

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6088527号
(P6088527)

(45) 発行日 平成29年3月1日 (2017.3.1)

(24) 登録日 平成29年2月10日 (2017.2.10)

(51) Int. Cl.

F I

G 0 6 F 9/52 (2006.01)

G 0 6 F 9/46 4 7 2 Z

請求項の数 11 (全 15 頁)

(21) 出願番号	特願2014-533604 (P2014-533604)	(73) 特許権者	502303739
(86) (22) 出願日	平成24年9月18日 (2012.9.18)		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2014-528609 (P2014-528609A)		アメリカ合衆国カリフォルニア州9406 5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26年10月27日 (2014.10.27)		
(86) 国際出願番号	PCT/US2012/055942	(74) 代理人	110001195
(87) 国際公開番号	W02013/048826		特許業務法人深見特許事務所
(87) 国際公開日	平成25年4月4日 (2013.4.4)	(72) 発明者	シェン, シュガン
審査請求日	平成27年8月11日 (2015.8.11)		中華人民共和国 ベイジン、フエンタイ・ ディストリクト、マー・ジア・プー・シー ・ルー、ナンバー・30、11-3-40 1
(31) 優先権主張番号	61/541,051		
(32) 優先日	平成23年9月29日 (2011.9.29)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	13/414,593		
(32) 優先日	平成24年3月7日 (2012.3.7)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 トランザクショナルミドルウェアマシン環境においてセルフチューニングロックメカニズム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】

トランザクショナルミドルウェアマシン環境においてロックメカニズムをサポートするためのシステムであって、

複数のプロセッサと、

共有メモリと、

前記複数のプロセッサ上で作動するオペレーティングシステムによって提供されるセマフォとを備え、

前記複数のプロセッサは前記共有メモリ中のデータにアクセスするように動作し、

1つ以上のプロセスに関連付けられるテストアンドセット (T A S) アセンブリコンポーネントをさらに備え、

各前記プロセスは、前記共有メモリ中のデータについてのロックを取得するために、前記 T A S アセンブリコンポーネントを用いて1回以上の T A S オペレーションを行なうように動作し、

前記 T A S アセンブリコンポーネントが特定数の T A S オペレーションを行ない、ロックの取得に失敗した後、プロセスは、前記セマフォ上でブロックされ、前記共有メモリ中のデータに対するロックの解除を待機するように動作し、

ロックメカニズムは、T A S オペレーションの許容される特定最大回数であるスピンカウントを使用し、

前記スピンカウントは、予め定められたプロセスを用いて周期的に決定される、システ

10

20

ム。

【請求項 2】

トランザクショナルミドルウェアマシン環境においてロックメカニズムをサポートする方法であって、

複数のプロセッサ上で作動するオペレーティングシステムに関連付けられたセマフォを提供することを備え、前記複数のプロセッサは、共有メモリ中のデータにアクセスするように動作し、さらに、

前記共有メモリ中のデータについてのロックを取得するために、1つ以上のプロセスによって、テストアンドセット(TAS)アセンブリコンポーネントを用いて1回以上のTASオペレーションを行なうことと、

前記TASアセンブリコンポーネントが特定数のTASオペレーションを行ない、ロックの取得に失敗した後に、プロセスを前記セマフォ上でブロックし、前記共有メモリ中のデータに対するロックの解除を待機することと、

TASオペレーションの許容される特定最大回数であるスピンカウントを使用することをロックメカニズムに許可することと、

予め定められたプロセスを用いて前記スピンカウントを周期的に決定することとをさらに備える、方法。

【請求項 3】

トランザクショナルミドルウェアマシン環境においてロックメカニズムをサポートする方法であって、

複数のプロセッサ上で作動するオペレーティングシステムに関連付けられたセマフォを提供することを備え、前記複数のプロセッサは、共有メモリ中のデータにアクセスするように動作し、さらに、

前記共有メモリ中のデータについてのロックを取得するために、1つ以上のプロセスによって、テストアンドセット(TAS)アセンブリコンポーネントを用いて1回以上のTASオペレーションを行なうことと、

前記TASアセンブリコンポーネントが特定数のTASオペレーションを行ない、ロックの取得に失敗した後に、プロセスを前記セマフォ上でブロックし、前記共有メモリ中のデータに対するロックの解除を待機することと、

TASオペレーションの許容される特定最大回数であるスピンカウントを使用することをロックメカニズムに許可することと、

ハードウェアコンフィギュレーションおよびアプリケーションシナリオの両方に基づいて前記スピンカウントを動的に決定することとをさらに備える、方法。

【請求項 4】

複数の同時トランザクションがある場合に、前記共有メモリ中のトランザクションデータを保護することをさらに備える、請求項 2 または 3 に記載の方法。

【請求項 5】

メタデータにおいて前記スピンカウントをあらかじめ構成することをさらに備える、請求項 2 ~ 4 のいずれか 1 項に記載の方法。

【請求項 6】

前記スピンカウントはアルゴリズムを用いて動的に決定され、前記アルゴリズムは、先の期間のスピン失敗の回数がスピン失敗限度を越え、前記先の期間のCPUアイドル比率がCPUアイドル比率限度未満である場合、前記スピンカウントは前記先の期間のスピンカウントから増大され、

前記CPUアイドル比率がCPUアイドル比率限度を越えた場合、前記スピンカウントは前記先の期間のスピンカウントから減少されることを特定する、請求項 2 に記載の方法。

【請求項 7】

前記TASオペレーションを特定回数試みた後でプロセスがデータに対するロックの取得に失敗した場合、スピン失敗が発生する、請求項 6 に記載の方法。

10

20

30

40

50

【請求項 8】

ロック所有者が覚醒してロックを解除する際に、前記セマフォを用いてロックを取得することをさらに備える、請求項 2 ～ 7 のいずれか 1 項に記載の方法。

【請求項 9】

最適値を見つけるために、前記スピンカウントを手動で微調整することをさらに備える、請求項 2 から 8 のいずれか 1 項に記載の方法。

【請求項 10】

請求項 2 ～ 9 のうちいずれか 1 項に記載の方法をコンピュータに行なわせる、プログラム。

【請求項 11】

トランザクショナルミドルウェアマシン環境においてロックメカニズムをサポートするためのシステムであって、

複数のプロセッサと、

共有メモリと、

前記複数のプロセッサ上で作動するオペレーティングシステムによって提供されるセマフォとを備え、

前記複数のプロセッサは前記共有メモリ中のデータにアクセスするように動作し、

1 つ以上のプロセスに関連付けられるテストアンドセット (TAS) アセンブリコンポーネントをさらに備え、

各前記プロセスは、前記共有メモリ中のデータについてのロックを取得するために、前記 TAS アセンブリコンポーネントを用いて 1 回以上の TAS オペレーションを行なうように動作し、

前記 TAS アセンブリコンポーネントが特定数の TAS オペレーションを行ない、ロックの取得に失敗した後、プロセスは、前記セマフォ上でブロックされ、前記共有メモリ中のデータに対するロックの解除を待機するように動作し、

ロックメカニズムは、TAS オペレーションの許容される特定最大回数であるスピンカウントを使用し、

前記スピンカウントは、ハードウェアコンフィギュレーションおよびアプリケーションシナリオの両方に基づいて動的に決定される、システム。

【発明の詳細な説明】

【技術分野】

【0001】

著作権表示

この特許文書の開示の一部は、著作権の保護下にある内容を含む。特許商標庁の特許ファイルまたは記録に現れる限りにおいては、著作権所有者は、誰でも当該特許文書または特許開示を複製することについて異議はないが、そうでなければ如何なる場合で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のようなソフトウェアに関し、特定的にはトランザクショナルミドルウェアマシン環境をサポートすることに関する。

【背景技術】

【0003】

背景

トランザクショナルミドルウェアシステム、すなわちトランザクション指向のミドルウェアは、組織内の様々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のような新たな技術の発展により、トランザクショナルミドルウェアの性能をさらに向上させる必要がある。これらは、一般に発明の実施例が対処することが意図される分野である。

【発明の概要】

【課題を解決するための手段】

【0004】

概要

本願明細書に記載されるのは、同時トランザクションがある場合に、共有メモリ中のトランザクションデータを保護するためにトランザクショナルミドルウェアシステムにおいてサポートされることができるロックメカニズムである。トランザクショナルミドルウェアマシン環境は、各々が共有メモリ中のデータにアクセスするように動作する複数のプロセッサ上で作動するオペレーティングシステムを含む。トランザクショナルミドルウェアマシン環境は、オペレーティングシステムによって提供されるセマフォ、および1つ以上のプロセスに関連付けられるテストアンドセット（TAS）アセンブリコンポーネントも含む。各プロセスは、共有メモリ中のデータについてのロックを取得するために、TASアセンブリコンポーネントを用いて1回以上のTASオペレーションを行なうことができる。さらに、TASコンポーネントが特定数のTASオペレーションを行なった後でロックの取得に失敗した後、プロセスはセマフォをブロックし、共有メモリ中のデータに対するロックの解除を待機することができる。

10

【図面の簡単な説明】

【0005】

【図1】発明の実施例に係る、セルフチューニングロックメカニズムをサポートするトランザクショナルミドルウェアマシン環境の例証を示す図である。

20

【図2】発明の実施例に係る、トランザクショナルミドルウェアマシン環境においてセルフチューニングロックメカニズムをサポートするための典型的なフローチャートを例示する図である。

【図3】発明の実施例に係る、トランザクショナルミドルウェアマシン環境においてロックメカニズムをサポートするためのシステムの機能ブロック図である。

【発明を実施するための形態】

【0006】

詳細な説明

本願明細書に記載されるのは、複数のプロセッサを有する高速マシンを利用することができるTuxedo（登録商標）のようなトランザクショナルミドルウェアシステムおよび高性能ネットワーク接続をサポートするためのシステムおよび方法である。同時トランザクションがある場合、共有メモリ中のトランザクションデータを保護するために、トランザクショナルミドルウェアシステムにおいてロックメカニズムをサポートすることができる。トランザクショナルミドルウェアマシン環境は、複数のプロセッサ上で作動するオペレーティングシステムによって提供されるセマフォを含む。複数のプロセッサは、共有メモリ中のデータにアクセスすることができる。トランザクショナルミドルウェアマシン環境は、1つ以上のプロセスに関連付けられるテストアンドセット（TAS）アセンブリコンポーネントも含む。各上記プロセスは、共有メモリ中のデータについてのロックを取得するために、TASアセンブリコンポーネントを用いて1回以上のTASオペレーションを行なうように動作する。その上、TASコンポーネントが多くのTASオペレーションを行ない、ロックの取得に失敗した後、プロセスは、セマフォ上でブロックされ、共有メモリ中のデータに対するロックの解除を待機する。

30

40

【0007】

本発明の実施例に従うと、当該システムは、高性能ハードウェア、たとえば64ビットプロセッサ技術、高性能な大きなメモリ、ならびに冗長なインフィニバンドおよびイーサネット（登録商標）ネットワーキングと、WebLogic Suiteのようなアプリケーションサーバまたはミドルウェア環境との組合せを含み、これにより、迅速に備えられ得るとともにオンデマンドでスケール変更可能な大規模並列メモリグリッドを含む完全なJava（登録商標）EEアプリケーションサーバコンプレックスを提供する。本発明の実施例に従うと、当該システムは、アプリケーションサーバグリッド、ストレージエリ

50

アネットワーク、およびインフィニバンド（IB）ネットワークを提供するフルラック、ハーフラック、もしくはクォーターラック、または他の構成としてデプロイされ得る。ミドルウェアマシンソフトウェアは、たとえばWebLogic Server、JRockitまたはHotspot JVM、Oracle Linux（登録商標）またはSolaris、およびOracle VMといった、アプリケーションサーバと、ミドルウェアと、他の機能性とを提供し得る。本発明の実施例に従うと、当該システムは、IBネットワークによって互いに通信する複数のコンピュータノードと、IBスイッチゲートウェイと、ストレージノードまたはユニットとを含み得る。ラック構成として実施される場合、当該ラックの未使用部分は、空のままとされるか、またはフィラー（filler）によって占有され得る。

10

【0008】

本願明細書において「Sun Oracle Exalogic」または「Exalogic」と称される本発明の実施例に従うと、当該システムは、Oracle Middleware SW suiteまたはWeblogicといったミドルウェアまたはアプリケーションサーバソフトウェアをホスティングするための、デプロイが容易なソリューションである。本願明細書に記載されるように、実施例に従うと、当該システムは、1つ以上のサーバと、ストレージユニットと、ストレージネットワーキングのためのIBファブリックと、ミドルウェアアプリケーションをホストするために要求されるすべての他のコンポーネントとを含む「グリッド・イン・ア・ボックス（grid in a box）」である。たとえばReal Application ClustersおよびExalogic open storageを用いて大規模並列グリッドアーキテクチャを活用することにより、すべてのタイプのミドルウェアアプリケーションのために有意な性能が与えられ得る。このシステムは、線形のI/Oスケラビリティとともに向上した性能を与え、使用および管理が簡易であり、ミッションクリティカルな可用性および信頼性を与える。

20

【0009】

発明の実施例に従うと、Tuxedo（登録商標）は、高性能の分散ビジネスアプリケーションの構築、実行および管理を可能にし、多くの多階層アプリケーション開発ツールによってトランザクショナルミドルウェアとして使用されてきた一組のソフトウェアモジュールである。Tuxedo（登録商標）は、分散コンピューティング環境において分散トランザクション処理を管理するために使用することができるミドルウェアプラットフォームである。無限のスケラビリティおよび規格に基づいたインターオペラビリティを提供しつつ、企業レガシーアプリケーションをアンロックし、それらをサービス指向のアーキテクチャに拡張するための検証済みのプラットフォームである。

30

【0010】

発明の実施例に従うと、Tuxedo（登録商標）システムのようなトランザクショナルミドルウェアシステムは、Exalogicミドルウェアマシンのような複数のプロセッサを有する高速マシン、およびインフィニバンド（IB）ネットワークのような高性能ネットワーク接続を利用することができる。

【0011】

セルフチューニングロックメカニズム

発明の実施例に従うと、同時トランザクションがある場合、セルフチューニングロックメカニズムは、共有メモリ中のトランザクションデータを保護するためにトランザクショナルミドルウェアシステムにおいてサポートされることができる。セルフチューニングロックメカニズムを用いると、トランザクショナルミドルウェアマシン環境は、大規模な同時トランザクションを有するアプリケーションのようなトランザクショナルアプリケーションシナリオにおける著しいスループットの向上を実現することができる。

40

【0012】

図1は、発明の実施例に係る、セルフチューニングロックメカニズムをサポートするトランザクショナルミドルウェアマシン環境の例証を示す。図1に示されるように、トラン

50

ザクショナルミドルウェアマシンは、オペレーティングシステム（OS）104をサポートする複数のCPU131～134と、様々なトランザクショナルデータ121～123を含む共有メモリ103とを含む。複数の同時トランザクション111～112を有するトランザクショナルアプリケーション101は、トランザクショナルサーバ102において複数のプロセス113～115上で作動することができ、その各々は、アトミックTAS（テストアンドセット）アセンブリ107を用いて、有効なロックメカニズムを実施することができる。同時トランザクションがある場合、ロックメカニズムは、共有メモリ中のトランザクションデータを保護することができる。その上、トランザクショナルアプリケーションにおけるプロセスは、OSによって提供されるセマフォメカニズム106を用いて、必要に応じてデータ122に対するロックを取得することができる。

10

【0013】

一実施例に従うと、プロセスがデータ122に対するロックの獲得を必要とする場合、プロセスはTASオペレーションを多数回行なうことができる。システムは、許容されるTASオペレーションの回数である目標スピンカウントを特定することができる。目標スピンカウントは、あらかじめ構成することができるか、または動的に決定される。

【0014】

目標スピンカウントに到達する前にロックが利用可能になった場合、プロセスは、OSによって提供されるセマフォメカニズムよりもはるかに低いコストでロックを取得することができる。一方、この期間中にロックが利用可能にならない場合、セマフォをブロックし、ロック所有者が覚醒してロックを解除するまで待機するようにプロセスを構成することができる。

20

【0015】

発明の実施例に従うと、目標スピンカウント値は、ハードウェアコンフィギュレーションのコンテキストとアプリケーションシナリオとにおいて決めることができる。ユーザは、最適値を見つけるために、スピンカウント値を手動で微調整することができる。CPUの使用とロックを獲得するための時間との間にトレードオフがあるため、決定はいくつかの状況では明白ではない場合がある。たとえば、ユーザは、より短い時間枠でロックを獲得するために、より多くのCPU電源を消費してより多くのTASオペレーションを行わなければならない場合がある。したがって、あらゆる場合について明白な最適目標スピンカウント値がない場合がある。

30

【0016】

ユーザレベルセマフォ実装例の欠点の一つは、トランザクショナルアプリケーションが、特定のマシンタイプに依存して目標スピンカウントをリアルタイムに動的に調節することができない場合がある点である。一般に、ユーザレベルセマフォは静的に構成された目標スピンカウント値のみを使用し、ユーザは、実験的慣習のみによって目標スピンカウント値を手動で調節することができる。最適なスピンカウント値はマシンに依存し、すべてのプラットフォームに当てはまる1つの値はないため、より好適なアプローチは、目標スピンカウント値を動的にかつリアルタイムに算出するためのメカニズムを採用することである。

【0017】

40

発明の実施例に従うと、目標スピンカウント値を共有メモリに記憶させることができる。Tuxedo（登録商標）デーモンプロセスのような特殊なプロセスは、先の期間に収集された動作情報に従ってスピンカウント値を周期的に変更することができる。たとえば、Tuxedo（登録商標）デーモンは、デフォルトで目標スピンカウント値を5秒につき一度更新することができる。

【0018】

一実施例に従うと、アルゴリズムを用いて目標スピンカウント値を設定することができる。CPUアイドルレートが低い、または過剰に多くのTASオペレーションがロックの取得に失敗し、システムがセマフォに切替えられた場合、アルゴリズムは、目標スピンカウント値を増大させることができる。さらに、CPUアイドルレートが高すぎる場合、

50

アルゴリズムは、目標スピンカウント値を低下させることができる。

【0019】

図2は、発明の実施例に係る、トランザクショナルミドルウェアマシン環境においてセルフチューニングロックメカニズムをサポートするための典型的なフローチャートを例示する。図2に示されるように、ステップ201において、システムは複数のプロセッサ上で作動するオペレーティングシステムに関連付けられたセマフォを提供することができ、複数のプロセッサは、共有メモリ中のデータにアクセスするように動作する。次いで、ステップ202において、1つ以上のプロセスのうちの1つのプロセスは、共有メモリ中のデータについてのロックを取得するために、テストアンドセット(TAS)アセンブリコンポーネントを用いて1回以上のTASオペレーションを行なうことができる。最後に、ステップ203において、TASコンポーネントが特定数のTASオペレーションを行ない、ロックの取得に失敗した後、プロセスはセマフォ上でブロックされ、共有メモリ中のデータに対するロックの解除を待機することができる。

10

【0020】

Tuxedo(登録商標)におけるSPINCOUNT値の構成

発明の実施例に従うと、Tuxedo(登録商標)コンフィギュレーションファイルにおけるSPINCOUNTパラメータのようなメタデータを用いて、目標スピンカウントを特定することができる。SPINCOUNTは、静的な構成値または動的な構成値のいずれかであり得る。

【0021】

20

たとえば、Tuxedo(登録商標)は、SPINCOUNTの静的に構成された値を用いて、システムレベルセマフォ上でブロックされる前に掲示板ロック待機が何回スピンしているかを判定する。このアルゴリズムの欠点は、ユーザによって設定された値が特定のプラットフォーム上のSPINCOUNTの最適値ではない点である。なぜなら、SPINCOUNTの最適値は、CPU量、作業負荷、掲示板(BB)ロックの待機数などといった多くの動的要因に依存するからである。

【0022】

その上、Tuxedo(登録商標)は、実行時環境を考慮に入れながらSPINCOUNTの値を動的に調整することができる。アルゴリズムを用いて、SPINCOUNTパラメータについての適切な値を決定することができる。過剰に多くのTASオペレーションが先の期間で失敗しており、システムがセマフォに切替えられ、十分なCPUアイドルレートがあった場合、システムはSPINCOUNTを増大させることができる。一方、CPUアイドルレートが高すぎた場合、システムはSPINCOUNTを減少させることができる。

30

【0023】

上記のアルゴリズムは、CPUの使用、たとえばCPUアイドルレートと、SPIN失敗の比率、たとえばロックを取得するための10000回のオペレーションにおけるSPIN失敗の比率とに基づく。SPIN失敗の比率は、TASオペレーションによる代わりに、セマフォによって何回ロックが取得されるかを示すことができる。

【0024】

40

プロセスがアクティブな状態にあるとき、プロセスはSPINモードにあることができる。プロセスがTASオペレーションを多数回行なおうとした後でロックの取得に失敗した、SPIN失敗と称される場合には、プロセスをセマフォ上でブロックすることができる。プロセスは、SPIN失敗が起こる前にロックを獲得するために、再三TASオペレーションを行なうことを試みることができる。構成可能なパラメータ、たとえばSPINCOUNTを用いて、呼び出され行なわれるべきTASオペレーションの回数を特定することができる。

【0025】

発明の実施例に従うと、最小アイドルCPUレートおよびSPIN失敗レートは、コンフィギュレーションファイルにおいてメタデータを用いて定義されることができる。たと

50

えば、Tuxedo（登録商標）コンフィギュレーションファイルには、最小アイドルCPUレートを定義するためのMINIDLECPUパラメータが存在し得る。その値は、1～100の範囲を有し、20がデフォルトである。その上、SPIN失敗レートを定義するためのFACTORパラメータが存在し得る。その値は、1～10000の範囲を有し、1000がデフォルトである。

【0026】

最小アイドルCPUレートおよびSPIN失敗レートが与えられると、システムは、各走査ユニットにおいてSPINCOUNTを調整することができる。次いで、SPIN失敗レートが高すぎる（たとえばSPIN失敗レートが $(1 / \text{FACTOR} * 1.1)$ ）よりも大きく設定される）か、またはアイドルCPU時間が十分であり（たとえば、アイドルレート $> \text{MINIDLECPU}\% + 0.05$ ）、新しい $\text{SPINCOUNT} = \text{古いSPINCOUNT} + \text{古いSPINCOUNT} \times (\text{cpu_idle_time} / \text{cpu_user_time})$ であり、最大SPINCOUNTを10,000,000に設定することができる場合、システムはSPINCOUNTを増大させることができる。その他の場合、アイドルCPUレートが低すぎ（たとえばアイドルレート $< \text{MINIDLECPU}\% - 0.05$ ）、新しい $\text{SPINCOUNT} = \text{古いSPINCOUNT} / 4$ であり、最小SPINCOUNTが50000に設定される場合、システムはSPINCOUNTを減少させることができる。

【0027】

以下のリスト1は、SPINTUNING構成のためのTuxedo（登録商標）の例である。

【0028】

10

20

【数 1】

```

*RESOURCES
IPCKEY      123456
DOMAINID    simpapp
MASTER      ALLEN
MAXACCESSERS 10
MAXSERVERS  5
MAXSERVICES 10
MODEL        SHM
LDBAL        N
OPTIONS      EXALOGIC,SPINTUNING

```

10

```

*MACHINES
ALLENHOST    LMID="ALLEN"
APPDIR="/home/allen/Workspace/Tuxedo11gR1PS2/simpdir"
TUXCONFIG="/home/allen/Workspace/Tuxedo11gR1PS2/simpdir/tuxconfig"
TUXDIR="/home/allen/Software/OraHome/tuxedo11gR1PS2"
SPINTUNING_TARGET=1000
SPINTUNING_MINIDLECPU=20
*GROUPS
GROUP1
LMID=ALLEN GRPNO=1 OPENINFO=NONE

```

20

```

*SERVERS
DEFAULT:
CLOPT="-A"
simplserv SRVGRP=GROUP1 SRVID=1

```

```

*SERVICES
TOUPPER

```

リスト 1

【0029】

30

上記の例に示されるように、コンフィギュレーションファイルの *MACHINES セクションは、構成チューニング目標に使用される属性「SPINTUNING_TARGET」を含む。「SPINTUNING_TARGET」の値は数値であり、「0」以上かつたとえば「10000」以下であり得る。0の値は、バイナリに組込まれた値が使用されることを示す。組込まれた値は1000である。「SPINTUNING_TARGET」のデフォルト値は0である。

【0030】

上記の例において、SPINTUNING_TARGETの値は、掲示板がロック1000回ごとにシステムセマフォによってロックされるべき回数がせいぜい1回であることを意味する。システムは、SPINTUNING_TARGETのより大きな値を満たすためにより多くのCPUが消費される状態で、SPINCOUNTの値を増大させることができる。オプションSPINTUNINGが特定される場合のみ、ゼロでない値で属性を設定することができる。

40

【0031】

上記の例において示されるように、コンフィギュレーションファイルの *MACHINES セクションは、アイドルCPUレートを特定するために使用される属性「SPINTUNING_MINIDLECPU」を含む。アイドルCPUレートをシステムによって使用して、好適なSPINCOUNTを動的に見つけることができる。SPINCOUNTが大きいほど、システムが使用するCPUが多くなる。ユーザは、「SPINTUNING_MINIDLECPU」によって最小アイドルCPUレートを設定して、過剰なC

50

P U消費を回避することができる。

【0032】

「SPINTUNING__MINIDLECPU」の値は数値であり、百分率である。「0」以上「100」以下であり得る。0の値は、与えられたバイナリに組込まれた値が使用されるべきであることを示す。たとえば、Tuxedo（登録商標）では、組込まれた値を20に設定することができ、「SPINTUNING__MINIDLECPU」のデフォルト値は0である。オプションSPINTUNINGが特定される場合にのみ、ゼロでない値に属性を設定することができる。

【0033】

いくつかの実施例に従うと、図3は、トランザクショナルミドルウェアマシン環境において、上述したような本発明の原理に従って構成されるロックメカニズムをサポートするためのシステム1000の機能ブロック図を示す。システム1000の機能ブロックは、本発明の原理を実施するために、ハードウェア、ソフトウェア、またはハードウェアおよびソフトウェアの組合せによって実現され得る。当業者であれば、図3に記載される機能ブロックは、本発明の原理を実現するために、組み合されてもよく、またはサブブロックへと分離されてもよいということが理解される。したがって、本願明細書における記載は、本願明細書において記載される機能ブロックの任意の可能な組合せ、分離、またはさらなる定義をサポートし得る。

【0034】

図3に示されるように、トランザクショナルミドルウェアマシン環境においてロックメカニズムをサポートするためのシステム1000は、提供ユニット1100およびTASアセンブリコンポーネント1200を備える。提供ユニット1100は、複数のプロセッサ上で作動するオペレーティングシステムに関連付けられたセマフォを提供し、複数のプロセッサは、共有メモリ中のデータにアクセスするように動作する。TASアセンブリコンポーネント1200は、1つ以上のプロセスに関連付けられ、各上記プロセスは、共有メモリ中のデータについてのロックを取得するために、TASアセンブリコンポーネント1200を用いて1回以上のTASオペレーションを行なうように動作する。TASコンポーネント1200が特定数のTASオペレーションを行ない、ロックの取得に失敗した後、プロセスは、セマフォ上でブロックされ、共有メモリ中のデータに対するロックの解除を待機するように動作する。

【0035】

いくつかの実施例において、システム1000は、複数の同時トランザクションがある場合、共有メモリ中のトランザクションデータを保護するための保護ユニット1300をさらに備える。

【0036】

いくつかの実施例において、システム1000は、TASオペレーションの許容される特定最大回数であるスピンカウントを使用することをロックメカニズムに許可するための許可ユニット1400をさらに備える。

【0037】

いくつかの実施例において、システム1000は、メタデータにおいてスピンカウントをあらかじめ構成するためのプリコンフィギュアリングユニット1500をさらに備える。

【0038】

いくつかの実施例において、システム1000は、ハードウェアコンフィギュレーションおよびアプリケーションシナリオの両方に基づいてスピンカウントを動的に決定するための動的決定ユニット1600をさらに備える。

【0039】

いくつかの実施例において、システム1000は、特殊なプロセスを用いてスピンカウントを周期的に決定するための周期的決定ユニット1700をさらに備える。

【0040】

いくつかの実施例において、アルゴリズムを用いてスピンカウントが動的に決定される。アルゴリズムは、先の期間におけるスピン失敗の回数がスピン失敗限度を越え、先の期間のCPUアイドル比率がCPUアイドル比率限度未満である場合、スピンカウントは先の期間のスピンカウントから増大され、CPUアイドル比率がCPUアイドル比率限度を越えた場合、スピンカウントは先の期間のスピンカウントから減少されることを特定する。

【0041】

いくつかの実施例において、TASオペレーションを特定回数試みた後でプロセスがデータに対するロックの取得に失敗すると、スピン失敗が発生する。

【0042】

いくつかの実施例において、システム1000は、ロック所有者が覚醒してロックを解除する際にセマフォを用いてロックを取得するための使用ユニット1800をさらに備える。

【0043】

いくつかの実施例において、システム1000は、スピンカウントを手動で微調整して最適値を見つけるための微調整ユニット1900をさらに備える。

【0044】

他の実施例は、トランザクショナルミドルウェアマシン環境においてロックメカニズムをサポートするための装置を含み、複数のプロセッサ上で作動するオペレーティングシステムに関連付けられたセマフォを提供するための手段を備え、複数のプロセッサは共有メモリ中のデータにアクセスするように動作し、さらに、共有メモリ中のデータについてのロックを取得するために、1つ以上のプロセスのうちの1つのプロセスによって、テストアンドセット(TAS)アセンブリコンポーネントを用いて、1回以上のTASオペレーションを行なうための手段と、TASコンポーネントが特定回数のTASオペレーションを行ない、ロックの取得に失敗した後、プロセスによってセマフォ上でブロックし、共有メモリ中のデータに対するロックの解除を待機するための手段とを備える。

【0045】

別の実施例において、装置は、複数の同時トランザクションがある場合に共有メモリ中のトランザクションデータを保護するための手段をさらに備える。

【0046】

別の実施例において、装置は、許容されるTASオペレーションの特定最大回数であるスピンカウントを使用することをロックメカニズムに許可するための手段をさらに備える。

【0047】

別の実施例において、装置は、メタデータにおいてスピンカウントをあらかじめ構成するための手段をさらに備える。

【0048】

別の実施例において、装置は、ハードウェアコンフィギュレーションおよびアプリケーションシナリオの両方に基づいてスピンカウントを動的に決定するための手段をさらに備える。

【0049】

別の実施例において、装置は、特殊なプロセスを用いてスピンカウントを周期的に決定するための手段をさらに備える。

【0050】

実施例において、アルゴリズムを用いてスピンカウントが動的に決定される装置をさらに備える。アルゴリズムは、先の期間のスピン失敗の回数がスピン失敗限度を越え、先の期間のCPUアイドル比率がCPUアイドル比率限度未満である場合、スピンカウントが先の期間のスピンカウントから増大され、CPUアイドル比率がCPUアイドル比率限度を越えた場合、スピンカウントが先の期間のスピンカウントから減少されることを特定する。

10

20

30

40

50

【 0 0 5 1 】

別の実施例において、装置では、T A S オペレーションを特定回数試みた後でプロセスがデータに対するロックの取得に失敗した場合、スピン失敗が発生する。

【 0 0 5 2 】

別の実施例において、装置は、ロック所有者が覚醒してロックを解除する際に、セマフォを用いてロックを取得するための手段をさらに備える。

【 0 0 5 3 】

別の実施例において、装置は、スピンカウントを手動で微調整して最適値を見つけるための手段をさらに備える。

【 0 0 5 4 】

さらに別の実施例は、トランザクショナルミドルウェアマシン環境においてロックメカニズムをサポートするためのシステムを含み、複数のプロセッサ上で作動するオペレーティングシステムに関連付けられたセマフォを提供するためのユニットを備え、複数のプロセッサは共有メモリ中のデータにアクセスするように動作し、さらに、1つ以上のプロセスに関連付けられたテストアンドセット(T A S)アセンブリコンポーネントを備え、各上記プロセスは、共有メモリ中のデータについてのロックを取得するために、T A Sアセンブリコンポーネントを用いて1回以上のT A Sオペレーション行なうように動作し、T A Sコンポーネントが特定数のT A Sオペレーションを行ない、ロックの取得に失敗した後、プロセスは、セマフォ上でブロックされ、共有メモリ中のデータに対するロックの解除を待機するように動作する。

【 0 0 5 5 】

別の実施例において、システムは、複数の同時トランザクションがある場合に、共有メモリ中のトランザクションデータを保護するためのユニットをさらに備える。

【 0 0 5 6 】

別の実施例において、システムは、T A Sオペレーションの許容される特定最大回数であるスピンカウントを使用することをロックメカニズムに許可するためのユニットをさらに備える。

【 0 0 5 7 】

別の実施例において、システムは、メタデータにおいてスピンカウントをあらかじめ構成するためのユニットをさらに備える。

【 0 0 5 8 】

別の実施例において、システムは、ハードウェアコンフィギュレーションおよびアプリケーションシナリオの両方に基づいてスピンカウントを動的に決定するためのユニットをさらに備える。

【 0 0 5 9 】

別の実施例において、システムは、特殊なプロセスを用いてスピンカウントを周期的に決定するためのユニットをさらに備える。

【 0 0 6 0 】

別の実施例は、スピンカウントがアルゴリズムを用いて動的に決定されるシステムを含み、アルゴリズムは、先の期間におけるスピン失敗の回数がスピン失敗限度を越え、先の期間のC P Uアイドル比率がC P Uアイドル比率限度未満である場合、スピンカウントは先の期間のスピンカウントから増大され、C P Uアイドル比率がC P Uアイドル比率限度を越えた場合、スピンカウントは先の期間のスピンカウントから減少されることを特定する。

【 0 0 6 1 】

別の実施例は、T A Sオペレーションを特定回数試みた後でプロセスがデータに対するロックの取得に失敗した場合、スピン失敗が発生するシステムを含む。

【 0 0 6 2 】

別の実施例は、ロック所有者が覚醒してロックを解除する際に、セマフォを用いてロックを取得するための手段をさらに備えるシステムを含む。

【 0 0 6 3 】

さらに別の実施例は、スピncountを手動で微調整して最適値を見つけるためのユニットをさらに備えるシステムを含む。

【 0 0 6 4 】

本発明は、1つ以上のプロセッサ、メモリ、および/または本開示の教示に従ってプログラムされたコンピュータ可読記憶媒体を含む1つ以上の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装置、マシン、またはマイクロプロセッサを用いて簡便に実施され得る。ソフトウェア技術の当業者には明らかであるように、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によって本開示の教示に基づき容易に用意され得る。

10

【 0 0 6 5 】

いくつかの実施例では、本発明は、本発明の処理のいずれかを実行するようコンピュータをプログラムするのに用いられ得る命令を格納した記憶媒体またはコンピュータ可読媒体であるコンピュータプログラムプロダクトを含む。当該記憶媒体は、フロッピー（登録商標）ディスク、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タイプのディスク、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、フラッシュメモリ素子、磁気または光学カード、ナノシステム（分子メモリICを含む）、または命令および/またはデータを格納するのに好適な任意のタイプの媒体もしくは装置を含み得るが、これらに限定されない。

【 0 0 6 6 】

20

本発明の上記の記載は、例示および説明目的で与えられている。網羅的であることまたは開示されたそのものの形態に本発明を限定することを意図したものではない。当業者にとっては、多くの修正例および変形例が明確であろう。上記の実施例は、本発明の原理およびその実質的な適用をもっともよく説明するために選択および記載されたものであり、これにより他の当業者が、特定の使用に好適なさまざまな修正例を考慮して、さまざまな実施例について本発明を理解するのが可能になる。本発明の範囲は、添付の特許請求の範囲およびそれらの均等物によって定義されることが意図される。

【図 3】

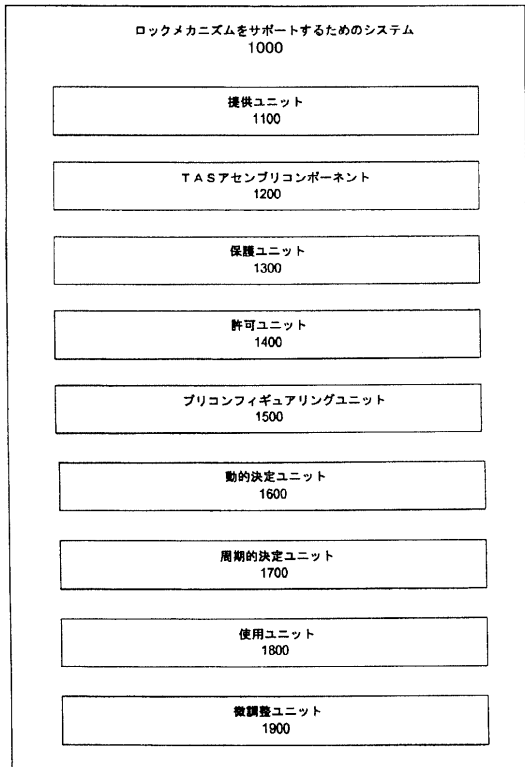


FIGURE 3

【図 1】

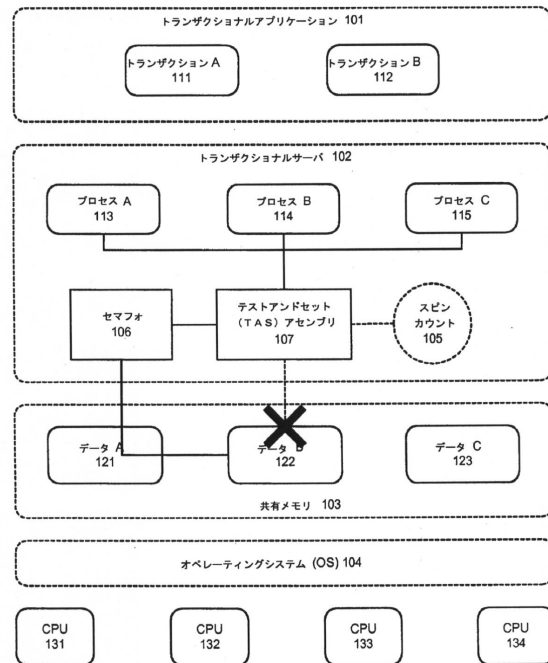


FIGURE 1

【図 2】

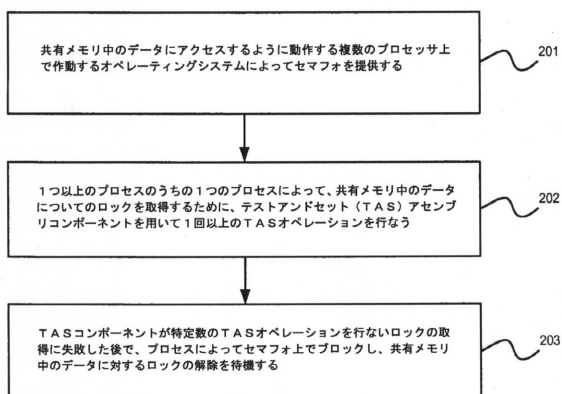


FIGURE 2

フロントページの続き

(72)発明者 リー, シアンドン

中華人民共和国 ペイジン、シェンウー・ディストリクト、チャンチュンジェシーリー、9 - 5 - 1

審査官 田中 幸雄

(56)参考文献 特開平8 - 3 2 8 9 3 3 (J P , A)

津島浩樹, UNIXプラットフォームでのOracle設定Tips, DB Magazine
 , 日本, 株式会社翔泳社, 2001年 4月 1日, 第10巻 第16号, 81 - 101ページ

(58)調査した分野(Int.Cl. , DB名)

G 0 6 F 9 / 5 2