

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6025753号
(P6025753)

(45) 発行日 平成28年11月16日(2016.11.16)

(24) 登録日 平成28年10月21日(2016.10.21)

(51) Int. Cl.		F I			
G06F 11/07	(2006.01)	G06F	11/07	154	
G06F 11/34	(2006.01)	G06F	11/07	166	
		G06F	11/34	147	

請求項の数 21 (全 16 頁)

(21) 出願番号	特願2013-553058 (P2013-553058)	(73) 特許権者	390009531
(86) (22) 出願日	平成24年2月8日(2012.2.8)		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2014-507727 (P2014-507727A)		INTERNATIONAL BUSINESS MACHINES CORPORATION
(43) 公表日	平成26年3月27日(2014.3.27)		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/IB2012/050569		New Orchard Road, Armonk, New York 10504, United States of America
(87) 国際公開番号	W02012/110918	(74) 代理人	100108501
(87) 国際公開日	平成24年8月23日(2012.8.23)		弁理士 上野 剛史
審査請求日	平成26年9月5日(2014.9.5)		最終頁に続く
(31) 優先権主張番号	13/026, 351		
(32) 優先日	平成23年2月14日(2011.2.14)		
(33) 優先権主張国	米国 (US)		

(54) 【発明の名称】 パフォーマンス・メトリックを監視するためのコンピュータによって実施される方法、コンピュータ可読記憶媒体、およびシステム

(57) 【特許請求の範囲】

【請求項1】

パフォーマンス・メトリックを監視するためのコンピュータによって実施される方法であって、

現在のサンプリング期間のパフォーマンス・メトリックの値を判定するステップと、前記パフォーマンス・メトリックの前記値が前記パフォーマンス・メトリックの予想される挙動の第1のモデルから導出された閾値を超えると判定すると、前記パフォーマンス・メトリックの予想される挙動の第2のモデルであって、メトリック履歴が、スパイクが観察されていた期間の間にサンプリングされたデータ値に限定される、第2のモデルに従ってパフォーマンス・メトリックの前記値を評価するステップと、

前記パフォーマンス・メトリックの前記値が前記第2のモデルから導出された閾値を超えると判定すると、警告メッセージを生成するステップとを含む方法。

【請求項2】

前記パフォーマンス・メトリックの前記値が前記第1のモデルから導出された前記閾値を超えなかったと判定すると、前記パフォーマンス・メトリックの前記サンプリングされた値に基づいて前記第1のモデルを更新するステップをさらに含む、請求項1に記載の方法。

【請求項3】

前記パフォーマンス・メトリックの前記値が前記パフォーマンス・メトリックの予想さ

れる挙動の前記第 1 のモデルから導出された前記閾値を超えたと判定すると、前記パフォーマンス・メトリックの前記サンプリングされた値に基づいて前記第 2 のモデルを更新するステップをさらに含む、請求項 1 に記載の方法。

【請求項 4】

前記パフォーマンス・メトリックの前記値が前記パフォーマンス・メトリックの予想される挙動の前記第 1 のモデルから導出された前記閾値を超えたと判定すると、前記サンプリング期間のサンプリング頻度を上げるステップをさらに含む、請求項 1 に記載の方法。

【請求項 5】

前記パフォーマンス・メトリックが、共有リソースの使用量に対応する、請求項 1 に記載の方法。

10

【請求項 6】

前記パフォーマンス・メトリックが、プロセッサ使用率、ストレージ・リソース消費量、およびメモリ消費量のうちの 1 つに対応する、請求項 1 に記載の方法。

【請求項 7】

前記第 1 の閾値が、指定された第 1 のトレーニング期間にわたって前記第 1 のモデルをトレーニングすることによって導出され、前記第 2 のモデルが、前記パフォーマンス・メトリック値が前記第 1 の閾値を超えるとときに前記パフォーマンス・メトリックをサンプリングすることによってトレーニングされる、請求項 1 に記載の方法。

【請求項 8】

プロセッサで実行されると、パフォーマンス・メトリックを監視するための動作を実行するアプリケーションを記憶するコンピュータ可読記憶媒体であって、前記動作が、

20

現在のサンプリング期間のパフォーマンス・メトリックの値を判定するステップと、

前記パフォーマンス・メトリックの前記値が前記パフォーマンス・メトリックの予想される挙動の第 1 のモデルから導出された閾値を超えると判定すると、前記パフォーマンス・メトリックの予想される挙動の第 2 のモデルであって、メトリック履歴が、スパイクが観察されていた期間の間にサンプリングされたデータ値に限定される、第 2 のモデルに従ってパフォーマンス・メトリックの前記値を評価するステップと、

前記パフォーマンス・メトリックの前記値が前記第 2 のモデルから導出された閾値を超えると判定すると、警告メッセージを生成するステップとを含む、コンピュータ可読記憶媒体。

30

【請求項 9】

前記動作が、前記パフォーマンス・メトリックの前記値が前記第 1 のモデルから導出された前記閾値を超えなかったと判定すると、前記パフォーマンス・メトリックの前記サンプリングされた値に基づいて前記第 1 のモデルを更新するステップをさらに含む、請求項 8 に記載のコンピュータ可読記憶媒体。

【請求項 10】

前記動作が、前記パフォーマンス・メトリックの前記値が前記パフォーマンス・メトリックの予想される挙動の前記第 1 のモデルから導出された前記閾値を超えたと判定すると、前記パフォーマンス・メトリックの前記サンプリングされた値に基づいて前記第 2 のモデルを更新するステップをさらに含む、請求項 8 に記載のコンピュータ可読記憶媒体。

40

【請求項 11】

前記動作が、前記パフォーマンス・メトリックの前記値が前記パフォーマンス・メトリックの予想される挙動の前記第 1 のモデルから導出された前記閾値を超えたと判定すると、前記サンプリング期間のサンプリング頻度を上げるステップをさらに含む、請求項 8 に記載のコンピュータ可読記憶媒体。

【請求項 12】

前記パフォーマンス・メトリックが、共有リソースの使用量に対応する、請求項 8 に記載のコンピュータ可読記憶媒体。

【請求項 13】

前記パフォーマンス・メトリックが、プロセッサ使用率、ストレージ・リソース消費量

50

、およびメモリ消費量のうちの1つに対応する、請求項8に記載のコンピュータ可読記憶媒体。

【請求項14】

前記第1の閾値が、指定された第1のトレーニング期間にわたって前記第1のモデルをトレーニングすることによって導出され、前記第2のモデルが、前記パフォーマンス・メトリック値が前記第1の閾値を超えるときに前記パフォーマンス・メトリックをサンプリングすることによってトレーニングされる、請求項8に記載のコンピュータ可読記憶媒体。

【請求項15】

プロセッサと、
前記プロセッサで実行されると、パフォーマンス・メトリックを監視するための動作を実行するアプリケーション・プログラムを記憶するメモリと
を含むシステムであって、前記動作が、

現在のサンプリング期間のパフォーマンス・メトリックの値を判定するステップと、
前記パフォーマンス・メトリックの前記値が前記パフォーマンス・メトリックの予想される挙動の第1のモデルから導出された閾値を超えると判定すると、前記パフォーマンス・メトリックの予想される挙動の第2のモデルであって、メトリック履歴が、スパイクが観察されていた期間の間にサンプリングされたデータ値に限定される、第2のモデルに従ってパフォーマンス・メトリックの前記値を評価するステップと、

前記パフォーマンス・メトリックの前記値が前記第2のモデルから導出された閾値を超えると判定すると、警告メッセージを生成するステップと
を含む、システム。

【請求項16】

前記動作が、前記パフォーマンス・メトリックの前記値が前記第1のモデルから導出された前記閾値を超えなかったと判定すると、前記パフォーマンス・メトリックの前記サンプリングされた値に基づいて前記第1のモデルを更新するステップをさらに含む、請求項15に記載のシステム。

【請求項17】

前記動作が、前記パフォーマンス・メトリックの前記値が前記パフォーマンス・メトリックの予想される挙動の前記第1のモデルから導出された前記閾値を超えたと判定すると、前記パフォーマンス・メトリックの前記サンプリングされた値に基づいて前記第2のモデルを更新するステップをさらに含む、請求項15に記載のシステム。

【請求項18】

前記動作が、前記パフォーマンス・メトリックの前記値が前記パフォーマンス・メトリックの予想される挙動の前記第1のモデルから導出された前記閾値を超えたと判定すると、前記サンプリング期間のサンプリング頻度を上げるステップをさらに含む、請求項15に記載のシステム。

【請求項19】

前記パフォーマンス・メトリックが、共有リソースの使用量に対応する、請求項15に記載のシステム。

【請求項20】

前記パフォーマンス・メトリックが、プロセッサ使用率、ストレージ・リソース消費量、およびメモリ消費量のうちの1つに対応する、請求項15に記載のシステム。

【請求項21】

前記第1の閾値が、指定された第1のトレーニング期間にわたって前記第1のモデルをトレーニングすることによって導出され、前記第2のモデルが、前記パフォーマンス・メトリック値が前記第1の閾値を超えるときに前記パフォーマンス・メトリックをサンプリングすることによってトレーニングされる、請求項15に記載のシステム。

【発明の詳細な説明】

【技術分野】

10

20

30

40

50

【0001】

本発明の実施形態は、予測モデリング解析の一部として使用され得る技法を対象とする。より詳細には、本発明の実施形態は、マルチモデリング・パラダイムを使用してコンピューティング・システムのパフォーマンス・メトリックを評価するための方法およびシステムを提供する。

【背景技術】

【0002】

大規模なコンピューティング配備では、1つのよくあるレジリエンシー問題は、コンピューティング・システムがクラッシュしない「ソフト障害」と呼ばれるものを解決しているが、正常に動作するのを停止するか、事実上機能しないところまで速度を落とすだけである。予測解析は、コンピューティング・システムについてのサンプリングされたメトリックの現在のセットが、将来イベントが発生する可能性があることをいつ示すかを特定するために（例えば、ソフト障害がいつ発生する可能性があるかを予測するために）使用される技法である。予測解析ツールは、履歴データに依存して、予想されるシステム挙動のモデルを導出する。

10

【0003】

そのようなツールの重要な側面は、誤検知を回避する機能である。誤検知は、予測解析ツールが問題を検出し、ユーザに警告するが、挙動は実際には正常なシステム挙動であるときに生じる。誤検知は、予測解析ツールに対するユーザの信頼を大幅に低下させる可能性がある。大型コンピュータ・システムでは、その挙動が「スパイク性がある（spikey）」ものである、つまり、アクティビティ・レートが作業負荷および時間帯、曜日などに応じて劇的に変化し得る、多くのタスクまたはジョブが実行中である場合がある。予測解析ツールは、システム上で収集された履歴データを解析し、機械学習アルゴリズムを使用して、システム上の異常な挙動を特定する。例えば、定期的な周期処理（週毎、隔週、月毎など）は、アクティビティにおいて、予測解析ツールによって異常な挙動と誤って特定される可能性がある正常なスパイクを引き起こす可能性がある。スパイクは、平均消費レートを使用して設定された消費閾値を超える傾向があるので、「スパイク性がある」挙動を示すジョブまたはプロセスは、誤検知を引き起こす傾向がある。さらに、スパイクのタイミングは、1カ月の日数、週末の数、祝日の数などが変化するために、パターン認識アルゴリズムによって検出可能なパターンに従わないことがある。

20

30

【発明の概要】

【発明が解決しようとする課題】

【0004】

本発明の目的は、マルチモデリング・パラダイムを使用してコンピューティング・システムのパフォーマンス・メトリックを提供するための技法を提供することにある。

【課題を解決するための手段】

【0005】

本発明の一実施形態は、パフォーマンス・メトリックを監視するための方法を含む。この方法は、一般的に、現在のサンプリング期間のパフォーマンス・メトリックの値を判定するステップを含み得る。パフォーマンス・メトリックの値がパフォーマンス・メトリックの予想される挙動の第1のモデルから導出された閾値を超えると判定すると、パフォーマンス・メトリックの値は、パフォーマンス・メトリックの予想される挙動の第2のモデルに従って評価される。そして、パフォーマンス・メトリックの値が第2のモデルから導出された閾値を超えると判定すると、警告メッセージが生成される。

40

【0006】

本発明の別の実施形態は、プロセッサで実行されると、パフォーマンス・メトリックを監視するための動作を実行するアプリケーションを記憶するコンピュータ可読記憶媒体を含む。この動作自体は、一般的に、現在のサンプリング期間のパフォーマンス・メトリックの値を判定するステップを含み得る。パフォーマンス・メトリックの値がパフォーマンス・メトリックの予想される挙動の第1のモデルから導出された閾値を超えると判定する

50

と、パフォーマンス・メトリックの値は、パフォーマンス・メトリックの予想される挙動の第2のモデルに従って評価される。そして、パフォーマンス・メトリックの値が第2のモデルから導出された閾値を超えると判定すると、警告メッセージが生成される。

【0007】

本発明のさらに別の実施形態は、プロセッサと、プロセッサで実行されると、パフォーマンス・メトリックを監視するための動作を実行するアプリケーション・プログラムを記憶するメモリとを有するシステムを含む。この動作自体は、一般的に、現在のサンプリング期間のパフォーマンス・メトリックの値を判定するステップを含み得る。パフォーマンス・メトリックの値がパフォーマンス・メトリックの予想される挙動の第1のモデルから導出された閾値を超えると判定すると、パフォーマンス・メトリックの値は、パフォーマンス・メトリックの予想される挙動の第2のモデルに従って評価される。そして、パフォーマンス・メトリックの値が第2のモデルから導出された閾値を超えると判定すると、警告メッセージが生成される。

10

【0008】

上に記載された態様を達成し、詳細に理解することができるように、上記で簡潔に概要を述べた本発明の実施形態のより詳細な説明は、添付の図面を参照することによって得ることができる。

【0009】

ただし、本発明は他の同様に有効な実施形態を認め得るので、添付の図面はこの発明の典型的な実施形態のみを示し、したがって、この発明の範囲を限定するものとみなされるべきではないことに留意されたい。

20

【図面の簡単な説明】

【0010】

【図1】本発明の実施形態が実施され得る例示的なコンピューティング・インフラを示す図である。

【図2】本発明の一実施形態による、マルチモデリング・パラダイムを使用してパフォーマンス・メトリックを監視するために使用される例示的なコンピューティング・システムを示す図である。

【図3】本発明の一実施形態による、マルチモデリング・パラダイムを使用して、「スパイク性がある」コンピューティング・ジョブまたはプロセスを監視するための方法を示す図である。

30

【図4】本発明の一実施形態による、マルチモデリング・パラダイムを使用して、予測解析を実行するための方法を示す図である。

【図5】本発明の一実施形態による、マルチモデリング・パラダイムを使用するように構成された予測解析ツールによって監視された例示的なデータ・セットを示す図である。

【図6】本発明の一実施形態による、マルチモデリング・パラダイムを使用するように構成された予測解析ツールによって監視された例示的なデータ・セットを示す図である。

【発明を実施するための形態】

【0011】

上述したように、いくつかのコンピューティング・タスクは、コンピューティング・タスクによって消費されるコンピューティング・リソースの量が急に劇的に変化する、いわゆる「スパイク性がある」挙動を定期的を示す可能性がある。例えば、ストレージ・スペース、メモリ割当、またはCPU使用率などの量は、正常なプログラム動作の一部として急に増加する（またはスパイクする）可能性がある。ただし、このことは、プロセスがクラッシュした（またはそうでなければ異常動作した）ときに生じる可能性がある。したがって、「スパイク性がある」挙動を示すジョブまたはプロセスにより、リソース消費量（例えばメモリ使用量またはプロセッサ使用率）における現在のスパイクが、システム機能またはシステム上で実行中のジョブの1つに何らかの異常があることを示すかどうかを判定することが困難になる。すなわち、問題は、ジョブ以外の何かに根付くものである可能性がある。例えば、通信デバイスの問題は、トランザクション応答時間の急な増加を引き

40

50

起こす可能性がある。したがって、予測解析ツールが、「スパイク性がある」ジョブまたはプロセスの正常な動作から生じた挙動における周期的なスパイクと、リソース消費量におけるスパイクをもたらすエラー状態を見分けることは困難である。

【0012】

これらのタイプの周期的な挙動をモデリングすることは、大量の履歴データの長期間の保持を必要とすることが多い。非常に大量のデータに対してモデリング・アルゴリズムを実行することは、ストレージ割当てと、履歴データに対して解析を実行するのに必要な時間（これは定期的なコンピューティング・タスクに利用可能な時間を減らす）の両方の点において、限られたシステム・リソースの容認できない量を消費する可能性がある。

【0013】

本発明の実施形態は、マルチモデリング・パラダイムを使用して、コンピューティング・システムのパフォーマンス・メトリックを評価するための方法およびシステムを提供する。一実施形態では、パフォーマンス・メトリックをモデリングするためのシステム・データは、複数のグループ、すなわち、パフォーマンス・メトリックの「標準（standard）」アクティビティを表す1つのグループおよびパフォーマンス・メトリックの「スパイク（spike）」アクティビティを表す1つ（または複数）の追加のグループとして記憶される。グループは別々にモデリングされて、パフォーマンス・メトリックの「正常 - 標準（normal-standard）」または予想値を表す1つの予測値およびスパイクの間に予想されるメトリックの「スパイク - 標準（spike-standard）」値を表す1つ（または複数）の予測値を考慮する。そうすることで、スパイクが生じたときに誤った例外を発行することを回避するが、依然として、パフォーマンス・メトリックの値がスパイクの間にモデリングされた「スパイク - 標準」値を外れているときに、有効な例外を投げるができる。

【0014】

この手法は、予測解析ツールのデータ保持要件を大幅に減らす。具体的には、より短い保持期間の間、「正常 - 標準」または標準値をモデリングするために使用されるデータを維持することができる。すなわち、非スパイク期間（すなわち、「正常 - 標準」）のリソース消費をモデリングするために使用される履歴データは、比較的短い時間ウィンドウ（例えば、1カ月の期間）に基づくものとして行うことができるが、スパイク期間（すなわち、「スパイク - 標準」）をモデリングするために使用されるデータは、代表的なサンプルを保持するためにはるかに長い期間（例えば、1年の期間）にわたって遡る可能性がある。しかし、スパイクがそれほど頻繁に起こらなくなると、「スパイク標準」の長期間のモデリングデータを記憶することは、ストレージ・リソースの容認できない量を予測解析ツールのために確保することをもたらさない。さらに、「正常 - 標準」モデルは、パフォーマンス・メトリックについてサンプリングされた値の長い履歴に基づかないので、この方法でデータを記憶することは、全体的な処理時間も短縮する。

【0015】

さらに、一実施形態では、ユーザは、「正常 - 標準」期間および「スパイク標準」期間に加えて、予想されるスパイクをモデリングするための具体的な期間を特定することができる。例えば、ユーザが毎月の1日に実行される反復性のジョブまたは処理タスクを作成し、しかも、関連するパフォーマンス・メトリックにおいて一時的なスパイクをもたらすジョブまたはタスクを実行すると仮定する。そのような場合、予測解析ツールは、「正常 - 標準」モデルおよび一時的な「スパイク - 標準」モデルに加えて、毎月の初めに起こることが知られているスパイクのモデルを作成することもできる。そうすることは、既知のスパイク期間が、「スパイク - 標準」モデルに従って依然として誤検知を引き起こすリソース消費レベルをもたらす場合に有益であり得る。

【0016】

以下では、本発明の実施形態を参照する。ただし、本発明は説明された特定の実施形態に限定されないことを理解されたい。その代わりに、以下の特徴および要素の任意の組合せが、異なる実施形態に関連するかどうかにかかわらず、本発明を実施し、実践するために企図される。さらに、本発明の実施形態は、他の考えられる解決策または従来技術ある

10

20

30

40

50

いはその両方に勝る利点を実現し得るが、特定の利点が所与の実施形態によって実現されるかどうかは、本発明を限定するものではない。したがって、以下の態様、特徴、実施形態および利点は例示的なものにすぎず、(1つまたは複数の)請求項に明示的に記載されている場合を除き、添付の特許請求の範囲の要素または限定とみなされない。同様に、「本発明」への言及は、本明細書に開示される任意の発明の主題の一般化として解釈されるものではなく、(1つまたは複数の)請求項に明示的に記載されている場合を除き、添付の特許請求の範囲の要素または限定であるものとみなされるものではない。

【0017】

当業者によって理解されるように、本発明の態様は、システム、方法またはコンピュータ・プログラム製品として具体化され得る。したがって、本発明の態様は、完全にハードウェアの実施形態、完全にソフトウェアの実施形態(ファームウェア、常駐ソフトウェア、マイクロ・コードなどを含む)またはすべてが一般的に本明細書において「回路」、「モジュール」または「システム」と呼ばれ得るソフトウェアの態様およびハードウェアの態様を組み合わせる実施形態の形をとり得る。さらに、本発明の態様は、具体化されるコンピュータ可読プログラム・コードを有する1つまたは複数のコンピュータ可読媒体で具体化されるコンピュータ・プログラム製品の形をとり得る。

【0018】

1つまたは複数のコンピュータ可読媒体の任意の組合せが利用され得る。コンピュータ可読媒体は、コンピュータ可読信号媒体またはコンピュータ可読記憶媒体であってもよい。コンピュータ可読記憶媒体は、例えば、電子、磁気、光学、電磁、赤外線、または半導体システム、装置、またはデバイス、あるいは上記の任意の適切な組合せであってもよいが、これらに限定されない。コンピュータ可読記憶媒体のより具体的な例(非網羅的なリスト)としては、1つまたは複数のワイヤを有する電気接続、携帯型コンピュータ・ディスクケット、ハード・ディスク、ランダム・アクセス・メモリ(RAM)、読取り専用メモリ(ROM)、消去可能プログラマブル読取り専用メモリ(EPROMまたはフラッシュ・メモリ)、光ファイバ、携帯型コンパクト・ディスク読取り専用メモリ(CD-ROM)、光記憶デバイス、磁気記憶デバイス、または上記の任意の適切な組合せが挙げられよう。この文書の文脈では、コンピュータ可読記憶媒体は、命令実行システム、装置もしくはデバイスによって、または命令実行システム、装置もしくはデバイスに関連して使用するためのプログラムを含む、または記憶することができる任意の有形媒体であってもよい。

【0019】

コンピュータ可読信号媒体は、例えば、ベースバンドにおいてまたは搬送波の一部として、その中で具体化されるコンピュータ可読プログラム・コードを有する伝搬データ信号を含み得る。そのような伝搬信号は、電磁、光学、またはこれらの任意の適切な組合せを含むがこれらに限定されない、様々な形のいずれかをとり得る。コンピュータ可読信号媒体は、コンピュータ可読記憶媒体ではなく、かつ命令実行システム、装置もしくはデバイスによってまたは命令実行システム、装置もしくはデバイスに関連して使用するためのプログラムを通信、伝搬、または伝送することができる、任意のコンピュータ可読媒体であってもよい。

【0020】

コンピュータ可読媒体で具体化されるプログラム・コードは、無線、有線、光ファイバケーブル、RFなど、または上記の任意の適切な組合せを含むがこれらに限定されない、任意の適切な媒体を使用して送信され得る。

【0021】

本発明の態様に関する操作を実行するためのコンピュータ・プログラム・コードを、Java(R)、Smalltalk、C++などのオブジェクト指向プログラミング言語および「C」プログラミング言語または類似のプログラミング言語などの従来の手続きプログラミング言語を含む、1つまたは複数のプログラミング言語の任意の組合せで書くことができる。プログラム・コードは、ユーザのコンピュータで全体的に、ユーザのコンピ

10

20

30

40

50

ユーザで部分的に、スタンドアロン・ソフトウェア・パッケージとして、ユーザのコンピュータで部分的にかつリモートのコンピュータで部分的に、またはリモートのコンピュータもしくはサーバで全体的に実行することができる。後者のシナリオでは、リモートのコンピュータは、ローカル・エリア・ネットワーク(LAN)またはワイド・エリア・ネットワーク(WAN)を含む任意のタイプのネットワークを介して、ユーザのコンピュータに接続され得るか、外部のコンピュータへの接続が(例えば、インターネット・サービス・プロバイダを使用してインターネットを介して)行われ得る。

【0022】

本発明の実施形態による方法、装置(システム)およびコンピュータ・プログラム製品のフローチャートの説明図またはブロック図あるいはその両方を参照しながら、本発明の態様を以下に説明する。フローチャートの説明図またはブロック図あるいはその両方のそれぞれのブロック、およびフローチャートの説明図またはブロック図あるいはその両方におけるブロックの組合せを、コンピュータ・プログラム命令によって実施することができることを理解されたい。これらのコンピュータ・プログラム命令は、コンピュータまたは他のプログラマブル・データ処理装置のプロセッサを介して実行する命令が、フローチャートまたはブロック図あるいはその両方の1つまたは複数のブロックにおいて指定された機能/行為を実施するための手段を作成するように、汎用コンピュータ、専用コンピュータ、または機械を作製するための他のプログラマブル・データ処理装置のプロセッサに提供され得る。

【0023】

また、これらのコンピュータ・プログラム命令は、コンピュータ可読媒体に記憶された命令が、フローチャートまたはブロック図あるいはその両方の1つまたは複数のブロックにおいて指定された機能/行為を実施する命令を含む製造品を作製するように、コンピュータ、他のプログラマブル・データ処理装置、または他のデバイスに特定の方法で機能するように命令することができるコンピュータ可読媒体に記憶され得る。

【0024】

また、コンピュータ・プログラム命令は、コンピュータまたは他のプログラマブル装置で実行する命令が、フローチャートまたはブロック図あるいはその両方の1つまたは複数のブロックにおいて指定された機能/行為を実施するためのプロセスを提供するように、コンピュータ、他のプログラマブル・データ処理装置、または他のデバイスに読み込まれて、コンピュータによって実施されるプロセスを作製するためにコンピュータ、他のプログラマブル装置または他のデバイスで一連の操作ステップを実行させ得る。

【0025】

本発明の実施形態は、クラウド・コンピューティング・インフラを介してエンド・ユーザに提供され得る。クラウド・コンピューティングは、一般的に、ネットワークを介したサービスとしてのスケラブルなコンピューティング・リソースの提供を指す。より正式には、クラウド・コンピューティングは、最小限の管理努力またはサービス・プロバイダの相互作用で迅速に提供し、リリースすることができる構成可能なコンピューティング・リソースの共有プールへの便利なオンデマンド・ネットワーク・アクセスを可能にする、コンピューティング・リソースとその基礎をなす技術的なアーキテクチャ(例えば、サーバ、ストレージ、ネットワーク)との間の抽象概念を提供するコンピューティング機能として定義され得る。したがって、クラウド・コンピューティングにより、ユーザは、コンピューティング・リソースを提供するために使用される基礎をなす物理的なシステム(またはこれらのシステムの場合)を考慮することなく、「クラウド」における仮想コンピューティング・リソースに(例えばストレージ、データ、アプリケーション、および完全仮想化コンピューティング・システムにさえ)アクセスすることができる。

【0026】

典型的には、クラウド・コンピューティング・リソースは、ユーザが実際に使用されたコンピューティング・リソース(例えば、ユーザによって消費されたストレージ・スペースの量またはユーザによってインスタンス化された仮想化システムの数)に対してのみ課

10

20

30

40

50

金されるペーパーコース (pay-per-use) 方式でユーザに提供される。ユーザは、いつでも、インターネットを介してどこからでも、クラウドに常駐するリソースのいずれにもアクセスすることができる。本発明の文脈では、ユーザは、クラウド環境に存在するアプリケーションまたは関連するデータを監視するためにアクセスすることができる。例えば、モニタリング・アプリケーションは、クラウドベースのサーバ配備における複数の仮想マシン・インスタンスで利用可能な共有メモリ (または他のリソース) の量を監視することができる。

【 0 0 2 7 】

図におけるフローチャートおよびブロック図は、本発明の様々な実施形態によるシステム、方法およびコンピュータ・プログラム製品の考えられる実装態様のアーキテクチャ、機能および動作を示す。この点において、フローチャートまたはブロック図におけるそれぞれのブロックは、(1つまたは複数の)指定された論理機能を実施するための1つまたは複数の実行可能な命令を含むモジュール、セグメントまたはコードの部分を表し得る。いくつかの代替実装態様では、ブロックに記載された機能は、図に記載された順序ではない順序で行われ得ることに留意されたい。例えば、連続して示されている2つのブロックは、実際には、実質的に同時に実行されることがあり、またはこれらのブロックは、時として、関与する機能に応じて、逆の順序で実行されることがある。ブロック図またはフローチャートの説明図あるいはその両方のそれぞれのブロック、およびブロック図またはフローチャートの説明図あるいはその両方におけるブロックの組合せを、指定された機能または行為を実行する専用ハードウェアベースシステム、または専用ハードウェアおよびコンピュータ命令の組合せによって実施することができることに留意されたい。

【 0 0 2 8 】

さらに、データ通信ネットワークを介してコンピューティング・システムのパフォーマンス・メトリックを監視する例を使用して、本発明の特定の実施形態を説明する。ただし、本明細書において記載される、複数のモデルのデータを使用してパフォーマンス・メトリックをモデリングするための技法は、コンピューティング・システムでパフォーマンス・メトリックをモデリングすることに加えて、様々な目的に適應され得ることを理解されたい。さらに、多くの場合において、予測解析ツールは、監視されているコンピューティング・システムで実行中であり得る。すなわち、予測解析ツールは、ローカルのコンピューティング・システムでリソース・パフォーマンス・メトリックを、ならびにリモートのシステムでリソースおよびパフォーマンス・メトリックを監視し得る。

【 0 0 2 9 】

図1は、本発明の実施形態が実施され得る例示的なコンピューティング・インフラ100を示す。示されるように、コンピューティング・インフラ100は、モニタリング・システム105およびそれぞれがネットワーク120に接続されたサーバ130₁、₂を含む。この例では、モニタリング・システム105は、ネットワーク120を介して通信して、サーバ130の進行中の状態を監視する。一例として、モニタリング・システム105を、それぞれのサーバ130での共有リソースの消費量を監視するように構成することができる。もちろん、モニタリング・システム105を、サーバ130の機能に関連する様々なパフォーマンス・メトリック (ならびにモニタリング・システム105のパフォーマンス・メトリック)、例えば、CPU使用率、共有 (または専用) ストレージ消費量、仮想ストレージ消費量、エラー・メッセージ・トラフィック、システム・メッセージ・ (コンソール・) トラフィック、ラッチング (ラッチの保持/解放)、トランザクション応答時間、ディスクI/O応答時間、ディスクI/Oアクティビティ (読取り、書込みなど) を監視するように構成することができる。さらに、当業者であれば、特定の場合に必要に応じて特定のメトリックが選択され得ることを認識されよう。

【 0 0 3 0 】

図2は、本発明の一実施形態による、マルチモデリング・パラダイムを使用してパフォーマンス・メトリックを監視するために使用されるモニタリング・アプリケーション222を含む例示的なコンピューティング・システム200を示す。示されるように、コンピ

ューティング・システム 200 は、限定することなく、中央処理装置 (CPU) 205、ネットワーク・インターフェース 215、インターコネクト 220、メモリ 225、およびストレージ 230 を含む。コンピューティング・システム 200 は、I/O デバイス 212 (例えば、キーボード、ディスプレイ・デバイスおよびマウス・デバイス) をコンピューティング・システム 200 に接続する I/O デバイス・インターフェース 210 も含み得る。

【0031】

一般的に、CPU 205 は、メモリ 225 に記憶されたプログラミング命令を取り出し、実行する。同様に、CPU 205 は、メモリ 225 に常駐するアプリケーション・データを記憶し、取り出す。インターコネクト 220 は、CPU 205、I/O デバイス・インターフェース 210、ストレージ 230、ネットワーク・インターフェース 215、およびメモリ 225 の間でプログラミング命令およびアプリケーション・データを送信するための通信経路を提供する。CPU 205 は、シングル CPU、マルチ CPU、複数のプロセッシング・コアを有する CPU などを表すように含まれる。また、メモリ 225 は、一般的に、ランダム・アクセス・メモリを表すように含まれる。ストレージ 230 は、ハード・ディスク・ドライブまたはソリッド・ステート・ストレージ・デバイス (SSD) であってもよい。さらに、単一のユニットとして示されているが、ストレージ 230 は、固定ディスク・ドライブ、フロッピー・ディスク・ドライブ、テープ・ドライブ、リムーバブル・メモリ・カード、光ストレージ、ネットワーク・アタッチト・ストレージ (NAS)、またはストレージ・エリアネットワーク (SAN) などの、固定またはリムーバブルあるいはその両方のストレージ・デバイスの組合せであってもよい。

【0032】

例示的には、メモリ 225 は、第 1 のモデル閾値 224 および第 2 のモデル閾値 226 とともに、モニタリング・アプリケーション 222 を記憶する。ストレージ 230 は、第 1 のモデルについてのサンプリングされたメトリック履歴 232、第 2 のモデルについてのサンプリングされたメトリック履歴 234、およびオプションの日/時スパイク・パターン・データ 236 を含む。一実施形態では、モニタリング・アプリケーション 222 は、パフォーマンス・メトリックが第 1 のモデル閾値 224 および第 2 のモデル閾値 226 によって指定された閾値を超える (または、適切な場合において、下回る) ときにアラーム (例えば、システム管理者に送られる警告メッセージ) を生成するように構成される。さらに、モニタリング・アプリケーション 222 は、第 1 のモデルについてのサンプリングされたメトリック履歴 232 を使用して、かつ第 2 のモデルについてのサンプリングされたメトリック履歴 234 を使用して、閾値 224、226 の値を導出するように構成され得る。例えば、第 1 のモデル閾値 224 は、スパイク期間に関連付けられていないサンプリングされた値に基づいて、パフォーマンス・メトリックの予想最大 (または最小) 値を提供し得る。したがって、第 1 のモデルについての、すなわち「正常 - 標準」値についてのサンプリングされたメトリック履歴 232 は、サンプリングされたメトリック値の比較的最近の履歴 (例えば、4 週間の期間) をカバーするサンプル・データを含み得る。

【0033】

同時に、第 2 のモデル閾値 226 についてのサンプリングされたメトリック履歴 234 は、サンプリングされたメトリック値の比較的長い履歴をカバーするデータを含み得る。しかし、第 2 のモデルについてのサンプリングされたメトリック履歴 234 は、パフォーマンス・メトリックにおけるスパイクが観察されていた期間の間にサンプリングされたデータ値に限定される。すなわち、サンプリングされたメトリック履歴 234 は、適切な「スパイク - 標準」閾値を決定するために使用される。

【0034】

一実施形態では、モニタリング・アプリケーション 222 は最初に、数時間 (または必要に応じて数日) のデータ収集を使用して、パフォーマンス・メトリックの基本アクティビティ・レベルを確立する。スパイクがこの時間の間に生じた場合、スパイクは「正常 - 標準」閾値の予想されるアクティビティの計算をわずかに、ただし一般的には全体的な結

10

20

30

40

50

果に影響を及ぼさない程度にゆがめることがある。

【0035】

例えば、データ・コレクタが2週間の期間で30分毎にメトリックをサンプリングし、メトリックが履歴データ・ファイルに(すなわち、第1のモデルについてのサンプリングされたメトリック履歴232として)記憶されると仮定する。2週間の期間にわたってサンプリングされたデータに基づいて、第1のモデル閾値224が特定される。この点で、十分な履歴データが収集されるまで、生じるいかなるスパイクも、異常な挙動として特定され、警告を引き起こすことになる。したがって、一実施形態では、警告をトリガする任意のパフォーマンス・メトリック値は、標準の正常な挙動の計算に対するあらゆる追加のゆがみを回避するために、スパイク・アクティビティ用の別個の履歴データ・ファイルに転送される、すなわち、観察されたスパイク期間の間にサンプリングされたパフォーマンス・メトリック値は、第2のモデルについてのサンプリングされたメトリック履歴234に転送される。

10

【0036】

したがって、サンプリングされたメトリック履歴234は、標準または「正常 - 標準」挙動と比較したときに警告を引き起こすのに十分高いメトリック値を記憶する。十分な数のメトリック値がサンプリングされたメトリック履歴234に記憶されると、予測モデリング・アルゴリズムを使用して、「スパイク - 標準」値を決定することができる。例えば、十分な量のデータは、少なくとも3回生じたスパイク挙動のデータを含む、4から6週間の期間にわたって収集されたスパイク・データであるとみなされ得る。もちろん、このトレーニング期間は、監視されるシステムの要件に応じて調整され得る。すなわち、モニタリング・アプリケーション222は、スパイクの間に見ると予想されるメトリック値を表す値を計算し得る。そのような値は、(予想される標準偏差および分散などの何らかの許容差を有する)単一の閾値として指定され得るが、正常な動作範囲として、または特定のパフォーマンス・メトリックに適した他の形で指定され得ることに留意されたい。

20

【0037】

この点で、サンプリングされたメトリック履歴232および234データは、予想されるパフォーマンス・メトリック値の予測値を生成するために使用され得る。第1の予測値は標準の正常なメトリック値を表し、追加の予測値はアクティビティにおけるスパイクの間の正常なメトリック値を表す。これらのモデルに基づいて、それぞれのタイプの挙動の閾値224、226をプログラマティックに生成することができる。閾値224、226の感度は、ユーザ構成可能なパラメータによって微調整され得る。

30

【0038】

第1のモデル閾値224および第2のモデル閾値226が確立されると、パフォーマンス・メトリックについての観察されたサンプル値が第1のモデル閾値224を超える場合、このサンプリングされた値は、サンプリングされたメトリック履歴234に記憶され(かつ定期的なアクティビティが再開するまで、予測モデルをより頻繁に更新するために使用され)得る。さらに、パフォーマンス・メトリックについてのその後に観察されたサンプル値がスパイク期間の間に第2のモデル閾値226を超える場合、警告メッセージ、例えば、さらなる問題(complications)を防止するためにアクションをとることができるように、システム・オペレータへの警告が生成され得る。もちろん、パフォーマンス・メトリックが第1のモデル閾値224、第2のモデル閾値226、またはその両方を超える(または下回る)ときに、多種多様な他のアクションをトリガすることができる。

40

【0039】

上に記載された手法は、リソース・スパイクに先立つ事前のユーザ知識および構成の必要をなくすものであるが、場合によっては、ユーザは一定のスパイク期間が予測され得る方法でジョブまたは処理タスクを常にスケジューリングすることができる。そのような場合、モニタリング・アプリケーション222は、追加の閾値およびサンプリングされたメトリック履歴を作成して、特定のジョブまたはタスクに関連付けられたスパイク期間をモデリングするように構成され得る。例えば、データ/時間スパイク・パターン・データは、特

50

定のスパイクが生じるおよび再び生じることが予想されるときを指定し得る。さらに、第1および第2の閾値は、いったん確立されると、動的であり得る。すなわち、いったん初期値に設定されると、「正常」期間および「スパイク」期間の両方の間の次のサンプル値は、経時的に閾値を更新するために使用され得る。

【0040】

図3は、本発明の一実施形態による、マルチモデリング・パラダイムを使用して、「スパイク性がある」コンピューティング・ジョブまたはプロセスを監視するための方法300を示す。示されるように、方法300は、モニタリング・アプリケーションがトレーニング期間にコンピューティング・ジョブまたはタスクの組に関連付けられたパフォーマンス・メトリックの監視を開始するステップ305で開始する。上述したように、監視されるパフォーマンス・メトリックの例は、例えば、システム使用率、プロセッサ（またはプロセッサ・コア）使用率、共有（または専用）ストレージ消費量、仮想ストレージ消費量、エラー・メッセージ・トラフィック、システム・メッセージ・（コンソール・）トラフィック、ラッチング（ラッチの保持/解放）、トランザクション応答時間、ディスクI/O応答時間、ディスクI/Oアクティビティ（読取り、書込みなど）を含む、コンピューティング・システム、グリッド、クラスタ、ネットワークなどの様々な態様を含むことができる。トレーニング期間により、モニタリング・システムは第1の閾値（すなわち、正常・標準）を決定するだけでなく、第2の閾値（すなわち、スパイク・標準）も決定することができる。

【0041】

ステップ310で、トレーニング期間の間に、監視されたパフォーマンス・メトリックのサンプリングされた値が第1の閾値についての値を超えると、モニタリング・システムはあらゆるアラームを抑制する。これが行われるのは、第1の閾値（すなわち、正常・標準閾値）が比較的迅速に（すなわち、数時間または数日の期間において）確立され得るからである。さらに、いったん確立されると、第1の閾値は、第2の閾値をモデリングするために使用されるデータがそれ自体で決定される期間を特定するために使用される。例えば、第2の閾値についてのデータは、監視されるパフォーマンス・メトリックが第1の閾値を超える期間（すなわち、スパイク・アクティビティの期間の間）に限定され得る。ステップ315で、モニタリング・システムは、第2の閾値（すなわち、スパイク・標準閾値）を決定するための十分なスパイク・データが観察されたかどうかを判定する。スパイク期間の頻度および継続時間に応じて、トレーニング期間は数週間または数カ月の期間継続し得る。スパイク期間の代表的なサンプルを観察した後、モニタリング・システムは、パフォーマンスにおける正常なスパイクとユーザ介入を必要とし得るイベントを区別するための第2のモデル閾値を決定する（ステップ320）。すなわち、モニタリング・システムは、第2のモデル閾値を決定する。

【0042】

図4は、本発明の一実施形態による、マルチモデリング・パラダイムを使用して、予測解析を実行するための方法400を示す。示されるように、方法400は、モニタリング・アプリケーションが現在のサンプリング期間のパフォーマンス・メトリックの値を判定するステップ405で開始する。もちろん、サンプリング頻度は、モニタリング・システムによって監視される特定のパフォーマンス・メトリックについて、必要に応じて設定され得る。

【0043】

ステップ410で、モニタリング・システムは、ステップ405でサンプリングされたパフォーマンス・メトリックの値が第1のモデルについての閾値を超える（または、適切な場合において、下回る）かどうかを判定する。閾値を超えない場合、システムは、次のサンプル期間の時間に達するまで、ステップ405に戻る。そうでなければ、サンプリングされたパフォーマンス・メトリック値が閾値を超える場合、システムは第2のモデルを使用して、監視されたメトリックの評価を開始する。一実施形態では、スパイクが観察されると、スパイク期間の間により綿密にパフォーマンス・メトリックを監視するために（

非スパイク期間の間のサンプリング頻度に対して) サンプリング頻度を上げることができることに留意されたい。

【 0 0 4 4 】

ステップ 4 2 0 で、パフォーマンス・メトリックが第 2 のモデルを使用して決定された第 2 の閾値 (すなわち、スパイク - 標準閾値) を超える場合、ステップ 4 2 5 で、パフォーマンス・メトリックについてアラーム・メッセージを送ることができる。そうでなければ、パフォーマンス・メトリックが、スパイクを経験している間、「正常な」スパイクを経験していることをパフォーマンス・メトリックの評価が示す場合、システムはステップ 4 0 5 に戻って、次のサンプリング期間を待つ。

【 0 0 4 5 】

マルチモデリング手法の例は、図 5 ~ 図 6 に示されている。より詳細には、図 5 ~ 図 6 は、本発明の一実施形態による、マルチモデリング・パラダイムを使用するように構成された予測解析ツールによって監視された例示的なデータ・セットを示す。図 5 は、2 週間の期間にわたって得られたメトリック値のサンプル 5 0 0 を示す。この例では、2 つのスパイク 5 1 0、5 1 5 の間を除き、パフォーマンス・メトリックについて、約 1 0 0 の値が全体的に得られる。この例について、スパイク 5 1 0 は (パフォーマンス・メトリックを介して) 監視されているコンピューティング・システムの正常なアクティビティから生じ、スパイク 5 1 5 はクラッシュまたは他のシステム動作不良から生じると仮定する。アラームの閾値が約 1 5 0 に設定された場合、アラームはスパイク 5 1 0 (誤検知) とスパイク 5 1 5 (実際の問題) の両方から生成される。したがって、上に記載したように、マルチモデリング手法は、パフォーマンス・メトリックのいわゆる「正常」値とスパイク期間について使用する別個のモデルの両方をモデリングするために使用され得る。この結果は図 6 に示されている。

【 0 0 4 6 】

図 6 に示されるように、パフォーマンス・メトリックについてのデータ 5 5 0 は、図 5 に示されるよりも長い期間キャプチャされる。さらに、第 1 の閾値 5 5 5 は約 1 5 0 に設定され、第 2 の閾値 5 6 0 は約 4 2 5 に設定される。トレーニング期間 5 6 5 の間、スパイク 5 8 0、5 8 5 についてのデータは、第 2 の閾値 5 6 0 についての値を決定するために使用される。トレーニングが完了すると、スパイク 5 7 5₁ は、第 2 の閾値 5 6 0 を超えないので、アラームを生成しない。対称的に、スパイク 5 7 5₂ は第 2 の閾値 5 6 0 を超え、アラームを引き起こす。

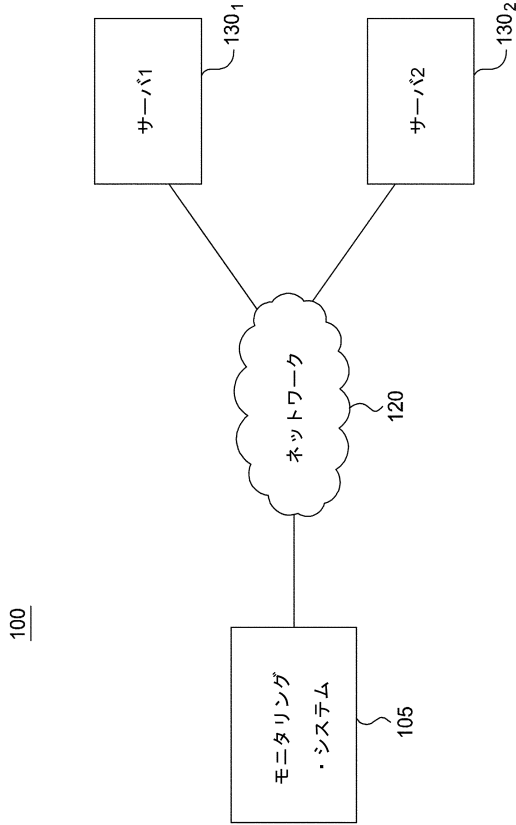
【 0 0 4 7 】

したがって、有利には、上に記載されたマルチモデリング手法は、正常なシステム処理の一部として生じるスパイクの間の警告の発行を回避することによって、予測解析を改善する。この手法は、監視されるコンピューティング・システムに関する予測解析の精度を高め、周期処理サイクルを定義するルールの作成を必要とせず、予測モデリングを実行するのに必要なデータの量を低減し、予測モデリングを実行するのに必要な CPU の量を低減する。

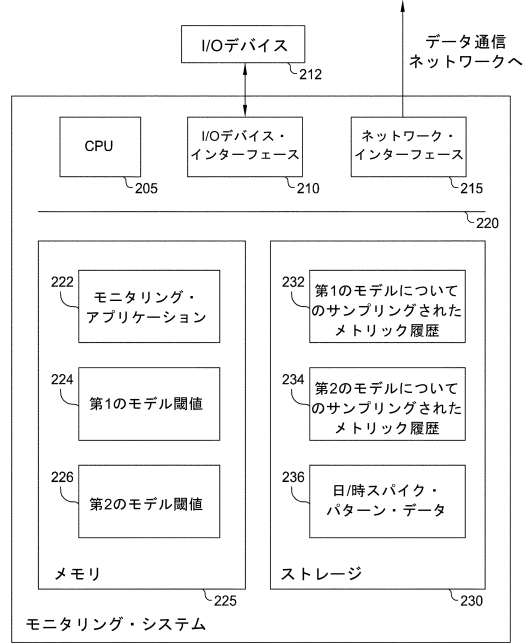
【 0 0 4 8 】

上記は本発明の実施形態を対象とするが、その基本範囲を逸脱することなく、本発明の他のおよびさらなる実施形態を考案することができ、その範囲は添付の特許請求の範囲によって決定される。

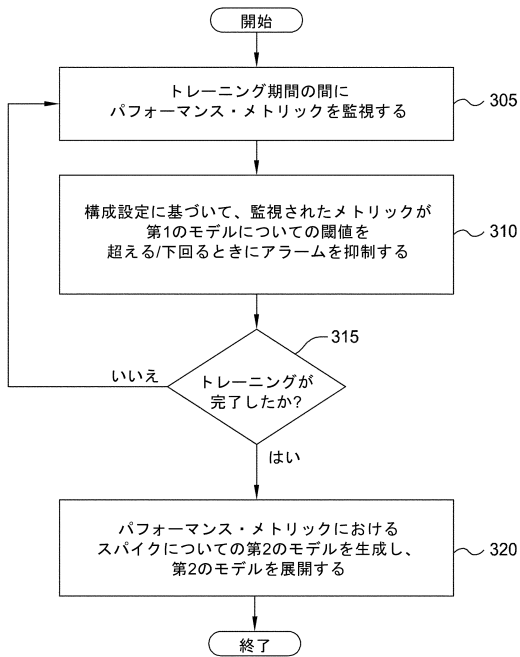
【図1】



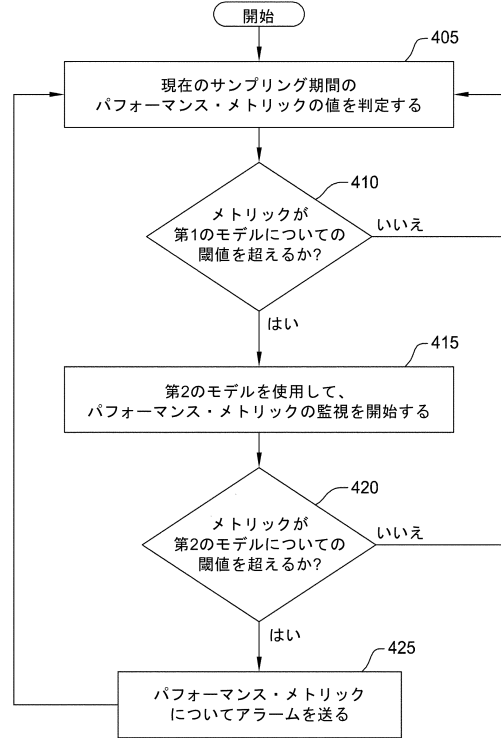
【図2】



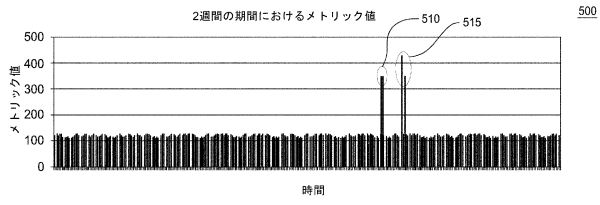
【図3】



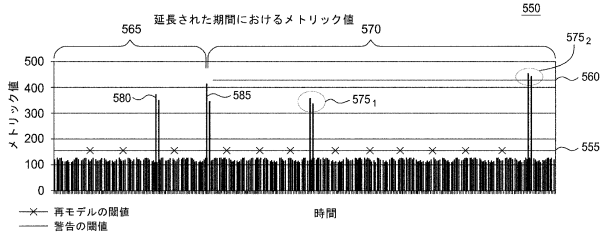
【図4】



【 図 5 】



【 図 6 】



フロントページの続き

(74)代理人 100112690

弁理士 太佐 種一

(72)発明者 ペイトン、アスペン、リー

アメリカ合衆国55901 ミネソタ州ロチェスター ハイウェイ52ノース 3605

(72)発明者 キャプリー、ジェームス

アメリカ合衆国12601 ニューヨーク州ポキプシー サウス・ロード2455

(72)発明者 アート、カーラ、ケイ

アメリカ合衆国55901 ミネソタ州ロチェスター ハイウェイ52ノース 3605

(72)発明者 パテール、ケユール

アメリカ合衆国12601-5400 ニューヨーク州ポキプシー サウス・ロード2455

審査官 多胡 滋

(56)参考文献 特開2005-316808(JP,A)

国際公開第2010/032701(WO,A1)

特開2001-142746(JP,A)

特開2009-003742(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 11/07

G06F 11/34