

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6250061号
(P6250061)

(45) 発行日 平成29年12月20日 (2017.12.20)

(24) 登録日 平成29年12月1日 (2017.12.1)

(51) Int. Cl.	F I
G06F 17/30 (2006.01)	G06F 17/30 220Z
G06F 12/00 (2006.01)	G06F 17/30 415
	G06F 12/00 513D
	G06F 12/00 547Z

請求項の数 9 (全 22 頁)

(21) 出願番号	特願2015-545815 (P2015-545815)	(73) 特許権者	502303739
(86) (22) 出願日	平成25年12月4日 (2013.12.4)		オラクル・インターナショナル・コーポレーション
(65) 公表番号	特表2016-504679 (P2016-504679A)		アメリカ合衆国カリフォルニア州94065レッドウッド・シティー, オラクル・パークウェイ500
(43) 公表日	平成28年2月12日 (2016.2.12)	(74) 代理人	110001195
(86) 国際出願番号	PCT/US2013/073086		特許業務法人深見特許事務所
(87) 国際公開番号	W02014/089190	(72) 発明者	デ・カストロ・アルベス, アレクサンダー
(87) 国際公開日	平成26年6月12日 (2014.6.12)		アメリカ合衆国、94065-1677
審査請求日	平成28年12月2日 (2016.12.2)		カリフォルニア州、レッドウッド・ショアーズ, オラクル・パークウェイ、500、エム/エス・5・オー・ピー・7
(31) 優先権主張番号	61/733,844	審査官	山本 俊介
(32) 優先日	平成24年12月5日 (2012.12.5)		最終頁に続く
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	61/806,744		
(32) 優先日	平成25年3月29日 (2013.3.29)		
(33) 優先権主張国	米国 (US)		

(54) 【発明の名称】 イベント処理とマップリデュースとの統合

(57) 【特許請求の範囲】

【請求項1】

第1の時間の時点で永続的データストアにすでに格納されているイベントデータに関して第1のクエリを実行する第1のバッチ演算を開始することと、

前記第1のバッチ演算の実行中、データストリームから連続的に受信されているイベントデータに関して第2のクエリを連続的に実行することと、

前記第1のバッチ演算の実行中、前記第2のクエリの結果を、完了した第2のバッチ演算の結果と連続的にマージすることと、

マージの結果を連続的に出力することと、

前記第1のバッチ演算の完了に回答して、前記第2のバッチ演算の結果を前記第1のバッチ演算の結果と置換することを含む、コンピュータによって実現される方法。

10

【請求項2】

前記第1のバッチ演算の実行中、前記永続的データストアに、前記データストリームから連続的に受信されている前記イベントデータを連続的に格納することと、

前記第1のバッチ演算の完了に回答して、前記第1の時間よりも遅い第2の時間の時点で前記永続的データストアにすでに格納されているイベントデータに関して前記第1のクエリを実行する第3のバッチ演算を開始することと、

前記第3のバッチ演算の実行中、データストリームから連続的に受信されているイベントデータに関して前記第2のクエリを連続的に実行することと、

前記第3のバッチ演算の実行中、前記第2のクエリの結果を、完了した前記第1のバ

20

子演算の結果と連続的にマージすることと、

前記第2のクエリと完了した前記第1のバッチ演算の結果とのマージの結果を連続的に出力することと、

前記第3のバッチ演算の完了に回答して、前記第1のバッチ演算の結果を前記第3のバッチ演算の結果と置換することとをさらに含む、請求項1に記載のコンピュータによって実現される方法。

【請求項3】

前記第1のバッチ演算はマププリデュース演算であり、前記第2のクエリは前記第1のクエリと同じ演算を行なう、請求項1または2に記載のコンピュータによって実現される方法。

10

【請求項4】

前記第2のクエリとして連続クエリ言語(CQL)クエリを受信することと、

前記CQLクエリに基づいて、前記CQLクエリの演算を行なうマププリデュース・プログラムを生成することとをさらに含む、請求項1～3のいずれか1項に記載のコンピュータによって実現される方法。

【請求項5】

前記第2のクエリとして連続クエリ言語(CQL)クエリを受信することと、

前記CQLクエリを構文解析することと、

構文解析に基づいて、前記CQLクエリを第1の組の演算子へと分離することと、

CQL演算子とピッグ・ラテン語演算子との間の特定されたマッピングに基づいて、第1の組のCQL演算子によって行なわれる演算子を行なう第2の組のピッグ・ラテン語演算子を生成することと、

20

前記ピッグ・ラテン語演算子に基づいて、前記第1のバッチ演算を行なうマププリデュース・プログラムを生成することとをさらに含む、請求項1～4のいずれか1項に記載のコンピュータによって実現される方法。

【請求項6】

前記第2のクエリとして連続クエリ言語(CQL)クエリを受信することと、

前記CQLクエリを構文解析することと、

構文解析に基づいて、前記CQLクエリを第1の組の演算子へと分離することと、

CQL演算子とピッグ・ラテン語演算子との間の特定されたマッピングに基づいて、第1の組のCQL演算子によって行なわれる演算子を行なう第2の組のピッグ・ラテン語演算子を生成することと、

30

前記第1の組からの演算子、前記第2の組からの演算子、前記第1の組からの演算子を表わすノードから前記第2の組からの演算子を表わすノードまでのエッジ、および前記第2の組からの演算子を表わすノードから前記第1の組からの演算子までのエッジを含む、ノードの有向グラフを生成することとをさらに含む、請求項1～5のいずれか1項に記載のコンピュータによって実現される方法。

【請求項7】

前記第2のクエリとして連続クエリ言語(CQL)クエリを受信することと、

前記CQLクエリを構文解析することと、

構文解析に基づいて、前記CQLクエリを第1の組の演算子へと分離することと、

CQL演算子とピッグ・ラテン語演算子との間の特定されたマッピングに基づいて、第1の組のCQL演算子によって行なわれる演算子を行なう第2の組のピッグ・ラテン語演算子を生成することと、

40

前記第1の組からの演算子、および前記第2の組からの演算子を含む、ノードの有向グラフを生成することと、

前記有向グラフにおける2つ以上のノードを、前記2つ以上のノードによって行なわれる演算間の類似点に基づいてマージすることとをさらに含む、請求項1～6のいずれか1項に記載のコンピュータによって実現される方法。

【請求項8】

50

請求項 1 ~ 7 のいずれかに記載の方法をコンピュータに実行させる、プログラム。

【請求項 9】

システムであって、

請求項 8 に記載のプログラムを格納するためのメモリと、

前記メモリに結合され、前記プログラムを実行するためのメモリとを備える、システム

【発明の詳細な説明】

【技術分野】

【0001】

関連出願との相互参照

本願は、2012年12月5日に出願された、「イベント処理をマッピングリデュースと統合するための方法」(A METHOD FOR INTEGRATING EVENT PROCESSING WITH MAP REDUCE)と題された米国仮出願第61/733,844号の、米国特許法第119条(e)による優先権を主張し、その内容全体はあらゆる目的のためにここに引用により援用される。本願はまた、2013年3月29日に出願された、「イベント処理とマッピングリデュースとの統合」(INTEGRATING EVENT PROCESSING WITH MAP REDUCE)と題された米国仮出願第61/806,744号の、米国特許法第119条(e)による優先権を主張し、その内容全体はあらゆる目的のためにここに引用により援用される。本願はまた、2013年11月13日に出願された、「イベント処理とマッピングリデュースとの統合」(INTEGRATING EVENT PROCESSING WITH MAP-REDUCE)と題された米国非仮出願第14/079,538号の、

10

20

【背景技術】

【0002】

発明の背景

連続イベントプロセッサは、イベントの連続ストリームを受信し、そこに含まれる各イベントを、そのイベントに連続イベント処理(continuous event processing: CEP)クエリを適用することによって処理することができる。そのようなCEPクエリは、構造化クエリ言語(structured query language: SQL)の延長である連続クエリ言語(continuous query language: CQL)などのCEPクエリ言語の構文に適合するようにフォーマット化され得る。SQLクエリは、リレーショナルデータベースのテーブルにすでに格納されたデータに(1つのユーザ要求につき)一度適用されることが多いが、CQLクエリは、入来するイベントストリームにおけるイベントが連続イベントプロセッサによって受信されると、それらのイベントに繰り返し適用される。

30

【発明の概要】

【発明が解決しようとする課題】

【0003】

たとえば、データストリームは、さまざまな会社についての株価を特定し得る。時間が経過するにつれて、それらの会社についての最も最近の株価が、データストリームに追加され得る。連続イベントプロセッサは、そのような株価の各々を受信し、その株価が到着するとその株価にCQLクエリを適用し得る。CQLクエリは、フィルタリング演算および集約演算を潜在的に含むさまざまな演算を特定し得る。連続イベントプロセッサは次に、CQLクエリの適用の結果を、興味を有するさまざまなリスナに出力し得る。

40

【図面の簡単な説明】

【0004】

【図1】この発明の一実施形態に従った、マッピングリデュース・アルゴリズムツールとCEPとの組合せを使用して、大量のデータに関するリアルタイムのクエリ結果を計算するための手法を示すフロー図である。

【図2】この発明の一実施形態に従った、マッピングリデュース・アルゴリズムツールとCEPとの組合せを使用して、大量のデータに関するリアルタイムのクエリ結果を計算す

50

るためのシステムを示すブロック図である。

【図3】この発明の一実施形態に従った、使用され得るシステム環境のコンポーネントを示す簡略化されたブロック図である。

【図4】この発明の実施形態に従った、使用され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5】この発明の一実施形態に従った、単一のクエリに基づいて連続処理コンポーネントおよびバッチ処理コンポーネントを生成するための手法の一例を示すフロー図である。

【図6】この発明の一実施形態に従った、クエリに基づいて機能的に同等の演算子を生成するための手法の一例を示すフロー図である。

【図7】この発明の実施形態に従った電子装置700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】

【0005】

詳細な説明

この発明の実施形態は、連続イベント処理の分野に関する。この発明の実施形態は、連続イベント処理とマップリデュース・アルゴリズムツールとの組合せを使用して、大量のデータを比較的迅速に処理し、またはクエリするための手法を伴う。一実施形態では、連続イベントプロセッサは、(a)現在実行中のマップリデュース・ジョブが開始されてから受信されたイベントからのCQLクエリ結果を、(b)最も最近完了したマップリデュース・ジョブによって生成された最も最近のクエリ結果とマージすることにより、リアルタイムの結果を連続的に生成する。現在実行中のマップリデュース・ジョブが完了すると、そのクエリ結果は格納されて連続イベントプロセッサにアクセス可能となり、最後のマップリデュース・ジョブの実行以降にサイズが大きくなったイベントデータに関して、新規のマップリデュース・ジョブが開始される。マップリデュース・アルゴリズムツールは、大量のデータを分析して処理するための便利な機構を提供する。そのような大量のデータは、「ビッグデータ」と呼ばれることもある。

【0006】

ビッグデータとは、情報技術媒体およびベンダーらによって使用される用語である。ビジネスは、テラバイト、さらにはペタバイトのデータといった膨大な量のデータを蓄積しており、市場動向および企業業績をより理解したいという、競争心の強い要望を有している。会社は、ビッグデータを管理し、分析したいという要望に直面している。現代のアプリケーションは、非常に大量のデータを扱う。この量の多さは、従来の処理機構を使用してデータを迅速に処理することを困難にする。

【0007】

ビッグデータの処理の一例として、営業部長が、先月といった長期間にわたって製品が購入され得る主要市場の各々における、会社のベストセラーの製品を識別したいという場合が考えられ得る。このタスクを遂行する上での課題は、そのような分析のために処理すべきデータの量である。データの量は、何千から何百万、または何十億もの売買トランザクション間で変わり得る。リレーショナルデータベースおよび視覚化ソフトウェアなどの一般的なツールは、典型的には、そのような量のデータに調整されていない。また、データは構造化されている場合と構造化されていない場合とがあり、それは、データ管理および分析ツールによって取扱われないかもしれない特質である。従来のリレーショナルデータベースシステムは、そのような大量のデータをタイムリーに処理することができない。

【0008】

このため、ビッグデータを管理しようとして、いくつかの試みを行なうことができる。大量の異なるデータを扱うための1つの潜在的なやり方は、より扱いやすいデータセットを作成するために、異なるサーバ間でデータを分割し、複製することであろう。しかしながら、データの複数のコピーを作成することは、情報の一貫性を維持することをより困難にする。なぜなら、1つのデータセットへの更新が、他のデータセットにコピーされないかもしれないためである。

【0009】

10

20

30

40

50

一貫性のそのような欠如は、たとえば、データを使用するアプリケーションが販売データのエントリを更新するものの、その更新が他の複製場所にコピーされない場合に、生じるかもしれない。データにアクセスする他のアプリケーションが、そのデータの一部を他のトランザクションですでに使用していた場合、状況は複雑になり得る。データが変わる場合、データの再分割が適切かもしれないが、そのような再分割を行なうべき態様が不明瞭であることが多い。

【 0 0 1 0 】

これらの複雑性に対処するための1つの可能なアプローチは、データを保護された不変データとして分散させることを伴い得る。たとえば、売買トランザクションが生じると、それらは何らかの持続性ストレージに格納され得る。アプリケーションは、その格納された生データを変えないように防止され得る。データを不変に保つことは一貫性の問題を回避し、分散は量の問題を緩和する。しかしながら、データを不変にすると、既存のデータ操作ツールが所望の態様で分散データを処理することができなくなるおそれがある。

10

【 0 0 1 1 】

ある特定の分散されたアルゴリズムックツールは、分散された不変の入力とともに機能することができ、分散されたシステムに関して、上述の販売例で説明されたものと同様のクエリを行なうことができる。この特定のアルゴリズムックツールは、マププリデュース・アルゴリズムックツールである。マププリデュース・アルゴリズムックツールのさまざまな異なる実現化例が存在する。たとえば、一実現化例は、オープンソースのApache Hadoop (アパッチ・ハドゥーブ)に見られる。Apache Hadoopを使用すると、ビッグデータ売買トランザクションを入力として使用し、次にベストセラーの製品を結果として返す、マププリデュース・ジョブを作成することが可能である。

20

【 0 0 1 2 】

マププリデュース・アルゴリズムックツールは、分散データに関して機能する。データは別々のノードへと分割可能であり、各ノードは、別個のコンピューティングマシン上で別個のファイルとして維持可能である。そのようなマシンは各々、そのマシン上で維持されたデータに関して、他のマシンと並行してクエリを実行できる。このため、各マシンは、データのその特定の区画に関する別々のクエリ結果を生成できる。次に、これらの別々の結果は、単一の一体化された結果へと減少(リデュース)され得る。マププリデュース・アルゴリズムックツールは、この減少を遂行できる。データへのエントリごとに、マププリデュース・アルゴリズムックツールは、そのエントリを結果にマップすることができる。たとえば、データエントリが製品を表わす場合、マププリデュース・アルゴリズムックツールは、各製品を、他の製品に対するその製品の売上げランキングを示す結果にマップすることができる。マププリデュース・アルゴリズムックツールは次に、これらの結果のすべてを、単一の一体化された結果へとマージすることができる。そのようなマージは、減少と呼ばれる。

30

【 0 0 1 3 】

マププリデュース・アルゴリズムックツールがデータに対してどのように演算できるかの一例について、複数の別々のウェブページ内のある特定の単語の出現数量の計算に関わる一例が考えられ得る。第1のウェブページは、特定の単語への言及を10個含むかもしれない。第2のウェブページも、特定の単語への言及を10個含むかもしれない。第3のウェブページは、特定の単語への言及を1個だけ含むかもしれない。第4のウェブページは、特定の単語への言及を2個含むかもしれない。マププリデュース・アルゴリズムックツールによって行なわれるマッピングプロセスは、各ウェブページを、そのウェブページ内の特定の単語の出現数量にマップすることができる。この例では、マププリデュース・アルゴリズムックツールによって行なわれる減少(リデュース)プロセスは単に、この例では特定の単語が結局23個出現したという最終結果を生成するために、各ウェブページにマップされた数量のすべてを合計することを伴う。

40

【 0 0 1 4 】

マププリデュース・アルゴリズムックツールが演算するデータの分散性質により、およ

50

び入力データのサイズに依存して、マップリデュース・ジョブは、 n 分ごとに、または、より高い可能性で n 時間ごとに、バッチ結果におけるベストセラーの品目を返すかもしれない(上述の販売例を参照)。ある状況下では、数分または数時間は、クエリ結果を待つには長すぎる。たとえば、いくつかの不正検出アプリケーションは、トランザクションが進行中で、かつ何千もの不正トランザクションが完了してしまう前に、即時の結果を要求する。別の例については、小売業者は、売れ行きが悪い製品の最新の知識を得て、顧客が店に入るときに値引きカードを顧客に配ることにより、店がそれらの製品の売上げを伸ばせるようにしたいかもしれない。その知識を待つために費やされる時間量が増加するにつれて、その知識が的外れになる可能性も増加する。マップリデュース・アルゴリズムの典型的な実現化例は、リアルタイムのクエリ結果を生成しない。

10

【0015】

この発明の一実施形態によれば、連続イベント処理(CEP)手法は、大きい分散データセットに対して実行されたクエリについてのクエリ結果をより迅速に生成するために、マップリデュース・アルゴリズムミックスツールとともに使用可能である。CEPを使用して、連続イベントプロセッサは、データストリームからリアルタイムでイベントを受信し、それらのイベントを受信されるとそれらに関して分析処理を行なう。たとえば、会社の店内で売買トランザクションが生じると、それらのトランザクションの各々は、それらのトランザクションの時点で、イベントとしてデータストリームに追加されてもよい。データストリームに配置されたイベントは、それらのイベントの受信に興味を有する、アプリケーションプログラムなどのエンティティによって聴かれ(listen)得る。そのようなイベントを受信するために、そのようなエンティティは、連続イベントプロセッサに登録可能である。加えて、そのような売買トランザクションが会社の店内で生じると、それらのトランザクションの各々は、リレーショナルデータベースなどの永続的データストアに追加されてもよく、それは、何らかの分割スキームに従って複数の別々の場所にわたって分散され得る。

20

【0016】

連続イベントプロセッサは、イベントストリームからのイベントがプロセッサに到着すると、それらのイベントにCEPクエリを連続的に適用することができる。たとえば、プロセッサは、売買トランザクションを表わすイベントに、特定された持続時間を有する時間窓にわたって、ベストセラーの製品を判断するクエリを適用することができる。そうするために、プロセッサは、その持続時間を有する最も最近生じた時間窓内でプロセッサがデータストリームから受信したすべてのイベントに関するデータを、コンピューティングメモリに維持することができる。しかしながら、この時間窓のこの長さは、連続イベントプロセッサが利用可能なコンピューティングメモリの量によって制限され得る。典型的には、連続イベントプロセッサは、たとえば特に、非常に大量の売買トランザクションが生じた場合、これまで生じたすべての売買トランザクションを含むように時間窓を延長するのに十分なメモリを有していないであろう。連続イベントプロセッサは、再び量に依存して、去年生じたすべての売買トランザクションを含むように時間窓を延長するのに十分なメモリさえ有していないかもしれない。加えて、時間窓のサイズが増加するにつれて、その窓内に該当するイベントのすべてに関して連続イベントプロセッサがクエリを実行するのにかかる時間量も増加する。CEP機構は従来、ほんの数時間または数分という大きさであることもある、比較的制限された持続時間を有する時間窓内で生じるイベントに関する結果を算出するために使用される。

30

40

【0017】

上述の制限のうちのいくつかを克服するために、この発明の実施形態は、CEPおよびマップリデュース・アルゴリズムミックスツールのパワーをともにマージする。マップリデュース・アルゴリズムミックスツールは、イベントデータの最も最近蓄積されたバッチについてクエリ結果を計算するために使用可能である。たとえば、マップリデュース・ジョブは、分散されたりレーショナルデータベースに格納されたイベントデータに関するクエリの結果を計算するように予定されていてもよく、そのようなデータは、典型的にはハードディ

50

スクドライブなどの磁気媒体上に、永続的で不揮発性の態様で格納されたものであってもよい。そのようなジョブは、完了するのに比較的長い時間量がかかるかもしれない。たとえば、分散データストレージにすでに蓄積されたイベントデータの量に依存して、ジョブは、完了するのに一日かかるかもしれない。この発明の一実施形態では、マップ・マップリデュース・ジョブが予定時間に一旦実行開始されると、マップリデュース・アルゴリズムツールは、予定時間までに分散データストレージに格納されたイベントデータのみを考慮する。マップリデュース・アルゴリズムツールがジョブを処理している際、マップリデュース・ジョブは、データストリームに現在到着しているイベントを含まない。一実施形態によれば、マップリデュース・ジョブは、マップリデュース・ジョブが実行開始された時間の時点で、データストリームからすでに受信され、分散データストレージに格納されたイベントからなる。

10

【0018】

各マップリデュース・ジョブは、バッチ技術の一例である。マップリデュース・ジョブが完了するのに特定の時間量がかかると仮定すると、その特定の時間量の間データストリームに到着して分散データストレージに格納されたイベントは、次に実行されるマップリデュース・ジョブ内で、すなわち次のバッチで考慮されるであろう。この発明の一実施形態によれば、連続イベントプロセッサは、現在実行中のマップリデュース・ジョブが処理され始めた後で（および、一実施形態では、後でのみ）データストリームに到着するイベントに関してCEPを連続的に行なう。そのような実施形態では、連続イベントプロセッサは、マップリデュース・アルゴリズムツールが、予定されたジョブの一環として、分散データストレージ内の以前格納されたイベントに関して実行しているのと同じまたは同様のクエリを、これらの新着イベントに関して実行する。連続イベントプロセッサは、新着イベントがデータストリームに到着すると、それらのイベントに関してこのクエリを実行でき、このため、CEPクエリはリアルタイムで実行可能である。この発明の一実施形態では、連続イベントプロセッサがクエリ結果を生成するたびに、プロセッサはそのクエリ結果を、最も最近完了したマップリデュース・ジョブが生成した最も最近の結果とマージして、一体化されたクエリ結果を生成する。一体化されたクエリ結果は、分散データストレージにすでに格納されたイベントデータ、およびそのストレージよりも最近到着したイベントデータに関して行われたクエリ処理を反映している。

20

【0019】

図1は、この発明の一実施形態に従った、マップリデュース・アルゴリズムツールとCEPとの組合せを使用して、大量のデータに関するリアルタイムのクエリ結果を計算するための手法100を示すフロー図である。手法100は、ある演算がある順序で行なわれることを伴って示されているが、この発明の代替的な実施形態は、追加の演算、より少ない演算、または異なる演算が、潜在的には異なる順序で行なわれることを伴い得る。

30

【0020】

ブロック102で、マップリデュース・アルゴリズムツール（たとえば、Apache Hadoop）は、分散データストレージに格納されたイベントデータに関して、マップリデュース・ジョブを行ない始める。一実施形態では、マップリデュース・ジョブは、分散データストレージに格納されたすべてのイベントの特定されたサブセットに関する、特定されたクエリの実行を伴う。このサブセットは、以前のマップリデュース・ジョブがもしあれば、その実行開始以降に分散データストレージに格納されたあらゆるものを含む、分散データストレージに格納されたイベントの全体集合を含み得る。

40

【0021】

ブロック104で、マップリデュース・アルゴリズムツールが分散データストレージに格納されたイベントデータに関してマップリデュース・ジョブを実行している間、連続イベントプロセッサは、データストリームを介して連続的に到着しているイベントに対してCQLクエリを連続的に実行する。CQLクエリは、現在の時刻で終了点を有する特定された時間窓内で受信されたすべてのイベントに作用できる。CQLクエリは、その実行において、そのようなイベントからデータを集約することができる。一実施形態では、

50

CQLクエリは、現在実行中のマップリデュース・ジョブによって、しかしながら別々のイベントに関して実行されているクエリと同じであり、または同様であり、または同じ機能を行なう。一実施形態では、CQLクエリは、その処理から、(現在実行中のマップリデュース・ジョブによって処理されている)分散データストレージにすでに格納されたすべてのイベントを除外する。上述のように、一実施形態では、そのようなイベントも分散データストレージに格納されるが、現在実行中のマップリデュース・ジョブの実行内で考慮されない。

【0022】

ブロック106で、連続イベントプロセッサは、特定された時間窓内にあるデータイベントに関するCQLクエリの最も最近の実行に基づいて、予備クエリ結果を算出する。

10

【0023】

ブロック108で、いずれかのマップリデュース・ジョブが完了した場合、連続イベントプロセッサは、予備クエリ結果を、最も最近完了したマップリデュース・ジョブの結果とマージする。このマージは、一体化されたクエリ結果を生成する。この発明の一実施形態では、マップリデュース・ジョブがまだ完了していない場合、予備クエリ結果が、一体化されたクエリ結果になる。

【0024】

ブロック110で、連続イベントプロセッサは、一体化されたクエリ結果を、興味を有するリスニングエンティティに出力する。たとえば、連続イベントプロセッサは、一体化されたクエリ結果を、そのような一体化されたクエリ結果を受信するために連続イベントプロセッサに登録したアプリケーションプログラムに出力できる。

20

【0025】

ブロック112で、最も最近開始されたマップリデュース・ジョブが完了したかどうかについて、判断が下される。最も最近開始されたマップリデュース・ジョブが完了した場合、制御はブロック114に進む。完了していない場合、制御はブロック104に進む。

【0026】

ブロック114で、マップリデュース・ジョブの結果が永続的に格納される。たとえば、マップリデュース・ジョブの結果は、連続イベントプロセッサがアクセスを有するハードディスクドライブ上に格納され得る。ブロック108に関連して上述されたように、連続イベントプロセッサはこの結果をプロセッサ自体の予備結果とマージして、一体化されたクエリ結果を生成することができる。一実施形態では、マップリデュース・ジョブの結果は、以前に実行されたマップリデュース・ジョブの以前に格納された結果を上書きして置換する。

30

【0027】

ブロック116で、マップリデュース・アルゴリズムツール(たとえば、Apache Hadoop)は、分散データストレージに格納されたイベントデータに関して、新規のマップリデュース・ジョブを行ない始める。一実施形態では、マップリデュース・ジョブは、分散データストレージに格納されたすべてのイベントの特定されたサブセットに関する、特定されたクエリの実行を伴う。一実施形態では、このサブセットは、以前のマップリデュース・ジョブの実行開始以降に分散データストレージに格納されたイベントを含む。制御はブロック104に戻る。

40

【0028】

たとえば、1組のウェブページにおけるある特定の単語の数量が23個であることを、最も最近完了したマップリデュース・ジョブの結果が示す場合、および、新規のウェブページがイベントとしてデータストリームに到着する場合、および、その新規のウェブページにおける特定の単語の数量が5個であると連続イベントプロセッサが判断した場合、連続イベントプロセッサは、この結果を、最も最近完了したマップリデュース・ジョブの結果とマージして、28個という一体化されたクエリ結果を生成することができる。(現在ではなく)次のマップリデュース・ジョブが、新規のウェブページを含むイベントデータに対して演算できるように、新規のウェブページは分散データストレージに追加され得

50

る。

【0029】

図2は、この発明の一実施形態に従った、マップリデュース・アルゴリズムツールとCEPとの組合せを使用して、大量のデータに関するリアルタイムのクエリ結果を計算するためのシステム200を示すブロック図である。一実施形態では、システム200は、イベントソース202と、イベントを含むデータストリーム204と、連続イベントプロセッサ206と、マップリデュース・アルゴリズムツール208と、分散データストレージ210と、リスニングエンティティ212を含む。一実施形態では、イベントソース202は、イベント（たとえば、売買トランザクション、株価など）が生じると、それらのイベントをデータストリーム204に連続的に追加できる。連続イベントプロセッサ206は、データストリーム204からイベントを連続的に読み出し、それらのイベントに関してCQLクエリを連続的に実行することができる。連続イベントプロセッサ206はまた、それが読み出すイベントを分散データストレージ210に連続的に格納できる。連続イベントプロセッサ206はまた、CQLクエリの実行の結果を、マップリデュース・アルゴリズムツール208によって実行された、完了したマップリデュース・ジョブの最も最近の結果と連続的にマージできる。マップリデュース・アルゴリズムツール208は、マップリデュース・ジョブを次々に実行できる。そのようなマップリデュース・ジョブは各々、そのマップリデュース・ジョブの実行開始時間の時点で分散データストレージ210に格納されたイベントに関するCQLクエリと同じである、または同様であるクエリを実行できる。そのようなマップリデュース・ジョブは各々、それが完了すると、そのクエリ実行の結果を、マージする目的のために、連続イベントプロセッサ206にアクセス可能である媒体（潜在的に、分散データストレージ210）上に永続的に格納できる。リスニングエンティティ212は、連続イベントプロセッサ206によって出力された、一体化されたクエリ結果を連続的に受信して、それらを使用することができ、そのような結果を人間のユーザに潜在的に表示する。

10

20

【0030】

一実施形態では、CQLクエリおよび対応するマップリデュース（たとえば、Hadoop）ジョブは、ユーザによって別々に手動で作成され、実行され得る。しかしながら、代替的な一実施形態では、CQLクエリおよび対応するマップリデュース・ジョブの作成および実行は、マップリデュース・ジョブ（またはプログラム）が、既存の対応するCQLクエリに基づいて自動的に生成され得るように、統合され得る。たとえば、一実施形態では、ユーザは、データストリームにおいて表わされた売買トランザクションイベントに基づいてベストセラーの製品を判断するCQLクエリを作成するために、CEPシステムを利用できる。次に、CQLクエリを入力として使用して、マップリデュース・プログラムが、マップリデュース・プログラムジェネレータによって自動的に生成され得る。プログラム（同じプログラム、または別個のプログラム）はまた、実行された各マップリデュース・ジョブによって返された結果を、CQLクエリの実行によって返された結果と連続的にマージするために、自動的に生成され得る。

30

【0031】

マップリデュース・ジョブは、ピッグ・ラテン語（Pig Latin）と呼ばれるある特定の言語で表わすことができる。ピッグ・ラテン語とは、高レベルの記述言語である。たとえば、ピッグ・ラテン語では、データをロードし、そのデータをフィルタリングし、そのデータを他のデータと結合するといったジョブまたはプログラムを表わすことができる。ピッグ・ラテン語プログラムは、演算子の観点から表わすことができる。これらの演算子は、データをロードする、データをフィルタリングする、データを結合するといった演算を行ってもよい。この発明の一実施形態では、ピッグ・ラテン語プログラムジェネレータは、CQLクエリを入力として受信し、そのクエリを、そのコンポーネント演算子へと自動的に分解する。そのようなCQL演算子は、フィルター演算子、結合演算子、射影演算子などを含んでいてもよい。この発明の一実施形態では、ピッグ・ラテン語プログラムジェネレータは、CQL演算子とピッグ・ラテン語演算子との間のマッピングを利用して、

40

50

1組の演算子から他のものに自動的に変換する。そのようなマッピングは、1つの演算子対1つの演算子のマッピング、1つの演算子対多くの演算子のマッピング、多くの演算子対1つの演算子のマッピング、および多くの演算子対多くの演算子のマッピングを含み得る。

【0032】

一実施形態では、1組のビッグ・ラテン語演算子が、対応するCQL演算子に基づいて自動的に生成された後で、ビッグ・ラテン語プログラムジェネレータは、CQL演算子およびビッグ・ラテン語演算子の双方を含み得るフローを自動的に生成して格納する。このフローは、CQL演算子の出力がビッグ・ラテン語演算子への入力となること、および/または、ビッグ・ラテン語演算子の出力がCQL演算子への入力となることを示し得る。このため、ビッグ・ラテン語演算子およびCQL演算子の双方を含む、演算子の有向グラフを生成することができる。この有向グラフでは、演算子ノード間のエッジは、演算子から演算子までのデータのフローを示し得る。潜在的に、1つの演算子が複数の演算子にデータを出力でき、複数の演算子が単一の演算子にデータを出力できる。グラフにおけるリーフノード演算子は、CQL演算結果とマププリデュース演算結果との最終マージを行なうことができる。有向グラフがこのように自動的に生成された後で、それらの演算子が行なう演算子のタイプに基づいた、グラフにおけるいくつかの演算子ノードのマージを潜在的に伴う、グラフの何らかの最適化を自動的に行なうことができる。たとえば、双方ともフィルタリング演算を行なうCQL演算子およびビッグ・ラテン語演算子を、単一の演算子へとともにマージすることができる。

【0033】

実行時、有向グラフは解釈され得る。ビッグ・ラテン語演算子、または1つ以上のエッジによってリンクされた1組のビッグ・ラテン語演算子がグラフに生じると、それらの演算子の演算を行なう対応するマププリデュース・プログラムが、自動的に実行され得る。マププリデュース・プログラムの結果は、入力イベントとして、有向グラフにおけるCQL演算子に対応するCQLクエリに送信され得る。

【0034】

単一の言語を使用した連続処理およびバッチ処理の結果の結合

上述のように、一実施形態では、連続クエリプロセッサは、イベントストリームに現在到着しているイベントに対してCQLクエリを連続的に処理することができ、一方、マププリデュース・プログラムは、以前にストリームから受信され、格納されたイベントを同時に処理する。大きいデータセットについて比較的新鮮な結果を生成するために、連続クエリプロセッサによって行なわれた処理の結果を、マププリデュース・プログラムによって行なわれた処理の結果と周期的にマージすることができる。

【0035】

一実施形態によれば、連続クエリプロセッサは、CQLなどの単一の高レベル言語で構成されるクエリを実行する。そのようなクエリのコンパイルの結果、2つの局面をランタイムで生成することができる。一方の局面は、イベントストリーム処理を行なう。同じコンパイル中、他方の局面として、追加の演算子を生成することができる。これらの追加の演算子は、Hadoopまたは何らかの他のマププリデュース・ツールへと統合され得る。マププリデュース・ツールは、バッチ系の処理を行なうことができる。ストリーム処理の結果は、バッチ処理の結果とマージされ得る。したがって、結果がリアルタイムで得られることも可能にしつつ、大きいデータ量が操作可能である。

【0036】

オラクル・イベント・プロセッシング(Oracle Event Processing: OEP)などのイベント処理アプリケーションは、イベントをリアルタイムで処理するように設計されている。しかしながら、そのようなアプリケーションは従来、比較的短命なイベントを処理するように設計されたものである。そのような処理は従来、あまり深くなかった。イベント処理アプリケーションは従来、処理が発生すると、発生しているイベントを処理してきた。それらのイベント処理アプリケーションは従来、何年も前にデータベースに格納された

かもしれないデータを検査していなかった。

【 0 0 3 7 】

Hadoopなどのマップリデュース・ツールは、ハードディスクドライブ上に物理的に含まれ得るデータベースなどのリポジトリに格納された大量のデータに対して演算できる。そのようなデータベースに格納されたデータは、比較的古い場合があり、何年も前のものである場合もある。そのようなデータベースに格納されたデータは、数年前に生成され、格納されたものかもしれない。それはそのような長期間蓄積していたかもしれないため、そのようなデータベースに格納されたデータの量は比較的大きい場合がある。マップリデュース・ツールは、データベースに格納された大きいデータセットを、分散された態様で処理できる。マップリデュース・ツールは、マップタスクおよびリデュースタスクを作成できる。これらのタスクは、分散された態様で実行可能である。大きいデータセットの処理は別々の局面へと分解可能であり、各局面についての処理は別々の処理要素間で分散可能である。マップリデュース・ツールはその処理を、それが大きいデータセットに関して行なうバッチジョブが開始および完了の双方を有するように、バッチ系の態様で行なうことができる。

10

【 0 0 3 8 】

マップリデュース・ツールはその処理を、連続イベントプロセッサによって同時に行なわれている処理から隔離されたやり方で行なうことができる。マップリデュース・ツールはその処理を、明確なタスク開始および終了を有するバッチ系の態様で行なうが、連続イベントプロセッサはその処理を、終了なしで連続的に行なう。イベントストリームにおけるイベントは、連続イベントプロセッサに連続的に到着する。この発明の実施形態は、連続イベントプロセッサの連続タイプの処理の結果と、マップリデュース・ツールのバッチタイプの処理の結果とを、ともに結合できる。さらに、この発明の実施形態は、CQLなどの単一の高レベル言語を使用して、この結合を遂行できる。

20

【 0 0 3 9 】

代替的なアプローチは、ストリーム処理を行なう1つのアプリケーションと、バッチ処理を行なうもう1つの別のアプリケーションとを伴い得る。異なるタイプの処理の結果をまとめようとして、第3のカスタムアプリケーションを書くことが可能である。そのようなアプローチは、1つはストリーム処理用、1つはバッチ処理用、1つは各タイプの処理の結果の結合用、といった3つの異なるプログラミング言語の使用を伴うかもしれない。

30

【 0 0 4 0 】

複数の異なる言語の使用を回避するために、この発明の一実施形態は、高レベルのCQLクエリを機能的に同等のマップリデュース・プログラムにコンパイルすることを伴う。CQLクエリの実行の結果は次に、マップリデュース・プログラムの結果とマージされる。CQLクエリは、結果をリアルタイムで生成でき、一方、マップリデュース・プログラムは、より大きいデータセットに対するより深い分析を行なうことができる。マージの結果はしたがって、リアルタイムであり、かつ深い。そのようなマージは、システムのリアルタイムコンポーネントおよび深い分析コンポーネントの双方を生成するためにコンパイルされる単一のユーザ指定クエリに基づいて達成可能である。異なる言語での別々のアプリケーションを構築する必要はない。

40

【 0 0 4 1 】

図5は、この発明の一実施形態に従った、単一のクエリに基づいて連続処理コンポーネントおよびバッチ処理コンポーネントを生成するための手法500の一例を示すフロー図である。ブロック502で、CQLで特定されたクエリが受信される。ブロック504で、コンパイラが、CQLクエリを、連続イベントプロセッサによって実行可能な連続イベント処理実行プランへとコンパイルする。ブロック506で、コンパイラは、CQLクエリを、連続イベント処理実行プランと機能的に同等であり、かつマップリデュース・プログラムによって分散された態様でバッチジョブとして実行可能な、マップタスクおよびリデュースタスクへとコンパイルする。ブロック508で、コンパイラは、連続イベントプロ

50

ロセッサによって生成された結果をマップリデュース・プログラムによって生成された結果と結合し得るマージコンポーネントを生成する。ブロック510で、連続イベントプロセッサは、イベントストリームからの現在到着しているイベントに関してプランを実行し、一方、マップリデュース・プログラムは、以前にデータベースに格納されたデータに関してマップタスクおよびリデュースタスクを実行する。ブロック512で、マージコンポーネントは、連続イベントプロセッサおよびマップリデュース・プログラムによって生成された結果を結合する。制御はブロック510に戻る。

【0042】

対応するリアルタイム演算子およびバッチ演算子の生成

例示的なシナリオでは、ユーザは、1組のテキスト内のある特定の単語への言及の数をカウントしたいかもしれない。Hadoopなどのマップリデュース・ツールは、特定の単語への言及の数をカウントするために、1組のファイルにおける各ファイルを処理することができる。マップリデュース・ツールによって処理された1組のファイルは、数年前にイベントストリームから格納された短いテキストを含むかもしれない。たとえば、1組のファイルは、比較的最近の過去（たとえば、直近の1分）を除き、10年にわたってストリームから受信された短いテキストを含むかもしれない。これが生じている間、連続イベントプロセッサは、同様の短いリアルタイムテキストを、フェイスブック（Facebook：登録商標）またはツイッター（Twitter：登録商標）などのオンラインアプリケーションから受信できる。連続イベントプロセッサは、それらのリアルタイムテキストが生成された直後に（たとえば、ユーザがフェイスブックまたはツイッターを使用してそのようなテキストを投稿した直後に）、これらのリアルタイムテキスト内の特定の単語への言及の数をカウントすることができる。マップリデュース・ツールによって計算されたカウントは、連続イベントプロセッサによって計算されたカウントと結合され得る。

【0043】

行なうべき単語カウントタスクを特定するために、ユーザは、CQLを使用してクエリを特定できる。構文的には、CQLは構造化クエリ言語（SQL）に似ている。クエリは、大まかには、FROM location GROUP BY word SELECT countといった形をとるかもしれない。そのようなクエリは、特定された場所からすべての文を集め、それらの文からの特有の単語を異なるグループへとグループ分けし、次に各グループにおける単語の数をカウントすることができる。このため、クエリは、比較的単純なフォーマットを使用して特定され得る。

【0044】

ユーザ指定CQLクエリを受信したコンパイラは、クエリを別々の演算子へと分解できる。これらはたとえば、グループ分け演算子、カウントを行なうための集約演算子、ソース演算子としても作用する出力演算子、などを含み得る。この組の演算子に基づいて、コンパイラは、(a)リアルタイムイベント処理用のクエリ実行プラン、および(b)大きいデータセットの処理用のバッチジョブ、の双方を生成できる。上述の特定の例示的な単語カウントシナリオを参照すると、バッチジョブの生成は、さまざまな文を別々の特有の単語へとマップするマップ演算子の生成を伴い得る。バッチジョブの生成はさらに、各組の特有の単語を、それらの単語の数を表わす対応する数へと減少させるリデュース演算子の生成を伴い得る。クエリ実行プランを生成する一環として、コンパイラは、バッチリデュース演算子の結果を連続集約演算子の結果とマージするマージ演算子を生成できる。バッチジョブは次に、リアルタイムの連続クエリと同時に実行され得る。マージ演算子は、マップリデュース・コンポーネントからバッチビューを、連続イベントプロセッサからリアルタイムビューを得る。

【0045】

図6は、この発明の一実施形態に従った、クエリに基づいて機能的に同等の演算子を生成するための手法600の一例を示すフロー図である。ブロック602で、クエリの現在の要素が、クエリの第1の要素に設定される。ブロック604で、現在の要素のタイプ（たとえば、グループ分け、集約、選択など）が判断される。ブロック606で、現在の要

10

20

30

40

50

素のタイプに基づいて、連続イベント処理演算子タイプが選択される。ブロック608で、現在の要素のタイプに基づいて、バッチ処理演算子タイプ（たとえば、マップ、リデュースなど）が選択される。ブロック610で、選択された連続イベント処理演算子タイプの演算子が生成され、連続イベント処理プランに追加される。ブロック612で、選択されたバッチ処理演算子タイプの演算子が生成され、バッチ処理プランに追加される。ブロック614で、クエリが現在の要素の後に要素を含むかどうかについて、判断が下される。要素を含む場合、制御はブロック616に進む。その他の場合、制御はブロック618に進む。

【0046】

ブロック616で、クエリの現在の要素が、クエリの次の要素に設定される。制御はブロック604に戻る。

10

【0047】

また、これに代えて、ブロック618で、マージ演算子が生成され、連続イベント処理プランに追加される。図6で示す手法は次に終了する。

【0048】**ステートレス演算子およびステートフル演算子**

一般に、CQLには2つの異なる組の演算子がある。そのような1組の演算子はステートレスである。そのようなステートレス演算子では、典型的には、相互接続された演算子のグラフにおいて、1つの入力が入力1つ以上の出力にマップされる。この発明の一実施形態によれば、ステートレスCQL演算子は、バッチ処理ドメインにおけるマップ演算子に対応している。もう1つの組のCQL演算子はステートフルである。そのようなステートレス演算子では、典型的には、相互接続された演算子のグラフにおいて、複数の入力が入力1つの出力にマップされる。この発明の一実施形態によれば、ステートフルCQL演算子は、バッチ処理ドメインにおけるリデュース演算子に対応している。

20

【0049】

このため、この発明の一実施形態では、コンパイラは、CQLクエリ要素が1つの入力を1つ以上の出力へとマップするかどうか判断する。マップする場合、コンパイラは、要素がステートレスイベント処理演算子に対応していると判断する。フィルター演算子は、ステートレスイベント処理演算子の一例である。イベント処理演算子がステートレス演算子であるという判断に回答して、コンパイラは、（選択されたマップリデュース言語で）バッチ処理ジョブに含むための機能的に同等のマップ演算子を生成する。また、これに代えて、コンパイラは、CQLクエリ要素が複数の入力を単一の出力へとマップするかどうか判断できる。マップする場合、コンパイラは、要素がステートフルイベント処理演算子に対応していると判断する。窓演算子は、ステートフルイベント処理演算子の一例である。イベント処理演算子がステートフル演算子であるという判断に回答して、コンパイラは、（選択されたマップリデュース言語で）バッチ処理ジョブに含むための機能的に同等のリデュース演算子を生成する。

30

【0050】

CQLクエリの要素は、ある特定のタイプの集約機能（たとえば、合計、平均、カウントなど）を行なってもよい。この発明の一実施形態では、コンパイラは、特定の集約タイプのCQLクエリ集約演算子を判断し、それに回答して、同じタイプの集約機能を行なうリデュース演算子を生成する。このため、たとえば、連続イベント演算子が合計演算を行なう場合、コンパイラはそれに回答して、合計演算を行なうリデュース演算子を生成するが、連続イベント演算子がカウント演算を行なう場合、コンパイラはそれに回答して、カウント演算を行なう集約演算子を生成する。

40

【0051】**マージされた結果における重複イベント処理の回避**

この発明の一実施形態によれば、マージ演算子は、リアルタイムの連続処理結果と非リアルタイムのバッチ処理結果との間で重複が生成されないような態様で設計されている。マージ演算子は、リアルタイムの局面および非リアルタイムの局面双方による処理に起因

50

してデータのあらゆる項目がそれ自体とマージされるのを防止する安全機能を含み得る。その結果、結合された結果の任意の特定の反復中、どの単一イベントも2回以上考慮されないであろう。この発明の一実施形態では、任意の特定の反復中、マージ演算子は、特定された時点（たとえば、1分前）よりも古い、非リアルタイムの（たとえば、マップリデュース）コンポーネントからの結果のみを受入れ、また、その特定された時点よりも古くない、リアルタイムの（たとえば、連続イベントプロセッサ）コンポーネントからの結果のみを受入れる。処理は繰り返し行なわれ得るため、時点は、反復が起こるたびに進み得る。

【0052】

たとえば、第1の反復において、マージ演算子は、現在の日付での12時50分45秒までに生じたイベントのみを含むように、大きいデータセットの境界を規定できる。マージ演算子は、その第1の反復において、現在の日付での12時50分45秒よりも最近に生じたイベントのみを含むように、連続イベントプロセッサによって取扱われるデータセットの境界を規定する。連続イベントプロセッサ用の時間窓が10秒であると仮定すると、第2の反復については、この境界は、12時50分45秒から12時50分55秒へと10秒進み得る。第1の反復中に受信されたイベントは、12時50分45秒から12時50分55秒にわたる10秒の間隔において、それらが到着すると大きいデータセットに追加され、イベントストリームからのそれらの到着時間を示すためにタイムスタンプされている。各反復の終わり、境界は、リアルタイムイベント処理の1単位に必要な時間量だけ進められ得る。この量は、連続イベントプロセッサが利用可能なメモリのサイズに基づいていてもよい。この発明の一実施形態では、この時間量は静的ではなく、異なる反復の後に境界が異なる量だけ進むことができるように、反復ごとに変わってもよい。そのような一実施形態では、各反復の境界は、一定時間の持続時間ではなく、連続イベントプロセッサによって処理すべき特定された量のイベントに基づいていてもよい。

【0053】

その結果、任意の特定の反復中、マージ演算子は、その特定の反復中に連続イベントプロセッサによって処理されたどの組のイベントも、その同じ特定の反復中にバッチプロセッサによって処理されたイベントを含むことができない、ということを確認にする。ただし、そのようなイベントは後に、次の反復中にバッチプロセッサによって処理されるであろう一組のイベント内に含まれ得る。本質的に、マージ演算子は、一体化された組におけるイベントのタイムスタンプの重複は許されないことを除き、連続イベントプロセッサおよびバッチプロセッサによって生成された結果の組の和集合を計算することができる。

【0054】

なお、ある状況下では、マージ演算子は、リアルタイム結果とバッチ結果とをただ連結するかまたは合計する以上のことをしてもよい。たとえば、平均演算子が処理に参与している場合、マージ演算子は、より複雑なマージを行なってもよい。より具体的な一例について、4個の数字の平均が4であると連続イベントプロセッサが判断した場合、かつ8個の数字の平均が10であるとバッチプロセッサが判断した場合、マージ演算子は、全部で12個の数字の平均は $(4 * 4 + 8 * 10) / 12 = 8$ であると判断するであろう。行なわれるマージの種類は、クエリで行なわれていた集約演算のタイプに依存していてもよい。

【0055】

ビッグ・ラテン語プログラムへのCQLクエリのコンパイル

ビッグ・ラテン語は、マップリデュースによって使用され得る言語である。この発明の一実施形態では、CQLクエリのコンパイルは、ビッグ・ラテン語での類似演算子の生成を伴い得る。これらの類似演算子に基づいて、機能的に同等のビッグ・ラテン語プログラムが、連続クエリプロセッサによる使用のためのCQLクエリのコンパイルと同時に、コンパイラによって自動的に生成され得る。マップリデュース・ツールは、連続クエリプロセッサのCQLクエリの実行と同時に、ビッグ・ラテン語プログラムを実行できる。

【0056】

10

20

30

40

50

しかしながら、この発明の代替的な一実施形態では、ピッグ・ラテン語演算子は生成されない。代わりに、マップリデュース・ツールによって使用可能なネイティブ演算子が生成され得る。この発明の実施形態は、ピッグ・ラテン語プログラムが生成される実施形態に限定されない。

【0057】

ハードウェア概要

図3は、この発明の一実施形態に従った、使用され得るシステム環境300のコンポーネントを示す簡略化されたブロック図である。図示されているように、システム環境300は1つ以上のクライアントコンピューティング装置302、304、306、308を含み、それらは、ウェブブラウザ、専用クライアント（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動作させるように構成されている。さまざまな実施形態では、クライアントコンピューティング装置302、304、306および308は、サーバ312とやり取りしてもよい。

10

【0058】

クライアントコンピューティング装置302、304、306、308は、汎用パーソナルコンピュータ（たとえば、マイクロソフトウィンドウズ（登録商標）および/またはアップルマッキントッシュ（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作動させ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）、携帯電話またはPDA（マイクロソフトウィンドウズモバイルなどのソフトウェアを作動させ、インターネット、eメール、SMS、ブラックベリー（登録商標）、または他の通信プロトコルに対応）、および/または、商業的に入手可能なさまざまなUNIX（登録商標）またはUNIX系オペレーティングシステムのうちのいずれかを作動させるワークステーションコンピュータ（さまざまなGNU/Linux（登録商標）オペレーティングシステムを限定なしで含む）であってもよい。また、これに代えて、クライアントコンピューティング装置302、304、306および308は、ネットワーク（たとえば、以下に説明するネットワーク310）を通して通信可能な、シンクライアントコンピュータ、インターネット対応ゲームシステム、および/または個人用メッセージング装置といった任意の他の電子装置であってもよい。例示的なシステム環境300は4つのクライアントコンピューティング装置を有して図示されているが、任意の数のクライアントコンピューティング装置がサポートされてもよい。センサ付き装置などの他の装置が、サーバ312とやり取りしてもよい。

20

30

【0059】

システム環境300は、ネットワーク310を含んでいてもよい。ネットワーク310は、TCP/IP、SNA、IPX、AppleTalkなどを限定なしで含む商業的に入手可能なさまざまなプロトコルのうちのいずれかを使用してデータ通信をサポート可能な、当業者によく知られているあらゆるタイプのネットワークであってもよい。ほんの一例として、ネットワーク310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などのローカルエリアネットワーク（local area network：LAN）；ワイドエリアネットワーク；仮想プライベートネットワーク（virtual private network：VPN）を限定なしで含む仮想ネットワーク；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、プロトコルのIEEE 802.11スイートのいずれか、当該技術分野で公知のBluetooth（登録商標）プロトコル、および/または任意の他の無線プロトコルの下で動作するネットワーク）；および/または、これらのおよび/または他のネットワークの組合せであってもよい。

40

【0060】

システム環境300はまた、1つ以上のサーバコンピュータ312を含み、それは、汎用コンピュータ、特化サーバコンピュータ（たとえば、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、ラック搭載サーバなどを含む）、サーバファーム、サーバクラスタ、もしくは任意の他の適切な構成および/または組合せであ

50

ってもよい。さまざまな実施形態では、サーバ312は、前述の開示に記載された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起動させるように適合されてもよい。たとえば、サーバ312は、図2の連続イベントプロセッサ206またはマップリデュース・アルゴリズムツール208に対応していてもよい。

【0061】

サーバ312は、上述のいずれかを含むオペレーティングシステム、および商業的に入手可能なあらゆるサーバオペレーティングシステムを起動させてもよい。サーバ312はまた、HTTPサーバ、FTPサーバ、CGIサーバ、JAVA（登録商標）サーバ、データベースサーバなどを含む、さまざまな追加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を起動させてもよい。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（Sybase）、IBMなどから商業的に入手可能なものを限定なしで含む。

【0062】

システム環境300はまた、1つ以上のデータベース314、316を含んでいてもよい。データベース314、316は、さまざまな場所に存在していてもよい。たとえば、データベース314、316のうちの1つ以上は、サーバ312にとってローカルな（および/またはサーバ312に常駐する）記憶媒体上に存在していてもよい。また、これに代えて、データベース314、316は、サーバ312からリモートであってもよく、ネットワークベースのまたは専用の接続を介してサーバ312と通信していてもよい。一組の実施形態では、データベース314、316は、当業者にはよく知られているストレージエリアネットワーク（storage area network：SAN）に存在していてもよい。同様に、サーバ312に起因する機能を行なうためのあらゆる必要なファイルが、サーバ312上にローカルにおよび/またはリモートで、適宜格納されてもよい。一組の実施形態では、データベース314、316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、検索するように適合された、オラクル10gなどのリレーショナルデータベースを含んでいてもよい。

【0063】

図4は、この発明の実施形態に従った、使用され得るコンピュータシステム400の簡略化されたブロック図である。たとえば、図2の連続イベントプロセッサ206またはマップリデュース・アルゴリズムツール208は、システム400などのシステムを用いて実現されてもよい。コンピュータシステム400は、バス424を介して電氣的に連結され得るハードウェア要素を含んで図示されている。ハードウェア要素は、1つ以上の中央処理装置（CPU）402と、1つ以上の入力装置404（たとえば、マウス、キーボードなど）と、1つ以上の出力装置406（たとえば、表示装置、プリンタなど）とを含んでいてもよい。コンピュータシステム400はまた、1つ以上の記憶装置408を含んでいてもよい。たとえば、記憶装置408は、ディスクドライブ、光学記憶装置、ならびに、ランダムアクセスメモリ（random access memory：RAM）および/または読み出し専用メモリ（read only memory：ROM）などのソリッドステート記憶装置などの装置を含んでいてもよく、それらはプログラム可能、フラッシュ更新可能なものなどであってもよい。

【0064】

コンピュータシステム400は加えて、コンピュータ読取可能記憶媒体リーダ412と、通信サブシステム414（たとえば、モデム、ネットワークカード（無線または有線）、赤外線通信装置など）と、上述のようなRAM装置およびROM装置を含み得る作業メモリ418とを含んでいてもよい。いくつかの実施形態では、コンピュータシステム400はまた、処理加速ユニット416を含んでいてもよく、それは、デジタル信号プロセッサ（digital signal processor：DSP）、専用プロセッサなどを含んでいてもよい。

【0065】

コンピュータ読取可能記憶媒体リーダ412はさらにコンピュータ読取可能記憶媒体410に接続可能であり、コンピュータ読取可能な情報を一時的におよび/またはより永続

10

20

30

40

50

的に含むためのリモートの、ローカルの、固定された、および/またはリムーバブルの記憶装置プラス記憶媒体を、ともに(および、オプションで記憶装置408と組合されて)包括的に表わす。通信システム414は、データが、ネットワークおよび/または上述した任意の他のコンピュータと交換されることを許可してもよい。

【0066】

コンピュータシステム400はまた、オペレーティングシステム420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、中間層アプリケーション、RDBMSなどであってもよい)などの他のコード422を含む、作業メモリ418内に現在位置するものとして図示されたソフトウェア要素を含んでいてもよい。例示的な一実施形態では、作業メモリ418は、上述の手法のために使用される(キャッシュなどの)実行可能コードおよび関連付けられたデータ構造を含んでいてもよい。コンピュータシステム400の代替的な実施形態は、上述のものからの多数の変形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。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、ソフトウェア(アプレットなどのポータブルソフトウェアを含む)、またはそれら双方で実現されてもよい。また、ネットワーク入力/出力装置などの他のコンピューティング装置への接続が採用されてもよい。

【0067】

コードまたはコードの一部を含むための記憶媒体およびコンピュータ読取可能媒体は、当該技術分野で公知のまたは使用されるあらゆる適切な媒体を含んでいてもよく、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、もしくは他のデータといった情報の格納および/または送信のための任意の方法あるいは技術で実現される、揮発性および不揮発性、リムーバブルおよび非リムーバブル媒体を含むもののそれらに限定されない記憶媒体および通信媒体を含み、RAM、ROM、EEPROM、フラッシュメモリまたは他のメモリ技術、CD-ROM、デジタル多用途ディスク(digital versatile disk: DVD)または他の光学ストレージ、磁気カセット、磁気テープ、磁気ディスク記憶装置または他の磁気記憶装置、データ信号、データ伝送、もしくは所望の情報を格納または送信するために使用可能であり、コンピュータによってアクセス可能な任意の他の媒体を含む。

【0068】

図7は、この発明の実施形態に従った電子装置700の簡略化されたブロック図である。電子装置700は、第1の時間の時点で永続的データストアにすでに格納されているイベントデータに関して第1のクエリを実行する第1のバッチ演算を開始するように構成された開始ユニット702と、第1のバッチ演算の実行中、データストリームから連続的に受信されているイベントデータに関して第2のクエリを連続的に実行するように構成された実行ユニット704と、第1のバッチ演算の実行中、第2のクエリの結果を、完了した第2のバッチ演算の結果と連続的にマージするように構成されたマージユニット706と、マージの結果を連続的に出力するように構成された出力ユニット708と、第1のバッチ演算の完了にตอบสนองして、第2のバッチ演算の結果を第1のバッチ演算の結果と置換するように構成された置換ユニット710とを含んでいてもよい。

【0069】

一例では、電子装置700は、第1のバッチ演算の実行中、永続的データストアに、データストリームから連続的に受信されているイベントデータを連続的に格納するように構成された格納ユニットと、第1のバッチ演算の完了にตอบสนองして、第1の時間よりも遅い第2の時間の時点で永続的データストアにすでに格納されているイベントデータに関して第1のクエリを実行する第3のバッチ演算を開始するように構成された第2の開始ユニットと、第3のバッチ演算の実行中、データストリームから連続的に受信されているイベントデータに関して第2のクエリを連続的に実行するように構成された第2の実行ユニットと、第3のバッチ演算の実行中、第2のクエリの結果を、完了した第1のバッチ演算の結果と連続的にマージするように構成された第2のマージユニットと、第2のクエリと完了し

10

20

30

40

50

た第1のバッチ演算の結果とのマージの結果を連続的に出力するように構成された第2の出力ユニットと、第3のバッチ演算の完了に回答して、第1のバッチ演算の結果を第3のバッチ演算の結果と置換するように構成された第2の置換ユニットとをさらに含む。

【0070】

一例では、第1のバッチ演算はマップリデュース演算であり、第2のクエリは第1のクエリと同じ演算を行なう。

【0071】

一例では、電子装置700は、第2のクエリとして連続クエリ言語(CQL)クエリを受信するように構成された受信ユニットと、CQLクエリに基づいて、CQLクエリの演算を行なうマップリデュース・プログラムを自動的に生成するように構成された生成ユニットとをさらに含む。

10

【0072】

一例では、電子装置700は、第2のクエリとして連続クエリ言語(CQL)クエリを受信するように構成された第2の受信ユニットと、CQLクエリを自動的に構文解析するように構成された構文解析ユニットと、構文解析に基づいて、CQLクエリを第1の組の演算子へと自動的に分離するように構成された分離ユニットと、CQL演算子とピッグ・ラテン語演算子との間の特定されたマッピングに基づいて、第1の組のCQL演算子によって行なわれる演算子を行なう第2の組のピッグ・ラテン語演算子を自動的に生成するように構成された第2の生成ユニットと、ピッグ・ラテン語演算子に基づいて、第1のバッチ演算を行なうマップリデュース・プログラムを自動的に生成するように構成された第3

20

【0073】

一例では、電子装置700は、第2のクエリとして連続クエリ言語(CQL)クエリを受信するように構成された第3の受信ユニットと、CQLクエリを自動的に構文解析するように構成された第2の構文解析ユニットと、構文解析に基づいて、CQLクエリを第1の組の演算子へと自動的に分離するように構成された第2の分離ユニットと、CQL演算子とピッグ・ラテン語演算子との間の特定されたマッピングに基づいて、第1の組のCQL演算子によって行なわれる演算子を行なう第2の組のピッグ・ラテン語演算子を自動的に生成するように構成された第4の生成ユニットと、第1の組からの演算子、第2の組からの演算子、第1の組からの演算子を表わすノードから第2の組からの演算子を表わすノードまでのエッジ、および第2の組からの演算子を表わすノードから第1の組からの演算子までのエッジを含む、ノードの有向グラフを自動的に生成するように構成された第5の生成ユニットとをさらに含む。

30

【0074】

ここに開示されたさまざまなユニットは、ハードウェア、ソフトウェア、またはそれらの組合せで実現または実行されてもよい。それらは、ここに説明された機能を行なうように設計された、汎用シングルまたはマルチチッププロセッサ、デジタル信号プロセッサ(digital signal processor: DSP)、特定用途向け集積回路(application specific integrated circuit: ASIC)、フィールドプログラマブルゲートアレイ(field programmable gate array: FPGA)または他のプログラマブルロジックデバイス、ディスクリットゲートまたはトランジスタロジック、ディスクリットハードウェアコンポーネント、またはそれらの任意の組合せで実現または実行されてもよい。汎用プロセッサは、マイクロプロセッサ、もしくは任意の従来プロセッサ、コントローラ、マイクロコントローラ、またはステートマシンであってもよい。プロセッサはまた、コンピューティング装置の組合せとして、たとえば、DSPとマイクロプロセッサとの組合せ、複数のマイクロプロセッサ、DSPコアと併用される1つ以上のマイクロプロセッサ、または他のそのような構成として実現されてもよい。いくつかの実現化例では、ユニットは、所与の機能に特有の回路によって実現されてもよい。

40

【0075】

この発明の特定の実施形態を説明してきたが、さまざまな修正、変更、代替的構成、お

50

よび同等物もこの発明の範囲内に包含される。これらの修正、変更、代替的構成、および同等物は、開示された特徴のあらゆる関連する組合せを含む。この発明の実施形態は、ある特定のデータ処理環境内での演算に制限されず、複数のデータ処理環境内で自由に演算できる。加えて、この発明の実施形態は、ある特定の一連のトランザクションおよびステップを使用して説明されてきたが、この発明の範囲が上述の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。

【0076】

また、この発明の実施形態はハードウェアとソフトウェアとのある特定の組合せを用いて説明されてきたが、ハードウェアとソフトウェアとの他の組合せもこの発明の範囲内にあることが認識されるべきである。この発明の実施形態は、ハードウェアのみで、またはソフトウェアのみで、またはそれらの組合せを用いて実現されてもよい。

10

【0077】

したがって、明細書および図面は、限定的ではなく例示的な意味であると考えられるべきである。しかしながら、請求項で述べられているようなより広い精神および範囲から逸脱することなく、追加、削減、削除、ならびに他の修正および変更がそれに行なわれてもよいことは、明らかであろう。

【図1】

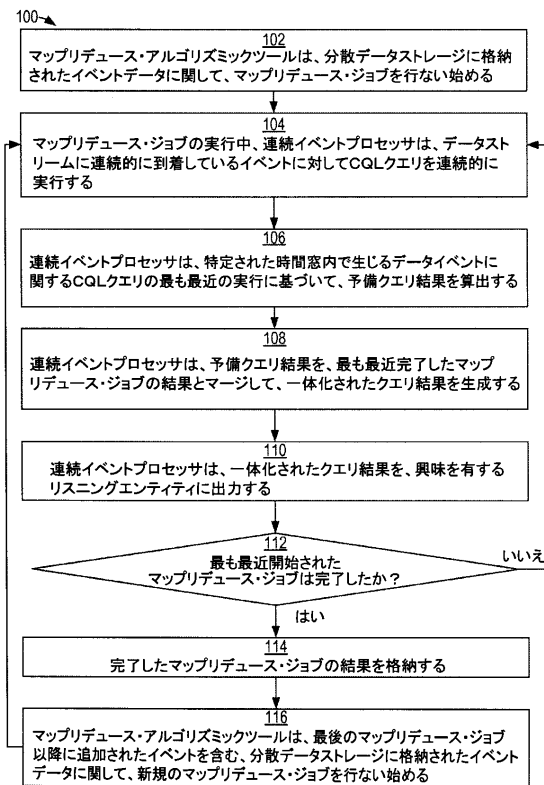


FIG. 1

【図2】

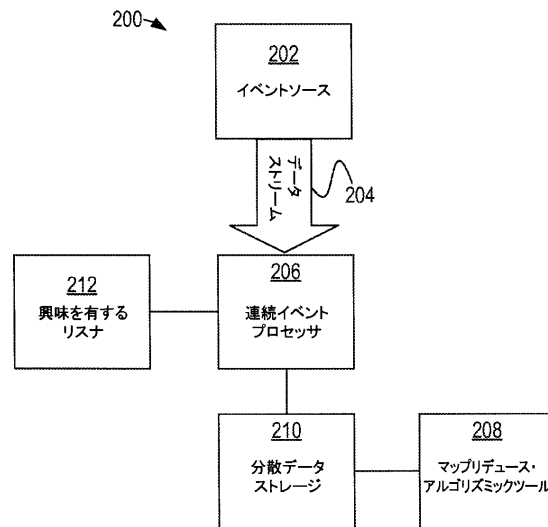


FIG. 2

【 図 3 】

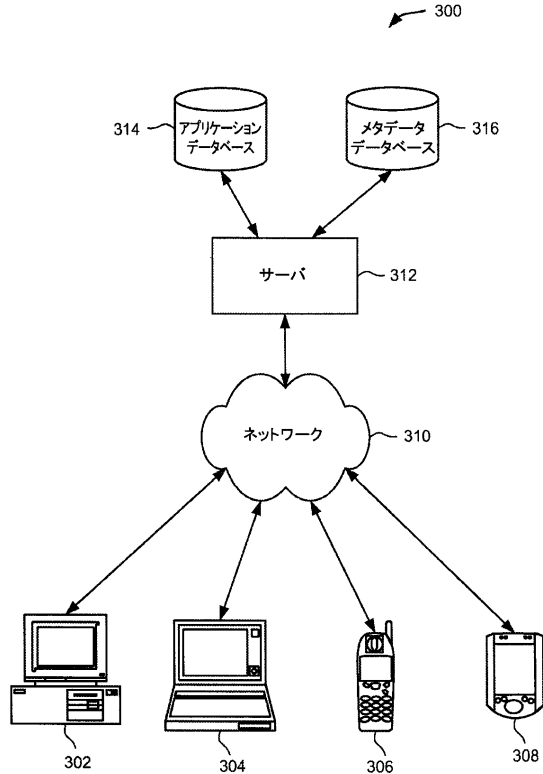


FIG. 3

【 図 4 】

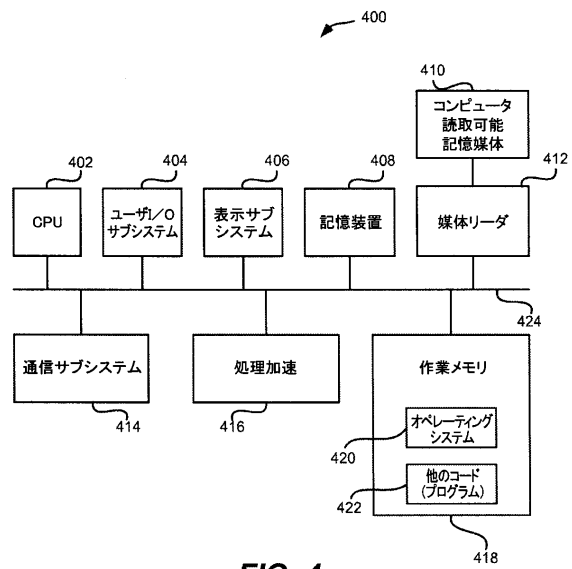


FIG. 4

【 図 5 】

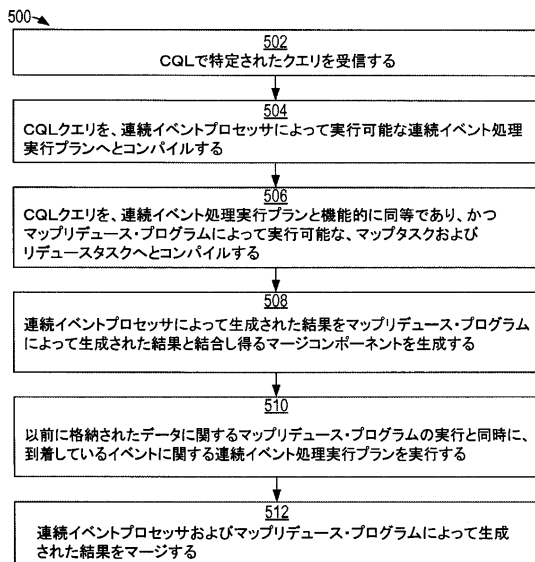


FIG. 5

【 図 6 】

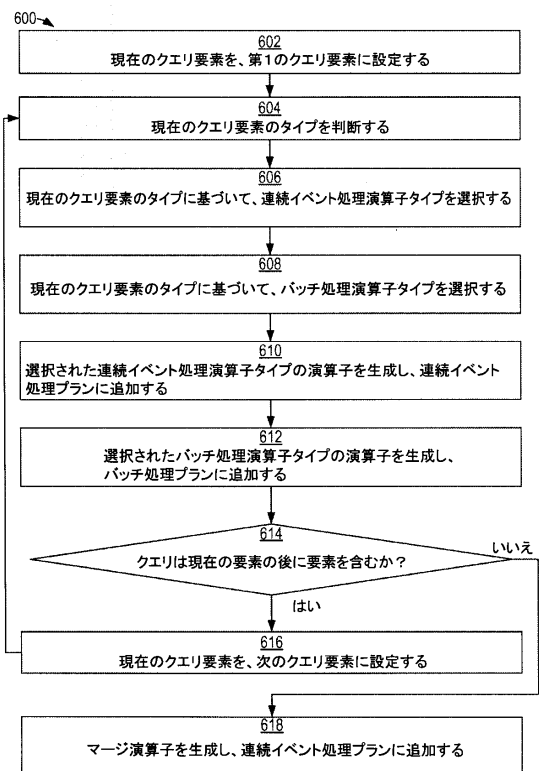


FIG. 6

【図7】

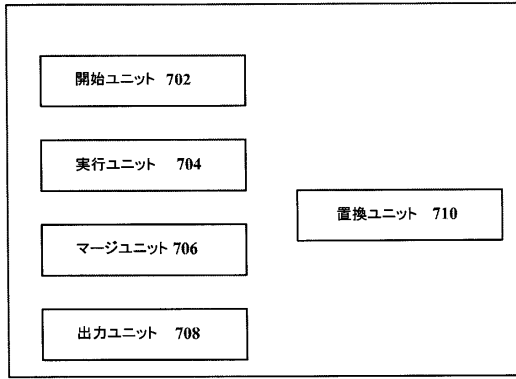


FIG. 7

フロントページの続き

(31)優先権主張番号 14/079,538

(32)優先日 平成25年11月13日(2013.11.13)

(33)優先権主張国 米国(US)

(56)参考文献 特開2010-108073(JP,A)

特開2006-338432(JP,A)

米国特許出願公開第2011/0302164(US,A1)

米国特許第6633867(US,B1)

Cagri Balkesen, Nesime Tatbul, Scalable Data Partitioning Techniques for Parallel Sliding Window Processing over Data Streams, 8th International Workshop on Data Management for Sensor Networks(DMSN 2011), 2011年 8月29日, [2017年10月19日検索], URL, <https://www.inf.ethz.ch/personal/cagri.balkesen/publications/dmsn2011.pdf>

勝沼 聡 他, ストリームデータ処理の分散並列化実行におけるマージ処理コスト削減方式, 第2回データ工学と情報マネジメントに関するフォーラム - DEIM 2010 - 論文集 [online], 日本, 電子情報通信学会データ工学研究専門委員会, 2010年 5月25日

(58)調査した分野(Int.Cl., DB名)

G06F 17/30

G06F 12/00