

(19)日本国特許庁(JP)

(12)公表特許公報(A)

(11)公表番号
特表2024-521730
(P2024-521730A)

(43)公表日 令和6年6月4日(2024.6.4)

(51)国際特許分類

F I

G 0 6 F 16/182 (2019.01)

G 0 6 F 16/182

G 0 6 F 9/50 (2006.01)

G 0 6 F 9/50 1 5 0 Z

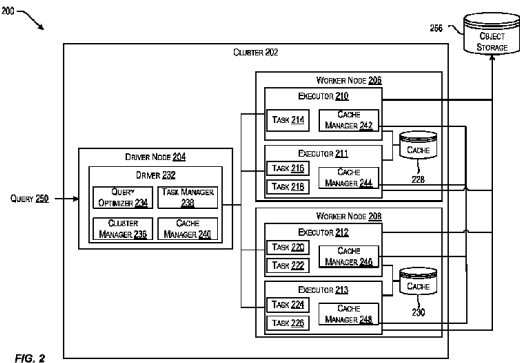
審査請求 未請求 予備審査請求 未請求 (全48頁)

(21)出願番号	特願2023-571975(P2023-571975)	(71)出願人	502303739
(86)(22)出願日	令和4年5月20日(2022.5.20)		オラクル・インターナショナル・コーポレーション
(85)翻訳文提出日	令和6年1月18日(2024.1.18)		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、オラクル・パークウェイ500
(86)国際出願番号	PCT/US2022/030337		
(87)国際公開番号	WO2022/246253		
(87)国際公開日	令和4年11月24日(2022.11.24)		
(31)優先権主張番号	202141022725	(74)代理人	110001195
(32)優先日	令和3年5月21日(2021.5.21)		弁理士法人深見特許事務所
(33)優先権主張国・地域又は機関	インド(IN)	(72)発明者	カバリ, ディーバラジュル
(81)指定国・地域	AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,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,最終頁に続く		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、オラクル・パークウェイ、500、オラクル・インターナショナル・コーポレーション
		(72)発明者	マールケード, アニーシュ
			最終頁に続く

(54)【発明の名称】 SQLクエリを加速するための決定的分散キャッシュのための技術

(57)【要約】

改良される分散キャッシュを提供するための技術が開示される。分散コンピューティングシステムは、クエリに関連するデータを処理するための1つまたは複数の実行部をホストするように構成されている複数のワーカーノードを含むクラスターで実装することができる。ワーカーノードは、実行部にアクセスできるキャッシュをホストできる。データは複数のデータセグメントとして処理することができる。ワーカーノードには、整数トークン値の範囲を規定する複数のトークン境界を均一に割り当てることができる。ハッシュアルゴリズムを使用して、クエリに関連付けられる各データセグメントのトークンを計算できる。タスクは、データセグメントを処理するタスクには、優先実行部に関連付けられているトークン境界内のトークンが含まれるように、実行部上で優先的に起動することができる。実行部は、関連するキャッシュをレビューして外れ値データセグメントを識別し、クラスター内の他のノードに通知するように指示できる。



【特許請求の範囲】

【請求項 1】

コンピュータに実行される方法であって、

分析データ処理サービスを提供する分散コンピューティングシステムが、複数のノードを含むクラスタを実装するステップと、

前記分散コンピューティングシステムが、前記複数のノードに均一に関連付けられている複数のトークン境界を含むクラスタの状態を維持するステップと、

前記複数のノードのうちのドライバノードが、実行のためのクエリを受信するステップと、

前記クエリに少なくとも部分的に基づいて前記ドライバノードが、前記クエリに対応する 1 つまたは複数のデータセグメントのセットを識別するステップと、

前記ドライバノードが、1 つまたは複数のデータセグメントの前記セットに対応するトークンのセットを計算するステップと、

前記複数のノードのうちの第 1 のワーカーノード上で実行する第 1 の実行部上で、1 つまたは複数のデータセグメントの前記セットから第 1 のデータセグメントを処理する第 1 のタスクを起動するステップと、を含み、選択される前記第 1 のワーカーノードは、前記複数のトークン境界のうちのトークン境界の第 1 の対に対応するトークンの前記セットの第 1 のトークンに少なくとも部分的に基づき、トークン境界の前記第 1 の対は前記第 1 のワーカーノードに関連付けられ、

前記方法は、

前記第 1 のワーカーノードが、前記第 1 のデータセグメントを取得するステップをさらに含む、方法。

【請求項 2】

前記クラスタの前記状態を維持するステップは、

1 つまたは複数の実行部の前記複数のノードへのマッピングを記憶するステップと、

前記マッピングに少なくとも部分的に基づいて、前記複数のノード上で実行する前記 1 つまたは複数の実行部に前記複数のトークン境界を割り当てるステップと、を含み、前記複数のトークン境界は、前記 1 つまたは複数の実行部に均一に分配され、前記複数のトークン境界の前記第 1 の対のトークン境界は、前記 1 つまたは複数の実行部のうちの第 1 の実行部に割り当てられる、請求項 1 に記載のコンピュータに実行される方法。

【請求項 3】

前記ドライバノードが、前記クラスタが変更されたという指示を受信するステップと、

前記指示に応答して、前記複数のトークン境界を更新するステップと、

前記更新されている複数のトークン境界を前記 1 つまたは複数の実行部に割り当てるステップと、をさらに含み、前記更新されている複数のトークン境界は前記 1 つまたは複数の実行部に均一に分配される、請求項 2 に記載のコンピュータに実行される方法。

【請求項 4】

前記第 1 のデータセグメントを取得するステップは、前記第 1 の実行部が、前記第 1 のワーカーノードに関連付けられている第 1 のキャッシュに前記第 1 のデータセグメントが存在することを決定するステップと、前記第 1 のキャッシュから前記第 1 のデータセグメントを読み取るステップとを含む、先行する請求項のいずれかに記載のコンピュータに実行される方法。

【請求項 5】

前記第 1 のデータセグメントを取得するステップは、

前記第 1 の実行部が、前記第 1 のデータセグメントが前記第 1 のワーカーノードに関連付けられている第 1 のキャッシュに存在しないと決定するステップと、

前記複数のノードのうちの 1 つまたは複数の隣接ノードに要求を送信するステップと、

前記要求に応答して、前記 1 つまたは複数の隣接ノードの第 2 のワーカーノード上で実行する第 2 の実行部が、前記第 1 のデータセグメントが前記第 2 のワーカーノードに関連付けられている第 2 のキャッシュに存在することを決定するステップと、

前記第 1 のワーカーノードが、前記第 1 のデータセグメントを前記第 2 のキャッシュから前記第 1 のキャッシュにコピーするステップと、

前記第 1 のキャッシュから前記第 1 のデータセグメントを読み取るステップと、を含む、請求項 1 ～請求項 3 のいずれかに記載のコンピュータに実行される方法。

【請求項 6】

トークンの前記セットを計算するステップは、前記 1 つまたは複数のデータセグメントを一意に識別するハッシュ値を計算するステップを含む、先行する請求項のいずれかに記載のコンピュータに実行される方法。

【請求項 7】

前記複数のトークン境界は、整数トークン値の範囲を分割する複数の整数値を含み、前記計算されるハッシュ値は、整数トークン値の前記範囲内にある、請求項 6 に記載のコンピュータに実行される方法。

10

【請求項 8】

前記ドライバノードが、ハウスキーピング要求を前記第 1 のワーカーノードに送信するステップと、

前記要求に応答して、前記第 1 の実行部が、前記第 1 のワーカーノードに関連付けられている第 1 のキャッシュに存在する 1 つまたは複数の外れ値データセグメントを決定するステップを含み、前記 1 つまたは複数の外れ値データセグメントは、前記第 1 の実行部に割り当てられるトークン境界の前記第 1 の対の外側にある前記 1 つまたは複数の外れ値データセグメントに対応するトークンの前記セットの 1 つまたは複数のトークンに少なくとも部分的に基づいて決定され、

20

前記第 1 の実行部が、前記 1 つまたは複数の外れ値データセグメントの識別子を前記複数のノードに送信するステップと、

前記複数のノードのうちの第 2 のワーカーノードが、前記識別子に少なくとも部分的に基づいて、前記 1 つまたは複数の外れ値データセグメントを前記第 2 のワーカーノードに関連付けられている第 2 のキャッシュにコピーするステップと、をさらに含む、先行する請求項のいずれかに記載のコンピュータに実行される方法。

【請求項 9】

前記ドライバノードが、ハウスキーピング要求を前記第 1 のワーカーノードに送信するステップと、

30

前記要求に応答して、前記第 1 の実行部によって有効なデータセグメントのセットを決定するステップと、

有効なデータセグメントの前記セットに少なくとも部分的に基づいて、前記第 1 のワーカーノードに関連付けられている第 1 のキャッシュ内に存在する 1 つまたは複数の無効なデータセグメントを決定するステップと、

前記 1 つまたは複数の無効なデータセグメントを前記第 1 のキャッシュから削除するステップと、

前記第 1 のワーカーノードが、前記第 1 のワーカーノードに関連付けられている現在の記憶可用性を決定するステップと、

40

前記現在の記憶可用性が閾値未満である場合：

前記第 1 の実行部が、前記第 1 のキャッシュ内に存在する 1 つまたは複数の標的データセグメントを決定するステップと、

前記標的データセグメントを削除するステップと、をさらに含む、先行する請求項のいずれかに記載のコンピュータに実行される方法。

【請求項 10】

前記 1 つまたは複数の標的データセグメントは、セグメント温度のセットに少なくとも部分的に基づいて決定される、請求項 9 に記載のコンピュータに実行される方法。

【請求項 11】

分析データ処理サービスを提供する分散コンピューティングシステムであって、

1 つまたは複数のプロセッサと、

50

前記 1 つまたは複数のプロセッサで実行されると、前記分散コンピューティングシステムに、

複数のノードを含むクラスタを実行し、

前記クラスタの状態を維持し、前記状態は、前記複数のノードに均一に関連付けられている複数のトークン境界を含み、

前記複数のノードのうちのドライバノードが、実行のためのクエリを受信し、

前記クエリに少なくとも部分的に基づいて、前記クエリに対応する 1 つまたは複数のデータセグメントのセットを識別し、

前記ドライバノードが、1 つまたは複数のデータセグメントの前記セットに対応するトークンのセットを計算し、

前記複数のノードのうちの第 1 のワーカーノード上で実行する第 1 の実行部上で、1 つまたは複数のデータセグメントの前記セットから第 1 のデータセグメントを処理する第 1 のタスクを起動し、前記第 1 のワーカーノードは、前記複数のトークン境界のうちのトークン境界の第 1 の対に対応するトークンの前記セットの第 1 のトークンに少なくとも部分的に基づいて選択され、トークン境界の前記第 1 の対は前記第 1 のワーカーノードに関連付けられ、

前記第 1 のワーカーノードが、前記第 1 のデータセグメントを取得する、ことを実行させるコンピュータ実行可能命令を記憶する 1 つまたは複数のメモリと、を含む分散コンピューティングシステム。

【請求項 1 2】

前記クラスタの前記状態を維持することは、

1 つまたは複数の実行部の前記複数のノードへのマッピングを記憶することと、

前記マッピングに少なくとも部分的に基づいて、前記複数のノード上で実行する前記 1 つまたは複数の実行部に前記複数のトークン境界を割り当てることと、を含み、前記複数のトークン境界は、前記 1 つまたは複数の実行部に均一に分配され、前記複数のトークン境界のトークン境界の前記第 1 の対は、前記 1 つまたは複数の実行部の第 1 の実行部に割り当てられる、請求項 1 1 に記載の分散コンピューティングシステム。

【請求項 1 3】

前記コンピュータ実行可能命令を実行すると、前記分散コンピューティングシステムは、

前記ドライバノードが、前記クラスタが変更されたという指示を受信し、

前記指示に応答して、前記複数のトークン境界を更新し、および

前記更新されている複数のトークン境界を前記 1 つまたは複数の実行部に割り当て、前記更新されている複数のトークン境界は前記 1 つまたは複数の実行部に均一に分配される、ことをさらに実行する、請求項 1 2 に記載の分散コンピューティングシステム。

【請求項 1 4】

前記第 1 のデータセグメントを取得することは、前記第 1 の実行部が、前記第 1 のワーカーノードに関連付けられている第 1 のキャッシュに前記第 1 のデータセグメントが存在することを決定することと、前記第 1 のキャッシュから前記第 1 のデータセグメントを読み取ることとを含む、請求項 1 1 ～請求項 1 3 のいずれかに記載の分散コンピューティングシステム。

【請求項 1 5】

前記第 1 のデータセグメントを取得することは、

前記第 1 の実行部が、前記第 1 のデータセグメントが前記第 1 のワーカーノードに関連付けられている第 1 のキャッシュに存在しないと決定することと、

前記複数のノードのうちの 1 つまたは複数の隣接ノードに要求を送信することと、

前記要求に応答して、前記 1 つまたは複数の隣接ノードの第 2 のワーカーノード上で実行する第 2 の実行部が、前記第 1 のデータセグメントが前記第 2 のワーカーノードに関連付けられる第 2 のキャッシュに存在することを決定することと、

前記第 1 のワーカーノードが、前記第 1 のデータセグメントを前記第 2 のキャッシュか

10

20

30

40

50

ら前記第 1 のキャッシュにコピーすることと、

前記第 1 のキャッシュから前記第 1 のデータセグメントを読み取ることと、を含む、請求項 1 1 ～請求項 1 3 のいずれかに記載の分散コンピューティングシステム。

【請求項 1 6】

1 つまたは複数のプロセッサによって実行されると、分散コンピュータシステムに少なくとも、

複数のノードを含むクラスタを実行し、

前記クラスタの状態を維持し、前記状態は、前記複数のノードに均一に関連付けられている複数のトークン境界を含み、

前記複数のノードのうちのドライバノードが、実行のためのクエリを受信し、

前記クエリに少なくとも部分的に基づいて、前記クエリに対応する 1 つまたは複数のデータセグメントのセットを識別し、

前記ドライバノードが、1 つまたは複数のデータセグメントの前記セットに対応するトークンのセットを計算し、

前記複数のノードのうちの第 1 のワーカーノード上で実行する第 1 の実行部上で、1 つまたは複数のデータセグメントの前記セットから第 1 のデータセグメントを処理する第 1 のタスクを起動し、前記第 1 のワーカーノードは、前記複数のトークン境界のうちのトークン境界の第 1 の対に対応するトークンの前記セットの第 1 のトークンに少なくとも部分的に基づいて選択され、トークン境界の前記第 1 の対は前記第 1 のワーカーノードに関連付けられ、

前記第 1 のワーカーノードが、前記第 1 のデータセグメントを取得する、ことを実行させるコンピュータ実行可能命令を記憶する非一時的なコンピュータ可読媒体。

【請求項 1 7】

前記クラスタの前記状態を維持することは、

1 つまたは複数の実行部の前記複数のノードへのマッピングを記憶することと、

前記マッピングに少なくとも部分的に基づいて、前記複数のノード上で実行する前記 1 つまたは複数の実行部に前記複数のトークン境界を割り当てることと、を含み、前記複数のトークン境界は、前記 1 つまたは複数の実行部に均一に分配され、前記複数のトークン境界のうちのトークン境界の前記第 1 の対は、前記 1 つまたは複数の実行部のうちの第 1 の実行部に割り当てられる、請求項 1 6 に記載の非一時的なコンピュータ可読媒体。

【請求項 1 8】

前記コンピュータ実行可能命令を実行すると、前記分散コンピューティングシステムは、

前記ドライバノードが、前記クラスタが変更されたという指示を受信し、

前記指示に応答して、前記複数のトークン境界を更新し、および

前記更新されている複数のトークン境界を前記 1 つまたは複数の実行部に割り当て、前記更新されている複数のトークン境界は前記 1 つまたは複数の実行部に均一に分配される、ことをさらに実行する、請求項 1 7 に記載の非一時的なコンピュータ可読媒体。

【請求項 1 9】

前記第 1 のデータセグメントを取得することは、前記第 1 の実行部が、前記第 1 のワーカーノードに関連付けられている第 1 のキャッシュに前記第 1 のデータセグメントが存在することを決定することと、前記第 1 のキャッシュから前記第 1 のデータセグメントを読み取ることとを含む、請求項 1 6 ～請求項 1 8 のいずれかに記載の非一時的なコンピュータ可読媒体。

【請求項 2 0】

前記第 1 のデータセグメントを取得することは、

前記第 1 の実行部が、前記第 1 のデータセグメントが前記第 1 のワーカーノードに関連付けられている第 1 のキャッシュに存在しないと決定することと、

前記複数のノードのうちの 1 つまたは複数の隣接ノードに要求を送信することと、

前記要求に応答して、前記 1 つまたは複数の隣接ノードの第 2 のワーカーノード上で実

10

20

30

40

50

行する第2の実行部が、前記第1のデータセグメントが前記第2のワーカーノードに関連付けられている第2のキャッシュに存在することを決定することと、

前記第1のワーカーノードが、前記第1のデータセグメントを前記第2のキャッシュから前記第1のキャッシュにコピーすることと、

前記第1のキャッシュから前記第1のデータセグメントを読み取ることと、を含む、請求項16～請求項18のいずれかに記載の非一時的なコンピュータ可読媒体。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

本出願は、2021年5月21日に提出され、「SQLクエリを加速するための決定的分散キャッシュのための技術」と題されているインド特許仮出願第202141022725号の非仮出願であり、優先権と利益を主張し、その全内容は、あらゆる目的のために参照により本明細書に組み込まれる。

【0002】

技術分野

本開示は、クラウドコンピューティングデータ分析に関する。より詳細には、本開示は、コンピューティングクラスタ内でデータの分散キャッシュを維持し、キャッシュされるデータの位置に対応するクラスタ内でコンピューティングタスクを実行することを対象とする。

【背景技術】

【0003】

背景

クラウドベースのサービスは、任意の数のテナントによって大量のデータを処理するためのソリューションを提供する。クラウドベースのデータ処理サービスは、ユーザ要求を扱うための動作を実行するノードを含む適切な数のコンピューティングクラスタを含む分散コンピューティングシステムによって実装され得る。高性能データ分析サービスは、ユーザからの対話型のほぼリアルタイムのクエリに基づいてクラスタ全体に分散データを処理できるが、グローバル記憶装置からのデータセグメントの取得が遅いことに依存している。対話型クエリのデータ処理の速度と効率を向上させるために、クラスタノードでのデータセグメントのキャッシュを改善する必要がある。

【発明の概要】

【0004】

簡単な概要

本開示の実施形態は、クラウドコンピューティング環境においてキャッシュを提供することに関する。より詳細には、いくつかの実施形態は、分散コンピューティング環境内のノード間で一貫性のある分散キャッシュを実装する方法およびシステムを提供する。キャッシングは、分散コンピューティング環境内の複数のノードの構成を可能にし、各々が関連するキャッシュを有する一貫したハッシュ技術を部分的に使用して提供することができる。ノード上で実行される1つまたは複数のタスクは、関連付けられているキャッシュを共有できるが、キャッシュに記憶されている一意のデータセグメントの所有権を持つ。所有権は、分散コンピューティングシステムによって処理されるクエリに関連付けられている各データセグメントのトークンを計算する一貫したハッシュ技術に基づくことができる。各データセグメントのタスク、ノード、およびトークンの間の対応関係により、クエリオプティマイザによって決定されるように、特定のノードをタスクの実行のための優先ノードとして識別できるようになり得る。

【課題を解決するための手段】

【0005】

一実施形態は、分析データ処理サービスを提供する分散コンピューティングシステムによって実行される方法を対象とする。この方法は、複数のノードを含むクラスタを実装す

10

20

30

40

50

るステップを含むことができる。分散コンピューティングシステムは、複数のノードに均一に関連付けられている複数のトークン境界を含むクラスタ状態を維持することができる。クラスタ状態を維持することは、1つまたは複数の実行部の複数のノードへのマッピングを記憶するステップと、マッピングに基づいて、複数のノード上で実行される1つまたは複数の実行部に複数のトークン境界を割り当てるステップとを含むことができる。この方法はまた、複数のノードのうちのドライバノードが、実行のためのクエリを受信するステップを含む。クエリに基づいて、クエリに対応する1つまたは複数のデータセグメントのセットを識別または決定することができる。1つまたは複数のデータセグメントのセットに対応するトークンのセットは、ハッシュアルゴリズムを使用して計算され、1つまたは複数のデータセグメントを一意に識別するハッシュ値を決定することができる。この方法はまた、複数のノードのうちの第1のワーカーノード上で実行する第1の実行部上で、1つまたは複数のデータセグメントのセットから第1のデータセグメントを処理する第1のタスクを起動するステップを含み、第1のワーカーノードは、第1のワーカーノードに関連付けられているトークン境界の第1の対に対応するトークンのセットの第1のトークンに少なくとも部分的に基づいて選択される。最後に、第1のワーカーノードは、第1のタスクによって処理されるデータセグメントを取得できる。データセグメントは、第1のワーカーノードに関連付けられているキャッシュ内に存在すること、またはノードから離れたデータストア、データベース、オブジェクト記憶装置、もしくは他のリポジトリ内に存在することができる。

10

20

【0006】

特定の実施形態によれば、分散コンピューティングシステムは、クラスタが変化したという指示を受信することができる。例えば、ハードウェアまたはソフトウェアの障害によりノードが動作不能になり得る。この方法は、複数のトークン境界を更新すること、および更新されているトークン境界を分散コンピューティングシステム内で実行している1つまたは複数の実行部に均一に割り当てることによってクラスタ状態を維持するステップを含むことができる。

【0007】

他のいくつかの実施形態によれば、この方法は、第2のワーカーノードに関連付けられている第2のキャッシュに存在する第1のデータセグメントを取得するステップを含むことができる。これらの実施形態では、第1の実行部は、第1のデータセグメントが第1のワーカーノードに関連付けられている第1のキャッシュに存在しないと決定することができる。第1の実行部は、複数のノードのうちの1つまたは複数の隣接ノードに要求を送信することができる。隣接ノード上で実行する実行部は、要求に基づいて、各隣接ノードに関連付けられているキャッシュを検査できる。1つまたは複数の隣接ノードの第2のワーカーノード上で実行する第2の実行部は、第1のデータセグメントが第2のキャッシュに存在することを決定することができる。分散コンピューティングシステム内のノード間のデータ転送を扱うように構成されているブロックマネージャまたはその他のモジュールにそのデータセグメントを配置し、次いで、第1のデータセグメントのIDを、要求を行った第1の実行部に送信することができる。次に、第1のワーカーノードは、第2のワーカーノードから第1のワーカーノードに関連付けられている第1のキャッシュにデータセグメントをコピーできる。

30

40

【0008】

いくつかの実施形態では、方法は、ハッシュアルゴリズムを使用して複数のトークン境界を計算するステップを含むことができる。計算の出力は、受信したクエリに関連付けられている各データセグメントからの識別情報、例えばデータセグメントのファイル名からハッシュとして計算される整数値にすることができる。各データセグメントのハッシュ値は、そのデータセグメントに対応するトークンにすることができる。複数のトークン境界は、可能な整数トークン値の範囲を分割する複数の整数値を含むことができる。したがって、トークン境界の各対は、トークン境界の対の値の間にある連続する整数値の範囲を規定する。

50

【0009】

他のいくつかの実施形態では、この方法は、分散コンピューティングシステム内の分散キャッシュを安定化するための動作を含むことができる。この方法は、ハウスキーピング要求を第1のワーカーノードに送信し、第1のワーカーノードに関連付けられている第1のキャッシュに存在する1つまたは複数の外れ値データセグメントを決定するように第1の実行部を促すステップを含んでもよい。キャッシュ内の外れ値データセグメントは、セグメントに関連付けられているトークンが第1の実行部に割り当てられるトークン境界内かどうかによって部分的に決定される。その後、第1の実行部は外れ値データセグメントの識別子を他のすべてのノードに送信できる。次に、第2のワーカーノードは、1つまたは複数の外れ値データセグメントを第1のワーカーノードの第1のキャッシュから第2のワーカーノードの第2のキャッシュにコピーできる。

10

【0010】

さらに他の実施形態では、この方法は、スペースおよび他のリソースを記憶するために分散キャッシュからデータセグメントを削除する動作を含むことができる。この方法は、ハウスキーピング要求を第1のワーカーノードに送信し、第1のキャッシュ内に存在する有効なデータセグメントのセットおよび1つまたは複数の無効なデータセグメントを決定するように第1の実行部に促すステップを含むことができる。1つまたは複数の無効なデータセグメントは、第1のキャッシュから削除され得る。この方法はまた、第1のワーカーノードに関連付けられている現在の記憶可用性を決定するステップを含むこともできる。現在の記憶可用性が閾値未満である場合、第1の実行部は、セグメント温度のセットに基づいて、第1のキャッシュ内に存在する1つまたは複数の標的データセグメントを決定することができる。その後、1つまたは複数の標的データセグメントをキャッシュから削除できる。

20

【0011】

別の実施形態は、1つまたは複数のプロセッサと、コンピュータ実行可能命令を記憶する1つまたは複数のメモリとを含む分散コンピューティングシステムを対象とし、1つまたは複数のプロセッサで実行されると、分散コンピューティングシステムに複数のノードを含むクラスタを実行させる。分散コンピューティングシステムは、複数のノードに均一に関連付けられている複数のトークン境界を含むクラスタ状態を維持することができる。クラスタ状態を維持することは、1つまたは複数の実行部の複数のノードへのマッピングを記憶するステップと、マッピングに基づいて、複数のノード上で実行される1つまたは複数の実行部に複数のトークン境界を割り当てるステップとを含むことができる。この命令はまた、分散コンピューティングシステムに、複数のノードのうちのドライバノードが、実行のためのクエリを受信させることもできる。クエリに基づいて、クエリに対応する1つまたは複数のデータセグメントのセットを識別または決定することができる。1つまたは複数のデータセグメントのセットに対応するトークンのセットは、ハッシュアルゴリズムを使用して計算され、1つまたは複数のデータセグメントを一意に識別するハッシュ値を決定することができる。また、命令は、分散コンピューティングシステムに、複数のノードのうちの第1のワーカーノード上で実行する第1の実行部上で、1つまたは複数のデータセグメントのセットから第1のデータセグメントを処理する第1のタスクを起動させることもでき、第1のワーカーノードは、第1のワーカーノードに関連付けられているトークン境界の第1の対に対応するトークンのセットの第1のトークンに少なくとも部分的に基づいて選択される。最後に、第1のワーカーノードは、第1のタスクによって処理されるデータセグメントを取得できる。データセグメントは、第1のワーカーノードに関連付けられているキャッシュ内に存在すること、またはノードから離れたデータストア、データベース、オブジェクト記憶装置、もしくは他のリポジトリ内に存在することができる。

30

40

【0012】

特定の実施形態によれば、分散コンピューティングシステムは、クラスタが変化したという指示を受信することができる。例えば、ハードウェアまたはソフトウェアの障害によ

50

りノードが動作不能になり得る。また、命令は、複数のトークン境界を更新し、更新されているトークン境界を分散コンピューティングシステム内で実行中の1つまたは複数の実行部に均一に割り当てることによって、分散コンピューティングシステムにクラスタ状態を維持させ得る。

【0013】

他のいくつかの実施形態によれば、命令は、分散コンピューティングシステムに、第2のワーカーノードに関連付けられている第2のキャッシュに存在する第1のデータセグメントを取得させることもできる。これらの実施形態では、第1の実行部は、第1のデータセグメントが第1のワーカーノードに関連付けられている第1のキャッシュに存在しないと決定することができる。第1の実行部は、複数のノードのうちの1つまたは複数の隣接ノードに要求を送信することができる。隣接ノード上で実行する実行部は、要求に基づいて、各隣接ノードに関連付けられているキャッシュを検査できる。1つまたは複数の隣接ノードの第2のワーカーノード上で実行する第2の実行部は、第1のデータセグメントが第2のキャッシュに存在することを決定することができ、分散コンピューティングシステム内のノード間のデータ転送を扱うように構成されているブロックマネージャまたはその他のモジュールにそのデータセグメントを配置し、次いで、第1のデータセグメントのIDを、要求を行った第1の実行部に送信することができる。次に、第1のワーカーノードは、第2のワーカーノードから第1のワーカーノードに関連付けられている第1のキャッシュにデータセグメントをコピーできる。

【0014】

別の実施形態は、コンピュータ実行可能命令を記憶する非一時的なコンピュータ可読媒体に向けられており、当該命令は、1つまたは複数のプロセッサによって実行されると、コンピュータシステムに、複数のノードを含むクラスタを実行させ、複数のノードに均一に関連付けられている複数のトークン境界を含むクラスタ状態を維持させ、複数のノードのうちのドライバノードによって実行のためのクエリを受信させ、クエリに基づいて、当該クエリに対応する1つまたは複数のデータセグメントのセットを決定させ、ハッシュアルゴリズムを使用して、1つまたは複数のデータセグメントのセットに対応するトークンのセットを計算して、1つまたは複数のデータセグメントを一意に識別するハッシュ値を決定させ、複数のノードのうちの第1のワーカーノード上で実行する第1の実行部上で、1つまたは複数のデータセグメントのセットから第1のデータセグメントを処理する第1のタスクを起動させ、第1のワーカーノードは、第1のワーカーノードに関連付けられているトークン境界の第1の対に対応するトークンのセットの第1のトークンに少なくとも部分的に基づいて選択され、当該命令は当該コンピュータシステムにさらに、第1のタスクによって処理されるデータセグメントを取得させる。データセグメントは、第1のワーカーノードに関連付けられているキャッシュ内に存在すること、またはノードから離れたデータストア、データベース、オブジェクト記憶装置、もしくは他のリポジトリ内に存在することができる。クラスタ状態を維持することは、1つまたは複数の実行部の複数のノードへのマッピングを記憶するステップと、マッピングに基づいて、複数のノード上で実行される1つまたは複数の実行部に複数のトークン境界を割り当てるステップとを含むことができる。

【0015】

特定の実施形態によれば、分散コンピューティングシステムは、クラスタが変化したという指示を受信することができる。例えば、ハードウェアまたはソフトウェアの障害によりノードが動作不能になり得る。この命令はまた、コンピュータシステムに、複数のトークン境界を更新し、更新されているトークン境界を分散コンピューティングシステム内で実行中の1つまたは複数の実行部に均一に割り当てることによって、クラスタ状態を維持させ得る。

【0016】

他のいくつかの実施形態によれば、命令は、コンピュータシステムに、第2のワーカーノードに関連付けられている第2のキャッシュに存在する第1のデータセグメントを取得

させることもできる。これらの実施形態では、第1の実行部は、第1のデータセグメントが第1のワーカーノードに関連付けられている第1のキャッシュに存在しないと決定することができる。第1の実行部は、複数のノードのうちの1つまたは複数の隣接ノードに要求を送信することができる。隣接ノード上で実行する実行部は、要求に基づいて、各隣接ノードに関連付けられているキャッシュを検査できる。1つまたは複数の隣接ノードの第2のワーカーノード上で実行する第2の実行部は、第1のデータセグメントが第2のキャッシュに存在することを決定することができ、分散コンピューティングシステム内のノード間のデータ転送を扱うように構成されているブロックマネージャまたはその他のモジュールにそのデータセグメントを配置し、次いで、第1のデータセグメントのIDを、要求を行った第1の実行部に送信することができる。次に、第1のワーカーノードは、第2のワーカーノードから第1のワーカーノードに関連付けられている第1のキャッシュにデータセグメントをコピーできる。

10

【図面の簡単な説明】

【0017】

【図1】いくつかの実施形態による、オブジェクト記憶システムから取得されるデータセグメントの分散キャッシュを含む、クラウドコンピューティング環境における分散コンピューティングシステムを示す図である。

【図2】いくつかの実施形態による、決定的キャッシュ技術を実装する分散コンピューティングシステム内のコンピューティングノードのクラスタを示す図である。

【図3A】いくつかの実施形態による、クラスタ内のノードへのトークン境界のマッピング例を示すコードの断片の図である。

20

【図3B】いくつかの実施形態による、クラスタ内の更新されるノードのセットへのトークン境界のマッピングの別の例を示すコードの別の断片の図である。

【図4】いくつかの実施形態による、決定的キャッシュを用いてクラスタ上でクエリを処理するための例示的なプロセスの簡略化されているフローチャートである。

【図5】いくつかの実施形態による、クラスタ状態が変化したときにクラスタ内の決定的キャッシュを安定化するための例示的なプロセスの別の簡略化されているフローチャートである。

【図6】タイマプロセスに従って決定的キャッシュをクリーニングするためのプロセス例のさらに別の簡略化されているフローチャートである。

30

【図7】少なくとも1つの実施形態による、サービスシステムとしてのクラウドインフラストラクチャを実装するための1つのパターンを示すブロック図である。

【図8】少なくとも1つの実施形態による、サービスシステムとしてのクラウドインフラストラクチャを実装するための別のパターンを示すブロック図である。

【図9】少なくとも1つの実施形態による、サービスシステムとしてのクラウドインフラストラクチャを実装するための別のパターンを示すブロック図である。

【図10】少なくとも1つの実施形態による、サービスシステムとしてのクラウドインフラストラクチャを実装するための別のパターンを示すブロック図である。

【図11】少なくとも1つの実施形態による例示的なコンピュータシステムを示すブロック図である。

40

【発明を実施するための形態】

【0018】

詳細な説明

以下の説明では、説明の目的で、特定の実施形態の完全な理解を提供するために特定の詳細が記載される。しかし、これらの特定の詳細がなくても、さまざまな実施形態を実施できることは明らかであろう。図および説明は、限定することを意図したものではない。

「例示的な」という言葉は、本明細書では「実施例、例、または実例として機能する」という意味で使用される。本明細書で「例示的」として説明される任意の実施形態または設計は、必ずしも他の実施形態または設計よりも好ましいまたは有利であると解釈されるべきではない。

50

【0019】

分散コンピューティングシステムは、データ分析においてますます一般的になってきており、大量のデータを処理するための高速で信頼性が高く、スケーラブルなソリューションを提供する機能を備えている。クラウドコンピューティング環境で分散コンピューティングシステムを提供すると、これらのデータ処理機能がクラウドコンピューティング環境の複数の異なる顧客（例えば、テナント）に提供される。分散コンピューティング空間における最近の開発（堅牢なクエリ最適化など）により、大規模なデータセットに対する対話型クエリをほぼリアルタイムで処理できるようになり、要求側ユーザはクエリを起動し、ユーザインターフェイスで結果を対話的に期待できる。独立したスケーラビリティを確保するために、クラウドコンピューティング環境ではデータ記憶とコンピューティングリソースが分離されている。この分離により、処理のためにデータが記憶装置から取得されるときに最適な性能が得られない可能性がある。本明細書に開示される技術は、分散コンピューティングシステム内に決定的分散キャッシュを提供して分析データ処理の性能を向上させるための方法、システム、およびコンピュータ可読記憶媒体を対象とする。

10

【0020】

分散コンピューティングシステムは、任意の適切な数のテナントによってさまざまな要求（例えば、データベース、クエリなどを維持するシステム内のデータの記憶および取得）を扱うために調整される方法で協働する、接続されるノード（例えば、コンピュータ、サーバ、仮想マシンなど）のコンピューティングクラスタを含むことができる。本明細書で使用される場合、「コンピューティングノード」（「ワーカーノード」および／または単に「ノード」とも呼ばれる）は、サーバ、コンピューティング装置、仮想マシン、またはコンピューティングクラスタの一部として動作を実行するように構成されている任意の適切な物理または仮想コンピューティングリソースが含まれ得る。例えば、コンピューティングクラスタは、1つまたは複数のマスターノード（本明細書では「ドライバノード」とも呼ばれる）および1つまたは複数のワーカーノードを含むことができる。いくつかの実施形態では、ドライバノードは、1つまたは複数のワーカーノードに対応するタスク割り当て、負荷分散、ノードプロビジョニング、ノード削除に関連する任意の適切な動作、またはコンピューティングクラスタの管理に対応する任意の適切な動作を実行することができる。ワーカーノードは、1つまたは複数のドライバノードによってワーカーノードに割り当てられているタスクに対応する動作を実行するように構成され得る。例えば、ワーカーノードは、ドライバノードによってワーカーノードに割り当てられているタスクの一部として、データベースに関連付けられているデータ記憶タスクおよび／またはデータ取得タスクを実行できる。

20

30

【0021】

分析データ処理サービス（例えば、オンライン分析処理（OLAP）サービス）を提供する分散コンピューティングシステム内では、データは、オブジェクト記憶装置データストア、データベース、または同様の「ディープ」記憶装置もしくは「オフライン」記憶装置に記憶され得る。通常、オブジェクト記憶システムは、データへの低頻度のアクセスをサポートし、さまざまな不揮発性メモリを備えているため、オブジェクト記憶装置へのデータの読み取りおよび／または書き込み動作に遅延が生じ得る。データは、データを多次元的に表す、より大きなデータ構造のセグメントとして記憶できる。データセグメントは、辞書を含む圧縮ファイルであってもよい。データセグメントは、データ処理サービスによってアトミックに取り扱われおよび処理されるデータの最小単位を表し得る。クエリを実行するには、データセグメントをオブジェクト記憶装置から取得し、ローカルファイルシステムまたは分散コンピューティングシステムの1つもしくは複数のノードに関連付けられている他のメモリに記憶し得る。データセグメントのローカルコピーは、分散コンピューティングシステム内の分散キャッシュを構成することができる。システム内の各ノードには、異なるデータセグメントのセットを記憶するキャッシュがあり得るため、本開示の技術は、とりわけ、データセグメントの一貫したキャッシュ、クラスタへの変更に応じたキャッシュの安定化、およびレプリケーションの適切な取り扱いのための技術を提供す

40

50

る。

【0022】

いくつかの実施形態では、ドライバモジュール（例えば、Apache Spark（登録商標）Thriftyサーバ）は、分散コンピューティングシステム内のクラスタのドライバノードまたはマスターノード上で動作し得る。ドライバモジュールは、ユーザクエリ（例えば、SQLクエリ）をタスクと呼ばれるより小さな実行単位に変換することができる。タスクは、クラスタ内の1つまたは複数のワーカーノードで実行される1つまたは複数の実行プロセス（「実行部」とも呼ばれる）によって実行されるように構成できる。各タスクは、実行部による1つのデータセグメントの処理に対応できる。ドライバモジュールは、とりわけ、タスクが対応するデータセグメントを「所有」できる（例えば、キャッシュされるコピーを記憶する関連キャッシュを有する）実行部の優先ワーカーノードでタスクを起動することによって、クエリの実行を最適化することができる。言い換えると、クラスタ内での以前のクエリ実行（対話型セッションでのユーザからの以前のクエリなど）により、特定のデータセグメントが特定のワーカーノードでキャッシュされ得る。優先ワーカーノードでタスクを起動すると、関連するキャッシュからデータセグメントが取得され得る。

10

【0023】

いくつかの実施形態では、ドライバノードは（ドライバモジュールを介して）クラスタ内の1つまたは複数のワーカーノードにトークン境界のセットを割り当てることができる。トークン境界は、セットのトークン分割を表すセットの整数値の分割を表すことができる。例えば、整数値は $0 \sim 2^{31} - 1$ の整数（つまり、32ビット符号付き整数のサイズ）にすることができる。トークン境界は、ワーカーノードの数やワーカーノード上で動作するように構成されている実行部の数など、クラスタの状態に少なくとも部分的に基づいて決定できる。例えば、各々2つの実行部をホストする3つのワーカーノードを持つクラスタの場合、トークン境界により整数値のセットをトークン分割の6つの連続するサブセットに分割できる。ドライバノードは、各ワーカーノードに構成されている実行部の数に従って、トークン境界をワーカーノードに割り当てることができる。クラスタの状態が変化した場合（例えば、ノードの追加または削除）、ドライバノードは更新されるトークン境界を決定し、クラスタ状態の変化に従って更新されているトークン境界をワーカーノードに割り当てることができる。いくつかの他の実施形態では、ドライバノードは、ワーカーノードが故障したという指示を受信し、そのワーカーノードに以前に割り当てられているトークン境界を、クラスタ内に残っている1つまたは複数のワーカーノードに割り当てることができる。ノードに障害が発生したときに、リング状の構造を使用してリング内の次のノードにトークン分割を割り当てることができる従来の一貫したハッシング技術とは異なり、本開示の技術は、トークン境界をワーカーノードに均一に分配し、その均一な分散をクラスタの存続期間全体にわたって維持しようとすることができる。

20

30

【0024】

クラスタのノード内でタスクをスケジュールするために、いくつかの実施形態では、ドライバノードは、クエリに応答して処理されるデータセグメントのトークンを計算することができる。トークンは、一意のセグメントキーまたはセグメント識別子（例えば、ファイル名など）に基づいて計算されるハッシュ値とすることができる。ハッシュ値は、割り当てられているトークン境界の範囲内の値を検索することによって、特定のワーカーノードにマッピングできる。対応するワーカーノードは、タスクの優先ノードであってもよく、その結果、タスクの起動を担当するスケジューラプロセスは、優先ノードにタスクを優先的に配置することができる。タスクを処理する実行部は、ワーカーノードに関連付けられているキャッシュからデータセグメントを取得するか、データセグメントがキャッシュに存在しない場合は、別のノードまたはオブジェクト記憶装置からデータセグメントを取得できる。データセグメントが優先ワーカーノードのキャッシュ内にあると、そのデータセグメントを処理する後続のタスクが優先ワーカーノードで起動され、データセグメントがキャッシュヒットする可能性が高くなり得る。

40

50

【0025】

いくつかの実施形態によれば、クラスタ内の1つまたは複数のワーカーノードに関連付けられているキャッシュは、さまざまなハウスキーピング技術を使用して一貫して維持され得る。ハウスキーピング方法には、1つまたは複数のキャッシュを更新するための能動的な技術と受動的な技術の両方が含まれ得る。クラスタの状態の任意の変化（例えば、ノードの追加または削除、ノードの障害、実行部の追加または削除など）に応じて、クラスタ内のドライバノードは、クラスタ内のすべての実行部に、実行部をホストしているワーカーノードに関連付けられているキャッシュをスキャンする命令を送信できる。ドライバノードはクラスタの状態が変化したときに更新されるトークン境界をワーカーノードに割り当てることができるため、キャッシュに記憶されるデータセグメントは別のワーカーノードに対応し得る。ハウスキーピング要求に応じて、実行部はこれらの外れ値データセグメントを識別し、ブロックマネージャまたはワーカーノード間でデータを転送するように構成されている他のプロセスに渡し、外れ値データセグメントの識別子をクラスタ内の他のすべての実行部に送信する。その後、実行部は外れ値データセグメントを取得してローカルキャッシュに記憶できる。

10

【0026】

他のいくつかの実施形態では、ハウスキーピング方法は、タイマに基づいて生じる動作を含むことができる。オブジェクト記憶装置内のデータセグメントへの変更がオブジェクト記憶装置にコミットされると、時間の経過とともに、キャッシュされるデータセグメントが無効になり得る。さらに、各ワーカーノードに割り当てられるメモリリソースは制限され得る。タイマに基づいて、ドライバノードはクラスタ内の実行部に命令を送信し、他の動作の中でも無効なデータセグメントをそのキャッシュから削除し、外れ値のデータセグメントを転送して削除し、キャッシュ内の空きスペースを維持するために、キャッシュからのアクセス頻度が低いデータセグメントを削除することができる。

20

【0027】

決定的分散キャッシュを維持することは、従来の技術に比べて多くの利点を提供する。上で簡単に説明したように、一般的な一貫したハッシュ技術では、ノード障害が発生した場合の負荷分散が不十分になり得る。障害が発生したノードにマッピングされるトークンは、単一ノードの現在の負荷を評価せずに、単一ノードに再マッピングされ得る。本明細書で説明する技術は、クラスタ状態の変化に応じて、クラスタ内のすべての実行部間でトークンの均一な分散を維持しようとする。このようにして、ノードが失われたときに単一のノードに過負荷がかからないようにすることで、クエリの実行速度が維持される。ノード優先は、優先ノードのキャッシュまたはオブジェクト記憶装置からデータセグメントをコピーする必要があるクラスタ内の他のノードではなく、優先ノード上で同じデータセグメントを使用して繰り返しタスクを起動するのに役立ち得るため、タスク実行用の優先ノードを作成するためにワーカーノードにトークン境界を割り当てると、クラスタ内のキャッシュされるデータセグメントの重複が制限され得る。さらに、クラスタ内の実行部は他の実行部と通信して、オブジェクト記憶装置からセグメントを取得する前に、隣接するノードがデータセグメントをキャッシュしているかどうかを確認することができるため、ディープ記憶装置からの取得数が減ることで、クエリ実行の速度と効率が向上する。

30

40

【0028】

図1は、いくつかの実施形態による、オブジェクト記憶システムから取得されるデータセグメントの分散キャッシュを含む、クラウドコンピューティング環境100内の分散コンピューティングシステム110を示す。分散コンピューティングシステム110は、分散コンピューティングシステムを実装するためにコンピュータ可読命令（例えば、コード、プログラム）を実行する1つまたは複数のコンピューティングシステムによって実装され得る。図1に示されるように、分散コンピューティングシステム110は、ロードバランサ112、ゲートウェイ114（例えば、マルチテナントゲートウェイ）、アプリケーションプログラミングインターフェイス（API）サーバ116、および1つまたは複数のコンピューティングクラスタ（複数可）122を含むさまざまなシステムを含む。図1

50

に示すシステムによって使用される、またはシステムによって生成されるデータまたは情報の部分は、オブジェクト記憶システム156に記憶され得る。図1に示されるシステムは、コンピューティングシステム、ハードウェア、またはそれらの組み合わせの1つまたは複数の処理装置（例えば、プロセッサ、コア）によって実行されるソフトウェア（例えば、コード、命令、プログラム）を使用して実装され得る。ソフトウェアは、非一時的な記憶媒体（例えば、メモリ装置）に記憶されてもよい。

【0029】

分散コンピューティングシステム110は、さまざまな異なる構成で実装することができる。図1に示される実施形態では、分散コンピューティングシステム110は、クラウドプロバイダネットワークの1つまたは複数のサーバ上に実装することができ、そのデータ処理およびデータ分析サービスをサブスクリプションベースでクラウドサービスの加入者に提供することができる。図1に示す分散コンピューティングシステム110を含むコンピューティング環境100は、単なる一例であり、請求される実施形態の範囲を不当に限定することを意図するものではない。当業者であれば、多くの可能な変形、代替、および修正を認識するであろう。例えば、一部の实装では、分散コンピューティングシステム110は、図1に示されるシステムよりも多いまたは少ないシステムを使用して実装することができ、2つ以上のシステムを組み合わせてもよいし、システムの異なる構成または配置を有してもよい。

【0030】

いくつかの実施形態では、コンピューティングクラスタ（例えば、クラスタ122）は、分散コンピューティングシステム110のテナントまたは顧客のために大量のデータを処理および分析するための分散コンピューティングエンジンを表すことができる。異なるコンピューティングクラスタを1つのテナントに関連付けることができる。例えば、図1に示される実施形態では、コンピューティングクラスタ（複数可）122は、分散コンピューティングシステムのテナント120に関連付けられている。1つまたは複数の異なるコンピューティングクラスタを追加のテナント（例えば、テナント118、119）に関連付けることができる。クラスタは、任意の適切な数のコンピューティングノードを利用して、調整される方法で動作を実行するように構成できる。前述したように、「コンピューティングノード」（本明細書では「ノード」とも呼ばれる）は、サーバ、コンピューティング装置、仮想マシン、またはコンピューティングクラスタの一部として動作を実行するように構成されている任意の適切な物理または仮想コンピューティングリソースが含まれ得る。例として、クラスタ122は、ドライバノード124および1つまたは複数のワーカーノード（複数可）128を含む複数のノードを含むことができ、両方ともコンピューティングノードの例である。いくつかの実施形態では、ドライバノード124は、負荷分散、ノードプロビジョニング、ノード削除、またはクラスタ122の管理に対応する任意の適切な動作など、ワーカーノードに対応するタスク割り当てに関連する任意の適切な動作を実行する。1つまたは複数のワーカーノード（複数可）128は、ドライバノード124によってワーカーノード（複数可）128に割り当てられているタスクに対応する動作を実行するように構成される。非限定的な例として、ワーカーノード（複数可）128は、ワーカーノード（複数可）128に特定の記憶または取得タスクを割り当てるドライバノード124の命令で、記憶システム／データベースに関連するデータ記憶および／またはデータ取得タスクを実行することができる。

【0031】

分散コンピューティングシステムのテナントに割り当てられるリソースは、階層的に配置されている複数のクラウドベースのリソースから選択される。例えば、図1に示すように、リソースは、クラウドインフラストラクチャサービス140のプールであり得る。クラウドインフラストラクチャサービス140は、サービスとしてのキーバリュースタイルデータベース（KaaS）142、サービスとしてのワークフロー（WaaS）144、Kubernetesエンジン（KE）146、IDおよびアクセス管理（IAM）サービス148、仮想クラウドネットワーク（VCN）150、ならびにデータカタログ152を含

10

20

30

40

50

むことができる。

【0032】

いくつかの実施形態によれば、コンピューティングクラスタ（例えば、クラスタ122）内のドライバノード（例えば、ドライバノード124）は、ドライバプログラム（ドライバモジュール、ドライバプロセス、または単にドライバとも呼ばれる。つまり、コンピューティングクラスタ上に構築されているアプリケーションを実行するプロセス）を実行するように構成され得、アプリケーションのコンテキストを作成するための動作を実行し得る。「アプリケーション」は、独立したプロセスとして実行され、ドライバノード124で実行されるドライバプログラム内のアプリケーションのコンテキストによって調整される、完全な実行可能なドライバプログラムを指し得る。アプリケーションのコンテキストは、クラスタ内のすべてのノードにシステムリソースを割り当てるクラスタマネージャ126に接続することができる。クラスタ122内の各ワーカーノード（例えば、ワーカーノード（複数可）128）は、1つまたは複数の実行部によって管理され得、これらは、ノードに割り当てられているタスクに対応する動作を実行するためにワーカーノード（複数可）128上で起動されるプロセス（実行エンジン）であり得る。アプリケーションコードはドライバプログラムから実行部に送信され、実行部は実行するコンテキストとさまざまなタスクを指定する。実行部は、データ共有または対話のためにドライバプログラムとやり取りする。実行部はさらに、ノード上のデータの計算および記憶とキャッシュの管理に関連する動作を実行することもできる。特定の実装では、コンピューティングクラスタは、分散コンピューティングエンジン（例えば、Apache Spark）を使用して実装でき、コンピューティングクラスタ内のクラスタマネージャ126は、Kubernetesなどのコンテナオーケストレーションプラットフォーム、またはMesosやYARNなどの別のクラスタ管理ソリューションを使用して実装することができる。

10

20

【0033】

いくつかの実施形態では、図1に示されるコンピューティングクラスタによって使用されるか生成されるデータまたは情報の部分は、分散コンピューティングシステム110の1つまたは複数の記憶システムに記憶されてもよい。図1に示される実施形態では、記憶システムにはオブジェクト記憶システム156が含まれる。オブジェクト記憶システム156は、分散コンピューティングシステム110の異なるテナントに関連付けられている異なるコンピューティングクラスタによって使用され、分析され、処理されるデータを記憶するためのディープ記憶またはオフライン記憶システムを表すことができる。一例として、オブジェクト記憶システム156は、オブジェクトと呼ばれる別個の単位としてデータを管理および操作することによってデータを記憶するためにオブジェクトベースの記憶装置を使用する記憶システムのタイプを表すことができる。オブジェクト記憶システム156は、データのオフライン記憶および／またはアーカイブに使用することができる（例えば、データへのアクセス頻度が低いバックアップまたは長期記憶に使用される）。

30

【0034】

特定の実施形態では、コンピューティングクラスタによって処理されるデータは、「データキューブ」としてオブジェクト記憶システムに表され、記憶され得る。「データキューブ」は、2次元、3次元、またはより高次元の表現など、何らかの関心のある尺度に沿ってデータを表現するために使用できるデータ構造を指し得る。データキューブは大量のデータを記憶できると同時に、ユーザに任意のデータポイントへの検索可能なアクセスを提供し、クエリを実行してリアルタイムの結果を提供できる。特定の例では、実行時に、コンピューティングクラスタは、データキューブインデックスの一部または全体を、コンピューティングクラスタを形成するコンピューティングノードのキャッシュメモリ（例えば、キャッシュ130）内の1つまたは複数のデータセグメントとしてキャッシュし得、あるいは、コンピューティングノードに関連付けられている接続される2次記憶装置（例えば、ランダムアクセスメモリ（RAM）、ソリッドステートドライブ（SSD）、またはハードディスクドライブ（HDD））にセグメントを記憶することができる。本明細書で使用される場合、「データセグメント」は、分散コンピューティングシステム110の

40

50

顧客／テナントに詳細な結果を提供するためにフィルタリングおよび分析することができるデータキューブの個々の次元を指し得る。

【0035】

いくつかの実施形態では、分散コンピューティングシステム110は、クラスタを構成するさまざまなコンピューティングノード上のデータセグメントの配置を決定するように構成され得る。クラスタ内の特定のコンピューティングノードにデータセグメントを配置すると、コンピューティングクラスタのノードで特定のデータセグメントを読み取る必要がある場合に、データセグメントをより高速に取得できるようになる。分散コンピューティングシステム110は、クラスタを構成するさまざまなコンピューティングノード上のデータセグメントの配置を決定するために、さまざまなアプローチを使用することができる。例えば、本明細書に記載のアプローチでは、分散コンピューティングシステム110は、特定のハッシュ戦略を利用して、データセグメントの特定のセットを処理できるノードを識別することができる。分散コンピューティングシステム110のユーザ（例えば、顧客（複数可）102）によってクエリが送信されると、クラスタ（例えば、クラスタ122）内のドライバノード（例えば、ドライバノード124）は、データセグメントを記憶するワーカーノード（例えば、ワーカーノード128）を（例えば、一貫したハッシュ戦略を使用して）識別するように構成され得、これを使用してクエリを実行し、次いで、実行のためにクエリをワーカーノードに送信できる。

10

【0036】

特定の実施形態では、ユーザ（例えば、顧客102）は、場合によってはパブリックネットワーク108（例えば、インターネット）を介して分散コンピューティングシステム110に通信可能に結合されているコンピューティング装置104を介して、分散コンピューティングシステム110と対話することができる。コンピューティング装置104は、携帯電話、タブレット、デスクトップコンピュータなどを含むがこれらに限定されないさまざまなタイプのものであってよい。ユーザは、コンピューティング装置によって実行されるアプリケーションのコンソールユーザインターフェイス（UI）（グラフィカルユーザインターフェイス（GUI）であってもよい）を使用して、または分散コンピューティングシステム110によって提供されるAPI動作を介して、クラウドコンピューティングシステムと対話することができる。例えば、ユーザは、分散コンピューティングシステム110と対話して、1つまたは複数のコンピューティングクラスタを作成し、記憶システムに記憶されている既存のデータに対して対話型クエリを実行し、クエリ処理の結果として結果を取得することができる。

20

30

【0037】

一例として、分散コンピューティングシステム110のテナント120に関連付けられているユーザは、クラスタ122を含む1つまたは複数のクラスタを作成する要求を分散コンピューティングシステム110に送信することによって、分散コンピューティングシステム110と対話することができる。クラスタ作成要求は、分散コンピューティングシステム110内のロードバランサ112によって受信され、分散コンピューティングシステム内のマルチテナントプロキシサービス、例えばゲートウェイ114に要求を送信することができる。ゲートウェイ114は、ユーザの要求を認証／認可し、コンピューティングクラスタを作成するための動作を実行するように構成され得るAPIサーバ116に要求をルーティングする責任を負い得る。特定の例では、ゲートウェイ114は、テナント用のコンピューティングクラスタの作成を可能にするために、ユーザを認可し、ユーザの要求をAPIサーバ116に送信する共有マルチテナントハイパーテキスト転送プロトコル（HTTP）プロキシサービスを表すことができる。特定の例では、前述したように、クラスタの作成には、ドライバノードとワーカーノードのセットを含むノードのプールの作成が含まれ得る。テナントの専用サブネットの下に1つまたは複数のクラスタを作成できる。したがって、分散コンピューティングシステム110は、異なるテナント、例えばテナント118、119に属するコンピューティングクラスタ間の分離を提供する機能を含む。

40

50

【0038】

クラスタ内でのクエリの処理と実行を加速するために、分散コンピューティングシステム110内のクラスタ122は、クエリの計算に必要なデータをクラスタの異なるノード内のメモリ（本明細書では「キャッシュメモリ」または「キャッシュ」とも呼ばれる）内にキャッシュする機能を含み得る。例えば、キャッシュ（例えば、キャッシュ130）は、コンピューティングノードの中央処理装置の近くに配置されている、非常に高速で高価な少量のダイナミックランダムアクセスメモリ（DRAM）を表すことができる。特定の実施形態では、クラスタ内のノードには、ノードに関連付けられているキャッシュから、例えばキャッシュされるデータセグメントを含むキャッシュされるデータを取得することによって、クラスタ内のデータの効率的な処理および分析を実行するための改良される機能が提供される。特に、クラスタ内のワーカーノードに実行用のタスクが割り当てられており、関連するデータセグメントがキャッシュにない場合は、ワーカーノードは、キャッシュ内にデータセグメントを有するクラスタ内の隣接ノードを決定し、オブジェクト記憶装置156からデータセグメントを取得する代わりに、隣接ノードからセグメントを取得することができる。関連するキャッシュ内のデータセグメントを取得し維持するためにノードによって実行される動作の追加の詳細は、図2で詳細に説明される。

10

【0039】

図2は、いくつかの実施形態による、決定的キャッシュ技術を実装する分散コンピューティングシステム200内のコンピューティングノードのクラスタ202を示す。クラスタ202は、図1を参照して上述したクラスタ122と同様とすることができる。クラスタ202は、ドライバノード204およびワーカーノード206、208を含む複数のノードを含んでもよい。クラスタ202内のノードは、ノードに割り当てられている1つまたは複数のタスクに対応する動作を実行するように構成されている任意の適切な数の実行部（例えば、実行エンジンまたは実行プロセス）をホストし得る。図2に示されるように、タスクは、実行部210～213に割り当てられているタスク214～226を含むことができる。ワーカーノード206は2つの実行部210、211を利用し、一方、ワーカーノード208は2つの実行部212、213を利用する。さまざまな実施形態によれば、より多くのまたはより少ない実行部を有するより多くのまたはより少ないノードが可能である。各実行部はさらに、ワーカーノードに関連付けられているキャッシュの一部であり得る実行部メモリを備えることができる。例えば、実行部210は、キャッシュ228の一部を実行部メモリとして有し得、一方、実行部211は、キャッシュ228の第2の部分を実行部メモリとして有することができる。同様に、ワーカーノード208上で動作する実行部212、213は、キャッシュ230を利用することができる。実行部210～213は、タスク214～226で使用されるデータセグメントを記憶するためにキャッシュメモリ、例えばキャッシュ228、230を使用することができる。キャッシュ228、230は、各実行部の専用メモリリソースを超える追加のデータ動作（例えば、データの分割およびシャッフル）をサポートするためにワーカーノード206、208に提供されるより大きなメモリリソースの一部であってもよい。

20

30

【0040】

前述したように、いくつかの実施形態では、ドライバノード（例えば、ドライバノード204）は、ドライバ232（ドライバプログラム、ドライバモジュール、ドライバプロセスとも呼ばれる。つまり、コンピューティングクラスタ上に構築されているアプリケーションを実行するプロセス）を実行するように構成され得、アプリケーションのコンテキストを作成するための動作を実行し得る。アプリケーションは、ドライバ232がクエリ250を最適化するためのクエリオプティマイザ234（例えば、Apache Spark Catalyst）を含むことができるように、分析データ処理アプリケーションであってもよい。ドライバ204は、クラスタ内のすべてのノードにシステムリソースを割り当てるクラスタマネージャ236を含むことができる。クラスタマネージャ236は、図1に関して説明したクラスタマネージャ126と同様のものとすることができる。クラスタマネージャ236は、クラスタの状態を維持するように構成することができる。例

40

50

例えば、クラスタ状態には、ワーカーノードの実行部へのマッピングや、トークン境界のワーカーノードへのマッピングが含まれ得るが、これらに限定されない。クラスタマネージャ236は、例えばワーカーノードに障害が発生した場合など、クラスタ状態が変化したという指示に応答してクラスタ状態を更新することができる。クラスタマネージャ236は、追加の実行部および／またはワーカーノードがクラスタ内にプロビジョニングされるときに、クラスタ状態を更新することもできる。クラスタマネージャ236は、外部クラスタサービス、例えば、Kubernetesサービス（例えば、図1のKubernetesエンジン146）と通信して、クラスタ状態を維持することができる。

【0041】

いくつかの実施形態では、ドライバ204は、クラスタ内の実行部上で1つまたは複数のタスク、例えばタスク214～226を起動するためのスケジューラであり得るタスクマネージャ238も含み得る。タスクのスケジューリングは、クエリオプティマイザ234によって提供されるクエリ最適化に基づき得る。タスクは、タスクが対応するデータセグメントのコピーを含むキャッシュを有するワーカーノードで起動されるように、本明細書で説明される一貫したハッシング技術に基づいて実行部上で優先的に起動され得る。例として、クエリ250に対する応答において、タスク214は、タスク214に対応するデータセグメントがワーカーノード206に割り当てられているトークン境界内にあるハッシュトークン値を有することの識別に少なくとも部分的に基づいて、ワーカーノード206上で動作する実行部210においてスケジューリングされ得る。実行部210がデータセグメントをフェッチするとき、データセグメントがローカルキャッシュ228に存在する可能性が高く、その結果、実行部210がオブジェクト記憶装置256からデータセグメントをフェッチしなければならなかった場合よりも高速な取得がもたらされる。

【0042】

いくつかの実施形態では、ドライバ204は、キャッシュマネージャ240を含み得る。キャッシュマネージャ240は、実行部210～213上で動作するキャッシュマネージャ242～248と通信するように構成され得る。キャッシュマネージャは、キャッシュマネージャ間で要求を渡すリモートプロシージャコール（RPC）エンドポイントを実装するように構成できる。例えば、クエリ250の実行中に、キャッシュマネージャ246は、ローカルキャッシュ228を検査するために、隣接するワーカーノード206上で動作する実行部210、211への要求を呼び出すことができ、実行部212上のタスク220またはタスク222のいずれかによって必要とされるデータセグメントの存在を識別する。要求の呼び出しは、データセグメントのハッシュトークン値がワーカーノード206に割り当てられているトークン境界内にあると決定するキャッシュマネージャ246に少なくとも部分的に基づいてよい。RPC要求に応答して、キャッシュマネージャ242は、キャッシュ228内でデータセグメントを見つけ出し、そのデータセグメントをブロックマネージャ（例えば、分散コンピューティングシステム内のブロック記憶装置またはノード間記憶システム）に配置し、ブロックIDを、要求側キャッシュマネージャ246に返すことができる。次に、実行部212は、ブロックマネージャからデータセグメントを取得し、そのデータセグメントをキャッシュ230にコピーすることができる。

【0043】

いくつかの実施形態では、キャッシュマネージャ240は、キャッシュマネージャ242～248と通信して、キャッシュ228、230上で1つまたは複数のハウスキーピング動作を実行することができる。ハウスキーピング動作には、キャッシュに記憶されているデータを同期するための能動的な動作を含めることができる。能動的なハウスキーピング動作は、クラスタの状態の変化（例えば、クラスタにノードや実行部が追加または削除される場合）に応じて発生することがある。さらに、ハウスキーピング動作には、ローカルキャッシュから無効なデータを削除したり、使用頻度の低いデータをキャッシュから削除したりして、キャッシュスペースの可用性を維持し、ノードリソースの使用率を削減する受動的動作を含めることができる。ハウスキーピング動作の具体的な詳細については、図5と図6を参照して以下でさらに詳しく説明する。

【0044】

図3Aおよび図3Bは、本明細書で説明される実施形態による一貫したハッシュ技術の一例を提供するコードの断片を示す。示されるコードの特定の言語、スタイル、および構文は、本開示を限定するものではない。当業者であれば、記載されている機能を実装するための他の言語および構文を認識するであろう。

【0045】

図3Aは、いくつかの実施形態による、クラスタ内のノードへのトークン境界のマッピング例を示すコード300の断片である。図示されているように、マッピングは、「host_0」、「host_1」、および「host_2」として識別される3つのワーカーノードを含むクラスタを表している。ワーカーノードは、図2のワーカーノード206、208を含む、本明細書で説明される他のワーカーノードと同様であってもよい。ワーカーノードは2つの実行部をホストするように構成されており、クラスタ内では合計6つの実行部になる。他の実施形態では、クラスタは、任意の適切な数の実行部をホストする1つ、2つ、または3つより多いワーカーノードを有してもよい。これらの実施形態では、トークン境界のマッピングは、図3Aに示されるものと同様となる。

【0046】

前述したように、トークン境界は、トークン分割のセットを表す整数値のセットの分割を表すことができる。例えば、整数値は $0 \sim 2^{31} - 1$ の整数（つまり、32ビット符号付き整数のサイズ）にすることができる。図3Aのマッピングに対応するクラスタについては、トークン境界は、整数値のセットをトークン分割の6つの連続するサブセットに分割することができる。各ワーカーノードは、整数トークン境界の対間のトークン分割の範囲に対応する2つの「TokenBounds」セットにマッピングされる。例えば、host_1は、「TokenBounds(0, 357913941)」として識別されるトークン境界302の第1の対と、「TokenBounds(715827882, 1073741823)」として識別されるトークン境界304の第2の対にマッピングされる。説明されるクラスタ構成について特定の整数値を用いて示されているが、他の値は本開示の他の実施形態から得られ得る。

【0047】

いくつかの実施形態では、ドライバノードのドライバプロセス（例えば、ドライバノード204のドライバ232）は、クラスタ内のワーカーノードにトークン境界を均一に割り当てることができる。トークン境界の均一な割り当ては、各ワーカーノードがホストする実行部の数に基づいて各ワーカーノードに割り当てられるトークン境界の数を有するように、各ワーカーノードに割り当てられるトークン分割の数のバランスをとることができる。図3Aに示すように、3つのワーカーノードには、ワーカーノード上に構成されている2つの実行部の各々に1つずつ、2つのトークン境界対が割り当てられる。

【0048】

図3Bは、いくつかの実施形態による、クラスタ内の更新されるノードのセットへのトークン境界のマッピングの別の例を示すコード310の別の断片である。更新されるノードのセットは、図3Aに表されるノードの更新の結果であり得る。図に示されているように、マッピング例は2つのワーカーノードhost_0とhost_2を持つクラスタに対応している。このマッピングは、host_1ノードの障害が原因で発生し得る。クラスタ状態の変化に応答して、host_1に以前に割り当てられているトークン境界302の第1の対は、ドライバノードによってhost_0に対応するワーカーノードに再割り当てされ得る。同様に、以前にhost_1に割り当てられているトークン境界の第2の対は、host_0に対応するワーカーノードに再割り当てできる。その結果、トークン分割が均一に分散される2つのワーカーノードが生成される。

【0049】

いくつかの実施形態では、トークン境界は、クラスタ状態の変化に応答して更新される。例えば、図3Aのhost_1に対応するワーカーノードが故障した場合、ドライバノードは4つの実行部を持つクラスタに対応するようにトークン境界を更新できる。更新さ

10

20

30

40

50

れているトークン境界は、次いで、クラスタ内に残っているワーカーノードに割り当てることができる。次いで、トークンマッピングには4対のトークン境界があり、各2対が残りの2つのワーカーノードに割り当てられる。同様に、ワーカーノードおよび／または実行部がクラスタに追加される場合、追加される実行部およびワーカーノードに対応するようにトークン境界を更新できる。

【0050】

図4は、いくつかの実施形態による、決定的キャッシュを用いてクラスタ上のクエリを処理する例示的なプロセス400の簡略化されているフローチャートである。クラスタは、図2のクラスタ202および図1のクラスタ122および分散コンピューティングシステム110を含む、本明細書で説明される分散コンピューティングシステムのいずれかを含む、分散コンピューティングシステムのクラスタであってもよい。プロセス400は、論理フローチャートとして示されており、その各動作は、ハードウェア、コンピュータ命令、またはそれらの組み合わせで実装できる一連の動作を表す。コンピュータ命令の文脈では、動作は、1つまたは複数のプロセッサによって実行されると、記載される動作を実行する1つまたは複数のコンピュータ可読記憶媒体に記憶されるコンピュータ実行可能命令を表す。一般に、コンピュータ実行可能命令には、特定の機能を実行する、または特定のデータ型を実装するルーチン、プログラム、オブジェクト、構成要素、データ構造などが含まれる。動作を説明する順序は制限として解釈されることを意図したものではなく、説明される動作の任意の数を省略したり、任意の順序および／または並行して組み合わせることでプロセスを実装することができる。

【0051】

プロセス400（または本明細書で説明される任意の他のプロセス、またはその変形および／もしくは組み合わせ）の一部、任意、またはすべては、実行可能命令で構成されている1つまたは複数のコンピュータシステムの制御下で実行され得、ハードウェアまたはそれらの組み合わせによって、1つまたは複数のプロセッサ上で集合的に実行されるコード（例えば、実行可能命令、1つもしくは複数のコンピュータプログラム、または1つもしくは複数のアプリケーション）として実装され得る。コードは、例えば、1つまたは複数のプロセッサによって実行可能な複数の命令を含むコンピュータプログラムの形式で、コンピュータ可読記憶媒体に記憶され得る。コンピュータ可読記憶媒体は非一時的であってもよい。

【0052】

プロセス400は、ブロック402で始まり、分散コンピューティングシステム（例えば、分散コンピューティングシステム110）が、複数のノード（例えば、ドライバノード204、ワーカーノード206、208）を含むクラスタ（例えば、クラスタ122またはクラスタ202）を実装する。クラスタの実装には、クラスタの一部として、任意の適切な数のコンピューティング装置（物理コンピューティング装置および／または1つもしくは複数の物理コンピューティング装置上で実行される仮想マシンを含むが、これらに限定されない）をプロビジョニングすることが含まれ得、これには、クラスタとしてのコンピューティング装置の動作（例えば、複数のノードをホストする）をサポートするために、コンピューティング装置上に適切なソフトウェアを展開することが含まれる。ブロック404で、分散コンピューティングシステムはクラスタの状態を維持する。クラスタの状態は、1つまたは複数の実行部の複数のノードへのマッピング、およびマッピングに基づいて、複数のノード上で実行する1つまたは複数の実行部に複数のトークン境界を割り当てるステップを含むことができる。クラスタの状態の維持には、例えば、クラスタ内でノードや実行部が追加または削除される場合など、クラスタ状態の変化に応じてマッピングを更新することが含まれ得る。クラスタ状態の維持には、複数のトークン境界を更新し、更新されているトークン境界を分散コンピューティングシステム内で実行中の1つまたは複数の実行部に均一に割り当てることも含まれ得る。

【0053】

ブロック406で、ドライバノードは実行のためのクエリを受信することができる。ク

エリは、クラスタのテナントに関連付けられているユーザによって送信されるクエリ（例えば、クエリ250）であり得る。クエリを実行するために、ドライバノードは、クラスタのワーカーノードによって処理される1つまたは複数のデータセグメントのセットを識別できる。データセグメントは、例えば、オフライン記憶装置、オブジェクト記憶装置、またはその他の同様の記憶システムなどのディープ記憶装置に残ることができる。特定の実施形態によれば、データセグメントは、クラスタ内のワーカーノードに関連付けられている1つまたは複数のキャッシュにも存在し得る。

【0054】

ブロック408で、ドライバノードは、1つまたは複数のデータセグメントのセットに対応するトークンのセットを計算することができる。各データセグメントは、データセグメントを一意に識別する対応するトークンを持つことができる。トークンは、MurmurHash3またはその他の適切なハッシュアルゴリズムを含むハッシュアルゴリズムを使用して計算できる。ハッシュアルゴリズムへの入力は、データセグメントに関連付けられている識別情報（例えば、ファイル名）にすることができる。

【0055】

ブロック412で、ドライバノードは、ドライバノードで実行されるタスクマネージャ、スケジューラ、または他の同様のプロセスと連携して、第1のワーカーノードで実行される第1の実行部上で第1のタスクを起動することができる。第1のタスクは、第1のデータセグメントに対する動作に対応し得るため、第1の実行部が第1のデータセグメントを取得し、要求される動作を実行し、タスクを完了することができる。第1のタスクは、クエリ実行を構成する複数のタスクのうちの1つのタスクであってもよい。第1のワーカーノードは、第1のワーカーノードに関連付けられているトークン境界の第1の対に対応するトークンのセットの第1のトークンに少なくとも部分的に基づいて選択することができる。例えば、第1のデータセグメントのハッシュ計算では、トークンの特定の整数値が返され得る。この値は、トークン境界の第1の対によって規定される整数値の範囲内に収まり得る。

【0056】

ブロック414で、第1のワーカーノードは第1のデータセグメントを取得することができる。第1のデータセグメントは、第1のワーカーノードに関連付けられているキャッシュ内に存在し得る。第1のデータセグメントは、クラスタ内の第2のワーカーノードに関連付けられているキャッシュに存在することも、ノードから離れたデータストア、データベース、オブジェクト記憶装置、ディープ記憶装置、またはその他のリポジトリに存在することもできる。第1のワーカーノードに関連付けられているキャッシュから第1のデータセグメントを取得することは、第1の実行部がキャッシュ内の第1のデータセグメントを識別し、その内容を能動的メモリに読み込むことを含むことができる。第2のワーカーノードに関連付けられているキャッシュから第1のデータセグメントを取得することは、第1の実行部が複数のノードのうちの1つまたは複数の隣接ノードに要求を送信することを含むことができる。隣接ノード上で実行する実行部は、要求に基づいて、各隣接ノードに関連付けられているキャッシュを検査できる。1つまたは複数の隣接ノードの第2のワーカーノード上で実行する第2の実行部は、第1のデータセグメントが第2のキャッシュに存在することを決定することができ、そのデータセグメントをブロックマネージャ（例えば、ブロック記憶装置またはノード間記憶システムに関連付けられている）または分散コンピューティングシステム内のノード間のデータ転送を扱うように構成されている他のモジュールに配置し、次に、第1のデータセグメントのIDを、要求を行った第1の実行部に送信できる。次に、第1のワーカーノードは、第2のワーカーノードから第1のワーカーノードに関連付けられている第1のキャッシュにデータセグメントをコピーできる。

【0057】

図5は、いくつかの実施形態による、クラスタ状態が変化したときにクラスタ内の決定的キャッシュを安定化するための例示的なプロセス500の別の簡略化されているフロー

チャートである。クラスタは、図 2 のクラスタ 2 0 2 および図 1 のクラスタ 1 2 2 および分散コンピューティングシステム 1 1 0 を含む、本明細書で説明される分散コンピューティングシステムのいずれかを含む、分散コンピューティングシステムのクラスタであってもよい。

【0058】

プロセス 5 0 0 は、ブロック 5 0 2 で、ドライバノード（例えばドライバノード 2 3 2）がクラスタ内の 1 つまたは複数のワーカーノード（例えばワーカーノード 2 0 6、2 0 8）にハウスキーピング要求を送信することで開始することができる。ハウスキーピング要求は、クラスタの状態の変化、例えば、ワーカーノードまたは実行部の追加または削除に応答して生成され、送信され得る。クラスタの状態が変化すると、新しい構成がクラスタ内のキャッシュされるデータセグメントの場所に影響を与え得る。ハウスキーピング動作を実行すると、キャッシュが能動的に安定化し得る。

10

【0059】

ハウスキーピング要求に応答して、ブロック 5 0 4 で、ワーカーノード上で動作する 1 つまたは複数の実行部は、データセグメントを求めてワーカーノードに関連付けられているキャッシュをスキャンすることができる。例えば、第 1 のワーカーノードで実行する第 1 の実行部は、第 1 のワーカーノードに関連付けられているキャッシュをスキャンして、データセグメントを識別することができる。同様に、第 1 のワーカーノードで実行されている別の実行部も、第 1 のワーカーノードに関連付けられているキャッシュをスキャンしてデータセグメントを識別できる。第 1 のワーカーノード上のキャッシュは第 1 の実行部と別の実行部によって共有されるため、各実行部はキャッシュ内のすべてのデータセグメントをスキャンして、データセグメントが実行部に割り当てられているトークン境界の対に対応するかどうかを決定できる。

20

【0060】

決定 5 0 6 で、実行部は、スキャンされるデータセグメントに関連付けられているトークンが実行部に割り当てられるトークン境界内かどうかを評価する。例えば、第 1 の実行部には、その実行部が「所有する」トークン値の範囲を規定するトークン境界の第 1 の対を割り当てることができる。第 1 のワーカーノードに関連付けられているキャッシュをスキャンするとき、第 1 の実行部は、存在するデータセグメントのトークンが第 1 の実行部のトークン境界内の値に対応するかどうかを決定できる。実行部は、クエリの実行時にドライバノードが使用するのと同じハッシュアルゴリズムを使用してトークン値を計算できる。いくつかの実施形態では、トークン値は、トークンのハッシュ化されるセグメントキーテーブルの一部としてドライバノードによって実行部に渡すことができる。データセグメントトークンが実行部のトークン境界内にある場合、セグメントはキャッシュに保持される。データセグメントトークンが実行部のトークン境界内でない場合、ブロック 5 1 0 で実行部はクラスタ内の他のすべての実行部にセグメント ID を送信する。

30

【0061】

ブロック 5 1 2 で、外れ値セグメント ID を受信する実行部は、他のワーカーノードからローカルキャッシュに外れ値セグメントをコピーすることができる。例えば、第 1 の実行部は、第 1 のワーカーノードに関連付けられているキャッシュ内の 1 つまたは複数の外れ値データセグメントを識別することができる。クラスタの現在の状態とトークン境界の現在の割り当てに基づいて、これらの外れ値データセグメントの 1 つまたは複数が、第 2 のワーカーノードの第 2 の実行部によって「所有」され得る。第 2 のワーカーノードは、第 1 のワーカーノードから第 2 のワーカーノードに関連付けられているキャッシュにデータセグメントをコピーできる。このようにして、クラスタの状態に応じてデータセグメントが予想されるキャッシュ内に存在することができ、クラスタの状態が変化した後キャッシュを安定させることができる。

40

【0062】

図 6 は、タイムプロセスに従って決定的キャッシュをクリーニングするための例示的なプロセス 6 0 0 の簡略化されているフローチャートである。クラスタは、図 2 のクラスタ

50

202 および図1のクラスタ122および分散コンピューティングシステム110を含む、本明細書で説明される分散コンピューティングシステムのいずれかを含む、分散コンピューティングシステムのクラスタであってもよい。プロセス600は、図5に関して上述した「能動的な」ハウスキーピング動作を補完する「受動的な」ハウスキーピング動作を表すことができる。

【0063】

入力601において、ドライバノード（例えば、ドライバノード232）は、ドライバノード上に実装されているタイマプロセスからタイマ指示を受信することができる。タイマは、固定スケジュールに従ってプロセス600の動作をトリガするように構成され得る。いくつかの実施形態では、タイマ間隔は、クラスタパラメータまたはクラスタに関連する他の条件（例えば、クラスタ負荷、クラスタ状態など）に従って変化し得る。タイマに応答して、ブロック602で、ドライバノードは、クラスタ内の1つまたは複数のワーカーノード（例えば、ワーカーノード206、208）にハウスキーピング要求を送信することができる。ハウスキーピング要求は、無効なデータセグメントやアクセス頻度の低いデータセグメントを削除するために分散キャッシュを「クリーニング」することで、クラスタ内のリソースの使用量を削減できる。

【0064】

ハウスキーピング要求に応答して、ブロック604で、ワーカーノード上で動作する1つまたは複数の実行部は、図5のブロック504と同様に、データセグメントについてワーカーノードに関連付けられているキャッシュをスキャンすることができる。決定606で、1つまたは複数の実行部は、キャッシュ内のデータセグメントが有効であるかどうかを評価する。クエリの実行中に、キャッシュされるデータセグメントがオブジェクト記憶装置に記憶されている現在のデータを表さなくなるように、データセグメントが更新または変更され得る。オブジェクト記憶装置内のデータと一致しなくなった分散キャッシュ内のデータセグメントは無効であるが、削除されるまでキャッシュ内に保持され得る。キャッシュされているデータセグメントが有効かどうかを決定するために、1つまたは複数の実行部はオブジェクト記憶装置内のデータキューブごとに有効なセグメントリストを構築できる。有効セグメントリストに基づいて、1つまたは複数の実行部は、キャッシュされているデータセグメントが有効かどうかを決定できる。データセグメントが有効である場合、ブロック608でそれらはキャッシュ内に保持される。データセグメントが無効である場合、それらはブロック610でキャッシュから退去させられる（例えば、削除される）。

【0065】

決定612で、ドライバノードは、利用可能なキャッシュ記憶が閾値未満であるかどうかを決定することができる。いくつかの実施形態では、利用可能な記憶が閾値未満であるという決定は、ノード上で実行する1つまたは複数の実行部またはキャッシュマネージャによって行われてもよい。いくつかの実施形態では、閾値はクラスタ構成の一部として構成することができ、例えば、キャッシュ内に構成されている合計スペースの80%とすることができる。利用可能なキャッシュスペースが閾値を超えている場合、ハウスキーピング動作はエンドポイント614で終了することができる。

【0066】

利用可能なキャッシュ記憶スペースが閾値未満である場合、プロセス600は決定616に進むことができる。図5に関して説明した動作と同様に、決定616で、1つまたは複数の実行部は、スキャンされるデータセグメントに関連付けられているトークンが実行部に割り当てられるトークン境界内かどうかを評価することができる。そうである場合、ブロック618で、セグメントはキャッシュ内に保持される。そうでない場合、ブロック620で、トークン境界の現在の割り当てに従って、外れ値セグメントをクラスタ内の他のワーカーノードにコピーすることができる。ブロック622で、外れ値データセグメントをローカルキャッシュから削除することができる。削除は、データセグメントが別のワーカーノードにコピーされた後に発生し得る。

10

20

30

40

50

【0067】

決定624では、決定612と同様に、利用可能なキャッシュ記憶スペースが再度チェックされ、それが閾値未満であるかどうか決定される。利用可能なスペースが閾値を超えている場合、ハウスキーピング動作はエンドポイント626で終了できる。使用可能なスペースが設定されている閾値を下回っているため、キャッシュにまだスペースの制約がある場合は、第3の技術を使用してキャッシュをクリーニングできる。ブロック628で、1つまたは複数の実行部は、分散キャッシュに記憶されるデータセグメントの温度を決定することができる。本明細書で使用する場合、「温度」とは、「高温」データセグメントがクラスタによって実行されるさまざまなクエリによって時間の経過とともに頻繁にアクセスされるように、キャッシュ内のデータセグメントへのアクセス頻度の尺度を指す。対照的に、「低温」データセグメントはアクセス頻度が低くキャッシュ内に存在する。いくつかの実施形態では、セグメントの温度は、「高温」、「温和な温度」、または「低温」に分類することができる。温度は、データセグメントへの以前のアクセス間の間隔を計算することによって決定できる。間隔が減少している場合（例えば、最近の読み取り頻度が高くなった場合）、そのセグメントの温度レベルはアップグレードされる（例えば、低温から温和な温度、または温和な温度から高温に）。間隔が増加している場合（例えば、読み取りの頻度が最近低下した場合）、そのセグメントの温度レベルは引き下げられる（例えば、高温から温和な温度、または温和な温度から低温）。例えば、データセグメントは10秒前にキャッシュ内で最初にアクセスされ、5秒前に2回目にアクセスされ、2秒前に3回目にアクセスされ得る。したがって、最後のアクセス間隔は減少するため、セグメントの温度をアップグレードすることができる（例えば、温和な温度から高温に）。

10

20

【0068】

ブロック630で、キャッシュ内のセグメントの温度に基づいて、1つまたは複数の実行部は、最も低温のセグメントをキャッシュから削除することができる。いくつかの実施形態では、これは、「低温」温度を有するデータセグメントの1つ、一部、またはすべてを排除することを意味し得る。最も低温のデータセグメントを削除した後、利用可能なキャッシュスペースが再度チェックされる。最も低温のデータセグメントを削除するプロセスは、利用可能なキャッシュスペースが閾値を超えるまで繰り返される。

【0069】

いくつかの実施形態では、データセグメントの追加の統計には、キャッシュヒット、キャッシュミス、無効化の数、サイズ、平均アクセス間隔、レプリケーションファクタ、およびセグメントがオブジェクト記憶装置から取得されたかノード間記憶装置から取得されたかが含まれるが、これらに限定されない。どのデータセグメントをキャッシュから削除するかを決定するための温度に加えて、データセグメント統計値の他の組み合わせを使用することもできる。

30

【0070】

サービスとしてのインフラストラクチャのアーキテクチャの例

上で述べたように、サービスとしてのインフラストラクチャ（IaaS）は、クラウドコンピューティングの特定のタイプの1つである。IaaSは、パブリックネットワーク（インターネットなど）経由で仮想化されるコンピューティングリソースを提供するように構成できる。IaaSモデルでは、クラウドコンピューティングプロバイダは、インフラストラクチャ構成要素（例えば、サーバ、記憶装置、ネットワークノード（例えばハードウェア）、展開ソフトウェア、プラットフォーム仮想化（例えばハイパーバイザ層）など）をホストすることができる。場合によっては、IaaSプロバイダは、これらのインフラストラクチャ構成要素に付随するさまざまなサービス（例えば、請求、監視、ログイン、負荷分散、およびクラスタリングなど）を提供することもできる。したがって、これらのサービスはポリシー推進型であり得るため、IaaSユーザは負荷分散を推進するポリシーを実装して、アプリケーションの可用性と性能を維持できる可能性がある。

40

【0071】

場合によっては、IaaS顧客は、インターネットなどのワイドエリアネットワーク（

50

W A N) を介してリソースおよびサービスにアクセスすることができ、クラウドプロバイダのサービスを使用してアプリケーションスタックの残りの要素をインストールすることができる。例えば、ユーザは I a a S プラットフォームにログインして、仮想マシン (V M (複数可)) の作成、各 V M へのオペレーティングシステム (O S) のインストール、データベースなどのミドルウェアの展開、ワークロードとバックアップ用の記憶バケットの作成、さらにはエンタープライズソフトウェアをその V M にインストールすることができる。その後、顧客はプロバイダのサービスを使用して、ネットワークトラフィックのバランス、アプリケーションの問題のトラブルシューティング、性能の監視、災害復旧の管理などのさまざまな機能を実行できる。

【 0 0 7 2 】

10

ほとんどの場合、クラウドコンピューティングモデルはクラウドプロバイダの参加を必要とし得る。クラウドプロバイダは、I a a S の提供 (例えば、提供、レンタル、販売) を専門とするサードパーティサービスであってもかまわないが、そうである必要はない。エンティティはプライベートクラウドを展開して、独自のインフラストラクチャサービスプロバイダになることも選択できる。

【 0 0 7 3 】

いくつかの例では、I a a S 展開は、新しいアプリケーション、またはアプリケーションの新しいバージョンを、準備されているアプリケーションサーバなどに配置するプロセスである。これには、サーバを準備するプロセス (例えば、ライブラリ、デーモンなどのインストールなど) も含まれ得る。これは多くの場合、ハイパーバイザ層 (例えば、サーバ、記憶装置、ネットワークハードウェア、および仮想化) の下のクラウドプロバイダによって管理される。したがって、顧客は、(O S)、ミドルウェア、および／またはアプリケーション展開 (例えば、セルフサービス仮想マシン (例えば、オンデマンドでスピンアップできる) など) などの取り扱いに責任を負うことができる。

20

【 0 0 7 4 】

いくつかの例では、I a a S プロビジョニングは、使用するコンピュータまたは仮想ホストを取得し、それらに所望のライブラリまたはサービスをインストールすることさえも指し得る。ほとんどの場合、展開にはプロビジョニングが含まれていないため、最初にプロビジョニングを実行することが望ましい場合がある。

【 0 0 7 5 】

30

場合によっては、I a a S プロビジョニングには2つの異なる課題がある。まず、何かを実行する前にインフラストラクチャの初期セットをプロビジョニングするという最初の課題がある。第2に、すべてがプロビジョニングされた後に、既存のインフラストラクチャを進化させるという課題がある (例えば、新しいサービスの追加、サービスの変更、サービスの削除など)。場合によっては、インフラストラクチャの構成を宣言的に規定できるようにすることで、これら2つの課題に対処できる場合がある。言い換えれば、インフラストラクチャ (例えば、どの構成要素が必要か、どのように対話するか) は1つまたは複数の構成ファイルによって規定できる。したがって、インフラストラクチャの全体的なトポロジ (例えば、どのリソースがどのリソースに依存するか、およびそれぞれがどのように連携するかなど) を宣言的に記述することができる。場合によっては、トポロジが規定されると、構成ファイルに記述されているさまざまな構成要素を作成および／または管理するワークフローを生成できる。

40

【 0 0 7 6 】

いくつかの例では、インフラストラクチャは、相互接続されている多くの要素を有し得る。例えば、コアネットワークとしても知られる、1つまたは複数の仮想プライベートクラウド (V P C (複数可)) (例えば、構成可能および／または共有コンピューティングリソースの潜在的にオンデマンドのプール) が存在し得る。いくつかの例では、ネットワークの受信トラフィックおよび／または送信トラフィックがどのように設定されるかを規定するためにプロビジョニングされる1つまたは複数の受信／送信トラフィックグループルールと、1つまたは複数の仮想マシン (V M (複数可)) が存在する場合もある。ロー

50

ドバランサ、データベースなどの他のインフラストラクチャ要素もプロビジョニングできる。より多くのインフラストラクチャ要素が望まれたり追加されたりするにつれて、インフラストラクチャは段階的に進化し得る。

【0077】

場合によっては、さまざまな仮想コンピューティング環境にわたるインフラストラクチャコードの展開を可能にするために、継続的展開技術が使用されてもよい。さらに、説明されている技術により、これらの環境内でのインフラストラクチャ管理が可能になる。いくつかの例では、サービスチームは、1つまたは複数の、しかし多くの場合、多くの異なる生産環境（例えば、さまざまな異なる地理的位置にまたがり、場合によっては全世界に及ぶ）に展開することが望ましいコードを書くことができる。しかし、例によっては、コードを展開するインフラストラクチャを最初にセットアップすることが望ましい場合がある。場合によっては、プロビジョニングは手動で行うことができ、プロビジョニングツールを利用してリソースをプロビジョニングすることができ、および／または展開ツールを利用して、インフラストラクチャがプロビジョニングされた後にコードを展開することができる。

10

【0078】

図7は、少なくとも1つの実施形態による、IaaSアーキテクチャのパターン例を示すブロック図700である。サービスオペレータ702は、仮想クラウドネットワーク（VCN）706およびセキュアホストサブネット708を含むことができるセキュアホストテナント704に通信可能に結合することができる。いくつかの例では、サービスオペレータ702は、1つまたは複数のクライアントコンピューティング装置を使用することができ、これは、ポータブルハンドヘルド装置（例えば、iPhone（登録商標）、携帯電話、iPad（登録商標）、コンピューティングタブレット、携帯情報端末（PDA））もしくはウェアラブル装置（例えば、Google（登録商標）Glassヘッドマウントディスプレイ）、Microsoft Windows Mobile（登録商標）などの実行ソフトウェア、および／またはiOS、Windows Phone、Android、BlackBerry 8、Palm OSなどのさまざまなモバイルオペレーティングシステム、ならびにインターネット、電子メール、ショートメッセージサービス（SMS）、BlackBerry（登録商標）、または有効な他の通信プロトコルであり得る。あるいは、クライアントコンピューティング装置は、例えば、さまざまなバージョンのMicrosoft Windows（登録商標）、Apple Macintosh（登録商標）、および／またはLinux（登録商標）オペレーティングシステム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汎用パーソナルコンピュータであってもよい。クライアントコンピューティング装置は、例えばGoogle Chrome OSなどのさまざまなGNU/Linuxオペレーティングシステムを含むがこれらに限定されない、さまざまな市販のUNIX（登録商標）またはUNIX類似のオペレーティングシステムのいずれかを実行するワークステーションコンピュータであり得る。代替として、または追加として、クライアントコンピューティング装置は任意の他の電子装置であってもよく、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（例えば、Kinect（登録商標）ジェスチャ入力装置であり、または無しのMicrosoft Xboxゲームコンソール）、および／またはVCN 706および／またはインターネットにアクセスできるネットワークを介して通信できるパーソナルメッセージング装置などである。

20

30

40

【0079】

VCN 706は、SSH VCN 712に含まれるLPG 710を介してセキュアシェル（SSH）VCN 712に通信可能に結合することができるローカルピアリングゲートウェイ（LPG）710を含むことができる。SSH VCN 712は、SSHサブネット714を含むことができ、SSH VCN 712は、制御プレーンVCN 716に含まれるLPG 710を介して制御プレーンVCN 716に通信可能に結合することができる。また、SSH VCN 712は、LPG 710を介してデータプレーンVCN 718に通信可能に結合することができる。制御プレーンVCN 716およびデータプレーンVCN 718

50

は、IaaSプロバイダが所有および／または動作できるサービステナント719に含めることができる。

【0080】

制御プレーンVCN716は、境界ネットワーク（例えば、企業イントラネットと外部ネットワークとの間の企業ネットワークの一部）として機能する制御プレーン非武装地帯（DMZ）層720を含むことができる。DMZベースのサーバは責任が制限されており、侵害を阻止するのに役立ち得る。さらに、DMZ層720は、1つまたは複数のロードバランサ（LB）サブネット（複数可）722、アプリサブネット（複数可）726を含むことができる制御プレーンアプリ層724、制御プレーンデータ層728を含むことができ、これには、データベース（DB）サブネット（複数可）730（例えば、フロントエンドDBサブネット（複数可）および／またはバックエンドDBサブネット（複数可））を含めることができる。制御プレーンDMZ層720に含まれるLBサブネット（複数可）722は、制御プレーンアプリ層724に含まれるアプリサブネット（複数可）726および制御プレーンVCN716に含まれ得るインターネットゲートウェイ734に通信可能に結合することができ、アプリサブネット（複数可）726は、制御プレーンデータ層728に含まれるDBサブネット（複数可）730、ならびにサービスゲートウェイ736およびネットワークアドレス変換（NAT）ゲートウェイ738に通信可能に結合することができる。制御プレーンVCN716は、サービスゲートウェイ736およびNATゲートウェイ738を含むことができる。

10

【0081】

制御プレーンVCN716は、アプリサブネット（複数可）726を含むことができるデータプレーンミラーアプリ層740を含むことができる。データプレーンミラーアプリ層740に含まれるアプリサブネット726（複数可）は、コンピューティングインスタンス744を実行できる仮想ネットワークインターフェイスコントローラ（VNIC）742を含むことができる。コンピューティングインスタンス744は、データプレーンミラーアプリ層740のアプリサブネット（複数可）726を、データプレーンアプリ層746に含めることができるアプリサブネット（複数可）726に通信可能に結合することができる。

20

【0082】

データプレーンVCN718は、データプレーンアプリ層746、データプレーンDMZ層748、およびデータプレーンデータ層750を含むことができる。データプレーンDMZ層748は、データプレーンアプリ層746のアプリサブネット（複数可）726およびデータプレーンVCN718のインターネットゲートウェイ734に通信可能に結合され得るLBサブネット（複数可）722を含むことができる。アプリサブネット（複数可）726は、データプレーンVCN718のサービスゲートウェイ736およびデータプレーンVCN718のNATゲートウェイ738に通信可能に結合することができる。データプレーンデータ層750は、データプレーンアプリ層746のアプリサブネット（複数可）726に通信可能に結合できるDBサブネット（複数可）730を含むこともできる。

30

【0083】

制御プレーンVCN716およびデータプレーンVCN718のインターネットゲートウェイ734は、パブリックインターネット754に通信可能に結合され得るメタデータ管理サービス752に通信可能に結合され得る。パブリックインターネット754は、制御プレーンVCN716およびデータプレーンVCN718のNATゲートウェイ738に通信可能に接続することができる。制御プレーンVCN716およびデータプレーンVCN718のサービスゲートウェイ736は、クラウドサービス756に通信可能に結合することができる。

40

【0084】

いくつかの例では、制御プレーンVCN716またはデータプレーンVCN718のサービスゲートウェイ736は、パブリックインターネット754を経由せずに、クラウド

50

サービス 756 へのアプリケーションプログラミングインターフェイス (API) 呼び出しを行うことができる。サービスゲートウェイ 736 からクラウドサービス 756 への API 呼び出しは一方方向であり得る：サービスゲートウェイ 736 はクラウドサービス 756 への API 呼び出しを行うことができ、クラウドサービス 756 は要求されるデータをサービスゲートウェイ 736 に送信することができる。しかし、クラウドサービス 756 は、サービスゲートウェイ 736 への API 呼び出しを開始できない場合がある。

【0085】

いくつかの例では、セキュアホストテナント 704 は、サービステナント 719 に直接接続することができ、そうでなければ分離され得る。セキュアホストサブネット 708 は、LPG 710 を介して SSH サブネット 714 と通信することができ、LPG 710 は、そうでなければ分離されるシステムを介した双方向通信を可能にすることができる。セキュアホストサブネット 708 を SSH サブネット 714 に接続すると、セキュアホストサブネット 708 にサービステナント 719 内の他のエンティティへのアクセスを与えることができる。

【0086】

制御プレーン VCN 716 により、サービステナント 719 のユーザが所望のリソースをセットアップまたはプロビジョニングできるようにすることができる。制御プレーン VCN 716 内にプロビジョニングされる所望のリソースは、データプレーン VCN 718 内に展開または使用され得る。いくつかの例では、制御プレーン VCN 716 はデータプレーン VCN 718 から分離することができ、制御プレーン VCN 716 のデータプレーンミラーアプリ層 740 は、データプレーンミラーアプリ層 740 およびデータプレーンアプリ層 746 に含まれることができる VNIC (複数可) 742 を介して、データプレーン VCN 718 のデータプレーンアプリ層 746 と通信することができる。

【0087】

いくつかの例では、システムのユーザまたは顧客は、要求をメタデータ管理サービス 752 に通信することができるパブリックインターネット 754 を介して、例えば作成、読み取り、更新、または削除 (CRUD) 動作などの要求を行うことができる。メタデータ管理サービス 752 は、インターネットゲートウェイ 734 を介して要求を制御プレーン VCN 716 に通信することができる。この要求は、制御プレーン DMZ 層 720 に含まれる LB サブネット (複数可) 722 によって受信され得る。LB サブネット (複数可) 722 は、要求が有効であると決定することができ、この決定に応答して、LB サブネット (複数可) 722 は、制御プレーンアプリ層 724 に含まれるアプリサブネット (複数可) 726 に要求を送信することができる。要求が検証され、パブリックインターネット 754 への呼び出しが必要な場合、パブリックインターネット 754 への呼び出しは、パブリックインターネット 754 への呼び出しを行うことができる NAT ゲートウェイ 738 に送信され得る。要求によって記憶されることが望ましい場合があるメモリは、DB サブネット (複数可) 730 に記憶できる。

【0088】

いくつかの例では、データプレーンミラーアプリ層 740 は、制御プレーン VCN 716 とデータプレーン VCN 718 との間の直接通信を容易にすることができる。例えば、構成に対する変更、更新、または他の適切な修正を、データプレーン VCN 718 に含まれるリソースに適用することが望ましい場合がある。VNIC 742 を介して、制御プレーン VCN 716 は、データプレーン VCN 718 に含まれるリソースと直接通信することができ、それにより、データプレーン VCN 718 に含まれるリソースに対する構成の変更、更新、または他の適切な修正を実行することができる。

【0089】

いくつかの実施形態では、制御プレーン VCN 716 およびデータプレーン VCN 718 は、サービステナント 719 に含めることができる。この場合、システムのユーザまたは顧客は、制御プレーン VCN 716 またはデータプレーン VCN 718 のいずれも所有または動作することはできない。代わりに、IaaS プロバイダは、制御プレーン VCN

716 およびデータプレーンVCN718を所有または動作することができ、これらは両方ともサービステナント719に含まれ得る。この実施形態は、ユーザまたは顧客が他のユーザまたは他の顧客のリソースと対話することを防止し得るネットワークの分離を可能にすることができる。また、この実施形態により、システムのユーザまたは顧客は、記憶のために所望のレベルの脅威防止を持たない可能性があるパブリックインターネット754に依存する必要なく、データベースをプライベートに記憶することができる。

【0090】

他の実施形態では、制御プレーンVCN716に含まれるLBサブネット（複数可）722は、サービスゲートウェイ736から信号を受信するように構成され得る。この実施形態では、制御プレーンVCN716およびデータプレーンVCN718は、パブリックインターネット754を呼び出すことなく、IaaSプロバイダの顧客によって呼び出されるように構成され得る。顧客が使用するデータベース（複数可）はIaaSプロバイダによって制御され、パブリックインターネット754から隔離され得るサービステナント719に記憶され得るため、IaaSプロバイダの顧客は、この実施形態を望む可能性がある。

【0091】

図8は、少なくとも1つの実施形態による、IaaSアーキテクチャの別のパターン例を示すブロック図800である。サービスオペレータ802（例えば、図7のサービスオペレータ702）は、セキュアホストテナント804（例えば、図7のセキュアホストテナント704）に通信可能に結合することができ、これは、仮想クラウドネットワーク（VCN）806（例えば、図7のVCN706）およびセキュアホストサブネット808（例えば、図7のセキュアホストサブネット708）を含むことができる。VCN806は、ローカルピアリングゲートウェイ（LPG）810（例えば、図7のLPG710）を含むことができ、これは、SSHVCN812に含まれるLPG710を介してセキュアシェル（SSH）VCN812（例えば、図7のSSHVCN712）に通信可能に結合することができる。SSHVCN812は、SSHサブネット814（例えば、図7のSSHサブネット714）を含むことができ、SSHVCN812は、制御プレーンVCN816に含まれるLPG810を介して、制御プレーンVCN816（例えば、図7の制御プレーンVCN716）に通信可能に結合することができる。制御プレーンVCN816は、サービステナント819（例えば、図7のサービステナント719）に含めることができ、データプレーンVCN818（例えば、図7のデータプレーンVCN718）は、システムのユーザまたは顧客によって所有または動作され得る顧客テナント821に含めることができる。

【0092】

制御プレーンVCN816は、LBサブネット（複数可）822（例えば、図7のLBサブネット（複数可）722）を含むことができる制御プレーンDMZ層820（例えば、図7の制御プレーンDMZ層720）、アプリサブネット（複数可）826（例えば、図7のアプリサブネット（複数可）726）を含むことができる制御プレーンアプリ層824（例えば、図7の制御プレーンアプリ層724）、データベース（DB）サブネット（複数可）830（例えば、図7のDBサブネット（複数可）730と同様）を含むことができる制御プレーンデータ層828（例えば、図7の制御プレーンデータ層728）を含むことができる。制御プレーンDMZ層820に含まれるLBサブネット（複数可）822は、制御プレーンアプリ層824に含まれるアプリサブネット（複数可）826と、制御プレーンVCN816に含まれ得るインターネットゲートウェイ834（例えば、図7のインターネットゲートウェイ734）とに通信可能に結合することができ、アプリサブネット（複数可）826は、制御プレーンデータ層828に含まれるDBサブネット（複数可）830、ならびにサービスゲートウェイ836（例えば、図7のサービスゲートウェイ）およびネットワークアドレス変換（NAT）ゲートウェイ838（例えば、図7のNATゲートウェイ738）に通信可能に結合することができる。制御プレーンVCN816は、サービスゲートウェイ836およびNATゲートウェイ838を含むことがで

きる。

【0093】

制御プレーンVCN816は、アプリサブネット（複数可）826を含むことができるデータプレーンミラーアプリ層840（例えば、図7のデータプレーンミラーアプリ層740）を含むことができる。データプレーンミラーアプリ層840に含まれるアプリサブネット（複数可）826は、コンピューティングインスタンス844（例えば、図7のコンピューティングインスタンス744と同様）を実行できる仮想ネットワークインターフェイスコントローラ（VNIC）842（例えば、742のVNIC）を含むことができる。コンピューティングインスタンス844は、データプレーンミラーアプリ層840のアプリサブネット（複数可）826とアプリサブネット（複数可）826との間の通信を容易にすることができ、これは、データプレーンミラーアプリ層840に含まれるVNIC842およびデータプレーンアプリ層846に含まれるVNIC842を介して、データプレーンアプリ層846（例えば、図7のデータプレーンアプリ層746）に含まれることができる。

10

【0094】

制御プレーンVCN816に含まれるインターネットゲートウェイ834は、メタデータ管理サービス852（例えば、図7のメタデータ管理サービス752）に通信可能に結合することができ、これは、パブリックインターネット854（例えば、図7のパブリックインターネット754）に通信可能に結合することができる。パブリックインターネット854は、制御プレーンVCN816に含まれるNATゲートウェイ838に通信可能に結合することができる。制御プレーンVCN816に含まれるサービスゲートウェイ836は、クラウドサービス856（例えば、図7のクラウドサービス756）に通信可能に結合することができる。

20

【0095】

いくつかの例では、データプレーンVCN818は、顧客テナント821に含まれることができる。この場合、IaaSプロバイダは、各顧客に対して制御プレーンVCN816を提供することができ、IaaSプロバイダは、各顧客に対して、サービステナント819に含まれる一意のコンピューティングインスタンス844をセットアップすることができる。各コンピューティングインスタンス844は、サービステナント819に含まれる制御プレーンVCN816と、顧客テナント821に含まれるデータプレーンVCN818との間の通信を可能にしてもよい。コンピューティングインスタンス844は、サービステナント819に含まれる制御プレーンVCN816内にプロビジョニングされるリソースが、顧客テナント821に含まれるデータプレーンVCN818内に展開されるか、そうでなければ使用されることを可能にし得る。

30

【0096】

他の例では、IaaSプロバイダの顧客は、顧客テナント821内に存在するデータベースを有し得る。この例では、制御プレーンVCN816は、アプリサブネット（複数可）826を含むことができるデータプレーンミラーアプリ層840を含むことができる。データプレーンミラーアプリ層840はデータプレーンVCN818内に存在することができるが、データプレーンミラーアプリ層840はデータプレーンVCN818内に存在しなくてもよい。つまり、データプレーンミラーアプリ層840は、顧客テナント821にアクセスできるが、データプレーンミラーアプリ層840は、データプレーンVCN818に存在しなくてもよいし、IaaSプロバイダの顧客によって所有または動作されてもよい。データプレーンミラーアプリ層840は、データプレーンVCN818への呼び出しを行うように構成されてもよいが、制御プレーンVCN816に含まれる任意のエントティへの呼び出しを行うように構成されなくてもよい。顧客は、制御プレーンVCN816内にプロビジョニングされるデータプレーンVCN818内のリソースを展開またはそうでなければ使用することを望む場合があり、データプレーンミラーアプリ層840は、顧客の所望の展開またはリソースの他の使用を容易にすることができる。

40

【0097】

50

いくつかの実施形態では、IaaSプロバイダの顧客は、データプレーンVCN818にフィルタを適用することができる。この実施形態では、顧客はデータプレーンVCN818が何にアクセスできるかを決定することができ、顧客はデータプレーンVCN818からパブリックインターネット854へのアクセスを制限することができる。IaaSプロバイダは、フィルタを適用したり、外部ネットワークまたはデータベースへのデータプレーンVCN818のアクセスを制御したりできない場合がある。顧客によるフィルタおよび制御を顧客テナント821に含まれるデータプレーンVCN818に適用すると、データプレーンVCN818を他の顧客およびパブリックインターネット854から分離するのに役立ち得る。

【0098】

いくつかの実施形態では、クラウドサービス856は、パブリックインターネット854上、制御プレーンVCN816上、またはデータプレーンVCN818上に存在しない可能性があるサービスにアクセスするために、サービスゲートウェイ836によって呼び出すことができる。クラウドサービス856と制御プレーンVCN816またはデータプレーンVCN818との間の接続は、ライブまたは継続的ではない可能性がある。クラウドサービス856は、IaaSプロバイダが所有または動作する別のネットワーク上に存在し得る。クラウドサービス856は、サービスゲートウェイ836から呼び出しを受信するように構成されてもよいし、パブリックインターネット854から呼び出しを受信しないように構成されてもよい。一部のクラウドサービス856は、他のクラウドサービス856から分離されてもよく、制御プレーンVCN816は、制御プレーンVCN816と同じ領域になくてもよいクラウドサービス856から分離されてもよい。例えば、制御プレーンVCN816は「領域1」に配置され、クラウドサービス「展開7」は領域1と「領域2」に配置され得る。展開7への呼び出しが、領域1にある制御プレーンVCN816に含まれるサービスゲートウェイ836によって行われた場合、その呼び出しは領域1の展開7に送信され得る。この例では、制御プレーンVCN816、または領域1の展開7は、領域2の展開7に通信可能に結合されていない、または通信していない可能性がある。

【0099】

図9は、少なくとも1つの実施形態による、IaaSアーキテクチャの別のパターン例を示すブロック図900である。サービスオペレータ902（例えば、図7のサービスオペレータ702）は、セキュアホストテナント904（例えば、図7のセキュアホストテナント704）に通信可能に結合することができ、これは、仮想クラウドネットワーク（VCN）906（例えば、図7のVCN706）およびセキュアホストサブネット908（例えば、図7のセキュアホストサブネット708）を含むことができる。VCN906は、SSHVCN912（例えば、図7のSSHVCN712）に含まれるLPG910を介してSSHVCN912に通信可能に結合され得るLPG910（例えば、図7のLPG710）を含むことができる。SSHVCN912は、SSHサブネット914（例えば、図7のSSHサブネット714）を含むことができ、SSHVCN912は、制御プレーンVCN916（例えば、図7の制御プレーンVCN716）に含まれるLPG910を介して制御プレーンVCN916に通信可能に結合することができ、データプレーンVCN918（例えば、図7のデータプレーン718）に含まれるLPG910を介してデータプレーンVCN918に通信可能に結合することができる。制御プレーンVCN916およびデータプレーンVCN918は、サービステナント919（例えば、図7のサービステナント719）に含めることができる。

【0100】

制御プレーンVCN916は、ロードバランサ（LB）サブネット（複数可）922（例えば、図7のLBサブネット（複数可）722）を含むことができる制御プレーンDMZ層920（例えば、図7の制御プレーンDMZ層720）、アプリサブネット（複数可）926（例えば、図7のアプリサブネット（複数可）726と同様）を含むことができる制御プレーンアプリ層924（例えば、図7の制御プレーンアプリ層724）、DBサ

10

20

30

40

50

ブネット（複数可）930を含むことができる制御プレーンデータ層928（例えば、図7の制御プレーンデータ層728）を含むことができる。制御プレーンDMZ層920に含まれるLBサブネット（複数可）922は、制御プレーンアプリ層924に含まれるアプリサブネット（複数可）926と、制御プレーンVCN916に含まれ得るインターネットゲートウェイ934（例えば、図7のインターネットゲートウェイ734）とに通信可能に結合され得、アプリサブネット（複数可）926は、制御プレーンデータ層928に含まれるDBサブネット（複数可）930、サービスゲートウェイ936（例えば、図7のサービスゲートウェイ）およびネットワークアドレス変換（NAT）ゲートウェイ938（例えば、図7のNATゲートウェイ738）に通信可能に結合することができる。制御プレーンVCN916は、サービスゲートウェイ936およびNATゲートウェイ938を含むことができる。

10

【0101】

データプレーンVCN918は、データプレーンアプリ層946（例えば、図7のデータプレーンアプリ層746）、データプレーンDMZ層948（例えば、図7のデータプレーンDMZ層748）、およびデータプレーンデータ層950（例えば、図7のデータプレーンデータ層750）を含むことができる。データプレーンDMZ層948は、LBサブネット（複数可）922を含むことができ、これは、データプレーンアプリ層946の信頼できるアプリサブネット（複数可）960と信頼できないアプリサブネット（複数可）962、およびデータプレーンVCN918に含まれるインターネットゲートウェイ934に通信可能に結合することができる。信頼できるアプリサブネット（複数可）960は、データプレーンVCN918に含まれるサービスゲートウェイ936、データプレーンVCN918に含まれるNATゲートウェイ938、およびデータプレーンデータ層950に含まれるDBサブネット（複数可）930に通信可能に結合することができる。信頼できないアプリサブネット（複数可）962は、データプレーンVCN918に含まれるサービスゲートウェイ936およびデータプレーンデータ層950に含まれるDBサブネット（複数可）930に通信可能に結合することができる。データプレーンデータ層950は、データプレーンVCN918に含まれるサービスゲートウェイ936に通信可能に結合できるDBサブネット（複数可）930を含むことができる。

20

【0102】

信頼できないアプリサブネット（複数可）962は、テナント仮想マシン（VM（複数可））966（1）～（N）に通信可能に結合することができる1つまたは複数の1次VNIC（複数可）964（1）～（N）を含むことができる。各テナントVM966（1）～（N）は、それぞれのアプリサブネット967（1）～（N）に通信可能に結合することができる、これは、それぞれの顧客テナント970（1）～（N）に含めることができるそれぞれのコンテナ出口VCN（複数可）968（1）～（N）に含めることができる。それぞれの2次VNIC（複数可）972（1）～（N）は、データプレーンVCN918に含まれる信頼できないアプリサブネット（複数可）962と、コンテナ出口VCN（複数可）968（1）～（N）に含まれるアプリサブネットとの間の通信を容易にすることができる。各コンテナ出口VCN（複数可）968（1）～（N）は、パブリックインターネット954（例えば、図7のパブリックインターネット754）に通信可能に結合できるNATゲートウェイ938を含むことができる。

30

40

【0103】

制御プレーンVCN916に含まれ、データプレーンVCN918に含まれるインターネットゲートウェイ934は、パブリックインターネット954に通信可能に結合できるメタデータ管理サービス952（例えば、図7のメタデータ管理システム752）に通信可能に結合することができる。パブリックインターネット954は、制御プレーンVCN916に含まれ、データプレーンVCN918に含まれるNATゲートウェイ938に通信可能に結合することができる。制御プレーンVCN916に含まれ、データプレーンVCN918に含まれるサービスゲートウェイ936は、クラウドサービス956に通信可能に結合することができる。

50

【0104】

いくつかの実施形態では、データプレーンVCN918は、顧客テナント970と統合することができる。この統合は、コード実行時のサポートが必要な場合など、IaaSプロバイダの顧客にとって有益または望ましい場合がある。顧客は、破壊的な可能性があるコード、他の顧客リソースと通信する可能性のあるコード、またはその他の望ましくない影響を引き起こす可能性のあるコードの実行を提供し得る。これに応じて、IaaSプロバイダは、顧客からIaaSプロバイダに提供されるコードを実行するかどうかを決定できる。

【0105】

いくつかの例では、IaaSプロバイダの顧客は、IaaSプロバイダに一時的なネットワークアクセスを許可し、データプレーン層アプリ946に付加される機能を要求することができる。機能を実行するコードは、VM（複数可）966（1）～（N）で実行することができる。コードはデータプレーンVCN918上の他の場所で実行するように構成することはできない。各VM（複数可）966（1）～（N）は、1つの顧客テナント970に接続できる。VM966（1）～（N）に含まれるそれぞれのコンテナ971（1）～（N）は、コードを実行するように構成され得る。この場合、二重隔離が存在する可能性があり（例えば、コンテナ971（1）～（N）のコード実行、コンテナ971（1）～（N）は、信頼できないアプリサブネット（複数可）962に含まれる少なくともVM966（1）～（N）に含まれ得る）、これは、間違ったコードや望ましくないコードがIaaSプロバイダのネットワークに損害を与えたり、別の顧客のネットワークに損害を与えたりすることを防ぐのに役立ち得る。コンテナ971（1）～（N）は、顧客テナント970に通信可能に結合されてもよく、顧客テナント970からデータを送信または受信するように構成されてもよい。コンテナ971（1）～（N）は、データプレーンVCN918内の任意の他のエンティティからデータを送信または受信するように構成されていない可能性がある。コードの実行が完了すると、IaaSプロバイダはコンテナ971（1）～（N）を強制終了するか、その他の方法で破棄することができる。

【0106】

いくつかの実施形態では、信頼できるアプリサブネット（複数可）960は、IaaSプロバイダによって所有または動作され得るコードを実行し得る。この実施形態では、信頼できるアプリサブネット（複数可）960は、DBサブネット（複数可）930に通信可能に結合され、DBサブネット（複数可）930内でCRUD動作を実行するように構成され得る。信頼できないアプリサブネット（複数可）962は、DBサブネット（複数可）930に通信可能に結合され得るが、この実施形態では、信頼できないアプリサブネット（複数可）は、DBサブネット（複数可）930において読み取り動作を実行するように構成され得る。各顧客のVM966（1）～（N）に含めることができ、顧客からのコードを実行することができるコンテナ971（1）～（N）は、DBサブネット（複数可）930と通信可能に結合されていなくてもよい。

【0107】

他の実施形態では、制御プレーンVCN916とデータプレーンVCN918は、直接通信可能に結合されていなくてもよい。この実施形態では、制御プレーンVCN916とデータプレーンVCN918との間に直接通信が存在しなくてもよい。しかし、通信は少なくとも1つの方法を通じて間接的に行うことができる。LPG910は、制御プレーンVCN916とデータプレーンVCN918との間の通信を容易にすることができるIaaSプロバイダによって確立され得る。別の例では、制御プレーンVCN916またはデータプレーンVCN918は、サービスゲートウェイ936を介してクラウドサービス956への呼び出しを行うことができる。例えば、制御プレーンVCN916からクラウドサービス956への呼び出しは、データプレーンVCN918と通信できるサービスに対する要求を含むことができる。

【0108】

図10は、少なくとも1つの実施形態による、IaaSアーキテクチャの別のパターン

10

20

30

40

50

例を示すブロック図 1000 である。サービスオペレータ 1002（例えば、図 7 のサービスオペレータ 702）は、セキュアホストテナント 1004（例えば、図 7 のセキュアホストテナント 704）に通信可能に結合することができ、これは、仮想クラウドネットワーク（VCN）1006（例えば、図 7 の VCN 706）およびセキュアホストサブネット 1008（例えば、図 7 のセキュアホストサブネット 708）を含むことができる。VCN 1006 は、SSHVCN 1012 に含まれる LPG 1010 を介して SSHVCN 1012（例えば、図 7 の SSHVCN 712）に通信可能に結合され得る LPG 1010（例えば、図 7 の LPG 710）を含むことができる。SSHVCN 1012 は、SSH サブネット 1014（例えば、図 7 の SSH サブネット 714）を含むことができ、SSHVCN 1012 は、制御プレーン VCN 1016（例えば、図 7 の制御プレーン VCN 716）に含まれる LPG 1010 を介して、制御プレーン VCN 1016 に通信可能に結合することができ、データプレーン VCN 1018（例えば、図 7 のデータプレーン 718）に含まれる LPG 1010 を介して、データプレーン VCN 1018 に通信可能に結合することができる。制御プレーン VCN 1016 およびデータプレーン VCN 1018 は、サービステナント 1019（例えば、図 7 のサービステナント 719）に含めることができる。

10

20

30

40

50

【0109】

制御プレーン VCN 1016 は、LB サブネット（複数可）1022（例えば、図 7 の LB サブネット（複数可）722）を含むことができる制御プレーン DMZ 層 1020（例えば、図 7 の制御プレーン DMZ 層 720）、アプリサブネット（複数可）1026（例えば、図 7 のアプリサブネット（複数可）726）を含むことができる制御プレーン アプリ層 1024（例えば、図 7 の制御プレーン アプリ層 724）、DB サブネット（複数可）1030（例えば、図 9 の DB サブネット（複数可）930）を含むことができる制御プレーン データ層 1028（例えば、図 7 の制御プレーン データ層 728）を含むことができる。制御プレーン DMZ 層 1020 に含まれる LB サブネット（複数可）1022 は、制御プレーン アプリ層 1024 に含まれるアプリサブネット（複数可）1026 に通信可能に結合することができ、制御プレーン VCN 1016 に含めることができるインターネットゲートウェイ 1034（例えば、図 7 のインターネットゲートウェイ 734）に通信可能に結合することができ、アプリサブネット（複数可）1026 は、制御プレーン データ層 1028 に含まれる DB サブネット（複数可）1030 に通信可能に結合することができ、サービスゲートウェイ 1036（例えば、図 7 のサービスゲートウェイ）およびネットワークアドレス変換（NAT）ゲートウェイ 1038（例えば、図 7 の NAT ゲートウェイ 738）に通信可能に結合することができる。制御プレーン VCN 1016 は、サービスゲートウェイ 1036 および NAT ゲートウェイ 1038 を含むことができる。

【0110】

データプレーン VCN 1018 は、データプレーン アプリ層 1046（例えば、図 7 のデータプレーン アプリ層 746）、データプレーン DMZ 層 1048（例えば、図 7 のデータプレーン DMZ 層 748）、およびデータプレーン データ層 1050（例えば、図 7 のデータプレーン データ層 750）を含むことができる。データプレーン DMZ 層 1048 は、LB サブネット（複数可）1022 を含むことができ、それは、信頼できるアプリサブネット（複数可）1060（例えば、図 9 の信頼できるアプリサブネット（複数可）960）に通信可能に結合され得、データプレーン アプリ層 1046 の信頼できないアプリサブネット（複数可）1062（例えば、図 9 の信頼できないアプリサブネット（複数可）962）およびデータプレーン VCN 1018 に含まれるインターネットゲートウェイ 1034 に通信可能に結合することができる。信頼できるアプリサブネット 1060（複数可）は、データプレーン VCN 1018 に含まれるサービスゲートウェイ 1036 に通信可能に結合することができ、データプレーン VCN 1018 に含まれる NAT ゲートウェイ 1038、およびデータプレーン データ層 1050 に含まれる DB サブネット（複数可）1030 に通信可能に結合することができる。信頼できないアプリサブネット（複

数可) 1062は、データプレーンVCN1018に含まれるサービスゲートウェイ1036およびデータプレーンデータ層1050に含まれるDBサブネット(複数可)1030に通信可能に接続することができる。データプレーンデータ層1050は、データプレーンVCN1018に含まれるサービスゲートウェイ1036に通信可能に結合できるDBサブネット(複数可)1030を含むことができる。

【0111】

信頼できないアプリサブネット(複数可)1062は、1次VNIC(複数可)1064(1)~(N)を含むことができ、これは、信頼できないアプリサブネット(複数可)1062内に存在するテナント仮想マシン(VM(複数可))1066(1)~(N)に通信可能に結合することができる。各テナントVM1066(1)~(N)は、それぞれのコンテナ1067(1)~(N)内でコードを実行することができ、コンテナ出口VCN1068に含めることができるデータプレーンアプリ層1046に含めることができるアプリサブネット1026に通信可能に結合することができる。それぞれの2次VNIC(複数可)1072(1)~(N)は、データプレーンVCN1018に含まれる信頼できないアプリサブネット(複数可)1062とコンテナ出口VCN1068に含まれるアプリサブネットとの間の通信を容易にすることができる。コンテナ出口VCNは、パブリックインターネット1054(例えば、図7のパブリックインターネット754)に通信可能に結合できるNATゲートウェイ1038を含むことができる。

【0112】

制御プレーンVCN1016に含まれ、データプレーンVCN1018に含まれるインターネットゲートウェイ1034は、パブリックインターネット1054に通信可能に結合できるメタデータ管理サービス1052(例えば、図7のメタデータ管理システム752)に通信可能に結合することができる。パブリックインターネット1054は、制御プレーンVCN1016に含まれ、データプレーンVCN1018に含まれるNATゲートウェイ1038に通信可能に結合することができる。制御プレーンVCN1016に含まれ、データプレーンVCN1018に含まれるサービスゲートウェイ1036は、クラウドサービス1056に通信可能に結合することができる。

【0113】

いくつかの例では、図10のブロック図1000のアーキテクチャによって示されるパターンは、図9のブロック図900のアーキテクチャによって示されるパターンの例外であると考えられ、IaaSプロバイダが顧客と直接通信できない場合(例えば、切断されている領域)、IaaSプロバイダの顧客にとっては望ましい場合がある。各顧客のVM(複数可)1066(1)~(N)に含まれるそれぞれのコンテナ1067(1)~(N)は、顧客によってリアルタイムでアクセス可能である。コンテナ1067(1)~(N)は、コンテナ出口VCN1068に含めることができるデータプレーンアプリ層1046のアプリサブネット(複数可)1026に含まれるそれぞれの2次VNIC(複数可)1072(1)~(N)への呼び出しを行うように構成され得る。2次VNIC(複数可)1072(1)~(N)は、呼び出しをパブリックインターネット1054に送信することができるNATゲートウェイ1038に呼び出しを送信することができる。この例では、顧客がリアルタイムでアクセスできるコンテナ1067(1)~(N)は、制御プレーンVCN1016から分離することができる、データプレーンVCN1018に含まれる他のエンティティから分離することができる。コンテナ1067(1)~(N)は、他の顧客からのリソースから隔離されてもよい。

【0114】

他の例では、顧客は、コンテナ1067(1)~(N)を使用して、クラウドサービス1056を呼び出すことができる。この例では、顧客は、クラウドサービス1056からサービスを要求するコードをコンテナ1067(1)~(N)内で実行することができる。コンテナ1067(1)~(N)は、この要求を2次VNIC1072(1)~(N)に送信することができ、2次VNIC(複数可)1072(1)~(N)は、要求をパブリックインターネット1054に送信することができるNATゲートウェイに要求を送信

することができる。パブリックインターネット1054は、インターネットゲートウェイ1034を介して、制御プレーンVCN1016に含まれるLBサブネット（複数可）1022に要求を送信することができる。要求が有効であるとの決定に応答して、LBサブネット（複数可）は、サービスゲートウェイ1036を介してクラウドサービス1056に要求を送信できるアプリサブネット（複数可）1026に要求を送信することができる。

【0115】

図に示されているIaaSアーキテクチャ700、800、900、1000は、示されているものの以外の構成要素を有し得ることを理解されたい。さらに、図に示される実施形態は、本開示の実施形態を組み込むことができるクラウドインフラストラクチャシステムのいくつかの例にすぎない。他のいくつかの実施形態では、IaaSシステムは、図示されているよりも多いもしくは少ない構成要素を有してもよく、2つ以上の構成要素を組み合わせてもよく、または構成要素の異なる構成もしくは配置を有してもよい。

10

【0116】

特定の実施形態では、本明細書で説明されるIaaSシステムは、セルフサービス、サブスクリプションベース、弾力的に拡張可能、信頼性が高く、可用性が高く、安全な方法で顧客に提供されるアプリケーション、ミドルウェア、およびデータベースサービス製品のスイートを含むことができる。このようなIaaSシステムの一例として、本譲受人が提供するOracle Cloud Infrastructure（OCI）がある。

【0117】

図11は、さまざまな実施形態が実装され得る例示的なコンピュータシステム1100を示す。システム1100は、上述のコンピュータシステムのいずれかを実装するために使用することができる。図に示すように、コンピュータシステム1100は、バスサブシステム1102を介して多くの周辺サブシステムと通信する処理装置1104を含む。これらの周辺サブシステムは、処理加速装置1106、I/Oサブシステム1108、記憶サブシステム1118、および通信サブシステム1124を含み得る。記憶サブシステム1118は、有形のコンピュータ可読記憶媒体1122およびシステムメモリ1110を含む。

20

【0118】

バスサブシステム1102は、コンピュータシステム1100のさまざまな構成要素およびサブシステムが意図したとおりに相互に通信できるようにする機構を提供する。バスサブシステム1102は単一のバスとして概略的に示されているが、バスサブシステムの代替実施形態では複数のバスを利用することができる。バスサブシステム1102は、さまざまなバスアーキテクチャのいずれかを使用するメモリバスまたはメモリコントローラ、周辺バス、およびローカルバスを含むいくつかのタイプのバス構造のいずれであってもよい。例えば、そのようなアーキテクチャには、Industry Standard Architecture（ISA）バス、Micro Channel Architecture（MCA）バス、Enhanced ISA（EISA）バス、Video Electronics Standards Association（VESA）ローカルバス、およびPeripheral Component Interconnect（PCI）バスが含まれ得る。これは、IEEE P1386.1標準に従って製造されるメザンバスとして実装できる。

30

【0119】

処理装置1104は、1つまたは複数の集積回路（例えば、従来のマイクロプロセッサまたはマイクロコントローラ）として実装することができ、コンピュータシステム1100の動作を制御する。1つまたは複数のプロセッサが処理装置1104に含まれてもよい。これらのプロセッサには、単一コアプロセッサまたはマルチコアプロセッサが含まれ得る。特定の実施形態では、処理装置1104は、各処理装置に含まれる単一コアプロセッサまたはマルチコアプロセッサを有する1つまたは複数の独立した処理装置1132および／または1134として実装され得る。他の実施形態では、処理装置1104は、2つ

40

50

のデュアルコアプロセッサを単一のチップに統合することによって形成されるクアッドコア処理装置として実装されてもよい。

【0120】

さまざまな実施形態において、処理装置1104は、プログラムコードに応答してさまざまなプログラムを実行することができ、複数の同時実行プログラムまたはプロセスを維持することができる。いつでも、実行されるプログラムコードの一部またはすべては、プロセッサ（複数可）1104および／または記憶サブシステム1118に存在することができる。適切なプログラミングを通じて、プロセッサ（複数可）1104は、上述のさまざまな機能を提供することができる。コンピュータシステム1100は、デジタルシグナルプロセッサ（DSP）、専用プロセッサ等を含むことができる処理加速装置1106をさらに含むことができる。

10

【0121】

I/Oサブシステム1108は、ユーザインターフェイス入力装置およびユーザインターフェイス出力装置を含むことができる。ユーザインターフェイス入力装置にはキーボード、マウスやトラックボールなどのポインティング装置、ディスプレイに組み込まれたタッチパッドやタッチスクリーン、スクロールホイール、クリックホイール、ダイヤル、ボタン、スイッチ、キーパッド、音声コマンド認識システムを備えたオーディオ入力装置、マイク、および他のタイプの入力装置が含まれ得る。ユーザインターフェイス入力装置には、例えば、Microsoft Kinect（登録商標）モーションセンサーなどのモーションセンシングおよび／またはジェスチャ認識装置が含まれ得、ユーザは、ジェスチャや音声コマンドを使用した自然なユーザインターフェイスを通じて、Microsoft Xbox（登録商標）360ゲームコントローラーなどの入力装置を制御し、対話できるようになる。ユーザインターフェイス入力装置には、ユーザからの目の活動（例えば、写真撮影中および／またはメニュー選択中の「瞬き」）を検出し、目のジェスチャを入力装置（例えば、Google Glass（登録商標））への入力として変換するGoogle Glass（登録商標）瞬き検出器などのアイジェスチャ認識装置も含まれ得る。さらに、ユーザインターフェイス入力装置は、ユーザが音声コマンドを通じて音声認識システム（例えば、Siri（登録商標）ナビゲータ）と対話できるようにする音声認識センシング装置を含んでもよい。

20

【0122】

ユーザインターフェイス入力装置には、限定されないが、3次元（3D）マウス、ジョイスティックまたはポインティングスティック、ゲームパッドおよびグラフィックタブレット、ならびにスピーカ、デジタルカメラ、デジタルビデオカメラなどのオーディオ／ビジュアル装置、ポータブルメディアプレーヤ、ウェブカメラ、イメージスキャナ、指紋スキャナ、バーコードリーダ3Dスキャナ、3Dプリンタ、レーザ距離計、および視線追跡装置なども含まれ得る。さらに、ユーザインターフェイス入力装置には、例えば、コンピュータ断層撮影法、磁気共鳴画像法、位置放射断層撮影法、医療用超音波検査装置などの医療用画像入力装置が含まれてもよい。ユーザインターフェイス入力装置には、例えば、MIDIキーボード、デジタル楽器などのオーディオ入力装置も含まれ得る。

30

【0123】

ユーザインターフェイス出力装置には、ディスプレイサブシステム、インジケータライト、またはオーディオ出力装置などの非視覚的ディスプレイが含まれてもよい。ディスプレイサブシステムは、陰極線管（CRT）、液晶ディスプレイ（LCD）またはプラズマディスプレイを使用するものなどのフラットパネル装置、投影装置、タッチスクリーンなどであってもよい。一般に、「出力装置」という用語の使用は、コンピュータシステム1100からユーザまたは他のコンピュータに情報を出力するためのあらゆる可能なタイプの装置および機構を含むことを意図している。例えば、ユーザインターフェイス出力装置には、モニタ、プリンタ、スピーカ、ヘッドフォン、自動車ナビゲーションシステム、プロッタ、音声出力装置、およびモデムなど、テキスト、グラフィックス、およびオーディオ／ビデオ情報を視覚的に伝達するさまざまな表示装置が含まれるが、これらに限定されない。

40

50

【0124】

コンピュータシステム1100は、システムメモリ1110内に現在配置されているように示されるソフトウェア要素を備える記憶サブシステム1118を備えることができる。システムメモリ1110は、処理装置1104上でロード可能および実行可能なプログラム命令、ならびにこれらのプログラムの実行中に生成されるデータを記憶することができる。

【0125】

コンピュータシステム1100の構成および種類に応じて、システムメモリ1110は、揮発性（ランダムアクセスメモリ（RAM）など）および／または不揮発性（読み取り専用メモリ（ROM）、フラッシュメモリなど）の場合がある。RAMは通常、処理装置1104に即座にアクセス可能な、ならびに／または処理装置1104によって現在動作および実行されているデータおよび／またはプログラムモジュールを含む。いくつかの実装形態では、システムメモリ1110は、スタティックランダムアクセスメモリ（SRAM）またはダイナミックランダムアクセスメモリ（DRAM）などの複数の異なるタイプのメモリを含み得る。いくつかの実装形態では、起動中など、コンピュータシステム1100内の要素間で情報を転送するのに役立つ基本ルーチンを含む基本入出力システム（BIOS）は、通常、ROMに記憶され得る。限定ではなく一例として、システムメモリ1110は、クライアントアプリケーション、ウェブブラウザ、中間層アプリケーション、リレーショナルデータベース管理システム（RDBMS）などを含み得るアプリケーションプログラム1112、プログラムデータ1114、およびオペレーティングシステム1116も示す。一例として、オペレーティングシステム1116には、さまざまなバージョンのMicrosoft Windows（登録商標）、Apple Macintosh（登録商標）、および／またはLinuxオペレーティングシステム、さまざまな市販のUNIX（登録商標）もしくはUNIX系オペレーティングシステム（さまざまなGNU/Linuxオペレーティングシステム、Google Chrome（登録商標）OSなどを含むがこれらに限定されない）、ならびに／またはiOS、Windows（登録商標）Phone、Android（登録商標）OS、BlackBerry（登録商標）OS、およびPalm（登録商標）OSオペレーティングシステムなどのモバイルオペレーティングシステムが含まれ得る。

【0126】

記憶サブシステム1118はまた、いくつかの実装形態の機能を提供する基本的なプログラミングおよびデータ構造を記憶するための有形のコンピュータ可読記憶媒体を提供することもできる。プロセッサによって実行されると、上述の機能を提供するソフトウェア（プログラム、コードモジュール、命令）は、記憶サブシステム1118に記憶され得る。これらのソフトウェアモジュールまたは命令は、処理装置1104によって実行され得る。記憶サブシステム1118はまた、本開示に従って使用されるデータを記憶するためのリポジトリを提供することもできる。

【0127】

記憶サブシステム1100はまた、コンピュータ可読記憶媒体1122にさらに接続できるコンピュータ可読記憶媒体リーダ1120を含んでもよい。一緒に、そして任意でシステムメモリ1110と組み合わせて、コンピュータ可読記憶媒体1122は、リモート、ローカル、固定、および／または取り外し可能な記憶装置に加えて、コンピュータ可読情報を一時的および／またはより永続的に収容、記憶、送信、および取得するための記憶媒体を包括的に表すことができる。

【0128】

コードまたはコードの一部を含むコンピュータ可読記憶媒体1122には、当技術分野で既知または使用されている任意の適切な媒体を含めることもでき、これには、情報の記憶および／または送信のための任意の方法または技術で実装される揮発性および不揮発性、取り外し可能なおよび取り外し不可能な媒体などの記憶媒体および通信媒体が含まれるが、これらに限定されない。これには、RAM、ROM、電子的に消去可能なプログラマブルROM（EEPROM）、フラッシュメモリもしくはその他のメモリテクノロジー、C

D-R O M、デジタルバーサタイルディスク（D V D）などの有形のコンピュータ可読記憶媒体、または他の光学記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置もしくは他の磁気記憶装置、または他の有形のコンピュータ可読媒体が含まれ得る。これには、データ信号、データ送信、または所望の情報を送信するために使用でき、コンピューティングシステム 1 1 0 0 によってアクセスできる任意の他の媒体などの、無形のコンピュータ可読媒体も含まれ得る。

【0 1 2 9】

一例として、コンピュータ可読記憶媒体 1 1 2 2 は、取り外し不可能な不揮発性磁気媒体に対して読み取りまたは書き込みを行うハードディスクドライブ、取り外し可能な不揮発性磁気ディスクに対して読み取りまたは書き込みを行う磁気ディスクドライブ、および C D R O M、D V D、B l u e R a y（登録商標）ディスクなどの取り外し可能な不揮発性光ディスク、またはその他の光媒体に対して読み取りまたは書き込みを行う光ディスクドライブを含むことができる。コンピュータ可読記憶媒体 1 1 2 2 には、Z i p（登録商標）ドライブ、フラッシュメモリカード、ユニバーサルシリアルバス（U S B）フラッシュドライブ、セキュアデジタル（S D）カード、D V D ディスク、デジタルビデオテープなどが含まれてもよいが、これらに限定されない。コンピュータ可読記憶媒体 1 1 2 2 には、フラッシュメモリベースの S S D などの不揮発性メモリに基づくソリッドステートドライブ（S S D）、エンタープライズフラッシュドライブ、ソリッドステート R O M など、ソリッドステート R A M、ダイナミック R A M、スタティック R A M などの揮発性メモリをベースにした S S D、D R A M ベースの S S D、磁気抵抗 R A M（M R A M）S S D、および D R A M とフラッシュメモリベースの S S D を組み合わせて使用するハイブリッド S S D も含まれ得る。ディスクドライブおよびそれらに関連するコンピュータ可読媒体は、コンピュータシステム 1 1 0 0 のためのコンピュータ可読命令、データ構造、プログラムモジュール、および他のデータの非揮発性記憶を提供し得る。

【0 1 3 0】

通信サブシステム 1 1 2 4 は、他のコンピュータシステムおよびネットワークへのインターフェイスを提供する。通信サブシステム 1 1 2 4 は、コンピュータシステム 1 1 0 0 から他のシステムとの間でデータを送受信するためのインターフェイスとして機能する。例えば、通信サブシステム 1 1 2 4 は、コンピュータシステム 1 1 0 0 がインターネットを介して 1 つまたは複数の装置に接続できるようにすることができる。いくつかの実施形態では、通信サブシステム 1 1 2 4 は、無線音声および／またはデータネットワークにアクセスするための無線周波数（R F）トランシーバ構成要素を含めることができる（例えば、携帯電話技術、3 G、4 G、または E D G E（地球規模の進化のための強化されるデータレート）などの高度なデータネットワーク技術を使用する）、W i F i（I E E E 8 0 2 . 1 1 ファミリ標準、または他のモバイル通信技術、またはそれらの任意の組み合わせ）、全地球測位システム（G P S）受信機構成要素、および／または他の構成要素を含むことができる。いくつかの実施形態では、通信サブシステム 1 1 2 4 は、無線インターフェイスに加えて、またはその代わりに、有線ネットワーク接続（例えば、イーサネット（登録商標））を提供することができる。

【0 1 3 1】

いくつかの実施形態では、通信サブシステム 1 1 2 4 は、コンピュータシステム 1 1 0 0 を使用することができる 1 人または複数のユーザに代わって、構造化および／または非構造化データフィード 1 1 2 6、イベントストリーム 1 1 2 8、イベント更新 1 1 3 0 などの形式で入力通信を受信することもできる。

【0 1 3 2】

一例として、通信サブシステム 1 1 2 4 は、ソーシャルネットワーク、および／または T w i t t e r（登録商標）フィード、F a c e b o o k（登録商標）更新、リッチサイトサマリー（R S S）フィードなどの W e b フィードなどのその他の通信サービス、ならびに／または 1 つもしくは複数のサードパーティ情報ソースからのリアルタイム更新のユーザからリアルタイムでデータフィード 1 1 2 6 を受信するように構成され得る。

【0133】

さらに、通信サブシステム1124は、連続データストリームの形式でデータを受信するように構成されてもよく、これには、リアルタイムイベントおよび／またはイベント更新1130のイベントストリーム1128が含まれ得、これらは、連続的または明示的な終わりのない本質的に無制限であり得る。連続データを生成するアプリケーションの例には、例えば、センサーデータアプリケーション、金融ティッカ、ネットワーク性能測定ツール（例えば、ネットワーク監視およびトラフィック管理アプリケーション）、クリックストリーム分析ツール、自動車交通監視などが含まれ得る。

【0134】

通信サブシステム1124はまた、構造化および／または非構造化データフィールド1126、イベントストリーム1128、イベント更新1130などを、コンピュータシステム1100に結合される1つまたは複数のストリーミングデータソースコンピュータと通信することができる1つまたは複数のデータベースに出力するように構成することもできる。

10

【0135】

コンピュータシステム1100は、ハンドヘルドポータブル装置（例えば、iPhone（登録商標）携帯電話、iPad（登録商標）コンピューティングタブレット、PDA）、ウェアラブル装置（例えば、Google Glass（登録商標）ヘッドマウントディスプレイ）、PC、ワークステーション、メインフレーム、キオスク、サーバラック、またはその他のデータ処理システムを含むさまざまなタイプのうちの1つであり得る。

20

【0136】

コンピュータおよびネットワークの絶え間なく変化する性質のため、図に示されるコンピュータシステム1100の説明は、特定の例としてのみ意図されている。図に示されているシステムよりも多くの、または少ない構成要素を備えた他の多くの構成が可能である。例えば、カスタマイズされているハードウェアも使用されたり、特定の要素がハードウェア、ファームウェア、ソフトウェア（アプレットを含む）、またはその組み合わせで実装されたりし得る。さらに、ネットワーク入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使用されてもよい。本明細書で提供される開示および教示に基づいて、当業者であれば、さまざまな実施形態を実装するための他の手法および／または方法を理解するであろう。

30

【0137】

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、代替構造、および均等物も本開示の範囲内に含まれる。実施形態は、特定の特有のデータ処理環境内での動作に限定されず、複数のデータ処理環境内で自由に動作することができる。さらに、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実施形態を説明したが、本開示の範囲が説明した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である。上述の実施形態のさまざまな特徴および態様は、個別にまたは組み合わせで使用することができる。

【0138】

さらに、ハードウェアとソフトウェアの特定の組み合わせを使用して実施形態を説明したが、ハードウェアとソフトウェアの他の組み合わせも本開示の範囲内であることを認識されたい。実施形態は、ハードウェアのみで、もしくはソフトウェアのみで、またはそれらの組み合わせを使用して実装することができる。本明細書で説明するさまざまなプロセスは、同じプロセッサ上で実装することも、異なるプロセッサを任意に組み合わせで実装することもできる。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説明されている場合、このような構成は、例えば、動作を実行する電子回路を設計することによって、動作を実行するようにプログラマブル電子回路（マイクロプロセッサなど）をプログラムすることによって、またはそれらの任意の組み合わせによって達成することができる。プロセスは、プロセス間通信のための従来の技術を含むがこれに限定されないさまざまな技術を使用して通信することができ、プロセスの異なる対

40

50

が異なる技術を使用したり、同じプロセスの対が異なる時点で異なる技術を使用したりすることができる。

【0139】

したがって、明細書および図面は、限定的な意味ではなく、例示的な意味としてみなされるべきである。しかし、特許請求の範囲に記載のより広い趣旨および範囲から逸脱することなく、追加、減算、削除、ならびにその他の修正および変更を行うことができることは明らかである。したがって、特定の開示実施形態について説明してきたが、これらは限定することを意図したものではない。さまざまな修正および均等物は、以下の特許請求の範囲内に含まれる。

【0140】

開示されている実施形態を説明する文脈における（特に、以下の特許請求の範囲の文脈において）用語「a」、「an」、「the」、および同様の指示対象の使用は、本明細書で別途指示するか、文脈と明らかに矛盾しない限り、単数形と複数形の両方をカバーすると解釈されるべきである。「含む」、「有する」、「含む」、および「含有する」という用語は、特に断りのない限り、無制限の用語（すなわち、「含むがこれらに限定されない」を意味する）として解釈されるべきである。「接続されている」という用語は、たとえば何かが介在している場合でも、部分的または全体的に内部に含まれている、取り付けられている、または結合されていると解釈される。本明細書における値の範囲の記載は、本明細書に別段の記載がない限り、その範囲内にある各々の個別の値を個別に参照する簡略的な方法として機能することを単に意図しており、個別の各値は、あたかも本明細書に個別に記載されているかのように明細書に組み込まれる。本明細書に記載されるすべての方法は、本明細書に別段の指示がない限り、または文脈と明らかに矛盾しない限り、任意の適切な順序で実行することができる。本明細書で提供されるあらゆる例、または例示的な文言（例えば、「など」）の使用は、単に実施形態をより良く説明することを目的としており、別段の請求がない限り、本開示の範囲に制限を課すものではない。本明細書のいかなる文言も、特許請求されていない任意の要素が本開示の実施に必須であることを示すものとして解釈されるべきではない。

【0141】

「X、Y、またはZのうちの少なくとも1つ」という句などの選言的表現は、特に別段の記載がない限り、項目、用語などがX、Y、またはZのいずれか、またはそれらの任意の組み合わせ（例えば、X、Y、および／またはZ）であり得ることを示すために一般に使用されると文脈内で理解されることを意図している。したがって、そのような選言的表現は、一般に、特定の実施形態が、Xの少なくとも1つ、Yの少なくとも1つ、またはZの少なくとも1つがそれぞれ存在することを必要とすることを意味することを意図したものではなく、また、意味するべきではない。

【0142】

本開示の好ましい実施形態が、本開示を実施するための既知の最良のモードを含めて、本明細書に記載される。これらの好ましい実施形態の変形は、前述の説明を読めば当業者には明らかになるであろう。当業者であれば、そのような変形を必要に応じて採用することができるはずであり、本開示は、本明細書に具体的に記載された以外の方法で実施することもできる。したがって、本開示には、適用される法律によって許可される、本明細書に添付の特許請求の範囲に記載されている主題のすべての修正および均等物が含まれる。さらに、本明細書で別段の指示がない限り、可能なすべての変形例における上述の要素の任意の組み合わせが本開示に包含される。

【0143】

本明細書に引用される刊行物、特許出願、および特許を含むすべての参考文献は、各参考文献が参照により組み込まれることが個別かつ具体的に示され、その全体が本明細書に記載されるのと同じ程度に、参照により本明細書に組み込まれる。

【0144】

前述の明細書では、本開示の態様は、その特定の実施形態を参照して説明されているが

、当業者は、本開示がそれに限定されないことを認識するであろう。上述の開示のさまざまな特徴および態様は、個別にまたは組み合わせて使用することができる。さらに、実施形態は、本明細書のより広範な趣旨および範囲から逸脱することなく、本明細書に記載されているものを超える任意の数の環境および用途において利用することができる。したがって、明細書および図面は、限定的なものではなく、例示的なものとみなされるべきである。

【図面】

【図 1】

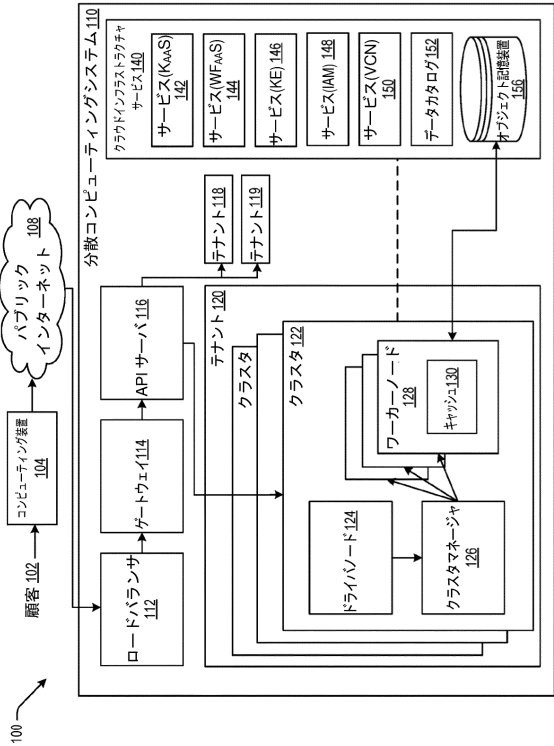


FIG. 1

【図 2】

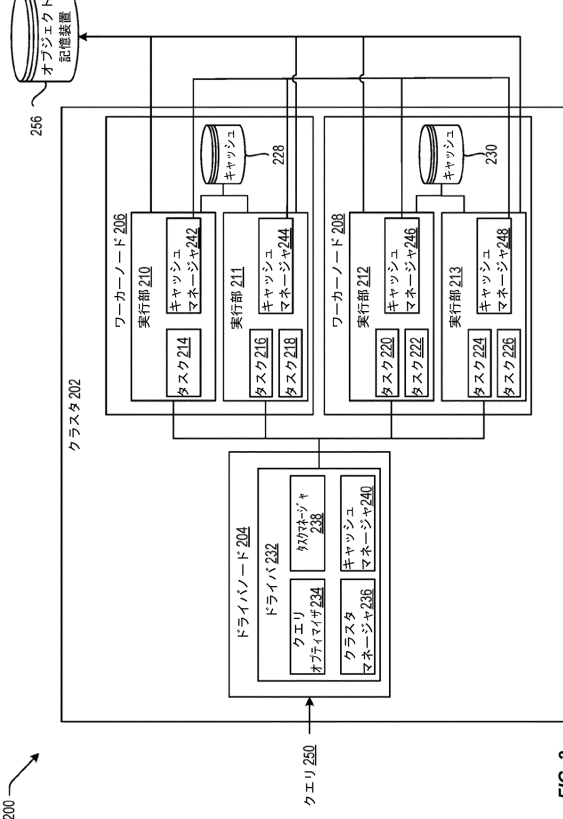


FIG. 2

【図 3 A】

```
300
hostTokenMap:
host_0 -> List(TokenBounds(357913941, 715827882),
               TokenBounds(1789569705, 2147483647)),
host_1 -> List(TokenBounds(0, 357913941),
               TokenBounds(715827882, 1073741823)),
host_2 -> List(TokenBounds(1431655764, 1789569705),
               TokenBounds(1073741823, 1431655764))
```

FIG. 3A

【図 3 B】

```
310
hostTokenMap:
host_0 -> List(TokenBounds(357913941, 715827882),
               TokenBounds(1789569705, 2147483647)),
host_1 -> List(TokenBounds(0, 357913941),
               TokenBounds(715827882, 1073741823)),
host_2 -> List(TokenBounds(1431655764, 1789569705),
               TokenBounds(1073741823, 1431655764))
```

FIG. 3B

【図 4】

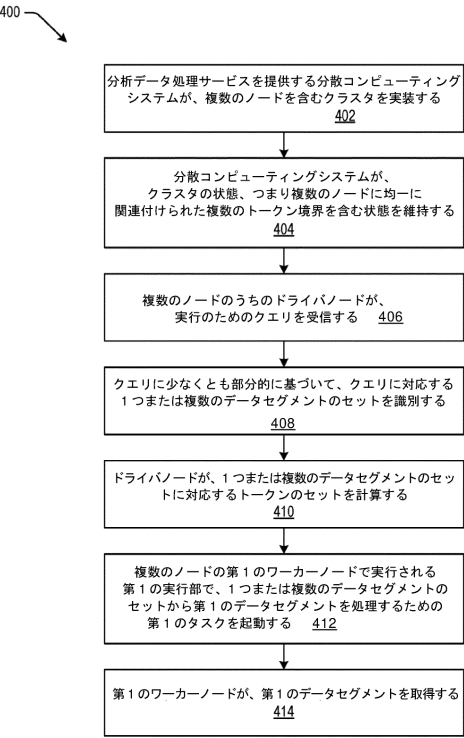


FIG. 4

【図 5】

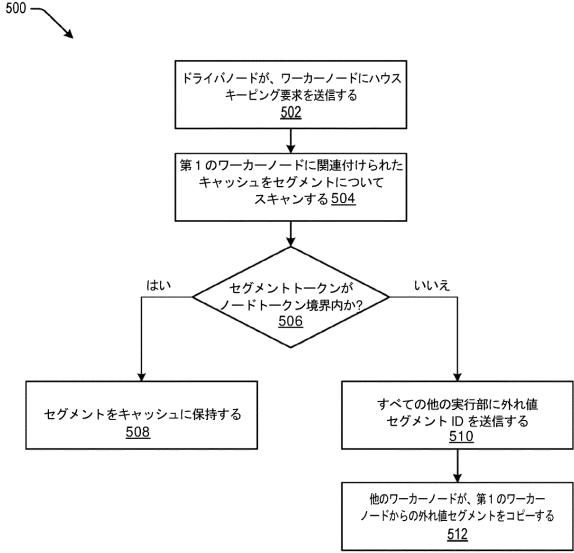


FIG. 5

【図 6】

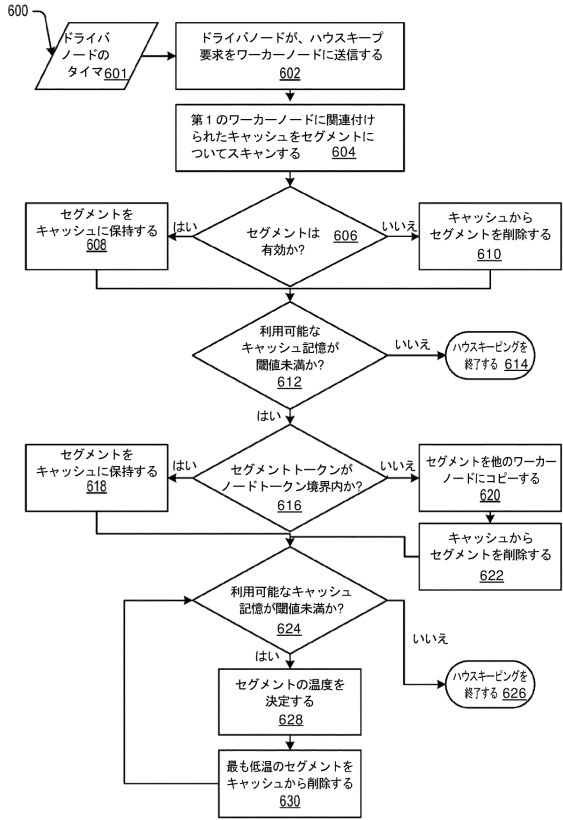


FIG. 6

【図 7】

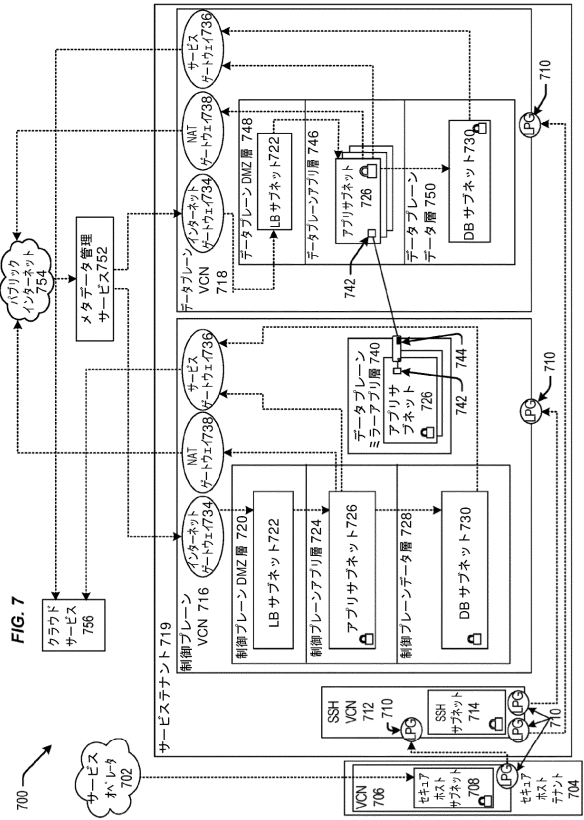


FIG. 7

10

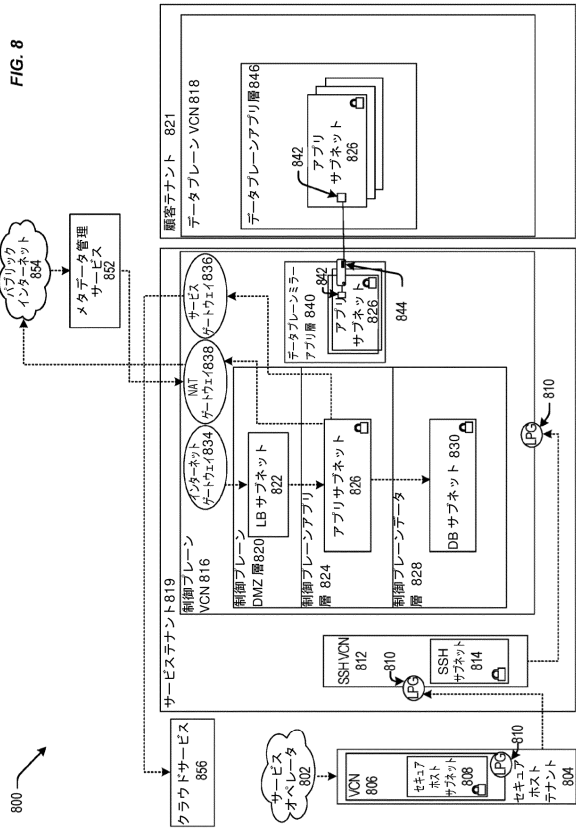
20

30

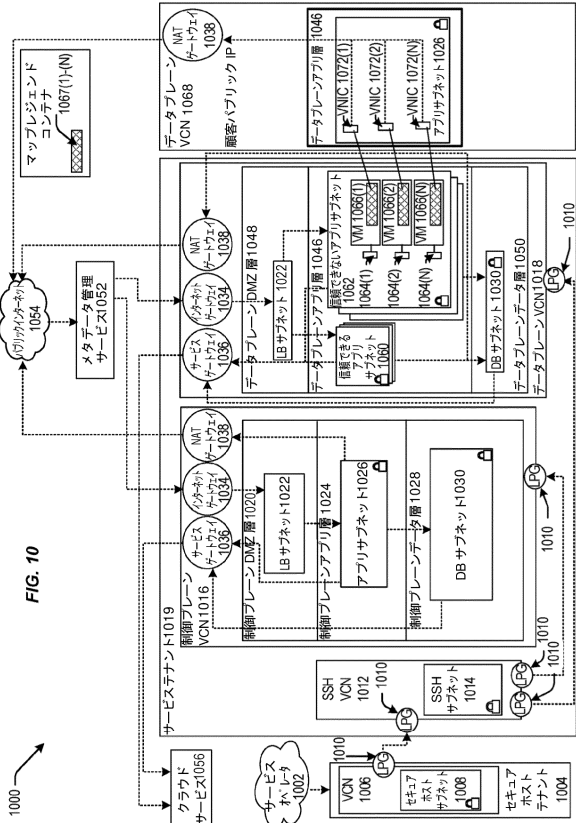
40

50

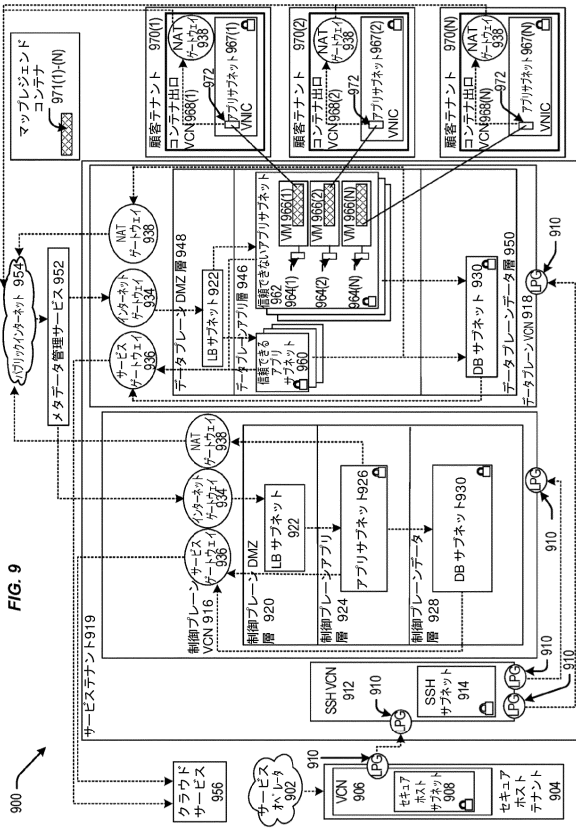
【図 8】



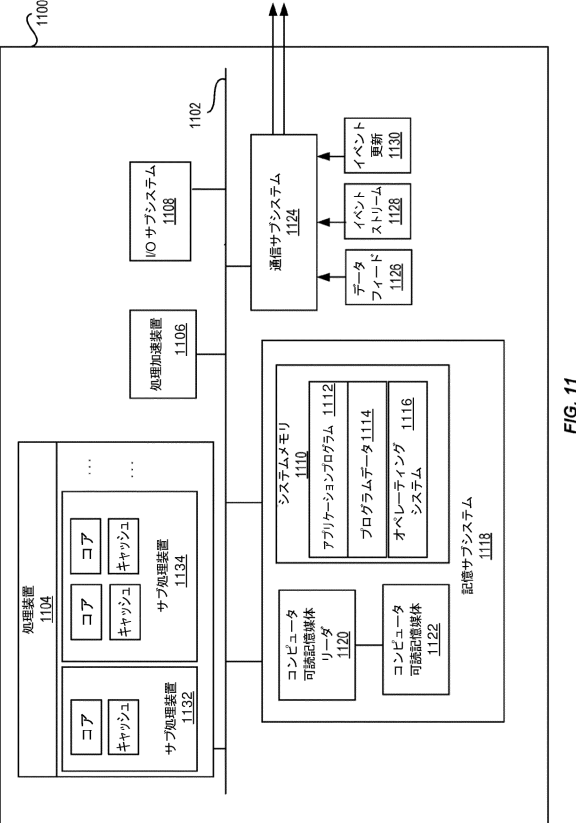
【図 10】



【図 9】



【図 11】



10

20

30

40

50

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2022/030337

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F16/2458 G06F11/30
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)
G06F

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)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2020/065303 A1 (BHATTACHARJEE ARINDAM [US] ET AL) 27 February 2020 (2020-02-27) abstract paragraph [0182] - paragraph [0188] paragraph [0289] - paragraph [0291] paragraph [0340] - paragraph [0346] paragraph [0483] - paragraph [0485] paragraph [0527] - paragraph [0528] paragraph [0494] - paragraph [0501] paragraph [0570] paragraph [0599] - paragraph [0604] paragraph [0662] - paragraph [0663] paragraph [0669] - paragraph [0679] paragraph [0703] - paragraph [0707] paragraph [0732] - paragraph [0756] paragraph [0771] - paragraph [0799] paragraph [0888] - paragraph [0896] paragraph [0903] - paragraph [0904] paragraph [1124] - paragraph [1128] -/--	1-20

☒ Further documents are listed in the continuation of Box C.

☒ 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 citation 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

24 August 2022

Date of mailing of the international search report

05/09/2022

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

Boyadzhiev, Yavor

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2022/030337

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	paragraph [1156] - paragraph [1157] ----- US 2019/102418 A1 (VASUDEVAN SANAL [AU] ET AL) 4 April 2019 (2019-04-04) paragraph [0098] paragraph [0114] paragraph [0187] paragraph [0236] paragraph [0252] -----	1-20
A	YANG WEIQING ET AL: "SHC: Distributed Query Processing for Non-Relational Data Store", 2018 IEEE 34TH INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE), IEEE, 16 April 2018 (2018-04-16), pages 1465-1476, XP033426908, DOI: 10.1109/ICDE.2018.00165 [retrieved on 2018-10-24] abstract page 1471, left-hand column, line 3 - page 1471, right-hand column, line 30 -----	1-20

1

10

20

30

40

50

INTERNATIONAL SEARCH REPORT				International application No PCT/US2022/030337	
Information on patent family members					
Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 2020065303	A1	27-02-2020	NONE		

US 2019102418	A1	04-04-2019	CN 110249321 A		17-09-2019
			EP 3676725 A1		08-07-2020
			JP 2020527264 A		03-09-2020
			JP 2022095645 A		28-06-2022
			US 2019102418 A1		04-04-2019
			WO 2019067911 A1		04-04-2019

10

20

30

40

50

フロントページの続き

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,N
E,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,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IQ,IR,IS,IT,JM,J
O,JP,KE,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,M
Z,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,ST,SV,SY,TH,
TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW

(特許庁注：以下のものは登録商標)

1. i O S

- 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
エイ、500、オラクル・インターナショナル・コーポレーション
- (72)発明者 チャクラボルティ，ソナック
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
エイ、500、オラクル・インターナショナル・コーポレーション
- (72)発明者 ブタニ，ハリッシュ・ラメシュ
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
エイ、500、オラクル・インターナショナル・コーポレーション
- (72)発明者 バスカー，ビベク
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
エイ、500、オラクル・インターナショナル・コーポレーション
- (72)発明者 アキナペッリ，サンディーブ
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
エイ、500、オラクル・インターナショナル・コーポレーション
- (72)発明者 ダス，ディー巴拉ジ
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
エイ、500、オラクル・インターナショナル・コーポレーション