(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3748339号
(P3748339)
(45)発行日　平成18年2月22日(2006.2.22)　(24)登録日　平成17年12月9日(2005.12.9)

(51)Int.Cl.　F I
G06F 9/52 (2006.01)　G06F 9/46 475C

請求項の数 3 (全 17 頁)

(21)出願番号　特願平11-120000
(22)出願日　平成11年4月27日(1999.4.27)
(65)公開番号　特開平11-345130
(43)公開日　平成11年12月14日(1999.12.14)
審査請求日　平成13年10月22日(2001.10.22)
(31)優先権主張番号　069481
(32)優先日　平成10年4月29日(1998.4.29)
(33)優先権主張国　米国(US)

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
HEWLETT-PACKARD COMPANY
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　3000
(74)代理人　100081721
弁理士　岡田　次生
(72)発明者　スティーブン・ジェイ・コンスタント
アメリカ合衆国80525コロラド州フォート・コリンズ、ウォーターフォード・レーン　1636

最終頁に続く

(54)【発明の名称】複数のデータストアの同期をとってデータ整合性を達成する方法

(57)【特許請求の範囲】

【請求項１】

非トランザクションの多重処理コンピュータシステムにおいて、複数のデータストアの同期をとってデータ整合性を達成する方法であって、

前記システムは、複数のデータストアと、１つまたは複数のデータ入力ソースと、該１つまたは複数のデータ入力ソースのうちの少なくとも１つのデータ入力ソースおよび該複数のデータストアのうちの少なくとも１つのデータストアの間の従属パス上にある複数のプロセスと、を有しており、該複数のプロセスのうちの少なくとも１つのプロセスは、該複数のプロセスのうちの少なくとも１つの他のプロセスに対してデータ従属性を持っており、

前記複数のプロセスのうちの任意の他のプロセスによって呼び出されず、かつ、前記１つまたは複数のデータ入力ソースのうちの少なくとも１つからデータを直接受け取るプロセスを含む第１のサブセットを中断するステップであって、該第１のサブセットのプロセスからのデータのすべてを、前記データストアのうちの１つまたは複数に、または中断されていない１つまたは複数のプロセスに流すことにより、該中断を達成する、ステップと、

前記データ従属性において前記第１のサブセットのプロセスに続くプロセスのサブセットであって、前記複数のプロセスのうち、中断されたプロセスによって呼び出され、かつ、中断されていないプロセスによっては呼び出されないプロセスを少なくとも１つ含むサブセットを、中断されないまま残っているプロセスが無くなるまで順番に中断するステッ

プであって、該サブセットのプロセスからのデータのすべてを、前記データストアのうちの１つまたは複数に、または中断されていない１つまたは複数のプロセスに流すことにより、該中断を達成する、ステップと、
を含む、方法。
【請求項２】
非トランザクションの複数データストア処理システムであって、
複数のデータストアと、
１つまたは複数のデータ入力ソースと、
前記１つまたは複数のデータ入力ソースのうちの少なくとも１つのデータ入力ソース、および前記複数データストアのうちの少なくとも１つのデータストアの間の従属パス上にある複数のプロセスと、を備え、
前記複数のプロセスのうちの少なくとも１つのプロセスは、該複数のプロセスのうちの少なくとも１つの他のプロセスに対してデータ従属性を持っており、
前記複数のプロセスのそれぞれは、前記複数のデータストアの修正を生じさせるサービス要求の受け入れを中断し、該複数のデータストアの修正を生じさせるすべての待ち状態タスクを完了させ、前記複数のプロセスのうちの中断されてないプロセスまたは前記複数のデータストアのうちの１つまたは複数に、内部的にキャッシュに格納されたデータのすべてを流す手段を有しており、
該システムは、さらに、前記複数のプロセスのそれぞれを順番に中断させるよう動作可能な中央管理機構であって、前記複数のプロセスのうちの任意の他のプロセスによって呼び出されず、かつ、前記１つまたは複数のデータ入力ソースのうちの少なくとも１つからデータを直接受け取るプロセスを含む第１のサブセットを中断させ、さらに、前記データ従属性において該第１のサブセットのプロセスに続くプロセスのサブセットであって、前記複数のプロセスのうち、中断されたプロセスによって呼び出され、かつ、中断されていないプロセスによっては呼び出されないプロセスを少なくとも１つ含むサブセットを、中断されないまま残っているプロセスが無くなるまで順番に中断させる、中央管理機構を備えており、
該中央管理機構は、前記第１のサブセットの前記中断を、該第１のサブセットのプロセスに、該第１のサブセットのプロセスからのデータのすべてを前記データストアのうちの１つまたは複数に、または中断されていない１つまたは複数のプロセスに流させることによって達成し、さらに、該第１のサブセットのプロセスに続くプロセスのサブセットの前記中断を、該サブセットのプロセスに、該サブセットのプロセスからのデータのすべてを前記データストアのうちの１つまたは複数に、または中断されていない１つまたは複数のプロセスに流させることによって達成する、
非トランザクションの複数データストア処理システム。
【請求項３】
非トランザクションの複数データストア処理システムにおける複数のデータストアの同期をとる方法であって、該システムは、
複数のデータストアと、
１つまたは複数のデータ入力ソースと、
前記１つまたは複数のデータ入力ソースのうちの少なくとも１つのデータ入力ソースおよび前記複数のデータストアのうちの少なくとも１つのデータストアの間の従属パス上にある複数のプロセスと、を備えており、
前記複数のプロセスのうちの少なくとも１つは、該複数のプロセスのうちの少なくとも１つの他のプロセスに対してデータ従属性を持っており、
前記複数のプロセスのそれぞれは、前記複数のデータストアの修正を生じさせるサービス要求の受け入れを中断し、該複数のデータストアの修正を生じさせる待ち状態タスクのすべてを完了し、内部的にキャッシュされたデータのすべてを、前記複数のプロセスのうち中断されていないプロセス、または前記複数のデータストアのうちの１つまたは複数に流す手段を有しており、

該同期をとる方法は、

中断要求に応答して、前記複数のプロセスのうちの任意の他のプロセスによって呼び出されず、かつ、前記１つまたは複数のデータ入力ソースのうちの少なくとも１つからデータを直接受け取るプロセスを含む第１のサブセットを中断するステップであって、該第１のサブセットからのデータのすべてを、前記データストアのうちの１つまたは複数に、または中断されていない１つまたは複数のプロセスに流すことによって、該中断を達成するステップと、

前記データ従属性において前記第１のサブセットのプロセスに続くプロセスのサブセットであって、前記複数のプロセスのうち、中断されたプロセスによって呼び出され、かつ中断されていないプロセスによっては呼び出されないプロセスを少なくとも１つ含むサブセットを、中断されないまま残っているプロセスが無くなるまで順番に中断するステップであって、該サブセットのプロセスからのデータのすべてを、前記データストアのうちの１つまたは複数に、または、中断されていない１つまたは複数のプロセスに流すことによって、該中断を達成する、ステップと、

を含む、方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

この発明は、一般に多重処理コンピュータシステムに関し、より具体的には、単一非トランザクション処理システムの複数のデータストア間に整合性および同期を達成するシステムおよび方法に関する。

【０００２】

【従来の技術】

多重処理コンピュータシステムは、同時に複数のプロセスを実行する。それぞれのプロセスは特定のタスクを実行し、複数のプロセスは、全体としてとらえた場合、より大きなタスクを実行し、これがアプリケーションと呼ばれる。これらのプロセスは、１つの中央コンピュータで実行することができ、または通信リンクを介して互いに接続された個別のコンピュータで走ることもできる（すなわち、分散環境）。多重処理コンピュータシステムは、しばしば複数のデータストアを備える。典型的に、それぞれのデータストアは、システムの１またはわずかの複数のプロセスにより直接アクセスされる。

【０００３】

アプリケーションが正確に機能するため、個々のプロセスは、プロセス間通信を介して協調されなければならない。通常、プロセス間通信はメッセージ通信システムとして実現される。メッセージ通信システムは、他のプロセスにメッセージを送り、他のプロセスからメッセージを受け取るプロセスにより特徴づけられる。複数のプロセスのうちのあるプロセスで失敗が起こると、それぞれのプロセスが他のプロセスの成功した操作に依存するので、しばしばアプリケーション全体を再度初期化しなければならない。このような場合には、それぞれのプロセスを、実行の初めの部分までロールバックしなければならない。

【０００４】

複数のデータストアの多重処理システムでは、システム・アプリケーションが、異なる時に複数のデータストアを更新することができる。したがって、任意の所与の時に、アプリケーションの現在の進行を反映するようデータストアのうちいくつかは更新されているが、いくつかは更新されていない（これは、まだ完了していないアプリケーションの一部を反映する）という事実のために、様々なデータストアが不整合な状態になることがある。アプリケーションのなかには、データの不整合性が問題となるものがある。たとえばバックアップのアプリケーションは、異なるデータストアのデータが、バックアップ媒体から復元される時に機能システムをサポートすることができる状態にあるよう要求する。これを実現するには、バックアップ中、データストアは整合性ある状態でなければならない。

【０００５】

複数のデータストア間の整合性および同期は、トランザクション処理を介して、いくつか

の従来技術のシステムで達成されてきた。トランザクション処理システムでは、アプリケーションは複数の「トランザクション」から構成される。トランザクションは、特定のシーケンスで行われる一連の独立した操作であり、特定の目的を達成する。トランザクションは、そのトランザクションにあるそれぞれの操作が完了するまで完了しない。ある操作を、他のプロセスで実行することができる。したがって、あるプロセスが他のプロセスを呼び出すと、呼び出したプロセスは、呼び出されたプロセスが完了するまで中断する。こうして、トランザクション処理システムは、一組の操作が自律している、すなわち一組の操作が集団として成功または失敗することを保証する。したがって、特定のトランザクションにより定義された一組の操作に含まれる操作のうちの１つが失敗すると、呼び出しの逆の順番で、つまりそれが実行されたのとは逆のシーケンスで、それぞれの操作を取り消す（アンドゥする）ことにより、トランザクション全体が整合性ある状態に容易にロールバックされる。

【０００６】

非トランザクション多重処理システムでは、アプリケーションは「高レベル操作（High Level Operation；ＨＬＯ）」を実行することができる。ＨＬＯは、一組の低レベルプロセスにより提供されるサービスを使用することにより達成される一連のタスクとして定義される。ＨＬＯは、複数のプロセス（それぞれが低レベルタスクを実行する）が、より大きいタスクを達成するよう協調するという点で、トランザクションシステムにおけるトランザクションと類似する。しかし、特定の順番で一組のプロセスを通じてシーケンスするトランザクションとは異なり、システムの複数のデータストア間に整合性ある状態を復元するのに、必ずしも操作シーケンスを逆にする必要がない。さらに、複数のＨＬＯは、同時かつ非同期に実行することができ、どのＨＬＯがどの順番で任意の所与のデータストアを更新したかを追跡するための備えが全く存在しない。

【０００７】

【発明が解決しようとする課題】

非トランザクションの複数データストアの多重処理システムでは、複数のデータストア間に整合性および同期を達成するための異なるアプローチが必要とされる。今日の非トランザクションの複数データストア処理システムでは、同期およびデータ整合性を、システム全体をシャットダウンする（電源を落とす）ことによってのみ達成することができる。しかし、サービスと時間の損失を招くこのアプローチは、ユーザには不都合であり、許容しがたい。したがって、システムを走らせたたままにすることができて、ユーザに対する時間およびサービスの損失を最小する非トランザクション処理システムの複数のデータストア間に同期および整合性を達成するシステムおよび方法が必要とされる。

【０００８】

【発明を解決するための手段】

この発明は、非トランザクションの多重処理コンピュータシステムにおいて、複数のデータストアの同期をとり、データ整合性を達成するシステムおよび方法であり、従来技術の問題点を解決する。

【０００９】

システムは、複数のデータストアと、１または複数のデータ入力ソースと、少なくとも１つのデータ入力ソースおよび少なくとも１つのデータストアの間の従属パス（経路）にあるすべてのプロセスを有する従属木に含まれる複数同時に実行するプロセスとを備える。従属木にあるプロセスの第１のサブセットが中断される。プロセスが中断するとき、そのプロセスは、任意のデータストアの修正を生じさせるサービス要求を受け取るのを停止し、任意のデータストアの修正を生じさせるすべての待ち状態(ペンディング：pending)タスクを完了させ、中断されていないプロセスまたは１または複数のデータストアに、内部的にキャッシュに格納されたデータのすべてを流す。第１のサブセットは、従属木の任意の他のプロセスにより呼び出されておらず、少なくとも１つのデータ入力ソースから直接データを受け取る従属木のプロセスのすべてを含む。従属木にある次のプロセスのサブセットが、次に中断される。次のサブセットは、すでに中断されたプロセスにより呼び出さ

れており、まだ中断されていないプロセスにより呼び出されていない少なくとも１つのプロセスを含む。それぞれのプロセスが中断されるとき、または代わりに、プロセスのそれぞれのサブセットがすべて中断されるとき、他の次のサブセットが同じ基準に基づいて選択され、その後中断される。この処理が、従属木のプロセスのすべてが中断されるまで続く。従属木のすべてのプロセスが中断された時、複数のデータストアのそれぞれにあるデータは同期がとられ、整合性がある。

【００１０】

システムの通常の操作は、中断されたのとは逆の順番で、中断されたプロセスのそれぞれを再開することにより、再開される。従属木にある最後のプロセスのサブセットが、再開される。最後のサブセットは、少なくとも１つのデータストアを直接アクセスする従属パスにあるすべてのプロセスを含む。従属木にある前のサブセットが、次に中断される。前のサブセットは、１または複数のすでに再開されたプロセスを呼び出しており、任意の現在中断されているプロセスを呼び出さない少なくとも１つのプロセスを含む。それぞれのプロセスが再開されるとき、または代わりに、プロセスのそれぞれのサブセットがすべて再開されるとき、他の前のサブセットが同じ基準に基づいて選択され、その後再開される。この処理が、従属木にあるプロセスのすべてが再開されるまで続く。データストアのうち少なくとも１つのデータストアの修正を生じさせるサービス要求を受け取り始めることにより、プロセスは再開する。

【００１１】

同期手段と通信できないが、少なくとも１つのデータ入力ソースと少なくとも１つのデータストアとの間の従属パスにある間接プロセスを含むシステムでは、代理プロセスが提供され、中断要求を、間接プロセスが応答する代理中断要求に翻訳する。

【００１２】

【発明の実施の形態】

図１は、非トランザクションの多重データストア処理システム１００の例を示すブロック図である。システム１００は、複数のプロセス１０２、１０４、１０６、１０８、１１２、１３０、１５０、１５２ａ、１５２ｂ、１５２ｃ、１６０、１６２ａおよび１６２ｂを含み、これらは互いに様々な相互従属関係を持つ。プロセス１０２、１０４、１０６、１０８、１１２、１３０、１５０、１５２ａ、１５２ｂ、１５２ｃ、１６０、１６２ａおよび１６２ｂは、同じ環境または分散環境で同時に実行される。また、システム１００は、複数のデータストア１２０～１２８を有し、これらは、複数のプロセスのうちの１つまたはわずかなプロセスにより直接アクセスされる。図に示される実施形態では、データストア１２２が、プロセス１０６によってのみ直接アクセスされ、データストア１２４は、プロセス１０２および１０８によってのみ直接アクセスされ、データストア１２６は、プロセス１１０および１５２ａ、１５２ｂ、１５２ｃによってのみ直接アクセスされ、データストア１２８は、プロセス１１０および１１２によってのみ直接アクセスされ、データストア１２０は、プロセス１６２ａおよび１６２ｂによってのみ直接アクセスされる。

【００１３】

システム１００では、複数の、独立しているが関連したデータストア１２０～１２８に、データが保持される。データの複製は、データストア１２０～１２８によっては全く提供されない。さらに、プロセス１０２、１０４、１０６、１０８、１１２は自律して動作し、個々のプロセス間の操作の同期をとるためのトランザクション制御が全く存在しない。

【００１４】

システム１００は、中央管理プロセス１３０を備える。中央管理プロセスは、プロセス１０２、１０４、１０６、１０８、１１２、１５０および１６０の間の相互従属関係を理解するよう構成される。中央管理プロセス１３０は、プロセス１０２、１０４、１０６、１０８、１１２のそれぞれの相互従属関係に従う順番で、プロセス１０２、１０４、１０６、１０８、１１２のそれぞれを中断し、また再開する手段を備える。これにより、異なるデータストアが確実に同期がとられ、データストア１２０～１２８のそれぞれのデータ整合性が維持される。

【００１５】
データストアを修正し、および（または）同期を要求するが、中央管理プロセス１３０と対話しないまたはできないプロセスをシステム１００は含むことができる。この種のプロセス（以下、「間接プロセス」という）を、間接プロセス１５２ａ、１５２ｂ、１５２ｃ、１６２ａおよび１６２ｂとしてシステム１００に示す。間接プロセスは、それらが中央管理プロセス１３０により認識されるものとは異なる通信プロトコルを実行するので、「間接」という。プロセス１５２ａ、１５２ｂおよび１５２ｃは、同一プロセスＡの３つの別々の例である。たとえば、プロセス１５２ａ、１５２ｂおよび１５２ｃは、ユーザーインターフェース・プロセスの３つの異なるオペレータ起動の別々のインスタンス（具象化）の結果であることができる。同様に、プロセス１６２ａおよび１６２ｂは、別の同一プロセスＢの２つの別々のインスタンスである。

【００１６】
中央管理プロセス１３０が、間接プロセス１５２ａ、１５２ｂおよび１５２ｃ、または１６２ａおよび１６２ｂと直接通信できないので、管理代理プロセス１５０および１６０を使用して、中央管理プロセス１３０と、それぞれの間接プロセス１５２ａ、１５２ｂ、１５２ｃ、１６２ａおよび１６２ｂとの間のインターフェースをとる。管理代理プロセス１５０および１６０は、中央管理プロセス１３０からの「中断」通知および「再開」通知を、それぞれの間接プロセス１５２ａ、１５２ｂ、１５２ｃ、１６２ａおよび１６２ｂにより認識される通信プロトコルを使用して、代理の「中断」通知および「再開」通知に翻訳する。

【００１７】
複数の同時に走る非同期のＨＬＯのために、データが複数のデータ入力ソースから流れることがある。例として、システムは、データストアを変更するためのソースとしてユーザー入力を受け取ることができ、それと同時に、システムが、１または複数の異なるデータストア１２０～１２８に対する修正を生じさせるシステム事象を獲得することができるとする。これらのデータ入力ソース１４２～１４６のそれぞれは、システム１００の異なるパスを通るデータフロー（したがって、異なる順番のプロセス）を生成する。

【００１８】
中央管理プロセス１３０は、任意の所与の時に、異なるプロセス１０２、１０４、１０６、１０８、１１２の間で何が起こっているかを追跡しない。しかし、中央管理プロセス１３０は、入力データソースとデータストアとの間の従属パスにどのプロセスがあるかを追跡するよう、また、従属関係にある様々なプロセスによりデータが操作される正しいシーケンスを追跡するよう構成される。

【００１９】
図に示される実施形態では、プロセスの従属関係の宣言から抽出される従属木を生成することにより、これが達成される。それぞれのプロセス１０２、１０４、１０６、１０８、１１２、１５０および１６０は、他のプロセスのそれぞれに関してその従属関係を宣言する。プロセスの設計者は、システムの他のプロセスおよび他のデータストアに対するプロセスの関係を理解する。それぞれのプロセスの宣言された相互従属関係は、相互従属構成ファイルに格納される。中央管理プロセス１３０は、相互従属構成ファイルを読み、システム１００の様々なプロセスの構造を反映する従属木を数学的に生成する。

【００２０】
提供された従属木の結果として、プロセス１０２は、プロセス１０８および１１２にコールすることを宣言する。プロセス１０４は、プロセス１０８および１１０にコールすることを宣言する。プロセス１０６は、プロセス１１０にコールすることを宣言する。プロセス１０８は、他のプロセスをコールしないことを宣言する。プロセス１１０は、間接プロセス１５２ａ、１５２ｂおよび１５２ｃにコールすることを宣言する。プロセス１１２は、他のプロセスを呼び出さないことを宣言する。管理代理プロセス１５０は、間接プロセス１５２ａ、１５２ｂおよび１５２ｃにコールすることを宣言する。管理代理プロセス１６０は、間接プロセス１６２ａおよび１６２ｂにコールすることを宣言する。

【００２１】
図２は、プロセス１０２～１１２、１５０および１６０の従属宣言から作られるシステム１００の異なるプロセス間の相互従属関係を示す従属木２００である。従属関係は、矢印の方向で示される。プロセス１５２ａ、１５２ｂ、１５２ｃ、１６２ａおよび１６２ｂは、中央管理プロセス１３０と直接通信しないので、図２には含まれない。プロセス１１０は、間接プロセス１５２ａ、１５２ｂおよび１５２ｃに依存するので、プロセス１５０に依存する。プロセス１０６および１０４は、プロセス１１０に依存する。プロセス１０４および１０２は、それぞれプロセス１０８に依存する。プロセス１０２は、プロセス１１２に依存する。プロセス１０２、１０４、１０６および１６０は、その他のプロセスには全く依存しない。

【００２２】
この発明によると、同期およびデータ整合性は、異なるプロセス間で定義された相互従属関係（すなわち、従属木２００のプロセスの位置のレベル）に従う順番で、プロセス１０２～１１２および１５０、１６０を中断および再開することにより達成される。異なるプロセス間の相互従属関係は、どのＨＬＯがシステム１００により実行されるかにかかわりなく同じである。中断および再開の順番は、プロセスの動作のしかたに固有である。すなわち、どのデータストアをそれぞれの特定のプロセスが修正するのか、どのプロセスがデータ入力ソースから直接入力を受け取るのか、従属木のどこにそれぞれのプロセスが属するのか、による。中央管理プロセス１３０が、プロセス１０２、１０４、１０６、１０８、１１２、１５０および１６０のそれぞれの相互従属関係を理解し、それぞれと通信するので、中央管理プロセス１３０は、異なるプロセス１０２～１１２、１５０および１６０の中断操作の順番を制御する能力を持つ。

【００２３】
この発明の「中断」操作は、トランザクション処理システムの停止操作と近似している。したがって、中央管理プロセス１３０は、通常のシャットダウン中にそれぞれのプロセスを停止させるのと同じ順番でプロセスのそれぞれと連絡をとり、停止以外のすべてを行うよう告げる。それぞれのプロセスが中断される順番は、停止が命じられる順番に直接マップされる。すなわち、定義されたプロセスの相互従属関係に従う。

【００２４】
代理管理プロセス１５０または１６０が、中央管理プロセス１３０から「中断」通知を受け取るとき、通信するそれぞれの間接プロセスに代理「中断」通知を送る。それぞれの管理代理プロセス１５０および１６０は、管理代理プロセスのそれぞれの間接プロセスの従属関係を理解し、従属関係の順番でそれらを中断する。間接プロセス１５２ａ、１５２ｂおよび１５２ｃ、または１６２ａおよび１６２ｂは、代理「中断」通知を受け取るとき、可能ならば自分自身を中断し、それぞれの管理代理プロセス１５０または１６０にその中断成功状態を知らせる。間接プロセスが、中断操作を実行する際に成功しないことがあるのは、間接プロセスが、要求された時に中断することのできないクライアント（すなわち、間接プロセスを潜在的に呼び出す他のプロセス）を持つ場合である。間接プロセスのクライアントのうちいずれかが、要求された時に中断できないならば、間接プロセスは適切な管理代理プロセス１５０または１６０に知らせる。その後、管理代理プロセス１５０または１６０は、中央管理プロセス１３０に応答する。中央管理プロセス１３０の責任は、中央管理プロセス１３０が中断するよう指図した間接プロセスに、「再開」通知を発行することである。その後、中央管理プロセス１３０は中断されたプロセスを再開し、同期を要求したＨＬＯに知らせることにより、成功しなかった中断に応答する。

【００２５】
当該技術分野の当業者により理解されるように、この発明は、本質的に、システムへのデータ入力ソースを遮断し、進行中の現在のデータフローを、異なるデータストアにつながる様々なデータフローのパスに流すよう動作する。このプロセスが完了するとき、複数のデータストアのそれぞれは同期が取られ、データは整合性ある状態になる。この発明は、入力データソースの数、異なるデータフローのパスおよびプロセスの従属関係にかかわり

なく、何らかの同様にモデル化されたシステムにおける整合性を保証するよう動作する。
【００２６】
図３は、この発明によるシステムの従属パスにある複数のプロセスを中断する方法３００のフローチャートである。方法３００は、従属木にあるそれぞれのプロセスが、中央管理プロセス１３０から「中断」通知を受け取って動作する手段を含むことを必要とする。「中断」通知を受け取った後にそれぞれのプロセスが行う詳細は、特定のプロセスの機能に依存し、したがってプロセスごとに異なることがある。しかし、「中断」通知を受け取る時、それぞれのプロセスは、最小限データストアの修正を生じさせるサービス要求またはコマンドを受け取るのを停止し、データストアを修正するタスクのすべての待ち状態タスクを完了させ、内部的にキャッシュに格納されたデータをデータストアに流さなければならない。プロセスがこれらのステップ（および、その特定のプロセスの中断機能が実行するよう設計されているどんな他のステップも）を完了すると、プロセスは、「中断」通知の要求を終えたということを、「中断完了」応答を介して中央管理１３０に通知する。システムのどのプロセスが、待ち状態の管理操作を通知されなければならないか、およびどの順番でプロセスに通知するかを理解するのは、中央管理プロセス１３０の責任である。
【００２７】
この発明の方法３００によると、システムの異なるデータストアの同期をとらなければならないとき、中央管理プロセス１３０は、ステップ３０２で、従属木にあり、任意の他のプロセスにより呼び出されておらず、少なくとも１つのデータ入力ソースからデータ入力を受け取るためシステムのソースの最初の点として動作するそれぞれのプロセスを中断する。
【００２８】
１または複数のプロセスが中断されると、中央管理プロセス１３０は、ステップ３０４で、中断されたプロセスにより潜在的に呼び出され、まだ中断されていないプロセスにより潜在的に呼び出されないプロセスであって、残りの中断されていないプロセスを含む次のプロセスのサブセットを決定する。
【００２９】
ステップ３０６では、中央管理プロセス１３０は、次のプロセスのサブセットのそれぞれを中断する。ステップ３０８では、中央管理プロセス１３０は、中断されずに残っているプロセスが従属木にあるかどうか判断する。残っていなければ、方法３００は完了する。そうでなければ、ステップ３０４から３０８が、従属木のそれぞれのプロセスが適切な順番で中断されるまで繰り返され、すべてのデータが、上記の異なるデータストアに流されるようにする。
【００３０】
方法３００をシステム１００に適用すると、プロセス１０２、１０４、１０６、１０８、１１２、１５０および１６０が中断される順番は、以下のようになる。１）プロセス１０２、１０４、１０６および１６０を中断する。２）プロセス１０２が中央管理プロセス１３０に応答するとき、プロセス１１２を中断する。３）プロセス１０２および１０４の両方とも中断を完了して、中央管理プロセス１３０に応答する時、プロセス１０８を中断する。４）プロセス１０４および１０６が両方とも中断を完了して中央管理プロセス１３０に応答する時、プロセス１１０を中断する。５）プロセス１１０が中断を完了して中央管理プロセス１３０に応答する時、プロセス１５０を中断する。６）すべての「中断完了」応答が中央管理プロセス１３０により受け取られた時、システムが中断する。
【００３１】
図４は、この発明によるシステムの従属木にある複数の中断されたプロセスを再開する方法４００のフローチャートである。方法４００は、従属木にあるそれぞれのプロセスが、中央管理プロセス１３０から「再開」通知を受けて動作する手段を含むことを必要とする。「再開」通知を受け取るとき、受け取りのプロセスは、サービス要求またはコマンドの受け取りおよび処理を始める。「再開」通知は「開始」通知と同じ順番で送られ、「中断」通知の逆の順番である。プロセスがこのステップ（および、プロセスの再開機能を完了

するためプロセスが完了しなければならない任意の他のステップ）を完了すると、プロセスは、「再開完了」応答を介して中央管理プロセス１３０に知らせる。システム１００のどのプロセスが、どの順番で、プロセスのそれぞれを再開するかを理解するのは、中央管理プロセス１３０の責任である。好ましい実施形態では、これは従属木を介して判断される。

【００３２】

方法４００によると、システムの、異なる中断されたプロセスが再開されるとき、中央管理プロセス１３０は、ステップ４０２で、従属木のプロセスのうち現在中断されているプロセスを呼び出さないそれぞれのプロセスを再開することから始める。これは、従属木の一番低いレベルのプロセスに「再開」通知を送ることにより達成される。中央管理プロセス１３０は、これらのプロセスのうち少なくとも１つのプロセスから「再開完了」応答を受け取るまで待つ。

【００３３】

従属木の一番低いレベルに置かれた１または複数のプロセスが再開されると、ステップ４０４で、中央管理プロセス１３０は、再開されたプロセスを潜在的に呼び出し、まだ中断されているプロセスを潜在的に呼び出さないプロセスであって、残っている中断されたプロセスを含む次のプロセスのセットを決定する。例示される実施形態では、従属木の次のレベルにあるそれぞれのプロセスがチェックされ、潜在的に呼び出すプロセスのそれぞれが、「再開完了」応答を返したかどうか判断する。

【００３４】

ステップ４０６では、中央管理プロセス１３０が、次のプロセスのセットのそれぞれを再開する。例示される実施形態では、これは、次のプロセスのセットのそれぞれに「再開」通知を送り、「再開完了」応答を待つことにより達成される。

【００３５】

ステップ４０８では、中央管理プロセス１３０が、プロセスが中断されたまま残っているプロセスが従属木にあるかどうか判断する。残っていなければ、方法４００は完了し、システムが再開される。そうでなければ、従属木にあるそれぞれのプロセスが従属木で定められた適切な順番で再開されるまで、ステップ４０４から４０８が繰り返される。

【００３６】

通常の操作を再開するため、プロセス１０２、１０４、１０６、１０８、１１２、１５０および１６０は、中断された順番に対して逆の順番で再開される。したがって、１）プロセス１０８、１１２、１５０および１６０を再開する。２）プロセス１５０が再開を完了した後、プロセス１１０を再開する。３）プロセス１１２および１０８が再開を完了した後、プロセス１０２を再開する。４）プロセス１０８および１１０が再開を完了した後、プロセス１０４を再開する。５）プロセス１１０が再開を完了した後、プロセス１０６を再開する。すべての「再開完了」応答が、中央管理プロセス１３０により受け取られた時、システムが再開される。

【００３７】

図５は、この発明の操作のある特定の例を示すネットワーク管理システム５００のブロック図である。システム５００は複数のソフトウェアプロセスを含み、それぞれが、例えば新しいネットワーク要素を見つけて、それをグラフィカルマップに加える「ネットワーク要素発見(Discover_Network_Element)」ＨＬＯの特定部分を制御するよう設計されている。システム５００は、新しい要素を発見する発見管理プロセス５０２、トポロジー管理プロセス５０４、集中オブジェクト記憶プロセス５０６（すなわち、意味情報管理）、および新しい要素を表示するグラフィカルマップ・プレゼンテーション管理プロセス５０８を備える。「ネットワーク要素発見」ＨＬＯは、以下の順番で操作シーケンスを実行する。（１）発見管理プロセス５０２は、新しいネットワーク構成要素を発見する。（２）発見管理プロセス５０２は、トポロジー管理プロセス５０４に新しいネットワーク構成要素について知らせる。（３）トポロジー管理プロセス５０４は、集中オブジェクト記憶プロセス５０６と連絡をとり、新しい情報でそれを更新する。（４）トポロジー管理プロセス５

０４は、自身のデータストアを更新する。（５）トポロジー管理プロセス５０４は、プレゼンテーション管理プロセス５０８と連絡をとり、新しい構成要素をグラフィカルマップに加える。（６）プレゼンテーション管理プロセス５０８は、自分のデータストアを更新する。

【００３８】

ネットワーク管理システム５００は、非トランザクションの複数データストア処理システムである。具体的には、データは複数の、独立しているが関連したデータストア５１４、５１６、５１８に保持される。また、トポロジー管理プロセス５０４は、自身のトポロジーデータストア５１４を保持し、集中オブジェクト記憶プロセス５０６は、自身の意味データストア５１８を保持し、プレゼンテーション管理プロセス５０８は、自身のグラフィカルマップ・プレゼンテーションデータストア５１６を保持する。この実施形態では、ＨＬＯが、ソフトウェアプロセス５０４、５０６、５０８へのＡＰＩを介して、異なる複数データストア５１４、５１６、５１８を更新する。ＡＰＩは、それぞれのデータストア５１４、５１６、５１８へのアクセスを制御する。さらに、プロセス５０２、５０４、５０６、５０８は自律して動作し、個々のプロセス間の操作の同期をとるトランザクション制御が全く無い。しかも、データストア５１４、５１６、５１８は、データ複写（すなわち冗長性）を全く提供しない。この実施形態では、全体のトランザクション制御が無いため、妨げられることのないシステム５００を、「ネットワーク要素発見」ＨＬＯは流れる。さらに、同時にシステム５００を流れる複数の異なるＨＬＯをもつこともでき、それぞれが、必要なプロセスのセットと連絡をとる（また、異なるデータストア５１４、５１６、５１８を潜在的に更新する）。たとえば、異なる「新規ノード追加(Add_New_Node)」ＨＬＯ（図示せず）は、オペレータに、グラフィカルマップ・プレゼンテーション管理プロセス５０８を介して新しいノードを直接追加させることもできる。このＨＬＯを使用して新規のノードを追加することにより、図５の「ネットワーク要素発見」のＨＬＯにより説明されるものとは反対方向に、システム５００を通るデータフローのパスが得られる。

【００３９】

データを管理する方法において、それぞれのプロセスが他のプロセスから独立しているとしても、様々なデータストアのデータ間には従属関係が存在する。たとえば、ネットワーク管理システム５００では、管理されるノードが、トポロジーデータストア５１４、グラフィカルマップ・プレゼンテーションデータストア５１６および意味データストア５１８において表される。データストア間の共通キーは、図５に示されるように、オブジェクトの識別子すなわち名前である。この実施形態では、オブジェクトＩＤが、トポロジー情報、意味情報およびプレゼンテーション情報を一緒に結びつける。新しいオブジェクトデータがネットワーク管理システム５００に加えられるとき（たとえば、新しいネットワークノードがネットワーク管理システム５００に加えられるとき）、トポロジー管理プロセス５０４およびグラフィカルマップ・プレゼンテーションプロセス５０８と順番に連絡をとる。したがって、ＨＬＯが完了にむけて実行しているとき、様々なデータストア５１４、５１６、５１８は、それらのうち更新されて新規のノードの存在を反映するものもあり、更新されずに反映しないものもあるという事実のため、不整合な状態にあるかもしれない。

【００４０】

同期したデータを要求するＨＬＯの例として、バックアップＨＬＯがある。データストア５１４、５１６、５１８の同期は、他のＨＬＯ（これは、従属木にあるプロセスを呼び出す）を、バックアップが進行中の間は実行させないことにより達成される。従属木にあるすべてのプロセスは、「中断」通知を受け取る手段を含み、この通知は、プロセスに同期の準備をするよう知らせる。すべてのプロセスが「中断」通知を受け取って動作すると、システムの複数のデータストアすべてが整合性ある状態にある（すなわち、複数のデータストア間に壊れた従属関係が全く存在しない）と考えられ、したがってそれらのデータストアをバックアップすることができる。

【００４１】

図６は、システム５００の従属木６００である。グラフィカルマップのプレゼンテーションプロセス５０８は、中央管理プロセス５３０（図７）と直接通信しない間接プロセスである。この例では、代理管理プロセス５４０（図７）が、そのクライアントプロセス（すなわち、プレゼンテーション管理プロセス５０８）についての相互従属関係の情報を供給し、このことは、プレゼンテーション管理プロセス５０８を呼び出すプロセスが従属木６００には無いので、プレゼンテーション管理プロセス５０８が従属木６００の一番高いレベル６０２にあることを示す。ネットワーク要素発見プロセス５０２が、データ入力ソースからデータ入力を直接受け取るシステム５００の唯一の直接プロセスなので、発見プロセス５０２は、従属木６００の次のレベル６０４に置かれる。トポロジー管理プロセス５０４は、発見プロセス５０２から要求を受け取る唯一のプロセスなので、従属木６００の次のレベル６０６に属する。最後に、集中オブジェクト記憶プロセス５０６が、トポロジー管理プロセス５０４から要求を受け取る唯一の残っているプロセスなので、従属木６００の次のレベル６０８に属する。

【００４２】

図７は、システム７００を通る「中断」通知の順番を示すフロー図であり、図５の異なるデータストア５１４、５１６、５１８のそれぞれのデータの同期をとる。システム７００は中央管理プロセス５３０を含み、中央管理プロセスは、システムのプロセス５０２、５０４、５０６、５０８の間の従属関係を理解し、データ整合性が確実に維持される順番でプロセスを開始、停止、中断および再開できるようにする。また、システム７００は、管理代理プロセス５４０を含む。プレゼンテーション管理プロセス５０８が、従属木６００の一番高いレベル６０２にあるので、管理代理プロセス５４０を介してプレゼンテーション管理プロセス５０８を最初に中断し、プレゼンテーション管理プロセス５０８の任意のクライアントプロセスが、要求された時に確実に中断可能な状態にあるようにしなければならない。

【００４３】

データストア５１４、５１６、５１８のデータの同期をとる方法は、第１のステップ７０２で開始し、このステップでは、同期したデータを必要とするプロセスの「中断実行可能」５５０が、中央管理プロセス５３０に、同期の準備をするためプロセスに通知し始めるよう要求する。ステップ７０４では、中央管理プロセス５３０が、管理代理プロセス５４０に「中断」通知を送る。ステップ７０６では、管理代理プロセス５４０が、そのクライアントプロセス（図示せず）のそれぞれに、代理「中断」プロセスを転送する。実際には、ある種類のプロセスの複数のインスタンスをもつことができる（グラフィカルマップのプレゼンテーションプロセス５０８のような）。また、システムは、多くの異なる種類の管理代理プロセス５４０を持つこともでき、それぞれが、クライアントの特定の種類の代理に合わせて調整される（たとえば、グラフィカル・ユーザー・インターフェース（ＧＵＩ）またはＳＮＭＰエージェント）。管理代理プロセス５４０の目的は、中央管理プロセス５３０と通信しないプロセスをサポートすることである。管理代理プロセス５４０は、定義されたプロトコルで中央管理プロセス５３０からコマンドを受け取り、そのクライアントプロセスが理解するプロトコル（例えばＡＰＩ）を使用して、クライアントプロセス（すなわち、プレゼンテーションプロセス５０８）と通信する。ステップ７０８では、グラフィカルマップ・プレゼンテーションプロセス５０８は、何らかの待ち状態にあるマップ変更をデータストア５１６に流すことにより、同期の準備をする。その中断機能が完了すると、プレゼンテーションプロセス５０８は、代理「中断完了」応答を用いて、管理代理プロセス５４０に応答する。代理「中断」通知を受け取る任意のプロセスは、否定応答を管理代理プロセス５４０に返すことができ、この場合には、中央管理プロセス５３０は、すべてのプロセスに「再開」通知を送ることにより中断操作全体を中止する。

【００４４】

管理代理プロセス５４０は、肯定応答（すなわち、代理「中断完了」の応答）をプレゼンテーションプロセス５０８から受け取るならば、ステップ７１０で中央管理プロセス５３０に「中断完了」応答を返す。中央管理プロセス５３０は、従属木６００により定義され

た順番に従って、すべての他のプロセスに「中断」通知を送る。したがって、ステップ７１２で中央管理プロセス５３０は発見プロセス５０２に「中断」通知を送り、ステップ７１４で「中断完了」応答を受け取る。その後、中央管理プロセス５３０は、ステップ７１６で集中オブジェクト記憶プロセス５０６に「中断」通知を送り、ステップ７１８で「中断完了」応答を受け取る。ステップ７２０では、中央管理プロセス５３０はトポロジー管理プロセス５０４に「中断」通知を送り、ステップ７２２で「中断完了」応答を受け取る。中央管理プロセス５３０がそれぞれのプロセスと連絡をとり、それぞれのプロセスから「中断完了」応答を受け取ると、中央管理プロセス５３０は、ステップ７２４で、「システムが中断された」メッセージを「中断実行可能」５５０に返し、システムおよびそのすべてのデータが整合性ある状態にあることを示す。

【００４５】

当該技術分野の当業者には、この発明が、「中断」通知または「再開」通知がトラベル(travel)することのできるレベル数を制限しないということを理解するであろう。この発明はＮレベルの中断／再開の階層を説明し、それぞれのレベルは次のレベルにメッセージを転送し、次のレベルから応答を受け取る。たとえば、ここで説明したグラフィカルマップ・プレゼンテーションプロセス５０８は、ＡＰＩを介して、マップを増やすことのできる他のアプリケーションとの通信をサポートすることができる。これらのアプリケーションには、代理「中断」通知を受け取って、アプリケーションが同期の準備をすることができるようにすることも必要である（すなわち、これらのアプリケーションは、メモリにマップの変更を格納することができ、メモリにおけるこれらの変更は、同期の前にグラフィカルマップ・プレゼンテーションプロセス５０８に送られるべきである）。この例では、グラフィカルマップ・プレゼンテーションプロセス５０８は、接続されるアプリケーションに「中断」通知を転送する。こうして、グラフィカルマップ・プレゼンテーションプロセス５０８は、そのクライアントが応じる能力により、大部分の「中断」通知に応じる能力を決定する。

【００４６】

すべてのプロセスが中断するよう通知されて中央管理プロセス５３０に応答を送り返すと、バックアップまたはデータのチェックポイントのような操作を実行することができる。このような操作の完了の後、中断されたプロセスに知らせて通常の操作を再開することは、中央管理プロセス５３０の責任である。「開始」通知を送るのとは逆の順番で、それらのプロセスに「再開」通知を送ることにより、これを行う。

【００４７】

非トランザクション処理で複数データストアの同期をとるシステムおよび方法を、上記に詳細に説明してきた。この発明は、例示した実施形態により説明したけれども、当該技術分野の当業者には理解できるように、この発明の精神および範囲から離れることなく、例示した実施形態に対して様々な変更および改良を行うことができる。この発明の範囲は例示した実施形態に制限されるものではなく、この発明は特許請求の範囲により定められる。

【００４８】

本発明は、例として以下の実施態様を含む。

【００４９】

（１）非トランザクション多重処理コンピュータシステム(100)において複数のデータストアの同期をとってデータ整合性を達成する方法であって、

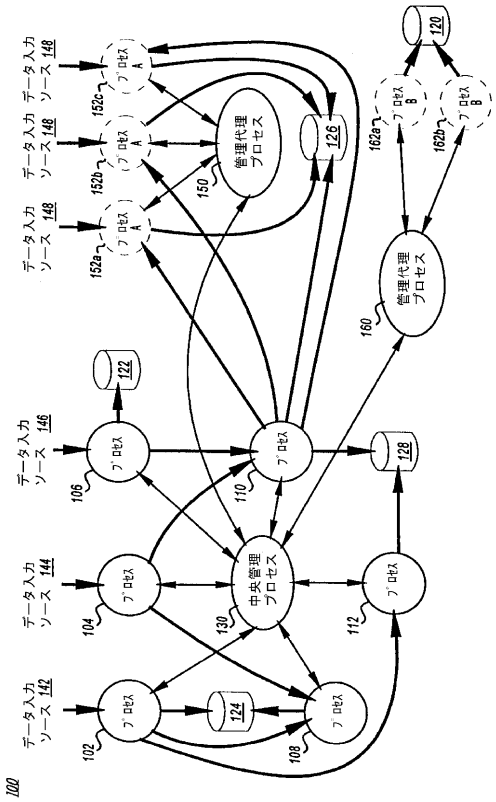
該システムが、複数のデータストア(120-128)と、１または複数のデータ入力ソース(142-148)と、該１または複数のデータ入力ソースのうちの少なくとも１つデータ入力ソースおよび前記複数のデータストアのうちの少なくとも１つデータストアの間の従属パスにある複数のプロセス(102-112)とを備え、

(a)前記複数のプロセスのうち、前記複数のプロセスの任意の他のプロセスにより呼び出されておらず、前記１または複数のデータ入力ソースのうちの少なくとも１つからデータを直接受け取るそれぞれのプロセスを備える前記複数のプロセスの第１のサブセット(102

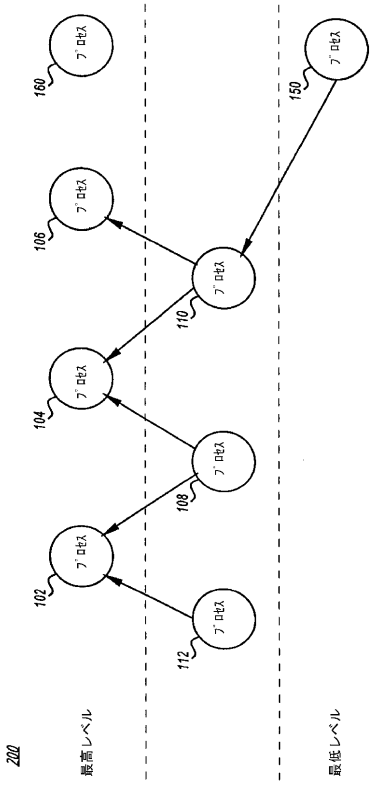
、104、106、160)を中断するステップと、
(b)前記複数のプロセスのうち、中断されたプロセスにより呼び出されており、中断されていないプロセスにより呼び出されていない少なくとも１つのプロセスを含む次のサブセットを中断するステップと、
(c)前記複数のプロセスのうち、いずれかのプロセスが中断されずに残っているならば、前記ステップ（ｂ）を繰り返すステップと、
を含む複数データストアの同期をとってデータ整合性を達成する方法。
【００５０】
（２）(d)前記複数のデータストアのうち任意のデータストアの修正を生じさせるサービス要求の受け入れを停止するステップと、
(e)前記複数のデータストアのうち任意のデータストアの修正を生じさせるすべての待ち状態タスクを完了させるステップと、
(f)中断されてないプロセスまたは前記複数のデータストアのうち１または複数のデータストアに、内部的にキャッシュに格納されたデータをすべて流すステップと、
を実行することにより、前記複数のプロセスのそれぞれが中断するようにした上記（１）に記載の方法。
【００５１】
（３）(g)前記複数のプロセスのうち、現在中断されているプロセスを呼び出さないそれぞれのプロセスを含む最後のサブセット(108、112、150、160)を再開するステップと、
(h)前記複数のプロセスのうち、１または複数の再開されたプロセスを呼び出し、現在中断されているプロセスを呼び出さない少なくとも１つのプロセスを含む前のサブセットを再開するステップと、
(i)前記複数のプロセスのうち、いずれかのプロセスが再開されずに残っているならば、前記ステップ（ｈ）を繰り返すステップと、
を実行することにより、前記システムの通常の操作が再開されるようにした上記（１）または（２）に記載の方法。
【００５２】
（４）(j)前記複数のデータストアのうち少なくとも１つデータストアの修正を生じさせるサービス要求の受け入れを可能にするステップにより、前記複数のプロセスがそれぞれが再開するようにした上記（３）に記載の方法。
【００５３】
（５）同期をとって整合性あるデータを達成する非トランザクションの複数データストア処理システムであって、
複数のデータストア(120-128)と、
１または複数のデータ入力ソース(142-148)と、
前記１または複数のデータ入力ソースのうちの少なくとも１つのデータ入力ソースと、前記複数データストアのうちの少なくとも１つデータストアとの間の従属パスにある複数のプロセスであって、それぞれが、前記データストアの修正を生じさせるサービス要求の受け入れを中断し、前記複数データストアを修正するすべての待ち状態タスクを完了させ、中断されてないプロセスまたは前記複数のデータストアのうち１または複数のデータストアのいずれかに、内部的にキャッシュに格納されたデータをすべて流す複数のプロセス(102-112)と、
前記複数プロセスのうち、任意の他のプロセスにより呼び出されておらず、前記１または複数のデータ入力ソースのうちの少なくとも１つから直接データを受け取るそれぞれのプロセスを含む第１のサブセットを中断し、中断されないまま残っているプロセスが無くなるまで、中断されたプロセスにより呼び出されており、中断されていないプロセスにより呼び出されていない少なくとも１つのプロセスを含む次のサブセットを繰り返し中断することにより、順番に、前記複数のプロセスのそれぞれを中断するよう動作することのできる同期をとる手段(130)と、
を備える非トランザクション複数データストア処理システム。

【００５４】
（６）前記複数のプロセスのそれぞれが、前記複数のデータストアのうち１または複数のデータストアの修正を生じさせるサービス要求の受け入れを可能にする再開手段を備える上記（５）に記載のシステム。
【００５５】
（７）前記同期をとる手段が、
前記複数のプロセスのうち、現在中断されているプロセスを呼び出していないそれぞれのプロセスを含む最後のサブセットを再開し、中断されたまま残っているプロセスが無くなるまで、１または複数の再開されたプロセスを呼び出し、中断されているプロセスを呼び出さない少なくとも１つのプロセスを含む次のサブセットを繰り返し再開することにより、
順番に、前記複数のプロセスのそれぞれを再開するよう動作することができるようにした上記（６）に記載のシステム。
【００５６】
（８）前記１または複数のデータ入力ソースのうちの少なくとも１つのデータ入力ソースと、前記複数データストアのうち少なくとも１つのデータストアと間の従属パスにある間接プロセス(152a-152c)を備え、
該間接プロセスが、前記複数のデータストアの修正を生じさせるサービス要求の受け入れを停止し、前記複数のデータストアを修正するすべての待ち状態タスクを完了させ、中断されていないプロセスまたは前記複数データストアの１または複数データストアに、内部的にキャッシュに格納されたデータのすべてを流すことにより、代理中断要求に応答し、前記間接プロセスへの代理中断要求を生成して送ることにより中断要求に応答する代理プロセス(150)を備え、
前記間接プロセスが中断されるとき、前記同期をとる手段(130)が前記代理プロセスに中断要求を送るようにした上記（５）から（７）に記載のシステム。
【００５７】
（９）前記代理中断要求が受け取られるとき、１または複数のクライアント・プロセスが現在中断可能ならば、前記間接プロセスが１または複数のクライアントプロセスと対話し、代理中断要求に応答して中断を実行する上記（８）に記載のシステム。
【００５８】
【発明の効果】
非トランザクション処理システムの複数のデータストア間に同期および整合性を達成することができるので、システムを走らせたたままにして、ユーザに対する時間およびサービスの損失を最小にすることができる。
【図面の簡単な説明】
【図１】この発明による非トランザクション複数データストア処理システムのブロック図。
【図２】図１のシステムの従属木を示す図。
【図３】この発明によるシステムのプロセスを中断する方法を示すフローチャート。
【図４】この発明によるシステムの中断されたプロセスを再開する方法を示すフローチャート。
【図５】ＨＬＯのシーケンシャルフローを示すこの発明のシステムのブロック図。
【図６】図５のシステムの従属木を示す図。
【図７】図５のシステムを通る中断通知の順番を示すフロー図。
【符号の説明】
１００　非トランザクション多重処理コンピュータシステム
１０２～１１２　プロセス
１２０～１２８　データストア
１４２～１４８　データ入力ソース

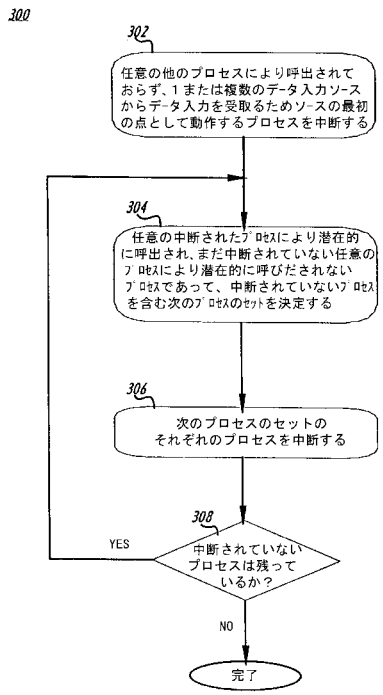
【図１】


データ入力ソース 148
データ入力ソース 146
データ入力ソース 144
データ入力ソース 142
プロセス A
プロセス B
管理代理プロセス
中央管理プロセス
プロセス
100

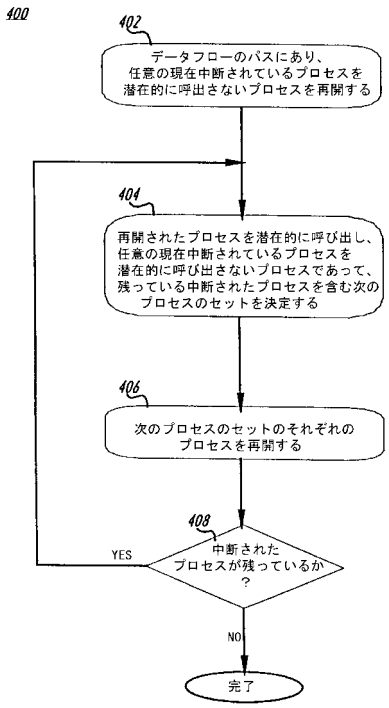
【図２】


プロセス
最高レベル
最低レベル
200

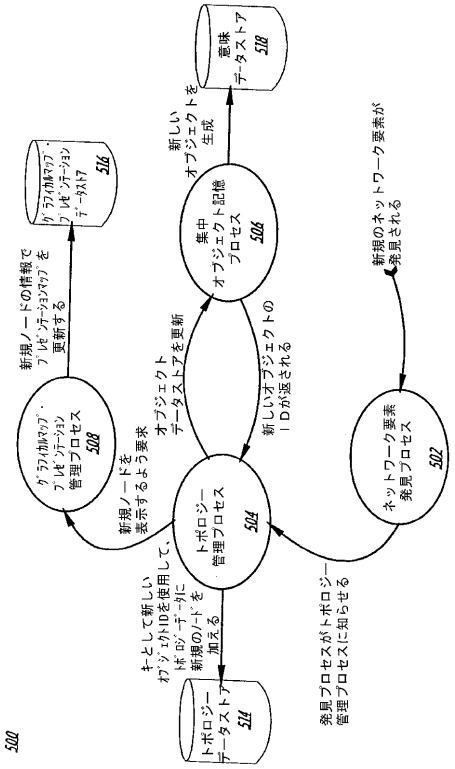
【図３】


300
302
任意の他のプロセスにより呼出されておらず、1または複数のデータ入力ソースからデータ入力を受取るためソースの最初の点として動作するプロセスを中断する
304
任意の中断されたプロセスにより潜在的に呼出され、まだ中断されていない任意のプロセスにより潜在的に呼びだされないプロセスであって、中断されていないプロセスを含む次のプロセスのセットを決定する
306
次のプロセスのセットのそれぞれのプロセスを中断する
308
中断されていないプロセスは残っているか？
YES
NO
完了

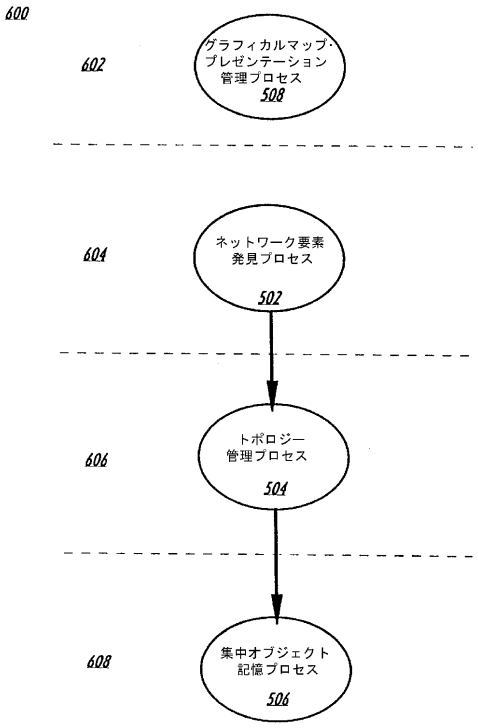
【図４】


400
402
データフローのパスにあり、任意の現在中断されているプロセスを潜在的に呼出さないプロセスを再開する
404
再開されたプロセスを潜在的に呼び出し、任意の現在中断されているプロセスを潜在的に呼び出さないプロセスであって、残っている中断されたプロセスを含む次のプロセスのセットを決定する
406
次のプロセスのセットのそれぞれのプロセスを再開する
408
中断されたプロセスが残っているか？
YES
NO
完了

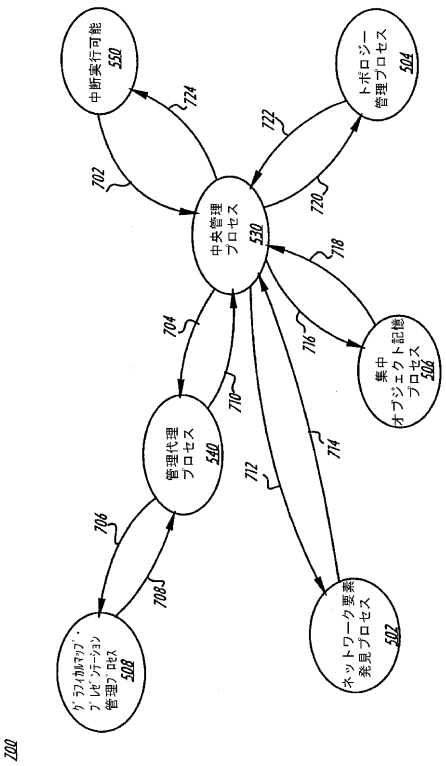
【図５】


500
グラフィカルマップ・プレゼンテーション データストア 516
新規ノードの情報で グラフィカルマップ・プレゼンテーションを 更新する
グラフィカルマップ・プレゼンテーション 管理プロセス 508
新規ノードを 表示するよう要求
オブジェクト データストアを更新
集中 オブジェクト記憶 プロセス 506
新しい オブジェクトを 生成
意味 データストア 518
新しいオブジェクトの IDが返される
新規のネットワーク要素が 発見される
トポロジー 管理プロセス 504
ネットワーク要素 発見プロセス 502
キーとして新しい オブジェクトIDを使用して、 トポロジーデータに 新規のノードを 加える
トポロジー データストア 514
発見プロセスがトポロジー 管理プロセスに知らせる

【図６】


600
602
グラフィカルマップ・プレゼンテーション 管理プロセス 508
604
ネットワーク要素 発見プロセス 502
606
トポロジー 管理プロセス 504
608
集中オブジェクト 記憶プロセス 506

【図７】


700
中断実行可能 550
724
722
トポロジー 管理プロセス 504
702
720
中央管理 プロセス 530
718
704
716
集中 オブジェクト記憶 プロセス 506
710
714
管理代理 プロセス 540
712
706
708
グラフィカルマップ・プレゼンテーション 管理プロセス 508
ネットワーク要素 発見プロセス 502

フロントページの続き

(72)発明者　トニー・アトキンソン
アメリカ合衆国８０５２４コロラド州フォート・コリンズ、スミス・ストリート　７０８
(72)発明者　スティーブン・シー・ブース
アメリカ合衆国８０５２８コロラド州フォート・コリンズ、ウォルデン・サークル　７８０５
(72)発明者　ジェームズ・アール・グリュエル
アメリカ合衆国８０５２５コロラド州フォート・コリンズ、ブルックウッド・ドライブ　２１１４
(72)発明者　ポール・エイチ・プライス
アメリカ合衆国８０５３７コロラド州ラブランド、ハイ・カントリー・ロード　４６００
(72)発明者　ロバート・ディー・シェトラー
アメリカ合衆国８０５２１コロラド州フォート・コリンズ、サマー・ストリート　１０２７
(72)発明者　ダレン・ディー・スミス
アメリカ合衆国８０５２６コロラド州フォート・コリンズ、ゴールデンリッジ・ウェイ　４２４２
(72)発明者　ジョン・ティー・ウォード
アメリカ合衆国８０５３７コロラド州ラブランド、ティアビ・コート　３７５

審査官　殿川　雅也

(56)参考文献　国際公開第９７／０１３２０６（WO，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　9/46-9/54
G06F 15/16-15/177
G06F 12/00