

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6220851号  
(P6220851)

(45) 発行日 平成29年10月25日(2017.10.25)

(24) 登録日 平成29年10月6日(2017.10.6)

(51) Int. Cl.		F I			
<b>G06F 12/00</b>	<b>(2006.01)</b>	G06F 12/00	518A		
<b>G06F 9/52</b>	<b>(2006.01)</b>	G06F 12/00	535C		
		G06F 9/46	475C		

請求項の数 10 (全 15 頁)

(21) 出願番号	特願2015-500669 (P2015-500669)	(73) 特許権者	502303739
(86) (22) 出願日	平成25年3月15日 (2013. 3. 15)		オラクル・インターナショナル・コーポレイション
(65) 公表番号	特表2015-514248 (P2015-514248A)		アメリカ合衆国カリフォルニア州94065レッドウッド・シティー, オラクル・パークウェイ500
(43) 公表日	平成27年5月18日 (2015. 5. 18)	(74) 代理人	110001195
(86) 国際出願番号	PCT/US2013/032473		特許業務法人深見特許事務所
(87) 国際公開番号	W02013/138774	(72) 発明者	パーキンソン, ポール
(87) 国際公開日	平成25年9月19日 (2013. 9. 19)		アメリカ合衆国、32828 フロリダ州、オーランド、バッサノ・ウェイ、1110
審査請求日	平成27年12月22日 (2015. 12. 22)	審査官	小林 哲雄
(31) 優先権主張番号	61/612, 144		
(32) 優先日	平成24年3月16日 (2012. 3. 16)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	61/774, 356		
(32) 優先日	平成25年3月7日 (2013. 3. 7)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 2フェーズコミットコールの厳密な順序付けに基づいたトランザクションリカバリをサポートするためのシステムおよび方法

(57) 【特許請求の範囲】

【請求項1】

複数のリソースマネージャに対して複数のトランザクションのリカバリをサポートする方法であって、前記方法は、

トランザクションマネージャが前記複数のリソースマネージャの中の1つのリソースマネージャを決定リソースマネージャとして指定することと、

2フェーズコミットプロトコルを用いた前記複数のトランザクションの処理中に、前記トランザクションマネージャが、各前記トランザクションごとに、前記決定リソースマネージャを除く全ての他のリソースマネージャについて準備アクションの成功の通知を受け取った後にのみ、前記決定リソースマネージャに準備メッセージを送信することと、

前記2フェーズコミットプロトコルを用いた前記複数のトランザクションの処理中に、前記トランザクションマネージャが、各前記トランザクションごとに、前記決定リソースマネージャを除く全ての他のリソースマネージャについてコミットアクションの成功の通知を受け取った後にのみ、前記決定リソースマネージャにコミットメッセージを送信することと、

障害の発生時に、前記トランザクションマネージャが、前記決定リソースマネージャを含む前記複数のリソースマネージャの中の各リソースマネージャからインダウトトランザクションのリストを取得することと、

障害の発生時に、前記トランザクションマネージャが、トランザクションログを用いることなく、前記決定リソースマネージャ以外の各リソースマネージャから取得したインダ

ウトランザクションのリストを、前記決定リソースマネージャから取得したインダウトランザクションのリストと比較することによって、全てのインダウトランザクションをリカバリすることを含む、方法。

【請求項 2】

複数のランザクションのリカバリをサポートするためのシステムであって、前記システムは、

ランザクションマネージャと、

中間層ランザクションシステムにおいて前記ランザクションマネージャと通信する複数のリソースマネージャとを備え、

前記ランザクションマネージャは以下のステップを実行するように動作し、前記ステップは、

前記複数のリソースマネージャの中の1つのリソースマネージャを決定リソースマネージャとして指定することと、

2フェーズコミットプロトコルを用いた前記複数のランザクションの処理中に、各前記ランザクションごとに、前記決定リソースマネージャを除く全ての他のリソースマネージャについて準備アクションの成功の通知を受け取った後にのみ、前記決定リソースマネージャに準備メッセージを送信することと、

前記2フェーズコミットプロトコルを用いた前記複数のランザクションの処理中に、各前記ランザクションごとに、前記決定リソースマネージャを除く全ての他のリソースマネージャについてコミットアクションの成功の通知を受け取った後にのみ、前記決定リ

ソースマネージャにコメントメッセージを送信することと、

障害の発生時に、前記決定リソースマネージャを含む前記複数のリソースマネージャの中の各リソースマネージャからインダウトランザクションのリストを取得することと、  
障害の発生時に、ランザクションログを用いることなく、前記決定リソースマネージャ以外の各リソースマネージャから取得したインダウトランザクションのリストを、前記決定リソースマネージャから取得したインダウトランザクションのリストと比較することによって、全てのインダウトランザクションをリカバリすることを含む、システム。

【請求項 3】

前記ランザクションマネージャがランザクションログを用いることなく全てのインダウトランザクションをリカバリする際に、前記ランザクションマネージャは、前記決定リソースマネージャ以外の各リソースマネージャから取得した前記インダウトランザクションのリストの中に現れるが前記決定リソースマネージャから取得した前記インダウトランザクションのリストに現れていない1つ以上のインダウトランザクションをロールバックするようにさらに動作する、請求項2に記載のシステム。

【請求項 4】

前記ランザクションマネージャがランザクションログを用いることなく全てのインダウトランザクションをリカバリする際に、前記ランザクションマネージャは、前記決定リソースマネージャ以外の各リソースマネージャから取得した前記インダウトランザクションのリストの中に現れ、かつ前記決定リソースマネージャから取得した前記インダウトランザクションのリストの中にも現れる1つ以上のインダウトランザクションをコミットするようにさらに動作する、請求項2または3に記載のシステム。

【請求項 5】

前記ランザクションマネージャは、ランザクション中間層システムのための構成において前記決定リソースマネージャの指定を記憶保持するように動作する、請求項2～4のいずれか1項に記載のシステム。

【請求項 6】

前記複数のリソースマネージャの中の少なくとも1つのリソースマネージャは前記複数のランザクションの不参加者である、請求項2～5のいずれか1項に記載のシステム。

【請求項 7】

10

20

30

40

50

前記トランザクションマネージャは、前記決定リソースマネージャから取得したインダウトトランザクションのリストを、前記複数のリソースマネージャの中の前記決定リソースマネージャを除く他のすべてのリソースマネージャをリカバリした後にコミットするように動作する、請求項 2 ~ 6 のいずれか 1 項に記載のシステム。

【請求項 8】

前記トランザクションマネージャは、

前記決定リソースマネージャから取得したインダウトトランザクションのリストから、グローバルトランザクションテーブル (GTT) をビルド/リビルドすることと、各インダウトトランザクションを前記 GTT と比較することとをさらに含むステップを実行するように動作する、請求項 2 ~ 6 のいずれか 1 項に記載のシステム。

10

【請求項 9】

前記トランザクションマネージャは、

前記決定リソースマネージャに前記複数のリソースマネージャの中の参加者リソースマネージャのリストを送ることと、

前記参加者リソースマネージャのリストがリカバリされるまで、前記決定リソースマネージャをリカバリしないで待機させるよう構成することとをさらに含むステップを実行するように動作する、請求項 2 ~ 6 のいずれか 1 項に記載のシステム。

【請求項 10】

実行されるとシステムに以下のステップを行なわせる機械読取可能なプログラムであって、前記ステップは、

20

トランザクションマネージャが複数のリソースマネージャの中の 1 つのリソースマネージャを決定リソースマネージャとして指定することと、

2 フェーズコミットプロトコルを用いた複数のトランザクションの処理中に、前記トランザクションマネージャが、各前記トランザクションごとに、前記決定リソースマネージャを除く全ての他のリソースマネージャについて準備アクションの成功の通知を受け取った後にのみ、前記決定リソースマネージャに準備メッセージを送信することと、

前記 2 フェーズコミットプロトコルを用いた前記複数のトランザクションの処理中に、前記トランザクションマネージャが、各前記トランザクションごとに、前記決定リソースマネージャを除く全ての他のリソースマネージャについてコミットアクションの成功の通知を受け取った後にのみ、前記決定リソースマネージャにコミットメッセージを送信することと、

30

障害の発生時に、前記トランザクションマネージャが、前記決定リソースマネージャを含む前記複数のリソースマネージャの中の各リソースマネージャからインダウトトランザクションのリストを取得することと、

障害の発生時に、前記トランザクションマネージャが、トランザクションログを用いることなく、前記決定リソースマネージャ以外の各リソースマネージャから取得したインダウトトランザクションのリストを、前記決定リソースマネージャから取得したインダウトトランザクションのリストと比較することによって、全てのインダウトトランザクションをリカバリすることを含む、機械読取可能なプログラム。

【発明の詳細な説明】

40

【技術分野】

【0001】

著作権に関する注意

本特許文献の開示の一部には、著作権保護の対象となるものが含まれている。著作権者は、この特許文献または特許開示の何者かによる複製が、特許商標庁の特許ファイルまたは記録にある限り、それに対して異議を唱えないが、そうでなければ、いかなる場合もすべての著作権を留保する。

【0002】

発明の分野

本発明は、概してコンピュータシステムおよびミドルウェア等のソフトウェアに関し、

50

特にトランザクションミドルウェアに関する。

【背景技術】

【0003】

背景

トランザクションミドルウェアシステム、またはトランザクション指向ミドルウェアは、ある組織内のさまざまなトランザクションを処理することができる企業アプリケーションサーバを含む。高性能ネットワークおよびマルチプロセッサコンピュータ等の新技術の開発に伴い、トランザクションミドルウェアの性能をより一層改善することが必要になっている。これらが、本発明の実施の形態が取組もうとしている一般的な分野である。

【発明の概要】

【課題を解決するための手段】

【0004】

概要

2フェーズコミットコールの厳密な順序付けに基づいたトランザクションリカバリをサポートするためのシステムおよび方法が提供される。2フェーズトランザクションの処理において中間層トランザクションログ(transaction log)(TLOG)の排除をサポートするために、中間層トランザクション環境内の少なくとも1つのリソースマネージャを「決定(determiner)リソース」として指定することができる。トランザクションマネージャは、決定リソースの前に、中間層トランザクションシステム内のその他すべてのリソースマネージャを、準備することができる。さらに、トランザクションマネージャは、トランザクションのリカバリのために、決定リソースから与えられた、コミットすべき未処理のトランザクションのリストに依拠することができる。トランザクションマネージャは、リソースマネージャから返されたインダウト(in-doubt)トランザクションが、決定リソースから返されたインダウトトランザクションのリストと一致する場合、このインダウトトランザクションをコミットすることができる。そうでなければ、トランザクションマネージャは上記インダウトトランザクションをロールバックすることができる。

【0005】

本発明のその他の目的および利点は、当業者には、さまざまな実施の形態に関する以下の詳細な説明を添付の図面に照らして読めば明らかになるであろう。

【図面の簡単な説明】

【0006】

【図1】本発明のある実施の形態に従うトランザクション環境の一例を示す。

【図2】トランザクションミドルウェアマシン環境における2フェーズコミットのサポートの一例を示す。

【図3】トランザクションミドルウェアマシン環境におけるインダウトトランザクションのリカバリの一例を示す。

【図4】本発明のある実施の形態に従う、トランザクションミドルウェアマシン環境におけるトランザクション処理のための2フェーズコミット(two-phase commit)(2PC)コールの厳密な順序付けのサポートの一例を示す。

【図5】本発明のある実施の形態に従う、トランザクションログ(TLOG)を使用しないグローバルトランザクションのリカバリの一例を示す。

【図6】本発明のある実施の形態に従う、トランザクションログ(TLOG)を使用しない2フェーズコミットトランザクションのリカバリのフローチャートの一例である。

【発明を実施するための形態】

【0007】

詳細な説明

本発明は、同様の参照符号が同様の構成要素を示している添付の図面に、限定のためではなく例示のために示されている。なお、本開示において「ある」または「一」または「いくつかの」実施の形態に言及している場合、それは必ずしも同一の実施の形態ではなく、こういった表現は、少なくとも1つという意味である。

10

20

30

40

50

## 【 0 0 0 8 】

本発明の以下の説明では、トランザクションミドルウェアマシン環境の一例としてTuxedo環境を用いている。当業者には、その他の種類のトランザクションミドルウェアマシン環境を、限定されることなく使用できることが、明らかであろう。

## 【 0 0 0 9 】

本明細書において、2フェーズコミットコールの厳密な順序付けに基づいたトランザクションリカバリをサポートするためのシステムおよび方法について説明する。2フェーズトランザクションの処理において中間層トランザクションログ(TLOG)の排除をサポートするために、中間層トランザクション環境内の少なくとも1つのリソースマネージャを「決定リソース」として指定することができる。トランザクションマネージャは、決定リソースの前に、中間層トランザクションシステム内のその他すべてのリソースマネージャを準備することができる。さらに、トランザクションマネージャは、トランザクションのリカバリのために、決定リソースから与えられた、コミットすべき未処理のトランザクションのリストに依拠することができる。トランザクションマネージャは、リソースマネージャから返されたインダウトトランザクションが、決定リソースから返されたインダウトトランザクションのリストと一致する場合、このインダウトトランザクションをコミットすることができる。そうでなければ、トランザクションマネージャは上記インダウトトランザクションをロールバックすることができる。

10

## 【 0 0 1 0 】

トランザクション環境およびグローバルトランザクション

20

図1は、本発明のある実施の形態に従うトランザクション環境の一例を示す。図1に示されるように、トランザクション環境100は、エンドユーザ101と、アプリケーションプログラム102と、1つ以上のトランザクションマネージャ(transaction manager)(TM)103と、複数のリソースマネージャ(resource manager)(RM)104~106と、1つ以上の永続性記憶装置、たとえばデータベース110とを含み得る。

## 【 0 0 1 1 】

本発明の一実施の形態に従うと、アプリケーションプログラム102は、あるトランザクションを構成するアクションを指定することができる。示されているように、アプリケーションプログラム102は、トランザクションマネージャ103と通信することにより、あるトランザクションを開始、コミット、またはアボートし、トランザクションマネージャ103は、アプリケーションプログラム102に、このトランザクションの開始、終了、および処置(disposition)を、送り返すことができる。さらに、アプリケーションプログラム102は、トランザクションマネージャ103に対してトランザクションの境界を定めることができ、トランザクションマネージャ103は、トランザクション情報を、複数のリソースマネージャ(RM)104~106と交換することができる。加えて、アプリケーションプログラム102は、複数のリソースマネージャ104~106と、埋込型構造化クエリ言語(Embedded Structured Query Language)(SQL)を介して通信することにより、有用な作業を行なうことができる。

30

## 【 0 0 1 2 】

複数のリソースマネージャ104~106は、永続性記憶装置、たとえばデータベース110へのアクセスを提供することができる。本発明のある実施の形態に従うと、複数のリソースマネージャ104~106は、XAインターフェイスを実装することにより、データベースの接続および切断を、永続性記憶装置にとってトランスペアレントな状態で、処理することができる。XAインターフェイスは、トランザクションのコーディネート、コミット、およびリカバリに関するプロトコルが記載された仕様書に基づくものであってもよい。あるトランザクションに参加するXAリソースは、XAリソースマネージャとバックエンド永続性記憶装置とを含み得る。

40

## 【 0 0 1 3 】

本発明のさまざまな実施の形態に従うと、トランザクションシステムは、2つ以上のサーバ上で実行できるグローバルトランザクションをサポートすることができ、かつ、2つ

50

以上のリソースマネージャからのデータにアクセスすることが可能である。

【 0 0 1 4 】

グローバルトランザクションは、アトミック性 (atomicity)、一貫性 (consistency)、独立性 (isolation)、および永続性 (durability) (ACID) という4つの特性によって特徴付けられるオペレーションの特定のシーケンスとして扱うことができる。グローバルトランザクションは、以下の特徴を有する作業の一論理単位であってもよい。

【 0 0 1 5 】

- ・すべての部分が成功か、すべての部分が無効かの、いずれかである。
- ・リソースを、ある一貫した状態から別の状態に正しく変換するオペレーションが実行される。

10

【 0 0 1 6 】

- ・同一トランザクションにおける他のプロセスはデータにアクセスし得るが、他のトランザクションは中間結果にアクセスできない。

【 0 0 1 7 】

- ・完了したシーケンスのすべての効果は、いかなる種類の障害でも変更され得ない。

さらに、グローバルトランザクションは、各々が1つのリソースマネージャにアクセスする、数個のローカルトランザクションを含み得る。1つのローカルトランザクションは、1つのデータベースまたはファイルにアクセスすることができ、このデータベースにおける同時実行性制御および更新のアトミック性を実現するリソースマネージャによって制御することができる。所与のローカルトランザクションは、そのアクセスの完了に成功するか失敗するかのいずれかであり得る。

20

【 0 0 1 8 】

加えて、トランザクションマネージャ 1 0 3 は、グローバルトランザクション識別子 (global transaction identifier) (GTRID) を、トランザクション環境 1 0 0 内のさまざまなトランザクションに割当てることができる。トランザクションマネージャ 1 0 3 は、これらの進行をモニタリングすることができ、トランザクションの完了を保証し障害をリカバリする役割を果たすことができる。加えて、トランザクションマネージャ 1 0 3 は、複数のリソースマネージャ 1 0 4 ~ 1 0 5 と X A インターフェイスを介して通信することにより、2 フェーズコミットコールをリソースマネージャ 1 0 4 ~ 1 0 5 に送る等のトランザクション情報を交換することができる。

30

【 0 0 1 9 】

2 フェーズコミット (two-phase commit) (2PC)

2 フェーズコミット (2PC) プロトコルを用いて、疎結合 (loosely-coupled) グローバルトランザクション等のトランザクションを実行することができる。2 フェーズコミットプロトコル (2PC) は、準備フェーズとコミットフェーズとを含み得る。準備フェーズでは、コーディネートトランザクションマネージャ (TM) が、参加しているリソースマネージャ (RM) に、トランザクションをコミットするかまたはアボートするために必要な措置を講じるよう指示する。コミットフェーズでは、トランザクションマネージャ (TM) が、準備フェーズの結果に基づいて、トランザクションをコミットするかアボートするかを判断する。

40

【 0 0 2 0 】

図 2 は、トランザクションミドルウェアマシン環境における 2 フェーズコミットのサポートの一例を示す。図 2 に示されるように、トランザクション環境 2 0 0 は、さまざまなトランザクションの実行をサポートするトランザクションマネージャ 2 0 1 と、1つ以上のデータソース、たとえばデータベース 2 0 5 を管理する1つ以上のリソースマネージャ 2 0 2 ~ 2 0 4 とを含み得る。

【 0 0 2 1 】

たとえば、トランザクションマネージャ 2 0 1 は、それぞれリソースマネージャ 2 0 2 ~ 2 0 4 に対して実行できるトランザクション分岐 A 2 1 1 とトランザクション分岐 B 2 1 2 とトランザクション分岐 C 2 1 3 とを含むトランザクションを、実行することができ

50

る。このトランザクションにおいていずれかの分岐に障害が発生した場合、トランザクションマネージャ201は、このトランザクションをコミットするかまたはロールバックするかをリソースマネージャ202~204が判断するのを支援することができる。

【0022】

図2に示されるように、トランザクションマネージャ201は、3つの分岐すべてにおいて、準備の指示をリソースマネージャ202~204に送信することができる(ステップ1、2、および3)。リソースマネージャ202~204が「OK」という投票を返す(ステップ4、5、および6)と、その後、トランザクションマネージャ201は、トランザクションログをデータベース205に書込むことができる(ステップ7)。

【0023】

トランザクションログ(transaction log)(TLOG)は、コミットフェーズ中にいずれかの分岐に障害が発生した場合にトランザクションをリカバリするのに十分な情報をトランザクションマネージャ201が持つことができるよう、ファイルまたはデータベースに書込んでおけばよい。

【0024】

次に、トランザクションマネージャ201は、リソースマネージャ202~204に、3つの分岐すべてをコミットするよう指示することができる(ステップ8、9、および10)。リソースマネージャ202~204は、コミットフェーズの完了に成功した後に、トランザクションマネージャ201に知らせることができる(ステップ11、12、および13)。

【0025】

トランザクションログ(TLOG)に基づいたトランザクションリカバリ

本発明の一実施の形態に従うと、トランザクションログ(TLOG)は、トランザクションマネージャが下した、トランザクションをコミットするための判断を保持することができる。たとえば、TLOGはリソースチェックポイント記録を格納することができる。トランザクションマネージャは、この記録を持続することによって、参加しているさまざまなXAリソースを追跡することができる。

【0026】

トランザクションは、準備フェーズ後にすべてのトランザクション分岐から成功という投票をトランザクションマネージャが受けたときに、TLOGに書込むことができる。TLOGにおけるトランザクション記録は、少なくとも、トランザクションマネージャによって割当てられたGRIDをさらに含むトランザクション識別子(XID)と、トランザクションが実行される場合にリソースマネージャによって割当てられるローカルXIDとを含み得る。

【0027】

さらに、TLOGは、コミットされることが指定されているインフライト(in-flight)トランザクションの状態の記録を保持し得る。TLOGは、インダウトトランザクションのリカバリに役立つ。インダウトトランザクションは、システムのクラッシュ後に、準備は完了しているが中間層トランザクションシステムにおいてまだコミットされていないトランザクションである。インダウトトランザクションがリカバリされなければ、クラッシュ後のシステムは不正確で一貫していない状態となり得る。

【0028】

たとえば、2フェーズコミットトランザクション中にシステムに障害が発生した場合、1つのバックエンドデータ記憶装置に対する更新は既にコミットされているかもしれないが、同一トランザクションにおける別のデータ記憶装置に対する更新のコミットはまだ指示されていないかもしれない、すなわち、データ記憶装置の更新は保留されたままである。システムにおいて障害が発生した部分が再起動されると、保留中の更新を保持しているデータ記憶装置は、この更新をコミットすべきかまたはロールバックすべきかわからない場合がある。

【0029】

10

20

30

40

50

図3は、トランザクションミドルウェアマシン環境におけるインダウトトランザクションのリカバリの一例を示す。図3に示されるように、トランザクションミドルウェアマシン環境300は、トランザクションマネージャ301と、複数のリソースマネージャ302～304と、永続性記憶装置、すなわちデータベース305とを含む。

【0030】

トランザクションマネージャ301は、TLOGを読み出すことによって、グローバルトランザクションがインダウトであるか否かを自動的に判断できる(ステップ1)。次に、トランザクションマネージャ301は、参加しているリソースマネージャ302～304の関連するバックエンドデータ記憶装置をポーリングすることができる。参加しているリソースマネージャ302～304は各々、ロールバックするのかコミットするのかトランザクションマネージャ301がわかっていないトランザクションであるインダウトトランザクションのリストを返すことができる(ステップ2、4、および6)。

10

【0031】

さらに、トランザクションマネージャ301は、各インダウトトランザクションをTLOGと突き合わせ、続いて、異なるリソースマネージャ302～304に対し、インダウトトランザクションをコミットまたはロールバックすることができる(ステップ3、5、および7)。たとえば、TLOGにインダウトトランザクションがあるときは、XAResource.commit()をコールして、TLOGにおいて指定されているリソースマネージャに、このインダウトトランザクションをコミットすることができる。一方、TLOGにインダウトトランザクションがないとき、すなわちクラッシュ前にコミットという判断がトランザクションに対して下されていないときは、XAResource.rollback()をコールして、トランザクション識別子(XID)に基づき、それをリソースマネージャにロールバックすることができる。

20

【0032】

図3に示されるように、トランザクションがリソースマネージャ302上に既に準備されておりリソースマネージャ302からトランザクションマネージャ301に成功という投票を送ることができるようになる前にシステムがクラッシュする場合がある。その場合、トランザクションマネージャ301は、すべてのトランザクション分岐から成功という投票を受けることができないかもしれず、そうするとトランザクションをTLOGにロギングできない。よって、トランザクションマネージャ301は、すべての分岐トランザクションをリソースマネージャ302～304にそれぞれロールバックすればよい。このようにして、トランザクションマネージャは、このような一貫しかつ予測可能なトランザクションリカバリ手法を用いて、コミットされる分岐とロールバックされる分岐がある場合に複雑でヒューリスティックな完了を回避することができる。

30

【0033】

トランザクションマネージャリカバリログの必要性の軽減

トランザクション環境では、リカバリを実行しACID特性が確実に2フェーズコミットトランザクションマネージャの設計要件となるようにするために、リカバリ情報を安定した記憶装置に持続させることができる。

【0034】

たとえば、(XA)2フェーズロギングアルゴリズムは以下のステップを含み得る。

1. アプリケーションは、あるトランザクションにおいて複数のXAリソースを使用し、このトランザクションについてコミットを発行する。

【0035】

2. トランザクションマネージャは、すべてのXAリソース参加者に対してXAResource.prepareを発行する。

【0036】

3. トランザクションマネージャは、フォーマットIDおよびこのトランザクションのグローバルトランザクションIDを含むトランザクションログを持続させる。

【0037】

40

50



4. トランザクションマネージャは、すべてのXAリソース参加者に対してXAResource.commitを発行する。

【0038】

5. トランザクションマネージャは（全般的にゆっくり、数回に分けて等）トランザクション記録をページノ削除する。

【0039】

6. 障害が発生した場合、持続させているトランザクション記録を、関与するすべてのリソースマネージャに対してなされたXAResource.recoverコールから返されたインダウトxidと突き合わせ、それに応じてコミットまたはロールバックによるリカバリを実行する。

10

【0040】

リカバリ情報を安定した記憶装置に持続させる必要性は、リソースマネージャに対する実際のプロトコルネットワークコールとともに、2フェーズコミットトランザクションに要するパフォーマンスコストを課す可能性がある。この必要性にはまた、非常に利用し易い安定した記憶装置のために必要な共有ファイルシステムまたはデータベース記憶装置に要するアセット容量コスト、および、この記憶装置の管理に要する管理コストがかかる。これらのコストは、（同期した）複製およびその管理も必要であるサイト全体の事故のリカバリの場合、より一層大きくなるであろう。よって、トランザクションマネージャリカバリログの必要性を軽減することにより、既存のシステムと比較して大幅な改善が可能である。

20

【0041】

本発明の一実施の形態に従うと、以下の技術を用いて、2フェーズコミットコールの厳密な順序付けと、1つのリソース参加者をリカバリの決定に分類することに基づいた、トランザクションリカバリとACID特性を与えることにより、トランザクションログの必要性を軽減することが可能である。

【0042】

1. アプリケーションは、あるトランザクションにおいて複数のXAリソースを使用し、このトランザクションについてコミットを発行する。

【0043】

2. 特定のリソースを、トランザクション結果の「決定」として指定されるように構成することができる。このように構成されるリソースがなければ、トランザクションシステムが何らかを指定しこれを構成において持続させる。

30

【0044】

3. 2フェーズトランザクションにおいて1つのリソースを最初に使用するときまたはリソースの新たな組合せを最初に使用するときには必ず、この組合せをトランザクション処理サブシステムによって自動的に構成に取込み持続させる。この構成の持続性は、このようなトランザクションについて最初の準備が発行される直前に生じる。この構成の変更に失敗した場合はトランザクションをロールバックすることができる。必要ではないが、このステップによって、リカバリ中XAResource.recoverコールを必要とするリソースの組を減じるので、一層の最適化がなされる。

40

【0045】

4. トランザクションマネージャはトランザクションロギングを行なわない。

5. トランザクションマネージャは、すべてのXAリソース参加者に対してXAResource.prepareを発行し、「決定」リソースを最後に準備する。

【0046】

6. トランザクションマネージャは、すべてのXAリソース参加者に対してXAResource.commitを発行し、「決定」リソースを最後にコミットする。

【0047】

7. 障害が発生した場合、構成内のすべての非「決定」リソースに対してXAResource.recoverスキャンを実行し、インダウトxidを、決定リソースのXAResource.recovery

50

から返されたインダウト X i d と突き合わせる。決定のリストに一致する / 現存する X i d があれば、コミットによるリカバリを行ない、なければロールバックによるリカバリを行なう。

【 0 0 4 8 】

8 . トランザクションシステムのクリーンシャットダウンまたはリスタート中に、スタートアップ中に不必要なリカバリコールおよび処理が行なわれないうに構成をページ / マークする。

【 0 0 4 9 】

上記技術を以下のセクションでより詳細に説明することができる。

2 フェーズコミット ( 2 P C ) コールの厳密な順序付け

本発明のある実施の形態に従うと、システムは、2 フェーズコミットコールの厳密な順序付けに基づいて、トランザクションの処理における中間層トランザクションログ ( T L O G ) を排除することができる。

【 0 0 5 0 】

図 4 は、本発明のある実施の形態に従う、トランザクションミドルウェアマシン環境におけるトランザクション処理のための 2 フェーズコミット ( 2 P C ) コールの厳密な順序付けのサポートの一例を示す。図 4 に示されるように、グローバルトランザクション 4 1 0 は、トランザクションマネージャ ( T M ) 4 0 1 と、複数のリソースマネージャ ( R M ) 4 0 2 ~ 4 0 4 とを含む中間層トランザクションシステム 4 0 0 内でサポートすることができる。リソースマネージャ ( R M ) 4 0 2 ~ 4 0 3 はグローバルトランザクション 4 1 0 に参加し、リソースマネージャ 4 0 4 はグローバルトランザクション 4 1 0 に参加しない。本発明のある実施の形態に従うと、トランザクション 4 1 0 は複数のトランザクションマネージャにまたがっていない。

【 0 0 5 1 】

トランザクションマネージャ 4 0 1 は、あるリソース、たとえばデータベース 4 0 5 と関連付けられているリソースマネージャ ( R M ) 4 0 2 を、決定リソースとして指定することができる ( ステップ 1 ) 。リソースマネージャ ( R M ) 4 0 2 の、決定リソースとしての指定は、トランザクションマシン環境のための構成において持続させることができる。構成においてトランザクションにおける決定リソースが指定されないとき、トランザクションマネージャ 4 0 1 は、あるデータソースを決定リソースとして指定し構成において持続させることができる。加えて、決定リソース 4 0 2 は、メッセージキュー等の、データベース以外のリソースであってもよい。

【 0 0 5 2 】

さらに、2 フェーズトランザクション 4 1 0 の処理において 1 つのリソースを最初に使用するときまたはリソースの新たな組合せを最初に使用するときには必ず、この新たなリソースまたはリソースの新たな組合せを、トランザクションマネージャ 4 0 1 によって自動的に構成ファイル等の構成に取込み持続させることができる。この構成の持続性は、2 フェーズトランザクションについて最初の準備要求が発行される直前に生じ得る。この構成の変更失敗した場合はトランザクション 4 1 0 をロールバックすることができる。

【 0 0 5 3 】

本発明のある実施の形態に従うと、トランザクションマネージャ 4 0 1 は、決定リソース、たとえばリソースマネージャ 4 0 2 が、2 フェーズトランザクション 4 1 0 についてのすべての参加者リソースマネージャの中で、最後に準備されコミットされることを、保証することができる。

【 0 0 5 4 】

図 4 に示されるように、トランザクションマネージャ 4 0 1 は、決定リソースを準備する ( ステップ 4 ) 前に、最初にリソースマネージャ ( R M ) 4 0 3 を準備することができる ( ステップ 2 ) 。参加しているリソースマネージャ ( R M ) 4 0 2 ~ 4 0 3 から「 O K 」という投票を受ける ( ステップ 3 および 5 ) と、その後トランザクションマネージャ 4 0 1 は、リソースマネージャ ( R M ) 4 0 3 をコミットし ( ステップ 6 ) 、続いて決定リ

10

20

30

40

50

ソースをコミットすることができる（ステップ8）。このトランザクションは、トランザクションマネージャ401がリソースマネージャ402～403各々から成功という表示を受けた（ステップ7および9）後に、完了することができる。

【0055】

このように、トランザクションマネージャ401は、リソース/チェックポイントのロギングを行わずに済ますことができ、これには、準備フェーズの処理に成功した後にTLOGを書込むことが含まれる。このシステムは、中間層トランザクションシステム400にトランザクション410のリカバリを行なわせつつ、トランザクションのパフォーマンスを改善することができる。

【0056】

TLOGを使用しないグローバルトランザクションのリカバリ

図5は、本発明のある実施の形態に従う、トランザクションログ（TLOG）を使用しないグローバルトランザクションのリカバリの一例を示す。図5に示されるように、グローバルトランザクション510は、トランザクションマネージャ（TM）501と、複数のリソースマネージャ（RM）502～504とを含む中間層トランザクションシステム500においてサポートすることができる。

【0057】

トランザクション510のリカバリが必要なとき、トランザクションマネージャ510は、たとえば、決定リソース502に対してXA\_recover()コールを行なうことによって、リカバリしようと試みることができる（ステップ1）。トランザクションマネージャ501は、決定リソース502から、インダウトトランザクション（準備されているがコミットされていないトランザクション）のリストを受け取ることができる（ステップ2）。さらに、トランザクションマネージャ501は、このインダウトトランザクションのリストを用いて、グローバルトランザクションテーブル（global transaction table）（GTT506）をビルド/リビルドすることができる（ステップ3）。

【0058】

本発明の一実施の形態に従うと、2フェーズコミットトランザクションにおいて決定リソース502の準備に成功したとき、インダウトトランザクションのリストは、参加しているリソースマネージャ502～503に対してコミットするようにトランザクションマネージャ501が指示した中間層の未処理のトランザクションと、同一であり得る。

【0059】

次に、トランザクションマネージャ501は、たとえば、トランザクション中間層システム500における、参加しているリソースマネージャ503（ステップ4）および参加していないリソースマネージャ504（ステップ10）を含む、その他すべてのリソースマネージャに対して、XA\_recover()コールを行なうことによって、リカバリしようと試みることができる。

【0060】

たとえば、トランザクションマネージャ501は、参加しているリソースマネージャ503から、インダウトトランザクションのリストを受け取ることができる（ステップ5）。トランザクションマネージャ501は、このリストの中のインダウトトランザクションを、GTT506テーブルと突き合わせることにより、リカバリリストを生成することができる（ステップ6および7）。次に、トランザクションマネージャ501は、リカバリリストに基づいて、参加しているリソースマネージャ503をリカバリすることができる（ステップ8）。たとえば、トランザクションマネージャ501は、GTT506テーブルと一致するインダウトトランザクションをコミットすることができ、GTT506テーブルと一致しないインダウトトランザクションをロールバックすることができる。

【0061】

上記手順を、参加していないリソースであるリソースマネージャ504に対して行なうことができる。図5に示されるように、リソースマネージャ504に対してXA\_recover()コールを行なった後、トランザクションマネージャ501は、インダウトトランザクシ

10

20

30

40

50

ンのリストを受けることができ（ステップ11）、これをGTT506と突き合わせてリカバリリストを生成することができる（ステップ12）。次に、トランザクションマネージャ501は、リソースマネージャ504に対し、インダウトトランザクションのコミット/ロールバックを行なうことができる（ステップ14および15）。

【0062】

その他すべてのリソースマネージャ503～504がリカバリされると（ステップ9および15）、トランザクションマネージャ501は、決定リソースマネージャに対して、準備されたトランザクションのリスト内のすべてのトランザクションをコミットすることができる（ステップ16および17）。最後に、トランザクションマネージャ501は、GTT506内のエントリを削除することができる（ステップ18）。

10

【0063】

本明細書に記載のように、たとえ1つ以上のリソースマネージャがトランザクションの一部でなくても、その他すべてのリソースマネージャ503～504がリカバリされるまで、決定リソースマネージャ502に対しては、インダウトトランザクションのリストにあるトランザクションはコミットされない。

【0064】

本発明の一実施の形態では、準備フェーズにおいて、決定リソース502に、参加リソース502、503に関する追加情報を伝達し持続させることができるのであれば、トランザクションマネージャ501はトランザクションの参加リソース502、503に関する知識を有することができる。この追加情報を用いて、トランザクションマネージャ501は、わかっているすべてのリソースマネージャ502～504がリカバリされるのを待つことなく、決定リソースマネージャ502から返されたインダウトトランザクションのリストの中のトランザクションをコミットすることができる。トランザクションマネージャ501は、トランザクションのすべての参加者502～503がリカバリされるのを待つだけでよい。

20

【0065】

加えて、トランザクション中間層システム500のクリーンシャットダウンまたはリスタート中に、スタートアップ中の不必要なリカバリおよび処理を回避できるよう構成をページまたはマークすることができる。

【0066】

図6は、本発明のある実施の形態に従う、トランザクションログ（TLOG）を使用せずに2フェーズコミットトランザクションをリカバリするフローチャートの一例である。図6に示されるように、ステップ601で、トランザクションマネージャは、決定リソースに対してXA\_recover()コールを実行することができ、中間層トランザクションシステムにおいてコミットすべき未処理のトランザクションのリストを受け取る。次に、ステップ602で、トランザクションマネージャは、準備されたトランザクションのリストを使用してグローバルトランザクションテーブル（GTT）テーブルをリビルドすることができる。次に、ステップ603で、トランザクションマネージャは、トランザクションの参加リソースマネージャおよび不参加リソースマネージャを含む、システム内のその他のリソースマネージャ各々に対して、XA\_recover()コールを実行することができ、これらその他のリソースマネージャ各々からインダウトトランザクションのリストを受け取ることができる。ステップ604で、トランザクションマネージャは、インダウトトランザクションのリスト内の各トランザクションを、GTTテーブルと突き合わせることによってリカバリすることができる。このトランザクションは、GTTテーブルにおいて一致が発見された場合は、このトランザクションの取得元のリソースマネージャに対してコミットすることができ、そうでなければロールバックすることができる。最後に、ステップ605で、トランザクションマネージャは、その他のリソースマネージャに対するすべてのインダウトトランザクションのリカバリ後に、決定リソースからのインダウトトランザクションの中のトランザクションをコミットすることができる。

30

40

【0067】

50

本発明は、本開示の教示に従いプログラムされた1つ以上のプロセッサ、メモリ、および/またはコンピュータ読取可能な記録媒体を含む、従来の汎用または専用デジタルコンピュータ、コンピューティングデバイス、マシン、またはマイクロプロセッサを1つ以上用いて、適宜実現し得る。適切なソフトウェアコーディングは、熟練したプログラマーが本開示の教示に基づいて容易に準備できる。これはソフトウェア技術における当業者には明らかであろう。

【0068】

実施の形態によっては、本発明は、本発明のプロセスのうちいずれかを実行するためにコンピュータをプログラムするのに使用できる命令が格納された記録媒体または(1つまたは複数の)コンピュータ読取可能な媒体であるコンピュータプログラムプロダクトを含む。この記録媒体は、フロッピーディスク(登録商標)、光ディスク、DVD、CD-ROM、マイクロドライブ、および光磁気ディスクを含む、任意の種類ディスク、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、フラッシュメモリデバイス、磁気もしくは光カード、ナノシステム(分子メモリICを含む)、または、命令および/またはデータを記憶するのに適した任意の種類媒体もしくはデバイスを含み得るもの、これらに限定されない。

【0069】

本発明に関するこれまでの記載は例示および説明を目的として提供されている。すべてを網羅するまたは本発明を開示された形態そのものに限定することは意図されていない。当業者には数多くの変更および変形が明らかであろう。実施の形態は、本発明の原理およびその実際の応用を最もうまく説明することによって他の当業者が本発明のさまざまな実施の形態および意図している特定の用途に適したさまざまな変形を理解できるようにするために、選択され説明されている。本発明の範囲は、以下の特許請求の範囲およびその均等物によって定められることが意図されている。

【図1】

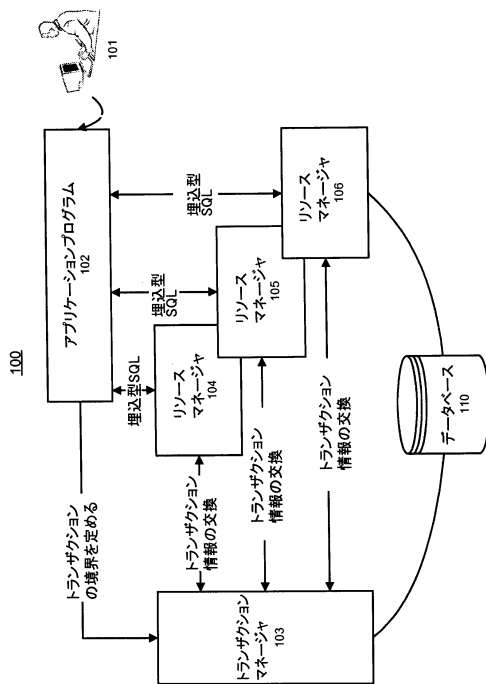


FIGURE 1

【図2】

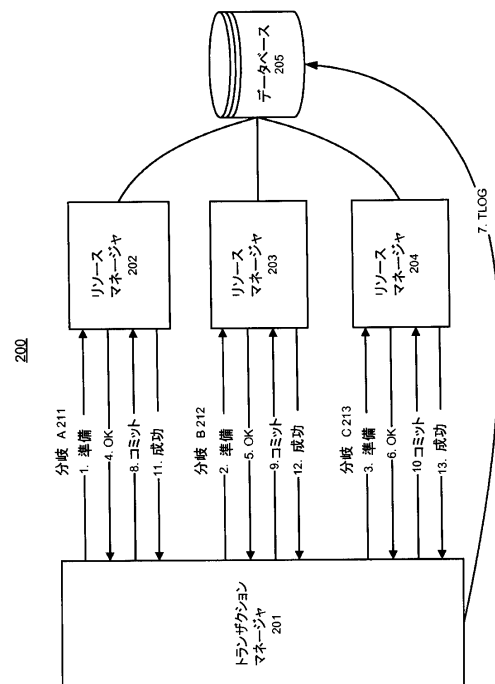


FIGURE 2

10

20



## フロントページの続き

(31)優先権主張番号 13/828,490

(32)優先日 平成25年3月14日(2013.3.14)

(33)優先権主張国 米国(US)

(31)優先権主張番号 13/828,590

(32)優先日 平成25年3月14日(2013.3.14)

(33)優先権主張国 米国(US)

(56)参考文献 米国特許出願公開第2005/0262077(US, A1)

特開平11-338746(JP, A)

特開平10-074155(JP, A)

特開平08-235039(JP, A)

米国特許出願公開第2005/0144299(US, A1)

特開平06-266592(JP, A)

樽澤広亨、田中孝清、目指せ！J2EEアーキテクト ミッションクリティカルシステム開発講座，月刊ジャバワールド，日本，東京：(株)IDGジャパン，2005年4月1日，第9巻第4号(通巻95号)，pp.130-141

(58)調査した分野(Int.Cl.，DB名)

G06F 12/00

G06F 9/46 - 9/54