

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6581108号
(P6581108)

(45) 発行日 令和1年9月25日(2019.9.25)

(24) 登録日 令和1年9月6日(2019.9.6)

(51) Int.Cl.

G06F 9/50 (2006.01)
G06F 16/27 (2019.01)

F 1

G06F 9/50
G06F 16/27

150Z

請求項の数 43 (全 27 頁)

(21) 出願番号 特願2016-562015 (P2016-562015)
 (86) (22) 出願日 平成27年4月16日 (2015.4.16)
 (65) 公表番号 特表2017-518561 (P2017-518561A)
 (43) 公表日 平成29年7月6日 (2017.7.6)
 (86) 國際出願番号 PCT/US2015/026088
 (87) 國際公開番号 WO2015/161025
 (87) 國際公開日 平成27年10月22日 (2015.10.22)
 審査請求日 平成30年4月16日 (2018.4.16)
 (31) 優先権主張番号 14/255,579
 (32) 優先日 平成26年4月17日 (2014.4.17)
 (33) 優先権主張国・地域又は機関
米国(US)

(73) 特許権者 509123208
アビニシオ テクノロジー エルエルシー
アメリカ合衆国 O 2 4 2 1 マサチュー
セツツ州 レキシントン スプリング ス
トリート 201
(74) 代理人 100079108
弁理士 稲葉 良幸
(74) 代理人 100109346
弁理士 大貫 敏史
(74) 代理人 100117189
弁理士 江口 昭彦
(74) 代理人 100134120
弁理士 内藤 和彦

最終頁に続く

(54) 【発明の名称】複数ソースからのデータの処理

(57) 【特許請求の範囲】

【請求項 1】

Hadoopクラスタのノードであって、データの第1の部分をHDFSデータストレージに記憶するノードにおいて、

前記Hadoopクラスタ外部のデータソースからデータを受信することが可能なデータ処理エンジンの第1のインスタンスを実行することと、

前記データ処理エンジンによって、a)前記Hadoopクラスタを表す少なくとも1つのコンポーネントと、b)前記Hadoopクラスタ外部の前記データソースを表す少なくとも1つのコンポーネントと、c)データ処理動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクを含むデータフローグラフを受信することと、

前記データ処理エンジンの前記第1のインスタンスによって前記データフローグラフの少なくとも一部を実行することと、

前記データ処理エンジンによって前記外部データソースからデータの第2の部分を受信することと、

前記データ処理エンジンによって、少なくとも前記データの第1の部分および前記データの第2の部分を使用する前記データ処理動作を行うことと、を含む方法。

【請求項 2】

前記Hadoopクラスタは各々が前記データ処理エンジンのインスタンスを実行するノードを含み、前記データ処理エンジンの前記インスタンスは、a)前記データの第1の

10

20

部分を含むデータの第1の本体であって、前記Hadoopクラスタの他のノードによって処理されるデータの他の部分も含む、データの第1の本体に対して、および、b)前記データの第2の部分を含むデータの第2の本体であって、関係データベースシステムに固有の形式で記憶され、前記Hadoopクラスタの前記ノードの揮発性メモリに各々が記憶可能な部分に分割されるデータの第2の本体に対して、共に並列でデータ処理動作を実行するために、同時に動く、請求項1に記載の方法。

【請求項3】

前記データフローグラフの少なくとも1つのコンポーネントは前記Hadoopクラスタからのデータフローを表すリンクに接続され、前記少なくとも1つのコンポーネントは前記データの第2の部分の前記ソースからのデータフローを表すリンクに接続される、請求項1に記載の方法。10

【請求項4】

前記データ処理エンジンはMapReduceプログラミングモデルを実装していない、請求項1に記載の方法。

【請求項5】

前記データの第2の部分は揮発性メモリに記憶される、請求項1に記載の方法。

【請求項6】

データベースクエリを受信することを含み、前記データベースクエリは前記Hadoopクラスタを含む少なくとも1つのデータソースから受信されたデータに対して行われるべき少なくとも1つの動作を含み、20

前記データフローグラフは前記データベースクエリに対応する動作を表すコンポーネントを含み、前記データフローグラフは、前記少なくとも1つのデータソースを表す少なくとも1つのコンポーネントと、少なくとも1つのデータソースから受信されたデータに対して行われるべき前記動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクとを含む、請求項1に記載の方法。

【請求項7】

前記データの第2の部分は前記データの第1の部分の特徴に基づいて選択された、請求項1に記載の方法。

【請求項8】

前記データの第2の部分は関係データベースの行のサブセットを含み、前記データの第2の部分は前記関係データベースの列のサブセットを含む、請求項1に記載の方法。30

【請求項9】

前記データの第2の部分は、前記外部データソースから前記Hadoopクラスタの第2のノードで受信されたデータの第3の部分とは別個である、請求項1に記載の方法。

【請求項10】

前記Hadoopクラスタの外側の前記データ処理エンジンの第2のインスタンスによって実行される前記データフローグラフの少なくとも一部のインスタンスと通信することを含む、請求項1に記載の方法。

【請求項11】

前記Hadoopクラスタの外側の前記データ処理エンジンの第2のインスタンスによって前記データフローグラフの少なくとも一部を実行することを含む、請求項1に記載の方法。40

【請求項12】

データの第1の部分をHDFSデータストレージに記憶するHadoopクラスタのノードに、

前記Hadoopクラスタ外部のデータソースからデータを受信することが可能なデータ処理エンジンの第1のインスタンスを実行すること、

前記データ処理エンジンによって、a)前記Hadoopクラスタを表す少なくとも1つのコンポーネントと、b)前記Hadoopクラスタ外部の前記データソースを表す少なくとも1つのコンポーネントと、c)データ処理動作に関連付けられた少なくとも1つ50

のデータフローを表す少なくとも1つのリンクを含むデータフローグラフを受信すること、

前記データ処理エンジンの前記第1のインスタンスによって前記データフローグラフの少なくとも一部を実行すること、

前記データ処理エンジンによって前記外部データソースからデータの第2の部分を受信すること、ならびに、

前記データ処理エンジンによって、少なくとも前記データの第1の部分および前記データの第2の部分を使用する前記データ処理動作を行うこと、を含む動作を実施させるための命令を含む、コンピュータ可読ストレージデバイス。

【請求項13】

10

データの第1の部分をHDFSストレージに記憶する、Hadoopクラスタのノードであって、

前記Hadoopクラスタ外部のデータソースからデータを受信することが可能なデータ処理エンジンの第1のインスタンスを実行すること、

前記データ処理エンジンによって、a)前記Hadoopクラスタを表す少なくとも1つのコンポーネントと、b)前記Hadoopクラスタ外部の前記データソースを表す少なくとも1つのコンポーネントと、c)データ処理動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクを含むデータフローグラフを受信すること、

前記データ処理エンジンの前記第1のインスタンスによって前記データフローグラフの少なくとも一部を実行すること、

前記データ処理エンジンによって前記外部データソースからデータの第2の部分を受信すること、ならびに、

前記データ処理エンジンによって、少なくとも前記データの第1の部分および前記データの第2の部分を使用する前記データ処理動作を行うこと、を含む動作を実施するように構成されたコンピュータ処理デバイスを含む、

Hadoopクラスタのノード。

【請求項14】

20

データの第1の部分をHDFSストレージに記憶する、Hadoopクラスタのノードであって、

30

前記Hadoopクラスタ外部のデータソースからデータを受信することが可能なデータ処理エンジンの第1のインスタンスを実行するための手段と、

前記データ処理エンジンによって、a)前記Hadoopクラスタを表す少なくとも1つのコンポーネントと、b)前記Hadoopクラスタ外部の前記データソースを表す少なくとも1つのコンポーネントと、c)データ処理動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクを含むデータフローグラフを受信するための手段と、

前記データ処理エンジンの前記第1のインスタンスによって前記データフローグラフの少なくとも一部を実行するための手段と、

前記データ処理エンジンによって前記外部データソースからデータの第2の部分を受信するための手段と、

40

前記データ処理エンジンによって、少なくとも前記データの第1の部分および前記データの第2の部分を使用する前記データ処理動作を行うための手段と、を含む、Hadoopクラスタのノード。

【請求項15】

ノードのクラスタのノードで、前記ノードはデータの第1の部分を記憶し、前記ノードのクラスタと共に1つまたは複数のデータ処理動作を実行するように構成され、前記クラスタは前記ノード全体からのデータの集合を記憶し、前記ノードは前記データの集合に対して並列に動作するように構成され、前記データの集合は複数の部分に分割され、各部分に対して前記クラスタのそれぞれの部分が動作し、

50

前記クラスタ外部のデータソースからデータを受信することが可能なデータ処理エンジンの第1のインスタンスを実行すること、

前記データ処理エンジンによって、a) Hadoop クラスタを表す少なくとも1つのコンポーネントと、b) 前記 Hadoop クラスタ外部の前記データソースを表す少なくとも1つのコンポーネントと、c) データ処理動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクを含むデータフローグラフを受信すること、

前記データ処理エンジンの前記第1のインスタンスによって前記データフローグラフの少なくとも一部を実行すること、

前記データの第1の部分の特徴に基づいて、前記データ処理エンジンによってデータの第2の部分を要求すること、

前記データ処理エンジンによって前記外部データソースから前記データの第2の部分を受信すること、

前記データの第2の部分を前記ノードの揮発性メモリに記憶すること、ならびに、

前記データ処理エンジンによって、少なくとも前記データの第1の部分および前記データの第2の部分を使用する前記データ処理動作を行うこと、
を含む、方法。

【請求項16】

Hadoop クラスタのノードのデータ処理エンジンで、前記データ処理エンジンによって実行される、a) 前記 Hadoop クラスタを表す少なくとも1つのコンポーネントと、b) 前記 Hadoop クラスタ外部のデータソースを表す少なくとも1つのコンポーネントと、c) データ処理動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクを含むデータフローグラフによって識別されたデータ処理動作を行うことを含む方法であって、前記データ処理動作は、前記ノードの HDFS データストレージに記憶されたデータの少なくとも第1の部分、および前記 Hadoop クラスタ外部のデータソースから受信されたデータの少なくとも第2の部分を使用して行われる、方法。

【請求項17】

Hadoop クラスタおよび関係データベースを含むデータソースを指定するSQLクエリを受信すること、

前記SQLクエリに対応する、a) 前記 Hadoop クラスタを表す少なくとも1つのコンポーネントと、b) 前記 Hadoop クラスタ外部の前記データソースを表す少なくとも1つのコンポーネントと、c) データ処理動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクを含むデータフローグラフを生成すること、

前記 Hadoop クラスタのノードのデータ処理エンジンで前記データフローグラフを実行すること、ならびに、

前記データ処理エンジンによって少なくとも前記 Hadoop クラスタのデータおよび前記関係データベースのデータを使用する前記データ処理動作を行うこと、
を含む、方法。

【請求項18】

Hadoop クラスタは前記データ処理エンジンのインスタンスを各々が実行するノードを含み、前記データ処理エンジンの前記インスタンスは、a) 前記データの第1の部分を含み、前記 Hadoop クラスタの他のノードにより処理されるデータの他の部分も含むデータの第1の本体に対して、および、b) 前記データの第2の部分を含み、関係データベースシステムに固有の形式で記憶され、前記 Hadoop クラスタの前記ノードの揮発性メモリに各々が記憶可能な部分に分割されるデータの第2の本体に対して、共に並列で前記データ処理動作を実行する、請求項12に記載のコンピュータ可読ストレージデバイス。

【請求項19】

前記データフローグラフの少なくとも1つのコンポーネントは前記 Hadoop クラスタからのデータフローを表すリンクに接続され、前記少なくとも1つのコンポーネントは

前記データの第2の部分の前記外部データソースからのデータフローを表すリンクに接続される、請求項12に記載のコンピュータ可読ストレージデバイス。

【請求項20】

前記データ処理エンジンはMapReduceプログラミングモデルを実装していない、請求項12に記載のコンピュータ可読ストレージデバイス。

【請求項21】

前記データの第2の部分は揮発性メモリに記憶される、請求項12に記載のコンピュータ可読ストレージデバイス。

【請求項22】

前記動作は、

10

データベースクエリを受信することを含み、前記データベースクエリは前記Hadoopクラスタを含む少なくとも1つのデータソースから受信されたデータに対して行われるべき少なくとも1つの動作を含み、

前記データフローグラフは、前記データベースクエリに対応する動作を表すコンポーネントを含み、前記データフローグラフは、前記少なくとも1つのデータソースを表す少なくとも1つのコンポーネントと、少なくとも1つのデータソースから受信されたデータに対して行われるべき前記動作に関連付けられる少なくとも1つのデータフローを表す少なくとも1つのリンクとを含む、

請求項12に記載のコンピュータ可読ストレージデバイス。

【請求項23】

20

前記データの第2の部分は前記データの第1の部分の特徴に基づいて選択された、請求項12に記載のコンピュータ可読ストレージデバイス。

【請求項24】

前記データの第2の部分は関係データベースの行のサブセットを含み、前記データの第2の部分は前記関係データベースの列のサブセットを含む、請求項12に記載のコンピュータ可読ストレージデバイス。

【請求項25】

前記データの第2の部分は、前記外部データソースから前記Hadoopクラスタの第2のノードで受信されたデータの第3の部分とは別個である、請求項12に記載のコンピュータ可読ストレージデバイス。

30

【請求項26】

前記動作は、Hadoopクラスタの外側の前記データ処理エンジンの第2のインスタンスによって実行される前記データフローグラフの少なくとも一部のインスタンスと通信することを含む、請求項12に記載のコンピュータ可読ストレージデバイス。

【請求項27】

前記動作は、前記Hadoopクラスタの外側の前記データ処理エンジンの第2のインスタンスによって前記データフローグラフの少なくとも一部を実行することを含む、請求項12に記載のコンピュータ可読ストレージデバイス。

【請求項28】

Hadoopクラスタは前記データ処理エンジンのインスタンスを各々が実行するノードを含み、前記データ処理エンジンの前記インスタンスは、a)前記データの第1の部分を含み、前記Hadoopクラスタの他のノードにより処理されるデータの他の部分も含むデータの第1の本体に対して、および、b)前記データの第2の部分を含み、関係データベースシステムに固有の形式で記憶され、前記Hadoopクラスタの前記ノードの揮発性メモリに各々が記憶可能な部分に分割されるデータの第2の本体に対して、共に並列で前記データ処理動作を実行する、請求項13に記載のノード。

40

【請求項29】

前記データフローグラフの少なくとも1つのコンポーネントは前記Hadoopクラスタからのデータフローを表すリンクに接続され、前記少なくとも1つのコンポーネントは前記データの第2の部分の前記外部データソースからのデータフローを表すリンクに接続

50

される、請求項 13 に記載のノード。

【請求項 30】

前記データ処理エンジンは MapReduce プログラミングモデルを実装していない、請求項 13 に記載のノード。

【請求項 31】

前記データの第 2 の部分は揮発性メモリに記憶される、請求項 13 に記載のノード。

【請求項 32】

前記動作は、

データベースクエリを受信することを含み、前記データベースクエリは前記 Hadoop クラスタを含む少なくとも 1 つのデータソースから受信されたデータに対して行われるべき少なくとも 1 つの動作を含み、

前記データフローグラフは、前記データベースクエリに対応する動作を表すコンポーネントを含み、前記データフローグラフは、前記少なくとも 1 つのデータソースを表す少なくとも 1 つのコンポーネントと、少なくとも 1 つのデータソースから受信されたデータに対して行われるべき前記動作に関連付けられる少なくとも 1 つのデータフローを表す少なくとも 1 つのリンクとを含む、

請求項 13 に記載のノード。

【請求項 33】

前記データの第 2 の部分は前記データの第 1 の部分の特徴に基づいて選択された、請求項 13 に記載のノード。

20

【請求項 34】

前記データの第 2 の部分は関係データベースの行のサブセットを含み、前記データの第 2 の部分は前記関係データベースの列のサブセットを含む、請求項 13 に記載のノード。

【請求項 35】

前記データの第 2 の部分は、前記外部データソースから前記 Hadoop クラスタの第 2 のノードで受信されたデータの第 3 の部分とは別個である、請求項 13 に記載のノード。

【請求項 36】

前記動作は、Hadoop クラスタの外側の前記データ処理エンジンの第 2 のインスタンスによって実行される前記データフローグラフの少なくとも一部のインスタンスと通信することを含む、請求項 13 に記載のノード。

30

【請求項 37】

前記動作は、前記 Hadoop クラスタの外側の前記データ処理エンジンの第 2 のインスタンスによって前記データフローグラフの少なくとも一部を実行することを含む、請求項 13 に記載のノード。

【請求項 38】

Hadoop クラスタは前記データ処理エンジンのインスタンスを各々が実行するノードを含み、前記データ処理エンジンの前記インスタンスは、a) 前記データの第 1 の部分を含み、前記 Hadoop クラスタの他のノードにより処理されるデータの他の部分も含むデータの第 1 の本体に対して、および、b) 前記データの第 2 の部分を含み、関係データベースシステムに固有の形式で記憶され、前記 Hadoop クラスタの前記ノードの揮発性メモリに各々が記憶可能な部分に分割されるデータの第 2 の本体に対して、共に並列で前記データ処理動作を実行する、請求項 15 に記載の方法。

40

【請求項 39】

前記データフローグラフの少なくとも 1 つのコンポーネントは前記 Hadoop クラスタからのデータフローを表すリンクに接続され、前記少なくとも 1 つのコンポーネントは前記データの第 2 の部分の前記外部データソースからのデータフローを表すリンクに接続される、請求項 15 に記載の方法。

【請求項 40】

Hadoop クラスタは前記データ処理エンジンのインスタンスを各々が実行するノー

50

ドを含み、前記データ処理エンジンの前記インスタンスは、a) 前記データの第1の部分を含み、前記Hadoopクラスタの他のノードにより処理されるデータの他の部分も含むデータの第1の本体に対して、および、b) 前記データの第2の部分を含み、関係データベースシステムに固有の形式で記憶され、前記Hadoopクラスタの前記ノードの揮発性メモリに各々が記憶可能な部分に分割されるデータの第2の本体に対して、共に並列で前記データ処理動作を実行する、請求項16に記載の方法。

【請求項41】

前記データフローグラフの少なくとも1つのコンポーネントは前記Hadoopクラスタからのデータフローを表すリンクに接続され、前記少なくとも1つのコンポーネントは前記データの第2の部分の前記外部データソースからのデータフローを表すリンクに接続される、請求項16に記載の方法。 10

【請求項42】

Hadoopクラスタは前記データ処理エンジンのインスタンスを各々が実行するノードを含み、前記データ処理エンジンの前記インスタンスは、a) 前記データの第1の部分を含み、前記Hadoopクラスタの他のノードにより処理されるデータの他の部分も含むデータの第1の本体に対して、および、b) 前記データの第2の部分を含み、関係データベースシステムに固有の形式で記憶され、前記Hadoopクラスタの前記ノードの揮発性メモリに各々が記憶可能な部分に分割されるデータの第2の本体に対して、共に並列で前記データ処理動作を実行する、請求項17に記載の方法。

【請求項43】

前記データフローグラフの少なくとも1つのコンポーネントは前記Hadoopクラスタからのデータフローを表すリンクに接続され、前記少なくとも1つのコンポーネントは前記関係データベースからのデータフローを表すリンクに接続される、請求項17に記載の方法。 20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数ソースからのデータの処理に関する。

【背景技術】

【0002】

データは、たとえばHDFS(Hadoop分散ファイルシステム)クラスタを含む、様々なソースに記憶することができる。データ処理システムは、HDFSクラスタから受信したデータに対して、および他のタイプのソースから受信したデータに対しても動作を行うことができる。 30

【発明の概要】

【0003】

第1の態様において、方法は、データの第1の部分をHDFSデータストレージに記憶する、Hadoopクラスタのノードで、Hadoopクラスタ外部のデータソースからデータを受信することが可能なデータ処理エンジンの第1のインスタンスを実行すること、データ処理エンジンによってコンピュータ実行可能プログラムを受信すること、データ処理エンジンの第1のインスタンスによってプログラムの少なくとも一部を実行すること、データ処理エンジンによって外部データソースからデータの第2の部分を受信すること、データの第2の部分をHDFSストレージ以外に記憶すること、ならびに、データ処理エンジンによって、少なくともデータの第1の部分およびデータの第2の部分を使用するプログラムによって識別されたデータ処理動作を行うことを含む。 40

【0004】

第2の態様において、方法は、データの第1の部分を記憶し、並列で動作可能なデータの集約を記憶する、ノードのクラスタと共に動作するノードで、クラスタ外部のデータソースからデータを受信することが可能なデータ処理エンジンの第1のインスタンスを実行すること、データ処理エンジンによってコンピュータ実行可能プログラムを受信すること 50

、データ処理エンジンの第1のインスタンスによってプログラムの少なくとも一部を実行すること、データ処理エンジンによって外部データソースからデータの第2の部分を受信すること、データの第2の部分をノードの揮発性メモリに記憶すること、ならびに、データ処理エンジンによって、少なくともデータの第1の部分およびデータの第2の部分を使用するプログラムによって識別されたデータ処理動作を行うことを含む。

【0005】

第1または第2の態様に従った第3の態様において、Hadoopクラスタはデータ処理エンジンのインスタンスを各々が実行するノードを含み、データ処理エンジンのインスタンスは、a)データの第1の部分を含むデータの第1の本体であり、Hadoopクラスタの他のノードによって処理されるデータの他の部分も含む、データの第1の本体上、および、b)データの第2の部分を含むデータの第2の本体であり、関係データベースシステムに固有の形式で記憶され、データの第2の本体はHadoopクラスタのノードの揮発性メモリに各々が記憶可能な部分に分割される、データの第2の本体上で、共に並列でデータ処理動作を実行するために、同時に行う。10

【0006】

第1から第3までの態様のいずれかに従った第4の態様において、コンピュータプログラムはデータ処理エンジンのグラフ実行エンジンによって実行されるデータフローグラフであり、データフローグラフは、a) Hadoopクラスタを表す少なくとも1つのコンポーネント、b)データの第2の部分のソースを表す少なくとも1つのコンポーネント、およびc)少なくとも1つのデータソースから受信されたデータに対して行われるべき動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクを含む。20

【0007】

第4の態様に従った第5の態様において、データフローグラフの少なくとも1つのコンポーネントはHadoopクラスタからのデータフローを表すリンクに接続され、少なくとも1つのコンポーネントはデータの第2の部分のソースからのデータフローを表すリンクに接続される。20

【0008】

第1から第5までの態様のいずれかに従った第6の態様において、データ処理エンジンはMapReduceプログラミングモデルを実装していない。30

【0009】

第1から第6までの態様のいずれかに従った第7の態様において、データの第2の部分は揮発性メモリに記憶される。

【0010】

第1から第7までの態様のいずれかに従った第8の態様において、方法は、データベースクエリを受信することを含み、データベースクエリはHadoopクラスタを含む少なくとも1つのデータソースから受信されたデータに対して行われるべき少なくとも1つの動作を含み、コンピュータプログラムはデータベースクエリに対応する動作を表すコンポーネントを含み、コンピュータプログラムは、少なくとも1つのデータソースを表す少なくとも1つのコンポーネントと、少なくとも1つのデータソースから受信されたデータに対して行われるべき動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクとを含む。40

【0011】

第1から第8までの態様のいずれかに従った第9の態様において、データの第2の部分はデータの第1の部分の特徴に基づいて選択された。

【0012】

第1から第9までの態様のいずれかに従った第10の態様において、データの第2の部分は関係データベースの行のサブセットを含み、データの第2の部分は関係データベースの列のサブセットを含む。

【0013】

第 1 から第 10 までの態様のいずれかに従った第 11 の態様において、データの第 2 の部分は、外部データソースから Hadoop クラスタの第 2 のノードで受信されたデータの第 3 の部分とは別個である。

【 0014 】

第 1 から第 11 までの態様のいずれかに従った第 12 の態様において、方法は、Hadoop クラスタの外側のデータ処理エンジンの第 2 のインスタンスによってプログラムの少なくとも一部を実行することを含む。

【 0015 】

第 1 から第 12 までの態様のいずれかに従った第 13 の態様において、方法は、Hadoop クラスタの外側のデータ処理エンジンの第 2 のインスタンスによって実行されるプログラムの少なくとも一部と通信することを含む。10

【 0016 】

第 14 の態様において、方法は、Hadoop クラスタのノードのデータ処理エンジンで、データ処理エンジンによって実行されるコンピュータ実行可能プログラムによって識別されたデータ処理動作を行うことを含み、データ処理動作は、ノードの HDFS データストレージに記憶されたデータの少なくとも第 1 の部分、および Hadoop クラスタ外部のデータソースから受信され、HDFS ストレージ以外に記憶されたデータの少なくとも第 2 の部分を使用して行われる。

【 0017 】

第 15 の態様において、方法は、Hadoop クラスタおよび関係データベースを含むデータソースを指定する SQL クエリを受信すること、SQL クエリに対応するコンピュータ実行可能プログラムを生成すること、Hadoop クラスタのノードのデータ処理エンジンでコンピュータ実行可能プログラムを実行すること、ならびに、少なくとも Hadoop クラスタのデータおよび関係データベースのデータを使用するコンピュータ実行可能プログラムによって識別されたデータ処理動作を、データ処理エンジンによって行うことを行ふことを含む。20

【 0018 】

1 つまたは複数の態様は、単独または組み合わせて、コンピュータシステムによって実行されたとき、態様の動作を実施する機械可読命令を含むコンピュータプログラム製品を記憶する、システムまたは装置として、あるいはコンピュータ可読ストレージデバイスとして、表すことができる。一例として、コンピュータ可読ストレージデバイスは、コンピュータシステムによって実行されたとき、第 1 から第 15 までの態様のいずれか 1 つに従って動作を実施する、機械可読命令を含む、コンピュータプログラム製品を記憶することができる。別の例として、1 つまたは複数のプロセッサを含むコンピュータシステムは、1 つまたは複数のプロセッサによって実行されたとき、第 1 から第 15 までの態様のいずれか 1 つに従って動作を実施する機械可読命令を含むコンピュータプログラム製品を記憶するコンピュータ可読ストレージデバイスを含むことができる。30

【 0019 】

上記態様のうちの 1 つまたは複数は、以下の利点を提供することができる。第 1 に、Hadoop ノードは揮発性メモリに記憶したデータに対して動作可能であり、データに対して動作を行う前にデータをディスクに書き込む必要はない。第 2 に、Hadoop ノードは複数タイプのデータソースからデータを受信するように構成可能である。第 3 に、Hadoop ノードは汎用データ処理オペレーティングシステム、たとえば Hadoop ノードに特有でないデータ処理オペレーティングシステムに関連して動作するように構成可能である。第 4 に、Hadoop ノードはデータ処理動作を実施するデータフローグラフで動作するように構成可能である。40

【 0020 】

本発明の他の特徴および利点は、以下の説明から、および特許請求の範囲から明らかとなろう。

【 図面の簡単な説明 】

【0021】

- 【図1】データ処理システムを示す図である。
- 【図2】データフローグラフを示す図である。
- 【図3】データフローグラフを示す図である。
- 【図4】データ処理手順を示すフロー・チャートである。
- 【図5】データ処理システムを示す図である。

【発明を実施するための形態】**【0022】**

図1は、1つのタイプのデータソース110から発信するデータ116a～dおよび別のタイプのデータソース120から発信するデータ104が処理され、処理されたデータ106が1つまたは複数の出力150に提供される、データ処理システム100を示す。
データに対して行われる動作が、データソース110、120のいずれのタイプから発信されたデータであるかに制限されないように、データを処理することができる。これを遂行する様式の1つによって、データソース110のうちの1つが他のデータソース120からデータ104を受信し、受信したデータ104をデータソース110に固有の技法を使用して処理することが可能となる。このように、データ処理の多くはデータソース110によって行われる。複数タイプのデータソースからのデータを処理するデータ処理システムは、連合データ処理システムと呼ばれることがある。

【0023】

1つのタイプのデータソース110は、Hadoop分散ファイルシステム（時にはHDFSと呼ばれる）クラスタに記憶されるファイルの集合である。HDFSは、各々が技法に準拠した様式でデータを記憶する複数のコンピュータシステムにわたってデータを分散するためには使用可能な、ファイルシステムを定義する技法である。単にHadoopクラスタとも呼ばれるHDFSクラスタは、データの一部に対して单一の動作を並列に（たとえばほぼ同時に）実施できるようにデータの一部を記憶している、コンピュータシステム（時にはノードと呼ばれる）の集合である。各ノードのデータは、HDFS技法によって定義されたファイルシステムを使用して記憶される。ファイルシステムはHDFSストレージと呼ばれるときもある。一般に、HDFSに従って動作するファイルシステムは、いずれの種類のデータファイルでも記憶することができる。時には、シーケンスファイルと呼ばれるHadoop固有のファイルタイプが、Hadoopノードに記憶されるデータのファイル形式として使用される。Hadoopクラスタは、何十何百もの（またはそれ以上の）ノードを有することができる。このようにHadoopクラスタは、それら何十何百ものノードにわたって单一のデータ処理動作を並列に実施することができ、各ノードはデータの一部に対して動作する。以下で説明するような技法を使用して、それ以外の方法で動作を行う異なるデータ処理システムではなく、Hadoopクラスタに対して、ほとんどまたはすべてのデータ処理動作を実施することができる。

【0024】

著者らは一般に、Hadoopノードをデータの一部を記憶するコンピュータシステムとして説明するが、Hadoopノードは他の形を取ることができる。データの特定部分がコンピュータハードウェアの特定部分に関連付けられる任意の配置構成を、Hadoopノードとすることができます。たとえば单一のHadoopノード自体を、ノードを形成するために共に動作する2つまたはそれ以上のコンピュータシステム、ノードを形成するために共に動作するマルチプロセッサコンピュータシステムの2つのプロセッサ、あるいは何らかの他の配置構成であるかどうかにかかわらず、複数のコンピュータシステムで構成することができる。单一のコンピュータシステムがHDFS技法に従って動作する2つの別個のファイルシステムを有しており、各々が独自のデータ部分を備えている場合、单一のコンピュータシステムは複数のHadoopノードとしても動作可能である。さらに筆者らが、ノードが特定のアクションを行うとする場合、それはノードが、説明するアクションを機能コンポーネントがその上で実施するプラットフォームとして働くことを意味する。たとえばノード上で実行するコンピュータプログラムは、アクションを実施す

ることができる。

【0025】

さらに筆者らは本明細書で Hadoop 技法を参照するが、Hadoop の名前を持たない、および / または HDFS データストレージ形式を使用しない他の同様の技法を、本明細書で説明する技法と共に使用することができる。このように、これらの同じ技法を他のタイプのクラスタと共に使用することができる。たとえばこれらの技法を、（たとえば、データの集約を個々のノードによって動作される部分に分けることによって）データの集約に対してデータ処理動作を実施するために、互いに関連して動作するノードによって並列に動作可能なデータの集約を記憶する別の種類のクラスタと共に使用することができる。

10

【0026】

Hadoop クラスタ内のデータを処理する 1 つの方法は、MapReduce プログラミングモデルを使用することである。一般に MapReduce プログラムは、フィルタリングおよび分類（大学生を名前によってキューに分類することなどであり、名前ごとに 1 つのキューがある）を行う Map 手順、および集計動作（それぞれのキュー内の大学生の数を数え、名前の頻度を引き出すことなど）を行う Reduce 手順を含む。システムのユーザは Map および Reduce の手順を指定するが、各手順（すなわちプロセス）のインスタンス（または呼び出し）の数、またはそれらを実行するノードは必ずしも決定しない。むしろ「MapReduce システム」（「インフラストラクチャ」、「フレームワーク」とも呼ばれる）は、分散ノードのセットを整列させ、様々なタスク（たとえば Map および Reduce 手順および関連付けられた通信）を並列に実行し、システムの様々な部分間でのすべての通信およびデータ転送を管理し、冗長性および障害に備え、全プロセスを総括的に管理することによって、調整する。MapReduce システムは、データ位置を認識することで、Map または Reduce 手順のインスタンスの実行をスケジューリングすることができる。

20

【0027】

他方のデータソース 120 は、関係データベース（時には関係データベース管理システム、または RDBMS と呼ばれる）、フラットファイル、ネットワークリソースからのデータのフィード、または、データ処理システムからの要求に応答してデータを提供できる任意の他のリソースなどの、データソースとすることができます。データ処理動作は、Hadoop クラスタ 112 に記憶されたデータと、他方のデータソース 120 から受信されたデータ 104 との組み合わせに対して行うことができる。データを Hadoop クラスタ 112 から、および他方のデータソース 120 から抽出するために、独立した処理システムを使用するのではなく、Hadoop クラスタ 112 のデータ処理機能を使用して、Hadoop クラスタ 112 に記憶されたデータ 116a~d と、他方のデータソース 120 から受信したデータ 104 との組み合わせを処理することができる。たとえばこれは、Hadoop クラスタ 112 のデータ処理機能を利用して実行できる。たとえば図 1 に示されるように、他方のデータソース 120 から受信したデータ 104 は、Hadoop クラスタ 112 に直接伝送される。ここで筆者らは、他方のデータソース 120 として関係データベース 122 の例を用いる。

30

【0028】

データを処理するために組み合わせる方法の 1 つは、Hadoop クラスタ 112 のノード 114a~d に記憶されたデータ 116a~d を関係データベースにコピーすることである。次いで関係データベース 122 に、たとえば関係データベース 122 に固有のデータ処理動作（たとえば、SQL などのクエリ言語に従って定義されたデータベース動作）を使用して、組み合わせたデータに対して動作するように命じる。しかしながらこの技法では、Hadoop クラスタ 112 の並列処理機能は失われる。データを処理するために組み合わせる方法の 1 つは、関係データベース 122 に記憶されたデータ 124a~c のほとんどまたはすべてを Hadoop クラスタ 112 にコピーし、その後、Hadoop クラスタ 112 に固有の技法を使用して、たとえば前述の MapReduce プログラミ

40

50

ングモデルを使用して、データを処理することである。

【0029】

実装の様式に応じて、どちらの技法も、潜在的に大量のデータを一方のデータソースから他方のデータソースにコピーすることが必要な可能性があり、a)典型的には少なくともいくつかのデータをディスクに書き込むことが必要である、b)典型的には、動作を行うのに必要な処理時間に比べて、データをコピーするためにはかなり多くの処理時間が必要である、およびc)コピーされたデータは古くなるリスクがある、すなわち、動作が実施されている間はデータが変更されないことを保証するためのステップが行われない限り、コピーされたデータはそのソースに比べて旧式になる。これらの制限のすべてが、いずれの技法の性能および効率にも影響を与える。

10

【0030】

別の技法では、ほとんどまたはすべてのデータをそれぞれの固有のデータソース 110、120 に記憶することが可能であるため、ごく少量のデータがデータソース間でコピーされることになる。言い換えればデータ処理動作は、1つのタイプのデータソースのみを利用する動作を実施するのではなく、両方のタイプのデータソース 110、120 のリソースを利用するための技法を使用するように実施される。実際の例として、Hadoop クラスタ 112 のノード 114 a ~ d は、タスクを完了するために必要な何らかの動作（たとえば、データの一部に対して変換を行う動作）を行うことができ、関係データベース 122 は、同じくタスクを完了するために必要な何らかの他の動作（たとえば、データの別の部分に対して変換を行う動作）を実施することができる。

20

【0031】

これらの技法の例として、Hadoop クラスタ 112 のノード 114 a ~ d がデータ処理動作を行うとき、各ノード 114 a ~ d は動作を実施することが必要なデータのみ、たとえばそれに対して動作が行われるデータのみにアクセスする。たとえばいくつかのデータを、列および行を有するデータベーステーブルに記憶することができる。ある列または行に特定の動作のみが適用される場合、それらの列または行のみがデータソース間で、たとえば関係データベース 122 からノード 114 a ~ d のうちの 1 つにコピーされる。実際の例として、関係データベース 122 が電話会社の顧客を表すデータを記憶しており、データ処理システムが電話番号のリストを必要とする動作を行っている場合、ノード 114 a ~ d は電話番号を記憶している関係データベース 122 の列のみにアクセスし、顧客の名前、住所、またはデータベースに記憶され得る他のデータを表す列にはアクセスする必要がない。関係データベース 122 は、特定の動作、たとえば Hadoop クラスタのノード 114 a ~ d によって実施される特定の動作に必要な、データベースの部分のみを戻すために必要な動作を行うことができる。

30

【0032】

さらに、Hadoop クラスタ 112 の個々のノード 114 a ~ d は、各々、Hadoop クラスタ 112 によって記憶される全データの一部のみを記憶することができる。各ノード 114 a ~ d は、そのデータの部分に関して動作を実施するために必要なような追加のデータでもそれのみにアクセス可能であり、それらの動作を実施するために必要な他のデータにアクセスする必要はない。たとえばノード 114 a が、データのその部分と、異なるデータソースからの他のデータとの両方を使用する動作を行っている場合、ノードは、データのその部分に対して行われる動作に適用可能な他のデータのサブセットのみにアクセスする。

40

【0033】

実際の例として、データ処理システムは、顧客のマスタリストおよび通話記録のデータベースを有する電話会社に代わって、データの本体を管理することができる。この例では、クラスタ 112 のノード 114 a は、米国内で発信または受信された通話を表すデータ 116 a を記憶し、他のノード 114 b ~ d は、他のすべての国で発信または受信された通話を表すデータ 116 b ~ d を記憶することができる。クラスタ 112 のノード 114 a ~ d とは分離されている関係データベース 122 は、電話会社の顧客のリストを表

50

すデータ124を記憶することができる。（通話のデータベースの現実世界での実装は何百何千というノードが必要な可能性があるため、これは例としてのみ使用される。）

【0034】

この例では、データ処理動作は、Hadoopクラスタ112のノード114a～dによって少なくとも部分的に行うことができる。たとえば動作は、特定の通話に関連付けられた顧客を識別する動作とすることができます。米国内で発信および受信された通話を表すデータを記憶するノード114aには、米国内で電話サービスを受けている顧客記録を表すデータ124aを提供し、任意の他の国で電話サービスを受けている顧客を表すいずれのデータ124b、124cも提供しないことができる。いくつかの例では、関係データベース122からのデータ124a～cを、Hadoopクラスタ112のそれぞれのノード114a～dに提供することができる。いくつかの例では、Hadoopクラスタのそれぞれのノード114a～dは、関係データベース122にデータ124a～cの一部を要求することができる。したがって、ノード114aがアクセスするデータの量は、顧客のデータベース全体、たとえば関係データベース122によって記憶されたデータ124a～cのすべてに比べて少ない。いくつかの例では、ノード114a～dによって受信されたデータ124a～cを、（たとえばノード114a～dによって）ノード114a～dに記憶されたデータの形式に適合する形式に変換することができる。

10

【0035】

各動作が行われるときに、比較的小量のデータのみがそれぞれのデータソースから受信されるため、動作は、ディスクなどの持続的な（不揮発性）ストレージとは対照的に、アクティブな（揮発性）メモリに記憶されたデータに対して行うことが可能である。多くのコンピューティング環境において、持続性ストレージはアクティブメモリよりも低速になる傾向があるため、これによってデータ処理動作がスピードアップすることになる。

20

【0036】

いくつかの実装において、データ処理システム100はグラフベース処理システムとすることができる。グラフベース処理システムは、データフローグラフを使用してデータを処理する。データフローグラフは、データのフローを表すコンポーネント間の入力データおよびリンクに対して行われるべき動作を表すコンポーネントを含む、コンピュータプログラムである。（コンポーネントは時にはノードと言い表されるが、Hadoopクラスタのノードとの混同を防ぐために、ここではコンポーネントと呼ばれる。）コンポーネントによって表される動作は、入力データを処理することによって、入力データに基づき出力データを生成する。コンポーネントは、コンポーネントが他のコンポーネントにリンクしている場合、他のコンポーネントに入力データを提供し、他のコンポーネントから出力データを受信することができ、2つのコンポーネント間の各リンクはコンポーネントのうちの1つから他方のコンポーネントへのデータフローを表す。データフローグラフがグラフベース処理システムによって実行されるとき、コンポーネントの各々が実行され、たとえばコンピュータプログラムまたはコンピュータプログラムの一部が実行され、コンポーネントによって表される動作を実施する。実行中、データフローグラフは、出力データを生成するために処理される（たとえば、データフローグラフのコンポーネントの動作によって動作される）入力データを受信する。グラフベースシステムの一例は、「Managing Parameters for Graph-Based Applications」という名称の米国公開第2007/0011668号に詳細に記載され、参照により本明細書に組み込まれる。グラフベースの計算を実行するためのシステムは、「Executing Computations Expressed as Graphs」という名称の米国特許第5,966,072号に記載され、参照により本明細書に組み込まれる。

30

【0037】

グラフの実行は、時にはグラフオペレーティングシステムと呼ばれる、専用オペレーティングシステムによって容易となることがある。グラフオペレーティングシステムは、デ

40

50

ータフローグラフの個々のコンポーネントの基礎をなす動作を実行することができるコンピュータプログラムである。たとえば、データフローグラフのコンポーネントがデータベースシステムによって実施されるべき動作を表す場合、グラfovオペレーティングシステムには、動作を実施するようにデータベースシステムに命じるタスクが与えられる。このため、グラfovオペレーティングシステムは、時にはグラフベースデータ処理システムと対話するシステム上で実行する。図1に示された例では、グラfovオペレーティングシステム130a～dのインスタンスは、Hadoopクラスタ112のノード114a～d上で実行可能である。Hadoopクラスタのノード上でグラfovオペレーティングシステムを実行するための技法の例は、「Parallel Access to Data in a Distributed File System」という名称の米国出願第14/090,434号に記載されており、参照により本明細書に組み込まれる。
10

【0038】

グラfovオペレーティングシステム130a～dまたは任意の他の汎用データ処理システムを使用して、Hadoopクラスタ112のノードが他のデータソースからデータを受信できるようにすることが可能である。たとえばグラfovオペレーティングシステムは、関係データベース122からのデータの受信を可能にする場合がある。この例では、グラfovオペレーティングシステムのインスタンスが関係データベース122からデータを受信し、これを、グラfovオペレーティングシステムのインスタンスが実行しているHadoopノード114a～dの適切な部分またはサブシステムに提供することができる。このように、Hadoopクラスタのノード114a～dは、関係データベースなどの別の種類のデータソースからデータを受信するためにいずれのカスタム機能（たとえば特注コード）も必要としない。いくつかの例では、グラfovオペレーティングシステム130a～dは、特定のデータソースからいかにしてデータを受信および解析するかを記述した「プラグイン」を受信する機能を有する。Hadoopノード114aが関係データベース122からデータを受信する例では、Hadoopノード114a上で実行しているグラfovオペレーティングシステム130aのインスタンスは、関係データベース122から受信したデータをいかにして解析するかを決定するために「プラグイン」にアクセスすることができる。
20

【0039】

いくつかの実装において、Hadoopクラスタ112のノード114a～d上で実行するグラfovオペレーティングシステム130a～dのインスタンスは、関係データベース122の機能と通信する。たとえば関係データベース122は、グラfovオペレーティングシステムなどの外部エンティティが関係データベース122によって記憶されたデータにアクセスできるようにする関数（たとえばデータベースコマンド）をサポートすることができる。
30

【0040】

いくつかの実装において、データ処理システム100には、データベースクエリ140を実施するタスクが与えられる。データベースクエリは、データベースコンテンツのサブセットおよびそのサブセット内のデータ上で行うアクションを記述する、命令のセットである。データベースクエリ140が、電話会社に関するデータを記憶している前述のシステムで使用されるデータベースクエリである場合、データベースクエリ140は、電話会社によって使用されるデータソースに記憶された通話のある記録に関する要求とすることができます。たとえば、いくつかのデータベースシステムは、構造化クエリ言語（SQL）などの専用データベースクエリ言語で作成されたデータベースクエリを行う。これらのデータベースシステムにおいて、SQLクエリはデータベースのコンテンツを操作するための主な手段である。
40

【0041】

いくつかの実装において、データベースクエリ140はSQLクエリである。SQLクエリは、構造化クエリ言語によって定義されたコマンドおよび構文を使用する。関係データベース122は、1つまたは複数のデータベーステーブルの集合を含み、データベース
50

テーブルは、 a) 各々が記録を表す行、および b) 各々が行に記録されたデータのカテゴリを表す列、に配置構成されたデータの集合である。たとえば「 current_customers」と呼ばれるデータベーステーブルは、各々がビジネスの現在の顧客を表す行を有し得、顧客の名前、顧客の住所、顧客が最後に購入した製品などの、データのカテゴリを表す列を有し得る。

【 0 0 4 2 】

関係データベース 122 は典型的には、クエリを解釈し、クエリに応答してデータを戻すための機能を含む。クエリを解釈することとクエリに応答してデータを戻すことの組み合わせは、時にはクエリの実行と呼ばれる。たとえばいくつかの関係データベースの実装は、 a) SQL クエリを解析し、 b) 構造化クエリ言語によって定義される動作を識別し、 c) 演算子のオペランドを識別し、 d) オペランドに従って動作を実施する（たとえば実行する）、エンジンを含む。SQL クエリの例は、「 SELECT last_name FROM current_customers」とすることができる。この SQL クエリは動作 SELECT を含み、これは SELECT 動作のオペランドに従ってデータを取り出すように関係データベースに命令する。SQL の構文において、オペランドは、関係データベースによって管理されるデータベーステーブルである「 current_customers」と、データベーステーブルの列である「 last_name」である。関係データベースがクエリを解釈し、クエリの動作を実行するとき、関係データベースはクエリに応答して last_name 列のデータ（ last_name 列に含まれるデータの各部分）を戻す。

10

20

【 0 0 4 3 】

データ処理システム 100 は、たとえデータベースクエリ 140 で識別されたデータソースが、データベースクエリ 140 の形のクエリを使用して動作するデータベースでない場合であっても、データベースクエリ 140 を実施することができる。たとえば Hadoop クラスタ 112 は通常、SQL の形で指定された命令を受け入れない場合がある。データベースクエリ 140 が SQL クエリであり、 Hadoop クラスタ 112 を参照する場合、グラフオペレーティングシステム 130a ~ d のインスタンスは、データベースクエリ 140 に取り込む仲介として共に動作することができ、各インスタンスはそれに応答して Hadoop クラスタ 112 で行うべき動作を決定することができる。たとえばデータフローグラフのコンポーネントは、データベースクエリ 140 の命令の代用として使うことができる。この代用に従った技法は、「 Managing Data Queries」という名称の米国公開第 2012 / 0284255 A1 号により詳細に記載され、参考により本明細書に組み込まれる。いくつかの実装において、データフローグラフはデータベースクエリ 140 から作成可能である。

30

【 0 0 4 4 】

いくつかの実装において、グラフオペレーティングシステム 130a ~ d の各インスタンスはコンピュータプログラム 134a ~ d の対応する部分を実行する。たとえばコンピュータプログラムは、実行可能コンポーネントで構成することが可能であり、グラフオペレーティングシステム 130a ~ d の各インスタンスはコンピュータプログラムのコンポーネントのいくつかを実行することができる。グラフオペレーティングシステム 130a ~ d のインスタンスは、たとえば互いにデータを送受信することによって、コンピュータプログラムのそれぞれの部分を実行し、それによって共にコンピュータプログラムを実行するように、互いに調整可能である。いくつかの例では、グラフオペレーティングシステム 130a ~ d の複数のインスタンスが、コンピュータプログラムの同じコンポーネントのインスタンスを実行する。たとえば Hadoop クラスタ 112 のノード 114a ~ d 上で実行するコンピュータプログラム 134a ~ d のインスタンスは、その各々が異なるデータ（たとえばそれぞれのノード 114a ~ d によって記憶されたデータ 116a ~ d ）上で動作する、同じデータ処理コンポーネントのインスタンスを各々実行することができる。いくつかの例では、コンピュータプログラムの一部が共にデータフローグラフを構成し得、コンピュータプログラムの一部はデータフローグラフのサブグラフ（たとえ

40

50

ば1つまたは複数のリンクされたコンポーネント)とすることができます。いくつかの実装において、グラフオペレーティングシステム130a～dのインスタンスは、コンピュータプログラム134a～dを生成することができる。

【0045】

このように、Hadoopクラスタは、動作を実施するために、関係データベースの機能、たとえば関係データベースのクエリ解釈機能に依拠しない技法を使用して、データベースクエリ140の動作を実施することができる(データベースクエリ140を実行するととも呼ばれる)。その代わりにクエリは、コンピュータプログラム134a～dのインスタンスを実行することによって実施可能である。コンピュータプログラム134a～dが生成されると、Hadoopクラスタのノード114a～d上でデータベースクエリ140の動作を実施するために、関係データベースのクエリ解釈機能は使用されない。10

【0046】

いくつかの実装において、コンピュータプログラム(たとえばデータフローグラフ、または任意の他の種類のプログラム)をパラメータで構成することができる。たとえばパラメータは、プログラムの挙動を変更するために変更可能な値とすることができます。特定の例として、パラメータは「filename」であってよく、パラメータの値はファイルシステム内のファイルの位置とすることができます。異なるファイルにアクセスするようにプログラムを構成するために、パラメータの値を異なるファイルの位置に変更することができます。同じプログラムの2つのインスタンスを異なるパラメータ値で構成することが可能であり、これによって同じプログラムの2つのインスタンスの挙動が変更される。20

【0047】

図1のシステムは、1つまたは複数のネットワークを使用して互いに通信することができます。たとえばHadoopクラスタ112のノード114a～dは、ローカルエリアネットワーク(LAN)などのネットワークを使用して互いに通信可能であるが、ワイドエリアネットワーク(WAN)、インターネット、または別の種類のネットワークを使用して互いに通信してもよい。さらにHadoopクラスタ112のノード114a～dは、LAN、WAN、インターネット、または、コンピュータシステム間での通信をサポートする任意の他の種類の通信ネットワークを使用して、関係データベース122および処理システム100と通信することができます。30

【0048】

さらに、図1には単一のHadoopクラスタ112が示されているが、図に示されるシステムでは複数のHadoopクラスタが使用できる。たとえば1つのHadoopクラスタは関係データベース122から何らかのデータを受信し、別のHadoopクラスタは関係データベース122から他のデータを受信することができる。複数のHadoopクラスタを含む他の構成が可能である。

【0049】

図2は、データフローグラフ200の例を示す。いくつかの例では、データフローグラフ200は、データフローグラフを閲覧、構成、および/または実行することができるユーザインターフェース内に表示可能である。このデータフローグラフ200は、図1に示されたHadoopクラスタ112およびデータ処理システム100によって行うことができるデータ処理動作を表す。この例では、データフローグラフは「結合」と呼ばれる動作を表すコンポーネント202を含み、結合コンポーネント202と呼ぶこともある。「結合」動作は2つのタイプのデータを組み合わせるものであり、たとえば1つのタイプのデータは1つのデータソースに格納され、別のタイプのデータは別のデータソースに格納されている。データフローグラフ200の他のコンポーネントは、Hadoopクラスタである1つのデータソースを含む、複数タイプのデータソースを使用して、データフローグラフ200が結合動作を実施できるようにする。さらにほとんどの処理は、Hadoopクラスタのノード上で実行される。40

【0050】

いくつかの例では、データフローグラフ200は、たとえば図1に示されたデータベー50

スクエリ 140などのデータベーススクエリから作成され得る。たとえばデータフローグラフ 200は、データベーススクエリを入力として受け取り、データフローグラフを出力として作成するエンジン（たとえば図1に示されるグラフォペレーティングシステム 130a～dのインスタンス）を使用して生成可能である。このように、図2に示されるデータフローグラフ 200などのデータフローグラフは、実行されたとき、対応するデータベーススクエリ 140（図1）の実行と同じ出力を作成することができる。このように、データベーススクエリ 140はSQLなどのデータベーススクエリ言語を使用して作成可能である。しかしながら、対応するデータ処理動作を実施するシステム、たとえばHadoopクラスタ 112は、データベーススクエリ 140を解析できる必要はない。たとえば、カスタム設計のデータベーススクエリ解析機能をHadoopクラスタ 112に提供する必要はない。
代わりに、Hadoop ノード 114a～d上で実行するグラフォペレーティングシステム 130a～dのインスタンス（図1）は、組み合わせて、データベーススクエリ 140の動作と等価である（たとえば、従来の関係データベースシステムにおいてデータベーススクエリ 140を実行することと同じ結果を達成する）動作を行うために、データフローグラフ 200を実行することができる。たとえば、データフローグラフ 200がグラフォペレーティングシステム 130a～dのインスタンスによって実行されるとき、データフローグラフ 200の出力は、データベーススクエリ 140を実行するがデータフローグラフ 200は実行しないシステム（Hadoop クラスタ 112以外）の出力と等価である。このように、データフローグラフ 200は、データベーススクエリ 140に対応するコンピュータプログラムの例である。
10
20

【0051】

データフローグラフ 200のコンポーネントは、データフローグラフ 200が、Hadoop クラスタを表すデータソースを含む、複数タイプのデータソースからのデータを処理することができるよう、配置構成される。1つのコンポーネント 204はHadoop クラスタ 112によって記憶されるデータを表し、別のコンポーネント 206は関係データベース 122によって記憶されるデータを表し、どちらも図1に示されている。Hadoop クラスタコンポーネント 204は結合コンポーネント 202にリンクされ、これはデータがHadoop クラスタコンポーネント 204から結合コンポーネント 202へと流れるため、Hadoop クラスタコンポーネント 204の出力が入力として結合コンポーネント 202に提供されることを意味する。さらに、Hadoop クラスタコンポーネント 204は、Hadoop クラスタのノードによって並列に実施可能な動作を表す。たとえばデータがHadoop クラスタコンポーネント 204から流れるとき、データの複数部分、たとえばノードによって処理されたデータが、Hadoop クラスタノードから同時に流れることができる。同様に、Hadoop クラスタコンポーネント 204によって行われる動作は、各々がHadoop クラスタノード上で行われる複数の同時動作の形で行われることができる。この技法は、並列処理と呼ばれることがある。図に示されている番号「4」は、下にあるHadoop クラスタ内のノードの数、したがって、並列処理のために動作を分割できる数を示す。
30

【0052】

データは、関係データベースコンポーネント 206から、データがHadoop クラスタの個々のノードに直接提供できるように流れる。データは、関係データベースコンポーネント 206からブロードキャストコンポーネント 210へと流れる。次いでデータは、結合コンポーネント 202へと流れる。この例では、結合コンポーネント 202の動作は、Hadoop クラスタの並列処理機能を利用するため、Hadoop クラスタコンポーネント 204によって表されるHadoop クラスタのノードによって記憶されたデータに対して実施される。
40

【0053】

ブロードキャストコンポーネント 210は、関係データベースによって記憶されたデータをHadoop クラスタコンポーネント 204によって表された各々のノードに伝送できるようにする、動作を表す。この例は関係データベースを使用しているが、図2に示さ
50

れる他のコンポーネントと同様に、ブロードキャストコンポーネント 210 は他の種類のデータソースと共に使用することができる。いくつかの実装において、ブロードキャストコンポーネント 210 は、結合動作を Hadoop クラスタの各ノード内でローカルに行えるようにするために、クラスタにわたるデータの各部分の狭められたハッシュテーブルをコピーする。このように、データが関係データベースコンポーネント 204 から結合コンポーネント 202 へと流れるとき、図 1 に示された関係データベース 122 に記憶された下にあるデータ 124a～c の一部を、Hadoop クラスタ 112 の特定のノード 114a～d に向けて送ることができる。このプロセスはファンアウト (fan-out) インジケータ 212 によって表され、これはデータが並列処理のために分割される（または「広がる」）ことを示す。いくつかの例では、それぞれのデータソース（たとえば関係データベース 122 および Hadoop クラスタ 112）に記憶されたデータを分析して、データフローグラフ 200 内で処理されるときなどにデータを分割する最適な様式を決定することができる。いくつかの例では、Hadoop クラスタのノード 114a～d 上で実行するグラフオペレーティングシステム 130a～d のインスタンスは、データフローグラフ 200 内で処理されるときなどにデータを分割する最適な様式を決定し、関係データベース 122 にデータの一部を要求することができる。一般にデータは、Hadoop ノードが各々、それぞれの Hadoop ノードのアクティブメモリに記憶できる量のデータ（たとえば関係データベースのいくつかの行および / または列のみ）を受信するように分割される。
10

【0054】

20

この例では、データは結合コンポーネント 202 からロールアップコンポーネント 214 へと流れる。ロールアップコンポーネントは、複数のソースからのデータを集約する。結合コンポーネント 202 は Hadoop クラスタの複数のノードによって実施される動作を表すため、ロールアップコンポーネント 214 は複数のノードからの出力を集約する。ファンインインジケータ 216 は、データフローグラフ 200 内のこの時点で、複数の並列フローの形でグラフを通って流れるデータが单一のフローに統合されることを示す。ファンインインジケータ 216 の後に現れるコンポーネント 218 は、Hadoop クラスタのノード以外のエンティティ（たとえば、図 5 に示されるデータ処理サブシステム 101）によって実施され得る動作を表す。

【0055】

30

データフローグラフの各コンポーネントは、レイアウトインジケータでマーク付けされている。レイアウトは、コンポーネントによって表される動作を実施する特定のシステムを指す。図に示されるように、いくつかのコンポーネントはレイアウト 1 とマーク付けされ、いくつかのコンポーネントはレイアウト 2 とマーク付けされ、いくつかのコンポーネントはレイアウト 3 とマーク付けされている。ここで、レイアウト 1、レイアウト 2、およびレイアウト 3 とマーク付けされたコンポーネントは、Hadoop クラスタ 112 に関連付けられたグラフオペレーティングシステム 130a～d のインスタンスによって実施される動作を表す。いくつかの例では、レイアウト 2 とマーク付けされたコンポーネントは、関係データベース 122 または関係データベース 122 に関連付けられたグラフオペレーティングシステムによって実施される動作を表す。いくつかの例では、レイアウト 3 とマーク付けされたコンポーネントは、Hadoop クラスタ 112 または関係データベース 122 以外のシステム、たとえば図 5 に示されるデータ処理サブシステム 101 などのシステムによって実施される動作を表す。
40

【0056】

したがって、データフローグラフ 200 が実行されるとき、データフローグラフ 200 の動作は、データ処理の多くが Hadoop ノードで生じるように、Hadoop クラスタに関連付けられたコンピュータシステムによって実施可能である。このように、Hadoop クラスタの並列処理機能が使用される。さらに、個々の Hadoop ノードにコピーされたデータ量はアクティブメモリ内に保持できるため、データをディスクにコピーする必要はなく、ディスクの読み / 書きによって生じる性能の減速は緩和される。
50

【0057】

図3は、データ処理システム100のシステムによって共に実施可能な別の一連の動作を表す、データフローグラフ300を示す。このデータフローグラフ300は、「付加」動作を実施するコンポーネント302を含む。「付加」動作は、1つのデータ量を別のデータ量に付加し、統合されたデータ量を形成する。このデータフローグラフ300は、ほとんどのデータ処理がHadoopノード上で生じる一連の処理動作の別の例を表す。

【0058】

データフローグラフ300のコンポーネントは、Hadoopクラスタを表すデータソースを含む複数タイプのデータソースからのデータを、データフローグラフ300が処理できるように配置構成される。1つのコンポーネント304はHadoopクラスタ112によって記憶されるデータを表し、別のコンポーネント306は関係データベース122によって記憶されるデータを表し、どちらも図1に示されている。10

【0059】

Hadoopクラスタコンポーネント304はフィルタコンポーネント308にリンクされ、フィルタコンポーネント308は付加コンポーネント302にリンクされる。したがって、データはHadoopクラスタコンポーネント304から流れ、付加コンポーネント302によって処理される前にフィルタコンポーネント308によってフィルタリングされる。入力データは、入力データの特徴ならびに各Hadoopノード上に記憶されたデータの特徴に基づいてフィルタリング可能である。たとえば、動作が商取引を表すデータに対して行われるものであり、特定のHadoopノードが合計で10ドルを超える取引に関するデータのみを記憶する場合、フィルタコンポーネント308は、(たとえば、コンポーネントの動作を制御するようにパラメータを修正することによって)合計で10ドルを超える購買に関連する入力データを渡すように構成することができる。別の例として、動作が商取引を表すデータに対して行われるものであり、動作自体が合計で10ドルを超える購買にのみ関連している場合、フィルタコンポーネント308は、(たとえば、コンポーネントの動作を制御するようにパラメータを修正することによって)合計で10ドルを超える取引に関連する入力データを渡すように構成することができる。20

【0060】

このデータフローグラフ300において、関係データベースコンポーネント306から流れるデータは、Hadoopクラスタのノードでの処理のために分割される。関係データベースコンポーネント306は区分コンポーネント310にリンクされ、区分コンポーネントは関係データベースコンポーネント306から流れるデータを区分(たとえば分割)する。たとえば、区分コンポーネント310はラウンドロビンと呼ばれる技法を使用し、この技法では、区分コンポーネント310によって区分されたデータの各新規部分が、固定シーケンスでHadoopクラスタのノードに提供される。言い換えれば、ラウンドロビン技法では、データの一部がノードに順番に分配される。このように、Hadoopクラスタの各ノードは、ノードのアクティブメモリ内に保存可能であり、ディスクに書き込む必要のないデータの一部を受信する。30

【0061】

したがって付加コンポーネント302は、各ノードが関係データベースコンポーネント306から受信したデータを、Hadoopクラスタの個々のノードに記憶されたデータに付加するように、Hadoopクラスタのノードによって実施される動作を表す。これらのすべての付加動作の結果はロールアップコンポーネント312に並列に提供され、ロールアップコンポーネント312はこの結果をさらに処理するために集約する。たとえば集約された出力は、その先のコンポーネント314によって処理することができる。ここでも、Hadoopクラスタのノード上で動作がディスクを大量に使用する必要のないように、ほとんどのデータ処理動作はHadoopクラスタ上で生じる。40

【0062】

図4は、データを処理するための手順400を表すフローチャートを示す。手順400は、たとえば図1に示されたデータ処理システム100のコンポーネントによって実施可50

能である。

【0063】

手順400は、HadoopクラスタのHadoopノードで、データ処理エンジンのインスタンスを実行する402。Hadoopノードはデータの第1の部分をHDFSデータストレージに記憶する。データ処理エンジンは、Hadoopクラスタ外部のデータソースからデータを受信することができる。たとえばノードは、図1に示されるHadoopクラスタ112のノード114a～dのうちの1つとすることができる。たとえばデータ処理エンジンは、図1に示されたグラフオペレーティングシステム130a～fとすることができる。

【0064】

データ処理エンジンの他のインスタンスは、Hadoopクラスタの他のノード上で実行可能であり、その各々が手順400の動作のそれら自身のインスタンス（たとえば動作402～412）を行う。いくつかの実装において、データ処理エンジンはMapReduceuceプログラミングモデルを実装しない。

【0065】

手順400はプログラムを受信する404。プログラムは、データ処理エンジンによって受信され得る。たとえばプログラムは、Hadoopクラスタを表す少なくとも1つのコンポーネント、受信したデータのソースを表す少なくとも1つのコンポーネント、および、少なくとも1つのデータソース（たとえばHadoopクラスタまたは別のデータソース）から受信したデータに対して行われるべき動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクを含む、データフローグラフとすることができる。グラフは、Hadoopクラスタからのデータフローを表すリンクに接続され、手順400で受信したデータのソースからのデータフローを表すリンクに接続された、少なくとも1つのコンポーネントを含むことができる。

【0066】

いくつかの実装において、コンピュータプログラムは、データベースクエリ、たとえばSQLクエリに対応する動作を表すコンポーネントを含む。データベースクエリを表すコンピュータプログラムは、データベースクエリ（たとえばHadoopクラスタを参照するデータベースクエリ）内で参照されるデータのソースを表す少なくとも1つのコンポーネント、および、データに対して行われるべき動作に関連付けられた少なくとも1つのデータフローを表す少なくとも1つのリンクを含む。たとえば動作は、手順400内で行われるデータ処理動作とすることができます。

【0067】

手順400は、コンピュータプログラムの少なくとも一部を実行する406。たとえばコンピュータプログラムの一部は、Hadoopクラスタ内のHadoopノード上で実行するデータ処理エンジンによって実行可能である。いくつかの例では、実行されるコンピュータプログラムの一部は、Hadoopクラスタを表す少なくとも1つのコンポーネント、およびデータ処理動作を表す少なくとも1つのコンポーネントを含む。たとえば実行される406コンピュータプログラムのコンポーネントは、コンピュータプログラムのレイアウトに含めることができる。いくつかの例では、ここで実行される406コンピュータプログラムの同じコンポーネントのインスタンスは、Hadoopクラスタの他のノード内で同時に実行される。いくつかの実装において、コンピュータプログラムは実行される前に構成される。たとえばコンピュータプログラムのコンポーネントは、変更可能な値を有するパラメータで構成可能である。いくつかの例では、コンピュータプログラムは別のプログラムにリンクされる。たとえば、コンピュータプログラムがグラフである場合、グラフを別のグラフ（たとえば、グラフオペレーティングシステム上で実行するかまたは使用可能なグラフ）にリンクすることができる。

【0068】

手順400は、外部のデータソースからデータの第2の部分を受信する408。たとえば外部データソースは、Hadoopクラスタのノード以外のソース、たとえば図1に示

10

20

30

40

50

されたデータソース 120（たとえば、関係データベース 122）とすることができます。受信されたデータは、Hadoop クラスタの他のノードで受信されたデータの他の部分（たとえば、データの他のソースから受信されたデータの他の部分）とは別個である。いくつかの例では、データの第 2 の部分は関係データベースの行のサブセットを含み、データの第 2 の部分は関係データベースの列のサブセットを含む。いくつかの例では、データの第 2 の部分は、外部データソースから Hadoop クラスタの第 2 のノードで受信したデータの第 3 の部分とは別個である。言い換えればデータの第 2 の部分は、データの第 3 の部分とは異なるデータ、たとえば異なる行および／または列を含む。

【0069】

手順 400 は、データの第 2 の部分を HDFS ストレージ以外に記憶する 410 ため、これはデータの第 2 の部分が HDFS ストレージ（たとえば、データの第 1 の部分を含む HDFS ストレージ）に記憶されないことを意味する。たとえばデータの第 2 の部分は、Hadoop ノードの揮発性メモリに記憶することができる。揮発性メモリは、ランダムアクセスメモリと呼ばれることがある。これに対して不揮発性メモリは、たとえばディスクドライブである。手順 400 によって受信される 408 データは、Hadoop クラスタのノードの揮発性メモリに収まるサイズを有し得る。10

【0070】

手順 400 は、少なくともデータの第 1 の部分およびデータの第 2 の部分を使用して、プログラムによって識別されたデータ処理動作を行う 412。データ処理動作は、少なくとも部分的に、データ処理エンジンの他のインスタンスと協働するデータ処理エンジンのインスタンスによって実施可能である。データ処理エンジンのインスタンスは共に並列でデータ処理動作を行うため、これはデータの異なる部分に対して同じデータ処理動作を行うために、データ処理エンジンのインスタンスが同時に実行することを意味する。「同時に実行すること」は、データ処理エンジンの 1 インスタンスが（たとえば、コンピュータプログラムの一部において）動作の集合の実施を開始する時点が、データ処理エンジンの別のインスタンスが同じ動作の集合の実施を開始する時点に依存せず、データ処理エンジンの両方のインスタンスで、同じ動作の少なくともいくつかは同時に、または互いに数ミリ秒以内に実施され得ることを意味する。いくつかの例では、インスタンスは、Hadoop クラスタのノードによって記憶されたデータの本体およびデータの別の本体に対してデータ処理動作を共に行うことができる。いくつかの例では、データの他の本体は関係データベースシステムに固有の形式（たとえば、行および列を含むテーブルの形、または関係データベースシステムのデフォルト形式である別の形）で記憶することができる。20

【0071】

いくつかの例では、手順 400 は、Hadoop クラスタ外部のデータ処理エンジンの第 2 のインスタンスによって、プログラムの少なくとも一部を実行する。たとえばデータ処理エンジンは、図 1 に示された他方のデータソース 120（たとえば、関係データベース 122）上で実行するグラフオペレーティングシステム 130f とすることができる。図には、Hadoop ノード上で実行するグラフ処理エンジンのインスタンスと、Hadoop クラスタ外部で実行するデータ処理エンジンのインスタンスとの間の通信を表すものとして、両矢印が示されている。いくつかの実装において、Hadoop クラスタのノード、たとえば手順の動作 402～412 を実施するノードは、Hadoop クラスタ外部のデータ処理エンジンの第 2 のインスタンス（たとえば、他方のデータソース 120 上で実行するデータ処理エンジンのインスタンス）によって実行されるプログラムの少なくとも一部と通信する。たとえば、プログラムの少なくとも一部（たとえば、プログラムの 1 つまたは複数のコンポーネント）は、Hadoop クラスタのノードとの間でデータを送受信することができる。30

【0072】

Hadoop クラスタのノードによって受信されるデータの一部は、ノード上に記憶されたデータの一部の特徴に基づいて選択することができる。たとえば、ノードによって受信されるデータの一部は、他のノードとは反対に、その特定のノード上でデータ処理動作4050

を実施するために必要となるデータに基づいて選択することができる。ノードによって受信されるデータの一部が関係データベースからのものである場合、データの一部は、関係データベースからのいくつかの列のみおよび／またはいくつかの行のみを含む場合がある。いくつかの例では、関係データベースは、特定のノード上に記憶されたデータの部分を識別する情報に基づいて、特定ノードを宛先とする出力データをフィルタリングするフィルタリング動作を行うことができる。

【0073】

図5は、1つまたは複数のデータソース110、120から発信されるデータ102、103が、データ処理サブシステム101によって処理される、データ処理システム100aの別のバージョンを示す。データ処理サブシステム101はデータに対して動作131を行い、処理されたデータ132を1つまたは複数の出力150に提供する。このデータ処理システム101は、複数タイプのデータソース110、120からのデータ102、103を処理し、データに対して行われる動作が、いずれのデータ102、103がいずれのタイプのデータソース110、120から着信したかによって制限されないように、データを処理することができる。これを遂行する様式の1つによって、データソース110のうちの1つ（たとえば、Hadoopクラスタ112）が他のデータソース120（たとえば、関係データベース122）からデータ104を受信し、受信したデータ104をデータソース110に固有の技法を使用して処理することが可能となる。このように、普通であればデータ処理システム101によって行われることになるデータ処理の多くは、代わりにデータソース110によって行われる。

【0074】

Hadoopクラスタ112は、他のタイプのデータソースと共に、データ処理システム101への入力データソース110として指定され得る。他方のデータソース120は、関係データベース、フラットファイル、ネットワークリソースからのデータのフィード、または、データ処理システムからの要求に応答してデータを提供できる任意の他のリソースなどの、データソースとすることができます。次いでデータ処理サブシステム101は、Hadoopクラスタ112からのデータ102と別のデータソース120からのデータ103との組み合わせに対して動作を行うことができる。Hadoopクラスタ112または他のデータソース120からデータを抽出するのではなく、データ処理サブシステム101は、Hadoopクラスタ112のデータ処理機能に依拠することができる。これは、Hadoopクラスタ112のデータ処理機能を利用して行うことができる。このように、データ処理サブシステム101は、Hadoopクラスタ112に比べて少ない動作を、かなり少量のデータに対して実施することができる。たとえば図5に示されるように、他方のデータソース120から受信するデータ104のほとんどは、Hadoopクラスタ112に（たとえば図1に関して上記で説明した技法を使用して）直接伝送され、少量のデータ103のみ（場合によっては全くなし）がデータ処理サブシステム101に伝送される。

【0075】

実際の例として、データ処理サブシステム101は、Hadoopクラスタ112のノード114a～dに、タスクを完了するために必要な何らかの動作（たとえば、データの一部に対して変換を行う動作）を行うように命令すること、および、関係データベース122に、同じくタスクを完了するために必要な何らかの他の動作（たとえば、データの別の部分に対して変換を行う動作）を実施するように命令することができる。

【0076】

これらの技法の例として、データ処理サブシステム101がデータ処理動作を行うとき、データ処理サブシステム101は動作を実施することが必要なデータのみ、たとえばそれに対して動作が行われるデータのみにアクセスする。他のデータ処理動作は、たとえばHadoopクラスタ112で実施することができる。

【0077】

いくつかの実装において、データ処理サブシステム101は、データを処理するために

10

20

30

30

40

グラフを実行する、グラフベースのデータ処理システムである。たとえばデータ処理サブシステム 101 は、データ処理動作を含む 1つまたは複数のコンピュータプログラム 134e を実行する、グラフォペレーティングシステム 130e のインスタンスを含むことができる。

【0078】

いくつかの実装において、グラフォペレーティングシステム 130f のさらなるインスタンスは、関係データベース 122 に関連して実行可能である。たとえばグラフォペレーティングシステム 130f のさらなるインスタンスは、関係データベース 122 を実行している同じコンピュータシステム（またはシステムの組み合わせ）上で実行可能であるか、またはグラフォペレーティングシステム 130f のさらなるインスタンスは、関係データベース 122 を実行しているコンピュータシステム（またはシステムの組み合わせ）と通信している別のコンピュータシステム 123 上で実行可能である。いくつかの実装において、グラフォペレーティングシステム 130f のさらなるインスタンスは使用されない。グラフォペレーティングシステム 130f のこのインスタンスは任意選択であるため、図 1 では破線で表されている。いくつかの実装において、Hadoop クラスタ 112 のノード 114a ~ d 上で実行するグラフォペレーティングシステム 130a ~ d のインスタンスは、関係データベース 122 に関連して実行しているグラフォペレーティングシステム 130f のさらなるインスタンスと通信する。

【0079】

いくつかの実装において、グラフォペレーティングシステム 130a ~ f の各インスタンスは、コンピュータプログラム 134a ~ f の対応する部分を実行する。たとえばコンピュータプログラムは実行可能コンポーネントで構成され得、グラフォペレーティングシステム 130a ~ f の各インスタンスは、コンピュータプログラムのコンポーネントのいくつかを実行することができる。グラフォペレーティングシステム 130a ~ f のインスタンスは、たとえば、コンピュータプログラムのそれらそれぞれの部分を実行し、したがってコンピュータプログラムをまとめて実行するために、互いにデータを送受信することによって互いに調整することができる。いくつかの例では、グラフォペレーティングシステム 130a ~ f の複数のインスタンスは、コンピュータプログラムの同じコンポーネントのインスタンスを実行する。たとえば、Hadoop クラスタ 112 のノード 114a ~ d 上で実行しているコンピュータプログラム 130a ~ d のインスタンスは、各々、それぞれが異なるデータ（たとえば、それぞれのノード 114a ~ d によって記憶されたデータ 116a ~ d）に対して動作する、同じデータ処理コンポーネントのインスタンスを実行することができる。いくつかの例では、コンピュータプログラムの一部が共にデータフローグラフを構成することができ、コンピュータプログラムの一部はデータフローグラフのサブグラフ（たとえば、1つまたは複数のリンクされたコンポーネント）とすることができます。いくつかの例では、グラフォペレーティングシステム 130a ~ f のインスタンスによって実行されるコンピュータプログラムまたはコンピュータプログラムの一部は、データ処理サブシステム 101 によって受信されるか、データ処理システム 100a の別のコンポーネントによって受信される、データベースクエリ 140 から生成される。前述の手法は、好適なソフトウェアを実行しているコンピューティングシステムを使用して実装可能である。たとえばソフトウェアは、各々が、少なくとも 1つのプロセッサ、少なくとも 1つのデータストレージシステム（揮発性および / または不揮発性のメモリ、ならびに / あるいはストレージ要素を含む）、少なくとも 1つのユーザインターフェース（少なくとも 1つの入力デバイスまたはポートを使用して入力を受信するため、および、少なくとも 1つの出力デバイスまたはポートを使用して出力を提供するため）を含む、1つまたは複数のプログラミングされたまたはプログラミング可能なコンピューティングシステム（分散型、クライアント / サーバ、またはグリッドなどの、様々なアーキテクチャであり得る）上で実行する、1つまたは複数のコンピュータプログラムに手順を含めることができる。ソフトウェアは、たとえばデータフローグラフの設計、構成、および実行に関するサービスを提供する、より大きなプログラムの 1つまたは複数のモジュールを含むこと

10

20

30

40

50

ができる。プログラムのモジュール（たとえば、データフローグラフの要素）は、データリポジトリに記憶されたデータモデルに準拠するデータ構造または他の組織的データとして実装可能である。

【 0 0 8 0 】

ソフトウェアは、C D - R O Mまたは他のコンピュータ読み取り可能媒体（たとえば汎用または特定用途向けのコンピューティングシステムまたはデバイスによって読み取り可能）などの、有形の持続性媒体上に提供すること、または、それが実行されるコンピューティングシステムの有形の持続性媒体へネットワークの通信媒体を介して送達する（たとえば伝搬信号内に符号化する）ことができる。処理の一部またはすべては、特定用途向けコンピュータ上で、あるいは、コプロセッサ、フィールドプログラマブルゲートアレイ（FPGA）、または専用の特定用途向け集積回路（ASIC）などの、特定用途向けハードウェアを使用して、行うことができる。処理は、ソフトウェアによって指定された計算の異なる部分が異なるコンピューティング要素によって行われる分散様式で実装することができる。こうした各コンピュータプログラムは、好ましくは、本明細書で説明する手順を行うために、ストレージデバイスがコンピュータシステムによって読み取られたときに、コンピュータを構成および動作するために、汎用または特定用途向けプログラマブルコンピュータによって読み取り可能なストレージデバイス（たとえば、ソリッドステートメモリまたは媒体、あるいは磁気または光媒体などの、持続性ストレージデバイス）上に記憶されるか、またはダウンロードされる。本発明のシステムは、コンピュータプログラムで構成される有形の持続性媒体として実装されるものとみなすことも可能であり、媒体がそのように構成されることで、本明細書で説明する処理ステップのうちの1つまたは複数を行うための特有な事前に定義された様式でコンピュータを動作させる。10

【 0 0 8 1 】

以上、本発明のいくつかの実施形態について説明してきた。ただし、前述の説明は例示的であり、以下の特許請求の範囲によって定義される本発明の範囲に限定されないことを理解されよう。したがって、他の実施形態も以下の特許請求の範囲内にある。たとえば、本発明の範囲を逸脱することなく、様々な修正が実行可能である。加えて、前述のステップのうちのいくつかは順序に依存しないものであり得るため、説明した順序とは異なる順序で行うことができる。20

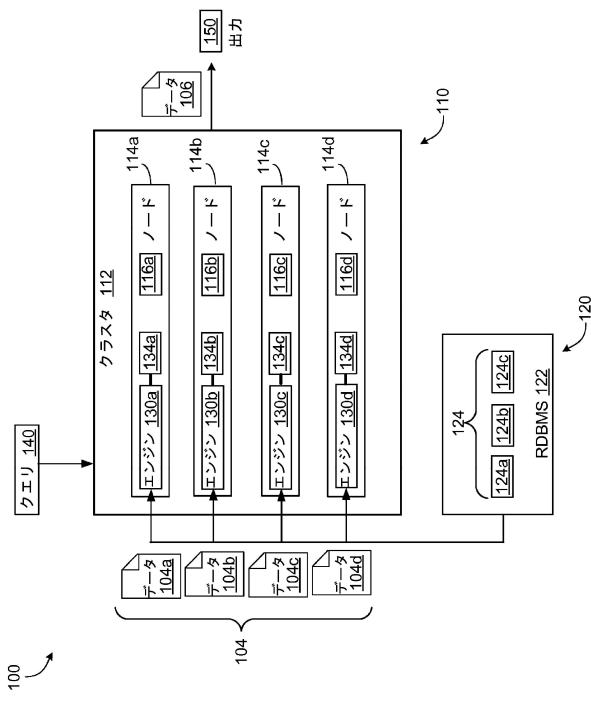
【 0 0 8 2 】

たとえば、上記の例は、H a d o o p クラスタ 1 1 2 または関係データベース 1 2 2 とは別個のデータ処理システム 1 0 0 を示しているが、いくつかの実装において、データ処理システム 1 0 0 は実際には、たとえば、グラフオペレーティングシステム 1 3 0 a ~ e のインスタンスとして、H a d o o p クラスタ 1 1 2 および / または関係データベース 1 2 2 にわたって機能的に分散されることが可能である。30

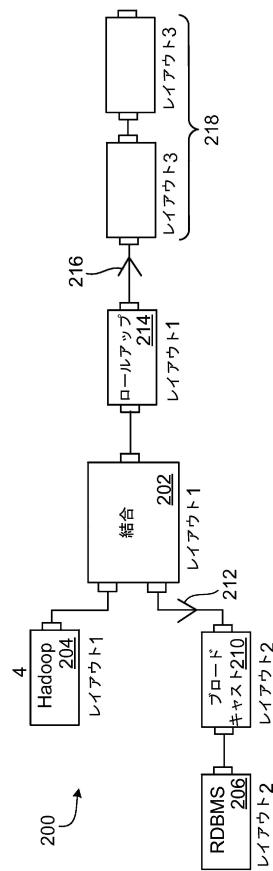
【 0 0 8 3 】

別の例として、図 1 ~ 4 に示された例は、単一のH a d o o p クラスタおよび単一の関係データベースの例を使用しているが、本明細書で説明する技法は、1つのH a d o o p クラスタから受信したデータ、および別の離れたH a d o o p クラスタから受信したデータに対して動作させるために使用することもできる。40

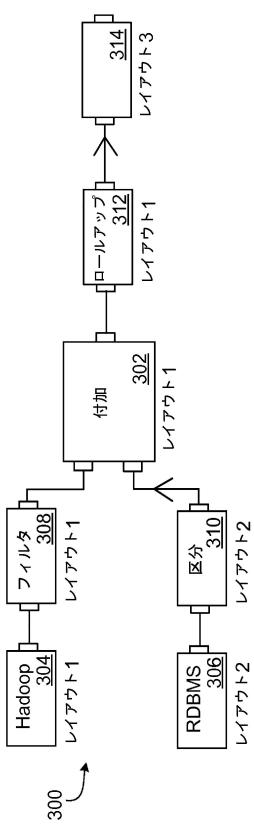
【図1】



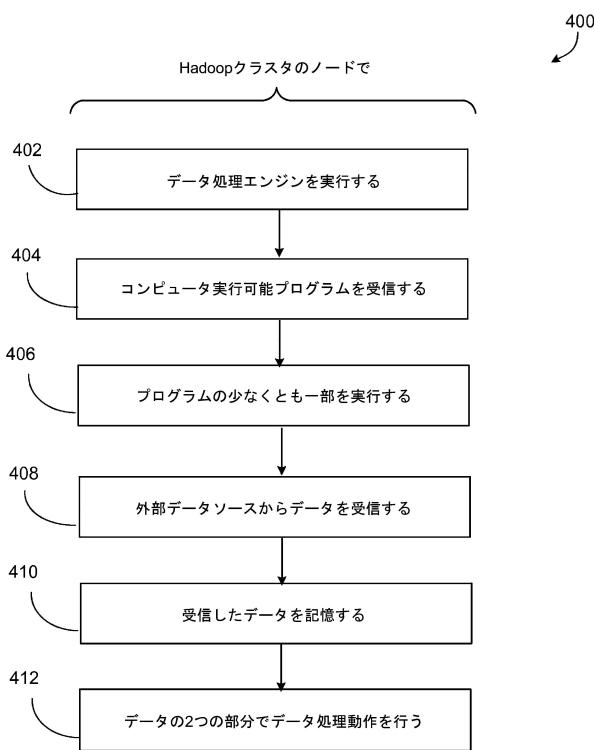
【図2】



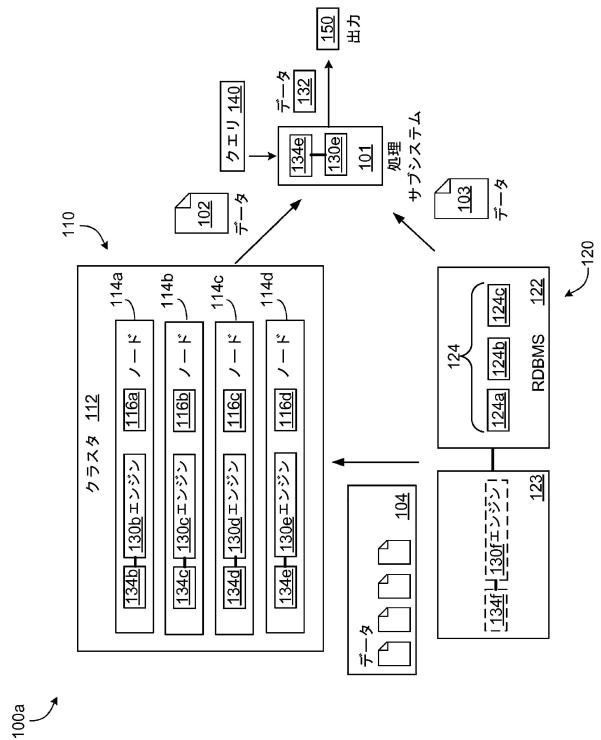
【図3】



【図4】



【図5】



フロントページの続き

(72)発明者 シェヒター , イアン

アメリカ合衆国 , マサチューセッツ州 02067 , シャロン , ブルック ロード 94

(72)発明者 ウェイクリング , ティム

アメリカ合衆国 , マサチューセッツ州 01810 , アンドーバー , アボット ストリート 11

(72)発明者 ウォルラス , アン エム .

アメリカ合衆国 , マサチューセッツ州 01450 , グロトン , ノースウッズ ロード 9

審査官 加藤 優一

(56)参考文献 国際公開第2012/151149 (WO , A2)

米国特許出願公開第2013/0198120 (US , A1)

米国特許出願公開第2013/0325814 (US , A1)

特開2014-78085 (JP , A)

上田 晴康 , 外2名 , Net COBOLのHadoop連携機能の開発と実践事例 , 情報処理学会デジタルプラクティス Vol. 5 No. 2 , 日本 , 一般社団法人情報処理学会 , 2014年

4月15日 , 第120頁 - 第129頁

(58)調査した分野(Int.Cl. , DB名)

G 06 F 9 / 50

G 06 F 16 / 27