(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6531056号
(P6531056)

(45)発行日　令和1年6月12日(2019.6.12)　(24)登録日　令和1年5月24日(2019.5.24)

(51)Int.Cl.　F I
G06F 16/00 (2019.01)　G06F 17/30 340Z
G06F 12/00 (2006.01)　G06F 12/00 513D

請求項の数 13　(全 17 頁)

(21)出願番号　特願2016-61823(P2016-61823)
(22)出願日　平成28年3月25日(2016.3.25)
(65)公開番号　特開2017-174290(P2017-174290A)
(43)公開日　平成29年9月28日(2017.9.28)
審査請求日　平成30年3月7日(2018.3.7)

(73)特許権者　000208891
ＫＤＤＩ株式会社
東京都新宿区西新宿二丁目３番２号
(74)代理人　100092772
弁理士　阪本　清孝
(74)代理人　100119688
弁理士　田邊　壽二
(72)発明者　斉藤　和広
埼玉県ふじみ野市大原二丁目１番１５号
株式会社ＫＤＤＩ研究所内

審査官　後藤　昂彦

最終頁に続く

(54)【発明の名称】クエリ処理制御装置、制御方法及び制御プログラム

(57)【特許請求の範囲】
【請求項１】
クライアントが投稿したクエリに対してストレージから必要なデータを取得し前記クエリの演算をバッファ上で処理して生成した演算結果をユーザに返すクエリ処理装置と、
前記クエリの演算処理に際して利用するメモリ上のバッファ領域を管理して前記バッファのサイズを指定するバッファ管理装置と、を備え、
前記バッファ管理装置は、
前記クエリ処理装置から実行するクエリの実行計画を取得して、事前設定された利用可能なメモリサイズとクエリ処理装置の演算の実装情報から演算毎のバッファの初期設定サイズを確保したバッファ情報を作成するバッファ計画部と、
前記クエリ処理装置から演算指定でバッファ情報の提供依頼を受け、前記バッファ情報を提供するとともに、前記クエリ処理装置から演算指定でバッファ増量依頼を受け、バッファ情報における当該バッファの利用状況に応じて終了した演算のバッファ領域を移行することでバッファのサイズを変更するバッファ制御部と、
前記クエリ処理装置から演算処理によって変化したバッファの利用状況を取得してバッファ情報を更新するバッファ利用量収集部と
を含むことを特徴とするクエリ処理制御装置。
【請求項２】
前記クエリ処理装置は、クエリ計画部とクエリ実行制御部と演算処理部を備え、
前記クエリ計画部は、クライアントが投稿したクエリを解釈し、クエリの実行計画を生

成して前記バッファ計画部及びクエリ実行制御部に実行計画を渡し、
前記クエリ実行制御部は、前記実行計画を基に前記バッファ制御部からバッファ情報を取得して演算が利用するバッファサイズを指定し、クエリ処理方式に沿って前記演算処理部に演算処理を依頼し、必要に応じて前記バッファ制御部にバッファの増量依頼し、
前記演算処理部は、前記クエリ実行制御部から依頼された演算を処理し、バッファの利用量の変化を前記バッファ利用量収集部に渡す
ことを特徴とする請求項１に記載のクエリ処理制御装置。
【請求項３】
前記バッファ制御部は、
前記クエリ処理装置から演算指定でバッファ増量依頼を受けた際に、増量対象が出力バッファである場合、当該演算の制御バッファを当該演算の出力バッファ領域に移行する請求項１又は請求項２に記載のクエリ処理制御装置。
【請求項４】
前記バッファ制御部は、
前記クエリ処理装置から演算指定でバッファ増量依頼を受けた際に、増量対象が制御バッファである場合、上限値が設定された当該演算の出力バッファを当該演算の制御バッファ領域に移行する請求項１又は請求項２に記載のクエリ処理制御装置。
【請求項５】
前記バッファ制御部は、
前記クエリ処理装置から演算指定でバッファ増量依頼を受けた際に、前記演算以外の他の演算の入力が完了している制御バッファを増量対象の演算のバッファ領域に移行する請求項１又は請求項２に記載のクエリ処理制御装置。
【請求項６】
前記バッファ制御部は、
前記クエリ処理装置から演算指定でバッファ増量依頼を受けた際に、上限値が設定された空きのバッファを増量対象の演算のバッファ領域に移行する請求項１又は請求項２に記載のクエリ処理制御装置。
【請求項７】
クライアントが投稿したクエリに対してストレージから必要なデータを取得し前記クエリの演算をバッファ上で処理して生成した演算結果をユーザに返すクエリ処理において、
前記クエリを解釈して生成したツリー構造の最下位の演算から順に処理する実体化型クエリ処理方式によるクエリ演算の実行計画を生成する手順と、
事前設定された利用可能なメモリサイズと演算の実装情報から前記クエリの演算毎のバッファサイズを指定する手順と、
演算結果が格納されたバッファが上限であり且つ下位の出力バッファにデータがある場合にバッファサイズの増量依頼を受け、バッファの利用状況に応じて終了した演算のバッファ領域を空き領域として検索して出力バッファまたは制御バッファのいずれかの対象バッファに付与する手順と、を備えることで、
クエリ実行時に、前記演算毎に利用するバッファサイズを動的に制御することで、ストレージ利用と演算切り換えの回数を削減する
ことを特徴とするクエリ処理制御方法。
【請求項８】
クライアントが投稿したクエリに対してストレージから必要なデータを取得し前記クエリの演算をバッファ上で処理して生成した演算結果をユーザに返すクエリ処理において、
前記クエリを解釈して生成したツリー構造の最上位から順に下位の演算にデータ生成を依頼して処理する要求駆動型クエリ処理方式によるクエリ演算の実行計画を生成する手順と、
事前設定された利用可能なメモリサイズと前記クエリ演算の実装情報から演算毎のバッファサイズを指定する手順と、
制御バッファが上限であり且つ下位の出力バッファにデータがある場合に制御バッファ

のバッファサイズの増量依頼を受けるとともに、演算結果が格納されたバッファ（出力バッファもしくは制御バッファ）が上限に達しておらず且つ直下の演算が終了していない場合に出力バッファのバッファサイズの増量依頼を受け、バッファの利用状況に応じて終了した演算のバッファ領域を空き領域として検索して出力バッファまたは制御バッファのいずれかの対象バッファに付与する手順と、を備えることで、

クエリ実行時に、前記演算毎に利用するバッファサイズを動的に制御することで、ストレージ利用と演算切り換えの回数を削減する

ことを特徴とするクエリ処理制御方法。

【請求項９】

演算に対してバッファサイズの増量依頼を受けた際に、増量対象が出力バッファである場合、当該演算の制御バッファを前記空き領域とする請求項７又は請求項８に記載のクエリ処理制御方法。

【請求項１０】

演算に対してバッファサイズの増量依頼を受けた際に、増量対象が制御バッファである場合、上限値が設定された当該演算の出力バッファを前記空き領域とする請求項７又は請求項８に記載のクエリ処理制御方法。

【請求項１１】

演算に対してバッファサイズの増量依頼を受けた際に、前記演算以外の他の演算の入力が完了している制御バッファを前記空き領域とする請求項７又は請求項８に記載のクエリ処理制御方法。

【請求項１２】

演算に対してバッファサイズの増量依頼を受けた際に、上限値が設定された空きのバッファを前記空き領域とする請求項７又は請求項８に記載のクエリ処理制御方法。

【請求項１３】

請求項７から請求項１２のいずれか１項に記載の各手順をコンピュータに実行させるクエリ処理制御プログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、SQLクエリを処理するデータベースシステムにおいて、利用可能なメモリサイズを超えるSQLクエリを高速に実行可能とするためのクエリ処理制御装置、制御方法及び制御プログラムに関する。

【背景技術】

【０００２】

データベースシステムにおけるクエリ処理装置は、ユーザが投稿したクエリを解釈し、必要なデータをストレージから取得してクエリを処理する。多くのデータベースシステムではSQLと呼ばれるクエリ言語を利用しており、クエリ処理装置はSQLクエリを解釈することで、関係代数に沿った演算を抽出し、各演算の処理順序を決めることができる。この処理順序は、関係代数式としてツリー構造をしたクエリ実行計画で表現される。

演算はデータに対する加工処理を表現している。選択演算や射影演算などはデータから不要な部分を削減するが、一方で結合演算や集合演算のように複数のデータを組み合せる演算や、集計演算を多数行う場合など、元のデータよりサイズを大きくする演算が存在する。

【０００３】

クエリ処理装置は、クエリ処理を装置のメモリ上で行う必要がある。以降、クエリ処理において演算が利用するメモリ領域をバッファと呼ぶ。メモリは有限であり、また多数のクエリを同時に実行する場合など、バッファに利用できる物理メモリが小さく制限される場合がある。元のデータが大規模である場合や、前述のようにデータサイズを大きくする演算によって、このようなバッファに利用可能なメモリサイズを超えるデータが発生する場合、OSのスワップ機構の利用によって大幅な遅延が発生する、又は、クエリ実行に失敗

する事態が発生する。
【０００４】
クエリ処理方式としては、実体化型クエリ処理方式（Materialized model又はBulk processing model）と、要求駆動型クエリ処理方式（Volcano model又はPipeline model）が提案されている。
実体化型クエリ処理方式は、非特許文献１に示されるように、クエリ実行計画におけるツリー構造の最下位の演算から順に処理する。演算は、入力される全てのデータを処理完了すると、次の上位の演算に処理を移行する。
本方式では、演算が利用するバッファのサイズに制限がないため、演算の処理中に対象のデータサイズが利用可能なメモリサイズを超える場合がある。それによって処理の大幅な遅延やクエリ実行失敗が発生する。
【０００５】
要求駆動型クエリ処理方式は、非特許文献２に示されるように、クエリ実行計画におけるツリー構造の最上位から順に下位の演算にデータ生成を依頼する。依頼された演算は、データを生成するために演算に必要なデータを同様に下位の演算に依頼する。入力データが得られた演算は、演算処理によって要求されたデータを生成できた段階で上位の演算に処理を移行する。
この要求単位は、１レコードもしくはストレージのディスクブロックなどの固定サイズであり、利用可能なメモリサイズを超える事象は発生しない。
【０００６】
しかし、大規模なデータを対象とする処理において、演算の切り替えが頻繁に発生し、性能が劣化する。また、仮に要求単位を大きく設定した場合、演算毎に所持するデータの規模が膨らみ、同時実行のクエリが増えた場合や、演算の数が多いクエリが実行された場合において、実体化型クエリ処理方式と同様に演算が利用するバッファの制御をしないため、利用可能なメモリサイズを超える現象が発生する。
更には、全データの蓄積を必要とする演算（整列演算等）や、二項演算（結合演算等）における片方のデータの蓄積が必要な実装（ハッシュ結合等）においては、実体化型クエリ処理方式と同様に、その蓄積データが利用可能なメモリサイズを超える現象が発生する。
【０００７】
本発明者は、実体化型クエリ処理方式をベースに、大規模データにおいても確実にクエリ処理を完了する分割実行型クエリ処理方式の提案を行った（特許文献１）。
この方式は、利用可能なメモリサイズを超えた場合に、OSのスワップ機構とは別の仕組みでストレージを活用することで、確実にクエリ処理を完了する。更に、ストレージの読み込み回数を最小化する演算処理手法を導入することで、性能劣化を抑止している。
【先行技術文献】
【特許文献】
【０００８】
【特許文献１】特願２０１５－００２２９１
【非特許文献】
【０００９】
【非特許文献１】F. Groffen, N. Nes, S. Nes, and S. Kersten, "MonetDB: Two Decades of Research in Column-oriented Database Architectures", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, pp. 1-6, 2012.
【非特許文献２】G. Graefe, "Volcano-an extensible and parallel query evaluation system", IEEE Transactions on Knowledge and Data Engineering, vol. 6, no. 1, pp. 120-135, 1994
【発明の概要】
【発明が解決しようとする課題】
【００１０】

しかしながら、特許文献１に記載の実体化型クエリ処理方式を利用した処理によれば、データサイズを削減するような演算においても、入力データが利用可能メモリサイズを超える場合、要求駆動型クエリ処理方式のように上位の演算に処理後のデータを渡すことをせず、処理後の少量データであっても一時的にストレージに格納するため、無駄なストレージ利用が発生し性能が悪化する可能性があった。

【００１１】

本発明は上記実情に鑑みて提案されたもので、要求駆動型クエリ処理方式及び実体化型クエリ処理方式のクエリ処理制御装置において、ストレージの利用回数と演算の切り替え回数を最小化して高速なクエリ処理を実現することで、大規模データを対象とするクエリ処理を確実に実行可能とするクエリ処理制御装置、制御方法及び制御プログラムを提供することを目的としている。

【課題を解決するための手段】

【００１２】

上記目的を達成するため本発明は、演算単位のバッファの利用量が上限に達した段階で次の演算に処理を推移させてストレージの利用回数を削減し、更にバッファサイズをクエリ実行時に動的に制御して演算の切り替えを最小限にすることで、確実にクエリ処理を完了しつつ性能の向上を図っている。

【００１３】

すなわち、請求項１のクエリ処理制御装置は、

クライアントが投稿したクエリに対してストレージから必要なデータを取得し前記クエリの演算をバッファ上で処理して生成した演算結果をユーザに返すクエリ処理装置と、

前記クエリの演算処理に際して利用するメモリ上のバッファ領域を管理して前記バッファのサイズを指定するバッファ管理装置と、を備え、

前記バッファ管理装置は、

前記クエリ処理装置から実行するクエリの実行計画を取得して、事前設定された利用可能なメモリサイズとクエリ処理装置の演算の実装情報から演算毎のバッファの初期設定サイズを確保したバッファ情報を作成するバッファ計画部と、

前記クエリ処理装置から演算指定でバッファ情報の提供依頼を受け、前記バッファ情報を提供するとともに、前記クエリ処理装置から演算指定でバッファ増量依頼を受け、バッファ情報における当該バッファの利用状況に応じて終了した演算のバッファ領域を移行することでバッファのサイズを変更するバッファ制御部と、

前記クエリ処理装置から演算処理によって変化したバッファの利用状況を取得してバッファ情報を更新するバッファ利用量収集部とを含むことを特徴としている。

【００１４】

請求項２は、請求項１のクエリ処理制御装置において、

前記クエリ処理装置は、クエリ計画部とクエリ実行制御部と演算処理部を備え、

前記クエリ計画部は、クライアントが投稿したクエリを解釈し、クエリの実行計画を生成して前記バッファ計画部及びクエリ実行制御部に実行計画を渡し、

前記クエリ実行制御部は、前記実行計画を基に前記バッファ制御部からバッファ情報を取得して演算が利用するバッファサイズを指定し、クエリ処理方式に沿って前記演算処理部に演算処理を依頼し、必要に応じて前記バッファ制御部にバッファの増量依頼し、

前記演算処理部は、前記クエリ実行制御部から依頼された演算を処理し、バッファの利用量の変化を前記バッファ利用量収集部に渡すことを特徴としている。

【００１５】

請求項３は、請求項１又は請求項２のクエリ処理制御装置において、

前記バッファ制御部は、

前記クエリ処理装置から演算指定でバッファ増量依頼を受けた際に、増量対象が出力バッファである場合、当該演算の制御バッファを当該演算の出力バッファ領域に移行することを特徴としている。

【００１６】

請求項４は、請求項１又は請求項２のクエリ処理制御装置において、
前記バッファ制御部は、
前記クエリ処理装置から演算指定でバッファ増量依頼を受けた際に、増量対象が制御バッファである場合、上限値が設定された当該演算の出力バッファを当該演算の制御バッファ領域に移行することを特徴としている。
【００１７】
請求項５は、請求項１又は請求項２のクエリ処理制御装置において、
前記バッファ制御部は、
前記クエリ処理装置から演算指定でバッファ増量依頼を受けた際に、前記演算以外の他の演算の入力が完了している制御バッファを増量対象の演算のバッファ領域に移行することを特徴としている。
【００１８】
請求項６は、請求項１又は請求項２のクエリ処理制御装置において、
前記バッファ制御部は、
前記クエリ処理装置から演算指定でバッファ増量依頼を受けた際に、上限値が設定された空きのバッファを増量対象の演算のバッファ領域に移行することを特徴としている。
【００１９】
請求項７のクエリ処理制御方法は、
クライアントが投稿したクエリに対してストレージから必要なデータを取得し前記クエリの演算をバッファ上で処理して生成した演算結果をユーザに返すクエリ処理において、
前記クエリを解釈して生成したツリー構造の最下位の演算から順に処理する実体化型クエリ処理方式によるクエリ演算の実行計画を生成する手順と、
事前設定された利用可能なメモリサイズと演算の実装情報から前記クエリの演算毎のバッファサイズを指定する手順と、
演算結果が格納されたバッファが上限であり且つ下位の出力バッファにデータがある場合にバッファサイズの増量依頼を受け、バッファの利用状況に応じて終了した演算のバッファ領域を空き領域として検索して出力バッファまたは制御バッファのいずれかの対象バッファに付与する手順と、を備えることで、
クエリ実行時に、前記演算毎に利用するバッファサイズを動的に制御することで、ストレージ利用と演算切り換えの回数を削減することを特徴としている。
【００２０】
請求項８のクエリ処理制御方法は、
クライアントが投稿したクエリに対してストレージから必要なデータを取得し前記クエリの演算をバッファ上で処理して生成した演算結果をユーザに返すクエリ処理において、
前記クエリを解釈して生成したツリー構造の最上位から順に下位の演算にデータ生成を依頼して処理する要求駆動型クエリ処理方式によるクエリ演算の実行計画を生成する手順と、
事前設定された利用可能なメモリサイズと前記クエリ演算の実装情報から演算毎のバッファサイズを指定する手順と、
制御バッファが上限であり且つ下位の出力バッファにデータがある場合に制御バッファのバッファサイズの増量依頼を受けるとともに、演算結果が格納されたバッファ（出力バッファもしくは制御バッファ）が上限に達しておらず且つ直下の演算が終了していない場合に出力バッファのバッファサイズの増量依頼を受け、バッファの利用状況に応じて終了した演算のバッファ領域を空き領域として検索して出力バッファまたは制御バッファのいずれかの対象バッファに付与する手順と、を備えることで、
クエリ実行時に、前記演算毎に利用するバッファサイズを動的に制御することで、ストレージ利用と演算切り換えの回数を削減することを特徴としている。
【００２１】
請求項９は、請求項７又は請求項８のクエリ処理制御方法において、
演算に対してバッファサイズの増量依頼を受けた際に、増量対象が出力バッファである

場合、当該演算の制御バッファを前記空き領域とすることを特徴としている。
【００２２】
請求項１０は、請求項７又は請求項８のクエリ処理制御方法において、
演算に対してバッファサイズの増量依頼を受けた際に、増量対象が制御バッファである場合、上限値が設定された当該演算の出力バッファを前記空き領域とすることを特徴としている。
【００２３】
請求項１１は、請求項７又は請求項８のクエリ処理制御方法において、
演算に対してバッファサイズの増量依頼を受けた際に、前記演算以外の他の演算の入力が完了している制御バッファを前記空き領域とすることを特徴としている。
【００２４】
請求項１２は、請求項７又は請求項８のクエリ処理制御方法において、
演算に対してバッファサイズの増量依頼を受けた際に、上限値が設定された空きのバッファを前記空き領域とすることを特徴としている。
【００２５】
請求項１３のクエリ処理制御プログラムは、請求項７から請求項１２のいずれか１項に記載の各手順をコンピュータに実行させることを特徴としている。
【発明の効果】
【００２６】
請求項１のクエリ処理制御装置によれば、クエリ処理装置から演算指定でバッファ増量依頼を受け、バッファ情報における当該バッファの利用状況に応じて終了した演算のバッファ領域を移行することでバッファのサイズを変更するので、クエリ処理実行時に演算毎に利用するメモリサイズを制御することができる。その結果、演算が利用するバッファを動的に制御して、利用可能なメモリサイズを超えることを抑制し、ストレージの利用と演算の切り替えを最小限にすることで、大規模データの確実なクエリ処理実行及び処理の高速化を実現することができる。
【００２７】
請求項２のクエリ処理制御装置によれば、クエリ実行制御部が実行計画を基にバッファ制御部からバッファ情報を取得して演算が利用するバッファサイズを指定し、クエリ処理方式に沿って演算処理部に演算処理を依頼し、必要に応じてバッファ制御部にバッファの増量依頼することで、バッファのサイズを変更することができる。
【００２８】
請求項７のクエリ処理制御方法によれば、クエリを解釈して生成したツリー構造の最下位の演算から順に処理する実体化型クエリ処理方式により、クエリ演算の実行計画を生成することができる。
【００２９】
請求項８のクエリ処理制御方法によれば、クエリを解釈して生成したツリー構造の最上位から順に下位の演算にデータ生成を依頼して処理する要求駆動型クエリ処理方式により、クエリ演算の実行計画を生成することができる。
【００３０】
請求項３のクエリ処理制御装置及び請求項９のクエリ処理制御方法によれば、演算の増量対象が出力バッファである場合に、当該演算の制御バッファを当該演算のバッファ領域に移行することでバッファのサイズを変更することができる。
【００３１】
請求項４のクエリ処理制御装置及び請求項１０のクエリ処理制御方法によれば、演算の増量対象が制御バッファである場合に、上限値が設定された当該演算の出力バッファを当該演算のバッファ領域に移行することでバッファのサイズを変更することができる。
【００３２】
請求項５のクエリ処理制御装置及び請求項１１のクエリ処理制御方法によれば、他の演算の入力が完了している制御バッファを増量対象の演算のバッファ領域に移行することで

バッファのサイズを変更することができる。
【００３３】
請求項６のクエリ処理制御装置及び請求項１２のクエリ処理制御方法によれば、上限値が設定された空きのバッファを増量対象の演算のバッファ領域に移行することでバッファのサイズを変更することができる。
【００３４】
請求項１３のクエリ処理制御プログラムによれば、コンピュータ上にクエリ処理制御装置を構築させることができる。
【図面の簡単な説明】
【００３５】
【図１】本発明の一実施形態に係るクエリ処理制御装置の構成を示すブロック図である。
【図２】バッファ情報の構成を表す説明図である。
【図３】バッファ計画部における処理フローを示すフローチャート図である。
【図４】バッファ制御部におけるバッファ増量処理の処理フローを示すフローチャート図である。
【図５】バッファ利用量収集部における処理フローを示すフローチャート図である。
【図６】本発明の他の実施形態に係るクエリ処理制御装置の構成を示すブロック図である。
【図７】実体化型クエリ処理方式におけるクエリ実行制御部の処理フローを示すフローチャート図である。
【図８】要求駆動型クエリ処理方式におけるクエリ実行制御部の処理フローを示すフローチャート図である。
【図９】実体化型クエリ処理方式及び要求駆動型クエリ処理方式における演算処理部の処理フローを示すフローチャート図である。
【発明を実施するための形態】
【００３６】
本発明の一実施形態のクエリ処理制御装置について、図１の構成ブロック図を参照しながら説明する。
クライアント１、ストレージ２、バッファ３、クエリ処理装置１０及びバッファ管理装置２０は、基本プログラムや各種の基本デバイスが記憶されたＲＯＭと、各種のプログラムやデータが記憶されるハードディスクドライブ装置（ＨＤＤ）と、ＣＲ－ＲＯＭやＤＶＤ等の記憶媒体からプログラムやデータを読み出すメディアドライブ装置と、プログラムを実行するＣＰＵと、このＣＰＵにワークエリアを提供するＲＡＭと、外部装置と通信するパラレル／シリアルＩ／Ｆとを主要部分とする一般的な構成を備えたコンピュータ上に構築されている。
例えば、上述した構成を有するコンピュータにおいて、クエリ処理を実行するためのクエリ処理制御プログラムがメディアドライブ装置を介してＨＤＤにインストールされることでクエリ処理制御装置が構築される。
【００３７】
クライアント１は、クエリ処理装置１０に対してクエリを投稿し、処理結果を取得する。クエリ処理装置１０は投稿クエリの処理結果を出力し、バッファ管理装置２０はクエリの演算処理を行うバッファ（バッファ領域）３を管理する。
クエリ処理装置１０は、クライアント１が投稿したクエリに対して、ストレージ２から必要なデータを取得して、バッファ管理装置２０が指定するサイズのバッファ（バッファ領域）３上でクエリの演算を処理することで、処理結果を生成し、ユーザに処理結果を返す。
【００３８】
ストレージ２は、クエリ処理装置１０上及びネットワーク経由先の磁気ディスク、SSD（Solid State Drive）、フラッシュメモリ、物理メモリ、外部記憶媒体を含む装置であり、クエリ処理対象のデータを所持している。

【００３９】
バッファ管理装置２０は、バッファ３のサイズを決定するバッファ計画部２１と、バッファ３の利用状況からサイズを変更するバッファ制御部２２と、バッファ３の利用状況を取得するバッファ利用量収集部２３から構成されている。
バッファ計画部２１は、クエリ処理装置１０から実行するクエリの実行計画を取得して、事前設定された利用可能なメモリサイズ（利用可能メモリ２６）と、クエリ処理装置１０の演算の実装情報（演算の実装情報２５）から演算毎のバッファ３の初期設定サイズを決定し、バッファ情報２７を作成する。
【００４０】
バッファ制御部２２は、クエリ処理装置１０から演算指定でバッファ情報提供依頼を受け、バッファ情報２７を提供するとともに、クエリ処理装置１０から演算指定でバッファ増量依頼を受け、バッファ情報２７における当該バッファの利用状況からバッファ３のサイズを変更する。バッファサイズを変更することで、演算が利用するバッファを動的に制御し、演算の切り替えを最小限にすることができる。また、バッファを増量することで、演算が利用可能なメモリサイズを超えることを抑制することができる。
【００４１】
バッファ利用量収集部２３は、クエリ処理装置１０から演算処理によって変化したバッファ３の利用状況を取得してバッファ情報２７の更新を行うことで、演算処理で利用するメモリ上のバッファ　領域を管理することができる。
【００４２】
次に、バッファ管理装置２０においてバッファ情報２７の作成及び変更を行う場合の処理について、図２～５を参照して説明する。
先ず、バッファ情報２７の構成について、図２を参照して説明する。ｏｐ[i]はi番目の演算処理（演算を実際に処理する動作）を示し、Ｂ[i]は演算処理ｏｐ[i]が利用するバッファを示す。
バッファＢは、投稿クエリの演算単位に生成される。バッファは、処理結果を格納する出力バッファoutbufと、演算処理に必要な中間結果を格納する制御バッファctrlbufから構成される。
演算（処理の内容）には、「結合」「直積」「集計」「整列」「集合」「選択」「射影」等の関係代数で定義されているものが存在し、演算専用のバッファが用意される。演算で処理される入力データは、その演算の直下の演算の出力バッファと、自身専用の制御バッファから取得する。
【００４３】
通常の実行計画では、最下位の演算はストレージ２からデータを取得するScan演算であり、最上位の演算はクライアント１に処理結果を返すOutput演算である。バッファ利用の例外として、Scan演算は入力先がストレージ２であり、Output演算は専用のバッファが存在しない。従って、演算毎のバッファの数は、演算の数－１となる。また、演算の実装アルゴリズムによって、制御バッファの有無が異なる。
例えば、結合演算のアルゴリズムがHash Joinの場合、片方の入力データから生成したハッシュテーブルを制御バッファに保管する必要がある。一方で、射影演算のように、一時的にデータを保存する必要のない演算の場合、制御バッファは不要となる。このために、クエリ処理装置１０が処理可能な演算それぞれに対する制御バッファの有無を、演算の実装情報２５に定義する。
【００４４】
バッファ管理装置２０のバッファ計画部２１における処理手順について、図３のフローチャート図を参照して説明する。
クエリ処理装置１０側からクエリの実行計画を取得し（ステップ３１）、演算の構成から、バッファ情報２７の元となるツリー構造を生成する。
次に、事前に設定された利用可能なメモリサイズと、演算の実装情報を元に、バッファ情報の各演算にバッファのサイズを設定していく（ステップ３２）。演算の実装情報には

、実行計画で出現する演算の種類毎に、制御バッファの有無や、サイズの設定方法が設定されている。

【００４５】

最も単純なバッファサイズの決定方法は、利用可能なメモリサイズをバッファの数で割った値をバッファサイズに設定する。制御バッファがあるバッファは、そのバッファサイズを半分にして出力バッファと制御バッファのサイズに設定する。

各演算のバッファをメモリ上に確保し、バッファ情報２７を作成する（ステップ３３）。

【００４６】

バッファ制御部２２におけるバッファ増量処理の処理手順について、図４のフローチャート図を参照して説明する。

バッファ制御部２２がクエリ処理装置１０から演算のバッファの増量依頼を受けると（ステップ４１）、バッファ情報２７から空いているバッファを検索し（ステップ４２）、バッファの空き領域の有無を判定する（ステップ４３）。

ステップ４３でバッファの空き領域を確保できた場合、対象のバッファに付与し（ステップ４４）、かつ当該空き領域を開放する。これらの変更をバッファ情報２７に適用する（ステップ４５）。

【００４７】

ステップ４３で空き領域を確保できなかった場合、増量対象が制御バッファか出力バッファのどちらなのかを判断する（ステップ４６）。

増量対象が制御バッファの場合は、ストレージ２の領域を付与して（ステップ４７）、バッファ情報を更新する（ステップ４５）。

増量対象が出力バッファの場合は、増量不可としてクエリ処理装置１０に返す（ステップ４８）。なお、出力バッファは拡張しないことで処理が止まることはないため、ストレージ２の領域を付与しない。そのため、ストレージ２のバッファ領域としての利用を最小限に抑えることができる。

【００４８】

ステップ４３におけるバッファの空き領域の有無の判定基準は複数存在し、以下に列挙する。これらは、それぞれ単体として利用可能であり、また組み合せて利用することも可能である。

・演算処理が終了しているバッファ（出力バッファ及び制御バッファ）の未使用領域を「空き」と判定する。

・増量依頼されたバッファが出力バッファの場合、当該演算の制御バッファの未使用領域を「空き」と判定する。ただし、制御バッファへの入力が完了していることを前提とする。

・増量依頼されたバッファの演算以外の入力が完了している制御バッファの未使用領域を「空き」と判定する。

・増量依頼されたバッファが制御バッファの場合、当該演算の出力バッファの未使用領域を「空き」と判定する。ただし、まだ利用される可能性があるため、全領域を取得されないために、空き領域として取得可能な上限サイズを出力バッファサイズの割合（例えば、５０％まで等）で設定する必要がある。

・増量依頼されたバッファの演算以外のバッファの未使用領域を「空き」と判定する。ただし、まだ利用される可能性があるため、全領域を取得されないために、空き領域として取得可能な上限サイズを出力バッファサイズの割合（例えば、５０％まで等）で設定する必要がある。

・増量依頼されたバッファが制御バッファの場合、外部記憶装置としてのストレージの空き領域を「空き」と判定してもよい。

【００４９】

次に、バッファ利用量収集部２３における処理手順について、図５のフローチャート図を参照して説明する。

まず、クエリ処理装置１０の演算処理におけるバッファの利用状況を取得する（ステップ５１）。取得する利用状況の情報は、入力元のバッファ（直下の演算の出力バッファもしくは当該演算の制御バッファ）、入力データサイズ、出力先のバッファ（当該演算の出力バッファもしくは制御バッファ）、出力データサイズ、該当の演算の処理完了フラグ、制御バッファがある場合の制御バッファへの入力完了フラグを含む。

次に、取得した利用状況の情報から、バッファ情報における入力元のバッファのサイズと出力先のバッファの利用サイズ、空き領域、完了状況を更新する（ステップ５２及びステップ５３）。

【００５０】

上述したクエリ処理制御装置によれば、クエリ処理装置１０でのクエリ処理実行時において、バッファ管理装置２０のバッファ情報２７で利用状況を管理することで、クエリの演算毎に利用するバッファサイズを動的に制御し、利用可能なメモリサイズを超えることを抑制し、ストレージ２の利用と演算の切り替えを最小限にすることで、大規模データの確実なクエリ処理実行及び処理の高速化を実現することができる。

【００５１】

続いて、クエリ処理装置１０の実施形態として、実体化型クエリ処理方式と、要求駆動型クエリ処理方式に適用した場合の構成について、図６を参照して説明する。

図６のクエリ処理制御装置のクエリ処理装置１０は、クエリの実行計画（ツリー構造）を生成するクエリ計画部１１と、バッファの増量依頼をするクエリ実行制御部１２と、バッファの利用量の変化を把握する演算処理部１３とを備えて構成されている。

【００５２】

クエリ計画部１１は、クライアント１が投稿したクエリを解釈し、クエリの実行計画を生成してバッファ計画部２１及びクエリ実行制御部１２に実行計画を渡す。

クエリ実行制御部１２は、実行計画を基に、バッファ制御部２２からバッファ情報を取得して演算が利用するバッファサイズを指定し、クエリ処理方式に沿って演算処理部１３に演算処理を依頼し、必要に応じてバッファ制御部２２にバッファの増量依頼をする。

演算処理部１３は、クエリ実行制御部１２から依頼された演算を処理し、バッファの利用量の変化をバッファ利用量収集部２３に渡す。

【００５３】

実体化型クエリ処理方式を採用した場合のクエリ処理制御装置におけるクエリ実行制御部１２の処理手順について、図７のフローチャート図を参照して説明する。

クエリ計画部１１から実行計画を取得し（ステップ７０１）、ツリー構造の最下位の演算から処理を開始する（ステップ７０２）。通常、最下位の演算は、ストレージ２からデータを取得するScan演算であり、これを選択する。

ここで、選択された演算とその直下の演算のバッファ情報をバッファ制御部２２から取得する（ステップ７０３）。なお、Scan演算の場合は直下の演算は存在しないため、直下の演算のバッファ情報は取得しない。同様にOutput演算も自身のバッファが存在しないため、取得しない。

【００５４】

バッファ情報を基に対象演算の実行判定をする（ステップ７０４）。

実行可能である場合、演算処理部１３に当該演算の実行を依頼する（ステップ７０５）。

実行完了後、その処理によるバッファ情報の変化を知るために、バッファ制御部２２からバッファ情報を再度取得する（ステップ７０６）。

ここで、当該演算の増量判定を行い（ステップ７０７）、増量が必要な場合は、バッファ制御部２２に増量を依頼する（ステップ７０８）。

増量ができた場合（ステップ７０９）、再度演算処理に戻る（ステップ７０５）。

増量ができなかった場合、対象の演算を直上の演算に変更し（ステップ７１０）、再度バッファ情報の取得（ステップ７０３）と実行判定（ステップ７０４）に戻る。

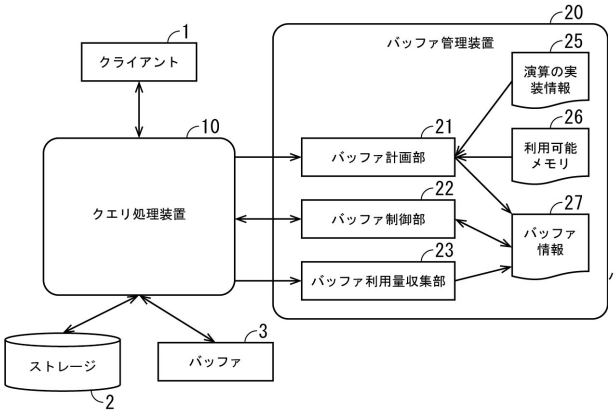
【００５５】

ステップ７０７のバッファの増量判定において、増量が不要であった場合、当該演算の終了判定を行う（ステップ７１１）。
終了したと判定された場合、当該演算を終了設定する（ステップ７１２）。ここでもし実行計画における最上位の演算であった場合、クエリ処理完了となる。
【００５６】
ステップ７１１で終了していないと判定された場合、次に処理する演算を判定（推移判定）し、対象の演算を直上の演算（ステップ７１０）又は直下の演算（ステップ７１４）に変更し、再度バッファ情報の取得（ステップ７０３）と実行判定（ステップ７０４）に戻る。
また、ステップ７０４の演算の実行判定において、実行しないと判定された場合は、推移判定（ステップ７１３）によって対象の演算を変更し（ステップ７１０又はステップ７１４）、再度バッファ情報の取得（ステップ７０３）と実行判定（ステップ７０４）に戻る。
【００５７】
実体化型クエリ処理方式におけるクエリ実行制御部１２の実行判定（ステップ７０４）は、以下の条件（１）及び（２）を全て満たす場合に「実行する」と判定される。全て満たさない場合は、「実行しない」と判定される。
（１）出力バッファが空である。
（２）下位の出力バッファにデータがある、または、入力データがストレージで読み込みデータの残りがある、または、制御バッファにデータがある。
【００５８】
実体化型クエリ処理方式におけるクエリ実行制御部１２の増量判定（ステップ７０７）は、以下の条件（１）及び（２）を全て満たす場合に増量すると判定される。満たさない場合は、増量しないと判定される。
（１）演算結果が格納されたバッファが上限である。
（２）下位の出力バッファにデータがある。
【００５９】
実体化型クエリ処理方式におけるクエリ実行制御部１２の終了判定（ステップ７１１）は、以下の条件（１）～（３）を全て満たす場合に「終了する」と判定される。全て満たさない場合は、「終了しない」と判定される。
（１）下位の演算が終了している、または、入力がストレージである。
（２）下位の出力バッファが空である、または、ストレージの読み込みデータの残りがない。
（３）制御バッファがない、または、制御バッファが空である。
【００６０】
実体化型クエリ処理方式におけるクエリ実行制御部１２の演算推移判定（ステップ７１３）は、以下の条件（１）及び（２）を全て満たす場合に「直下の演算」に推移すると判定される。全て満たさない場合は、「直上の演算」に推移すると判定される。
（１）出力バッファが空である。
（２）直下の演算が未終了である。
【００６１】
要求駆動型クエリ処理方式を採用した場合のクエリ処理制御装置におけるクエリ実行制御部１２の処理手順について、図８のフローチャート図を参照して説明する。
クエリ計画部１１から実行計画を取得し（ステップ８０１）、ツリー構造の最上位の演算から処理を開始する（ステップ８０２）。通常、最上位の演算は、クライアントに処理結果を渡すOutput演算であり、これを選択する。
ここで、選択された演算とその直下の演算のバッファ情報をバッファ制御部２２から取得する（ステップ８０３）。なお、Scan演算の場合は直下の演算は存在しないため、直下の演算のバッファ情報は取得しない。同様にOutput演算も自身のバッファが存在しないため、取得しない。

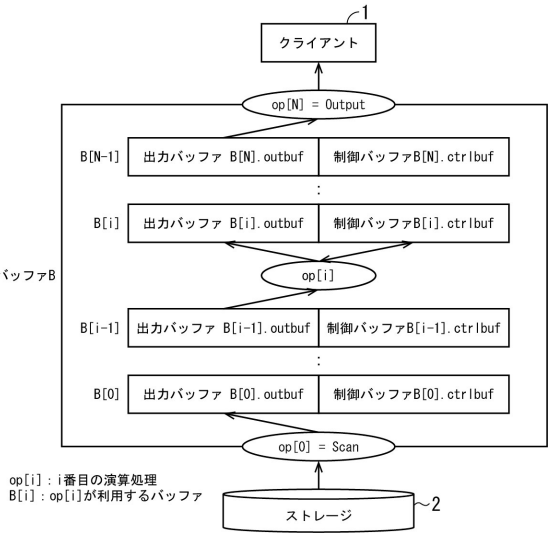
【００６２】
バッファ情報を基に対象演算の実行判定をする（ステップ８０４）。
実行不可である場合、直下の演算を対象の演算とし（ステップ８１４）、出力バッファのサイズ分のデータ作成を依頼して、再度バッファ情報の取得（ステップ８０３）と実行判定（ステップ８０４）に戻る。
実行可能である場合、演算処理部１３に当該演算の実行を依頼する（ステップ８０５）。
実行完了後、その処理によるバッファ情報の変化を知るために、バッファ制御部２２からバッファ情報を再度取得する（ステップ８０６）。
【００６３】
ここで、当該演算の制御バッファサイズと、直下の演算の出力バッファサイズ（作成要求するデータサイズ）の増量判定を行う（ステップ８０７）。
当該演算の制御バッファサイズに増量が必要な場合は、バッファ制御部２２にこの増量を依頼する（ステップ８０８）。
ステップ８０９で制御バッファサイズの増量ができた場合は、再度演算処理に戻る（ステップ８０５）。制御バッファサイズの増量ができなかった場合は、対象の演算を直上の演算に変更し（ステップ８１０）、再度バッファ情報の取得（ステップ８０３）と実行判定（ステップ８０４）に戻る。
【００６４】
ステップ８０７において直下の演算の出力バッファサイズの増量が必要な場合も、同様にバッファ制御部２２にこの増量を依頼する（ステップ８０８）。
ステップ８０９で出力バッファサイズの増量できなかった場合は、同様に直上の演算に移行する（ステップ８１０）が、増量できた場合は、次に処理する演算を判定（推移判定）する（ステップ８１２）。
ステップ８１２で直上の演算と判断され場合は、対象の演算を変更して（ステップ８１０）、再度バッファ情報の取得（ステップ８０３）と実行判定（ステップ８０４）に戻る。
ステップ８１２で「直下の演算」と判断された場合は、「直下の演算」にデータ生成を依頼する（ステップ８１４）。
バッファの増量判定（ステップ８０７）において、増量が不要であった場合、当該演算の終了判定を行う（ステップ８１１）。
【００６５】
終了したと判定された場合、当該演算を終了設定する（ステップ８１３）。ここでもし実行計画における最上位の演算であった場合、クエリ処理完了となる。
ステップ８１１で終了していないと判定された場合、次に処理する演算を判定（推移判定）し（ステップ８１２）、対象の演算を変更して（ステップ８１０、８１４）、再度バッファ情報の取得（ステップ８０３）と実行判定（ステップ８０４）に戻る。
【００６６】
要求駆動型クエリ処理方式におけるクエリ実行制御部１２の実行判定（ステップ８０４）は、以下の条件（１）及び（２）を全て満たす場合に「実行する」と判定される。全て満たさない場合は、「実行しない」と判定される。
（１）出力バッファが上限でない。
（２）下位の出力バッファにデータがある、または、入力データがストレージで読み込みデータの残りがある。
【００６７】
要求駆動型クエリ処理方式におけるクエリ実行制御部１２の増量判定（ステップ８０７）は、以下の条件を全て満たす場合に、対象の演算の制御バッファを「増量する」と判定される。全てを満たさない場合は、「増量しない」と判定される。
（１）制御バッファが上限である。
（２）下位の出力バッファにデータがある。

一方で、以下の条件を全て満たす場合に、直下の演算の出力バッファを「増量する」と判定される。
（１）演算結果が格納されたバッファ（出力バッファもしくは制御バッファ）が上限に達してない。
（２）直下の演算が終了していない。
【００６８】
要求駆動型クエリ処理方式におけるクエリ実行制御部１２の終了判定（ステップ８１１）は、以下の条件（１）～（３）を全て満たす場合に「終了する」と判定される。全て満たさない場合は、「終了しない」と判定される。
（１）下位の演算が終了している、または、入力がストレージである。
（２）下位の出力バッファが空である、または、ストレージの読み込みデータの残りがない。
（３）制御バッファがない、または、制御バッファが空である。
【００６９】
要求駆動型クエリ処理方式におけるクエリ実行制御部１２の推移判定（ステップ８１２）は、以下の条件（１）及び（２）を全て満たす場合に直上に推移すると判定される。全て満たさない場合は、直下に推移すると判定される。
（１）出力バッファが上限に達する（要求量のデータ作成完了する）、または、下位の演算が終了する。
（２）出力バッファが上限に達する（要求量のデータ作成完了する）、または、出力バッファが空である。
【００７０】
実体化型クエリ処理方式及び要求駆動型クエリ処理方式における演算処理部１３の処理手順について、図９のフローチャート図を参照して説明する。
クエリ実行制御部１２から演算の実行依頼を受け（ステップ９１）、演算の種類に応じて処理を開始する（ステップ９２）。
処理開始後、出力先のバッファ（演算の出力バッファ、もしくは制御バッファ）が上限に達するか、入力先のバッファ（演算の制御バッファ、もしくは直下の演算の出力バッファ）が空になった場合、演算処理を終了する（ステップ９３）。
その後、バッファ利用量収集部２３に、バッファ利用量収集部２３が収集している利用状況の情報を渡す（ステップ９４）。
【符号の説明】
【００７１】
１…クライアント、　２…ストレージ、　３…バッファ、　１０…クエリ処理装置、１１…クエリ計画部、　１２…クエリ実行制御部、　１３…演算処理部、　２０…バッファ管理装置、　２１…バッファ計画部、　２２…バッファ制御部、　２３…バッファ利用量収集部、　２５…演算の実装情報、　２６…利用可能メモリ、　２７…バッファ情報。

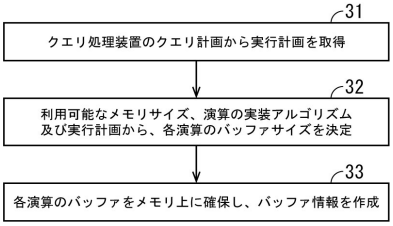
【図１】



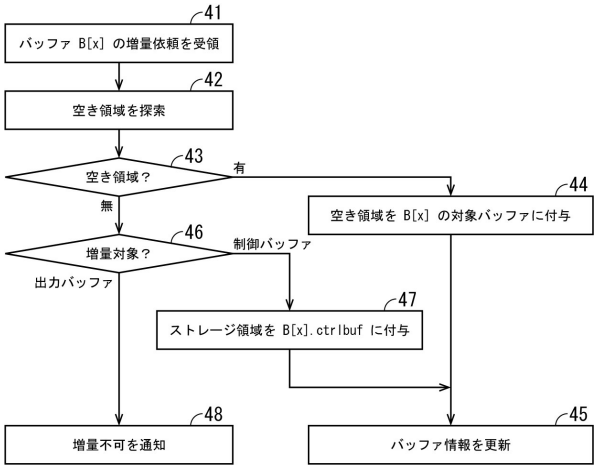
【図２】



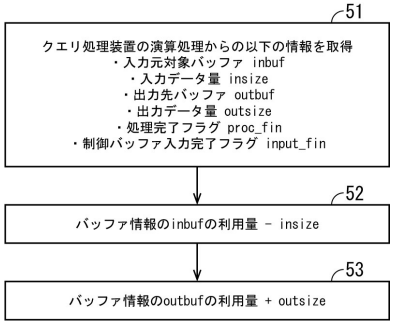
【図３】



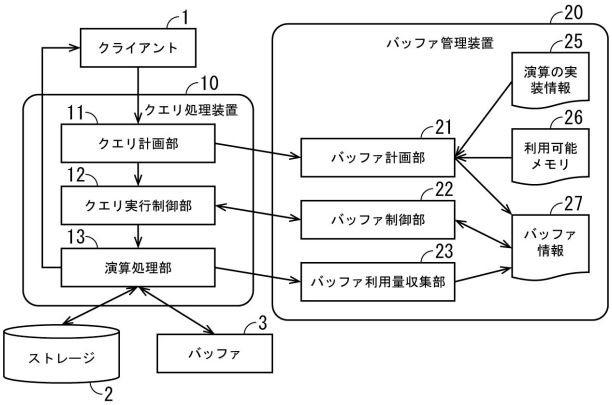
【図４】



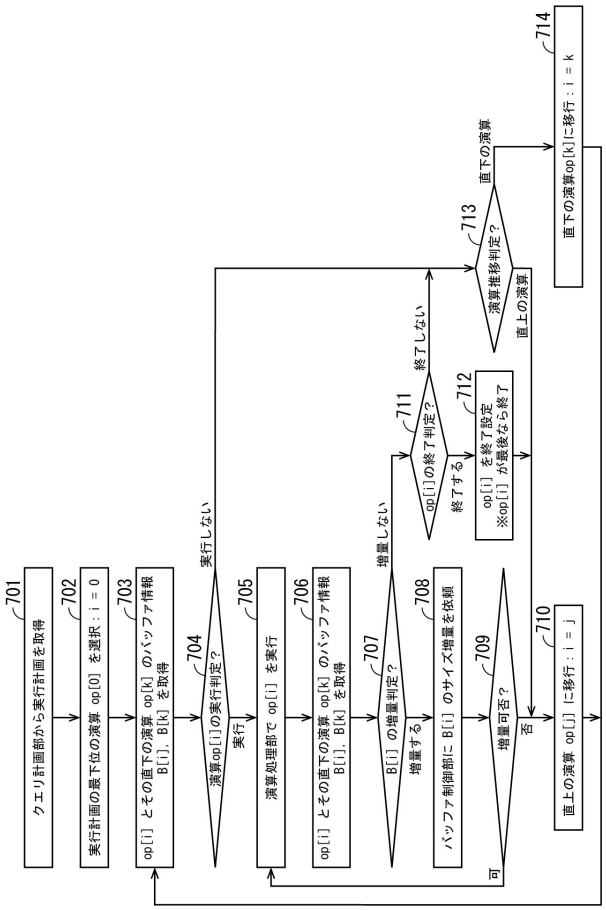
【図５】



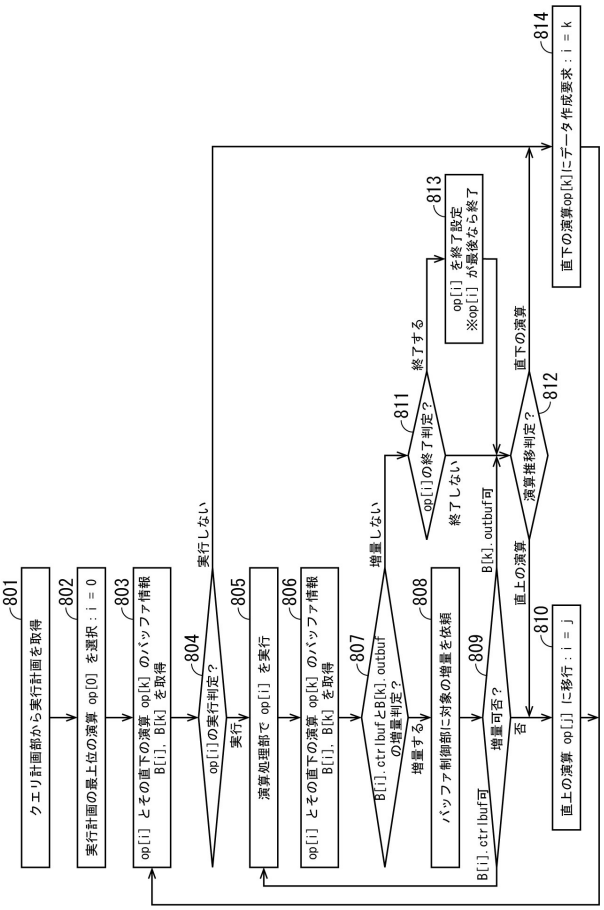
【図６】



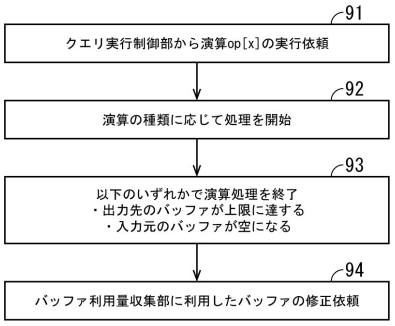
【図７】


701
クエリ計画部から実行計画を取得
702
実行計画の最下位の演算 op[0] を選択：i = 0
703
op[i] とその直下の演算 op[k] のバッファ情報 B[i]，B[k] を取得
704
演算op[i]の実行判定？
実行
実行しない
705
演算処理部で op[i] を実行
706
op[i] とその直下の演算 op[k] のバッファ情報 B[i]，B[k] を取得
707
B[i] の増量判定？
増量する
増量しない
708
バッファ制御部に B[i] のサイズ増量を依頼
709
増量可否？
可
否
710
直上の演算 op[j] に移行：i = j
711
op[i]の終了判定？
終了する
終了しない
712
op[i] を終了設定
※op[i] が最後なら終了
713
演算推移判定？
直上の演算
直下の演算
714
直下の演算op[k]に移行：i = k

【図８】


801
クエリ計画部から実行計画を取得
802
実行計画の最上位の演算 op[0] を選択：i = 0
803
op[i] とその直下の演算 op[k] のバッファ情報 B[i]，B[k] を取得
804
op[i]の実行判定？
実行
実行しない
805
演算処理部で op[i] を実行
806
op[i] とその直下の演算 op[k] のバッファ情報 B[i]，B[k] を取得
807
B[i].ctrlbufとB[k].outbuf の増量判定？
増量する
増量しない
808
バッファ制御部に対象の増量を依頼
809
増量可否？
B[i].ctrlbuf可
B[k].outbuf可
否
810
直上の演算 op[j] に移行：i = j
811
op[i]の終了判定？
終了する
終了しない
812
演算推移判定？
直上の演算
直下の演算
813
op[i] を終了設定
※op[i] が最後なら終了
814
直下の演算op[k]にデータ作成要求：i = k

【図９】


91
クエリ実行制御部から演算op[x]の実行依頼
92
演算の種類に応じて処理を開始
93
以下のいずれかで演算処理を終了
・出力先のバッファが上限に達する
・入力元のバッファが空になる
94
バッファ利用量収集部に利用したバッファの修正依頼

フロントページの続き

(56)参考文献 特開平０９－３０５６１４（ＪＰ，Ａ）
米国特許第９１０４６６３（ＵＳ，Ｂ１）
米国特許出願公開第２００８／０１４７５９９（ＵＳ，Ａ１）
国際公開第２０１３／１６１０８１（WO，Ａ１）
大西　元、他，“データベースの多様な応用分野を対象とする並列処理システムＳＭＡＳＨ－メモリ資源割り当ての計算方式の実現－”，第３８回（平成元年前期）全国大会講演論文集（ＩＩ），社団法人情報処理学会，１９８９年　3月１７日，ｐｐ．９６６－９６７

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１６／００－１６／９５８
Ｇ０６Ｆ　１２／００