(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3696832号
(P3696832)
(45)発行日　平成17年9月21日(2005.9.21)　(24)登録日　平成17年7月8日(2005.7.8)

(51)Int.Cl.[7]　F I
G06F 11/28　G06F 11/28　３１５Ａ
G06F 11/14　G06F 11/14　３１０Ｂ

請求項の数 48　(全 22 頁)

(21)出願番号　特願2001-544143(P2001-544143)
(86)(22)出願日　平成12年12月5日(2000.12.5)
(65)公表番号　特表2003-516581(P2003-516581A)
(43)公表日　平成15年5月13日(2003.5.13)
(86)国際出願番号　PCT/US2000/042609
(87)国際公開番号　WO2001/042920
(87)国際公開日　平成13年6月14日(2001.6.14)
審査請求日　平成14年5月31日(2002.5.31)
(31)優先権主張番号　60/169,097
(32)優先日　平成11年12月6日(1999.12.6)
(33)優先権主張国　米国(US)
(31)優先権主張番号　09/608,995
(32)優先日　平成12年6月30日(2000.6.30)
(33)優先権主張国　米国(US)

(73)特許権者　501278238
エービー　イニティオ　ソフトウェア　コーポレーション
アメリカ合衆国　０２４２１　マサチューセッツ州　レキシントン，スプリング　ストリート　２０１
(74)代理人　100088971
弁理士　大庭　咲夫
(74)代理人　100115185
弁理士　加藤　慎治
(72)発明者　スタンフィル　クレイグ　ダブリュ
アメリカ合衆国　マサチューセッツ州　０２４５３　ウォルサム　フローレンス　ロード　６２

最終頁に続く

(54)【発明の名称】連続フローチェックポイント設定データ処理法

(57)【特許請求の範囲】
【請求項１】
一つまたは複数のコンピュータを備えていて、前記コンピュータによるプログラムの実行により、データフローおよびそのデータフローによって連結された少なくとも二つのプロセスを含む少なくとも一つのプロセスステージを実現したデータ処理システムにおいて、前記コンピュータによるプログラムの実行により、
（ａ）少なくとも一つのコマンドメッセージを、プロセスステージを通じてデータフローの一部として伝搬させ、
（ｂ）各プロセスによる少なくとも一つのコマンドメッセージの受取りに応答して、プロセスステージ内の各プロセスにチェックポイントを設定する
ことを含む連続したデータフローのチェックポイント設定を行う方法。
【請求項２】
（ａ）データ処理システムの状態を復元する必要があることを確認し、
（ｂ）各プロセスを対応する保存された状態に復元する
ことをさらに含む請求項１に記載の方法。
【請求項３】
前記連結されたプロセスの少なくとも一つがソースまたはシンクである請求項１に記載の方法。
【請求項４】
チェックポイントの設定が、通常の処理を一時停止し、対応する状態を保存し、次いで

通常の処理に戻すことを含む請求項１に記載の方法。
【請求項５】
一つまたは複数のコンピュータを備えていて、前記コンピュータによるプログラムの実行により、入力データを受け取り記憶する一つまたは複数のソース、一つもしくは複数のソースまたは前のプロセスからデータを受け取って処理する一つまたは複数のプロセス、および処理されたデータを一つもしくは複数のプロセスまたはソースから受け取り次いで処理されたデータを発表する一つまたは複数のシンクを実現したデータ処理システムにおいて、前記コンピュータによるプログラムの実行により、
（ａ）チェックポイント要求メッセージをあらゆるソースに伝送し、
（ｂ）前記チェックポイント要求メッセージを受け取ってこれに応答して、各ソースの通常のデータ処理を一時停止し、前記ソースの状態を復元するのに十分な現在のチェックポイントレコードを保存し、チェックポイントメッセージを、前記ソースからのデータを消費するプロセスに前記ソースから伝搬させ、次いで各ソースにおける通常のデータ処理を再開し、
（ｃ）あらゆるソースまたは前のプロセスからのデータを消費する各プロセスにおける通常のデータ処理を、あらゆるソースまたは前のプロセスからチェックポイントメッセージを受け取りこれに応答して一時停止し、前記プロセスの状態を復元するのに十分な現在のチェックポイントレコードを保存し、チェックポイントメッセージを、前記プロセスからのデータを消費するプロセスまたはシンクに前記プロセスから伝搬させ、次いで前記プロセスにおける通常のデータ処理を再開し、
（ｄ）あらゆるプロセスからのデータを消費する各シンクにおける通常のデータ処理を、あらゆるプロセスからチェックポイントメッセージを受け取りこれに応答して一時停止し、前記シンクの状態を復元するのに十分な現在のチェックポイントレコードを保存し、未発表データを保存し、次いで前記シンクにおける通常のデータ処理を再開することを含む連続したフローのチェックポイント設定を行う方法。
【請求項６】
各ソース、プロセスまたはシンクの通常のデータ処理を一時停止する前に、その各ソース、プロセスまたはシンクについて保留中の計算を完了することをさらに含む請求項５に記載の方法。
【請求項７】
現在のチェックポイントレコードを記憶する前に、各ソース、プロセスまたはシンクの古いチェックポイントレコードを削除することをさらに含む請求項５に記載の方法。
【請求項８】
削除された古いチェックポイントレコードに関連する記憶スペースを再要求することをさらに含む請求項７に記載の方法。
【請求項９】
各ソースがデータ待ち行列を含み、そして現在のチェックポイントレコードを保存することが、現在のデータ待ち行列の位置を保存することを含んでいて、
さらに前記保存された現在のデータ待ち行列の位置の前に出現するデータに関連する記憶スペースを再要求することを含む請求項５に記載の方法。
【請求項１０】
各シンクがデータ待ち行列を含み、そして現在のチェックポイントレコードを保存することが、現在のデータ待ち行列の位置を保存することを含んでいて、
さらに前記保存された現在のデータ待ち行列の位置の前に出現するデータと関連する記憶スペースを再要求することを含む請求項５に記載の方法。
【請求項１１】
さらに、
（ａ）データ処理システムの状態を復元する必要があることを確認し、
（ｂ）各ソース、プロセスおよびシンクの状態を、対応する現在のチェックポイントレコードから復元する

ことを含んでいる請求項５に記載の方法。
【請求項１２】
チェックポイントトリガーイベントの検出に応答してチェックポイント要求メッセージを生成することをさらに含んでいる請求項５に記載の方法。
【請求項１３】
チェックポイントトリガーイベントが定期的に起こる請求項１２に記載の方法。
【請求項１４】
チェックポイントトリガーイベントが外部刺激に基づいている請求項１２に記載の方法。
【請求項１５】
チェックポイントトリガーイベントが、処理されている入力データレコード内またはそのデータレコードから誘導される選択されたデータ値の出現に基づいている請求項１２に記載の方法。
【請求項１６】
さらに、
（ａ）各ソース内の入力データを、選択されたデータ値について走査し、
（ｂ）選択されたデータ値を各ソース内に検出したとき、前記ソースからのデータを消費するプロセスに、データの終わりが出現したことを示していてチェックポイントの設定を要求する制御メッセージを伝送し、
（ｃ）制御メッセージがあらゆるソースによって伝送されると直ちにチェックポイントトリガーイベントが起こったことを確認する
ことを含む請求項１２に記載の方法。
【請求項１７】
さらに、
（ａ）各ソース内の入力データレコードを検査し、次いでその検査に基づいて選択されたデータ値を確認し、
（ｂ）前記選択されたデータ値を各ソースに提供し、
（ｃ）各ソース内の入力データレコードを、前記選択されたデータ値について走査し、
（ｄ）前記選択されたデータ値を各ソース内に検出すると、データの終わりが出現したことを示していてチェックポイントの設定を要求する制御メッセージを、前記ソースからのデータを消費するプロセスに伝送し、
（ｅ）制御メッセージがあらゆるソースによって伝送されると直ちにチェックポイントトリガーイベントが起こったことを確認する
ことを含む請求項１２に記載の方法。
【請求項１８】
シンクからの出力を定期的に生成することによって、チェックポイント設定を調整することをさらに含む請求項５に記載の方法。
【請求項１９】
（ａ）各ソース、プロセスおよびシンクを通じて、ジョブ終了の指示を伝搬させ、
（ｂ）通常のデータ処理を再開する代わりにジョブの終了の指示に応答して、各ソース、プロセスおよびシンク内のデータ処理から抜け出ることによってデータ処理を終了することをさらに含む請求項５に記載の方法。
【請求項２０】
通常のデータ処理を再開するほとんど直前に前記データ値を発表することをさらに含む請求項５に記載の方法。
【請求項２１】
未発表のデータ値が決定的であることを確認し、次にこのようなデータ値を、前記未発表データを保存するほとんど直後に発表することをさらに含む請求項５に記載の方法。
【請求項２２】
未発表データ値が決定的でかつ順序付けられていることを確認し、次いで前記シンクが

、あらゆるプロセスからのデータを消費するそのあらゆるプロセスからチェックポイントメッセージを受け取った後で、通常のデータ処理を再開する前の時点に前記データ値を発表することをさらに含む請求項５に記載の方法。

【請求項２３】

再び発表するデータ値が許容可能であることを確認し、次いで前記シンクが、あらゆるプロセスからのデータを消費するそのあらゆるプロセスからチェックポイントメッセージを受け取った後で、通常のデータ処理を再開する前の時点に前記データ値を発表することをさらに含む請求項５に記載の方法。

【請求項２４】

一つまたは複数のコンピュータを備え、前記コンピュータによるプログラムの実行により、入力データを受け取り次いで記憶する一つまたは複数のソース、データを一つもしくは複数のソースまたは前のプロセスから受け取って処理する一つまたは複数のプロセス、および処理されたデータを一つもしくは複数のプロセスまたはソースから受け取り次いで処理されたデータを発表する一つまたは複数のシンクを実現したデータ処理システムにおいて、前記コンピュータによるプログラムの実行により、

（ａ）チェックポイント要求メッセージをあらゆるソースに伝送し、

（ｂ）前記チェックポイント要求メッセージを受け取りこれに応答して各ソースの通常データ処理を一時停止し、前記ソースの状態を復元するのに十分な現在のチェックポイントレコードを保存し、チェックポイントメッセージを前記ソースからのデータを消費するプロセスに前記ソースから伝搬させ、次いで各ソースにおいて通常のデータ処理を再開し、

（ｃ）前記プロセスが、あらゆるソースまたは前のプロセスからのデータを消費するそのあらゆるソースまたは前のプロセスからチェックポイントメッセージを受け取りこれに応答して各プロセスの通常のデータ処理を一時停止し、前記プロセスの状態を復元するのに十分な現在のチェックポイントレコードを保存し、前記プロセスからのチェックポイントメッセージを、前記プロセスからのデータを消費するプロセスまたはシンクに伝搬させ、次いで前記プロセスにおける通常データ処理を再開し、

（ｄ）前記シンクが、あらゆるプロセスからのデータを消費するそのあらゆるプロセスからのチェックポイントメッセージを受け取りこれに応答して、各シンクの通常データ処理を一時停止し、前記シンクの状態を復元するのに十分な現在のチェックポイントレコードを保存し、未発表データを保存し、次いでチェックポイントメッセージをチェックポイントプロセッサに各シンクから伝搬させ、

（ｅ）チェックポイントメッセージをすべてのシンクから受け取り、そしてその受け取ったことに応答し、すべてのソース、プロセスおよびシンクでのチェックポイント設定が完了したことを示す記憶された値を更新し、次にその記憶された値を各シンクに伝送し、次いで

（ｆ）前記記憶された値を各シンクにおいて受け取って、その受取りに応答して前記シンクに関連する未発表データを発表し、次に前記シンクにおける通常のデータ処理を再開する

ことを含む連続したデータフローのチェックポイント設定を行う方法。

【請求項２５】

一つまたは複数のコンピュータを備え、前記コンピュータによるプログラムの実行により、データフローおよびそのデータフローで連結された少なくとも二つのプロセスを含む少なくとも一つのプロセスステージを実現したデータ処理システムにおいて、連続したデータフローのチェックポイント設定を行うために、

前記コンピュータに、

（ａ）少なくとも一つのコマンドメッセージを、プロセスステージを通じてデータフローの一部として伝搬させる機能、および

（ｂ）各プロセスが少なくとも一つのコマンドメッセージを受け取り、その受取りに応答して、前記プロセスステージ内の各プロセスにチェックポイントを設定する機能

を、実現させるためのプログラムを記録したコンピュータ読取り可能な記録媒体。

【請求項２６】
前記コンピュータに、さらに、
（ａ）データ処理システムの状態を復元する必要があることを確認する機能、および
（ｂ）各プロセスを、対応する保存された状態に復元する機能
を、実現させるようにしたプログラムを記録した請求項２５に記載のコンピュータ読取り可能な記録媒体。
【請求項２７】
前記連結されたプロセスの少なくとも一つがソースまたはシンクである請求項２５に記載のコンピュータ読取り可能な記録媒体。
【請求項２８】
前記チェックポイントを設定する機能が、通常の処理を一時停止し、対応する状態を保存し、次いで通常の処理に戻す機能を含んでいる請求項２５に記載のコンピュータ読取り可能な記録媒体。
【請求項２９】
一つまたは複数のコンピュータを備え、前記コンピュータによるプログラムの実行により、入力データを受け取り入力されたデータを記憶する一つまたは複数のソース、一つもしくは複数のソースまたは前のプロセスからデータを受け取って処理する一つもしくは複数のプロセス、および一つもしくは複数のプロセスまたはソースから処理されたデータを受け取り次いで処理されたデータを発表する一つまたは複数のシンクを実現したデータ処理システムにおいて、連続したデータフローのチェックポイント設定を行うために、
前記コンピュータに、
（ａ）チェックポイント要求メッセージをあらゆるソースに伝送する機能
（ｂ）前記チェックポイント要求メッセージを受け取りそれに応答して各ソースの通常データ処理を一時停止し、前記ソースの状態を復元するのに十分な現在のチェックポイントレコードを保存し、チェックポイントメッセージを、前記ソースからのデータを消費するプロセスに前記ソースから伝搬させ、次いで各ソースにおける通常データ処理を再開する機能、
（ｃ）あらゆるソースまたは前のプロセスからのデータを消費する各プロセスの通常データ処理を、あらゆるソースまたは前のプロセスからのチェックポイントメッセージを受け取りそれに応答して一時停止し、前記プロセスの状態を復元するのに十分な現在のチェックポイントレコードを保存し、チェックポイントメッセージを前記プロセスからのデータを消費するプロセスまたはシンクに前記プロセスから伝搬させ、次いで前記プロセスにおける通常データ処理を再開する機能、および
（ｄ）あらゆるプロセスからのデータを消費する各シンクの通常データ処理を、あらゆるプロセスからのチェックポイントメッセージを受け取りこれに応答して一時停止し、前記シンクの状態を復元するのに十分な現在のチェックポイントレコードを保存し、未発表のデータを保存し、次いで前記シンクにおける通常データ処理を再開する機能
を、実現させるためのプログラムを記録したコンピュータ読取り可能な記録媒体。
【請求項３０】
前記コンピュータに、さらに、
各ソース、プロセスまたはシンクの通常データ処理を一時停止する前に、前記ソース、プロセスまたはシンクの保留中の計算を完了する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。
【請求項３１】
前記コンピュータに、さらに、
現在のチェックポイントレコードを記憶する前に、各ソース、プロセスまたはシンクの古いチェックポイントレコードを削除する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。

【請求項３２】
前記コンピュータに、さらに、
削除された古いチェックポイントレコードに関連する記憶スペースを再要求する機能を、実現させるようにしたプログラムを記録した請求項３１に記載のコンピュータ読取り可能な記録媒体。
【請求項３３】
各ソースがデータ待ち行列を含み、前記現在のチェックポイントレコードを保存する機能が、現在のデータ待ち行列の位置を保存する機能を含んでおり、
前記コンピュータに、さらに、
前記保存された現在のデータ待ち行列の位置の前に出現するデータに関連する記憶スペースを再要求する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。
【請求項３４】
各シンクがデータ待ち行列を含み、前記現在のチェックポイントレコードを保存する機能が、現在のデータ待ち行列の位置を保存する機能を含んでおり、
前記コンピュータに、さらに、
前記保存された現在のデータ待ちの行列の位置の前に出現するデータに関連する記憶スペースを再要求する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。
【請求項３５】
前記コンピュータに、さらに、
（ａ）データ処理システムの状態を復元する必要があることを確認する機能、および
（ｂ）各ソース、プロセスおよびシンクの状態を対応する現在のチェックポイントレコードから復元する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。
【請求項３６】
前記コンピュータに、さらに、
チェックポイントトリガーイベントの検出に応答して、チェックポイント要求メッセージを生成する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。
【請求項３７】
チェックポイントトリガーイベントが定期的に出現する請求項３６に記載のコンピュータ読取り可能な記録媒体。
【請求項３８】
チェックポイントトリガーイベントが外部刺激に基づいている請求項３６に記載のコンピュータ読取り可能な記録媒体。
【請求項３９】
チェックポイントトリガーイベントが、処理中の入力データレコード内またはその入力データレコードから誘導される選択されたデータ値の出現に基づいている請求項３６に記載のコンピュータ読取り可能な記録媒体。
【請求項４０】
前記コンピュータに、さらに、
（ａ）各ソース内の入力データレコードを、選択されたデータ値について走査する機能、
（ｂ）前記選択されたデータ値を各ソース内に検出すると、データの終わりが出現したことを示していてチェックポイントの設定を要求する制御メッセージを、前記ソースからのデータを消費するプロセスに伝送する機能、および

（ｃ）制御メッセージがあらゆるソースによって伝送されると直ちに、チェックポイントトリガーイベントが出現したことを確認する機能
を、実現させるようにしたプログラムを記録した請求項３６に記載のコンピュータ読取り可能な記録媒体。
【請求項４１】
前記コンピュータに、さらに、
（ａ）各ソース内の入力データを検査して、前記検査に基づいて選択されたデータ値を確認する機能、
（ｂ）前記選択されたデータ値を各ソースに提供する機能、
（ｃ）各ソース内の入力データレコードを、前記選択されたデータ値について走査する機能、
（ｄ）前記選択されたデータ値を各ソース内に検出すると、データの終りが出現したことを示していてチェックポイントの設定を要求する制御メッセージを、前記ソースからのデータを消費するプロセスに伝送する機能、および
（ｅ）制御メッセージがあらゆるソースによって伝送されると直ちにチェックポイントトリガーイベントが出現したことを確認する機能
を、実現させるようにしたプログラムを記録した請求項３６に記載のコンピュータ読取り可能な記録媒体。
【請求項４２】
前記コンピュータに、さらに、
シンクから出力を定期的に生成することによってチェックポイントの設定を調整する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。
【請求項４３】
前記コンピュータに、さらに、
（ａ）ジョブの終りの指示を、各ソース、プロセスおよびシンクを通じて伝搬させ、
（ｂ）ジョブの終りの指示に応答して、通常データ処理を再開する代わりに、各ソース、プロセスおよびシンクのデータ処理から抜け出ることによってデータ処理を終了する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。
【請求項４４】
前記コンピュータに、さらに、
通常データ処理を再開するほとんど直前に、前記データ値を発表する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。
【請求項４５】
前記コンピュータに、さらに、
未発表データ値が決定的であることを確認し、そしてその未発表データを保存したほとんど直後に前記データ値を発表する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。
【請求項４６】
前記コンピュータに、さらに、
未発表データ値が決定的でかつ順序付けられていることを確認し、次いで前記シンクが、あらゆるプロセスからのデータを消費するそのあらゆるプロセスからのチェックポイントメッセージを受け取った後で、通常のデータ処理を再開する前の時点に前記データ値を発表する機能
を、実現させるようにしたプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。

【請求項４７】
前記コンピュータに、さらに、
再発表中のデータ値が許容可能であることを確認し、次いで前記シンクが、あらゆるプロセスからのデータを消費するそのあらゆるプロセスからのチェックポイントメッセージを受け取った後で、通常のデータ処理を再開する前の時点に前記データ値を発表する機能を、実現させるプログラムを記録した請求項２９に記載のコンピュータ読取り可能な記録媒体。
【請求項４８】
一つまたは複数のコンピュータを備え、前記コンピュータによるプログラムの実行により、入力データを受け取って記憶する一つまたは複数のソース、一つもしくは複数のソースまたは前のプロセスからデータを受け取って処理する一つまたは複数のプロセス、および一つもしくは複数のプロセスまたはソースから処理されたデータを受け取り次いで処理されたデータを発表する一つまたは複数のシンクを実現したデータ処理システムにおいて、連続したデータフローのチェックポイント設定を行うために、
前記コンピュータに、
（ａ）チェックポイント要求メッセージをあらゆるソースに伝送する機能、
（ｂ）前記チェックポイント要求メッセージを受け取ってそれに応答し各ソースの通常のデータ処理を一時停止し、前記ソースの状態を復元するのに十分な現在のチェックポイントレコードを保存し、チェックポイントメッセージを、前記ソースからのデータを消費するプロセスに前記ソースから伝搬させ、次いで各ソースにおける通常のデータ処理を再開する機能、
（ｃ）前記プロセスが、あらゆるソースまたは前のプロセスからのデータを消費するそのあらゆるソースまたは前のソースからチェックポイントメッセージを受け取ってこれに応答して各プロセスの通常のデータ処理を一時停止し、前記プロセスの状態を復元するのに十分な現在のチェックポイントレコードを保存し、チェックポイントメッセージを、前記プロセスからのデータを消費するプロセスまたはシンクに前記プロセスから伝搬させ、次いで前記プロセスにおける通常のデータ処理を再開する機能、
（ｄ）前記シンクが、あらゆるプロセスからのデータを消費するそのあらゆるプロセスからチェックポイントメッセージを受け取ってこれに応答して各シンクの通常のデータ処理を一時停止し、前記シンクの状態を復元するのに十分な現在のチェックポイントレコードを保存し、未発表データを保存し、次いでチェックポイントメッセージをチェックポイントプロセッサに各シンクから伝搬させる機能、
（ｅ）前記チェックポイントメッセージをすべてのシンクから受け取り、その受取りに応答して、すべてのソース、プロセスおよびシンクでのチェックポイント設定の完了を示す記憶された値を更新し、次いでその記憶された値を各シンクに伝送する機能、および
（ｆ）各シンクに記憶された値を受け取り、その受取りに応答して前記シンクに関連する未発表データを発表し、次に前記シンクにおける通常のデータ処理を再開する機能を、実現させるためのプログラムを記録したコンピュータ読取り可能な記録媒体。
【発明の詳細な説明】
【０００１】
【優先権の主張】
本願は、３５ＵＳＣ§１１９（ｅ）に基づいて、１９９９年１２月６日付けで出願された米国特許願第６０／１６９，０９７号の優先権を主張し、かつその全内容を本願に援用するものである。
【０００２】
【発明の属する技術分野】
本発明はデータ処理に関し、さらに詳しくは連続フローチェックポイント設定データ処理を行うシステム、方法およびコンピュータプログラムに関する。
【０００３】
【従来の技術】

データ・アクセスや電子商取引を行うためのインターネットに極めて高い人気があるため、大量のデータをほとんど連続的にかつほとんどリアルタイムで（すなわち利用者の入力に対し数秒間～数分間以内に応答して）処理させる高性能で障害許容性の「内部管理事務（back-office）」処理能力が要望されている。このような処理能力は、故障が起って処理が停止したときに処理を続けることができるように堅固なもの（すなわち障害許容性のあるもの）でなければならない。このような性能は、大規模なインターネットベースデータ処理を行うのに有用であるが、私設のネットワークや通信システム（例えば航空便予約システム、会社内「イントラネット」など）に関する通常タイプのデータ処理に利用できることも多い。

【０００４】

格別な量のデータに対して高い性能を達成することは、並列処理システムを利用して、データを妥当な応答時間内に処理することを意味することが多い。多数の並列処理システムが知られている。例えば、図１は、典型的な従来技術の多重プロセスデータ処理システム１００のブロック図である。最終ソース１０１（例えばウェブ・サーバ）からのデータは、少なくとも一つのデータ待ち行列１０２に伝達される。データは、初期プロセス１０４によって時々、読み出されすなわち「消費され」、その初期プロセス１０４は処理されたデータを一つまたは複数のデータ待ち行列１０６、１０６’に出力する。前記プロセス１０４は、既知の方式で、典型的には、二相コミットプロトコルを使用して、入力データの消費と出力データの伝搬を調整する単一プロセスである。次のプロセス１０８、１０８’が連結されて（並列になっている）、追加の処理と次のデータ待ち行列１１０、１１０’への出力が行われる。そのデータは、関係データベース管理システム（ＲＤＢＭＳ）などの最終消費者１１２に、最終的に出力される。このようなシステムは、実際には、多数のプロセスおよび図示されているより多くの並列性を有している。さらに、各プロセスは、多重データ待ち行列からのデータを消費し次いでデータを多重データ待ち行列に出力する。

【０００５】

障害許容力を得るため、これらのシステムは、計算システムを、データと機械状態の既知の良好な組合せに「ロールバック」させる「チェックポイント設定」法を利用している。詳しく述べると、チェックポイント設定法は、全アプリケーションを最初から再実行するのではなくて、アプリケーションを計算の中間状態を捕獲するチェックポイントから続けさせる。チェックポイント設定システムの例は、発明の名称が「Overpositioning System and Method for Increasing checkpoints in Component-Based Parallel Applications」の米国特許第５，８１９，０２１号および発明の名称が「Methods and Systems for Reconstructing the state of a Computation」の米国特許第５，７１２，９７１号に記載されている。なおこの両特許は本発明の譲渡人に譲渡されている。

【０００６】

ほとんど連続したデータ処理が要望される場合（例えばインターネットデータ処理）、データに対して伝統的なチェックポイント設定法を利用するときの問題は、チェックポイントが、システムが静止しているときのみ、すなわちプロセスが実行中でないときのみつくられるという点である。したがって、あらゆるプロセスが、その状態を保存するため最良の時間を必要とするプロセスによって要求される時間にわたって、実行を一時中断しなければならない。このような一時中断は、データの連続処理に、有害な影響を与える。

【０００７】

したがって、本発明の発明者らは、他のプロセスの状態にチェックポイントを設定するために他のプロセスが必要とする時間とは無関係に、チェックポイントを設定し、次いで各プロセスを、チェックポイント設定後もとに戻して作動させることによってデータ処理の連続的なフロー（流れ）を可能にするデータ処理システムと方法を提供する必要があると判断した。本発明は、これらのおよび他の利点を提供する方法、システムおよびコンピュータを提供するものである。

【０００８】

【発明の概要】
本発明は、チェックポイントを設定し、そして他のプロセスの状態にチェックポイントを設定するために他のプロセスが必要とする時間とは無関係に、チェックポイント設定後、各プロセスを動作状態に戻すことによってデータ処理の連続的フローを可能にする、データ処理システムと方法を含んでいる。
【０００９】
特に、本発明のチェックポイント設定法は、データソース（データ送信装置）からプロセスを通じてデータシンク（データ受信装置）へ、プロセス・ステージを通って逐次伝搬し、各プロセスをトリガーしてその状態にチェックポイントを設定し、次に、チェックポイント設定メッセージを、接続された「下流」プロセスに伝える、チェックポイント・プロセッサからのコマンドメッセージを使用する。この方法は、チェックポイントを設定し、次いで他のプロセスの状態のチェックポイントを設定するため他のプロセスが必要とする時間とは無関係に、チェックポイントを設定後、各プロセスを通常の動作に戻すことによって、データ処理の連続的フローを可能にする。この方法は、各プロセスステージに対する「終端間待ち時間」（すなわちプロセスステージにおけるデータソースからデータシンクに至るデータの全処理時間）を減らし、その結果、全データ処理システムの終端間待ち時間を減らす。
【００１０】
さらに詳しく述べると、本発明は、データフローおよびそのデータフローによって連結された少なくとも二つのプロセスを含む少なくとも一つのプロセスステージを有するデータ処理システムにおいて、連続フローのチェックポイント設定を行う方法、システムおよびコンピュータプログラムであって、そのプロセスステージを通じてデータフローの一部として少なくとも一つのコマンドメッセージを伝搬させ、そして少なくとも一つのコマンドメッセージの各プロセスによる受信に応答して、前記プロセスステージ内の各プロセスにチェックポイントを設定することを含んでなる方法、システムおよびコンピュータプログラムを含むものである。
【００１１】
本発明の一つ以上の実施態様の詳細を、添付図面と以下の説明で述べる。本発明の他の特徴、目的および利点は前記説明と図面および特許請求の範囲から明らかになるであろう。
【００１２】
【発明の実施の形態】
外観
図２は本発明の連続フローデータ処理システム２００のブロック図である。最終ソース２０１（例えばウェブサーバ）からのデータが少なくとも一つのデータ待ち行列２０２に伝達される。データは、次々と連結されているプロセスの一つ以上の並列セットからなる初期プロセスステージ２０４によって時々、読み出されすなわち「消費され」、そしてそれらプロセスは各々、処理されたデータを一つまたは複数のデータ待ち行列２０６、２０６’に出力する。次のプロセスステージ２０８、２０８’が連結され（並列に図示されている）、追加の処理と次のデータ待ち行列２１０、２１０’への出力を行う。そのデータは、関係データベース管理システム（ＲＤＢＭＳ）などの最終消費者２１２に、最終的に出力される。プロセスの全セットは非循環グラフを形成する。プロセスステージ内で、連結されたプロセスの各並列セットによって実行されているプロセスは同じである。このようなシステムは、実際には、図示されているより多くのプロセスステージと並列プロセスを備えている。さらに、各プロセスは、多重データ待ち行列からのデータを消費しそしてデータを多重データ待ち行列に出力する。
【００１３】
図３は本発明の一プロセスステージのブロック図である。各プロセスステージは非循環グラフを形成する。チェックポイントプロセッサ３００が、通信チャネル３０１によって、プロセスステージを構成するすべてのデータソース３０２、３０２’、プロセス３０４、３０４’…３０６、３０６’およびデータシンク３０８、３０８’に連結されている。一

般に、通信チャネル３０１はすべて双方向性である。
【００１４】
データソース（「ソース」）３０２、３０２’は、入力されたデータを受け取り記憶するため関連データ待ち行列（３０３）（例えば、使用者待ち行列）にアクセスし、そしてそれからデータを読み取ることができるプロセスである。各ソースは、データ待ち行列内のデータにチェックポイントを設定し、その結果、システムが故障したとき、チェックポイントの次のデータを再び読み取ることができる。
【００１５】
データシンク（「シンク」）３０８、３０８’は、処理されたデータを受け取るため関連するデータ待ち行列３０９にアクセスし、そしてデータをそれから時々、出力するかまたは発表する（例えば印刷する、表示するまたは記憶する）ことができるプロセスである。各シンクは、そのデータ待ち行列内のデータにチェックポイントを設定し、その結果、システムが故障したとき、チェックポイントの次のデータを再出力することができる。
【００１６】
プロセス３０４、３０４’…３０６、３０６’は、一つまたは複数のソース３０２、３０２’からの入力を直接または間接的に受け取り、そして出力結果を、最終的にシンク３０８、３０８’に出力する。プロセスは、そのデータと処理状態にチェックポイントを設定して、システムが故障したとき、その状態を復元して、データを損失することなしに、処理を、最後のチェックポイントから続けることができる。
【００１７】
ステージ内の最初の並列プロセス３０４、３０４’は、類似のデータ型を含む多重区分ソース３０４、３０４’に並列に連結することができ、そして異なるデータ型を含む多重独立ソースに連結することもできる。ステージ内の最終並列プロセス３０６、３０６’は多重区分シンク３０８、３０８’に並列に連結することができ、そして多重独立シンクに連結することもできる。一プロセスに対するシンクは次のプロセスステージに対するソースである。データフローは、ソースからプロセスを通過してシンクへ一方向性である。プロセスを任意に省略して、シンクをソースに直接接続してもよい。ステージを通じて伝搬する制御メッセージはデータをバイパスしないが、出現の順番で処理される。
【００１８】
図４は本発明に使用するのに適切なデータ待ちの行列の線図である。データレコードが論理的に線形の待ち行列に記憶され、次に、古いものから新しいものへと時系列で配列される。実際の物理的記憶装置は、ランダム・アクセス・メモリ内または媒体上にある。その待ち行列は典型的には、一組の現在レコード４００、破棄可能レコード４０２および利用不能レコード４０４に分割されている。利用不能レコード４０４は、例えば、外部ソースから受け取って記憶されたが、処理するために利用可能になっていないデータ・レコードである。破棄可能なレコード４０２は、消費されたかまたは発表されたが、その記憶スペースが再要求（リクレーム）されていないレコードである。
【００１９】
現在位置のポイントを使用して現在レコードを示すことができる。「開始」ポインタは、典型的には、破棄されたレコードを、現在レコードから明確に区別するのに使用され、そして「終了」ポインタは、典型的には、現在レコードを利用不能レコードから明確に区別するために使用される。
【００２０】
連続フローのチェックポイント設定
本発明の重要な側面は、チェックポイント設定が、チェックポイントプロセッサ３００からのコマンド・メッセージを利用し、ここでコマンドメッセージは、ソース３０２からプロセス３０４、３０６を介したシンク３０８へのプロセス・ステージを通して次々と伝搬し、各プロセスを、その状態にチェックポイントを設定し次いでチェックポイント設定メッセージを接続された「下流」プロセスに伝えるようにトリガーするという側面である。この方法は、チェックポイント設定を行い、次いで、他のプロセスがその状態にチェック

ポイントを設定するのに必要な時間とは無関係に、各トリガーされたプロセスをチェックポイント設定後、通常の動作に戻すことによって、連続フローのデータ処理を可能にする。この方法は、各プロセスステージの「終端間待ち時間」（すなわちプロセスステージのソースからシンクへのデータの全処理時間）を減らし、その結果、全データ処理システム２００の終端間待ち時間を減らす。したがって、グラフは、ジョブが実行中に、利用できる出力を生成することができる。さらに、最後の保存された状態（連続フローの用語の「コミットされたチェックポイント」）の前の時点で入力されたデータは、前記グラフはもはや必要としないので、安全に削除することができる。

【００２１】

図５Ａ～５Ｄは、連続フローチェックポイント設定データ処理システムのプロセスステージでチェックポイントを開始する一実施態様を示すフローチャートである。図６は、連続フローチェックポイント設定データ処理システムの入力データ待ち行列を示すダイアグラムである。図７は、連続フローチェックポイント設定データ処理システムの出力データ待ち行列のダイアグラムである。

【００２２】

１．チェックポイントプロセッサ

【００２３】

ステップ５００：チェックポイント（ＣＰ）トリガーイベントが発生したことを判定する。チェックポイントは、以下に考察するいくつもの方法でトリガーされる。

【００２４】

ステップ５０２：チェックポイント要求メッセージを生成してそれを各ソースに伝送する。チェックポイント要求メッセージは、「要求されたチェックポイント」（ＲＣＰ）フラグおよび「コミットされたチェックポイント」（ＣＣＰ）フラグの値を含んでいる（ステップ５０２）。好ましい実施態様で、これらの値は数値であり、そしてＲＣＰの値はＣＣＰの値より常に大きい。例えば、最初、第一チェックポイントイベントに対するＲＣＰとＣＣＰそれぞれの値は「１、０」である。

【００２５】

２．ソース－チェックポイント要求メッセージを受け取ったときの各ソース

【００２６】

ステップ５０４：保留状態の計算を完成するか、または保留状態の計算を一時停止し、それをチェックポイント設定の一部として保存する。

【００２７】

ステップ５０６：チェックポイント要求メッセージに示されている現在のＣＣＰ値と一致しない既存のチェックポイントレコードを削除し、次いで、関連する記憶スペースを任意に再要求（リクレーム）する。数値を利用する場合、現在のＣＣＰ値より小さいＣＣＰタグもしくはＣＣＰインデックスを有するレコードを削除することができる。二つのチェックポイント値に対して数値が使用されない場合（例えばブールフラグ（Boolean flag）が使用される場合）、「古い」チェックポイントレコードに相当するレコードを、現在のＣＣＰ値を古いＣＣＰ値の上に記憶する前に、必ず削除しなければならない。チェックポイントレコード記憶の再要求（リクレーミング）は異なる順序で行われるが、そのプロセスのこのポイントでの削除によって、後期ではなくて早期に記憶スペースが開放される。

【００２８】

ステップ５０８：不揮発性記憶装置（例えばハードディスクの装置などの磁気媒体）内にチェックポイントレコードを十分詳細につくって、チェックポイントレコードを生成するようにソースの状態を再構築する（すなわち、それは「状態を保存する」）。各チェックポイントレコードは、ソースのデータ待ち行列内の現在の「読取り」位置を含みかつチェックポイント要求メッセージからのＲＣＰのタグもしくはインデックス（例えば値「１」）がつけられている。図６は、ＲＣＰチェックポイントレコードイベントが発生する場合の、入力データ待ち行列内の潜在的に未処理のデータ６００の待ち行列内のデータレコードに対するポインターＲＣＰを示す。第二のポインターＣＣＰが、前のチェックポイント

設定イベントの前にすでに処理されたデータから潜在的に未処理のデータを分離するデータレコードを示す（ポイントする）。ＲＣＰポインターとＣＣＰポインターの間の待ち行列のデータレコードは保存された状態の一部である。次のチェックポイント設定動作のため、ＲＣＰポインターが前のＣＣＰポインターの値として処理されることに留意すべきである（これは、ＲＣＰフラグとＣＣＰフラグのための数値を使用して、値の単純な算術比較を行うことによって自動的に起こる）。

【００２９】

ステップ５１０：任意に、現在のＣＣＰ値によって示される保存されたデータ待ち行列ＣＣＰの前に出現するデータ待ち行列内に記憶スペースを再要求（リクレーム）する。多重チェックポイント設定動作が同時に進行中の場合、多重ＣＣＰポインターがあることに注目すべきである。ＲＣＰフラグおよびＣＣＰフラグのための数値を使用すると、単純な算術比較を利用することによって、対応するポインターが整合される。

【００３０】

ステップ５１２：ソースのデータ待ち行列からのデータを消費するプロセスに向かって下流へチェックポイントメッセージを伝搬させる。そのチェックポイントメッセージは、元のチェックポイント要求メッセージ由来のＲＣＰとＣＣＰを含んでいる。

【００３１】

ステップ５１４：処理を再開する。したがって、下流のプロセスが状態を保存している間、ソースはデータを受け取り、プロセスにデータを与える準備の中で他の特定用途向け機能を実行する。

【００３２】

３．プロセス－各チェックポイントメッセージを受け取ったときの各プロセス：

【００３３】

ステップ５１６：保留中の計算を完了するか、または保留中の計算を一次停止して、それをチェックポイント設定の一部として保存する。

【００３４】

ステップ５１８：受け取ったチェックポイントメッセージ各々を不揮発性記憶装置内に保存する。

【００３５】

ステップ５２０：対応するソースからの読取りを一時停止する。

【００３６】

ステップ５２２：チェックポイントメッセージをすべての接続されたソースまたは上流のプロセスから受け取ったとき（保存されたチェックポイントメッセージから確認したとき）、チェックポイント要求メッセージに示されている現在のＣＣＰ値に一致しない既存のチェックポイントレコードを削除し、次に、関連する記憶スペースを任意に再要求（リクレーム）する。

【００３７】

ステップ５２４：現在の処理状態を含みそして現在のＲＣＰ値のタブまたはインデックスをつけたチェックポイントレコードを不揮発性記憶装置内に作成する。

【００３８】

ステップ５２６：チェックポイントメッセージを、接続されたプロセスまたはシンクに向かって「下流」へ伝送する。やはり、チェックポイントメッセージは元のチェックポイント要求メッセージ由来のＲＣＰとＣＣＰを含んでいる。

【００３９】

ステップ５２８：処理を再開する。

【００４０】

４．シンク－チェックポイントメッセージをすべての接続されたプロセスまたはソースから受け取ったときの各シンク：

【００４１】

ステップ５３０：保留中の計算を完了するか、または保留中の計算を一時停止して、それ

をチェックポイント設定の一部として保存する。
【００４２】
ステップ５３２：チェックポイント要求メッセージ中に示される現在のＣＣＰ値に一致しない既存のチェックポイントレコードを削除し、次いで関連する記憶スペースを任意に再要求（リクレーム）する。
【００４３】
ステップ５３４：シンクデータの待ち行列内に現在の「発表可能な」位置を含み現在のＲＣＰ値（例えば「１」）のタグまたはインデックスをつけたチェックポイントレコードを不揮発性記憶装置内に作成する。図７は、ＲＣＰチェックポイントレコードイベントが発生する出力データ待ち行列内の発表可能なデータ７００の待ち行列内のデータレコードに対するポインターＲＣＰを示す。第二ポインターＣＣＰは、前のチェックポイント設定イベントの前にすでに発表されたデータから発表可能なデータを分離するデータレコードを示す（ポイントする）。
【００４４】
ステップ５３６：任意に、現在ＣＣＰの値によって示される、保存されたデータ待ち行列位置ＣＣＰの前に出現するデータ待ち行列内に記憶スペースを再要求（リクレーム）する。このようなデータはすでに発表されていた。多重チェックポイント設定動作が同時に進行中の場合、多重ＣＣＰポインターがあることに注目すべきである。ＲＣＰとＣＣＰフラグのための数値を使用すると、簡単な算術比較を使用することによって、対応するポインターは、一層容易に整合される。
【００４５】
ステップ５３８：まだ発表されていない既存の出力を不揮発性記憶装置に記憶させて（バッファされて）、その出力に現在ＲＣＰの値のタグをつける。このような出力はＲＣＰポインターとＣＣＰポインターの間の待ち行列にレコードを含んでいる。
【００４６】
ステップ５４０：チェックポイントメッセージをチェックポイントプロセッサに伝送する。チェックポイントメッセージはやはり、元のチェックポイント要求メッセージ由来のＲＣＰとＣＣＰを含んでいる。
【００４７】
５．チェックポイントプロセッサ－チェックポイントメッセージをすべての接続されたシンクから受け取ったときのチェックポイントプロセッサ：
【００４８】
ステップ５４２：ＣＣＰ変数の記憶された値を更新または増加させ、次いでその新しいＣＣＰ値を不揮発性記憶装置に記憶させる（ＣＣＰ変数は、チェックポイントメッセージ由来のＲＣＰ変数の現在の値に等しく設定される）。例えば、チェックポイントイベントに対するＲＣＰとＣＣＰそれぞれの値が「１、０」の場合、それらの値はこのステップの後、「１、１」となる。
【００４９】
ステップ５４４：新しいＣＣＰ値をすべてのシンクに伝送する。
【００５０】
６．シンク－各シンク：
【００５１】
ステップ５４６：受け取った新しいＣＣＰ値に等しいＲＣＰ値のタグをつけたバッファされたデータをすべて発表する。例示実施例のこのようなデータは、図７に示すＲＣＰポインター（「１」という値に相当する）とＣＣＰポインター（「０」という値に相当する）の間の待ち行列にレコードを含んでいる。
【００５２】
ステップ５４８：処理を再開する。
【００５３】
これでチェックポイント設定プロセスが終る。いくつかのステップは異なる順序で実行で

きることに注目すべきである。例えば、各シンクは、ＲＣＰチェックポイントレコードイベントが発生した後で利用可能になるデータについて、バッファされたタグ付きデータを発表する（ステップ５４６）前または未発表データを記憶した（ステップ５３８）後に処理を再開できる（ステップ５３８）。

【００５４】

故障からの回復

図８は、システムの故障がチェックポイントイベントの後に起こった時に状態を回復する方法のフローチャートである。

【００５５】

ステップ８００：すべてのプロセスを再始動する。

【００５６】

ステップ８０２：ソース、プロセス、シンクおよびチェックポイントプロセッサの間のすべての通信リンクを再確立する。

【００５７】

ステップ８０４：現在ＣＣＰ値を含む回復メッセージを、チェックポイントプロセッサから、すべてのセンサ、プロセスおよびシンクに伝送する。

【００５８】

ステップ８０６：特に、各ソース、プロセスおよびシンクが、受け取ったＣＣＰ値に対応するチェックポイントレコード内に、定義どおりにその状態を復元する。

【００５９】

各シンクは、受け取ったＣＣＰ値に対応するそのチェックポイントレコードが示す位置の前に出現するデータを発表し、そしてすでに発表したデータを再び発表しないように注意しながら、その位置の後に出現するデータを破棄する。このステップは、例えば、シンクが新しいＣＣＰ値を受け取った後であるがその保留データのレコードを発表する時間があった前に、故障が起こったときに必要である。

【００６０】

各ソースは、受け取ったＣＣＰ値に対応するそのチェックポイントレコードが示す位置に、その読取り動作を「巻き戻す」。

【００６１】

チェックポイントのトリガリング

チェックポイント動作はいくつもの方法でトリガーできる。例えば、チェックポイントの設定は、時間に基づいて定期的に実行されるか、または外部刺激（例えばネットワークの停止を保留するというネットワークオペレーティングシステムのメッセージ）に基づいている。本発明の一実施態様のチェックポイント設定は、処理されるレコード以内またはそのレコードから誘導されるデータ値に基づいている。例えば、そのデータレコードはタイムスタンプ（ＴＳ）またはブレークポイント（ＢＰ）値を含んでいる。ブレークポイントの値または範囲に対する特定のタイムスタンプを有するデータに関する計算を完了した後に、チェックポイントを設計することが望ましい。図９は、すべてのプロセス（すなわち、ソース、プロセスおよびシンク）に「知られている」予め確認されたデータ値に基づいてチェックポイント設定を調整する一方法のフローチャートである。

【００６２】

ステップ９００：各ソースが、入力データを、予め定められたＴＳ／ＢＰ値について走査する。トリガリング値が少なくとも一つのソースによって検出されたならば直ちに次のステップを実行する。

【００６３】

ステップ９０２：各ソースは保留中の計算を任意に完了する。

【００６４】

ステップ９０４：トリガーされたソースは各々「データ終了」（ＥＯＤ）の制御メッセージをすべての出力に送る。これによって、第一のソースがトリガリングＴＳ／ＢＰ値を検出して、データレコードの下流のプロセスへの提供を停止する特定のデッドロックシナリ

オが回避される。第二のソースは、トリガリングＴＳ／ＢＰ値に到達しておらず、その出力バッファを、下流のプロセスが消費するデータレコードで満たしている。これら二つのソースに接続されている消費プロセスは、第一ソース（新しいデータの提供を停止している）から次のデータが入ってくるまで無駄に待っており、そして第二ソースからのバッファされたデータレコード（その出力バッファは満たされているので、いくつものまだ処理しなければならないデータレコードのトリガリングＴＳ／ＢＰ値に到達できない）を決して消費しない。その結果、デッドロックが起こる。明示的ＥＯＤメッセージを用いることによって、下流のプロセスは、データがもはや第一ソースから入ってこないということを知らされ、その結果、このようなデータを無駄に待つことはない。

【００６５】

ステップ９０６：各ソースは制御メッセージをチェックポイントプロセッサに送って、チェックポイント設定を要求する。

【００６６】

ステップ９０８：チェックポイントプロセッサは、制御メッセージをすべてのソースから受け取ると、チェックポイント要求メッセージを発行し、そしてチェックポイント設定が上記のように進行する。

【００６７】

上記プロセスの機能を強化すると、ソースとチェックポイントプロセッサが、チェックポイント設定の開始を折衝する手続きを提供する。この手続は、出力の生成を調整しなければならない場合（例えばデータの特定の集合がチェックポイントを設定する前に生成しなければならない場合）有用であり、そしてどんな特別のＢＰ／ＴＳ値を、チェックポイント設定のトリガーとして使用しなければならないかというより進歩した知識はない。このような場合、チェックポイントプロセッサは、各ソースをポーリングして、タイムスタンプまたはブレークポイントに基づいたトリガーを決定するために使用できる一つまたは複数フィールドの現在値を確認する。次に、そのチェックポイントプロセッサは、適切な大域ＢＰ／ＴＳ値を決定してその値を各ソースにブロードキャストする。次いで、処理は、図９について記載したように進行できる。

【００６８】

本発明のいくつかの実施態様の他の側面は、それら実施態様が、出力（例えばいくつものレコードを消費して一つのレコードを生成する集合）を定期的に生成させてチェックポイントイベントを調整することによって、チェックポイント設定のオーバーヘッド負荷（overhead burden）を減らすことができることである。例えば、データを集めて発表し次にチェックポイントを実行することがより有効であり、その結果、保存すべき状態の量が減少する（例えば保存しなければならない進行中のデータが少なくなる）。したがって、チェックポイント要求メッセージの発行は、データ低減または集合操作の発表によって調整することができる。このような調整はプログラマーがセットアップすることができる。あるいは、調整は、データの低減または集合操作を実行した後、各シンクに、制御メッセージをチェックポイントプロセッサへ送らせることによって自動的にトリガーすることができる。次に、チェックポイントプロセッサは、制御メッセージを各シンクから受け取った後、チェックポイント設定動作を開始することができる。他の選択肢として、チェックポイントメッセージを使用して、データの低減または集合操作の発表をトリガーすることができる。すなわち、チェックポイントメッセージは、一グループに集められるかまたは減らされるすべてのデータレコードが受け取られたことを示すインディケータとして働く。

【００６９】

ジョブの停止

上記タイプの本発明の実施態様で、明示的「ジョブ停止」の手続きを使用することが好ましい。このような手続きは、データ処理システムの各プロセスが、処理の正常停止を、上流プロセスの故障から識別することを保証する。利用される一つの手続きは、チェックポイントプロセッサに、各ソースに処理を停止することを通知させることである。例えば、チェックポイントプロセッサは、スケジュールに基づいてまたはオペレーションシステム

メッセージなどの外部トリガーから停止するように通知される。その結果、チェックポイントプロセッサは、チェックポイント設定動作を開始して、「ジョブ停止」（ＥＯＪ）のメッセージをすべてのソースに送ることができる。ＥＯＪメッセージを送る便利な方法は、通常のチェックポイント要求メッセージにＥＯＪフラグのタグをつける方法である。ＥＯＪフラグを受け取ると、各ソースは、通常のチェックポイントルーチンを実行するが、動作を再開する代りに抜け出る。上記のように、各ソースは、チェックポイントメッセージを下流に伝搬する。各プロセスは、同様に、通常のチェックポイントルーチンを実行し、次いで動作を再開する代わりに抜け出る。チェックポイントメッセージがシンクに伝搬すると、各シンクが似かよったように通常のチェックポイントルーチンを実行する。しかし、各シンクは、チェックポイントプロセッサが、上記のように、新ＣＣＰ値を戻し、次いでそのシンクが保留データを発表した後に単に抜け出る。

【００７０】

早い発表

場合によっては、シンクが、上記手順に示されているより早くデータを発表してもよい。早い発表は、終端間の待ち時間をさらに改善する。例えば、シンクは、下記の条件下で、データレコードを早く発表することができる。

【００７１】

保留中の未発表レコード値が決定的である場合、それらレコード値はステップ５３８（図５Ｃ）の後に発表することができる。用語「決定的」は、同じデータレコードが、順序は異なるが、再始動の後に生成されることを意味する。この条件が当てはまるかどうかは、各用途の特性であり、プログラマーが決定する。故障から回復するための再始動動作の場合、シンクは、早く発表されたデータレコードを上書きした、再計算データレコードを破棄し、その結果、重複レコードは発表されない。

【００７２】

保留中の未発表レコード値が決定的なＡＮＤ順序の値である（すなわち、これらの値は単調に増加し減少する）場合、これらレコード値はチェックポイントメッセージを受け取った後、いつでも発表することができる。これによって待ち時間が一層短くなる。やはり、この条件が当てはまるかどうかは、各用途の特性であり、そしてシンクは、早く発表されたデータレコードを上書きする再始動中に計算された再計算データレコードを破棄する。

【００７３】

同じデータレコードを再発表することが許容できる場合、これらデータレコードは、チェックポイントメッセージを受け取った後、いつでも発表することができる。このような状況は、適時性が正確さより重要である場合（例えばリアルタイムが統計値を「ヒット」する場合）に生じる。

【００７４】

実行

本発明は、ハードウェアまたはソフトウェアまたは両者の組合せ（例えばプログラマブルロジックアレイ）で実行することができる。特にことわらない限り、本発明の一部として含まれているアルゴリズムは、特定のコンピュータなどの装置に本質的には関連していない。特に、各種の汎用機が、本願の教示にしたがって書きこまれたプログラムを用いて使用されるか、または要求された方法のステップを実施するためより専用的な装置をつくることが便利なことがある。しかし、本発明は、好ましくは、各々、少なくとも一つのプロセッサ、少なくとも一つのデータ記憶システム（揮発性および不揮発性のメモリおよび／または記憶素子を含む）、少なくとも一つの入力装置もしくは入力ポートおよび少なくとも一つの出力装置もしくは出力ポートを備えた一つまたは複数のプログラマブルコンピュータシステム上で実行される一つまたは複数のコンピュータプログラムで実施される。そのプログラムレコードは、プロセッサで実行されて本願に記載の機能を実行する。

【００７５】

このようなプログラムは各々、望ましいコンピュータ言語（機械、アセンブリ、または高レベル手続、論理もしくはオブジェクト指向プログラムミング言語を含む）で実施されて

、コンピュータシステムと通信する。いずれの場合も、その言語は、コンパイルされた言語かまたはインタプリタされた言語である。

【００７６】

このようなコンピュータプログラムは、好ましくは汎用または専用のプログラマブルコンピュータシステムが読み取ることができる記憶媒体または記憶装置（例えば、ＲＯＭ、ＣＤ－ＲＯＭ、磁気媒体または光媒体）に記憶されている。このコンピュータプログラムの記憶は、記憶媒体または記憶装置がコンピュータシステムに読み取られて、本願に記載されている手続きが実行されるとき、コンピュータを構成しかつ作動させるためになされている。また、本発明のシステムは、コンピュータプログラムで構成された、コンピュータ読取り可能記憶媒体として実施することも考えられる。この場合、このように構成された記憶媒体はコンピュータシステムを特別の予め定義された方式で作動させて本願に記載の機能をなしとげる。

【００７７】

本発明のいくつもの実施態様を説明してきた。しかしながら、各種の変形を、本発明の精神と範囲を逸脱することなく行うことができると解される。例えば、上記のいくつもの機能ステップは全処理に対して実質的に影響することなく、異なる順序で実施することができる。したがって、他の実施態様は本願の特許請求の範囲内に入っている。

【図面の簡単な説明】

【図１】　本発明とともに使用できる従来技術のデータ処理システムのブロック図である。

【図２】　本発明の連続フローデータ処理システムのブロック図である。

【図３】　本発明の一プロセスステージのブロック図である。

【図４】　本発明が使用するのに適切なデータ待ち行列の線図である。

【図５Ａ】　連続フローチェックポイント設定データ処理システムのプロセスステージにおけるチェックポイントを開始する方法の一実施態様を示すフローチャートである。

【図５Ｂ】　連続フローチェックポイント設定データ処理システムのプロセスステージにおけるチェックポイントを開始する方法の一実施態様を示すフローチャートである。

【図５Ｃ】　連続フローチェックポイント設定データ処理システムのプロセスステージにおけるチェックポイントを開始する方法の一実施態様を示すフローチャートである。

【図５Ｄ】　連続フローチェックポイント設定データ処理システムのプロセスステージにおけるチェックポイントを開始する方法の一実施態様を示すフローチャートである。

【図６】　連続フローチェックポイント設定データ処理システムのための入力データ待ち行列の線図である。

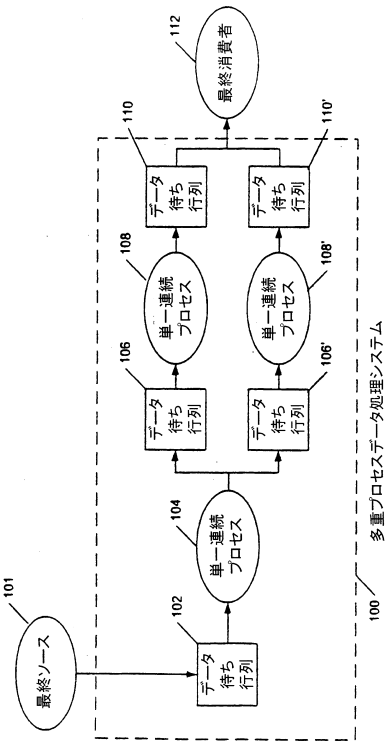
【図７】　連続フローチェックポイント設定データ処理システムのための出力データ待ち行列の線図である。

【図８】　システムの故障がチェックポイント設定イベントの後に起こったときに状態を回復する方法のフローチャートである。

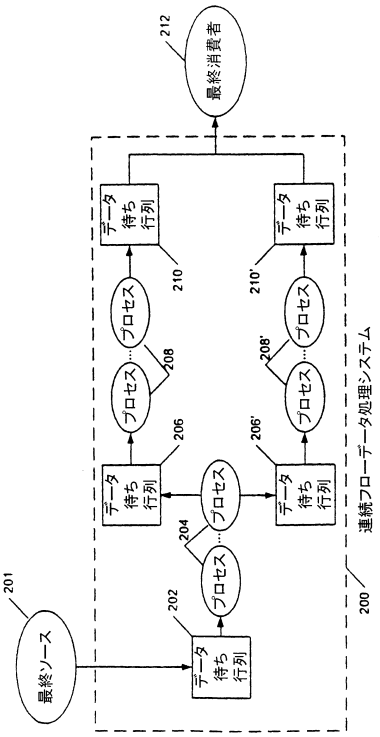
【図９】　データ値に基づいてチェックポイント設定を調整する一方法のフローチャートである。

各種図面中の同じ参照番号と表示は同じ素子を示す。

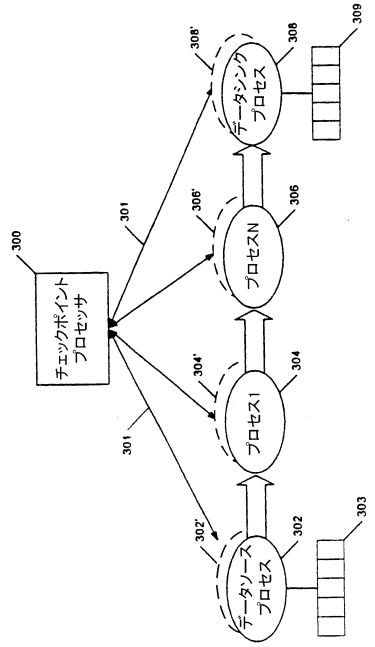
【図１】


最終ソース
101
データ待ち行列
102
単一連続プロセス
104
データ待ち行列
106
データ待ち行列
106'
単一連続プロセス
108
単一連続プロセス
108'
データ待ち行列
110
データ待ち行列
110'
最終消費者
112
多重プロセスデータ処理システム
100

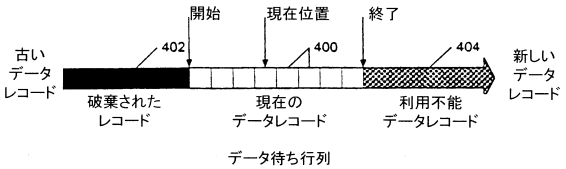
【図２】


最終ソース
201
データ待ち行列
202
プロセス
204
データ待ち行列
206
プロセス
208
データ待ち行列
210
プロセス
208'
データ待ち行列
206'
プロセス
204'
データ待ち行列
210'
最終消費者
212
連続フローデータ処理システム
200

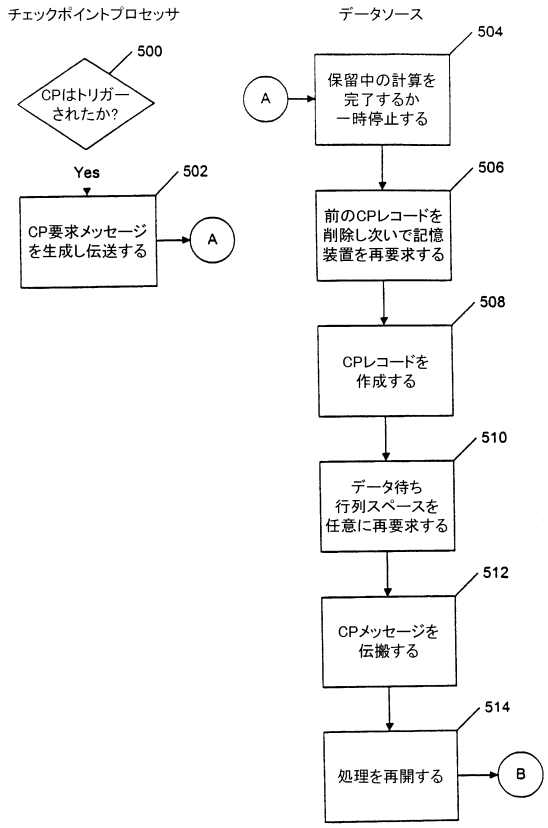
【図３】


チェックポイントプロセッサ
300
301
データソースプロセス
302
302'
303
プロセス1
304
304'
プロセスN
306
306'
データシンクプロセス
308
308'
309

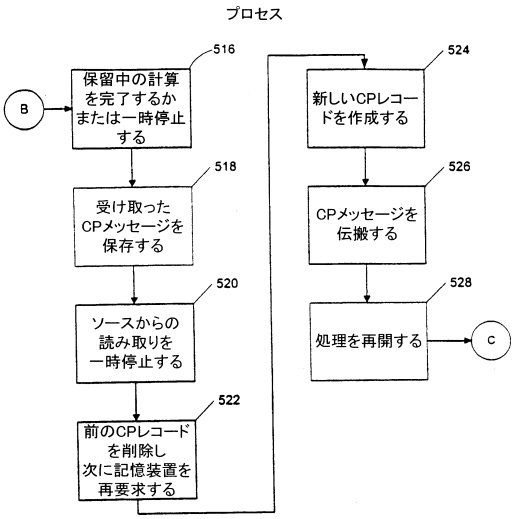
【図４】


開始
現在位置
終了
古いデータレコード
402
400
404
新しいデータレコード
破棄されたレコード
現在のデータレコード
利用不能データレコード
データ待ち行列

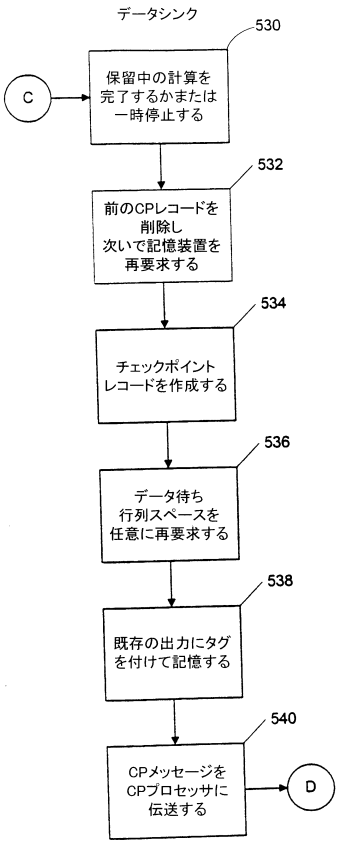
【図５Ａ】



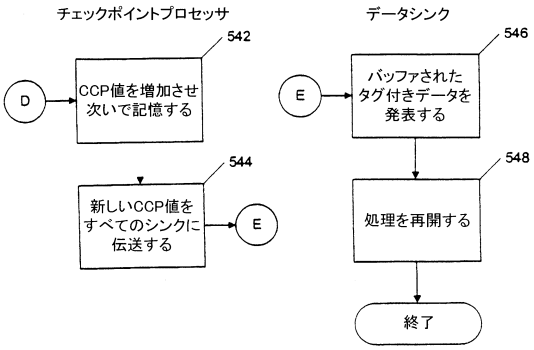
【図５Ｂ】



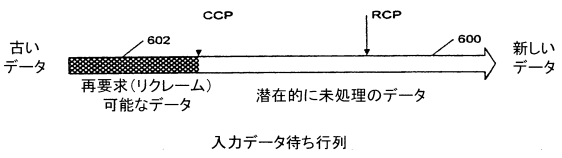
【図５Ｃ】



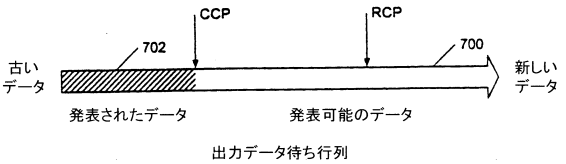
【図５Ｄ】



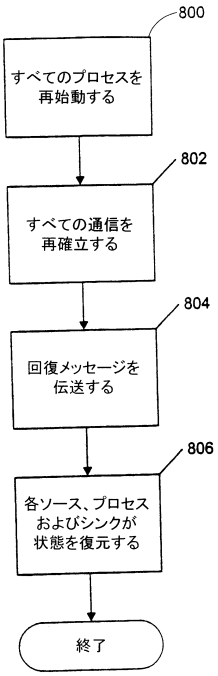
【図６】



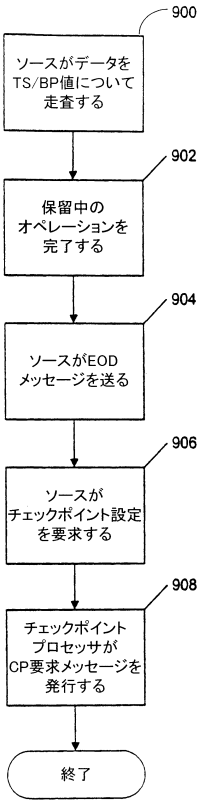
【図７】


CCP
RCP
702
700
古い
データ
新しい
データ
発表されたデータ
発表可能のデータ
出力データ待ち行列

【図８】


800
すべてのプロセスを
再始動する
802
すべての通信を
再確立する
804
回復メッセージを
伝送する
806
各ソース、プロセス
およびシンクが
状態を復元する
終了

【図９】


900
ソースがデータを
TS/BP値について
走査する
902
保留中の
オペレーションを
完了する
904
ソースがEOD
メッセージを送る
906
ソースが
チェックポイント設定
を要求する
908
チェックポイント
プロセッサが
CP要求メッセージを
発行する
終了

フロントページの続き

(72)発明者　ヴィシュニアック　ユーフライム
アメリカ合衆国　マサチューセッツ州　０２４７４　アーリントン　ゴーハム　ストリート　３８
(72)発明者　ベイ　ポール
アメリカ合衆国　マサチューセッツ州　０２４７４　アーリントン　ゴーハム　ストリート　７

審査官　坂庭　剛史

(56)参考文献　特開平０９－１３８７５４（ＪＰ，Ａ）
特開昭６４－０８８８４６（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 11/28
G06F 11/14
G06F 9/46