。

【0104】

図3は、本発明の実施形態に従い、継続時間可変の時間ベースウィンドウ処理を実現するために使用し得るモジュールおよびデータ構造を示す。図3に示されるモジュールは、ソフトウェア、またはハードウェア、またはその組合わせにおいて実現し得る。さらに、図3に示されるモジュールおよびデータ構造は限定を意図したものではない。これに代わる実施形態は、さまざまな配置および組み合わせの、図3に示されるものよりも多いかまたは少ないモジュールを有していてもよい。

【0105】

図3に示される実施形態において、モジュールは、継続時間および満了時間評価器モジ

10

20

30

40

50

ルール 302 と、時間ベースウィンドウオペレータモジュール 304 とを含む。継続時間および満了時間評価器モジュール 302 は、新たに受信した各イベントに対し、この受信イベントの継続時間および満了時間を決定するように構成されてもよい。受信イベントはその後、このイベントに対して計算された満了時間とともに、さらなる処理のために時間ベースウィンドウオペレータ 304 に渡されてもよい。一実施形態において、継続時間および満了時間評価器 302 は、図 2 の 204、206、および 208 に示され先に説明された処理を実行するように構成されてもよい。

【0106】

時間ベースウィンドウオペレータ 304 は、時間ベースのウィンドウを管理するように構成されてもよい。これは、新たに受信したイベントを時間ベースのウィンドウに挿入することと、時間ベースのウィンドウ内のイベントがいつ満了したか判断することと、満了したイベントを時間ベースのウィンドウから削除または除去することとを含み得る。一実施形態において、時間ベースウィンドウオペレータ 304 は、210 および 212 に示され先に説明された処理を実行するように構成されてもよい。

10

【0107】

さまざまなデータ構造を用いて継続時間可変の時間ベースウィンドウを実現してもよい。一実施形態において、優先度キューが使用され、優先度はウィンドウ内のイベントに対して計算された満了時間によって指示される。新たに受信されたイベントはキューに追加され、満了したイベントはキューから削除される。キューの中にあるイベントは、それぞれに関連付けられた満了時間に基づいてソートされてもよい。一実施形態において、イベントは、満了時間がより早いイベントがキューの先頭により近く満了時間がより遅いイベントがキューの末尾により近くなるように、ソートされる。どの時点でも、優先度キューは、その時点で時間ベースのウィンドウ内にあるゼロ以上のイベントを表わすゼロ以上のイベントを含み得る。

20

【0108】

図 4 は、上記のように優先度キューを用いて継続時間可変の時間ベースウィンドウを実現する実施形態において時間ベースウィンドウオペレータが実行し得る処理を示す簡略化されたフローチャート 400 を示す。図 4 に示される処理は、1 つ以上のプロセッサによって実行されるソフトウェア（たとえばコード、命令、プログラム）で実施されてもよく、ハードウェアで実施されてもよく、またはその組み合わせにおいて実施されてもよい。ソフトウェアは、メモリに（たとえばメモリ素子に、非一時的なコンピュータ読取可能な記録媒体に）格納されていてもよい。図 4 に示される特定の一連の処理ステップは、限定を意図したものではない。

30

【0109】

402 で、新たに受信されたイベントは、関連付けられた満了時間とともに受信されてもよい。このイベントの満了時間は、継続時間および満了時間評価器 302 によって既に計算されていてもよい。

【0110】

404 で、キューは空か否か判断される。空のキューは、現在ウィンドウ内にイベントがないことを示す。404 でキューが空であると判断された場合、処理は 412 に続き、そうでなければ処理は 406 に続く。

40

【0111】

406 で、優先度キューの先頭にあるイベントがアクセスされる。優先度キューの中にあるイベント要素は常に、これらイベントに関連付けられた満了時間に基づいてソートされ、満了時間がより早いイベントはキューの先頭により近く、満了時間がより遅いイベントはキューの末尾により近いので、キューの先頭にあるイベントは、ウィンドウ内で満了時間が最も早いイベントを表わす。

【0112】

408 で、406 でアクセスされたイベントが満了しているか否か判断される。一実施形態において、アクセスされたイベントに関連付けられた満了時間が現在の時間に等しい

50

かまたはそれよりも早い場合、このイベントは満了しているとみなされる。たとえば、現在の時間が10秒の標識にあるとすると、アクセスされたイベントに関連付けられた満了時間が10秒または10秒未満であれば、このアクセスされたイベントは満了しているとみなされる。408で、アクセスされたイベントが満了していると判断された場合、410で、アクセスされたイベントは優先度キューから除去または削除され、処理は404に続く。408で、アクセスされたイベントが満了していないと判断された場合、処理は412に続く。このようにして、404、406、408、および410の処理は、キューの中の満了したイベントすべてがキューから除去されるまで、または、キューが空になるまで、繰り返される。

【0113】

404でキューは空であると判断されるかまたは406でアクセスされたイベントが満了していないと判断された場合、処理は412に続く。412では、402で受信されたイベントが、このイベントに対して計算された満了時間に基づいてキューに追加される。イベントは、キューのソートされた（すなわち満了時間に基づいてソートされた）状態が維持されるように挿入される。次に、処理は414に続き、414でEPS102はイベントストリーム内の次のイベントを待つ。イベントが来ると、処理は402に続く。

【0114】

時間ベースの区画ウィンドウ

ある実施形態において、「PARTITION BY」というCQL句によって、1つのイベントストリームを、このストリームの属性に基づいて、論理的に複数のサブストリームに分割することができる。次に、範囲（range）Tのスライディングウィンドウが、各サブストリームに対して独立して計算される。

【0115】

たとえば、ストリームSに対し、時間ベースの区画ウィンドウは次のように定義されてもよい。

【0116】

S [partition by A1,..., Ak rows N range T]

ストリームSに対する上記分割スライディングウィンドウは、イベントの正の整数Nと、ストリームの属性のサブセット{A1,..., Ak}とをパラメータとして取り、(1)属性A1,..., Akが同等であることに基づいて、論理的にSを異なるサブストリームに分割し、(2)各サブストリームに対し独立してサイズがNで範囲/継続時間がTのイベントベースのスライディングウィンドウを計算する。イベントベースのスライディングウィンドウは独立して各区画に適用される。このようなウィンドウを「行 時間範囲ウィンドウ」と呼んでもよい。なぜなら、ある区画に対するウィンドウについて、イベントは、「range（範囲）T」によって制御されるそれぞれのイベントに関連付けられた満了時間に基づいて、または、「rows（行）N」によって制御されるウィンドウ内のイベントの数に基づいて、ウィンドウから満了するからである。上記Nは行 時間範囲ウィンドウ内のイベントの最大数を定める。

【0117】

過去の実装例では、範囲/継続時間パラメータ「T」は、一定または固定されており可変ではなかった。本発明のある実施形態では、PARTITION BY句によって作成された1つの区画のための行 時間範囲ウィンドウ内に1つのイベントが留まる資格がある継続時間を指定する範囲/継続時間パラメータTを、異なるイベントごとに異なるようにすることができる。結果として、特定の入力イベントストリームに対しEPS102に登録されたクエリについて、1つの区画のためのウィンドウ内にイベントが留まる継続時間を、異なるイベントごとに異なるようにすることができる。

【0118】

たとえば、継続時間可変区画ウィンドウを有する以下のCQLクエリについて考える。

CREATE QUERY Q1 AS

SELECT *

FROM S [PARTITION BY DATA ROWS M RANGE <range_expression>]

上記CQLコードは、ストリームSを、「DATA」の値に基づいて区画すなわちサブストリームに分割する。各区画について、CQLコードは行 時間範囲ウィンドウを指定し、この行 時間範囲ウィンドウ内のイベントの最大数は「M」であり、1つのイベントがこの行 時間範囲ウィンドウ内に留まる資格がある継続時間は<range__expression>で表わされる。この例において、「range」についてのパラメータは、式「<range__expression>」として指定され、range__expressionは、変数または算術式であってもよい。ある実施形態において、<range__expression>は、イベント属性に基づくものであってもよい。他の実施形態において、<range__expression>は、イベントストリームスキーマの1つ以上の属性に基づき得る算術式であってもよい。

10

【0119】

例として以下のものについて考える。

FROM S [PARTITION BY DATA ROWS 2 RANGE RANGE_COLUMN]

この場合も、イベントストリームSは、属性「DATA」の値に基づいて区画に分割される。CQLコードは、各区画に対し、行 時間範囲ウィンドウを指定する。1つの区画のためのウィンドウ内のイベントの最大数は2であり、1つのイベントがこのウィンドウ内に留まる資格がある継続時間は、イベント属性であるRANGE__COLUMNで表わされる。特定の区画に属する1つのイベントについて、このイベントがこの区画のための行 時間範囲ウィンドウ内に留まる資格がある継続時間は、このイベントの属性RANGE__COLUMNの値と、この区画ウィンドウ内のイベントの数（最大で2）とによって制御される。したがって、属性RANGE__COLUMNの値が互いに異なるイベントは、互いに異なる継続時間ウィンドウ内に留まる資格があることになる。

20

【0120】

以下の例において、区画時間ベースの継続時間パラメータは、以下の算術式で表わされる。

【0121】

CREATE QUERY Q1 AS

SELECT * FROM S [PARTITION BY DATA ROWS 2 RANGE <range_expression>]

この場合、イベントストリームSは、属性「DATA」の値に基づいて区画に分割される。CQLコードは、各区画に対して行 時間範囲ウィンドウを指定し、1つのウィンドウ内のイベントの最大数は2であり、1つのイベントがこのウィンドウ内に留まる資格がある継続時間は、式<range__expression>で表わされる。この行 時間範囲ウィンドウは、独立して各区画に適用される。ある区画の受信イベントに対して計算される満了時間はしたがって(t+range__expression__value)であり、「range__expression__value」は、イベントの受信時に<range__expression>の値を求めることによって計算される値である。

30

【0122】

以下の例は、イベントの受信時に、異なる区画に適用される継続時間可変の時間ベース範囲ウィンドウの効果を示す。この例について、入力イベントストリームSは2つの属性c1およびc2を有するスキーマを有すると想定する。

40

【0123】

S (c1 character(2), c2 bigint)

ストリームSに対する連続CQLクエリは次のように指定してもよい。

【0124】

SELECT * from S [PARTITION BY c1 ROWS 2 RANGE c2]

上記CQLコードによると、入力ストリームSは、属性c1の値に基づいて区画に分割される。CQLコードは、各区画に対し行 時間範囲ウィンドウを指定し、1つの区画のためのウィンドウ内のイベントの最大数は、(「ROWS 2」なので)2であり、イベントがこの区画のためのウィンドウ内に留まる資格がある継続時間は、(「RANGE c

50

2」なので)このイベントの属性 c 2 の値によって決定される。

【 0 1 2 5 】

以下の表 B は、タブルの入カストリーム、ならびに、タブルの内容および入力イベントストリーム S を介してイベントを受信するときのさまざまな時点において複数の区画に適用される行 時間範囲ウィンドウの内容を示す。

【 0 1 2 6 】

【 表 2 】

入力イベントストリーム スキーマ:(タイムスタンプ, c1, c2)	行-時間範囲ウィンドウの内容 スキーマ:(タイムスタンプ, c1, c2)
(1000,a,2)	Pa: +(1000,a,2) ^{3s}
(1000,a,9)	Pa: (1000,a,2) ^{3s} , +(1000,a,9) ^{10s}
(2000,a,1)	Pa: -(1000,a,2) ^{3s} , (1000,a,9) ^{10s} , +(2000,a,1) ^{3s}
(3000,b,1)	Pa: -(2000,a,1) ^{3s} , (1000,a,9) ^{10s} , Pb: +(3000,b,1) ^{4s}
(4000,a,2)	Pa: (1000,a,9) ^{10s} , +(4000,a,2) ^{6s} ,
(4000,a,3)	Pa: -(1000,a,9) ^{10s} , (4000,a,2) ^{6s} , +(4000,a,3) ^{7s}
(4000,b,3)	Pb: -(3000,b,1) ^{4s} , +(4000,b,3) ^{7s}
(5000,b,2)	Pa: (4000,a,2) ^{6s} , (4000,a,3) ^{7s} Pb: (4000,b,3) ^{7s} , +(5000,b,2) ^{7s}
(6000,<heartbeat event>)	Pa: -(4000,a,2) ^{6s} , (4000,a,3) ^{7s} Pb: (4000,b,3) ^{7s} , (5000,b,2) ^{7s}
(7000,c,1)	Pa: -(4000,a,3) ^{7s} Pb: -(4000,b,3) ^{7s} , -(5000,b,2) ^{7s} Pc: +(7000,c,1) ^{8s}
(8000,<heartbeat event>)	Pa: Pb: Pc: -(7000,c,1) ^{8s}

表B: 区画処理を伴う継続時間可変の時間ベース範囲ウィンドウ

【 0 1 2 7 】

上記表 B において、左の列はスキーマ (タイムスタンプ, c 1, c 2) を有するストリーム S を介して受信されたイベントのストリームを示す。タイムスタンプはミリ秒 (m s e c s) で示され、1 秒 = 1 0 0 0 m s e c s である。一実施形態において、あるイベントに関連付けられたタイムスタンプは、このイベントの受信時の時間またはこのイベントの発生時の時間を表わし得る。表 B の右の列は、各区画について、さまざまな時点における、この区画のための行 時間範囲ウィンドウの内容 (すなわちイベント) を示す。イベントの前の「+」表記は、そのイベントが区画のための行 時間範囲ウィンドウに追加されたことを示すために使用される。イベントの前の「-」表記は、このイベントが、このイベントが満了したために、区画のための行 時間範囲ウィンドウから削除されたことを

示すために使用される。

【 0 1 2 8 】

一実施形態において、イベントが受信されると、このイベントのための区画が決定される。このイベントの満了時間が計算される。イベントに対して計算される満了時間は、このイベントが満了しその区画のための行 時間範囲ウィンドウから削除されることになる時間を示す。その後このイベントは、適切な区画のための行 時間範囲ウィンドウに追加される。また、各区画のための行 時間範囲ウィンドウが検査されて、この行 時間範囲ウィンドウ内のいずれかのイベントを、このイベントが満了したためまたは R O W S 条件のために、ウィンドウから削除する必要があるか否か判断される。

【 0 1 2 9 】

関連付けられたタイムスタンプが「 t 」であり時間範囲または継続時間が「 T 」であるイベントの場合、このイベントの満了時間は (t + T) である。以下のクエリの場合、範囲の値は、受信したイベントの属性 c 2 の値に基づく。

【 0 1 3 0 】

SELECT * from S [PARTITION BY c1 ROWS 2 RANGE c2]

したがって、受信したイベントについて、このイベントの満了時間は (t + c 2 の値) である。上記表 B において、各受信イベントの満了時間 (expiration time) は、このイベントの添え字として示されている。

【 0 1 3 1 】

【 数 2 】

(たとえば (タイムスタンプ, c 1, c 2) ^{Expiration-time})

【 0 1 3 2 】

表 B に示されるように、イベント (1 0 0 0 , a , 2) が受信される。さまざまな区画のための行 時間範囲ウィンドウ (すなわち各区画のためのウィンドウが出力するリレーションセット) が検査されて、いずれかのイベントをそのイベントが満了しているので行 時間範囲ウィンドウから削除する必要があるか否か判断される。表 B に示される例のために、さまざまな区画のための行 時間範囲ウィンドウは、イベント (1 0 0 0 , a , 2) の到着前は空であると想定する。このイベントの c 2 属性の値に基づいて、このイベントは、属性値「 a 」に対応する区画 (「 P a 」) のための行 時間範囲ウィンドウ内に、2 秒間留まることになっていると判断される。イベント (1 0 0 0 , a , 2) の満了時間は 3 秒標識 (このイベントのタイムスタンプ + c 2 の値、すなわち 1 + 2 = 3 秒) であると計算される。次に、このイベント (1 0 0 0 , a , 2) は、P a のための行 時間範囲ウィンドウに追加される (「 + 」で示される)。イベント (1 0 0 0 , a , 2) は、区画 P a のための行 時間範囲ウィンドウ内に 2 秒間留まる資格がある。このイベントは、3 秒標識の時点で (またはこの区画ウィンドウの「 R O W S 2 」条件のためにそれよりも早く) 行 時間範囲ウィンドウから削除されることになっている。P a のための行 時間範囲ウィンドウはその後検査されて、いずれかのイベントを「 R O W S 2 」条件のために行 時間範囲ウィンドウから削除する必要があるか否か判断される。削除する必要があるイベントはないと判断される。

【 0 1 3 3 】

関連付けられたタイムスタンプ情報が同一であるイベント (1 0 0 0 , a , 9) が受信される。このイベントの c 2 属性の値に基づいて、このイベントは P a のための行 時間範囲ウィンドウ内に 9 秒間留まることになっていると判断される。このイベントの満了時間は 1 0 秒標識 (このイベントのタイムスタンプ + c 2 の値、すなわち 1 + 9 = 1 0 秒) であると計算される。次に、このイベント (1 0 0 0 , a , 9) は、区画 P a のための行 時間範囲ウィンドウに追加される (「 + 」で示される)。イベント (1 0 0 0 , a , 9) は、区画 P a のための行 時間範囲ウィンドウ内に 9 秒間留まる資格がある。このイベントは、1 0 秒標識の時点で (またはこの区画ウィンドウの「 R O W S 2 」条件のためにそれよりも早く) 行 時間範囲ウィンドウから削除されることになっている。P a のた

10

20

30

40

50

めの行 時間範囲ウィンドウが検査されて、いずれかのイベントを、イベント(1000, a, 9)を挿入して「ROWS 2」条件を満たすようにしたために、行 時間範囲ウィンドウから削除する必要があるか否か判断される。削除する必要があるイベントはないと判断される。この時点で、区画Paのための行 時間範囲ウィンドウは、2つのイベント(1000, a, 2)および(1000, a, 9)を含む。

【0134】

次にイベント(2000, a, 1)が受信される。さまざまな区画のための行 時間範囲ウィンドウが検査されて、いずれかのイベントをそのイベントの満了のために行 時間範囲ウィンドウから削除する必要があるか否か判断される。満了したイベントはないと判断される。上記イベントのc2属性の値に基づいて、このイベントはPaのための行 時間範囲ウィンドウ内に1秒間留まることになっていると判断される。イベント(a, 1)の満了時間は3秒標識(このイベントのタイムスタンプ+c2の値、すなわち2+1=3)であると計算される。次に、このイベント(2000, a, 1)は、区画Paのための行 時間範囲ウィンドウに追加される(「+」で示される)。イベント(2000, a, 1)は区画Paのための行 時間範囲ウィンドウ内に1秒間留まる資格がある。このイベントは、3秒標識の時点で(またはこの区画ウィンドウの「ROWS 2」条件のためにそれよりも早く)行 時間範囲ウィンドウから削除されることになっている。次に、Paのための行 時間範囲ウィンドウが検査されて、いずれかのイベントを、イベント(2000, a, 1)を挿入して「ROWS 2」条件を満たすようにしたために、Paのための行 時間範囲ウィンドウから削除する必要があるか否か判断される。Paのための行 時間範囲ウィンドウの「ROWS 2」条件のために、イベント(2000, a, 1)を追加したことによってイベント(1000, a, 2)は区画Paのための行 時間範囲ウィンドウから削除されることになり、区画Paのための行 時間範囲ウィンドウ内には、2つのイベント(1000, a, 9)および(2000, a, 1)が残される。

【0135】

次にイベント(3000, b, 1)が受信される。さまざまな区画のための行 時間範囲ウィンドウが検査されて、いずれかのイベントをそのイベントの満了のために行 時間範囲ウィンドウから削除する必要があるか否か判断される。区画Paのための行 時間範囲ウィンドウ内のイベント(2000, a, 1)は満了しておりこのウィンドウから削除される(「-」で示される)と判断される。c2属性の値に基づいて、イベント(3000, b, 1)は、属性値「b」に対応する区画(「Pb」)のための行 時間範囲ウィンドウ内に、1秒間留まることになっていると判断される。イベント(3000, b, 1)の満了時間は4秒標識(このイベントのタイムスタンプ+c2の値、すなわち3+1=4秒)であると計算される。次に、このイベント(3000, b, 1)は、Pbのための行 時間範囲ウィンドウに追加される(「+」で示される)。イベント(3000, b, 1)は、区画Pbのための行 時間範囲ウィンドウ内に1秒間留まる資格がある。このイベントは、4秒標識の時点で(またはこの区画ウィンドウの「ROWS 2」条件のためにそれよりも早く)行 時間範囲ウィンドウから削除されることになっている。Pbのための行 時間範囲ウィンドウが検査されて、いずれかのイベントを、イベント(3000, b, 1)を挿入して「ROWS 2」条件を満たすようにしたために、行 時間範囲ウィンドウから削除する必要があるか否か判断される。削除する必要があるイベントはないと判断される。この時点で、区画Paのための行 時間範囲ウィンドウはイベント(1000, a, 9)を含み、区画Pbのための行 時間範囲ウィンドウはイベント(3000, b, 1)を含む。

【0136】

次に、関連付けられたタイムスタンプが同一である3つのイベント(4000, a, 2)、(4000, a, 3)および(4000, b, 3)が受信される。さまざまな区画のための行 時間範囲ウィンドウが検査されて、いずれかのイベントをそのイベントの満了のために行 時間範囲ウィンドウから削除する必要があるか否か判断される。区画Pbのための行 時間範囲ウィンドウ内のイベント(3000, b, 1)は満了しておりこのウ

インドウから削除される(「-」で示される)と判断される。c 2 属性の値に基づいて、イベント(4000, a, 2)は、Paのための行 時間範囲ウィンドウ内に2秒間留まることになっていると判断される。イベント(4000, a, 2)の満了時間は6秒標識(このイベントのタイムスタンプ+c 2の値、すなわち4+2=6秒)であると計算される。次に、このイベント(4000, a, 2)は、区画Paのための行 時間範囲ウィンドウに追加される(「+」で示される)。イベント(a, 2)は、区画Paのための行 時間範囲ウィンドウ内に2秒間留まる資格があり、6秒標識の時点で(またはこの区画ウィンドウの「ROWS 2」条件のためにそれよりも早く)このウィンドウから削除されることになっている。Paのための行 時間範囲ウィンドウが検査されて、いずれかのイベントを、イベント(4000, a, 2)を挿入して「ROWS 2」条件を満たすようにしたために、行 時間範囲ウィンドウから削除する必要があるか否か判断される。削除する必要があるイベントはないと判断される。

10

【0137】

イベント(4000, a, 3)について、c 2 属性の値に基づいて、イベント(4000, a, 3)は、Paのための行 時間範囲ウィンドウ内に3秒間留まることになっていると判断される。イベント(4000, a, 3)の満了時間は7秒標識(このイベントのタイムスタンプ+c 2の値、すなわち4+3=7秒)であると計算される。次に、このイベント(4000, a, 3)は、区画Paのための行 時間範囲ウィンドウに追加される(「+」で示される)。イベント(4000, a, 3)は、区画Paのための行 時間範囲ウィンドウ内に3秒間留まる資格があり、7秒標識の時点で(またはこの区画ウィンドウの「ROWS 2」条件のためにそれよりも早く)このウィンドウから削除されることになっている。次に、Paのための行 時間範囲ウィンドウが検査されて、いずれかのイベントを、イベント(4000, a, 3)を挿入して「ROWS 2」条件を満たすようにしたために、Paのための行 時間範囲ウィンドウから削除する必要があるか否か判断される。イベント(4000, a, 3)を追加したことによってイベント(1000, a, 9)は区画Paのための行 時間範囲ウィンドウから削除されることになる(「-」で示される)。

20

【0138】

イベント(4000, b, 3)について、c 2 属性の値に基づいて、イベント(4000, b, 3)は、区画Pbのための行 時間範囲ウィンドウ内に3秒間留まることになっていると判断される。イベント(4000, b, 3)の満了時間は7秒標識(このイベントのタイムスタンプ+c 2の値、すなわち4+3=7秒)であると計算される。次に、このイベント(4000, b, 3)は、区画Pbのための行 時間範囲ウィンドウに追加される(「+」で示される)。イベント(b, 3)は、区画Pbのための行 時間範囲ウィンドウ内に3秒間留まる資格があり、7秒標識の時点で(またはこの区画ウィンドウの「ROWS 2」条件のためにそれよりも早く)このウィンドウから削除されることになっている。Pbのための行 時間範囲ウィンドウが検査されて、いずれかのイベントを、イベント(4000, b, 3)を挿入して「ROWS 2」条件を満たすようにしたために、この行 時間範囲ウィンドウから削除する必要があるか否か判断される。削除する必要があるイベントはないと判断される。この時点で、区画Paのための行 時間範囲ウィンドウはイベント(4000, a, 2)および(4000, a, 3)を含み、区画Pbのためのウィンドウはイベント(4000, b, 3)を含む。

30

40

【0139】

次にイベント(5000, b, 2)が受信される。さまざまな区画のための行 時間範囲ウィンドウが検査されて、いずれかのイベントをそのイベントの満了のために行 時間範囲ウィンドウから削除する必要があるか否か判断される。満了したイベントはないと判断される。c 2 属性の値に基づいて、このイベント(5000, b, 2)は区画Pbのための行 時間範囲ウィンドウ内に2秒間留まることになっていると判断される。イベント(5000, b, 2)の満了時間は7秒標識(このイベントのタイムスタンプ+c 2の値、すなわち5+2=7)であると計算される。次に、イベント(5000, b, 2)は、

50

区画 P b のための行 時間範囲ウィンドウに追加される（「+」で示される）。イベント（5000, b, 2）は区画 P b のための行 時間範囲ウィンドウ内に 2 秒間留まる資格があり、7 秒標識の時点で（またはこの区画ウィンドウの「ROWS 2」条件のためにそれよりも早く）このウィンドウから削除されることになっている。区画 P b のための行 時間範囲ウィンドウが検査されて、いずれかのイベントを、イベント（50000, b, 2）を挿入して「ROWS 2」条件を満たすようにしたために、行 時間範囲ウィンドウから削除する必要があるか否か判断される。削除する必要があるイベントはないと判断される。この時点で、区画 P a のための行 時間範囲ウィンドウはイベント（4000, a, 2）および（4000, a, 3）を含み、区画 P b のための行 時間範囲ウィンドウはイベント（4000, b, 3）および（5000, b, 2）を含む。

10

【0140】

次に、関連付けられたタイムスタンプが 6000 ミリ秒であるハートビートイベントが受信される。一実施形態において、ハートビートイベントは、単に時間の経過情報を伝えるために使用される特殊な入力イベントである。ハートビートイベントには、他の属性の値はない。イベント処理システムを、システムのユーザによって構成できる周期的な間隔でハートビートを自動的に送信するように構成してもよい。そうすれば、ハートビートイベントは、その継続時間内に他の実際のイベントが受信されなければ（すなわちイベントストリームはその継続時間サイレント状態である）、自動的に生成され周期的な間隔で送信される。このハートビートイベントが受信されると、ウィンドウに関連する処理が実行される。したがって、ハートビートイベント（6000, <heartbeats__event>）が受信されたとき、さまざまな区画のための行 時間範囲ウィンドウが検査されて、いずれかのイベントをそのイベントが満了しているで行 時間範囲ウィンドウから削除する必要があるか否か判断される。区画 P a のための行 時間範囲ウィンドウ内のイベント（4000, a, 2）は満了しておりこのウィンドウから削除される（「-」で示される）と判断される。この時点で、区画 P a のための行 時間範囲ウィンドウはイベント（4000, a, 3）を含み、区画 P b のための行 時間範囲ウィンドウはイベント（4000, b, 3）および（5000, b, 2）を含む。

20

【0141】

次にイベント（7000, c, 1）が受信される。さまざまな区画のための行 時間範囲ウィンドウが検査されて、いずれかのイベントをそのイベントの満了のために行 時間範囲ウィンドウから削除する必要があるか否か判断される。区画 P a のための行 時間範囲ウィンドウ内のイベント（4000, a, 3）ならびに区画 P b のための行 時間範囲ウィンドウ内のイベント（4000, b, 3）および（5000, b, 2）は満了しておりウィンドウから削除される（「-」で示される）と判断される。c 2 属性の値に基づいて、イベント（7000, c, 1）は、属性値「c」に対応する区画（「Pc」）のための行 時間範囲ウィンドウ内に、1 秒間留まることになっていると判断される。イベント（7000, c, 1）の満了時間は 8 秒標識（このイベントのタイムスタンプ + c 2 の値、すなわち $7 + 1 = 8$ 秒）であると計算される。次に、このイベント（7000, c, 1）は、属性値「c」に対応する区画 Pc のための行 時間範囲ウィンドウに追加される（「+」で示される）。イベント（7000, c, 1）は、区画 Pc のための行 時間範囲ウィンドウ内に 1 秒間留まる資格があり、8 秒標識の時点で（またはこの区画ウィンドウの「ROWS 2」条件のためにそれよりも早く）ウィンドウから削除されることになっている。区画 Pc のための行 時間範囲ウィンドウが検査されて、いずれかのイベントを、イベント（7000, c, 1）を挿入して「ROWS 2」条件を満たすようにしたために、行 時間範囲ウィンドウから削除する必要があるか否か判断される。削除する必要があるイベントはないと判断される。この時点で、区画 P a および P b のための行 時間範囲ウィンドウは空であり区画 Pc のための行 時間範囲ウィンドウはイベント（7000, c, 1）を含む。

30

40

【0142】

すべてのシステムがハートビートイベント送信機構を有する必要はないことに注意しな

50

ければならない。ハートビートイベントを送信しないシステムにおいて、ウィンドウに関連する処理は、イベントストリームを介して通常の入力イベントが受信されると実行される。たとえば、表 B に示されるイベントストリームの場合、ハートビートイベント送信機能がないシステムでは、6000 ミリ秒標識および 7000 ミリ秒標識の時点で実行される処理は、新たなイベント (7000, c, 1) が受信されたときに 7000 ミリ秒標識の時点で実行されてもよい。

【0143】

次に、ハートビートイベント (8000, <heartbeat_event>) が受信される。さまざまな区画のための行 時間範囲ウィンドウが検査されて、いずれかのイベントをそのイベントの満了のために行 時間範囲ウィンドウから削除する必要があるか
10
否が判断される。区画 P c のための行 時間範囲ウィンドウ内のイベント (7000, c, 1) は満了しておりウィンドウから削除される (「-」で示される) と判断される。この時点で、区画 P a、P b、および P c のための行 時間範囲ウィンドウは空である。

【0144】

表 B に関して先に説明したように、入力イベントの受信時に実行される処理は、この入力イベントの満了時間を決定することと、このイベントが属する区画を決定することと、この入力イベントをこの区画のための行 時間範囲ウィンドウに追加することと、さまざま
20
な区画のための行 時間範囲ウィンドウを検査して、満了したイベント、または、R O W S 条件の結果として区画ウィンドウから除去する必要があるイベントを識別し削除する。ある区画のための行 時間範囲ウィンドウ内に 1 つのイベントが留まる継続時間は、入力イベントの属性の関数であってもよく、これは異なる入力イベントごとに異なり得る。したがって、イベントがある区画のための行 時間範囲ウィンドウ内に留まる資格がある継続時間は、異なるイベントごとに異なり得る。

【0145】

図 5 は、本発明の実施形態に従い、区画のための継続時間可変ウィンドウを管理するための方法を示す簡略化されたフローチャート 500 を示す。図 5 に示される処理は、1 つ以上のプロセッサによって実行されるソフトウェア (たとえばコード、命令、プログラム) で実施されてもよく、ハードウェアで実施されてもよく、またはその組み合わせにおいて実施されてもよい。ソフトウェアは、メモリに (たとえばメモリ素子に、非一時的なコン
30
ピュータ読取可能な記録媒体に) 格納されていてもよい。図 5 に示される特定の一連の処理ステップは、限定を意図したものではない。

【0146】

ある実施形態において、図 5 に示される処理は、イベントストリームを介して入力イベント (実際のイベントまたはハートビートイベントであり得る) が受信される度に、連続クエリ実行の一部として実行されてもよい。502 で、イベントストリーム内の入力イベント (タプル) が受信されてもよい。このイベントは、関連付けられた時間情報 (たとえばタイムスタンプ) を有していてもよい。いくつかの例において、あるイベントに関連付けられた時間は、このイベントの受信時の時間を示すものであってもよい。

【0147】

504 で、受信されたイベントについて、このイベントが区画のための行 時間範囲ウィンドウ内に留まる資格がある継続時間が計算される。504 の一部として、C Q L コードによって行 時間範囲ウィンドウに対して指定された範囲パラメータが決定され継続時間を決定するのに使用される。先に説明したように、いくつかの実施形態において、連続クエリは、この継続時間を、受信された入力イベントの 1 つ以上の属性の値の関数として定めてもよい。このようなシナリオにおいて、504 で継続時間を決定することは、502 で受信されたイベントの 1 つ以上の属性の値を決定することと、このイベントについての継続時間をこれら値の関数として計算することとを含み得る。継続時間の計算に使用される 1 つ以上の属性の値は、異なるイベントごとに異なり得るので、この継続時間は個々のイベントごとに異なる可能性がある。

【0148】

10

20

30

40

50

たとえば、ある実施形態において、この継続時間は、イベントの特定の属性の値に設定されてもよい。この特定の属性の値は、504で決定されてもよく、入力イベントに対する行 時間範囲ウィンドウは、決定された値に設定される。他のいくつかの実施形態において、連続クエリは、行 時間範囲ウィンドウの継続時間を、1つ以上の属性値に基づいて算術式として定めてもよい。このようなシナリオにおいて、504で、受信されたイベントの1つ以上の属性値が決定されてもよく、その後、決定された値を用いて算術式の値が求められ、それにより、受信されたイベントの継続時間が計算されてもよい。

【0149】

506で、受信されたイベントについて、504で決定された継続時間に基づいて満了時間が計算される。一実施形態において、

10

イベントの満了時間 = $t + T$

であり、「 t 」は502で受信されたイベントに関連付けられた時間（たとえばイベントの到着時間）であり、「 T 」は504で決定された継続時間である。

【0150】

508で、506で計算された満了時間は、502で受信されたイベントに関連付けられてもよい。たとえば、満了時間情報を格納しておいて、EPS102がイベントを与えられたときにこのイベントについて計算される満了時間を容易に求めることができるようにしてもよい。

【0151】

510で、既存の各区画について、この区画のための行 時間範囲ウィンドウが検査されて、満了したイベントが識別され削除される。満了していると識別されたイベントについては、このイベントはその区画の行 時間範囲ウィンドウから削除または除去される。一実施形態において、行 時間範囲ウィンドウ内のイベントは、このイベントについて計算された満了時間が現在の時間と等しいかまたはそれよりも前であれば、満了しているとみなされる。上記のように、あるイベントの満了時間は $(t + T)$ と計算される。したがって、現在の時間 (t_p) が $(t + T)$ と同一かまたはそれよりも後であれば、イベントは、満了しているとみなされる。

20

【0152】

512で、502で受信された入力イベントについて区画が決定される。このイベントが属する区画は、このイベントの属性の値に応じて決まるものであってもよい。したがって、512で、受信された入力イベントについて、入力イベントストリームを区画に分割するために使用された属性の値が検査され、この値に基づいて、イベントが属する区画が求められる。

30

【0153】

514で、502で受信されたイベントが、512で決定された区画の行 時間範囲ウィンドウに挿入または追加される。

【0154】

上記のように、以下の句

S [partition by A_1, \dots, A_k rows N range T]

の場合、区画からのイベントの除去は、このイベントの満了時間に基づいて、または、行 時間範囲ウィンドウ内のイベントの総数に基づいて制御される。たとえば、「... rows N ...」は、最大「 N 」個のイベントを1つの区画のための行 時間範囲ウィンドウ内に入れることが可能であることを意味し、新たなイベントが受信されてこのウィンドウに追加されたときに、「rows N 」という条件を満たすためにこの区画内の最も早いイベントが行 時間範囲ウィンドウから削除される。したがって、516では、512で決定された区画のための行 時間範囲ウィンドウが検査されて、「rows N 」条件を満たすためにイベントが識別され削除される。いくつかの実施形態において、516の処理は、514の処理よりも前に実行されてもよく、または、これに代えて、514および516で実行される処理は一緒に実行されてもよい。

40

【0155】

50

518で、処理はその後イベントストリーム内の次のイベントを待つ。次のイベントが受信されると、処理は502から繰り返される。

【0156】

図3に示され先に説明されたさまざまなモジュールは、本発明の実施形態に従う区画ベースで継続時間可変の時間ベースウィンドウ処理においても使用し得る。一実施形態において、継続時間および満了時間評価器モジュール302を、図5の502、504、506、および508の処理を実行するように適応させてもよい。時間ベースウィンドウオペレータモジュール304は、上記さまざまな機能に加えて、場合によっては複数の区画行時間範囲ウィンドウを管理するように適応させてもよい。たとえば、時間ベースウィンドウオペレータ304は、図5の510、512、514、および516に従う処理を実行するように構成されてもよい。

10

【0157】

さまざまなデータ構造を用いてさまざまな区画のための行時間範囲ウィンドウを実現してもよい。一実施形態において、一組の優先度キューが使用されてもよく、各優先度キューは、特定の区画のための行時間範囲ウィンドウに対応する。ある区画のための行時間範囲ウィンドウを表わす優先度キューにおいて、優先度は行時間範囲ウィンドウ内のイベントに対して計算された満了時間によって指示されてもよい。ある区画に属する、新たに受信されたイベントは、この区画のための優先度キューに追加される。ある区画について、満了したイベントおよび/または「row N」条件のために区画の行時間範囲ウィンドウから除去されるイベントは、対応する優先度キューから削除される。

20

【0158】

ある区画のためのキューの中において、このキューの中にあるイベントは、それぞれに関連付けられた満了時間に基づいてソートされてもよい。一実施形態において、イベントは、満了時間がより早いイベントがキューの先頭により近く満了時間がより遅いイベントがキューの末尾により近くなるように、ソートされる。どの時点でも、ある区画のための優先度キューは、その時点でその区画のためのウィンドウ内にあるゼロ以上のイベントを表わすゼロ以上のイベントを含み得る。

【0159】

図6は、本発明の実施形態に従い、優先度キューを用いて区画のための継続時間可変ウィンドウを扱うために時間ベースウィンドウオペレータ304が実行し得る処理を示す簡略化されたフローチャート600を示す。図6に示される処理は、1つ以上のプロセッサによって実行されるソフトウェア（たとえばコード、命令、プログラム）で実施されてもよく、ハードウェアで実施されてもよく、またはその組み合わせにおいて実施されてもよい。ソフトウェアは、メモリに（たとえばメモリ素子に、非一時的なコンピュータ読取可能な記録媒体に）格納されていてもよい。図6に示される特定の一連の処理ステップは、限定を意図したものではない。

30

【0160】

602で、新たに受信された入力イベントは、関連付けられた満了時間とともに受信されてもよい。このイベントの満了時間は、継続時間および満了時間評価器302によって既に計算されていてもよい。

40

【0161】

604、606、608、610、614、および612による処理は、既存の各区画に対して実行される。604で、未処理の区画のための行時間範囲ウィンドウに対応する優先度キューがアクセスされる。606で、このキューが空か否か判断される。空のキューは、この区画の行時間範囲ウィンドウ内に現在イベントがないことを示す。606でキューが空であると判断された場合、処理は614に続き、614で、次の未処理の区画に対する処理が604から再び開始される。606でキューが空でないと判断された場合、処理は608に続く。

【0162】

608で、処理されている区画のための優先度キューの先頭にあるイベントがアクセス

50

される。この区画のための優先度キューの中にあるイベント要素は常に、これらイベントに関連付けられた満了時間に基づいてソートされ、満了時間がより早いイベントはキューの先頭により近く、満了時間がより遅いイベントはキューの末尾により近いので、キューの先頭にあるイベントは、その区画のウィンドウ内で満了時間が最も早いイベントを表わす。

【 0 1 6 3 】

6 1 0 で、6 0 8 でアクセスされたイベントが満了しているか否か判断される。一実施形態において、アクセスされたイベントに関連付けられた満了時間が現在の時間に等しいかまたはそれよりも早い場合、このイベントは満了しているとみなされる。たとえば、現在の時間が 1 0 秒の標識にあるとすると、アクセスされたイベントに関連付けられた満了時間が 1 0 秒または 1 0 秒未満であれば、このアクセスされたイベントは満了しているとみなされる。6 1 0 で、アクセスされたイベントが満了していると判断された場合、6 1 2 で、アクセスされたイベントはこの区画のための優先度キューから除去または削除され、処理は 6 0 6 に続く。6 1 0 で、アクセスされたイベントが満了していないと判断された場合、処理は 6 1 4 に続く。このようにして、6 0 4、6 0 6、6 0 8、6 1 0、6 1 2、および 6 1 4 の処理は、さまざまな区画のための行 時間範囲ウィンドウから、満了したイベントすべてが削除されるまで、繰り返される。その後処理は 6 1 6 に続く。

10

【 0 1 6 4 】

6 1 6 で、6 0 2 で受信されたイベントに対し区画が決定される。このイベントのための区画は、入力されたイベントの属性の値に基づいて決定されてもよい。

20

【 0 1 6 5 】

6 1 8 で、6 1 6 で決定された区画のための行 時間範囲ウィンドウに対応する優先度キューがアクセスされる。6 2 0 で、6 1 8 でアクセスされた優先度キューにイベントが挿入または追加される。一実施形態において、キューの要素がこのキューの中のイベントに対して計算された満了時間に基づいてソートされた状態を保つように、イベントはこのイベントのために計算された満了時間に基づいて挿入されてもよい。

【 0 1 6 6 】

ある実施形態において、ハッシュ関数を用いて、6 1 6 で決定された特定の区画に対応するキューがアクセスされてもよい。入力イベントストリームを区画に分割するために使用された属性の値が入力としてハッシュ関数に与えられてもよく、そうすると、ハッシュ関数はその区画のための行 時間範囲に対応するキューに対するリファレンス（たとえばポインタ）を返す。

30

【 0 1 6 7 】

6 2 2 で、6 1 8 でアクセスされたキューが処理されて、「rows N」条件を満たすためにキューのイベントが識別されキューから削除される。6 2 4 で、処理は、イベントストリーム内の次のイベントを待つ。次のイベントが受信されると、処理は 6 0 2 から繰り返される。

【 0 1 6 8 】

上記のやり方で、1つのイベントが時間ベースのウィンドウ内に留まる資格がある継続時間がイベントごとに異なるようにすることができる。さらにある実施形態において、あるイベントの継続時間は、このイベント自体の1つ以上の属性に基づくものでありかつこの1つ以上の属性の関数であってもよい。あるイベントが時間ベースのウィンドウ（通常の時間ベースのウィンドウまたは区画のための行 時間範囲ウィンドウのいずれか）内に留まる資格がある継続時間は、イベントごとに異なるようにすることができ、したがってイベント固有のものである。

40

【 0 1 6 9 】

イベントがそれぞれ異なる継続時間ベースのウィンドウ内に留まるという状況は、数通りある。1つの状況は、2つのイベントが同時に受信された場合（たとえば関連付けられたタイムスタンプが同一）、これらのイベントが時間ベースのウィンドウ内にある継続時間は互いに異なることがある。その結果、これら2つのイベントの満了時間も互いに異な

50

ることがある。たとえば、あるイベントストリームに対して、時間ベースのウィンドウの継続時間がこのストリームのためのスキーマ内の属性「a t t r」の値の関数である、連続クエリが定められる場合がある。2つのイベントのうち第1のイベント(e_1)が時間 t_1 で受信され第2のイベント(e_2)も時間 t_1 で受信される場合について考える。さらに、 e_1 の属性「a t t r」の値が a_1 であり e_2 の属性「a t t r」の値が a_2 であり a_2 は a_1 と同一でないと仮定する。そうすると次のようになる。

【0170】

e_1 のための時間ベースのウィンドウの継続時間 = a_1

e_1 の満了時間 = $t_1 + a_1$

e_2 のための時間ベースのウィンドウの継続時間 = a_2

e_2 の満了時間 = $t_1 + a_2$

a_1 と a_2 は同一でないので、 e_1 の満了時間（すなわち $t_1 + a_1$ ）は、 e_2 の満了時間（すなわち $t_1 + a_2$ ）と同一ではない。このため、これら2つのイベントは同時に受信されているが（すなわち関連付けられた時間情報が同一）、2つのイベントは異なる2つの時間に満了し、したがってウィンドウ内に留まる継続時間が互いに異なる。これは、同時に受信した2つのイベントは常に同時にウィンドウから満了する、以前の時間ベースウィンドウの実装例では不可能であった。

【0171】

継続時間可変の時間ベースウィンドウのもう1つの状況として、先に受信されたイベントに関連付けられた満了時間が後に受信されたイベントに関連付けられた満了時間よりも遅い、すなわち言い換えると、後に受信されたイベントが先に受信されたイベントよりも早くウィンドウから満了する可能性がある。たとえば、上記例において、第1のイベント(e_1)が時間 t_1 に受信され第2のイベント(e_2)が t_1 よりも後の t_2 で受信されたとする。さらに、 e_1 の属性「a t t r」の値が a_1 であり e_2 の属性「a t t r」の値が a_2 であると仮定する。そうすると次のようになる。

【0172】

e_1 のための時間ベースのウィンドウの継続時間 = a_1

e_1 の満了時間 = $t_1 + a_1$

e_2 のための時間ベースのウィンドウの継続時間 = a_2

e_2 の満了時間 = $t_2 + a_2$

e_2 の満了時間（すなわち $t_2 + a_2$ ）が e_1 の満了時間（すなわち $t_1 + a_1$ ）よりも前、すなわち（ $t_2 + a_2$ ）が（ $t_1 + a_1$ ）よりも前である可能性がある。したがって、イベント e_2 は後で到着したのに e_1 よりも前に満了する。これは、以前の時間ベースウィンドウの実装例では不可能であった。以前の実装例では、先に受信されたイベントは、常に後に受信されたイベントよりも前に満了し、その理由はこれら2つのイベントのウィンドウ継続時間が同一であるからである。

【0173】

継続時間可変の時間ベースウィンドウによって、ユーザは、あるイベントがウィンドウ内にある継続時間を制御することができ、この継続時間はイベントごとに異なるようにすることができる。これは、異なるさまざまな応用例において有益である。たとえば、製品関連情報に関するイベントのストリームを処理するように構成された応用例を考える。各製品の有効期間がそれぞれ異なるということが起こり得る（たとえば生鮮食品の有効期間は非生鮮食品の有効期間よりも短い）。製品の有効期間に基づいて時間ベースのウィンドウの継続時間を変える必要があるアプリケーションでは、ストリームのためのスキーマは、有効期間が属性としてスキーマに加えられるように指定されてもよい。そうすると、あるイベントのための時間ベースのウィンドウは、有効期間という属性によって特定される値に基づくことになるであろう。継続時間可変の時間ベースウィンドウを利用することができる他の応用例は、例として、関連付けられているアラームタイマがそれぞれ異なるタスクのためのイベントに関連する応用例、完了日がそれぞれ異なるタスクのためのイベントに関連する応用例などであるが、これらに限定されない。

【 0 1 7 4 】

上記のように、あるイベントについてのウィンドウ継続時間は、このイベント自体の1つ以上の属性に基づくものであってもよい。ある実施形態において、継続時間の基になる属性は、入力イベントストリームスキーマ自体の一部であってもよい。しかしながら、いくつかの実施形態において、このような属性は受信された入力ストリームの一部でなくともよい。このような一実施形態において、ウィンドウの継続時間を決定するために使用される1つ以上の属性は、元のストリームから新たなストリームを導き出すことによってイベントストリームに追加されてもよく、継続時間計算の基になる1つ以上の属性は、導き出されたイベントストリームのスキーマの一部である。

【 0 1 7 5 】

図7は、本発明の実施形態に従い使用し得るシステム環境700の構成要素を示す簡略化されたブロック図である。図に示されているように、システム環境700は、1つ以上のクライアント計算装置702、704、706、708を含み、これら装置は、ウェブブラウザ、専用のクライアント（たとえばOracle Forms）等のクライアントアプリケーションを操作するように構成されている。さまざまな実施形態において、クライアント計算装置702、704、706、および708は、イベント処理システム712と情報をやり取りし得る。

【 0 1 7 6 】

クライアント計算装置702、704、706、708は、汎用のパーソナルコンピュータ（例として、さまざまなバージョンのMicrosoft Windows（登録商標）および/またはApple Macintosh（登録商標）オペレーティングシステム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）、携帯電話もしくはPDA（Microsoft Windows Mobile等のソフトウェアを実行し、インターネット、電子メール、SMS、ブラックベリー（Blackberry）（登録商標）、またはその他の通信プロトコルを使用可能）、および/またはさまざまな市販のUNIX（登録商標）もしくはUNIXのようなオペレーティングシステム（さまざまなGNU/Linux（登録商標）オペレーティングシステムを含むがこれに限定されない）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ってもよい。これに代えて、クライアント計算装置702、704、706、および708は、ネットワーク（たとえば以下で説明するネットワーク710）を通して通信可能なシンクライアントコンピュータ、インターネットを使用可能なゲーム機、および/またはパーソナルメッセージングデバイス等のその他の電子デバイスであってもよい。典型的なシステム環境700が4つのクライアント計算装置とともに示されているが、クライアント計算装置の数はいくつであってもそれらをサポートし得る。センサ等を有する装置のような他の装置が、システム712と情報をやり取りしてもよい。

【 0 1 7 7 】

ネットワーク710は、クライアント702、704、706、および708とイベント処理システム712との間でのデータの通信および交換を容易にし得る。ネットワーク710は、TCP/IP、SNA、IPX、AppleTalk等を含むがこれらに限定されないさまざまな市販のプロトコルのいずれかを用いたデータ通信をサポートできる、当業者によく知られているネットワークであれば、どのような種類のネットワークであってもよい。単なる例として、ネットワーク710は、イーサネット（登録商標）ネットワーク、トークンリングネットワーク等のローカルエリアネットワーク（LAN）、ワイドエリアネットワーク、バーチャルプライベートネットワーク（VPN）を含むがこれに限定されないバーチャルネットワーク、インターネット、イントラネット、エクストラネット、公衆電話交換回線網（PSTN）、赤外線ネットワーク、ワイヤレスネットワーク（たとえば、IEEE 802.1Xプロトコル群のいずれかの下で動作するネットワーク、当該技術分野において公知のブルートゥース（登録商標）プロトコル、および/またはその他のワイヤレスプロトコル）、および/または、これらおよび/または他のネットワークの任意の組み合わせであってもよい。

【 0 1 7 8 】

イベント処理システム 712 は 1 つ以上のコンピュータおよび / またはサーバを含み得る。これは、汎用コンピュータ、専用のサーバコンピュータ（例として、PCサーバ、UNIXサーバ、ミッドレンジサーバ、メインフレームコンピュータ、ラックマウント式（rack-mounted）のサーバ等を含む）、サーバファーム、サーバクラスタ、またはその他の適切な配置および / または組み合わせであってもよい。さまざまな実施形態において、システム 712 は、本願のこれまでの開示において説明された 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されていてもよい。

【0179】

システム 712 は、上述したオペレーティングシステムのうちのいずれかおよび任意の市販のサーバオペレーティングシステムを含むオペレーティングシステムを実行してもよい。また、システム 712 は、HTTPサーバ、FTPサーバ、CGIサーバ、Javaサーバ、およびデータベースサーバ等を含むさまざまなその他のサーバアプリケーション、および / またはミッドティア（mid-tier）アプリケーションのうちのいずれかを実行してもよい。典型的なデータベースサーバは、Oracle、Microsoft、Sybase、IBM等から市販されているものを含むが、これらに限定されない。

【0180】

また、システム環境 700 は 1 つ以上のデータベース 714 および 716 を含み得る。データベース 714 および 716 はさまざまな場所に存在し得る。例として、データベース 714 および 716 のうちの 1 つ以上は、システム 712 に対してローカルな（および / またはシステム 712 内に常駐している）記録媒体に存在していてもよい。これに代えて、データベース 714 および 716 は、システム 712 から離れていてネットワークベースまたは専用回線を介してシステム 712 と通信してもよい。一組の実施形態において、データベース 714 および 716 は、当業者によく知られているストレージエリアネットワーク（SAN）に存在していてもよい。同様に、システム 712 による機能を実行するために必要なファイルは、必要に応じて、システム 712 に対してローカルにおよび / または遠隔で格納されてもよい。一組の実施形態において、データベース 714 および 716 は、SQLフォーマットのコマンドに回答してデータを格納し、更新し、取出すようにされている、Oracle 11g等のリレーショナルデータベースを含み得る。

【0181】

図 8 は、本発明のある実施形態に従い使用し得るコンピュータシステム 800 の簡略化されたブロック図である。たとえば、システム 800 を用いて図 1 に示されるイベント処理システム 100 を実現してもよい。バス 824 を介して電氣的に接続し得るさまざまな構成要素を含むコンピュータシステム 800 が示されている。この構成要素は、1 つ以上の処理部 802 と、入力サブシステム 804 と、出力サブシステム 806 と、記憶装置 808 と、コンピュータ読取可能な記録媒体 810 に接続されたコンピュータ読取可能記録媒体リーダー 812 と、通信サブシステム 814 と、処理加速サブシステム 816 と、ワーキングメモリ 818 とを含み得る。

【0182】

バスサブシステム 824 は、コンピュータシステム 800 のさまざまな構成要素およびサブシステムを意図された通りに相互通信させるためのメカニズムを提供する。バスサブシステム 824 は概略的に 1 本のバスとして示されているが、これに代わるバスサブシステムの実施形態は複数のバスを利用してもよい。

【0183】

入力サブシステム 804 は、マウス、キーボード、ポインティングデバイス、タッチパッド等といった 1 つ以上の入力装置を含み得る。一般的に、入力サブシステム 804 は、コンピュータシステム 800 に情報を入力するための装置またはメカニズムを含み得る。

【0184】

出力サブシステム 806 は、コンピュータシステム 800 から情報を出力するための 1 つ以上の出力装置を含み得る。出力装置の例は、表示装置、プリンタ、投影装置等を含むがこれらに限定されない。一般的に、出力サブシステム 806 は、コンピュータシステム

10

20

30

40

50

８００から情報を出力するための装置またはメカニズムを含み得る。

【０１８５】

処理部８０２は、１つ以上のプロセッサ、プロセッサの１つ以上のコア、およびその組み合わせ等を含み得る。いくつかの実施形態において、処理部８０２は、汎用一次プロセッサと、グラフィックプロセッサ、デジタル信号プロセッサ等といった１つ以上の専用コプロセッサとを含み得る。いくつかの実施形態において、処理部８０２のうちのいくつかまたはすべてを、特定用途向け集積回路（ＡＳＩＣ）またはフィールドプログラマブルゲートアレイ（ＦＰＧＡ）のようなカスタマイズされた回路を用いて実現することができる。いくつかの実施形態において、このような集積回路は、回路そのものに格納された命令を実行する。他の実施形態において、処理部８０２は、ワーキングメモリ８１８にまたは記憶装置８０８に格納された命令を実行してもよい。さまざまな実施形態において、処理部８０２は、多様なプログラムまたはコード命令を実行することができ、現在実行している複数のプログラムまたはプロセスを管理することができる。どの時点でも、これから実行するプログラムコードのうちのいくつかまたはすべては、システムワーキングメモリ８１８、記憶装置８０８、および／またはコンピュータ読取可能な記録媒体８１０に常駐していてもよい。適切なプログラミングによって、処理部８０２は、イベントストリームに関連する処理を実行するための上記さまざまな機能を提供することができる。いくつかの実施形態において、コンピュータシステム８００はまた、デジタル信号プロセッサ（ＤＳＰ）、専用プロセッサ、および／またはそれと同様のものを含み得る処理加速部８１６を含んでいてもよい。

【０１８６】

記憶装置８０８は、ディスクドライブ、光学式記憶装置等のメモリデバイス、ならびに、プログラム可能で、一瞬で更新可能で（flash-updatable）、および／またはそれと同じようなことが可能なランダムアクセスメモリ（ＲＡＭ）および／または読出専用メモリ（ＲＯＭ）等のソリッドステート記憶装置を含み得る。処理部８０２によって実行されると上記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、記憶装置８０８に格納されてもよい。記憶装置８０８も、本発明の実施形態に従い使用されるデータを格納するためのリポジトリを提供し得る。

【０１８７】

コンピュータ読取可能な記録媒体リーダ８１２はさらに、コンピュータ読取可能な記録媒体８１０に接続されてもよい。コンピュータ読取可能な記録媒体８１０はともに、（かつ任意に記憶装置８０８と組合わせて）リモート式、ローカル式、固定式および／または着脱交換式の記憶装置、および、一時的および／またはより永続的にコンピュータ読取可能な情報を収容するための記録媒体を総合的に意味している。

【０１８８】

通信サブシステム８１４によって、ネットワーク７１０および／またはシステム環境７００に関連する上述のその他のコンピュータと、データをやり取りすることが可能であってもよい。通信サブシステム８１４は、コンピュータシステム８００からの他のシステムからデータを受信しこのシステムにデータを送信するためのインターフェースの役割を果たす。通信は、有線または無線プロトコルを用いて提供し得る。たとえば、通信サブシステム８１４によって、コンピュータ８００がインターネットを介してクライアント装置と接続できるようにしてもよい。通信サブシステム８１４は、モデム、ネットワークカード（無線または有線）、赤外線通信装置、ＧＰＳ受信機等を含み得る。

【０１８９】

ワーキングメモリサブシステム８１８は、プログラム実行中の命令およびデータの格納のためのメインランダムアクセスメモリ（ＲＡＭ）と、固定命令が格納される読出専用メモリ（ＲＯＭ）とを含み得る。オペレーティングシステム８２０および／またはアプリケーションプログラム（クライアントアプリケーション、ウェブブラウザ、ミッドティアアプリケーション、ＲＤＢＭＳ等であってもよい）等のその他のコード８２２は、ワーキングメモリ８１８に格納されていてもよい。典型的な実施形態において、ワーキングメモリ

８１８は、上記のようにイベントを処理し継続時間可変のウィンドウ処理を可能にするために使用される、実行可能なコードおよび関連するデータ構造（キャッシュ等）を含み得る。

【０１９０】

コンピュータシステム８００の代替実施形態は、数多くの変形を伴う、上記構成要素よりも多いまたは少ない構成要素を有し得ることが、理解されるはずである。たとえば、カスタマイズされたハードウェアを使用してもよく、および／または特定の要素を、ハードウェア、ソフトウェア（アプレット等のポータブルソフトウェアを含む）、またはこれら双方で実現してもよい。さらに、ネットワーク入出力装置等の他の計算装置との接続を用いてもよい。

10

【０１９１】

図９は、本発明のある実施形態に従い使用し得る計算装置９００の簡略化されたブロック図である。計算装置９００のブロックを、ハードウェア、ソフトウェア、またはハードウェアとソフトウェアの組み合わせにおいて実現することによって本発明の原理を実行してもよい。図９に示されるブロックを組み合わせるまたはサブブロックに分離することによって上記本発明の原理を実行してもよいことが、当業者には理解される。したがって、本明細書での説明は、本明細書に記載の機能ブロックの可能な組み合わせまたは分離またはさらに他の定義を裏付け得るものである。

【０１９２】

図９に示されるように、示されている計算装置９００は、第１の決定部９０２および第２の決定部９０３を含む。任意で、計算装置９００は、受信部９０１、第１の除去部９０４、および第２の除去部９０５も含むことができる。

20

【０１９３】

一実施形態において、受信部９０１は、イベントストリームを介して第１のイベントおよび第２のイベントを受信するように構成し得る。第１の決定部９０２は、イベントストリームを介して受信した第１のイベントの第１の継続時間を決定するように構成し得る。第１の継続時間は、イベントストリームに対して指定された時間ベースのウィンドウ内に第１のイベントが留まる資格がある継続時間を示す。第２の決定部９０３は、イベントストリームを介して受信した第２のイベントの第２の継続時間を決定するように構成し得る。第２の継続時間は、時間ベースのウィンドウ内に第２のイベントが留まる資格がある継続時間を示す。第２の継続時間は第１の継続時間と異なってもよい。

30

【０１９４】

一実施形態において、第１の除去部９０４は、第１の時間イベントが時間ベースのウィンドウ内に第１の継続時間あった後に時間ベースのウィンドウから第１のイベントを除去するように構成し得る。第２の除去部９０５は、第２の時間イベントが時間ベースのウィンドウ内に第２の継続時間あった後に時間ベースのウィンドウから第２のイベントを除去するように構成し得る。

【０１９５】

一実施形態において、第１のイベントは関連付けられた第１の時間を有し、第２のイベントは関連付けられた第２の時間を有し、第１の時間は第２の時間と同一である。

40

【０１９６】

一実施形態において、第１のイベントは関連付けられた第１の時間を有し、第２のイベントは関連付けられた第２の時間を有し、第１の時間は第２の時間と異なる。

【０１９７】

一実施形態において、第１の決定部９０２は、第１の継続時間を決定するときに第１のイベントの１つ以上の属性の１つ以上の値に基づいて第１の継続時間を計算するように構成し得る。第２の決定部９０３は、第２の継続時間を決定するときに第２のイベントの１つ以上の属性の１つ以上の値に基づいて第２の継続時間を計算するように構成し得る。

【０１９８】

一実施形態において、第１の決定部９０２は、第１のイベントの第１の属性の値に基づ

50

いて、第 1 のイベントのための第 1 の区画を決定するように構成し得る。第 2 の決定部 903 は、第 2 のイベントの第 1 の属性の値に基づいて、第 2 のイベントが第 1 の区画に属すると決定するように構成し得る。この時間ベースのウィンドウは第 1 の区画のためのものである。

【0199】

一実施形態において、第 1 の決定部 902 は、第 1 のイベントに関連付けられた第 1 の時間と第 1 の継続時間とに基づいて、第 1 のイベントの第 1 の満了時間を決定するように構成し得る。第 2 の決定部 903 は、第 2 のイベントに関連付けられ第 2 の時間と第 2 の継続時間とに基づいて、第 2 のイベントの第 2 の満了時間を決定するように構成し得る。第 1 の除去部 904 は、第 1 の満了時間の到来時または到来後に時間ベースのウィンドウから第 1 のイベントを除去するように構成し得る。第 2 の除去部 905 は、第 2 の満了時間の到来時または到来後に時間ベースのウィンドウから第 2 のイベントを除去するように構成し得る。

10

【0200】

本発明の特定の実施形態について説明してきたが、さまざまな修正、変更、代替構成、および均等物も、本発明の範囲に包含される。本発明の実施形態は、ある特定のデータ処理環境内での動作に限定されるものではなく、複数のデータ処理環境内で自在に機能する。加えて、本発明の実施形態について、特定の一連のトランザクションおよびステップを用いて説明してきたが、本発明の範囲が、記載されている一連のトランザクションおよびステップに限定されないことは、当業者には明らかなはずである。

20

【0201】

さらに、本発明の実施形態について、ハードウェアとソフトウェアの特定の組み合わせを用いて説明してきたが、ハードウェアとソフトウェアの他の組み合わせも本発明の範囲内であることが認識されるはずである。本発明の実施形態は、ハードウェアのみで実現されてもよく、ソフトウェアのみで実現されてもよく、これらの組み合わせを用いて実現されてもよい。本明細書に記載のさまざまなプロセスは、同一のプロセッサ上で実現されてもよく、または任意に組合わされた異なるプロセッサ上で実現されてもよい。したがって、構成要素またはモジュールが、特定の動作を実行するように構成されているという記載があった場合、このような構成は、たとえば、電子回路をこの動作を実行するように設計することによって、または、プログラム可能な電子回路（マイクロプロセッサ等）をこの動作を実行するようにプログラムすることによって、または、これらを任意に組合わせて実現することができる。プロセスは従来のプロセス間通信の技術を含むがこれに限定されないさまざまな技術を用いて通信することができ、異なるプロセス対は異なる技術を使用してもよく、または、同一のプロセス対が異なる技術を異なる時間に使用してもよい。

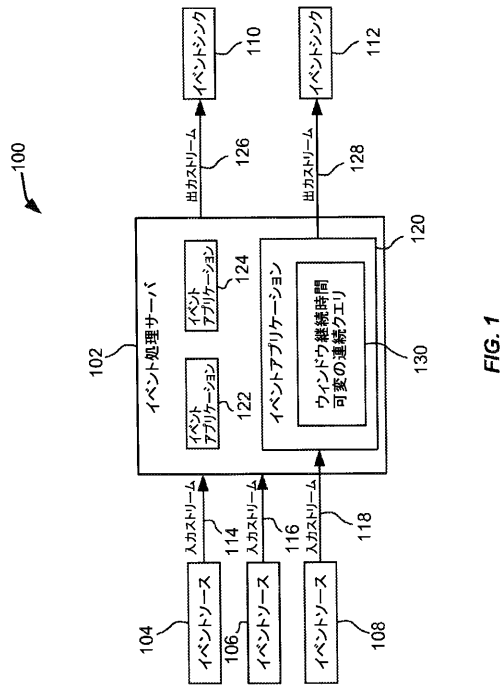
30

【0202】

明細書および図面はしたがって、限定的な意味ではなく例示的な意味で考慮されねばならない。しかしながら、これに対し、追加、控除、削除、およびその他の修正および変更を、請求項に記載のより広い精神および範囲から逸脱することなく行ない得ることは明らかであろう。このように、具体的な発明の実施形態について説明してきたが、これは限定を意図したものではない。さまざまな変形および均等物は以下の請求項の範囲に含まれる。

40

【図 1】



【図 2】

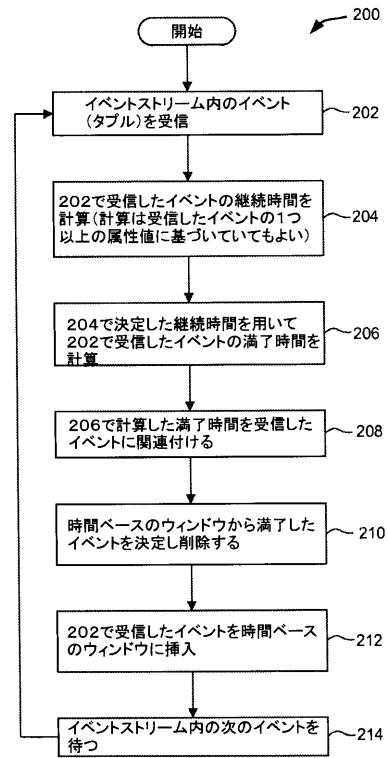


FIG. 2

【図 3】

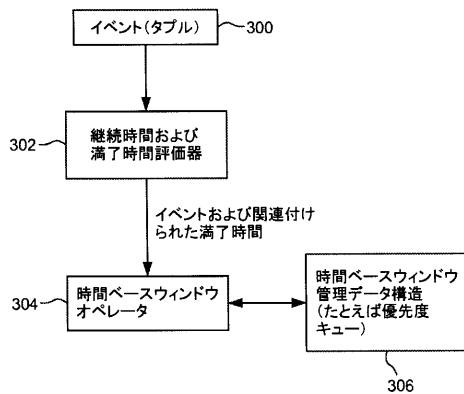


FIG. 3

【図 4】

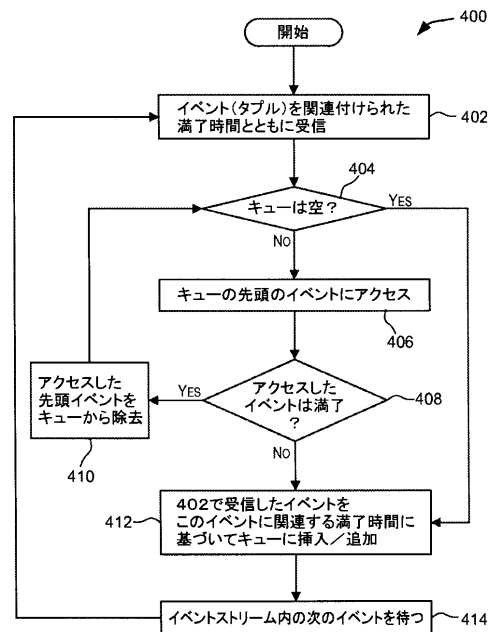


FIG. 4

【図 5】

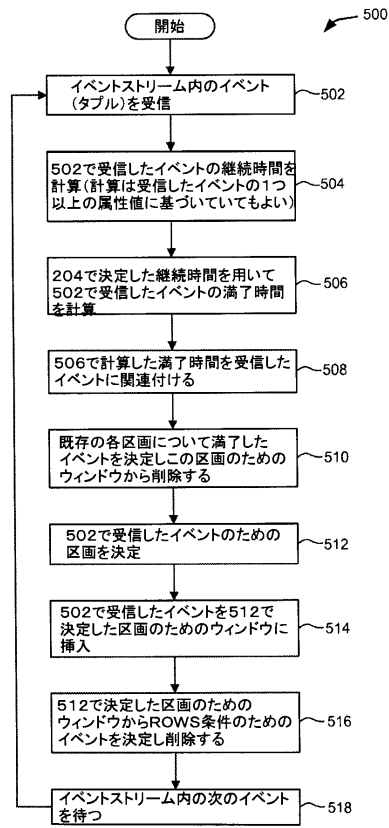


FIG. 5

【図 6】

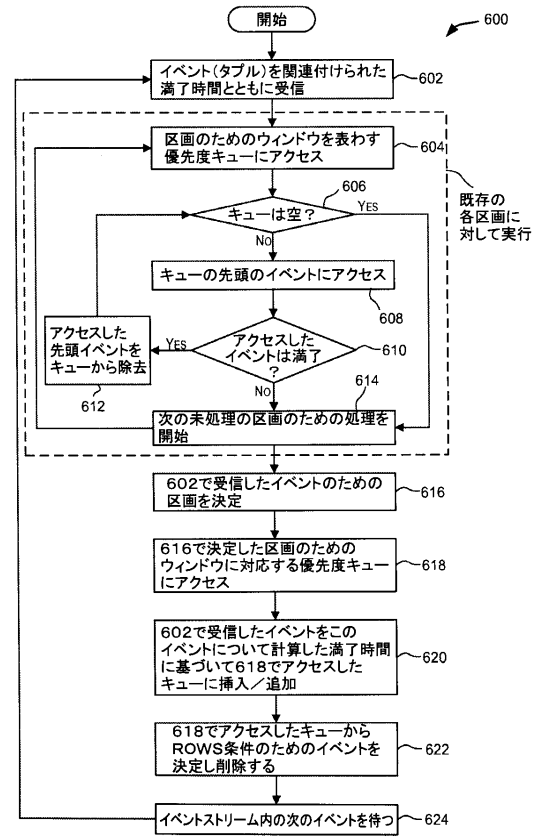


FIG. 6

【図 7】

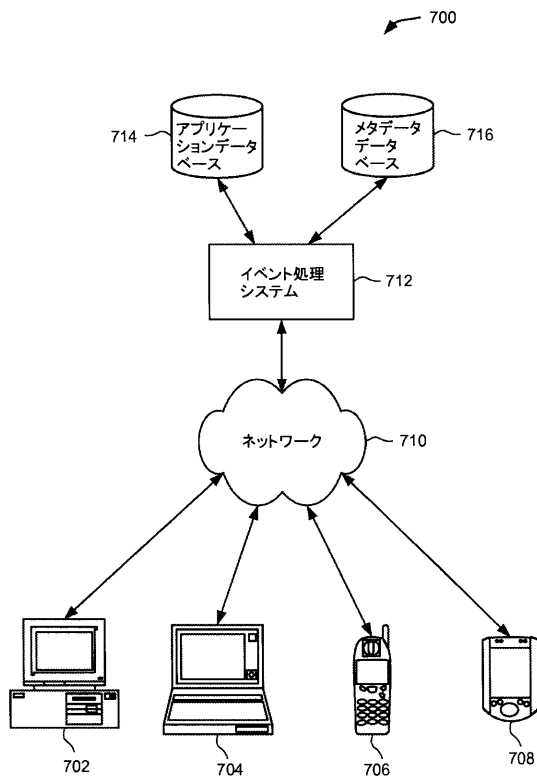


FIG. 7

【図 8】

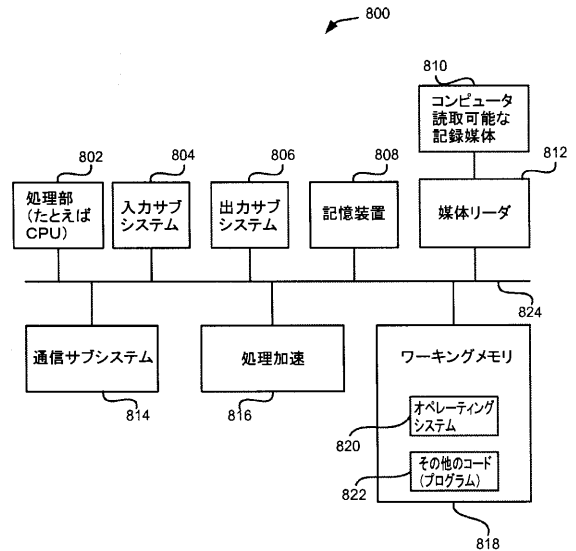
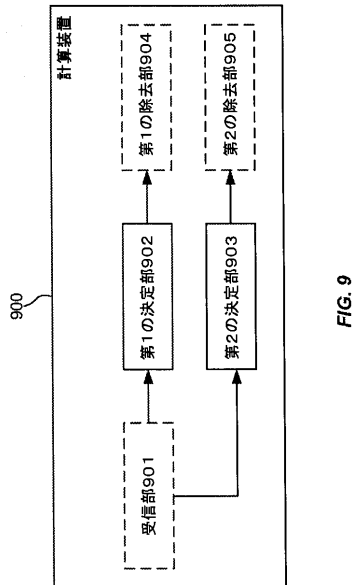


FIG. 8

【 図 9 】



【 手続補正書 】

【 提出日 】平成27年4月23日 (2015.4.23)

【 手続補正 1 】

【 補正対象書類名 】特許請求の範囲

【 補正対象項目名 】全文

【 補正方法 】変更

【 補正の内容 】

【 特許請求の範囲 】

【 請求項 1 】

方法であって、

イベントストリームを介して受信した第1のイベントの第1の継続時間を計算装置によって決定することを含み、前記第1の継続時間は、前記イベントストリームに対して指定された時間ベースのウィンドウ内に前記第1のイベントが留まる資格がある第1の時間量を示し、前記時間ベースのウィンドウは前記イベントストリームに対して指定されたクエリによってクエリされる資格がある前記イベントストリームの一組のイベントを特定し、

前記イベントストリームを介して受信した第2のイベントの第2の継続時間を前記計算装置によって決定することを含み、前記第2の継続時間は、前記時間ベースのウィンドウ内に前記第2のイベントが留まる資格がある第2の時間量を示し、前記第2の時間量は前記第1の時間量と異なり、

前記クエリが実行されるときに前記時間ベースのウィンドウ内にある前記イベントストリームを介して受信した一組のイベントに対する前記クエリを前記計算装置によって繰返し実行することを含む、方法。

【 請求項 2 】

前記第1のイベントが前記時間ベースのウィンドウ内に前記第1の時間量の間留まった後に、前記第1のイベントを前記時間ベースのウィンドウから前記計算装置によって除去

することと、

前記第 2 のイベントが前記時間ベースのウィンドウ内に前記第 2 の時間量の間留まった後に、前記第 2 のイベントを前記時間ベースのウィンドウから前記計算装置によって除去することとをさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記第 1 のイベントは関連付けられた第 1 の時間を有し、
前記第 2 のイベントは関連付けられた第 2 の時間を有し、
前記第 1 の時間は前記第 2 の時間と同一である、請求項 1 または 2 に記載の方法。

【請求項 4】

前記第 1 のイベントは関連付けられた第 1 の時間を有し、
前記第 2 のイベントは関連付けられた第 2 の時間を有し、
前記第 1 の時間は前記第 2 の時間と異なる、請求項 1 または 2 に記載の方法。

【請求項 5】

前記第 1 の継続時間を決定することは、前記第 1 のイベントの 1 つ以上の属性の 1 つ以上の値に基づいており、
前記第 2 の継続時間を決定することは、前記第 2 のイベントの 1 つ以上の属性の 1 つ以上の値に基づいている、請求項 1 ~ 4 のいずれか一項に記載の方法。

【請求項 6】

前記第 1 のイベントの第 1 の属性の値に基づいて前記第 1 のイベントのための第 1 の区画を決定することと、
前記第 2 のイベントの第 1 の属性の値に基づいて前記第 2 のイベントが前記第 1 の区画に属すると決定することとをさらに含み、
前記時間ベースのウィンドウは前記第 1 の区画のためのものである、請求項 1 ~ 5 のいずれか一項に記載の方法。

【請求項 7】

前記第 1 のイベントに関連付けられた第 1 の時間と前記第 1 の継続時間とに基づいて前記第 1 のイベントの第 1 の満了時間を前記計算装置によって決定することと、
前記第 2 のイベントに関連付けられた第 2 の時間と前記第 2 の継続時間とに基づいて前記第 2 のイベントの第 2 の満了時間を前記計算装置によって決定することと、
前記第 1 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 1 のイベントを除去することと、
前記第 2 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 2 のイベントを除去することとをさらに含む、請求項 1 ~ 6 のいずれか一項に記載の方法。

【請求項 8】

計算装置であって、
メモリと、
一組の処理部とを備え、前記一組の処理部のうちの 1 つ以上の処理部は、
イベントストリームを介して受信した第 1 のイベントの第 1 の継続時間を決定するように適合され、前記第 1 の継続時間は、前記イベントストリームに対して指定された時間ベースのウィンドウ内に前記第 1 のイベントが留まる資格がある第 1 の時間量を示し、前記時間ベースのウィンドウは前記イベントストリームに対して指定されたクエリによってクエリされる資格がある前記イベントストリームの一組のイベントを特定し、
前記イベントストリームを介して受信した第 2 のイベントの第 2 の継続時間を決定するように適合され、前記第 2 の継続時間は、前記時間ベースのウィンドウ内に前記第 2 のイベントが留まる資格がある第 2 の時間量を示し、前記第 2 の時間量は前記第 1 の時間量と異なっており、

前記クエリが実行されるときに前記時間ベースのウィンドウ内にある前記イベントストリームを介して受信した一組のイベントに対する前記クエリを繰返し実行するように適合される、計算装置。

【請求項 9】

前記一組の処理部のうちの1つ以上の処理部は、

前記第1のイベントが前記時間ベースのウィンドウ内に前記第1の時間量の間留まった後に、前記第1のイベントを前記時間ベースのウィンドウから除去するように適合される、

前記第2のイベントが前記時間ベースのウィンドウ内に前記第2の時間量の間留まった後に、前記第2のイベントを前記時間ベースのウィンドウから除去するように適合される、請求項8に記載の計算装置。

【請求項10】

前記第1のイベントは関連付けられた第1の時間を有し、

前記第2のイベントは関連付けられた第2の時間を有し、

前記第1の時間は前記第2の時間と同一である、請求項8または9に記載の計算装置。

【請求項11】

前記第1のイベントは関連付けられた第1の時間を有し、

前記第2のイベントは関連付けられた第2の時間を有し、

前記第1の時間は前記第2の時間と異なる、請求項8または9に記載の計算装置。

【請求項12】

前記一組の処理部のうちの1つ以上の処理部は、

前記第1のイベントの1つ以上の属性の1つ以上の値に基づいて前記第1の継続時間を決定するように適合され、

前記第2のイベントの1つ以上の属性の1つ以上の値に基づいて前記第2の継続時間を決定するように適合される、請求項8～11のいずれか一項に記載の計算装置。

【請求項13】

前記一組の処理部のうちの1つ以上の処理部は、

前記第1のイベントの第1の属性の値に基づいて前記第1のイベントのための第1の区画を決定するように適合され、

前記第2のイベントの第1の属性の値に基づいて前記第2のイベントが前記第1の区画に属すると決定するように適合され、

前記時間ベースのウィンドウは前記第1の区画のためのものである、請求項8～12のいずれか一項に記載の計算装置。

【請求項14】

前記一組の処理部のうちの1つ以上の処理部は、

前記第1のイベントに関連付けられた第1の時間と前記第1の継続時間とに基づいて前記第1のイベントの第1の満了時間を決定するように適合され、

前記第2のイベントに関連付けられた第2の時間と前記第2の継続時間とに基づいて前記第2のイベントの第2の満了時間を決定するように適合され、

前記第1の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第1のイベントを除去するように適合され、

前記第2の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第2のイベントを除去するように適合される、請求項8～13のいずれか一項に記載の計算装置。

【請求項15】

1つ以上の処理部によって実行可能な複数の命令を格納するコンピュータ読取可能なメモリであって、前記複数の命令は、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、イベントストリームを介して受信した第1のイベントの第1の継続時間を決定させる命令を含み、前記第1の継続時間は、前記イベントストリームに対して指定された時間ベースのウィンドウ内に前記第1のイベントが留まる資格がある第1の時間量を示し、前記時間ベースのウィンドウは前記イベントストリームに対して指定されたクエリによってクエリされる資格がある前記イベントストリームの一組のイベントを特定し、

前記1つ以上の処理部のうちの少なくとも1つの処理部に、前記イベントストリームを

介して受信した第 2 のイベントの第 2 の継続時間を決定させる命令を含み、前記第 2 の継続時間は、前記時間ベースのウィンドウ内に前記第 2 のイベントが留まる資格がある第 2 の時間量を示し、前記第 2 の時間量は前記第 1 の時間量と異なっており、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記クエリが実行されるときに前記時間ベースのウィンドウ内にある前記イベントストリームを介して受信した一組のイベントに対する前記クエリを繰返し実行させる命令を含む、コンピュータ読取可能なメモリ。

【請求項 16】

前記複数の命令は、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 1 のイベントが前記時間ベースのウィンドウ内に前記第 1 の時間量の間留まった後に、前記第 1 のイベントを前記時間ベースのウィンドウから除去させる命令と、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 2 のイベントが前記時間ベースのウィンドウ内に前記第 2 の時間量の間留まった後に、前記第 2 のイベントを前記時間ベースのウィンドウから除去させる命令とを含む、請求項 15 に記載のコンピュータ読取可能なメモリ。

【請求項 17】

前記第 1 のイベントは関連付けられた第 1 の時間を有し、

前記第 2 のイベントは関連付けられた第 2 の時間を有し、

前記第 1 の時間は前記第 2 の時間と同一である、請求項 15 または 16 に記載のコンピュータ読取可能なメモリ。

【請求項 18】

前記第 1 のイベントは関連付けられた第 1 の時間を有し、

前記第 2 のイベントは関連付けられた第 2 の時間を有し、

前記第 1 の時間は前記第 2 の時間と異なる、請求項 15 または 16 に記載のコンピュータ読取可能なメモリ。

【請求項 19】

前記複数の命令は、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 1 のイベントの 1 つ以上の属性の 1 つ以上の値に基づいて前記第 1 の継続時間を決定させる命令と、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 2 のイベントの 1 つ以上の属性の 1 つ以上の値に基づいて前記第 2 の継続時間を決定させる命令とを含む、請求項 15 ~ 18 のいずれか一項に記載のコンピュータ読取可能なメモリ。

【請求項 20】

前記複数の命令は、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 1 のイベントの第 1 の属性の値に基づいて前記第 1 のイベントのための第 1 の区画を決定させる命令と、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 2 のイベントの第 1 の属性の値に基づいて前記第 2 のイベントが前記第 1 の区画に属すると決定させる命令とを含み、

前記時間ベースのウィンドウは前記第 1 の区画のためのものである、請求項 15 ~ 19 のいずれか一項に記載のコンピュータ読取可能なメモリ。

【請求項 21】

前記複数の命令は、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 1 のイベントに関連付けられた第 1 の時間と前記第 1 の継続時間とに基づいて前記第 1 のイベントの第 1 の満了時間を決定させる命令と、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 2 のイベントに関連付けられた第 2 の時間と前記第 2 の継続時間とに基づいて前記第 2 のイベントの第 2 の満了時間を決定させる命令と、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 1 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 1 のイベントを除去させる命令と、

前記 1 つ以上の処理部のうちの少なくとも 1 つの処理部に、前記第 2 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 2 のイベントを除去させる命令とを含む、請求項 15 ~ 20 のいずれか一項に記載のコンピュータ読取可能なメモリ。

【請求項 22】

システムであって、

イベントストリームを介して受信した第 1 のイベントの第 1 の継続時間を決定するための手段を備え、前記第 1 の継続時間は、前記イベントストリームに対して指定された時間ベースのウィンドウ内に前記第 1 のイベントが留まる資格がある第 1 の時間量を示し、前記時間ベースのウィンドウは前記イベントストリームに対して指定されたクエリによってクエリされる資格がある前記イベントストリームの一組のイベントを特定し、

前記イベントストリームを介して受信した第 2 のイベントの第 2 の継続時間を決定するための手段を備え、前記第 2 の継続時間は、前記時間ベースのウィンドウ内に前記第 2 のイベントが留まる資格がある第 2 の時間量を示し、前記第 2 の時間量は前記第 1 の時間量と異なっており、

前記クエリが実行されるときに前記時間ベースのウィンドウ内にある前記イベントストリームを介して受信した一組のイベントに対する前記クエリを繰返し実行するための手段を備える、システム。

【請求項 23】

前記第 1 のイベントが前記時間ベースのウィンドウ内に前記第 1 の時間量の間留まった後に、前記第 1 のイベントを前記時間ベースのウィンドウから除去するための手段と、

前記第 2 のイベントが前記時間ベースのウィンドウ内に前記第 2 の時間量の間留まった後に、前記第 2 のイベントを前記時間ベースのウィンドウから除去するための手段とをさらに備える、請求項 22 に記載のシステム。

【請求項 24】

前記第 1 のイベントは関連付けられた第 1 の時間を有し、

前記第 2 のイベントは関連付けられた第 2 の時間を有し、

前記第 1 の時間は前記第 2 の時間と同一である、請求項 22 または 23 に記載のシステム。

【請求項 25】

前記第 1 のイベントは関連付けられた第 1 の時間を有し、

前記第 2 のイベントは関連付けられた第 2 の時間を有し、

前記第 1 の時間は前記第 2 の時間と異なる、請求項 22 または 23 に記載のシステム。

【請求項 26】

前記第 1 の継続時間を決定するための手段は、前記第 1 のイベントの 1 つ以上の属性の 1 つ以上の値に基づいて前記第 1 の継続時間を計算するための手段を含み、

前記第 2 の継続時間を決定するための手段は、前記第 2 のイベントの 1 つ以上の属性の 1 つ以上の値に基づいて前記第 2 の継続時間を計算するための手段を含む、請求項 22 ~ 25 のいずれか一項に記載のシステム。

【請求項 27】

前記第 1 のイベントの第 1 の属性の値に基づいて前記第 1 のイベントのための第 1 の区画を決定するための手段と、

前記第 2 のイベントの第 1 の属性の値に基づいて前記第 2 のイベントが前記第 1 の区画に属すると決定するための手段とをさらに備え、

前記時間ベースのウィンドウは前記第 1 の区画のためのものである、請求項 22 ~ 26 のいずれか一項に記載のシステム。

【請求項 28】

前記第 1 のイベントに関連付けられた第 1 の時間と前記第 1 の継続時間とに基づいて前

記第 1 のイベントの第 1 の満了時間を決定するための手段と、

前記第 2 のイベントに関連付けられた第 2 の時間と前記第 2 の継続時間とに基づいて前記第 2 のイベントの第 2 の満了時間を決定するための手段と、

前記第 1 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 1 のイベントを除去するための手段と、

前記第 2 の満了時間の到来時または到来後に前記時間ベースのウィンドウから前記第 2 のイベントを除去するための手段とをさらに備える、請求項 22 ~ 27 のいずれか一項に記載のシステム。

【請求項 29】

前記第 1 の継続時間を決定することは、前記イベントストリームに対して指定されたクエリに基づいており、前記第 2 の継続時間を決定することは、前記イベントストリームに対して指定されたクエリに基づいている、請求項 1 ~ 7 のいずれか一項に記載の方法。

【請求項 30】

前記クエリは、前記イベントストリームを介して受信した各イベントについて継続時間をどのように計算すべきか示す式を指定し、前記第 1 の継続時間を決定することは前記式の値を求めることを含み、前記第 2 の継続時間を決定することは前記式の値を求めることを含む、請求項 29 に記載の方法。

【請求項 31】

前記式は、前記イベントストリームを介して受信した各イベントの 1 つ以上の属性の 1 つ以上の値に基づいている、請求項 30 に記載の方法。

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/010920

A. CLASSIFICATION OF SUBJECT MATTER INV. H04L12/24 ADD.		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) H04L		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EP0-Internal, WPI Data		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2009/313198 A1 (KUDO YUTAKA [US] ET AL) 17 December 2009 (2009-12-17) figures 1,6,7,11,14 paragraph [0033] paragraph [0053] paragraph [0054] - paragraph [0055]	1-28
X A	US 2012/278473 A1 (GRIFFITHS KURT EVAN [US]) 1 November 2012 (2012-11-01) paragraph [0041] figure 10	1,8,15, 22 2-7, 9-14, 16-21, 23-28
A	WO 2007/122347 A1 (FRANCE TELECOM [FR]; DOUSSON CHRISTOPHE [FR]; LE MAIGAT PIERRE [FR]) 1 November 2007 (2007-11-01) figures 4a-e, 5a-c page 13, line 3 - page 27, line 19	1-28
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another claim or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
4 December 2014		15/12/2014
Name and mailing address of the ISA/ European Patent Office, P.B. 5618 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer
		Nold, Michael

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2014/010920

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009313198 A1	17-12-2009	EP 2291742 A1	09-03-2011
		EP 2341434 A1	06-07-2011
		JP 5102366 B2	19-12-2012
		JP 5356586 B2	04-12-2013
		JP 2011518359 A	23-06-2011
		JP 2012256355 A	27-12-2012
		US 2009313198 A1	17-12-2009
		US 2012117573 A1	10-05-2012
		US 2014025621 A1	23-01-2014
		US 2014229419 A1	14-08-2014
		WO 2009153901 A1	23-12-2009

US 2012278473 A1	01-11-2012	US 2012278473 A1	01-11-2012
		US 2014280852 A1	18-09-2014

WO 2007122347 A1	01-11-2007	NONE	

フロントページの続き

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

(72)発明者 スリニバサン, アナンド

アメリカ合衆国、 9 4 0 6 5 カリフォルニア州、 レッドウッド・シティー、 オラクル・パークウェイ、 5 0 0

Fターム(参考) 5B061 FF06