

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4866861号
(P4866861)

(45) 発行日 平成24年2月1日(2012.2.1)

(24) 登録日 平成23年11月18日(2011.11.18)

(51) Int. Cl. F I
G06F 11/34 (2006.01) G O 6 F 11/34 S
G06F 11/30 (2006.01) G O 6 F 11/34 B
 G O 6 F 11/30 3 1 O H

請求項の数 8 (全 15 頁)

(21) 出願番号	特願2007-547411 (P2007-547411)	(73) 特許権者	390009531
(86) (22) 出願日	平成17年10月26日 (2005.10.26)		インターナショナル・ビジネス・マシーンズ・コーポレーション
(65) 公表番号	特表2008-537610 (P2008-537610A)		INTERNATIONAL BUSINESS MACHINES CORPORATION
(43) 公表日	平成20年9月18日 (2008.9.18)		アメリカ合衆国10504 ニューヨーク州 アーモンク ニュー オーチャードロード
(86) 国際出願番号	PCT/EP2005/055580		
(87) 国際公開番号	W02006/066990	(74) 代理人	100108501
(87) 国際公開日	平成18年6月29日 (2006.6.29)		弁理士 上野 剛史
審査請求日	平成20年6月30日 (2008.6.30)	(74) 代理人	100112690
(31) 優先権主張番号	04300953.9		弁理士 太佐 種一
(32) 優先日	平成16年12月24日 (2004.12.24)	(74) 代理人	100091568
(33) 優先権主張国	欧州特許庁 (EP)		弁理士 市位 嘉宏

最終頁に続く

(54) 【発明の名称】 トランザクション・ベースのシステムを監視するための方法及びシステム

(57) 【特許請求の範囲】

【請求項1】

システム・リソース・モニタ及びトランザクション性能モニタを有するトランザクション・ベースのシステムを監視する方法であって、これらのモニタは、それぞれシステム・リソース・メトリック値及びトランザクション性能メトリック値を時間周期ベースで周期的に収集し、これらのモニタはまた、システム・リソース又はトランザクション性能メトリックの閾値が越えられるときを検出し、

ある時間周期の外において、トランザクション性能メトリック閾値が越えられるたびに、システム・リソース・メトリック値を追加収集するように、前記トランザクション性能モニタから前記システム・リソース・モニタを起動するステップと、

各々の周期的に収集されたシステム・リソース・メトリック値の平均値を計算するステップと、

各々の追加収集されたシステム・リソース・メトリック値の平均値を計算するステップと、

システム・リソース・メトリック値の追加収集によって得られた平均値と、時間周期ベースで行われたシステム・リソース・メトリック値の収集によって得られた平均値との間に、重要な差が存在する場合のシステム・リソース・メトリックを識別するステップと、

前記識別されたシステム・リソース・メトリックに関して、前記2つの計算された平均値の間に新しい閾値を設定するステップとを含む方法。

【請求項 2】

前記起動するステップは、前記トランザクション性能モニタ自体が、トランザクション性能メトリック閾値が越えられるたびに、追加収集を示す識別子を含んだトランザクション性能メトリック値の追加収集を実行するステップをさらに含み、この識別子は前記トランザクション性能モニタに与えられてメトリック値の前記追加収集に含まれる、請求項 1 に記載の方法。

【請求項 3】

前記新しい閾値をシステムの監視のために用いるステップと、
前記システムの監視が満足されるまで、前記方法に係る全ステップを必要なだけ何度も繰り返すステップと

10

をさらに含む、請求項 1 又は請求項 2 のいずれかに記載の方法。

【請求項 4】

メトリックの閾値を用いてシステム・リソース及びトランザクション性能をローカルに監視することができる、トランザクション・ベースのサーバを管理サーバから監視するためのサービスを提供する方法であって、前記システムによって実行される、

管理されるサーバに前記システム・リソース及びトランザクション性能を監視するツールをローカルにインストールするステップと、

管理されるサーバ上で、前記システム・リソース・データ及びトランザクション性能データを時間周期ベースでローカルに収集するステップと、

管理されるサーバ上で、ある時間周期外において、トランザクション性能メトリック閾値が越えられるたびに、システム・リソース・メトリック値を追加収集するように前記システム・リソースをローカルに監視するモニタを前記トランザクション性能をローカルに監視するモニタからローカルに起動するステップと、

20

前記管理されるサーバから前記管理サーバにデータを送信するステップと、

前記管理サーバ上で、各々の周期的に収集されたシステム・リソース・メトリック値の平均値を計算するステップと、

前記管理サーバ上で、各々の追加収集されたシステム・リソース・メトリック値の平均値を計算するステップと、

前記管理サーバ上で、システム・リソース・メトリック値の追加収集によって得られた平均値と、時間周期ベースで行われたシステム・リソース・メトリック値の収集によって得られた平均値との間に、重要な差が存在する場合のシステム・リソース・メトリックを識別するステップと、

30

前記識別されたシステム・リソース・メトリックに関して、前記 2 つの計算された平均値の間に新しい閾値を設定するステップと、

前記システム・リソース・メトリックの新しい閾値を前記管理サーバから前記管理されるサーバに送るステップと

を含む、方法。

【請求項 5】

前記起動するステップは、前記トランザクション性能モニタ自体が、トランザクション性能メトリック閾値が越えられるたびに、追加収集を示す識別子を含んだトランザクション性能メトリック値の追加収集を実行するステップをさらに含み、この識別子は前記トランザクション性能モニタに与えられてメトリック値のシステム・モニタ追加収集に含まれる、請求項 4 に記載の方法。

40

【請求項 6】

前記管理されるサーバ上で、前記計算された閾値をシステム監視のためにローカルに使用するステップと、

前記システム・リソースの監視が満足されるまで、前記収集するステップから始まる前記方法の全ステップを必要なだけ何度も繰り返すステップと、

前記管理されるサーバ上で前記トランザクション性能データ監視ツールをローカルにインストールするステップと

50

をさらに含む、請求項 4 又は請求項 5 のいずれかに記載の方法。

【請求項 7】

コンピュータに、請求項 1 又は請求項 3 のいずれかに記載のステップを実行させるためのコンピュータ・プログラム。

【請求項 8】

請求項 7 に記載のコンピュータ・プログラムを実行するシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般に、システムの性能に関し、より具体的には、本発明はトランザクションの性能に影響を与える問題から症状のシグニチャを自動的に引き出すための方法及びシステムに関する。

10

【背景技術】

【0002】

ITリソースを使って事業を営むときに最も重要なことは、システムが、それを効率化するサービスの段階で、ビジネス・アプリケーションをサポートするのに必要な機能を提供することである。このためには、トランザクションがタイムリーに実行される必要がある。通常、トランザクションの応答時間の問題が発生するときには、まずエンド・ユーザが問題を経験し、ITスタッフは、既にビジネスに影響が及んだ後に、その問題の警告を受け取る。その後で、故障修理のプロセスが始まり、どこに問題があるのかを発見するためにITリソースが調べられる。

20

【0003】

洗練されたトランザクション監視ツールが存在し、トランザクションの応答時間が所定の閾値を超えるとすぐに、リアルタイムでITスタッフに警告する。これは通常の（以前の）状況に対しては大きな改良であるが、このようなツールはまだ普及しておらず、またこのようなツールを使うことには、ある影響が予想される。例えば、トランザクションの継続時間を計るためのトランザクション監視ツールを呼び出すためのアプリケーションを搭載する必要があり、この搭載がトランザクション自体のパス長に加わり、それ故にトランザクションの実行を遅くする。トランザクション自体の実行に直接影響を及ぼす方法を使わずに、劣化した応答時間を検出できることが好ましい。

30

【0004】

商用ビジネス・アプリケーションでは、トランザクションが完了できない場合、又は極度に遅くなる場合には、ビジネス・アプリケーション（例えば、データベース、ファイル・システム、サーバなど）をサポートする1つ又は複数のITリソースに必ず問題が存在する。トランザクションの応答時間の劣化を生じるITリソース内の問題は限定的で反復可能である。ITリソースの状態を観察することによってサービス劣化を生じる条件を認識することができれば、トランザクション監視プログラムの広範囲の配備を必要とせずに、またそのような監視がもたらすパス長のオーバーヘッドなしに、問題状況に対して速やかに対処することが可能になる。

【0005】

40

システム管理の監視スペースには、2つの異なる監視のカテゴリ、即ち、リソースの監視及びトランザクションの監視がある。トランザクションの実行性能を監視するのに用いられるITリソースの状態及び挙動を監視するために異なる技術が適用され、2つの規則が、異なるツール及び実施法で実行される。そのため、管理者が、システム内で実行中のトランザクションの性能についての情報、及び種々のITリソースがいかに動いているかについての情報を得ることができるにもかかわらず、その2つを合わせて、両セットのデータを強化して観測された問題の根本原因を特定することは極めて困難である。それゆえ、リソースの監視のみによってトランザクション性能の低下を生じる問題を特定することは不可能である。

【0006】

50

トランザクションの応答時間を改善する既存の方法を探すと、次のアドレス、
[Http://www.microsoft.com/technet/prodtech/hnol/sscomm/reskit/rkcapmit.mspx](http://www.microsoft.com/technet/prodtech/hnol/sscomm/reskit/rkcapmit.mspx)
におけるマイクロソフト（マイクロソフトは、特定の国におけるマイクロソフト・コーポレーションの商標である）のサイトのウェブ・ページ上で見いだすことができる。これらのページは、インターネット・トランザクションのキャパシティ・モデルを説明している。この方法は、システム監視パラメータを測定してリソース消費の静的予想曲線を作ることによってトランザクション性能の改善を試みる。次に、トランザクションの応答時間及び処理量がシステム・リソース消費を予想するモデルに対して入力データとして与えられる。このキャパシティ計画の方法は、トランザクションをサポートするシステムを計量することにおいては役立つとしても、リアルタイムでトランザクション性能を監視することには役立たない。

10

【発明の開示】

【発明が解決しようとする課題】

【0007】

従って、オペレータがリアルタイムでシステム・リソースに反応することを自動的にガイドする、トランザクション監視の方法及びシステムを提供することが本発明の目的である。

トランザクションの速度を低下させる割込みコードを作らないトランザクション監視の方法及びシステムを取得することが、本発明の第二の目的である。

20

【課題を解決するための手段】

【0008】

これらの目的は、請求項1により、メトリックの閾値を用いてシステム・リソース及びトランザクション性能データを監視する、トランザクション・ベースのシステムを監視する方法を用いて達成され、この方法は、

システム・リソース・データ及びトランザクション性能データを周期ベースで収集し、トランザクション性能データのメトリックが越えられるたびに、システム・リソースの収集を起動し、トランザクション性能データとシステム・リソース・データを関連付ける識別子をストアするステップと、

収集されたシステム・リソース及びトランザクション性能データを組み合わせるステップと、

30

組み合わせられたデータのシステム・リソース又はトランザクション性能メトリックの各々の平均値を計算するステップと、

トリガ機能により又はトリガ機能によらずに得られた平均値の間に重要な差があるメトリックを識別するステップと、

識別されたメトリックの平均値によって、監視に用いるべきシステム・リソース・メトリックの新しい閾値を計算するステップと、
を含む。

【0009】

これらの目的はまた、請求項2により、
システムの監視のために計算された閾値を使用するステップと、
システムの監視が満足されるまで、その方法に係る全ステップを必要なだけ何度も繰り返すステップと、
をさらに含む請求項1の方法を用いて達成される。

40

【0010】

これらの目的はまた、請求項3により、メトリック閾値を用いてそれらのシステム・リソース及びトランザクション性能をローカルに監視することができる、トランザクション・ベースのサーバを管理サーバから監視するサービスを提供する方法であって、前述のシステムによって実行される、

管理されるサーバにシステム・リソース・データ及びトランザクション性能データの監

50

視ツールをローカルにインストールするステップと、

管理されるサーバ上で、システム・リソース・データ及びトランザクション性能データを時間周期ベースでローカルに収集し、トランザクション性能データのメトリックが越えられるたびに、システム・リソースの収集を起動し、トランザクション性能データとシステム・リソース・データを関連付ける識別子をストアするステップと、

管理されるサーバから管理サーバにデータを送信するステップと、

収集されたシステム・リソース及びトランザクション性能データを管理サーバ上で組み合わせるステップと、

組み合わせられたデータのシステム・リソース又はトランザクション性能メトリックの各々の平均値を管理サーバ上で計算するステップと、

トリガ機能により又はトリガ機能によらずに得られた平均値の間に大きな差があるメトリックを管理サーバ上で識別するステップと、

識別されたメトリックの平均値によって、監視に用いるべきシステム・リソース・メトリックの新しい閾値を管理サーバ上で計算するステップと、

システム・リソース・メトリックの新しい閾値を前記管理サーバから管理されるサーバに送信するステップと、

を含む、方法によって達成される。

【 0 0 1 1 】

これらの目的は、請求項 4 により、

システム監視のために計算された閾値を管理されるサーバ上でローカルに使用するステップと、

システム・リソースの監視が満足されるまで、収集するステップから始まる方法の全ステップを必要なだけ何度も繰り返すステップと、

管理されるサーバ上でトランザクション性能データ監視ツールをローカルにアンインストールするステップと、

をさらに含む請求項 3 の方法を用いて達成することができる。

【 0 0 1 2 】

これらの目的は、請求項 5 により、コンピュータ上で実行されるとき、請求項 1 又は 2 のいずれかによる方法のステップを実行するためのプログラミング・コード命令を含むコンピュータ・プログラムを用いて達成される。

これらの目的は、請求項 6 により、請求項 1 又は 2 のいずれかによる方法を実行するために適合させた手段を含むシステムを用いて達成される。

【 0 0 1 3 】

本発明は、トランザクション及びリソース監視の両方の機能を強化し、リソースの挙動をトランザクション性能に関連づけることに関連するコンテキストにおいて両方のソースからの情報を組み合わせる。特に、本発明は、動作が低下するときにリソースが何をしているのかを明らかにして、性能問題の根本原因が存在する領域の非常に迅速な特定を可能にするために、低下したトランザクション性能が観測されたまさにその瞬間にリソースの挙動のスナップショットを取得する。具体的には、本発明は、トランザクション監視問題が IT リソースの挙動によって表されることを可能にして、トランザクション・モニタが配備されていないシステムにおいてさえも低下したトランザクション性能を生じる条件の検出を可能にする。

【 発明を実施するための最良の形態 】

【 0 0 1 4 】

図 1 は、好ましい実施形態による本発明の方法の環境を示す。メイン・システム (1 0 0) は、ネットワーク (1 1 5) を通じて、ローカル・エリア・ネットワーク (1 2 5 、 1 3 5 、 1 4 5) を形成することができるサーバ (1 5 5 、 1 2 5 、 1 3 5 、 1 4 5) に接続されている。メイン・システムは、それ自体のシステム・リソース及びサーバの中心点としてのトランザクション性能を監視する。好ましい実施形態においては、メイン・システムにインストールされたソフトウェア・ツール IBM Tivoli Monitor

10

20

30

40

50

r (ITM) (120) 及び IBM Tivoli Transaction Monitor (130) が、それぞれ Information Technology (IT) システム・リソース及びトランザクション性能を監視する。システム・リソースを監視するための任意の他のツール (RM 又はシステム・リソース・モニタ又はリソース・モニタとも呼ばれる) 又はトランザクション性能を測定するための任意の他のツールは、本発明を実施するために使用できることに留意されたい。メイン・システム上で動作するシグニチャ・ビルダ (110) ソフトウェア層も、本発明の好ましい実施形態の新しい機能を含む。マニュアル監視操作のためにオペレータ・コンソールがメイン・システムに接続される。

【0015】

図2は、好ましい実施形態の方法の一態様である、TP 及び RM データベースの組み合わせを示す。図2は、組み合わせられたトランザクション性能モニタ (200) 及びリソース・モニタ (220) を示す。トランザクション性能モニタ (200) は、図1の IBM Tivoli Monitor (ITM) (130) とすることが可能であるが、今日 TP 製品に使われている多くの方法の1つによってトランザクション応答時間を収集する。IT リソース・モニタは、図1の IBM Tivoli Monitor (120) とすることが可能であるが、今日リソース監視製品に使われている多くの方法の1つを用いて IT リソースの利用可能性及び性能に関するメトリックを収集する。この2つのエンティティは、独立に、それらのデータを共有データベース (230) に書き込む。データは、周期的に書き込まれ、この実施例のためには60秒の監視サイクルを仮定する。

【0016】

図3は、TP 及び RM データベースが組み合わせられるときにモニタリング・データベースにストアされる測定結果を示す。図3には、3つの異常なトランザクション性能が観察されているが、IT リソース・モニタが60秒のサイクルでデータを収集するので、トランザクション問題が観察されるのと同時にリソース・メトリックが収集されるとしても全くの偶然である。2つのエンティティによって収集される情報を組み合わせることは、以下の実施例のようなレポートを可能にすることができる。この例は意図的に過度に簡略化したものであり、非常に少ないメトリックが示されている。示されているレポートはまた、トランザクション性能及び IT リソース監視データの収集の時間同期化を想定しているので、非現実的でもある。実際、今日このことは如何なるシステムによっても達成されないが、本発明の好ましい実施形態の方法によって加えられる。しかし、本実施例においては明確にするために、同期化が想定されている。

【0017】

図4は、好ましい実施形態の方法の相関関係子によるトリガ機能が用いられない場合の、モニタリング・データベースの測定結果における2つの不完全な測定値を示す。トランザクション性能問題がデータの収集時には発生しないため、このレポートには如何なる問題の痕跡もない。性能低下が観察されたときにトランザクション性能モニタは記録を書き込むことはできるが、対応するリソース・データ・メトリックが無ければ何が問題を生じたかの指示はない。図4の測定結果レポートは、この状況を示している。

ここでは、2つの問題が観察されたことはわかるが、リソース・メトリックがないため、何がトランザクション性能低下の原因なのかについて如何なる結論も導き出すことができない。

【0018】

図5は、好ましい実施形態の方法による、TP から RM への相関関係子によるトリガ機能を示す。好ましい実施形態の解決法は、問題が観察された瞬間の状況の完全なスナップショットを取得するためのトランザクション性能及び IT リソース監視エンティティの協働を付け加えることである。

【0019】

図5においては、2つの監視エンティティは、独立にそれらの専門のドメインを監視し、独立にデータを共有データベースに書き込む。トランザクション性能モニタは問題を観

10

20

30

40

50

察する（応答時間が閾値を超える）と、ITリソース・モニタを起動して、通常の監視サイクルの外でデータを収集する。さらに、相関関係子をリソース・モニタに送って、トランザクション及びリソース・データが相互に結合できることを確実にして、時間同期化を確実にする。それゆえ、好ましい実施形態の方法により、以前のレポートには欠けていたリソース・メトリックを捕捉することができる。

【0020】

好ましい実施形態の方法においては、問題が観察された瞬間の状況の完全なスナップショットを取得するためのトランザクション性能及びITリソース監視エンティティの協働が付け加えられる。2つの監視エンティティは、独立に、それらの専門ドメインを監視し、独立にデータをデータベース（このデータベースは物理的共有データベース、又はトランザクション・データのための1つ及びリソース・データのための1つの2つの独立したデータベースとすることができ、そのデータは後のステップで組み合わせられることに留意されたい）に書き込む。トランザクション性能モニタが問題（応答時間が閾値を超える）を観察すると、ITリソース・モニタを起動させて通常の監視サイクルの外でデータを収集する。さらに、相関関係子をリソース・モニタに送って、トランザクション及びリソース・データが相互に結合できることを確実にして、時間同期化を確実にする。

10

【0021】

このトリガ機能が機能するためには、リソース・モニタはインターフェイスを公開して、外部プロセスがデータの収集を起動することができるようにしなければならない。このインターフェイスは、別々のデータ・レコードを組み合わせることを可能にする相関関係子を取得するために、入力データ内にパラメータを求めることになる。この相関関係子は、トリガの条件を表す独特の識別子であり、トランザクション・モニタによって生成される。最も簡単な場合、相関関係子は増分カウンタとすることができる。

20

【0022】

完全な測定結果のレポートは、図6に示されるようなものになり得る。図6は、好ましい実施形態の方法により、TP及びRMデータベースが組み合わせられ、かつ相関関係子によるトリガ機能が用いられるときにモニタリング・データベース（230）にストアされる測定結果を示す。

【0023】

ここでは、トランザクション応答時間が許容レベルを超えて増加するとき、使用可能なメモリの量が、トランザクションが許容範囲内で動作しているときの正常値から、急激に減少することがわかる。さらに、CPU使用率が正常であって、問題の原因はCPU使用率ではなく、恐らく使用可能メモリの低下であることがわかる。適切な瞬間に、全ての関連データを集め得ること、及び異なるソースからのデータを関連付け得ることは、性能問題の根本原因を特定するために重要であり、2つの監視コンポーネントの協働によって可能となる。

30

【0024】

次の方法が、どのメトリック値が応答時間に悪影響を与える問題の症状であるかを自動的に決定するために用いられる。

各々の収集時間間隔に対してレコードが生成される。各レコードは、以下の情報、タイムスタンプと、

40

各々がメトリックのタイプ及びその時間間隔内で観察された値を示す、重要値の対のリストと、

相関関係子（トランザクション性能モニタによって収集が起動されないかぎりヌル値となる）と

を含む。

【0025】

非ヌル値の相関関係子を有するレコードに含まれる各メトリックの平均値が計算される。同様に、ヌル値の相関関係子を有するレコードに含まれる各メトリックの平均値が計算される。それら2つが比較される。例外条件及び正常条件において観察されたメトリック

50

の値の間に差がほとんどなければ、その特定のメトリックは、トランザクション応答時間に悪影響を与える問題の症状ではないことは明らかである。その2つの条件におけるメトリック値に顕著な差があれば、そのメトリックは、応答を遅らせている問題の症状として使うことができる可能性が高い。

【0026】

上述の方法は、応答時間の劣化に直面しているときのシステムの状態を、リソースのメトリック及び値を用いて示す「問題のシグニチャ」を作成するために用いることができる。この方法は、システムにトランザクション・モニタがインストールされておらずアクティブでない場合でも、ITリソース・モニタが応答時間の問題を引き起こしている可能性のある状況を検出できるため、極めて有用である。さらに、トランザクションの監視は、命令をトランザクション・パス長に加えるので、トランザクション応答時間を監視することは、トランザクションを非明示的に遅くする（良好に実施されているモニタではわずかな程度であるが）が、トランザクションのパスを配備することなしにトランザクション応答時間の問題を検出することは、如何なるパス長をも加えることはない。

【0027】

問題のシグニチャは、正常状態と例外状態の間で値の顕著な差を示すメトリック、及びそれらが例外状態において示す値を含むことになる。リソース・モニタが、全てのメトリックが例外状態のレベルに合致する値を示していることを検出する場合には、システムは恐らくトランザクション応答時間の劣化に直面していると報告することになる。各メトリックに関連する値は、以前に計算された平均値とはならない。その値が使われる場合には、定義により、個々のメトリックに関する「問題状態」の半分しか認識されないことになり、全体の問題シグニチャではさらに少なく認識されることになる。より優れた方法は、正常作動状態の平均値と例外状態の平均値の中間の値、又は2つの平均値の間の連続体上の他のある値を使うことである。その連続体上の正確な箇所は設定可能であるが、好ましい実施形態においては、例外状態の平均値からの標準偏差値に基づくものとする。

【0028】

本明細書でもまもなく説明される図中に用いられる実施例は、少数の性能メトリックが収集される単一のシステム上で実行される単一のトランザクションを示す。これは、極めて単純な場合である。好ましい実施形態の方法はまた、多くのサブ・トランザクションから成る複数のシステムに及ぶビジネス・トランザクションに適用できる。

【0029】

ウェブ・ユーザは、例えば、ウェブサーバ上、アプリケーション・サーバ上、及びデータベース上にサブ・トランザクションを作成する操作を呼び出すことができる。トランザクション・モニタは相関関係子を使用してサブ・トランザクションをつなぎ合わせるできるので、同じ相関関係子は、それぞれ個々のシステム上のリソース・モニタ・データをサブ・トランザクションに、従ってまた全体のビジネス・トランザクションに関連させるために用いることができる。それゆえ、好ましい実施形態の方法は、多くのシステムにわたるビジネス・トランザクションをサポートする全てのリソースのリソース挙動のスナップショットを取得して複雑な状況における問題の原因を明らかにすることができる。

【0030】

ITリソース監視の例外的なトリガ機能に対する一つの拡張機能は、各々のコンポーネントからのデータの正確な連結を可能にするために相関関係子を常に送ることができるように、常にリソース・データ収集を起動するためにTPトリガ機構を用いることである。これにはまた、トランザクション・モニタがトランザクションの開始を観察するとすぐに、リソース・モニタがリソース・メトリックを収集するように命令することができるという利点がある。これは、メトリック値が、トランザクションが実行されている時点での状況を正確に反映するという意味を意味する。以前に説明された方法では、メトリックは、トランザクション・モニタが遅い応答時間に気づいた後に（即ち、トランザクションが完了した後に）収集される。殆どの場合、リソース・メトリックの収集における少しの遅れはデータの有効性に影響を与えないが、非常に動的な環境においては、状況が急速に変化

10

20

30

40

50

することがあり、最小の遅れであってもパターン識別を困難にする可能性がある。

【 0 0 3 1 】

図7は、好ましい実施形態による、それから問題シグニチャが抽出される、モニタリング・データベース(230)にストアされた測定結果を示す。次の実施例は、例外及び非例外状態の間で値があまり変わらないメトリックがいかに容易に識別できるかを示している。この場合には、メトリックA、C及びDが問題シグニチャを構成し、一方メトリックB及びEは、この問題に関して有用な情報を何も提供しない。

【 0 0 3 2 】

図8は、好ましい実施形態による方法の一般的なフロー・チャートである。サーバを監視するためのITリソース・モニタ(220)及びシグニチャ・ビルダ(110)がすでに動作しているメイン・システムに、TPコード(200)がインストールされる(800)。次に、TP及びRMデータが監視されるサーバから収集される(810)。収集されたデータがモニタリング・データベース内で組み合わせられる(820)。TP及びRMレコードの両方に含まれる相関関係子が、関連する情報を関連付けるために用いられる。各々のメトリックに対して平均値が計算される(830)。重要なメトリックが識別される。「正常」及び「例外」レコードにおいて一貫して異なるメトリックは、問題シグニチャに寄与する故に識別される。各々のメトリックに対して閾値が計算される(850)。「最適」値は、任意にアルゴリズムを使って、又は代替としてオペレータ・コンソールを通じた顧客の選択を可能にすることにより、計算される。好ましい実施形態により、問題シグニチャをカプセル化するリソース・モデルが形成される(850)。本開示で説明されるリソース・モデルの概念は、IBM Tivoli Monitor(ITM)のようなシステム・リソース・モニタにすでに適用されている概念である。リソース・モデルは、特定の状況を識別する条件を記述する機械可読パッケージである。ITMの場合には、単純なRMは、システムが監視されている状況に適合する状態にあるかどうかを判断するのに必要なデータ(メトリック)を収集するためにITM内で動作する実行可能なスクリプトと、メトリック値を比較するための多数の閾値と、比較の結果を組み合わせることで監視されている状況を示す組み合わせを識別するブル・ロジックとを含む。ITMのRMはもっと複雑なロジックを実行できるが、本発明の方法の目的のためには、リソース・モデルは、システム・リソース・モニタによって消費され、特定のシステム状況を識別することを可能にする機械可読パッケージを表す。

【 0 0 3 3 】

リソース・モデルは、次に、管理されるサーバに配備される(870)。問題を識別するのにより長いトレーニング期間が必要であれば、データが再び収集され、同じループが実行される。ユーザがもはやこのサービスを必要としない場合には、この方法は終了する。トレーニングはある一定期間実施され、その後(テスト880にイエスと答える)、TPコードが削除され(885)、生成されたシグニチャがリソース監視ツールによって用いられる。あるいは(テスト880にノーと答える)、トレーニングを続行することができて、シグニチャは継続して最新のデータで更新される。この後者の方法は、システム上でTPエージェントをアクティブのままにしておける場合に適切である。

【 0 0 3 4 】

図9は、トレーニング期間中に、好ましい実施形態の方法を実行するためのシステム環境を示す。この図9は、独立したデータベース(940, 930, 920)上のデータを収集し、そのデータをそれぞれの管理サーバ・コンポーネントに送る、TP及びRMエージェントを示している。この構成においては、両エージェントは、トレーニング期間が終了するまで、全ての管理されるサーバ上で動作する。一旦レコードがデータベース内で持続すると、新しいロジックがモニタリング・データベース(230)内でレコードを組み合わせることでシグニチャを形成する。一旦トレーニング期間が終了すると、シグニチャはRMエージェントに配備されることになる。

【 0 0 3 5 】

図10は、プロダクション・モードで好ましい実施形態の方法を実行するためのシステ

10

20

30

40

50

ム環境を示す。この図10は、1つの管理されるサーバ上のみで動作するTPエージェントを示している。この構成は、全ての管理サーバが、同じトランザクションを処理し、全ての点で非常に類似している（それらはHTTPサーバのクラスタを形成する）場合に適切である。「トレーニング」は、単一サーバに限定されたトレーニングのオーバーヘッドにより続行することができる。シグニチャが、1つだけのサーバ上で収集された情報から形成されていても、サーバが非常に類似しているので生成されたシグニチャは全部に適用可能である。

【0036】

本発明は、サービスの提供として実施することができる。リソース監視ツールの使用は非常に普及しており、ほとんどの企業顧客は、伝統的なリソース健全状態/利用可能性の監視を実施するために1つのツールを使っていることになる。トランザクション監視ツールの使用は、トランザクションの性能がエンド・ユーザに直接影響を与え、運営されている事業に直接影響を与える可能性があるにも関わらず、遥かに少なくしか普及していない。サービスの提供は、顧客が購入すべき追加のソフトウェアを必要とせず、劣化した応答時間の状況を検出するリソース監視シグニチャを生成することとすることができる。これらのシグニチャは、顧客の環境におけるトレーニング期間中に収集されたデータから生成されることになるように、顧客のシステムに合わせて調整される。

【0037】

サービス・チームは、トランザクション監視サーバを顧客のシステムにインストールするか、又はサービス・チーム所有のシステムにプリロードしたものを持って行き、TPエージェントを顧客の管理されるサーバにインストールしてデータの収集を開始する。十分なデータが収集された後、サービス・チームは、分析コードを実行させてシグニチャを作成し、次いで、このシグニチャが顧客に配信されてリソース・モニター・エージェントに配備されることになる。

【0038】

図11は、サービス提供として実施されるときに好ましい実施形態の方法の一般的なフロー・チャートである。初めにTPエージェントが監視されるべきサーバにインストールされる(1100)。次に、監視されるサーバからTP及びRMデータが収集される(1110)。その後、データはエージェントによって管理サーバに送られる(1120)。次のステップ(1130)は管理サーバ内で実施される。TP及びRMエージェントからのデータはモニタリング・データベース内で組み合わされる(1115)。TP及びRMレコードの両方に含まれる相関関係子が関連情報を関連付けるために用いられる。各々のメトリックに対して平均値が計算される(1125)。重要なメトリックが識別される(1135)。「正常」及び「例外」レコードにおいて一貫して異なるメトリックは、問題のシグニチャに寄与するので識別される。各々のメトリックに対して閾値が計算される(1145)。「最適」値が、任意にアルゴリズムを使って、又は代替として顧客の選択を可能にすることによって計算される。問題のシグニチャをカプセル化するリソース・モデルが形成される(1155)。次いで、リソース・モデルが顧客に配備される(1130)。ウェブサービスのユーザが問題を識別するためのトレーニング期間の続行を希望する場合には、データが再び収集され、同じループが実施される。ユーザがもはやこのサービスを必要としなくなる場合は、この方法は終了する。トレーニングはある一定期間実施することができ、次にTPエージェントが削除され(1140)、本監視方法が終了し(1150)、生成されたシグニチャはリソース監視エージェントによって用いられる。あるいは、トレーニングは続行することができて、シグニチャは継続して最新のデータで更新される。この後者の方法は、管理されるサーバのサブセット上でTPエージェントをアクティブのままにしておける場合に適切である。

【図面の簡単な説明】

【0039】

【図1】好ましい実施形態による方法を実施するためのコンピュータ・システムの概観を示す。

10

20

30

40

50

【図2】好ましい実施形態による方法の一態様であるTP及びRMデータベースの組み合わせを示す。

【図3】TP及びRMデータベースが組み合わされるときにモニタリング・データベースにストアされる測定結果を示す。

【図4】好ましい実施形態による方法の相関関係子によるトリガ機能が用いられない場合の、モニタリング・データベースの測定結果における2つの不完全な測定値を示す。

【図5】好ましい実施形態の方法によるTPからRMへの相関関係子の機能によるトリガ機能を示す。

【図6】好ましい実施形態の方法による、TP及びRMデータベースが組み合わされるとき、かつ、相関関係子によるトリガ機能が用いられるときに、モニタリング・データベースにストアされる測定結果を示す。

10

【図7】好ましい実施形態による、それから問題のシグニチャが抽出されるモニタリング・データベースにストアされた測定結果を示す。

【図8】好ましい実施形態による方法の一般的なフロー・チャートである。

【図9】トレーニング期間中に、好ましい実施形態の方法を実施するためのシステム環境を示す。

【図10】実働モードにおいて、好ましい実施形態の方法を実施するためのシステム環境を示す。

【図11】サービスの提供として実施されるときに好ましい実施形態の方法の一般的なフロー・チャートである。

20

【符号の説明】

【0040】

100：メイン・システム

110：シグニチャ・ビルダ

115：ネットワーク

120：IBM Tivoli モニタ (ITM)

125、135、145、155：サーバ

130：IBM Tivoli トランザクション性能モニタ (TPM)

200：トランザクション性能モニタ

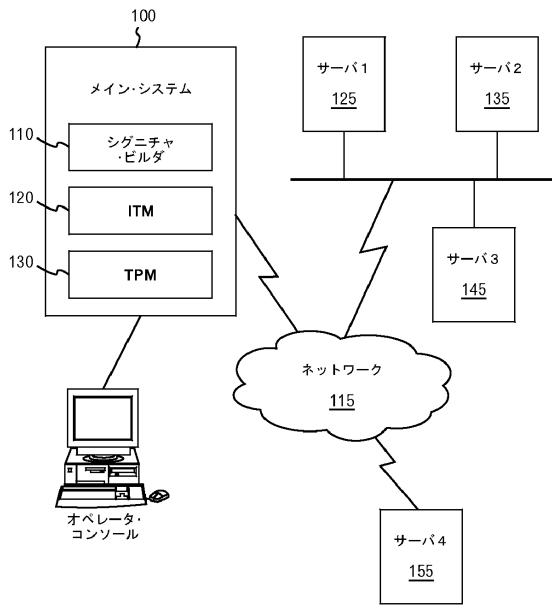
220：ITリソース・モニタ

30

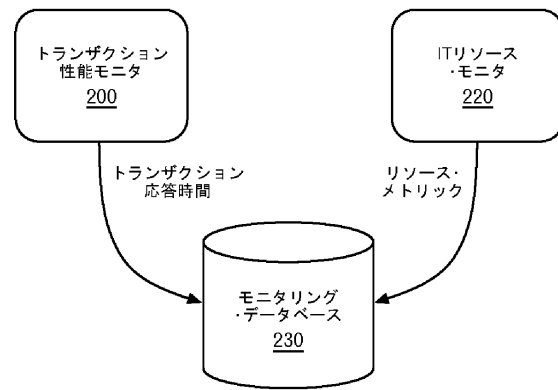
230：モニタリング・データベース (共有データベース)

920、930、940：データベース

【 図 1 】



【 図 2 】



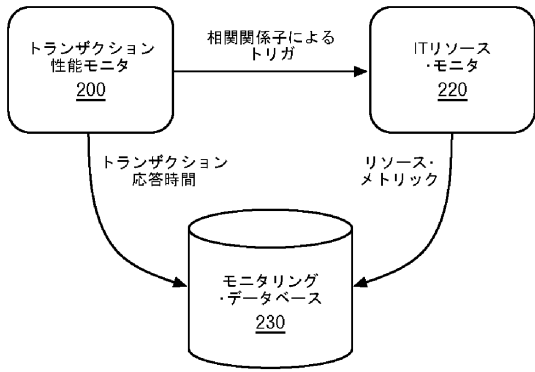
【 図 3 】

タイムスタンプ	トランザクション 応答時間	CPU %	利用可能なメモリ (K)
12:00:00	0.25	48	304473
12:01:00	0.3	62	267594
12:02:00	0.27	55	287309
12:03:00	0.26	58	298305
12:04:00	0.31	65	276811
12:05:00	0.24	74	349277
12:06:00	0.26	80	298347
12:07:00	0.25	76	295038
12:08:00	0.27	63	286602
12:09:00	0.24	59	300219
12:10:00	0.27	61	294117
12:11:00	0.3	75	280499
12:12:00	0.25	77	282991
12:13:00	0.23	54	329832
12:14:00	0.25	48	315830
12:15:00	0.27	60	294381
12:16:00	0.24	78	319982
12:17:00	0.3	84	278455
12:18:00	0.29	83	288201
12:19:00	0.27	76	290006
12:20:00	0.26	69	303395

【 図 4 】

タイムスタンプ	トランザクション 応答時間	CPU %	利用可能なメモリ (K)
12:00:00	0.25	48	304473
12:01:00	0.3	62	267594
12:02:00	0.27	55	287309
12:03:00	0.26	58	298305
12:03:17	1.27		
12:04:00	0.31	65	276811
12:05:00	0.24	74	349277
12:06:00	0.26	80	298347
12:07:00	0.25	76	295038
12:08:00	0.27	63	286602
12:09:00	0.24	59	300219
12:10:00	0.27	61	294117
12:11:00	0.3	75	280499
12:12:00	0.25	77	282991
12:13:00	0.23	54	329832
12:13:35	1.19		
12:14:00	0.25	48	315830
12:15:00	0.27	60	294381
12:16:00	0.24	78	319982
12:17:00	0.3	84	278455
12:18:00	0.29	83	288201
12:19:00	0.27	76	290006
12:20:00	0.26	69	303395

【図5】



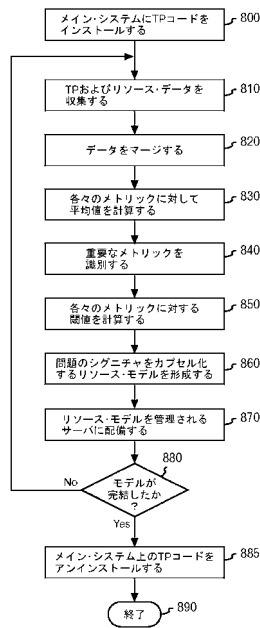
【図6】

タイムスタンプ	トランザクション 応答時間	CPU %	利用可能なメモリ (K)
12:00:00	0.25	48	304473
12:01:00	0.3	62	267594
12:02:00	0.27	55	287309
12:03:00	0.26	58	298305
12:03:17	1.27	62	12729
12:04:00	0.31	65	276811
12:05:00	0.24	74	349277
12:06:00	0.26	80	298347
12:07:00	0.25	76	295038
12:08:00	0.27	63	286602
12:09:00	0.24	59	300219
12:10:00	0.27	61	294117
12:11:00	0.3	75	280499
12:12:00	0.25	77	282991
12:13:00	0.23	54	329832
12:13:35	1.19	59	15924
12:14:00	0.25	48	315830
12:15:00	0.27	60	294381
12:16:00	0.24	78	319982
12:17:00	0.3	84	278455
12:18:00	0.29	83	288201
12:19:00	0.27	76	290006
12:20:00	0.26	69	303395

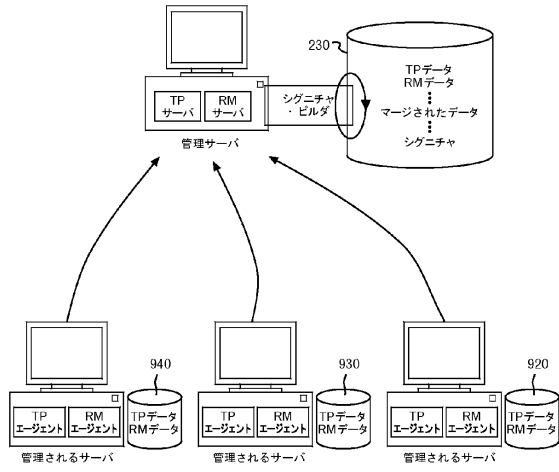
【図7】

レコード 番号	タイムスタンプ	相関 関係子	メトリック A	メトリック B	メトリック C	メトリック D	メトリック E
1	12:00:00		10.1	707	20	80	0.39
2	12:01:00		9.8	732	21	90	0.42
3	12:02:00	xyz123	43.7	722	6	604	0.40
4	12:03:00		9.7	705	22	85	0.41
5	12:04:00		9.9	724	23	85	0.39
6	12:05:00		10.2	716	20	89	0.37
7	12:06:00		10.0	719	20	84	0.44
8	12:07:00	xyz124	63.5	720	6	566	0.38
9	12:08:00		9.9	714	19	83	0.40
10	12:09:00		9.9	730	23	78	0.43
11	12:10:00		10.1	721	24	87	0.38
12	12:11:00		9.8	709	23	86	0.40
13	12:12:00	xyz125	50.1	702	5	597	0.39
14	12:13:00		9.9	720	21	84	0.37
相関関係子を有するレコードの平均			52.4	715	6	589	0.39
相関関係子を有しないレコードの平均			9.9	718	21	84	0.40

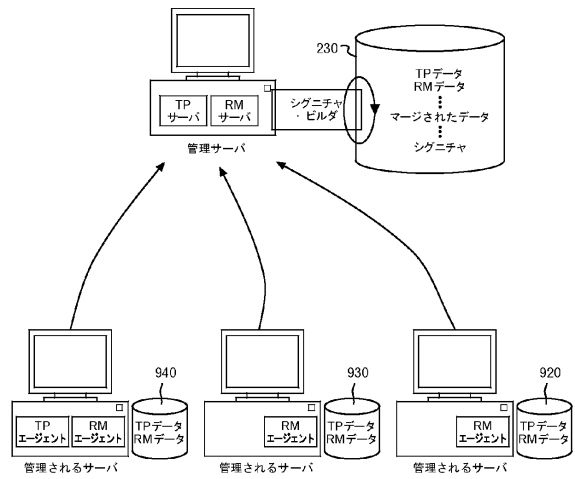
【図8】



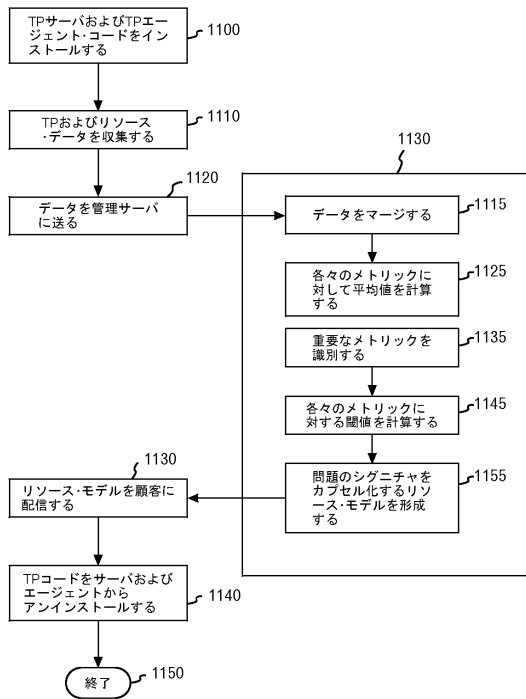
【図9】



【図10】



【図11】



フロントページの続き

(74)代理人 100086243

弁理士 坂口 博

(72)発明者 マクレラン、スコット

イタリア共和国 I - 0 0 1 2 6 ローマ ヴィーア・デイ・モンティ・ディ・サン・パオロ 5
5 / オー

(72)発明者 シャッカ、ヴィンチェンツォ

イタリア共和国 I - 0 0 1 4 4 ローマ ヴィーア・ジュゼッペ・ペレゴ 7 5

審査官 新井 寛

(56)参考文献 特開2003 - 263342 (JP, A)

米国特許出願公開第2002 / 0198985 (US, A1)

特開平09 - 244998 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 11/30-11/34