

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-146543

(P2010-146543A)

(43) 公開日 平成22年7月1日(2010.7.1)

(51) Int.Cl.		F I		テーマコード (参考)
G06F 11/30	(2006.01)	G06F 11/30	320C	5B042
G06F 11/34	(2006.01)	G06F 11/34	L	

審査請求 有 請求項の数 26 O L (全 20 頁)

(21) 出願番号 特願2009-120245 (P2009-120245)
 (22) 出願日 平成21年5月18日 (2009.5.18)
 (31) 優先権主張番号 12/336, 221
 (32) 優先日 平成20年12月16日 (2008.12.16)
 (33) 優先権主張国 米国 (US)

(71) 出願人 300015447
 エスアーベー アーゲー
 SAP AG
 ドイツ連邦共和国, 69190 バルドルフ,
 ディートマルーホップーアレー 16
 Dietmar-Hopp-Allee
 16, 69190 Walldorf,
 Germany
 (74) 代理人 100108453
 弁理士 村山 靖彦
 (74) 代理人 100064908
 弁理士 志賀 正武
 (74) 代理人 100089037
 弁理士 渡邊 隆

最終頁に続く

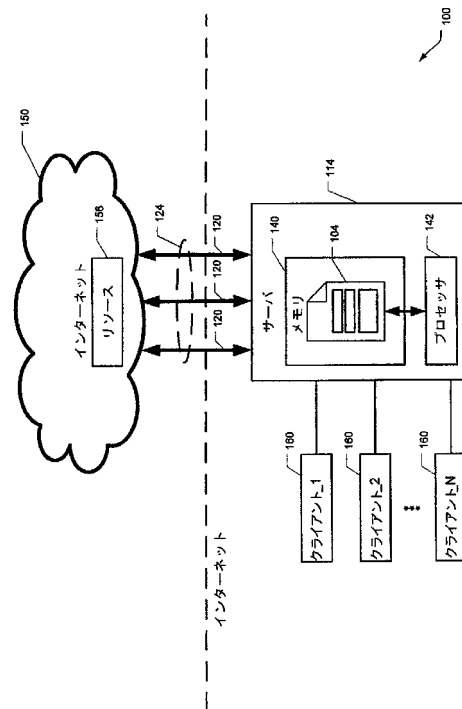
(54) 【発明の名称】 メモリ消費監視

(57) 【要約】

【課題】 相関分析に基づきメモリ消費監視を実行する。

【解決手段】 メモリガーベージコレクションイベント及びノードコンテンツサービスリクエストの発生を監視し、ノードコンテンツサービスリクエストとメモリガーベージコレクションイベントとの間の相関を決定し、かつ、ノードコンテンツサービスリクエストのうちの疑わしいリクエストと関連した情報を報告するように動作する、装置、システム、及び、方法が開示されている。疑わしいリクエストは、相関と関連した予め選択された基準を満足するリクエストとして識別してもよい。追加的な装置、システム、及び、方法が開示されている。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

コンピュータ実行可能な方法であって、

メモリガーベージコレクションイベントと、ノードコンテンツサービスリクエストとの発生を監視するステップと、

前記ノードコンテンツサービスリクエストと、前記メモリガーベージコレクションイベントとの間の相関を決定するステップと、

前記相関と関連した予め選択された基準を満足する前記ノードコンテンツサービスリクエストのうちの疑わしいものと関連した情報を報告するステップと

を具備する方法。

10

【請求項 2】

前記監視するステップは、前記メモリガーベージコレクションイベントの開始時間又は終了時間のうちの少なくとも1つを記録するステップを含むことを特徴とする請求項1に記載の方法。

【請求項 3】

前記監視するステップは、前記ノードコンテンツサービスリクエストの開始時間又は終了時間のうちの少なくとも1つを記録するステップを含むことを特徴とする請求項1に記載の方法。

【請求項 4】

前記監視するステップは、前記ノードコンテンツサービスリクエストをキューで記録するステップを含むことを特徴とする請求項1に記載の方法。

20

【請求項 5】

前記監視するステップは、前記ノードコンテンツサービスリクエストの一部として、ユニフォームリソースロケータ (URL) 情報を記録するステップを含むことを特徴とする請求項1に記載の方法。

【請求項 6】

前記決定するステップは、前記メモリガーベージコレクションイベントによって定義された時間間隔中の前記ノードコンテンツサービスリクエストの数の間の相関を決定するステップを含むことを特徴とする請求項1に記載の方法。

【請求項 7】

前記決定するステップは、前記メモリガーベージコレクションイベントによって定義された時間間隔中の選択されたユニフォームリソースロケータを含む前記ノードコンテンツサービスリクエストの包含的な平均数と、前記選択されたユニフォームリソースロケータを含まない前記ノードコンテンツサービスリクエストの排他的な平均数との間の相関を決定するステップを含むことを特徴とする請求項1に記載の方法。

30

【請求項 8】

前記包含的な平均と前記排他的な平均との差異として、前記相関を計算し、前記包含的な平均と前記排他的な平均との最大値によって除算するステップをさらに含むことを特徴とする請求項7に記載の方法。

【請求項 9】

前記時間間隔は、ガーベージコレクションイベントの開始時間、及び、ガーベージコレクションイベントの終了時間のうちの1つによって定義されることを特徴とする請求項7に記載の方法。

40

【請求項 10】

前記決定するステップは、前記メモリガーベージコレクションイベントによって定義された時間間隔中の前記ノードコンテンツサービスリクエストの数の間の前記相関を決定するステップを含み、

ノードコンテンツサービスリクエストの数は、複数の共通ユニフォームリソースロケータ値に基づくことを特徴とする請求項1に記載の方法。

【請求項 11】

50

前記サービスリクエストの定量分析を実行して、前記相関のため、少なくとも1つの重み付き係数を提供するステップをさらに含むことを特徴とする請求項1に記載の方法。

【請求項12】

コンピュータ実行可能な方法であって、
メモリガーベージコレクションイベントと、ユニフォームリソースロケータを含むノードコンテンツサービスリクエストとの間のタイミング関係を監視するステップと、
前記ノードコンテンツサービスリクエストと、前記メモリガーベージコレクションイベントとの間の相関を決定するステップと、
前記相関と関連した予め選択された基準を満足する前記ノードコンテンツサービスリクエストのうちの疑わしいものと関連した情報を報告するステップと
を含むことを特徴とする方法。

10

【請求項13】

前記決定するステップは、選択された数の前記メモリガーベージコレクションイベントによって定義された間隔中の前記ノードコンテンツサービスリクエストの包含的な平均数と、前記選択された数の前記メモリガーベージコレクションイベントによって定義された間隔中の前記ノードコンテンツサービスリクエストの排他的な平均数との間の前記相関を決定するステップを含むことを特徴とする請求項12に記載の方法。

【請求項14】

前記報告するステップは、グラフィックユーザインターフェースの一部として、ウェブページウィンドウ、又は、スライドバーウィンドウのうちの少なくとも1つを表示するステップを含むことを特徴とする請求項12に記載の方法。

20

【請求項15】

前記ノードコンテンツサービスリクエストが前記ノードによってもはや処理されないように、ネットワーク内のノードを命令して、維持モードに入るステップをさらに含むことを特徴とする請求項12に記載の方法。

【請求項16】

前記ノードを命令して、プロファイラモードに入るステップと、
前記ノードコンテンツサービスリクエストのうちの前記疑わしいもののうちの少なくとも1つに対して、前記ノード内に割り当てられた第1メモリ量を決定するステップと
をさらに含むことを特徴とする請求項15に記載の方法。

30

【請求項17】

前記メモリガーベージコレクションイベントのうちの1つによって割り当てを解除された第2メモリ量を決定するステップをさらに含むことを特徴とする請求項16に記載の方法。

【請求項18】

前記第1メモリ量と、前記第2メモリ量とを比較して、メモリリークが存在するか否かを決定するステップをさらに含むことを特徴とする請求項17に記載の方法。

【請求項19】

前記メモリガーベージコレクションイベントを選択的に無効にするステップをさらに含むことを特徴とする請求項12に記載の方法。

40

【請求項20】

複数のノードコンテンツサービスリクエストと関連した前記ユニフォームリソースロケータ情報を組み合わせて、前記複数のノードコンテンツサービスリクエストのうちの個別のものとの間の前記相関を決定するステップをさらに含むことを特徴とする請求項12に記載の方法。

【請求項21】

コンピュータに方法を実行させるために格納された命令を含むコンピュータ可読媒体であって、前記方法は、
メモリガーベージコレクションイベントと、ノードコンテンツサービスリクエストとの発生を監視するステップと、

50

前記ノードコンテンツサービスリクエストと、前記メモリガーベージコレクションイベントとの間の相関を決定するステップと、

前記相関と関連した予め選択された基準を満足する前記ノードコンテンツサービスリクエストのうちの疑わしいものと関連した情報を報告するステップと
を具備することを特徴とするコンピュータ可読媒体。

【請求項 2 2】

前記方法は、

前記相関が前記予め選択された基準と関連した閾値を超過したことを検出するステップと、

前記ノードコンテンツサービスリクエストと、前記メモリガーベージコレクションイベントとの間の前記相関が、前記ノードコンテンツサービスリクエストのうちの前記疑わしいものの追加的な分析をトリガするのに十分であることを示すステップと

を含むこと特徴とする請求項 2 1 に記載のコンピュータ可読媒体。

【請求項 2 3】

前記報告するステップは、グラフィカルユーザインターフェースの一部として、前記ノードコンテンツサービスリクエストのうちの前記疑わしいものと関連した、前記相関又は前記情報のうちの少なくとも 1 つを表示するステップを含むこと特徴とする請求項 2 1 に記載のコンピュータ可読媒体。

【請求項 2 4】

装置であって、

メモリガーベージコレクションイベントと、ノードコンテンツサービスリクエストとの発生を検出するための監視モジュールと、

前記ノードコンテンツサービスリクエストと、前記メモリガーベージコレクションイベントとの間の相関を決定するためのプロセッサ実行可能な相関モジュールと、

前記相関と関連した予め選択された基準を満足する前記ノードコンテンツサービスリクエストのうちの疑わしいものと関連した情報を報告するための報告モジュールと
を含むことを特徴とする装置。

【請求項 2 5】

前記メモリガーベージコレクションイベントと関連したタイミング情報を記録するためのメモリをさらに含むことを特徴とする請求項 2 4 に記載の装置。

【請求項 2 6】

前記ノードコンテンツサービスリクエストのうちの疑わしいものと関連した前記情報に関して、アラートメッセージを表示するためのディスプレイをさらに含むことを特徴とする請求項 2 4 に記載の装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、メモリ消費監視に関する。

【背景技術】

【0002】

Java（登録商標）アプリケーションサーバなどのアプリケーションサーバは、多くの場合、多数のサービスリクエストを処理するようにタスク化される。したがって、数個のリクエストだけのときでさえ、サーバ性能は、劇的に低下し、比較的多量のメモリを消費する結果となり得る。このため、そのようなシステムにおけるメモリ消費監視は、システム全体の安定性の維持に有用である。しかし、システムが高度に利用されるとき、メモリ監視活動自体がシステム性能に影響を与え得る。

【先行技術文献】

【非特許文献】

【0003】

【非特許文献 1】 Analysis of Performance Problems in SOA: Find the Right Track (

10

20

30

40

50

オンライン), URL: <http://www.ap-verlag.de/Online-Artikel/SOA/200609%20Opnet%20Performance%20in%20SOA.htm>

【発明の概要】

【課題を解決するための手段】

【0004】

コンピュータ実行可能な方法であって、
メモリガーベージコレクションイベントと、ノードコンテンツサービスリクエストとの発生を監視するステップと、
前記ノードコンテンツサービスリクエストと、前記メモリガーベージコレクションイベントとの間の相関を決定するステップと、
前記相関と関連した予め選択された基準を満足する前記ノードコンテンツサービスリクエストのうちの疑わしいものと関連した情報を報告するステップと
を具備する。

10

【図面の簡単な説明】

【0005】

【図1】本発明の多様な実施例によるメモリ消費監視のプロセス図である。

【図2】本発明の多様な実施例によるメモリ消費監視のための装置のブロック図である。

【図3】本発明の多様な実施例によるメモリ消費監視のためのシステムのブロック図である。

【図4】本発明の多様な実施例によるメモリ消費監視の方法を図示したフローチャートである。

20

【図5】本発明の多様な実施例によるメモリ消費監視の追加的な方法を図示したフローチャートである。

【図6】本発明の多様な実施例による例示的な形態のコンピュータシステムにおける装置のブロック図である。

【発明を実施するための形態】

【0006】

「セッション」は、通信エンドポイント間の一連のインタラクションとして特徴付けることができる(特定の接続がアクティブな間に生じる)。セッションを開始するために、クライアントなどの1つのノードは、サーバなどの他のノードとの接続をリクエストしてもよい。ノードが合意した場合、コマンド及びデータを交換することができる。この交換はセッションの一部を構成する。セッションは、接続が両エンドで確立されるときに開始し、かつ、接続が終了されるときに終了する。追加的なセッションを開始するための更なるリクエストは共通である。

30

【0007】

システム性能に対して比較的影響の少ない大規模なサービスリクエストを適応するため、本明細書に記載されたメカニズムを使用して、相関分析に基づくメモリ消費監視を実行することができる。例えば、多くの実施例では、ガーベージコレクション(GC)イベントとサービスリクエスト活動との間の相関の統計的な評価を使用して、疑わしいリクエスト活動(過度のメモリ使用をもたらすように見える)を識別することができる。そのようなリクエストを識別及び規制することによって、システムオペレーションの安定性を向上させることができる。例えば、一部の実施例では、これらのメカニズムを管理インターフェースの一部として使用して、メモリ消費(JAVA(登録商標)アプリケーションサーバにおいて生じる)を監視することができる。

40

【0008】

JAVA(登録商標)アプリケーションの使用、及び、JAVA(登録商標)アプリケーションサーバが本明細書において一例として説明されるが、説明される多様な実施例はそれに限定されないことに注意しなければならない。メモリ(データ処理システム内のサービスのためのリクエストを適応するためのリソースとして使用することができ、プログラミング言語と関連して動作する)を備えた、どのようなデバイスでも、本明細書に説明され

50

るような過度のメモリ消費を監視することができる。

【 0 0 0 9 】

本明細書の目的上、「アプリケーション」は、プロセッサによって実行することができる命令の集合を意味する。そのようなアプリケーションは、ブラウザ、ワードプロセッサ、スプレッドシートプロセッサ、プレゼンテーションプロセッサ、データベースプロセッサ、ドローイングプロセッサなどを含む。アプリケーションを使用して、電子コンテンツを処理してもよい。

【 0 0 1 0 】

「電子コンテンツ」は、ユーザに示される（例えば、視覚的又は聴覚的に示される）デジタルデータを含む（電子ドキュメント、ページ記述的電子ドキュメントなどのページ記述的電子コンテンツ、メディアストリーム、ウェブページ、ハイパーテキストドキュメント、画像、デジタルビデオ又はビデオレコーディング、デジタルオーディオ又はオーディオレコーディング、アニメーション、例えば、ハイパーテキスト・マークアップ言語（HTML）又はXMLを含むドキュメントなどのマークアップ言語ドキュメント、記入されるコンポーネントを備えたフォーム、又は、グラフィカルユーザインターフェース（GUI）を記述するデータなど）。電子コンテンツは多様なコンテンツ要素である。

【 0 0 1 1 】

「コンテンツ要素」は、電子コンテンツの部分又は一部（部分又は一部として定義されるか又は認められる）を含む。例えば、コンテンツ要素は、コンテンツ要素自体の特徴（例えば、電子ドキュメントの段落又はファイルフォーマット指定）を自動的に識別してもよく、又は、ユーザによって手動で定義してもよい（例えば、電子ドキュメント内のユーザが選択したワードの集合、デジタル画像のユーザが選択した部分）。コンテンツ要素の実施例は、ページ記述的ドキュメント又は他の電子ドキュメントの一部を含む（例えば、電子テキスト又は他のマテリアルの部分など、電子ドキュメント内の命令のグループなど、デジタルビデオのセクション、又は、デジタルビデオ又はデジタルオーディオのフレーム又はフレームセットなどのメディアストリームの部分の形態のダイナミックコンテンツ、アニメーションのセグメント又はフレームの形態のダイナミックコンテンツ、電子フォーム、フォームテンプレート、フォーム要素、フォームデータ、動作可能な要素仕様又は実行可能な命令、及び、電子コンテンツ内のユーザによって表示可能又はアクセス可能な多様な要素（スクリプト化及び非スクリプト化されたダイナミックコンテンツのインスタンスなどを含む））。

【 0 0 1 2 】

多数のモジュールを含むいくつかのアプリケーション又はプロセスが本明細書に説明される。「モジュール」は、特徴のある機能のユニット（他のモジュールに情報を提供し、かつ、他のモジュールから情報を受け取ることができる）である。したがって、説明されたモジュールは、通信的に接続されているものとして取り扱ってもよい。また、モジュールは、入力又は出力デバイスとの通信を開始し、かつ、リソースに作用することができる（例えば、情報の収集）。モジュールは、必要に応じて、多様な実施例の特定の実行のため、ハードウェア回路、光学コンポーネント、単一又は複数のプロセッサ回路、メモリ回路、ソフトウェアプログラムモジュール及びオブジェクト、ファームウェア、及び、それらの組み合わせを含んでもよい。用語「モジュール」は、コード、データ、又は、計算オブジェクトの識別可能な部分を含んでもよい（例えば、特定の機能、動作、処理、又は、手順を実行するように設計された、ドキュメントアプリケーション）。

【 0 0 1 3 】

「ノード」は、サーバ、クライアント、又は、サービスリクエストを受信可能な独立計算デバイスである。

【 0 0 1 4 】

「リソース」は、ウェブページ、ディスク上のファイル、又は、コンピュータシステム内部の表示（manifestation）である。リソースの他の実施例は、とりわけ、アドレスブックアイテム、データベースエントリ、及び、カレンダーアポイントメントを含んでいる。

【0015】

本明細書に開示された多様な実施例は、監視されているシステムに対して比較的小さな影響で、メモリ消費監視を可能にするように動作することができる。このタイプの動作を含む一部の代表的な実施例がここでより詳細に説明される。

【0016】

図1は、本発明の多様な実施例によるメモリ消費を監視するプロセス100を図示している。例えば、サーバ114がアプリケーション104を実行する場合を検討する。サーバ114は、1又は複数のネットワーク124を使用して、インターネット150に接続してもよい。多様なリソース156にアクセスする一部として、多数のユニフォームリソースロケータ（URL）サービスリクエスト120をネットワーク124上に入力してもよい。そのようなリクエストは、多数のクライアント160（例えば、特定のウェブサイトに連続的にアクセスするよう動作する）によるブラウザの使用によって生じてもよい。サービスリクエスト120の一部は、サーバ114内のメモリ140の著しい計算を生じる。

10

【0017】

Java（登録商標）仮想マシン（JVM）では、変数などのアイテムを格納するために使用されたメモリ140は、ヒープスペースと呼ばれる。ヒープスペースは、世代（generation）と呼ばれるより小さなスペースに分割することができる。ヒープ内には、若い世代スペース（エデン（Eden）スペース、生存（Survivor）0スペース、生存1スペースを含む）を含むいくつかの世代が一般に存在する。

20

【0018】

特定の世代（オブジェクトを分類する）は、その年齢に依存する。一般に、メモリ内に配置された新規に生まれたメモリオブジェクトは、第1にエデンスペース内に作成される。エデンスペースが満杯であるとき、まだ生きているメモリアイテムは生存0スペースに移動され、かつ、古いオブジェクト（もはや必要とされていない）はJVMによってエデンスペースからクリアされる。これによって、新規に入ってくるオブジェクトに対して余地を設ける。生存0スペースが満杯であるとき、そのスペース内でまだ生きているオブジェクトは生存1スペースに移動され、かつ、生存0スペースは一掃される。

【0019】

メモリ140内のスペースは、GC活動を通じて、JVMによって部分的に管理される。ガーベジコレクタは、メモリヒープスペース周囲の物を移動させ、かつ、あるアイテムが保存されるか又は廃棄されるかを決定する。GC活動がなければ、使用可能なメモリ140は比較的速く使い果たされる。したがって、URLリクエストなどの多数のサービスリクエスト120の後、GCイベントが起こり、もはや必要ではないメモリ140を解放する。

30

【0020】

この開示の目的上、同じサービスリクエストが多かれ少なかれ同じ量のメモリ消費を生じることを仮定することができる。これは、クラスローディング、キャッシュフィリングなどにより割り当てられた追加的なメモリが存在するので、例えば、初回（URLがアクセスされる）に関して、真か、又は、真ではなくてもよい。しかし、第1アクセス動作中に消費されたメモリの合計量は廃棄するか、又は、恐らく、特別な手順の一部として考慮してもよい。次のURLサービスリクエストは、平均的な消費の値をより表さなければならない。

40

【0021】

JVM及びGCルーチンを利用する他のプロセスのため、いったん定義されたメモリ領域（例えば、エデンスペース）が使い果たされると、CGルーチンがトリガされることを仮定することができる。その結果、当該サーバ又は他の機械が、GCイベントを経験することなく、固定された数のサービスリクエストを取り扱うことができることを予期することができる。したがって、あるリクエスト（例えば、特定のURLにアクセスする）が多くのメモリを使用した場合、CG間隔の間のリクエストの合計数は一般に減少する。したがって、相関分析を使用して、間隔の分散（このタイプの「疑わしい」リクエストを含む、及び、含まない）を見出すことができる。

50

【 0 0 2 2 】

図 2 は、本発明の多様な実施例によるメモリ消費を監視するための装置214のブロック図である。装置214はノードを備え、恐らく、図 1 のサーバ114に類似しているか又は同一であってもよい。図 2 に戻ると、装置214は、ハードウェア（いくつかのモジュール内に入力としてデータを供給する）を備えていることが分かる。この場合には、ハードウェアはプロセッサ及びメモリ216を備え、かつ、データはGCイベント226、及び、入ってくるサービスリクエスト228（例えば、URLアクセスのためのリクエスト）を含んでいる。モジュールは、監視モジュール244、関連モジュール248、及び、分析/報告モジュール252を含んでいる。

【 0 0 2 3 】

GCイベント226及び入ってくるリクエスト228は、プロセッサ及びメモリ216の方では、活動の結果として提供することができる。入ってくるリクエスト228が発生し、かつ、メモリが応答して割り当てられるので、GCイベント226も発生する。リクエスト228及びイベント226が実行されるので、それらのアピアランス（appearance）に関する情報は、非常に小さな性能影響でアーカイブすることができる。

【 0 0 2 4 】

更なる分析のための疑わしいリクエストを識別するため、CGイベント226（サービスリクエストがより少なく発生する）の間の時間間隔に注目が集まる。これは、これらの状態がメモリ消費活動の特徴であるためである。したがって、個別のサービスリクエストは、一部の選択された数の間隔INT（CGイベント226の周期的な発生によって定義された）を評価することができる。

【 0 0 2 5 】

一部の実施例では、メモリ消費の監視は、次のステップによるGCイベント226及び入ってくるリクエスト228の処理を含んでもよい。

1．監視モジュール224を使用して、イベント226及びリクエスト228をアーカイブし、かつ、恐らく、所定のタイプ（所定のサービスリクエストタイプ（例えば、URL参照を含んだリクエスト）を含んだ選択された数のGCイベント間の間隔INTにおいて発生する）のリクエストXの平均数を計算するステップ

2．監視モジュール224を使用して、所定のリクエストを含まない間隔INTにおけるリクエストYの平均数を計算するステップ

3．関連モジュール244を使用して、 $r = (X - Y) / \max(X, Y)$ （ここで、 $\max(X, Y)$ はX又はYのどちらか一方の最大値である）として相関値を計算するステップ。これによって、特定のリクエストタイプ（例えば、特定のURLを含む）及びGC間隔の発生の相関を与える。

【 0 0 2 6 】

負の相関値（例えば、-1に近似されるr値）は、評価されたリクエスト（例えば、所定のリクエスト）が、GCイベント226間のリクエスト数を顕著に削減するように動作し、したがって、他よりもよりメモリを消費するようになることを示している。したがって、負の相関値rを発生するサービスリクエストは、「疑わしいリクエスト（suspicious request）」と考えてもよい。疑わしいリクエストは、分析/報告モジュール252を使用して、ユーザ及び/又は管理者に報告してもよい。

【 0 0 2 7 】

一部の実施例では、疑わしいリクエストのより詳細な分析を実行して（恐らくはオフラインで）、実際のメモリ量（特定にリクエストに関連して割り当てられている）を調査することができる。分析（分析/報告モジュール252を使用して実行してもよい）は、疑わしいリクエストから1又は複数のURLを抽出するステップと、影響を受けたノードを、それをメンテナンスモードに配置することによって、負荷バランシングサービスから除外するステップと、次いで、ノード性能の詳細な分析のため、プロファイラ（profiler）モードに切り替えるステップとを含んでもよい。このとき、各クラスに対するメモリ消費は集約することができ、かつ、ソースコード（余剰メモリ消費に責任のある）は隔離するこ

10

20

30

40

50

とができる。

【0028】

一部の実施例では、相関は、サービスリクエストの選択された組み合わせに対して計算される。例えば、相関は、一部の選択された数のGC間隔INT（一緒に現れた複数のURLリクエスト：X (URL1, URL2, ...)及び(URL1, URL2, ...)を有する)を決定することができる。上記したような同じ計算プロセスを適用することができる（2以上のリクエストのグループが同時に処理される場合を除く）。そのような組み合わせの分析は、リクエストが密結合であるときか、又は、2以上のサービスリクエストがURL（ほぼ同じ量のメモリを使用する）に関連付けられるとき、有用であり、かつ、組み合わせは、他のリクエストに関連付けられたメモリ使用よりも、顕著により多くのメモリ使用を生じる。

10

【0029】

GC活動のない装置214のため（例えば、GCイベント226が生成されない）、実質的に一定の長さを有する一連の時間間隔は、GC間隔と同じ方法で取り扱うことができる。したがって、システムがガーベジコレクタを備えていない場合、同じ計算を実行して、時間の各間隔中にメモリバランスを決定することができる。他の実施例を真に理解することができる。

【0030】

例えば、上記した相関分析の他に、定量分析も実行することができる（恐らく並行して）。これは、相関分析によって与えられた相関の重要性を推定するのに役立てることができる。

20

【0031】

次の実施例（ここで、3つのリクエストタイプA（サイズ=8ユニット）、B（サイズ=4ユニット）及びC（サイズ=2ユニット）が受信され、かつ、ここで、受信したリクエストの現在の取り合わせによって占有されたメモリサイズが、20ユニット（例えば、GC（サイズ=20））であるとき、ガーベジコレクションが開始する。）を検討する。3つのGC間隔にわたって、次の数のリクエストが受信されると仮定する。

【0032】

【表1】

GC	A	B	C
1	1	1	4
2	0	1	8
3	1	2	2

30

【0033】

次の組を各行（即ち、各GC間隔）に対して計算することができる。

SIZE_MIN = このリクエストに対する最少サイズを推定する。

SIZE_AVG = 全てのリクエストが同じサイズを有すると仮定した平均値を推定する。

SIZE_MAX = 現在のリクエストを除く全てがサイズゼロであると仮定したサイズを推定する。

40

【0034】

各計算中、2つの推定（（a）各リクエストは同じ量のメモリを消費し、かつ、（b）各リクエストはメモリを消費しない。）を各リクエストに対して実行することができる。したがって、各GC間隔に対して、各リクエストを推定することができる3つの可能性のある値が存在する。

（1）分析されたリクエストはゼロメモリを消費する。

（2）間隔内の全てのリクエストは、同じサイズのメモリ消費である。又は、

（3）分析されたリクエストを除く、全てのリクエストはゼロメモリ消費である。

【0035】

（1）から、第1値を推定することができ、SIZE_MIN = 0である。（2）から、第2値

50

を推定することができ、 $SIZE_AVG = 1 / (\text{全てのリクエスト数})$ である。(3)から、第3値を推定することができ、 $SIZE_MAX = 1 / (\text{分析されたリクエストの数})$ である。この計算は、各CG間隔に対して実行することができる。

【0036】

各組の計算の後、MIN-MAX計算を使用して、前の組の値は次のように組み合わせられる。

$SIZE_MIN = \text{MAX}(\text{全てのMINのうちの})$

$SIZE_AVG = \text{MAX}(\text{全てのAVGのうちの})$ (MAXよりも小さい)

$SIZE_MAX = \text{MIN}(\text{全てのMAXのうちの})$

【0037】

したがって、この検討に対して、各リクエストの $SIZE_MIN = 0$ 及び $SIZE_AVG$ は、次のように組み合わせられる。

$\text{MAX}(\text{MAX}(\text{SIZE_AVG}(\text{実際}), \text{SIZE_AVG}(\text{組み合わせ})), \text{SIZE_MAX}(\text{組み合わせ}))$

次いで、 $SIZE_MAX$ は、 $\text{MIN}(\text{SIZE_MAX}(\text{実際}), \text{SIZE_MAX}(\text{組み合わせ}))$ として計算される。このような方法で、第1間隔に対する計算の結果は、第2間隔からの結果となる計算の結果と組み合わせられる。次いで、第3間隔に対する計算の結果は、第1及び第2間隔からの結果の組み合わせの結果と組み合わせられる。

【0038】

上記の実施例から、第1計算において、次の組が決定される。

【0039】

【表2】

	SIZE_MIN	SIZE_AVG	SIZE_MAX
A	0	1/6	1
B	0	1/6	1
C	0	1/6	1/4

【0040】

第2計算において、タイプCのリクエストに対して、これらの値は次のように決定される。

MIN = 詳細な情報なしに、Cをゼロであると推定することができる。

AVG = 全てのリクエストが同じサイズであると推定する。

MAX = 全てのリクエストが、Cを除き、ゼロであると推定する。

【0041】

【表3】

	SIZE_MIN	SIZE_AVG	SIZE_MAX
A	0	1	1
B	0	1/9	1
C	0	1/9	1/8

【0042】

次いで、第3計算において、第1及び第2計算の結果が組み合わせられる。

【0043】

10

20

30

40

【表 4】

	SIZE_MIN	SIZE_AVG	SIZE_MAX
A	0	1/6	1
B	0	1/6	1
C	0	1/8	1/8

【0044】

次いで、第4計算において、タイプCのリクエストに対して、SIZE_AVGは、 $SIZE_AVG = \text{MIN}(\text{MAX}(1/6, 1/9), 1/8) = 1/8$ として計算される。最少リクエストの値が非常に高く推定され、かつ、他の全てが非常に低く推定されるので、SIZE_AVGの値は増加しなければならない。全てのリクエストが同じ量のメモリを消費すると仮定された場合、次いで、より少ないメモリを使用したリクエストは非常に高く推定され、かつ、より多くのメモリを使用したリクエストは非常に低く推定されている。より少ないメモリを使用したリクエストが時間とともにSIZE_MAXによって減少されるが、より多くのメモリを使用したリクエストが、MAX計算を使用したSIZE_AVGによって大きくなることを、計算は許可する。このような方法で、前の仮定は時間とともに改善する。

10

【0045】

【表 5】

	SIZE_MIN	SIZE_AVG	SIZE_MAX
A	0	1/5	1
B	0	1/5	1/2
C	0	1/5	1/2

20

【0046】

第5計算において、第4計算の結果は、第3計算の結果と組み合わせられる。

【0047】

【表 6】

	SIZE_MIN	SIZE_AVG	SIZE_MAX
A	0	1/5	1
B	0	1/5	1/2
C	0	1/8	1/8

30

【0048】

その結果、各リクエストのサイズを推定することができる（より多くのGC間隔を用いて、推定はより正確になる）。

【0049】

【表 7】

	Estimated (SIZE_MIN+SIZE_MAX)/2	Size
A	0,6	0,4
B	0,35	0,2
C	0,125	0,1

40

【0050】

定量分析値結果が既知であるので、相関の定数値を除外し、かつ、重み付き相関を設定するのに有用であるか否かに関する決定を行うことができる。重み付き相関を使用して、最少のURLリクエストを除外した以降のGC間隔内のメモリ消費が既知であることを考

50

慮することができる。URLリクエストの別の発生は、所定のGCメモリ間隔に作用し、ファクタとして相関関数内に追加することができる。例えば、タイプCのリクエストがCGメモリのうちの1/10を占有すると仮定すると、計算からタイプCのリクエストを除外することは、タイプA及びBのリクエストの発生が、オリジナルのメモリスパースのうちの $(1-2*0.1 = 0.8)$ を占有することを意味する。さらに他の実施例が真に理解される。

【0051】

例えば、図3は、本発明の多様な実施例によるメモリ消費を監視するためのシステム310のブロック図である。システム310内で動作する装置314（例えば、サーバとして動作するノード）の一部の実施例は、メモリGCイベント及びノードコンテンツサービスリクエストの発生を検出するための監視モジュール344と、ノードコンテンツサービスリクエストとメモリGCイベントとの間の相関を決定するためのプロセッサ実行相関モジュール348とを備えている。装置314は、ノードコンテンツサービスリクエスト（相関（例えば、相関 $r < 0$ 又は $r < -0.5$ ）と関連した予め選択された基準を満足する）のうちの疑わしいリクエストと関連した情報を報告するための報告モジュール352も備えている。監視モジュール344と報告モジュール352とは、単一のモジュール内に一体化するか、又は、2つの別々のモジュール（図示したように）として存在してもよい。

10

【0052】

一部の実施例では、装置314は、メモリGCイベントと関連したタイミング情報を記録するためのメモリ340を備えている。したがって、多様なメモリGCイベントと関連した開始時間、終了時間、又は、両方は、メモリ340内に格納することができる。

20

【0053】

また、装置314は、装置314内に一体化されるか、又は、それに遠隔に接続されるか（図示したように、有線又は無線ネットワーク324を使用して）のどちらかのディスプレイ334を備えている。装置314に接続されたデバイス318（例えば、ノード）を使用して、ディスプレイ334を収容してもよい。したがって、ディスプレイ334及びモジュール344,348,352は、クライアント及びサーバなどの別々のデバイス（図示したように）に備えてもよい。また、ディスプレイ334及びモジュール344,348,352は、単一のデバイス（例えば、サーバ又はクライアント）又は独立したデバイス（例えば、ラップトップコンピュータ、PDA、又は、セルラー電話）内に一体化してもよい。

【0054】

ディスプレイ334を使用して、疑わしいノードコンテンツサービスリクエストに関連付けられた情報に関するアラートメッセージ374を表示することができる。このような方法で、高い相関値（例えば、 $r < -0.5$ ）を使用して、視覚的なアラームを作動させて、システム310内の過度のメモリ使用を、システム管理者に警告することができる。

30

【0055】

さらに実施例が真に理解される。これらの一部は多様な方法を含んでいる。

【0056】

例えば、図4は、本発明の多様な実施例によるメモリ消費を監視するための方法411を図示したフローチャートである。方法411は、機械アクセス可能かつ読取可能な媒体において実行され、かつ、ネットワーク内部及び間のプロセスにわたって動作可能である。ネットワークは、有線、無線、又は、有線及び無線の組み合わせであってもよい。方法411は、命令（機械によってアクセスされるとき、図1から図3に図示された要素及びプロセスに関して表示及び説明された処理を実行する）として実行してもよい。このコンテキストを仮定して、メモリ消費の監視が図4を参照して検討される。

40

【0057】

一部の実施例では、メモリ消費の監視のコンピュータ実行方法411は、ブロック421で開始し、かつ、メモリGCイベント及びノードコンテンツサービスリクエストの発生の監視を含むブロック425に継続してもよい。相関を決定するため、特定の時間間隔（GCイベント開始時間、GCイベント終了時間、又は、両方によって定義された）の間に、ノードコンテンツサービスリクエストの数が計算されるように、GCイベントのタイミングを記

50

録することは有用である。したがって、ブロック425の監視活動は、メモリGCイベントの開始時間又は終了時間のうちの少なくとも1つを記録するステップを含んでもよい。

【0058】

同様に、それらがメモリGCイベントの発生に関係しているので、ノードコンテンツサービスリクエストのタイミングを記録することが可能である。したがって、ノードコンテンツサービスリクエストのタイミングを記録することができ、かつ、間隔（ノードコンテンツサービスリクエストタイミングによって定義された）中に発生するガーベジコレクションの数を記録することができる。したがって、ブロック425の監視活動は、ノードコンテンツサービスリクエストの開始時間又は終了時間のうちの少なくとも1つを記録するステップを含んでもよい。

10

【0059】

そのようなアプローチは、プロセッサ活動がメモリ消費と関連する環境には有用である。しかし、各場合において、経過時間をプロセッサ使用時間から区別しなければならない（例えば、アイドルスレッドはメモリを消費しない）。

【0060】

ノードコンテンツサービスリクエストの存在及び/又はタイミングは、キューで記録することができる（相関分析が所望されるとき、関係する情報を周期的な抽出に利用可能にさせる）。したがって、ブロック425の監視活動は、ノードコンテンツサービスリクエストをキューで記録するステップを含んでもよい。

【0061】

ノードコンテンツサービスリクエスト内に含まれた特定の種類の情報として、URL情報も記録することができる。したがって、ブロック425の監視活動は、ノードコンテンツサービスリクエストの一部として、URL情報を記録するステップを含んでもよい。

20

【0062】

方法411は、ノードコンテンツサービスリクエストとメモリGCイベントとの間の相関を決定するステップを含むブロック429に続いてよい。上記したように、相関は、時間間隔（メモリGCイベントによって定義された）中のノードコンテンツサービスリクエストの数の間で決定してもよい。

【0063】

一部の実施例では、例えば、URLがサービスリクエストの一部を構成するとき、相関は、ノードコンテンツサービスリクエストの平均数（選択されたURLを含む）、及び、ノードコンテンツサービスリクエストの平均数（選択されたURLを含まない）を使用して決定される。したがって、ブロック429の活動は、時間間隔（メモリGCイベントによって定義された）中のノードコンテンツサービスリクエストの包含的な平均数（選択されたURLを含む）と、ノードコンテンツサービスリクエストの排他的な平均数（選択されたURLを含まない）との間の相関を決定するステップを含んでもよい。

30

【0064】

一部の実施例では、上記したように、平均は、相関を決定するための特定の方法で組み合わせてもよい。したがって、ブロック429の活動は、包含的な平均又は排他的な平均の最大値によって除算された、包含的な平均と排他的な平均との間の差異として、相関を計算するステップを含んでもよい。

40

【0065】

一部の実施例では、GCコレクション間隔は、第1GCイベントの開始時間（第2GCイベントの開始時間まで）か、又は、第1GCイベントの終了時間（第2GCイベントの終了時間まで）の間の間隔として定義することができる。したがって、時間間隔（ブロック429の相関を決定するのに使用された）は、GCイベントの開始時間及びGCイベントの終了時間のうちの1つによって定義してもよい。

【0066】

2以上の密結合されたサービスリクエストが過度のメモリ使用を発生させているか否かを決定するため、ブロック429の活動は、時間間隔（メモリGCイベントによって定義さ

50

れた)中のノードコンテンツサービスリクエストの一部の数の間の相関を決定するステップを含んでもよい(例えば、ノードコンテンツサービスリクエストの数は、複数の共通URL値に基づいている)。

【0067】

方法411は、予め選択された基準(ノードコンテンツサービスリクエストとメモリGCイベントとの間の相関が、ノードコンテンツサービスリクエストのうちの疑わしいリクエストの追加的な分析をトリガするのに十分であることを示す)と関連した閾値を超えていることを検出するステップを含むブロック433に続いてよい。例えば、相関が所定の閾値(例えば、 $r < -0.5$)をクロスするとき、疑わしいノードコンテンツサービスリクエストの追加的な分析を開始することができる。

10

【0068】

したがって、ブロック433で疑わしいリクエストが見つからない場合、次いで、方法411は、ブロック425に戻り、CGイベント及び入ってくるノードコンテンツサービスリクエストの監視を継続してもよい。そうでなければ、相関決定と関連した予め選択された基準の満足により、疑わしいリクエストであるとリクエストが識別された場合、方法411は、疑わしいノードコンテンツサービスリクエストと関連した情報(例えば、1又は複数のURL)を報告するステップを含むブロック437に続いてよい。ブロック437の報告するステップの活動は、GUIの一部として、相関又は情報(ノードコンテンツサービスリクエストのうちの疑わしいリクエストと関連した)のうちの少なくとも1つを表示するステップをさらに含んでもよい。

20

【0069】

このような方法で、GCイベント及び入ってくるリクエストを監視することによって、あるサーバコンテンツサービスリクエストに関係した疑わしいメモリ消費活動を、システム性能に対して比較的小さな影響で、認識及び報告することができる。さらなる実施例が真に理解される。

【0070】

例えば、図5は、本発明の多様な実施例によるメモリ消費を監視するための方法511を図示したフローチャートである。方法511は、機械アクセス可能かつ読取可能な媒体において実行され、かつ、ネットワーク内部及び間のプロセスにわたって動作可能である。ネットワークは、有線、無線、又は、有線及び無線の組み合わせであってもよい。方法511は、命令(機械によってアクセスされる時、図1から図4に図示された要素及びプロセスに関して表示及び説明された処理を実行する)として実行してもよい。このコンテキストを仮定して、メモリ消費の監視が図5を参照して検討される。

30

【0071】

一部の実施例では、メモリ消費を監視するコンピュータ実行方法511は、ブロック521で開始し、かつ、メモリGCイベントとノードコンテンツサービスリクエスト(URL情報を含む)との間のタイミング関係を監視するステップを含むブロック525に続く。

【0072】

方法511は、ノードコンテンツサービスリクエストとメモリGCイベントとの間の相関を決定するステップを含むブロック529に続いてよい。例えば、相関は、時間間隔(GCイベントによって定義された)中のノードコンテンツサービスリクエストの平均を使用して決定することができる。したがって、ブロック529の活動は、選択された数の間隔(メモリGCイベントによって定義された)中のノードコンテンツサービスリクエストの包含的な平均数と、選択された数の間隔(メモリGCイベントによって定義された)中のノードコンテンツサービスリクエストの排他的な平均数との間の相関を決定するステップを含んでもよい。

40

【0073】

上記したように、リクエストの組み合わせからのURLを相関分析に使用して、選択されたURLが共に動作して、大量のメモリを消費するか否かを決定することができる。したがって、ブロック529の活動は、複数のノードコンテンツサービスリクエストと関連し

50

たURL情報を組み合わせて、複数のノードコンテンツサービスリクエストのうちの個別のリクエスト間の相関を決定するステップを含んでもよい。

【0074】

方法511は、ブロック533に続き、疑わしいリクエストが識別されたか否かを決定してもよい。そのようなリクエストが見つけれなかった場合、次いで、方法511は、ブロック525に戻り、GCイベント及びノードアクセスリクエストの監視を継続してもよい。そうでなければ、リクエストが相関と関連した予め選択された基準を満足すると識別された場合、疑わしいリクエストとして特徴付けるため、次いで、方法511は、疑わしいノードコンテンツサービスリクエストと関連した情報を報告するステップを含むブロック537に続いてよい。したがって、一部の実施例では、方法511は、リクエスト（URLを含むか、又は、含まない）の発生よりもむしろ、GCタイミング関係（特にURLの使用を含む）を監視するように動作してもよい。

10

【0075】

疑わしいリクエストと関連した情報は、ディスプレイ表面（例えば、フラットパネルディスプレイ）上に、インターネットブラウザ又はパナーを使用して表示することができる。したがって、ブロック537の報告する活動は、GUI（例えば、図3のアラートメッセージ374を参照）の一部として、ウェブページウィンドウ又はサイドバーウィンドウ又は両方を表示するステップを含んでもよい。

【0076】

方法511は、ノードコンテンツサービスリクエストがノードによってもはや処理されないように、ネットワークノードを命令して、維持モードに入るステップを含むブロック541に続いてよい。このような方法で、疑わしいリクエストを受信するノードは、維持モードを使用したさらなる分析のために、ネットワークを外すことができる。ブロック541の活動の一部として、方法511は、ノードを命令して、プロファイラモードに入るステップを含んでもよい。

20

【0077】

選択された回数で、GCイベントを停止して、ノードコンテンツサービスリクエストに回答して、ふるまいをより詳細に分析してもよい。例えば、JVMの実施例では、Java（登録商標）の方法「System.gc」を無効化することができる（Java（登録商標）仮想マシン（JVM）ツールインターフェース（TI）（Jvmti->ForceGarbageCollection）を使用して、GCをまだトリガすることができるが）。したがって、方法511は、ブロック545で、メモリGCイベントを選択的に無効化するステップを含んでもよい。

30

【0078】

一部の実施例では、疑わしいリクエストの結果として使用されたメモリは、プロファイラモードにおいて決定することができ、かつ、GCイベントによって格納されたメモリも、このモードにおいて決定することができる。したがって、方法511は、ブロック549に進み、1又は複数の疑わしいノードコンテンツサービスリクエストに、ノード内で、割り当てられた第1メモリ量を決定するステップと、メモリGCイベントによって割り当てを解除された（deallocated）第2メモリ量を決定するステップとを含んでもよい。

【0079】

割り当てられたメモリが格納されたメモリよりも一貫して大きい場合、次いで、メモリリーク、又は、時間とともに一貫したメモリの損失（調査中である疑わしいノードコンテンツサービスリクエストによる）が存在する可能性がある。したがって、方法511は、ブロック553で、第1メモリ量と第2メモリ量とを比較して、メモリリークが存在する否かを決定するステップを含んでもよい。次いで、方法511は、ブロック557で終了してもよい。

40

【0080】

図5に示された各方法の要素は、図4に示された方法の要素のいずれかに追加するか、又は、置き換えてもよいことが当業者には真に理解される。加えて、図4から図5の各方法の要素は、多様な方法で組み合わせて、直列、並列、ループ、及び/又は、繰り返し方

50

法で、各図からの要素を使用する多様な方法を構成してもよいことも当業者には真に理解される。

【0081】

図6は、本発明の多様な実施例によるコンピュータシステム600の代表的な形態における機械602のブロック図である。コンピュータシステム600は、本明細書において検討された1又は複数の活動(図4から5に図示された活動を含む)を、システム600に実行させるための1組の命令612を含んでもよい。システム600は、図3のシステム310に類似している又は同一であるか、又は、その要素のうちのいずれか(例えば、装置314又はデバイス318)であつてもよい。

【0082】

図6では、機械602のコンポーネントは、メインメモリ620、静的又は不揮発性メモリ624、大容量記憶装置606を含んでもよいことが分かる。プロセッサ604に接続された他のコンポーネントは、出力デバイス628(ビデオディスプレイなど)、入力デバイス632(キーボードなど)、カーソル制御デバイス636(マウスなど)を含んでもよい。また、ネットワークインターフェースデバイス640(ネットワーク644に、プロセッサ604及び他のコンポーネントを接続する)は、バス616に接続してもよい。

【0083】

プロセッサ604、メモリ620,624、及び、記憶装置606は、命令612(実行されるとき、機械602に、本明細書に記載された1又は複数の方法を実行させる)をそれぞれ含んでもよい。命令612は、多数の周知の転送プロトコル(例えば、ハイパーテキスト転送プロトコル)のうちのいずれか1つを利用して、ネットワークインターフェースデバイス640を介して、ネットワーク644上にさらに転送するか、又は、受信してもよい。バス616に接続されたこれらの要素のうちのいずれかは、実現される固有の実施例によって、不在、単一で存在、又は、複数の数で存在してもよい。

【0084】

一部の実施例では、機械602は、スタンドアロンデバイスとして動作するか、又は、他の機械に接続してもよい(例えば、ネットワーク化)。ネットワーク化された環境では、機械602は、ノード(例えば、サーバ-クライアントネットワーク環境におけるサーバ又はクライアントデバイス、又は、ピアツーピア(又は分散した)環境におけるピアマシン)の機能で動作してもよい。機械602は、パーソナルコンピュータ(PC)、タブレットPC、セットトップボックス(STB)、PDA、セルラー電話、ウェブアプライアンス、ネットワークルータ、スイッチ又はブリッジ、その機械によって行われるアクションを指定する1組の命令(シーケンシャルか又はそうではないもの)を実行可能な機械を含んでもよい。さらに、単一の機械602だけが図示されているが、用語「機械」は、個別に、又は、共同で1組(又は複数の組)の命令を実行して、本明細書で検討された1又は複数の方法を実行する機械の集合を含んで利用しなければならない。

【0085】

機械可読媒体608は、単一の媒体として示されるが、用語「機械可読媒体」は、1又は複数の命令612を格納する単一媒体又は複数の媒体(例えば、集中又は分散データベース、及び/又は、関係するキャッシュ及びサーバ、プロセッサ604レジスタ、メモリ620,624及び記憶装置606などの多様な記憶媒体)を含んで利用しなければならない。用語「機械可読媒体」は、機械によって実行するための1組の命令を格納、コード化又はキャリアが可能であり、かつ、機械602に、本発明の1又は複数の方法のうちのいずれかを実行させ、又は、そのような1組の命令によって利用されたか、又は、関連付けられたデータ構造を格納、コード化又はキャリアが可能である媒体を含むように利用しなければならない。したがって、用語「機械可読媒体」又は「コンピュータ可読媒体」は、固体記憶装置及び光学及び磁気媒体などの有形の媒体を含んで利用しなければならない。

【0086】

最後に、上記したように、メモリ消費が相関分析を使用して監視されるとき、システム性能に及ぶ影響は比較的小さいことが分かる。したがって、多くのインスタンスでは、分

10

20

30

40

50

析結果が迅速（恐らくは、メモリ消費状態（そうでなければ、システム動作の崩壊となる恐れのある）を修復するのに間に合う）に利用可能であるので、全システム性能は、これらの解決策（メモリ動作とより直接的に相互作用する）によって、向上させることができる。

【 0 0 8 7 】

本発明の実施例は、多様なアーキテクチャのプラットフォーム、オペレーティング及びサーバシステム、デバイス、システム、又は、アプリケーションにおいて、実行することができる。本明細書に示された特定のアーキテクチャのレイアウト又は実装は、説明及び理解のためだけに、このように提供され、かつ、多様な実施例を限定することを意図していない。

10

【 0 0 8 8 】

要約は、37C.F.R.1.72(b)に従うように提供され、かつ、読者が、技術的な開示の性質を迅速に確認することを可能にする。それは、特許請求の範囲の範囲又は意味を、解釈又は限定することに使用されないという理解の下に提出されている。

【 0 0 8 9 】

この発明の詳細な説明では、多数の特徴が、説明を合理化する目的で、単一の実施例と一緒にグループ化されている。説明のこの方法は、包含（請求された実施例が各請求項に明白に列挙されたよりも多くの特徴を有する）として解釈されてはならない。むしろ、特許請求の範囲が反映するので、発明の主題は、単一の開示された実施例の全ての特徴には決してない。したがって、特許請求の範囲は、別々の実施例としてそれら自体に基づく各請求項を備えて、これによって、発明の詳細な説明に組み込まれている。

20

【 符号の説明 】

【 0 0 9 0 】

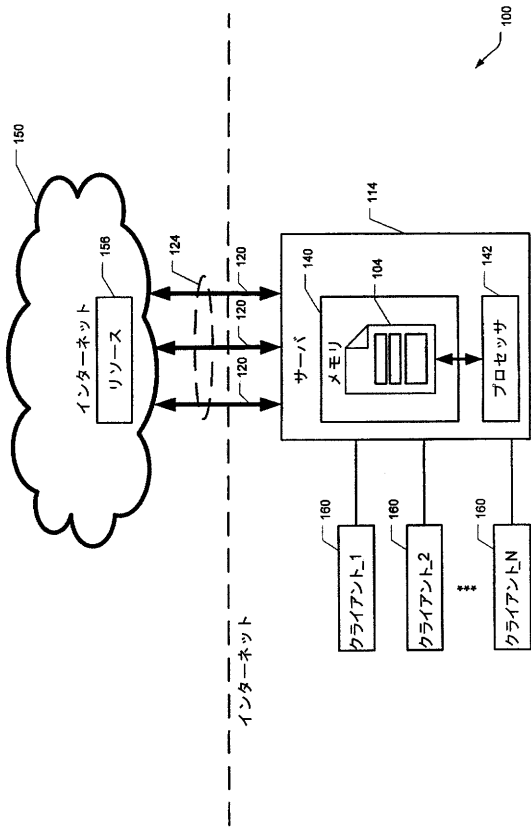
- 100 プロセス
- 104 アプリケーション
- 114 サーバ
- 120 サービスリクエスト
- 124 ネットワーク
- 140 メモリ
- 142 プロセッサ
- 150 インターネット
- 156 リソース
- 160 クライアント
- 214 装置
- 216 プロセッサ及びメモリ
- 226 GC イベント
- 228 入ってくるサービスリクエスト
- 244 監視モジュール
- 248 相関モジュール
- 252 分析 / 報告モジュール
- 310 システム
- 314 装置
- 318 デバイス
- 340 メモリ
- 342 プロセッサ
- 344 監視モジュール
- 348 相関モジュール
- 352 報告モジュール
- 334 ディスプレイ
- 374 アラートメッセージ

30

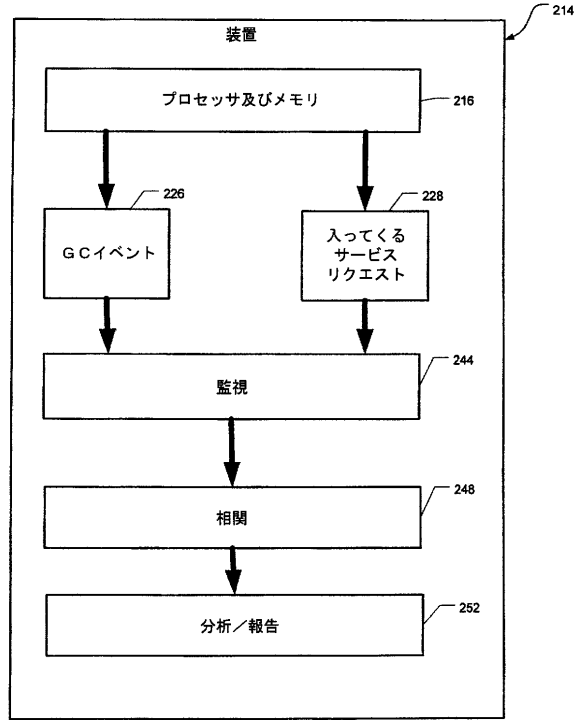
40

50

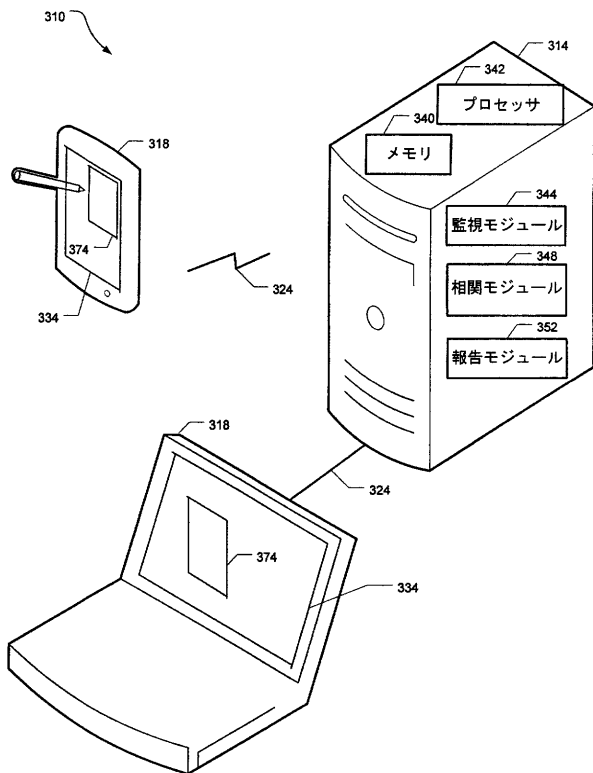
【図1】



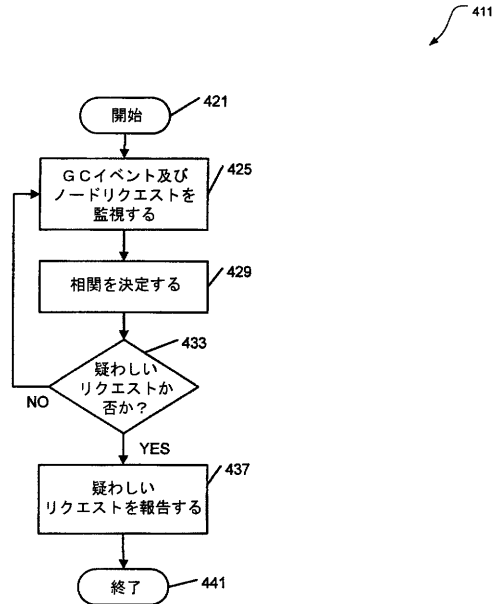
【図2】



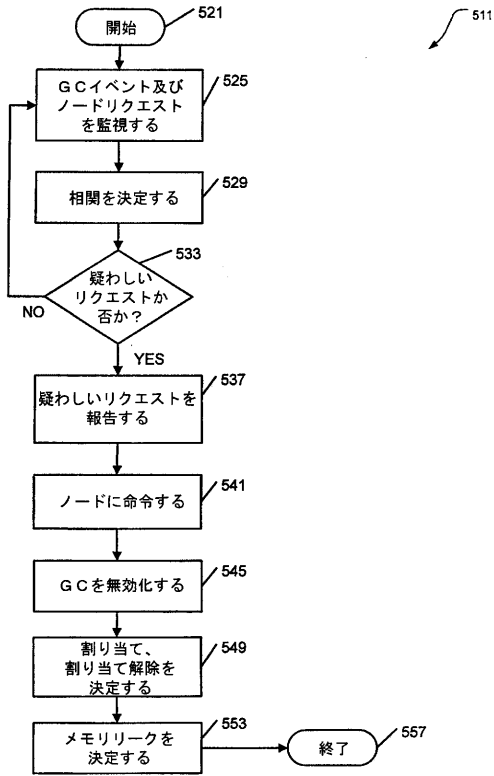
【図3】



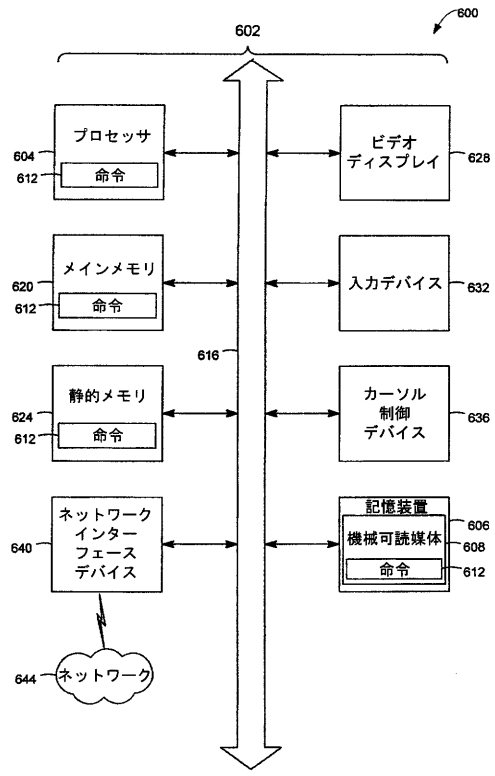
【図4】



【図5】



【図6】



フロントページの続き

(74)代理人 100110364

弁理士 実広 信哉

(72)発明者 アルベルト・ゼドリッツ

ドイツ・69168・ヴィースロッホ・ミュールシュトラッセ・44

(72)発明者 ジャン・バティスト・ドゥディア

ドイツ・68165・マンハイム・ヴィントミュールシュトラッセ・2 - 4

Fターム(参考) 5B042 GA33 GC08 JJ44 MA08 MA14 MB01 MC07 MC09 MC22 MC23

MC40