

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5308403号  
(P5308403)

(45) 発行日 平成25年10月9日(2013.10.9)

(24) 登録日 平成25年7月5日(2013.7.5)

(51) Int.Cl. F 1  
G 0 6 F 1 2 / 0 0 ( 2 0 0 6 . 0 1 ) G 0 6 F 1 2 / 0 0 5 3 1 R

請求項の数 15 (全 20 頁)

(21) 出願番号	特願2010-136099 (P2010-136099)	(73) 特許権者	000005108 株式会社日立製作所 東京都千代田区丸の内一丁目6番6号
(22) 出願日	平成22年6月15日(2010.6.15)	(74) 代理人	110000350 ポレール特許業務法人
(65) 公開番号	特開2012-3394 (P2012-3394A)	(72) 発明者	櫻井 隆雄 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内
(43) 公開日	平成24年1月5日(2012.1.5)	(72) 発明者	恵木 正史 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内
審査請求日	平成25年1月8日(2013.1.8)	(72) 発明者	今木 常之 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内

最終頁に続く

(54) 【発明の名称】 データ処理の障害回復方法、システムおよびプログラム

(57) 【特許請求の範囲】

【請求項1】

計算機を用いたストリームデータ処理の障害回復方法であって、  
前記計算機は、

ストリームデータ処理を構成するオペレータ中、実行状態を保持するオペレータ各々の回復ポイントに基づき、当該回復ポイントより以降の回復ポイントを持つ前記実行状態を保持するオペレータの、最古の時刻からのストリームデータの容量と、当該回復ポイントより前の回復ポイントを持つ前記実行状態を保持するオペレータの複製データの容量を取得し、前記ストリームデータの容量と前記複製データの容量の合計値が最少となる前記回復ポイントを算出し、算出した前記回復ポイントにおいて前記ストリームデータと前記複製データを記録する、

10

ことを特徴とするストリームデータ処理の障害回復方法。

【請求項2】

請求項1に記載のデータ処理の障害回復方法であって、  
前記容量の指標が前記ストリームデータのデータ数である、  
ことを特徴とするデータ処理の障害回復方法。

【請求項3】

請求項1に記載のデータ処理の障害回復方法であって、  
前記計算機は、

前記実行状態の記録を、任意の時間に実行する、一定時間ごとに実行する、あるいは前回

20

の記録から一定量の入力データが与えられたときに実行する、  
ことを特徴とするデータ処理の障害回復方法。

【請求項 4】

請求項 1 に記載のデータ処理の障害回復方法であって、  
前記実行状態を保持するオペレータが、時間ウィンドウ、個数ウィンドウ、あるいは永続  
ウィンドウである、  
ことを特徴とするデータ処理の障害回復方法。

【請求項 5】

請求項 1 に記載のデータ処理の障害回復方法であって、  
前記計算機は、  
障害回復のための実行状態再現時において、算出した前記回復ポイントから前記ストリー  
ムデータを流し込み、その後、前記複製データを記録した、前記実行状態を保持するオペ  
レータに前記複製データを上書きし、その後、バックアップデータ取得後のストリームデ  
ータ処理を行う、  
ことを特徴とするデータ処理の障害回復方法。

10

【請求項 6】

処理部と記憶部とを備えた計算機により実行されるストリームデータ処理の障害回復シス  
テムであって、  
前記計算機の処理部は、  
クエリに対応するストリームデータ処理を行うオペレータ中、実行状態を保持するオペレ  
ータと、回復ポイントを解析するクエリ解析部と、  
前記クエリ解析部が解析した、各々の前記回復ポイントに対し、当該回復ポイントより以  
降の回復ポイントを持つ実行状態を保持するオペレータの最古の時刻からのストリームデ  
ータの容量と、当該回復ポイントより前の回復ポイントを持つ実行状態を保持するオペレ  
ータの複製データの容量を取得し、前記回復ポイント各々における、前記ストリームデー  
タの容量と、前記複製データの容量との合計値が最少となる回復ポイントを決定するバッ  
クアップデータ管理部とを備え、  
前記バックアップデータ管理部が決定した回復ポイントにおいてストリームデータ処理の  
実行状態を前記記憶部に記録する、  
ことを特徴とする障害回復システム。

20

30

【請求項 7】

請求項 6 に記載のデータ処理の障害回復システムであって、  
前記容量の指標が前記ストリームデータのデータ数である、  
ことを特徴とするデータ処理の障害回復システム。

【請求項 8】

請求項 6 に記載のデータ処理の障害回復システムであって、  
前記処理部は、  
前記実行状態の記録を、任意の時間に実行する、一定時間ごとに実行する、あるいは前回  
の記録から一定量の入力データが与えられたときに実行する、  
ことを特徴とするデータ処理の障害回復システム。

40

【請求項 9】

請求項 6 に記載のデータ処理の障害回復システムであって、  
前記実行状態を保持するオペレータが、時間ウィンドウ、個数ウィンドウ、あるいは永続  
ウィンドウである、  
ことを特徴とするデータ処理の障害回復システム。

【請求項 10】

請求項 6 に記載のデータ処理の障害回復システムであって、  
前記処理部は、  
障害回復のための実行状態再現時において、算出した前記回復ポイントから前記ストリー  
ムデータを流し込み、その後、前記複製データを記録した、前記実行状態を保持するオペ  
レータが、時間ウィンドウ、個数ウィンドウ、あるいは永続ウィンドウである、  
ことを特徴とするデータ処理の障害回復システム。

50

レータに前記複製データを上書きし、その後、バックアップデータ取得後のストリームデータ処理を行う、  
ことを特徴とするデータ処理の障害回復システム。

【請求項 1 1】

クエリに基づきストリームデータ処理を実行する計算機の処理部で実行されるデータ処理の障害回復プログラムであって、

前記処理部を、

クエリに対応するストリームデータ処理を行うオペレータ中、実行状態を保持するオペレータと、回復ポイントを解析し、

解析した前記回復ポイント各々に対し、当該回復ポイントより以降の回復ポイントを持つ実行状態を保持するオペレータの最古の時刻からのストリームデータの容量と、当該回復ポイントより前の回復ポイントを持つ実行状態を保持するオペレータの複製データの容量を取得し、

前記回復ポイント各々における、前記ストリームデータの容量と、前記複製データの容量との合計値が最少となる回復ポイントを決定し、

決定した回復ポイントにおいてストリームデータ処理の実行状態を記録する、

よう動作させる、

ことを特徴とするデータ処理の障害回復プログラム。

【請求項 1 2】

請求項 1 1 に記載のデータ処理の障害回復プログラムであって、

前記容量の指標が前記ストリームデータのデータ数である、

ことを特徴とするデータ処理の障害回復プログラム。

【請求項 1 3】

請求項 1 1 に記載のデータ処理の障害回復プログラムであって、

前記処理部を、

前記実行状態の記録を、任意の時間に実行する、一定時間ごとに実行する、あるいは前回の記録から一定量の入力データが与えられたときに実行させる、

よう動作させる、

ことを特徴とするデータ処理の障害回復プログラム。

【請求項 1 4】

請求項 1 1 に記載のデータ処理の障害回復プログラムであって、

前記実行状態を保持するオペレータが、時間ウィンドウ、個数ウィンドウ、あるいは永続ウィンドウである、

ことを特徴とするデータ処理の障害回復プログラム。

【請求項 1 5】

請求項 1 1 に記載のデータ処理の障害回復プログラムであって、

前記処理部を、

障害回復のための実行状態再現時において、算出した前記回復ポイントから前記ストリームデータを流し込み、その後、前記複製データを記録した、前記実行状態を保持するオペレータに前記複製データを上書きし、その後、バックアップデータ取得後のストリームデータ処理を行う、

よう動作させることを特徴とするデータ処理の障害回復プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データ処理の障害回復技術に関し、特に、ストリームデータ処理における障害回復に必要な再現データの保存技術に関する。

【背景技術】

【0002】

自動株取引、高度な交通情報処理、多地点から得たセンサ情報の解析といった、継続的

10

20

30

40

50

に発生する多量のデータをリアルタイムに解析し即座に対応ために、ストリームデータ処理が注目されている。ストリームデータ処理は、様々な形式のデータのリアルタイム処理に適用可能な汎用ミドルウェア技術であるため、個別案件ごとにシステムを構築するのは間に合わないようなビジネス環境の急激な変化にも応えつつ、実世界のデータをリアルタイムにビジネスに反映することを可能とする。このストリームデータ処理の原理、実現方式は非特許文献 1 に開示されている。

【 0 0 0 3 】

ストリームデータ処理は、前述のように多量のデータのリアルタイム処理であるため、処理結果の出力データも多量かつ継続的に発生することになる。従って、障害が発生してから復旧までに要する時間は、可能な限り短くすることが求められる。このとき、復旧されたサーバの実行状態は初期状態であるため、障害発生前の実行状態を復旧後のサーバにも再現する、実行状態再現が必要とされている。

10

【 0 0 0 4 】

実行状態再現の一つ目の方法として、正常動作中から入力ストリームをバックアップしておき、復旧時にはバックアップデータを待機系サーバで再実行して現用系サーバの実行状態に追付かせる、Upstream Backup方式が非特許文献 2 に開示されている。処理時間が長くなるほど、バックアップに必要なディスクやメモリなどの記憶容量は増大するが、次の理由で容量は一定以内に収まることが仮定できる。

【 0 0 0 5 】

ストリームデータ処理では、データ系列から直近の一部分を切り出すウィンドウ演算を利用することが可能である。ウィンドウ演算の定義は非特許文献 3 に開示されている。例えば、時間幅 1 分のウィンドウ演算によって切り出したデータに対して平均を算出する集約演算を適用すると、1 分間の移動平均を算出する動作となる。この例においては、1 分間データを流し続けるとウィンドウ内のデータが刷新されることになるため、初期状態から開始する復旧時においても直近 1 分間のデータを処理することで、障害発生前と同じ実行状態になる。このように、Upstream Backup方式においては、保持しておくべきデータの範囲が処理の進行に伴って未来に進むことを前提とすることで、バックアップのための記憶容量が一定以内に収まることが仮定できる。

20

【 0 0 0 6 】

実行状態再現の二つ目の方法として、次のようなものが存在する。まず、運用中のサーバを定期的に一時停止して実行状態を静止化し、その実行状態を複製（スナップショット）として保存する。そして、障害が発生し、復旧した時に保存したスナップショットから実行状態を再現する。静止化してスナップショットを保存する方法は、データベースやトランザクションシステムで広く利用されている方法である。インメモリデータベースにおける静止化を利用した再現方法が、特許文献 1 に開示されている。

30

【 先行技術文献 】

【 特許文献 】

【 0 0 0 7 】

【 特許文献 1 】 特開 2 0 0 9 - 1 5 7 7 8 5 号公報

【 非特許文献 】

40

【 0 0 0 8 】

【 非特許文献 1 】 B . B a b c o c k , S . B a b u , M . D a t a r , R . M o t w a n i a n d J . W i d o m , “ M o d e l s a n d i s s u e s i n d a t a s t r e a m s y s t e m s ” , I n P r o c . o f P O D S 2 0 0 2 , p p . 1 - 1 6 . ( 2 0 0 2 )

【 非特許文献 2 】 J . H . H w a n g , M . B a l a z i n s k a , A . R a s i n , U . C e t i n t e m e l , M . S t o n e b r a k e r a n d S . B . Z d o n i k , “ H i g h - A v a i l a b i l i t y A l g o r i t h m s f o r D i s t r i b u t e d S t r e a m P r o c e s s i n g ” , I n P r o c . o f I C D E 2 0 0 5 , p p . 7 7 9 - 7 9 0 . ( 2 0 0 5 )

50

【非特許文献3】A. Arasu, S. Babu and J. Widom. "The CQL Continuous Query Language: Semantic Foundations and Query Execution", (2005)

【発明の概要】

【発明が解決しようとする課題】

【0009】

前述のUpstream Backup方式による実行状態再現において次のような問題がある。ストリームデータ処理システムが処理するウィンドウ演算としては、前述の時間ウィンドウ(Rangeウィンドウ)以外にも、個数ウィンドウ(Rowsウィンドウ)、グループ別個数ウィンドウ(Partitionウィンドウ)、永続ウィンドウ(Unboundedウィンドウ)などが存在する。時間ウィンドウとは異なり、これらのウィンドウでは時間の経過のみではウィンドウが刷新されない可能性がある。例えば、証券取引の分析において銘柄毎に直近100件の出来高統計を算出する処理は、グループ別個数ウィンドウの利用により容易に定義できる。このとき、取引が低調な銘柄が存在すると、その銘柄の取引データがウィンドウに残り続けることになる。また、分析開始から全取引の集計を算出するといった処理は、永続ウィンドウを利用することで容易に定義できるが、同ウィンドウには処理開始以降の全てのデータが残り、全く刷新されない。

10

【0010】

このようなケースにUpstream Backup方式を適用すると、保持しておくべきデータ範囲の起点が進行しないため、データの保持に必要な記憶容量が際限なく増大し、いずれオーバーフローすることになる。

20

【0011】

一方で、スナップショットを利用する実行状態再現方式では、全てのウィンドウ演算を利用可能である。但し、動作中のサーバを静止化する期間、結果の出力が停止するため、アプリケーションに対して処理の停止として影響を与えてしまうことになる。実行状態に「過去数分間に送られた全データ」といった非常にサイズの大きなものが複数含まれていた場合、スナップショットの取得に非常に大きな記憶容量を必要とする。

【0012】

本発明の解決すべき課題は、ストリームデータ処理の実行状態再現において、バックアップデータ取得に必要な記憶容量を最小限にとどめた上で、時間ウィンドウに限らず全てのウィンドウ演算の利用を実現することである。

30

【0013】

すなわち、本発明の目的は、上記の課題を解決できるデータ処理障害回復方法、システムおよびプログラムを提供することにある。

【課題を解決するための手段】

【0014】

上記の目的を達成するため、本発明においては、計算機を用いたストリームデータ処理の障害回復方法であって、計算機は、ストリームデータ処理を構成するオペレータ中、実行状態を保持するオペレータ各々の回復ポイントに基づき、当該回復ポイントより以降の回復ポイントを持つ実行状態を保持するオペレータの最古の時刻からのストリームデータの容量と、当該回復ポイントより前の回復ポイントを持つ実行状態を保持するオペレータの複製データの容量を取得しストリームデータの容量と複製データの容量の合計値が最少となる回復ポイントを算出し、算出した回復ポイントにおいてストリームデータと複製データを記録するストリームデータ処理の障害回復方法を提供する。

40

【0015】

また、上記の目的を達成するため、本発明においては、処理部と記憶部とを備えた計算機により実行されるストリームデータ処理の障害回復システムであって、計算機の処理部は、クエリに対応するストリームデータ処理を行うオペレータ中、実行状態を保持するオペレータと、その回復ポイントを解析するクエリ解析部と、クエリ解析部が解析した、各々の回復ポイントに基づき、当該回復ポイントより以降の回復ポイントを持つ実行状態を

50

保持するオペレータの最古の時刻からのストリームデータの容量と、当該回復ポイントより前の回復ポイントを持つ実行状態を保持するオペレータの複製データの容量を取得し、各回復ポイントにおける、ストリームデータの容量と複製データの容量との合計値が最少となる回復ポイントを決定するバックアップデータ管理部とを備え、決定した回復ポイントにおいてストリームデータ処理の実行状態を記憶部に記録する障害回復システムを提供する。

【0016】

更に、上記の目的を達成するため、本発明においては、クエリに基づきストリームデータ処理を実行する計算機の処理部で実行される障害回復プログラムであって、処理部を、クエリに対応するストリームデータ処理を行うオペレータ中、実行状態を保持するオペレータと、その回復ポイントを解析し、解析した、各々の回復ポイントに基づき、当該回復ポイントより以降の回復ポイントを持つ実行状態を保持するオペレータの最古の時刻からのストリームデータの容量と、当該回復ポイントより前の回復ポイントを持つ実行状態を保持するオペレータの複製データの容量を取得し、各回復ポイントにおける、ストリームデータの容量と複製データの容量との合計値が最少となる回復ポイントを決定し、決定した回復ポイントにおいてストリームデータ処理の実行状態を記録するよう動作させる障害回復プログラムを提供する。

10

【0017】

また更に、本発明の好適なデータ処理の障害回復方式においては、前述の課題を解決するために以下の手順で実行状態を再現する。

20

【0018】

(1) ストリームデータ処理の中に含まれる全てのウィンドウ等の実行状態を保持するオペレータは、時間・個数・グループ別などの種類を問わず、それぞれが現在の状態を再現するために必要な最も古いデータが入力された時刻を `Upstream Backup` 方式で再現可能な回復ポイントとして管理する。

【0019】

(2) 全てのウィンドウ等の実行状態を保持するオペレータの回復ポイント各々について、その回復ポイントより以降の回復ポイントを持つウィンドウ等の実行状態を保持するオペレータについては、バックアップデータを保持する `Upstream Backup` 方式、その回復ポイントより前の回復ポイントを持つウィンドウ等の実行状態を保持するオペレータについては複製(スナップショット)を取得する方式で、実行状態を再現するために必要な記憶領域の大きさを計算し管理する。

30

【0020】

(3) 計算した全ての回復ポイントにおける実行状態再現に必要な記憶領域の総和の中で、容量がもっとも小さい回復ポイントを選択する。そして、その回復ポイント以降のストリームデータのバックアップデータを保持すると同時に、その回復ポイントより前の回復ポイントを持つウィンドウの複製(スナップショット)を取得する。

【0021】

(4) 障害回復のための実行状態再現時において、まず当該回復ポイントからデータを流し込み、その部分の処理が終わったら複製(スナップショット)のあるウィンドウはスナップショットからデータを上書きし、その後バックアップデータ取得後のストリームの処理を始める。

40

【発明の効果】

【0022】

本発明により、ストリームデータ処理の実行状態再現において、バックアップデータ取得に必要な記憶容量を最小限にとどめた上で、時間ウィンドウに限らず全ての実行状態を保持するオペレータが利用可能となる。より具体的には、実行状態を保持するオペレータごとにスナップショットを取得すべきか `Upstream Backup` 方式により再現するかを比較し、より記憶領域が小さくなる方を選択することが可能となる。

【図面の簡単な説明】

50

## 【 0 0 2 3 】

【図 1】第 1 の実施例のストリームデータ処理サーバが利用される計算機環境の構成を示す図である。

【図 2】第 1 の実施例のストリームデータ処理サーバの構成の一例を示す図である。

【図 3】第 1 の実施例に係る、データ処理定義の一例を示す図である。

【図 4】図 3 に示すデータ処理定義をクエリグラフに変換した結果を示す図である。

【図 5】第 1 の実施例に係る、図 4 に示すクエリグラフの例における、実行状態の例を示す図である。

【図 6】第 1 の実施例に係る、ストリームデータ処理における実行状態記録方式の例を示す図である。

10

【図 7】第 1 の実施例に係る、バックアップ要求がされた際の動作を示すフローチャートを示す図である。

【図 8】第 1 の実施例に係る、スナップショット対象の選定がされる際の動作を示すフローチャートを示す図である。

【図 9】第 1 の実施例に係る、バックアップデータ取得時刻における各オペレータの実行状態と記憶量、回復ポイントを例示する図である。

【図 10】第 1 の実施例に係る、ストリームデータ処理システム起動直後からバックアップデータ取得時刻までの入力データと各オペレータの回復ポイント時のデータ量を例示する図である。

【図 11】第 1 の実施例に係る、各オペレータの回復ポイント選択時のバックアップに必要な記憶容量の一覧を例示する図である。

20

【図 12】第 1 の実施例に係る、選択された回復ポイントと入力データから実行状態を再現するオペレータとスナップショットから実行状態を再現するオペレータのリストを例示する図である。

【図 13 A】第 1 の実施例に係る、再現用のバックアップデータを例示する図である。

【図 13 B】第 1 の実施例に係る、再現用のバックアップデータを例示する図である。

【図 14】第 1 の実施例に係る、ストリームデータ処理システムにより復旧要求がされた場合の動作を示すフローチャートを示す図である。

【図 15】第 1 の実施例に係る、復旧要求時にストリームデータ処理システムの実行状態をバックアップデータから再現する動作を示すフローチャートを示す図である。

30

【図 16】第 1 の実施例に係る、初期状態のストリームデータ処理システムに対して入力データのバックアップを処理させる動作を例示する図である。

【図 17】第 1 の実施例に係る、入力データのバックアップを処理した後の実行状態を例示する図である。

【図 18】第 1 の実施例に係る、入力データのバックアップ後にスナップショットをコピーする動作を例示する図である。

【図 19】第 1 の実施例に係る、バックアップデータ取得におけるパラメータを設定する GUI を例示する図である。

## 【発明を実施するための形態】

## 【 0 0 2 4 】

40

以下、本発明の実施の形態を図面に基づいて詳細に説明する。なお、実施の形態を説明するための全図において、同一の部材には原則として同一の符号を付し、その繰り返しの説明は省略する。また、後で説明するように、本明細書において、オペレータには、Scan オペレータ、フィルタオペレータ等に加え、各種のウィンドウ演算も含まるので、留意されたい。

## 【実施例 1】

## 【 0 0 2 5 】

まず、図 1 および図 2 を用いて、第 1 の実施例に係る、ストリームデータ処理システムの基本構成を説明する。

## 【 0 0 2 6 】

50

図1に示すように、ネットワーク104にストリームデータ処理サーバ100と計算機101、102、103が接続されている。ストリームデータ処理サーバ100は、ネットワーク104を介して、データソース107が動作する計算機102からデータ108を受け取り、処理結果のデータ110を計算機103上の結果利用アプリケーション109に送信する。また、計算機101上では、クエリ登録コマンド実行インタフェース105が動作する。

【0027】

図2に示すように、ストリームデータ処理サーバ100は、計算機200および210から構成され、計算機200および210は、記憶部であるメモリ202および212、処理部である中央処理部(Central Processing Unit: CPU)201および211、ネットワークインタフェース(Interface: I/F)204および214、記憶部であるストレージ203および213、およびそれらを結合するバス205および215によって構成される。メモリ202上に、ストリームデータ処理の論理動作を定義する、ストリームデータ処理システム206を配置する。ストリームデータ処理システム206は、後で詳述するようにCPU201によって解釈実行可能な実行イメージである。

10

【0028】

図2に示すように、ストリームデータ処理サーバ100を構成する計算機200および210は、ネットワークI/F204および214を介して外部のネットワーク104に接続される。

20

【0029】

ネットワーク104に接続された計算機101上で動作する、クエリ登録コマンド実行インタフェース105を介して、ユーザによって定義されたクエリ106を、ストリームデータ処理サーバ100を構成する計算機200が受取ると、ストリームデータ処理システム206は、この定義に従ってストリームデータ処理を実行可能なクエリグラフを自身の内部に構成する。この後、ネットワーク104に接続された計算機102上で動作するデータソース107によって送信されるデータ108を、ストリームデータ処理サーバ100を構成する計算機200が受取ると、このクエリグラフに従って処理し、結果データ110を生成し、計算機103上で動作する結果利用アプリケーション109に送信する。ストレージ203は、ストリームデータ処理システム206の他、一度受取ったクエリ106を保存する。ストリームデータ処理システム206は、起動時にストレージ203からこの定義をロードし、クエリグラフを構成することも可能である。

30

【0030】

計算機210を構成するメモリ212には、ストリームデータ処理システム206に不具合が発生した際の復旧用にバックアップ用ストレージシステム(BSS)216が記憶されている。また、計算機210を構成するメモリ212およびストレージ213のいずれかもしくは双方は、ストリームデータ処理システム206に不具合が発生した際に復旧させるために必要な再現用データ217および218を保持している。

【0031】

なお、ここで説明した本実施例のストリームデータ処理サーバの構成は一例であり、計算機200と210は一台の計算機であって、処理部であるCPU201および211は、同一計算機上の二つのプロセッサであっても構わない。あるいは、一つのマルチコアCPUにおける二つの計算コアであっても構わない。また、メモリ202および212、ネットワークI/F204および214、ストレージ203および213は、それぞれが一つであって、一つの計算機に接続されるのであっても、あるいは二つの計算機に接続されて共有されるのであっても構わない。本明細書において、計算機とはいずれの場合も含み、処理部、更に記憶部も同様である。

40

【0032】

次に、図3および図4を用いて、本実施例のストリームデータ処理におけるクエリとクエリグラフの一例を説明する。

50



## 【 0 0 3 3 】

図3に示すように、クエリ300は、2つの入力ストリーム  $s_a$  および  $s_b$ 、3つのクエリ  $q_1$ 、 $q_2$  および  $q_3$  を定義するクエリである。

## 【 0 0 3 4 】

図4に示す通り、ストリームデータ処理システムは、クエリ300の定義を受取ると、自身の実行領域中に確保したクエリ実行ワークエリア420上に、オペレータ400～410によって構成される、クエリグラフを生成する。このオペレータには、スキャン (Scan) オペレータ400、403、フィルタ (Filter) オペレータ402、405、結合オペレータ406、ストリーム化演算オペレータ407などに加え、各種のウィンドウ (Window) 401、404、408等も含まれる。オペレータ400は入力ストリーム  $s_a$  をデータソースから受取るScanオペレータ、オペレータ403は入力ストリーム  $s_b$  をデータソースから受取るScanオペレータである。ストリーム  $s_a$  および  $s_b$  は共に、文字列型のカラム  $id$  と、整数型のカラム  $val$  の二つのカラムから構成されるデータの系列である。

## 【 0 0 3 5 】

オペレータ401、402、404、405、406および407は、クエリ  $q_1$  に対応する部分クエリグラフを構成するオペレータ群である。オペレータ401は、ストリーム  $s_a$  に対して施されるグループ別個数ウィンドウ (PARTITION BY  $id$  ROWS 2) であり、カラム  $id$  別に最新2個のデータを切り出す。オペレータ404は、ストリーム  $s_b$  に対して施される時間ウィンドウ (RANGE 5 MINUTES) であり、直近5分以内のデータを切り出す。オペレータ402は、ウィンドウ401で切り出したデータに対して施されるフィルタオペレータ ( $s_a.val > 100$ ) であり、カラム  $val$  の値が100より大きいデータのみを通過させる。オペレータ405は、ウィンドウ404で切り出したデータに対して施されるフィルタオペレータ ( $s_b.val < -1$ ) であり、カラム  $val$  の値が-1以外のデータを通過させる。オペレータ406は、結合オペレータ ( $s_a.id = s_b.id$ ) であり、オペレータ402および405を通過したデータにおいて、カラム  $id$  が一致する組合せを生成する。オペレータ407は、クエリの結果を正規化するストリーム化演算である。

## 【 0 0 3 6 】

オペレータ408および409は、クエリ  $q_2$  に対応する部分クエリグラフを構成するオペレータ群である。オペレータ408は、永続ウィンドウ (UNBOUNDED) であり、クエリ  $q_1$  の結果データを全て保持する。オペレータ409は集約オペレータであり、カラム  $id$  別に  $s_a.val$  と  $s_b.val$  の最大値を算出する。また、オペレータ410は、クエリ  $q_3$  に対応する部分クエリグラフを構成するストリーム化演算オペレータである。

## 【 0 0 3 7 】

一時保持領域 (Temporal Store) 411および412は、それぞれ結合オペレータ406および集約オペレータ409の実行状態を保持する領域である。一時保持領域411は、オペレータ406の左入力と右入力それぞれにおける、生存中のデータを保持する。これらは、反対側の入力に到来したデータの結合相手となる。一時保持領域412は、グループ別に集約結果のデータを一つずつ保持する。

## 【 0 0 3 8 】

前述したように、一時保存領域を持つ結合オペレータ、集約オペレータ以外に、ウィンドウ演算も、実行状態を保持するオペレータである。ウィンドウ演算は、個々の入力データに対して生存期間を定義し、生存中のデータを保持する。これら以外の、フィルタオペレータ、射影オペレータ、ストリーム化演算、Scanオペレータ等のオペレータについては、実行状態を保持する必要はない。

## 【 0 0 3 9 】

次に、図5を用いて、図4のクエリグラフの例における実行状態の一例を説明する。ウィンドウ演算  $W1$  401にデータ501～506を保持し、ウィンドウ演算  $W2$  404

10

20

30

40

50

にデータ511～517を保持している状態を表している。各データの長楕円はデータのタイムスタンプを表し、左側の四角はカラムidの値を、右側の四角はカラムvalの値を表している。グループ別ウィンドウ401は、カラムid別に、最大2個のデータを保持している。時間ウィンドウ404は、タイムスタンプが9:55～9:59までのデータを保持している。

#### 【0040】

一時保持領域W3411は、左入力における生存中のデータ501、503、504、505、および右入力における生存中のデータ512、513、514、516、517を保持している。それぞれ、ウィンドウ演算401に保持しているデータ集合のうち、フィルタ条件 $sa.val > 100$ を満たすデータの集合、およびウィンドウ演算404に保持しているデータ集合のうち、フィルタ条件 $sb.val < -1$ を満たすデータの集合である。また、結合条件がカラムidに関する等号条件であるため、カラムidの値をキーとして索引付けしており、カラムidの値別にグループ分けして保持している。

10

#### 【0041】

ウィンドウ演算W4408は、一時保持領域411に保持する、左入力のデータ集合と右入力のデータ集合の直積において、結合条件 $sa.id = sb.id$ を満たす組合せデータ521～531を保持している。これらのデータのタイムスタンプは、組合せた左右データのうち遅い方のタイムスタンプをとる。ウィンドウ演算408は永続ウィンドウであるため、処理を開始した時刻から全てのデータを保持している。そのため、組合せデータ521のように非常に古いデータもウィンドウ内に存在する。

20

#### 【0042】

一時保持領域W5412は、ウィンドウ演算408に保持しているデータをカラムid別にグループ分けして集約したデータを、各グループにつき一つずつ保持している。カラムidがa、bおよびcそれぞれについて、データ541、542、および543を保持している。ここで一時保持領域W5412には、カラムid別に各グループの平均値、最大値、最小値等を保持するよう設定することが可能である。図5の場合、一時保持領域W5412には最大値が保持されるよう設定されている。

#### 【0043】

続いて、図6を用いて本実施例のストリームデータ処理を実現するソフトウェアのブロック構成の一例を説明する。なお、同図において、太線のブロックはCPUで実行される各種のソフトウェア機能を、細線のブロックはソフトウェアの実行の際、メモリ上に形成される各種のデータの保存領域を模式的に示している。

30

#### 【0044】

同図において、ストリームデータ処理システム206は、それぞれ、入力データ108を受信する入力データ受信部601、クエリグラフとオペレータの実行状態を保持するクエリ実行ワークエリア420、クエリ実行ワークエリア420のデータに基づいてクエリを実行するクエリ実行部602、クエリ実行結果110を出力する出力データ送信部605を備える。クエリ実行ワークエリア420には、それぞれ、オペレータ毎の実行状態を保持するオペレータ実行状態保持領域621～623および各オペレータ実行状態保持領域621～623に対して各オペレータにおいてその内部状態に使用されている最古の入力データの時刻を示す回復ポイントとそれらをスナップショットとして記録したときの記憶量を記録したオペレータ回復ポイント記憶領域624～626を確保する。

40

#### 【0045】

さらに、ストリームデータ処理システム206は、クエリ106を解析してクエリ実行ワークエリア上にクエリグラフを生成するクエリ解析部606を備える。クエリ解析部606は、クエリグラフ上のオペレータ群において、実行状態のスナップショットを取得するオペレータを選定する、スナップショット対象選定部607を含む。スナップショット対象選定部607で選定したオペレータ群は、スナップショット対象リスト記憶領域608に記憶する。

#### 【0046】

50

加えて、ストリームデータ処理システム206は、入力データ受信部601で受信した入力データ108の複製をバックアップ用ストレージシステム216に送信する、もしくはバックアップ用ストレージシステム216から送られた復旧用の複製入力データを受信し入力データ受信部601に送信する複製データ通信部609、復旧用のデータをバックアップ用ストレージシステム216から送信するよう要求する復旧要求送信部610、バックアップ用ストレージシステム216から送信されたバックアップ要求を受信するバックアップ通知受信部611、オペレータの実行状態とスナップショット対象リストを一時的に保存するコピーバッファ領域612、バックアップ用ストレージシステム216に対しオペレータの実行状態およびスナップショット対象リストを送受信するワークエリアデータ通信部613を備える。

10

**【0047】**

ここで、クエリ実行部602は、各オペレータ実行状態保持領域621～623の保持内容をスナップショット対象リスト記憶領域608に従いコピーバッファ領域612にコピーする実行状態書出部603と、コピーバッファ領域612にある保持内容を各オペレータ実行状態保持領域621～623の保持内容にコピーする実行状態書込部604を備える。

**【0048】**

一方、バックアップ用ストレージシステム216はストレージデータ処理システム206と入力データ108の複製を授受する複製データ通信部657、ストレージデータ処理システム206から送られた復旧要求を受信する復旧要求受信部658、バックアップ処理をストレージデータ処理システム206に要求するバックアップ通知送信部659、オペレータの実行状態とスナップショット対象リストを一時的に保存するコピーバッファ領域660、ストレージデータ処理システム206に対しオペレータの実行状態およびスナップショット対象リストを送受信するワークエリアデータ通信部661を備える。

20

**【0049】**

さらに、バックアップ用ストレージシステム216は複製された入力データを保存しておく入力データ記憶領域655、スナップショットの対象リストを記憶するスナップショット対象リスト記憶領域656、スナップショットを記憶するスナップショット記憶領域654を備える。ここで、スナップショット記憶領域654はオペレータ実行状態記憶領域671～673を備える。

30

**【0050】**

加えて、バックアップ用ストレージシステム216はバックアップデータ管理部652を備える。バックアップデータ管理部652は入力データ記憶領域655の容量を監視する入力データ容量管理部653を備える。

**【0051】**

次に、図7、図8において、本実施例におけるバックアップ用データの更新処理フローの一例を示す。

**【0052】**

まず、図7はバックアップ用ストレージシステム216からバックアップ要求を送信し、バックアップ用データがストリームデータ処理システム206から送信され、バックアップ用ストレージシステム216の保持するバックアップ用データを更新する際のフローである。

40

**【0053】**

処理700では入力データ容量管理部653が「入力データ容量が規定値に達した」、「前のバックアップから一定時間が経過した」、等を理由にバックアップ要求をバックアップ通知送信部659に送信する。続いて処理701ではバックアップ通知送信部659がバックアップ要求をストリームデータ処理システム206に送信する。次いで処理702ではバックアップ通知受信部611でバックアップ要求を受信したストリームデータ処理システム206がスナップショット対象選定部607で、実行状態を保持するオペレータの中から、スナップショット対象のオペレータを選定する。処理703でストリームデ

50

ータ処理システム 206 が選定されたオペレータのスナップショットと回復ポイントデータをバックアップ用ストレージシステム 216 に送信する。最後に処理 704 ではバックアップ用ストレージシステム 216 でスナップショットを保存するとともに、送られた回復ポイント以前の複製された入力データを削除する。

【0054】

続いて、図 8 は上述の処理 702 の詳細である。まず、処理 800、801、812、813 でオペレータ通番 I が対象オペレータの数に達するまで処理 802 ~ 811 の処理を繰り返す。まず処理 816 で、オペレータ通番 I のオペレータが実行状態を保持しているかをチェックし、保持している場合、処理 802 ではオペレータ通番 I の回復ポイント I をオペレータ回復ポイント記憶領域から読み出す。続いて、処理 803 では回復ポイント I 以降の入力データの記憶容量を、入力データ容量管理部 653 に問い合わせそれを必要記憶容量 I の初期値とする。

10

【0055】

次いで、処理 804、805、810、811 でオペレータ通番 J が対象オペレータの数に達するまで処理 806 ~ 809 の処理を繰り返す。まず、処理 817 では、オペレータ通番 J が実行状態を保持しているかをチェックし、保持している場合、処理 806 でオペレータ通番 J の回復ポイント J をオペレータ回復ポイント記憶領域から読み出す。処理 807 でオペレータ通番 I の回復ポイント I とオペレータ通番 J の回復ポイント J を比較し、回復ポイント I の方が回復ポイント J より現在時刻に近い場合は処理 810 に進み、そうでない場合は処理 808 に進む。処理 808 ではオペレータ通番 J を回復ポイント I 選択時のスナップショット対象に指定する。続いて処理 809 ではオペレータ通番 J のスナップショットの記憶量を必要記憶量 I に加算する。全てのオペレータ通番 J に対して処理 806 ~ 809 を繰り返す。そして、これを全てのオペレータ通番 I に対して繰り返す。

20

【0056】

処理 814 において全てのオペレータ通番に対して最も小さい必要記憶容量を選択し、その回復ポイント K を決定する。続いて回復ポイント K 時のスナップショット対象をスナップショット対象リスト記憶領域 608 に記憶する。

【0057】

続いて図 9、図 10、図 11、図 12、図 13A、図 13B を用いて、本実施例におけるスナップショット対象の選定の具体的な動作例を示す。

30

【0058】

まず、図 9 は、図 4 で示した 400 ~ 412 で構成されるクエリグラフ、図 5 で示した各オペレータの持つウィンドウの実行状態をもとに、それぞれのウィンドウの実行状態にスナップショット取得時の記憶量と回復ポイントを加えて図示したものである。図 9 において、記憶量はストリームデータのデータ数を示しているが、これに限定するものでなく、各データを記憶するメモリの記憶容量などであって良いことはいうまでもない。

【0059】

この例では、ストリームデータ処理システムが時刻 6 : 30 から処理を実行し、現在時刻 9 50 が 10 : 00 のときにバックアップ処理を実施するものとする。このとき、ウィンドウ W1 401 においてデータは 501 ~ 506 の 6 つ存在し、最も時刻の古いデータは「時刻 9 : 48、ID = b、VAL = 97」のデータ 502 である。そのため、ウィンドウ W1 401 のスナップショットに必要な記憶量 901 は 6、回復ポイント 902 は 9 : 48 となる。同様に W2 404 の記憶量 911 は 6、回復ポイント 912 は 9 : 55、W3 411 の記憶量 921 は 9、回復ポイント 922 は 9 : 50 となる。W4 408 は永続ウィンドウであるためストリームデータ処理システムが起動してから W4 に送られたデータすべてを記録している。

40

【0060】

そのため、記憶量 931 は 100 と大きく、回復ポイント 932 も最古のデータである 521 と合わせて 6 : 30 と非常に前の時刻となっている。W5 412 ではそれぞれの

50

IDの最大値を記録しているため、記憶量941は3と小さいが、そのID = bの最大値データ542の由来となるデータは6:45に入力されたデータ522であるため、回復ポイント942は522と同じ6:45となる。このように各オペレータの持つウィンドウの実行状態の記憶量、回復ポイントが決められる。

【0061】

続いて図10は入力データ記憶領域655に記録された入力データ108のバックアップと、図9で示した各オペレータにおける実行状態の回復ポイント以降のデータ数を示している。

【0062】

データ群sa1001はScan400に入力されるデータ群でデータ501~506およびデータ1020~1023等から構成されている。データ群sb1002はScan430に入力されるデータ群でデータ511~517およびデータ1030~1035から構成されている。これを各回復ポイントで記録する場合、W4408の回復ポイント932である6:30から保存する場合、記憶するデータ数1010は1000となる。同様にW5412の回復ポイント942である6:45から保存する場合、記憶するデータ数1011は900となり、W1401の回復ポイント902である9:48の場合はデータ数1012が17、W3411の回復ポイント922の9:50の場合、データ数1013は14、W2404の回復ポイント912の9:55の場合はデータ数1014が9となる。

【0063】

図11ではこれらの情報を用いて処理800~813を行った結果をまとめたものを示した。W1の回復ポイント902である9:48を選択した場合は、W2の回復ポイントが9:55、W3の回復ポイントが9:50であるため、W1、W2、W3は入力データのバックアップから実行状態を再現できる。一方、W4とW5の回復ポイントはW1より古いため、入力データのバックアップから再現できない。そこで、W4とW5はスナップショットが必要となる。

【0064】

その結果、この場合の必要記憶容量1101はW1の回復ポイント902での入力データバックアップのデータ数1012である17とW4とW5のスナップショットの記憶量931、941の合計である120となる。同様の処理をするとW2の回復ポイント選択時の必要記憶容量1102は127、W3の必要記憶容量1103は123、W4の必要記憶容量1103は1000、W5の必要記憶容量1104は1000となる。

【0065】

図12では処理814、815により必要記憶容量の最も少ないW1の回復ポイントが選択された時の回復ポイントとスナップショットで再現するオペレータのリストである。

【0066】

このときの回復ポイント1201はW1の回復ポイントである9:48、入力データのバックアップから再現するオペレータ1202はW1、W2、W3、スナップショットから再現するオペレータ1203はW4、W5となる。

【0067】

図13A、図13Bそれぞれが、本具体例における、記録される入力データのバックアップ1300とスナップショット1310を示している。入力データのバックアップ1300は回復ポイントである9:48以降のデータ、スナップショット1310はW4とW5の実行状態を記録している。

【0068】

続いて図14を用いて、本実施例における、入力データのバックアップとスナップショットから初期状態のストリームデータ処理システムに実行状態を再現する手順のフローチャートを示す。

【0069】

処理1400においてストリームデータ処理システム206の復旧要求送信部610が

10

20

30

40

50

バックアップ用ストレージシステム 216 に復旧要求を送信する。それを受けて処理 1401 においてバックアップ用ストレージシステム 216 が入力データのバックアップとスナップショットをストリームデータ処理システム 206 に送信する。処理 1402 において入力データのバックアップとスナップショットを送られたストリームデータ処理システム 206 は障害前の実行状態を復旧する。最後に処理 1403 において障害後の入力データから処理を継続する。

【0070】

図 15 に図 14 の処理 1402 の詳細を示した。最初に処理 1500 において回復ポイントからバックアップデータ取得時刻までの入力データのバックアップを初期状態のストリームデータ処理システム 206 で処理する。続いて処理 1501 ~ 1504 においてスナップショットを取得しているオペレータ全てにスナップショットの実行状態をコピーする。最後にバックアップデータ取得後から障害発生直前までの入力データのバックアップをストリームデータ処理システム 206 で処理する。

10

【0071】

図 16、図 17、図 18 を用いて、図 13 で取得したスナップショットから初期状態のストリームデータ処理システムに対して図 15 のフローチャートに示した手順でバックアップデータ取得時の実行状態を再現する例を示す。

【0072】

図 16 では初期状態のストリームデータ処理システムに対し処理 1500 の回復ポイントからバックアップデータ取得時までの入力データのバックアップ 1300 を入力している。

20

【0073】

図 17 がその結果である。この場合、入力データのバックアップから実行状態の再現できる W1 401、W2 404、W3 411 の 3 つはバックアップデータ取得時刻 1750 である 10:00 の実行状態が再現されている。一方、W4 408 は本来 6:30 からのデータを保存していたため 9:48 のデータからではデータ量が足りず、W5 412 は 6:30 からのデータの最大値を記憶していたため、9:48 からの最大値であるデータ 1701 ~ 1703 は本来のものと値が異なっている。

【0074】

図 18 で図 17 の状態に対し処理 1501 ~ 1504 を行う例を示す。入力データのバックアップ 1300 から再現できない W4 408、W5 412 の実行状態についてスナップショット 1310 から実行状態をコピーする。その結果、W4 408、W5 412 を含め全てのオペレータに対し図 9 と同様のバックアップデータ取得時の実行状態が再現される。

30

【0075】

この後は処理 1505 にあるようにバックアップデータ取得後の入力データのバックアップを処理すれば障害直前の実行状態が再現される。

【0076】

ここまでで、スナップショットの取得の処理は一定間隔、または入力データのバックアップの容量が一定値に達した場合に自動で行われてもかまわない。

40

【0077】

また、図 19 に示すように (Graphic User Interface: GUI) 1900 を用いて、バックアップデータ取得化の最適化機能の使用の有無 1901、一定間隔の時刻 1902、バックアップデータの容量の限界値 1903 などを設定できるよう構成しても良い。なお、1094 はユーザが、所望する任意の時間に、直ちに最適化を実行するために用いる「最適化実施」ボタンを示す。

【0078】

以上の詳述した本発明の処理手順により、最小限の記憶領域でストリームデータ処理システムの実行状態を再現する手段が実現できる。

【産業上の利用可能性】

50

## 【 0 0 7 9 】

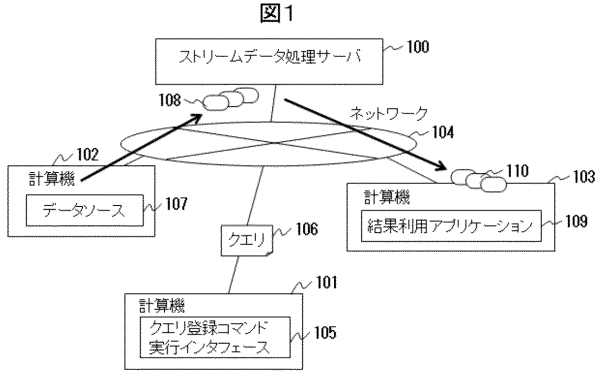
本発明は、ストリームデータ処理における障害回復技術に関し、特に、障害回復に必要な再現データの保存技術として有用である。

## 【 符号の説明 】

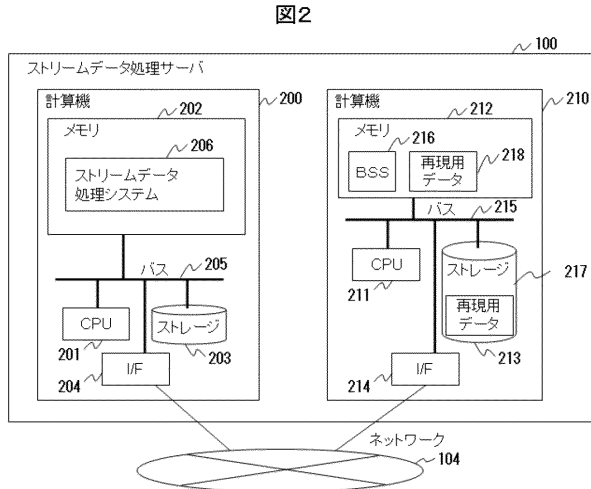
## 【 0 0 8 0 】

1 0 0 ...ストリーム処理サーバ	
1 0 1、1 0 2、1 0 3、2 0 0、2 1 0 ... 計算機	
1 0 4 ... ネットワーク	
2 0 1、2 1 1 ... C P U	
2 0 2、2 1 2 ... メモリ	10
2 0 3、2 1 3 ... ストレージ装置	
2 0 4、2 1 4 ... ネットワーク I / F	
2 0 5、2 1 5 ... 計算機内部バス	
2 0 6 ... ストリームデータ処理システム	
2 1 6 ... バックアップ用ストレージシステム ( B S S )	
2 1 7、2 1 8 ... 再現用バックアップデータ	
4 0 0 ~ 4 1 0 ... オペレータ	
4 1 1、4 1 2 ... 一時保持領域	
6 0 1 ... 入力データ受信部	
6 0 2 ... クエリ実行部	20
6 0 5 ... 出力データ送信部	
6 0 6 ... クエリ解析部	
6 0 8、6 5 6 ... スナップショット対象リスト記憶領域	
6 0 9、6 5 7 ... 複製データ通信部	
6 1 0 ... 復旧要求送信部	
6 1 1 ... バックアップ通知受信部	
6 1 2、6 6 0 ... コピーバッファ領域	
6 1 3、6 6 1 ... ワークエリアデータ通信部	
6 5 2 ... バックアップデータ管理部	
6 5 5 ... 入力データ記憶領域	30
6 5 8 ... 復旧要求受信部	
6 5 9 ... バックアップ通知送信部	
6 2 1、6 2 2、6 2 3 ... オペレータ実行状態保持領域	
6 2 4、6 2 5、6 2 6 ... オペレータ回復ポイント記録領域	
6 7 1、6 7 2、6 7 3 ... オペレータ実行状態記憶領域	
5 0 1 ~ 5 0 6、5 1 1 ~ 5 1 7、5 2 1 ~ 5 3 1、5 4 1 ~ 5 4 3、1 0 2 0 ~ 1 0 2 3、1 0 3 0 ~ 1 0 3 5、1 7 0 1 ~ 1 7 0 3 ... データ	
9 0 1、9 1 1、9 2 1、9 3 1、9 4 1 ... スナップショット記憶量	
9 0 2、9 1 2、9 2 2、9 3 2、9 4 2 ... 回復ポイント	
1 3 0 0 ... 入力データバックアップ	40
1 3 0 1 ... スナップショットデータ	
1 9 0 0 ... バックアップ方式設定 G U I。	

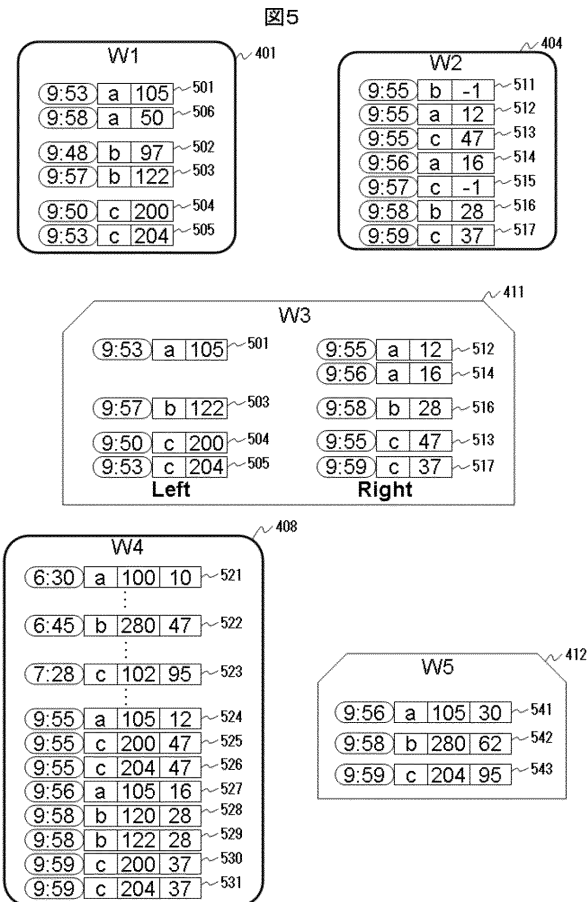
【 図 1 】



【 図 2 】



【 図 5 】



【 図 3 】

```

REGISTER STREAM sa(id VARCHAR(30), va INT);
REGISTER STREAM sb(id VARCHAR(30), va INT);

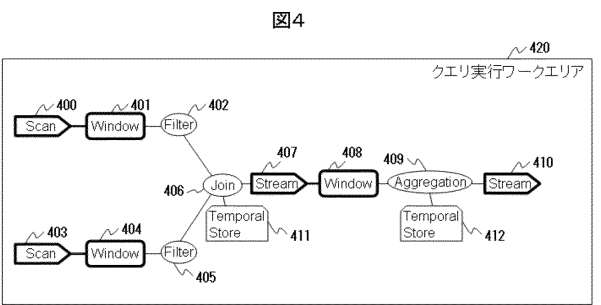
REGISTER QUERY q1
  ISTREAM(
  SELECT sa.id, sa.va AS va, sb.va AS vb
  FROM sa[PARTITION BY id ROWS 2], sb[RANGE 5 MINUTES]
  WHERE sa.id = sb.id AND sa.va > 100 AND sb.va <> -1
  );

REGISTER QUERY q2
  SELECT id, MAX(va) AS va, MAX(vb) AS vb
  FROM q1[UNBOUNDED]
  GROUP BY id;

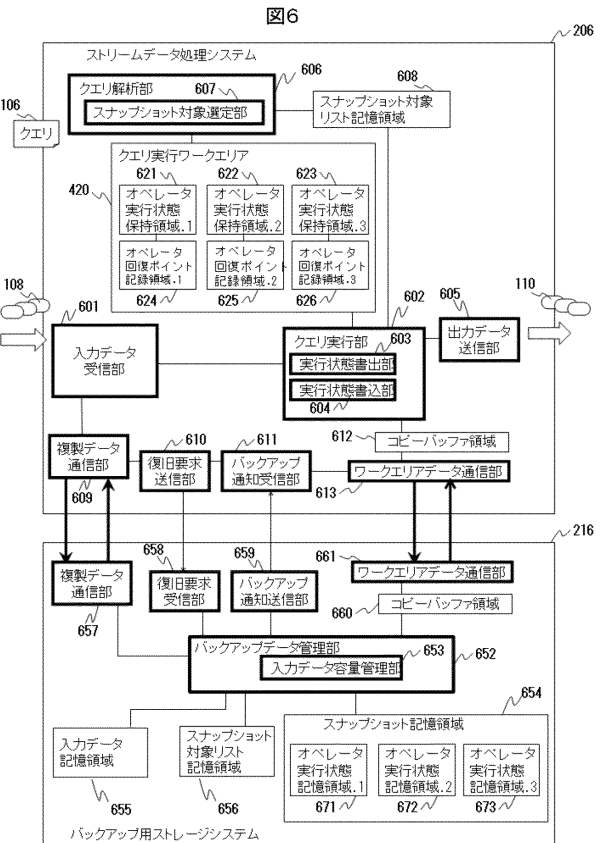
REGISTER QUERY q3
  ISTREAM(SELECT * FROM q2);

```

【 図 4 】

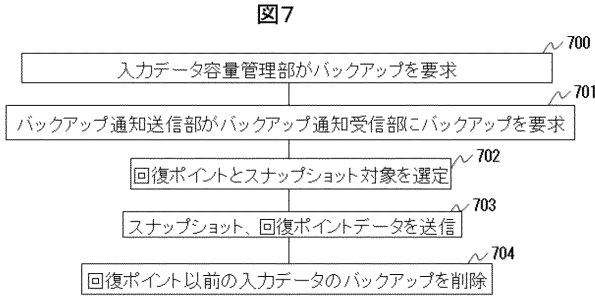


【 図 6 】

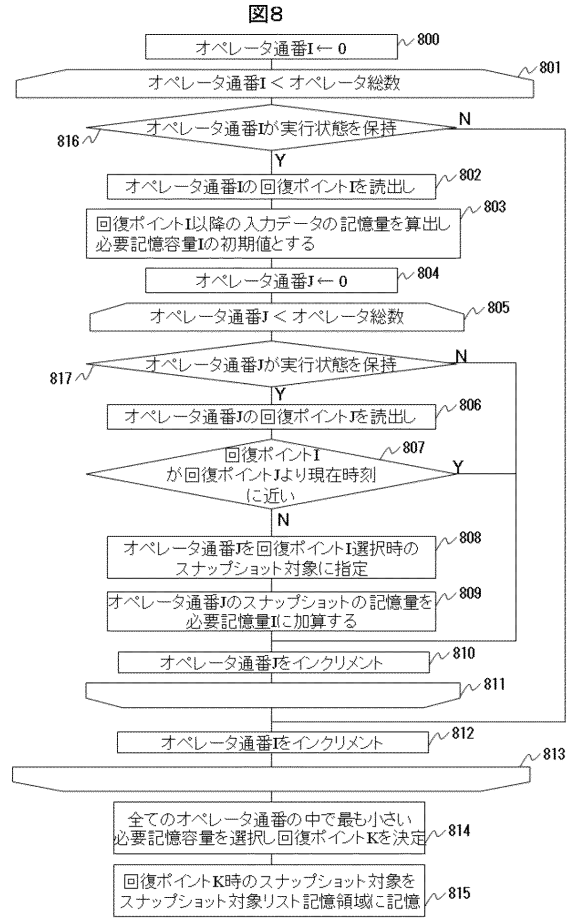




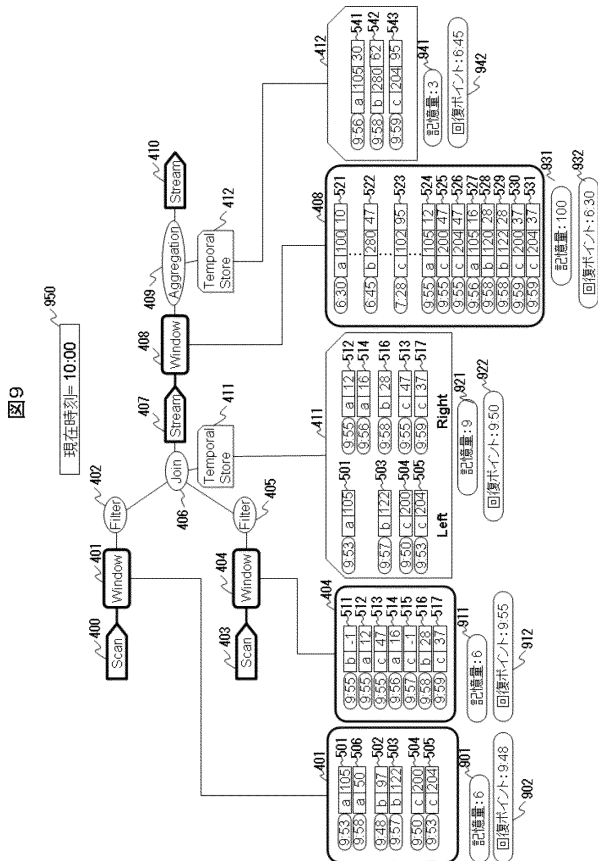
【 図 7 】



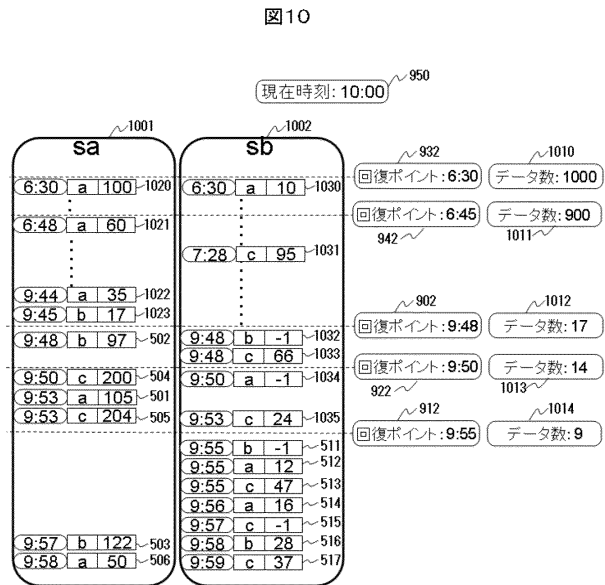
【 図 8 】



【 図 9 】



【 図 10 】



【図11】

図11

I	回復ポイント	入力データ記憶量	W1	W2	W3	W4	W5	必要記憶容量
W1	9:48	17	0	0	0	100	3	120
W2	9:55	9	6	0	9	100	3	127
W3	9:50	14	6	0	0	100	3	123
W4	6:30	1000	0	0	0	0	0	1000
W5	6:48	900	0	0	0	100	0	1000

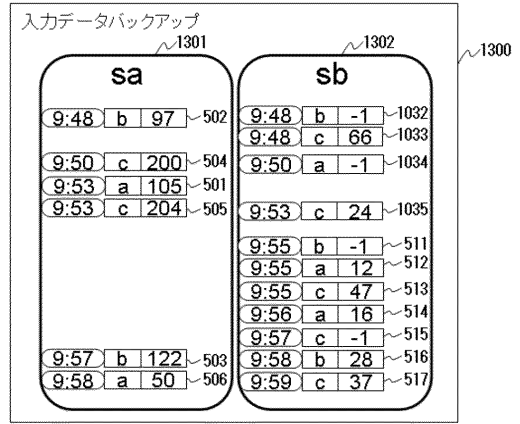
【図12】

図12

回復ポイント	入力データから再現	スナップショットで再現
9:48(W1)	W1, W2, W3	W4, W5

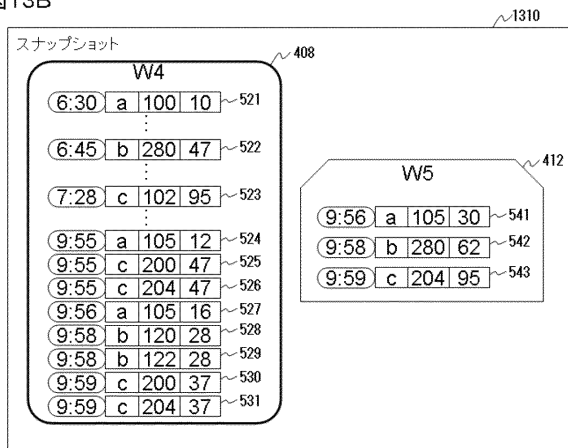
【図13A】

図13A



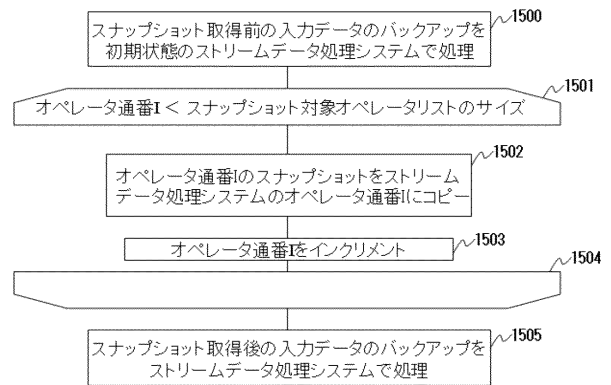
【図13B】

図13B



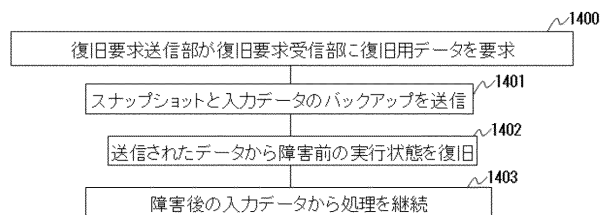
【図15】

図15

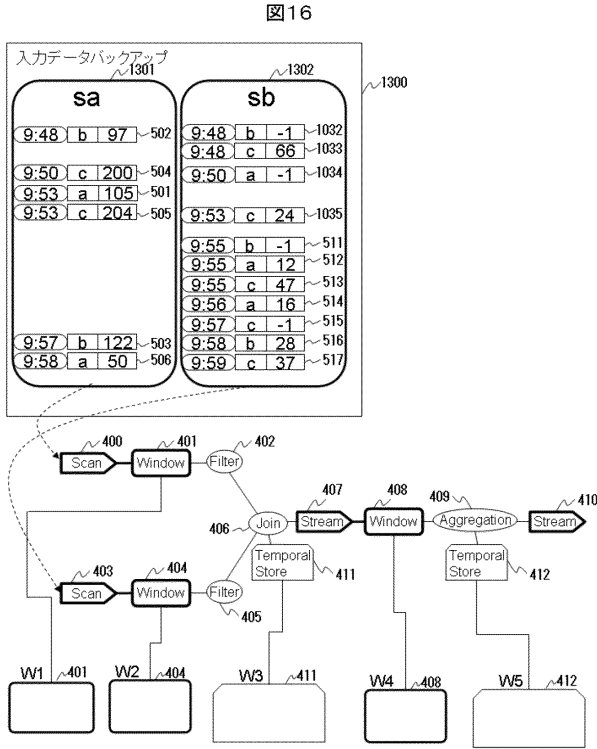


【図14】

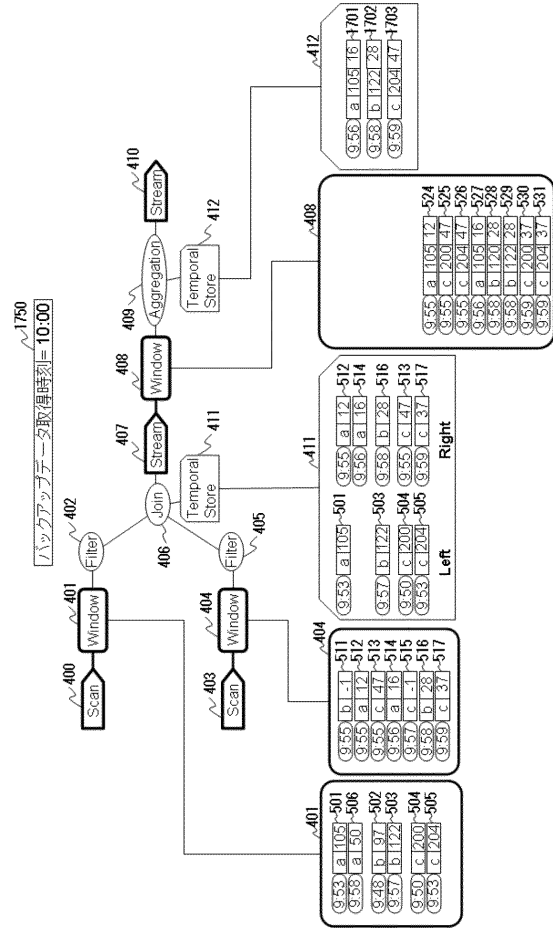
図14



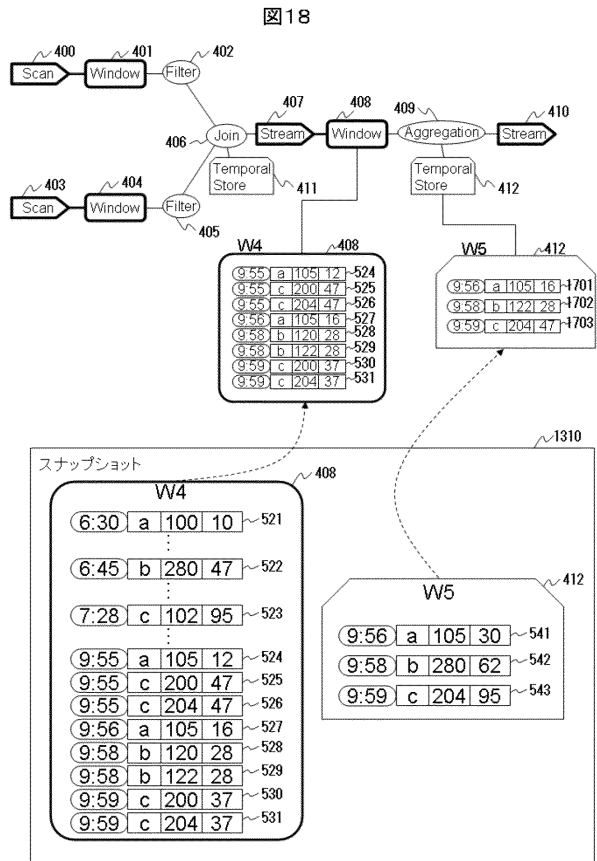
【図16】



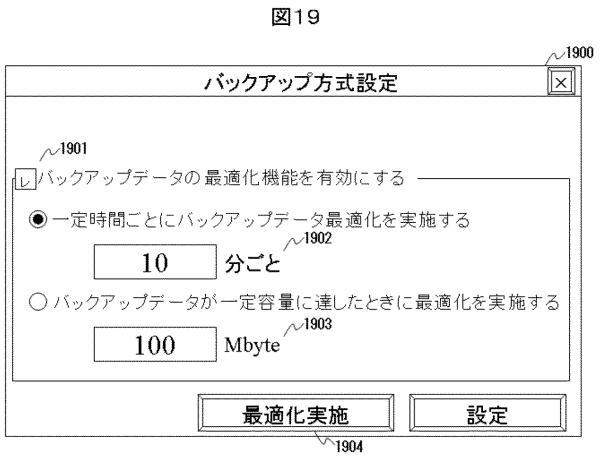
【図17】



【図18】



【図19】



---

フロントページの続き

審査官 桜井 茂行

(56)参考文献 特開2006-338432(JP,A)

Jeong-Hyon Hwang et al., "A Comparison of Stream-Oriented High-Availability Algorithms", Technical Report CS-03-17, 米国, 2003年 9月, P. 1-13

Jeong-Hyon Hwang et al., "A Cooperative, Self-Configuring High-Availability Solution for Stream Processing", Proceedings of 2007 IEEE 23rd International Conference on Data Engineering, 米国, IEEE, 2007年 4月20日, P. 176-185

Gabriela Jacques-Silva et al., "Language-Level Checkpointing Support for Stream Processing Applications", Proceedings of 2009 IEEE/IFIP International Conference on Dependable Systems and Networks, 米国, IEEE, 2009年 7月 2日, P. 145-154

(58)調査した分野(Int.Cl., DB名)

G06F 12/00

G06F 17/30