

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6064571号
(P6064571)

(45) 発行日 平成29年1月25日(2017.1.25)

(24) 登録日 平成29年1月6日(2017.1.6)

(51) Int.Cl.

F 1

GO6F 11/36	(2006.01)	GO6F 11/36	1 6 0
GO6F 11/34	(2006.01)	GO6F 11/34	1 0 9
GO6F 11/07	(2006.01)	GO6F 11/34	1 1 9
		GO6F 11/07	1 9 0
		GO6F 11/07	1 4 0 E

請求項の数 6 (全 18 頁)

(21) 出願番号

特願2012-269416 (P2012-269416)

(22) 出願日

平成24年12月10日 (2012.12.10)

(65) 公開番号

特開2014-115826 (P2014-115826A)

(43) 公開日

平成26年6月26日 (2014.6.26)

審査請求日

平成27年8月4日 (2015.8.4)

(73) 特許権者 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(74) 代理人 100105142

弁理士 下田 憲次

(72) 発明者 武田 俊男

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 荏原 陽一

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 山崎 純一

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 处理プログラム、処理方法および処理装置

(57) 【特許請求の範囲】

【請求項 1】

コンピュータに、

アプリケーションサーバの処理状況を示す処理情報と、記憶手段に記憶された構成情報に基づいて特定される前記アプリケーションサーバに含まれる複数のコンポーネントそれぞれの動作状況を示す動作情報を収集し、

収集された前記処理情報に基づいて前記アプリケーションサーバの処理時間の遅延を検知したときに、収集された前記処理情報と前記動作情報に基づき、前記複数のコンポーネントそれぞれについて、前記処理状況とコンポーネントの動作状況との相関の有無を判定し、

判定した結果に基づいて、前記複数のコンポーネントの中から、動作状況が前記処理状況と相関を有するコンポーネントを抽出し、

前記複数のコンポーネントの中から抽出された、相関を有する前記コンポーネントと関連する関連コンポーネントを、前記構成情報に基づいて特定し、

前記関連コンポーネントの動作状況を示す情報をさらに収集し、

収集された前記処理情報と、前記関連コンポーネントの動作状況を示す前記動作情報と、に基づき、動作状況が前記処理状況と相関を有すると判定された関連コンポーネントを抽出する、

ことを実行させることを特徴とする処理プログラム。

【請求項 2】

前記処理プログラムは、

前記コンピュータに、

前記アプリケーションサーバがアクセスする外部リソースの動作情報をさらに収集し、収集された前記処理情報と、外部リソースの動作情報と、に基づき、前記処理状況と前記外部リソースの動作状況との相関の有無を判定し、

動作状況が前記処理状況と相関を有する外部リソースを抽出する、

ことを実行させることを特徴とする請求項1記載の処理プログラム。

【請求項3】

前記処理情報は、所定時間間隔で収集された前記アプリケーションサーバにより実行されるアプリケーションの処理時間を含み、

前記複数のコンポーネントの動作情報、および前記関連コンポーネントの動作情報は、所定時間間隔で収集されたコンポーネントそれぞれの処理時間を含む、

ことを特徴とする請求項1記載の処理プログラム。

【請求項4】

前記処理プログラムは、

前記コンピュータに、

前記複数のコンポーネントについて、コンポーネントの処理状況であって、処理時間とは異なる第2の動作情報をさらに収集し、

前記関連コンポーネントが特定されなかった場合、前記処理情報と前記第2の動作情報とにに基づいて、前記処理状況と前記第2の動作情報が収集されたコンポーネントの動作状況との相関の有無を判定し、

前記処理状況と前記第2の動作情報が収集されたコンポーネントの動作状況との相関があると判定された場合、処理状況が前記処理状況との相関があると判定された、前記第2の動作情報が収集されたコンポーネントを抽出する、

ことを実行させることを特徴とする請求項1記載の処理プログラム。

【請求項5】

コンピュータが、

アプリケーションサーバの処理状況を示す処理情報と、記憶手段に記憶された構成情報に基づいて特定される前記アプリケーションサーバに含まれる複数のコンポーネントそれぞれの動作状況を示す動作情報を収集し、

収集された前記処理情報に基づいて前記アプリケーションサーバの処理時間の遅延を検知したときに、収集された前記処理情報と前記動作情報に基づき、前記複数のコンポーネントそれぞれについて、前記処理状況とコンポーネントの動作状況との相関の有無を判定し、

判定した結果に基づいて、前記複数のコンポーネントの中から、動作状況が前記処理状況と相関を有するコンポーネントを抽出し、

前記複数のコンポーネントの中から抽出された、相関を有する前記コンポーネントと関連する関連コンポーネントを、前記構成情報に基づいて特定し、

前記関連コンポーネントの動作状況を示す情報をさらに収集し、

収集された前記処理情報と、前記関連コンポーネントの動作状況を示す前記動作情報と、に基づき、動作状況が前記処理状況と相関を有すると判定された関連コンポーネントを抽出する、

ことを特徴とする処理方法。

【請求項6】

アプリケーションサーバの構成情報を記憶する記憶手段と、

アプリケーションサーバの処理状況を示す処理情報と、前記記憶手段に記憶された構成情報に基づいて特定される前記アプリケーションサーバに含まれる複数のコンポーネントそれぞれの動作状況を示す動作情報を収集する収集手段と、

収集された前記処理情報に基づいて前記アプリケーションサーバの処理時間の遅延を検知したときに、収集された前記処理情報と前記動作情報に基づき、前記複数のコンポーネ

10

20

30

40

50

ントそれぞれについて、前記処理状況とコンポーネントの動作状況との相関の有無を判定する判定手段と、

判定した結果に基づいて、前記複数のコンポーネントの中から、動作状況が前記処理状況と相関を有するコンポーネントを抽出する抽出手段と、

前記特定手段はさらに、前記複数のコンポーネントの中から抽出された、相関を有する前記コンポーネントと関連する関連コンポーネントを、前記構成情報に基づいて特定し、

前記収集手段はさらに、前記関連コンポーネントの動作状況を示す情報をさらに収集し

前記抽出手段はさらに、収集された前記処理情報と、前記関連コンポーネントの動作状況を示す前記動作情報と、に基づき、動作状況が前記処理状況と相関を有すると判定された関連コンポーネントを抽出する、

ことを特徴とする処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理システムに適用する処理プログラムに関する。

【背景技術】

【0002】

クライアント装置がサーバ装置にアプリケーションの実行を要求し、サーバ装置が、クライアント装置からの要求に応じてアプリケーションを実行し、結果をクライアント装置に送信する技術が存在する。このような技術に用いられるサーバ装置は、例えば、アプリケーションサーバと呼ばれる。アプリケーションサーバは、情報処理システムの中で複数台用いられることがある。

【0003】

アプリケーションの実行時には、種々の原因によって、サーバ装置がクライアント装置から要求を受けてから、クライアント装置に実行結果を送信するまでの処理に遅延が発生することがある。処理の遅延を解消し、情報処理システムを安定して運用させるためには、遅延の原因を特定し、システム管理者などによる迅速な対処が求められる。このような事象に対し、従来、アプリケーションの実行に用いられる複数のアプリケーションサーバの内、遅延の原因となるアプリケーションサーバを特定し、情報処理システムに発生した遅延の解消が解消されるまでの時間を短縮する技術が存在する。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特開2011-258057号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

アプリケーションサーバが実行する処理には、複数の実行単位（プロセス）、また実行単位ごとの工程（コンポーネント）が存在する。したがって、遅延解消のためには、例えば、アプリケーションサーバの処理の遅延の原因がどのコンポーネントであるかを特定し、遅延原因と特定されたコンポーネントに対して対処を施す必要がある。

【0006】

しかしながら、従来技術では、処理に遅延が発生しているアプリケーションサーバを特定することはできるものの、特定されたアプリケーションサーバにおいてどのコンポーネントが遅延の原因であるかを特定することができない。したがって、どのコンポーネントが遅延の原因であるかは、例えばシステム管理者の手作業により特定する必要があるが、このため、遅延解消までの時間が長くなってしまうおそれがある。特に、アプリケーションサーバで実行されるアプリケーションやアプリケーション内のコンポーネントの数が多くなると、手作業がより煩雑になるため、遅延解消までの時間が長くなってしまう。

10

20

30

40

50

【0007】

1つの側面では、本発明は、アプリケーションサーバにより実行されるアプリケーションの動作遅延の原因を迅速に特定し、情報処理システムに発生した遅延が解消されるまでの時間を短縮することを目的とする。

【課題を解決するための手段】**【0008】**

1つの案では、コンピュータに、アプリケーションサーバの処理状況を示す処理情報と、記憶手段に記憶された構成情報に基づいて特定される前記アプリケーションサーバに含まれる複数のコンポーネントそれぞれの動作状況を示す動作情報を収集し、収集された前記処理情報に基づいて前記アプリケーションサーバの処理時間の遅延を検知したときに、収集された前記処理情報と前記動作情報に基づき、前記複数のコンポーネントそれぞれについて、前記処理状況とコンポーネントの動作状況との相関の有無を判定し、判定した結果に基づいて、前記複数のコンポーネントの中から、動作状況が前記処理状況と相関を有するコンポーネントを抽出し、前記複数のコンポーネントの中から抽出された、相関を有する前記コンポーネントと関連する関連コンポーネントを、前記構成情報に基づいて特定し、前記関連コンポーネントの動作状況を示す情報をさらに収集し、収集された前記処理情報と、前記関連コンポーネントの動作状況を示す前記動作情報と、に基づき、動作状況が前記処理状況と相関を有すると判定された関連コンポーネントを抽出することを実行させる。

【発明の効果】**【0009】**

本発明の1つの態様によれば、アプリケーションの処理の遅延の検知した場合に、複数の要素の中から、動作状況が前記処理時間と相関を有する要素が抽出される。したがって、例えば、アプリケーションの動作遅延の原因となる要素を特定することができる。

【図面の簡単な説明】**【0010】**

【図1】情報処理システム100を示す。

【図2】管理サーバ400のハードウェア構成の一例を示す。

【図3】管理サーバ400の機能ブロック図を示す。

【図4】構成情報記憶部423が記憶する情報の一例を示す。

【図5】アプリケーションサーバ200の定義情報の一例を示す(1/2)。

【図6】アプリケーションサーバ200の定義情報の一例を示す(2/2)。

【図7】遅延原因を特定する処理対象のアプリケーションの態様を示す。

【図8】性能情報1100を示す。

【図9】閾値情報1200を示す。

【図10】管理サーバ400が実行する、遅延原因を特定する処理のフローチャートを示す(1/2)。

【図11】管理サーバ400が実行する、遅延原因を特定する処理のフローチャートを示す(2/2)。

【図12】相関関数を算出する式を示す。

【図13】スレッドプールのキュー格納数の推移を示す情報の一例を示す。

【図14】解析部414により正規化が実行された後の、応答時間とキュー格納数を示す。

【発明を実施するための最良の形態】**【0011】**

本発明を実施する実施例について、図面とともに説明する。

【0012】

図1は、本実施例の情報処理システム100を示す図である。情報処理システム100は、アプリケーションサーバ200、データベース300、管理サーバ400、クライアント装置500、管理端末600を備える。アプリケーションサーバ200、データベー

10

20

30

40

50

ス300、管理サーバ400、クライアント装置500、管理端末600は、例えばLAN(Lo cal Area Network)などを用いたネットワーク700を介して、相互に通信可能に接続される。アプリケーションサーバ200、データベース300、管理サーバ400のそれぞれは、例えばサーバ装置を用いて実現することが可能である。また、クライアント装置500、管理端末600には、例えば、PC(Personal Computer)を用いて良い。

【0013】

アプリケーションサーバ200は、情報処理システム100においてアプリケーションを実行する装置である。データベース300は、アプリケーションサーバ200が実行するアプリケーションにおいて利用されるデータを格納する装置である。管理サーバ400は、アプリケーションサーバ200、およびデータベース300の管理、制御を行なう装置である。クライアント装置500は、例えば、アプリケーションサーバ200が実行するアプリケーションの実行結果が出力される装置である。管理端末600は、管理サーバ105が実行する処理の結果が出力される装置である。

【0014】

図2は、管理サーバ400のハードウェア構成の一例を示す図である。管理サーバ400は、CPU(Central Processing Unit)401、RAM(Random Access Memory)402、HDD(Hard Disk Drive)403、ROM(Read Only Memory)ドライブ404、アダプタ406を備え、それぞれはバスによって相互に通信可能に接続されている。CPU401は、管理サーバ400が実行する演算処理等を実行する。RAM402は、例えば、CPU401が実行する演算処理に用いられるデータやプログラムがロードされる。HDD403は、サーバ装置101が実行する処理に用いられるデータが記憶される。また、HDD403は、CPU401が実行するプログラムを記憶することが可能である。尚、HDD403は、例えば半導体記憶装置や、磁気テープ記憶装置などの、HDD以外の記憶装置であっても良い。ROM(Read Only Memory)ドライブ404は、例えばディスクROM405に記憶された情報を読み込み、RAM402にロードすることが可能な読み取り装置である。ディスクROM405は、管理サーバ400が実行するプログラムを記憶することが可能である。したがって、CPU401は、ディスクROM405に記憶されたプログラムを、ドライブ404を介してRAM402にロードし、ロードされたプログラムを実行することも可能である。アダプタ406は、サーバ装置をネットワーク110に接続するための接続機器である。例えば、ネットワーク110がLANである場合、アダプタ406としてLANアダプタを用いれば良い。

【0015】

尚、アプリケーションサーバ200およびデータベース300にサーバ装置を用いる場合、アプリケーションサーバ200およびデータベース300のハードウェア構成は図2に示す管理サーバ400のハードウェア構成と同様であって良い。

【0016】

図3は、管理サーバ400の機能ブロック図である。管理サーバ400は、機能部として、通信部411、収集部412、検出部413、解析部414、判定部415、構成管理部416を有する。

【0017】

通信部411は、ネットワーク700を介して、アプリケーションサーバ400、データベース300、クライアント装置500、管理端末600とのデータの送受信を実行する。

【0018】

収集部412は、アプリケーションの実行中に、アプリケーションサーバ200、データベース300の性能情報を収集する。性能情報については詳細を後述する。

【0019】

検出部413は、アプリケーションのクライアント装置500に対する応答時間の情報

10

20

30

30

40

50

を収集し、応答時間の遅延を検出する。

【0020】

解析部414は、検出部413によって応答時間の遅延が検出されると、後述する性能情報記憶部422および構成情報記憶部423に記憶された情報に基づいて、～～

判定部415は、アプリケーションの各要素について、クライアント装置500への応答時間と相関があるか否かを判定する。また、判定部415は、クライアント装置500への応答時間と相関がある要素を、応答時間の遅延の原因として抽出する。

【0021】

構成管理部416は、アプリケーションの実行に関わるコンポーネントおよびコンポーネント同士の接続関係を示す情報を管理する。構成管理部416が管理する情報は後述する構成情報記憶部423に記憶される。

10

【0022】

収集部412、検出部413、解析部414、判定部415、構成管理部416のそれぞれは、例えば、図2に示すCPU401が、HDD403に記憶されたプログラムを実行することにより実現されても良い。その際、HDD403に記憶されたプログラムが、RAM402にロードされることとしても良い。また、通信部411として、例えば図2に示すアダプタ406のような接続機器を用いても良い。

【0023】

また、管理サーバ400は、閾値情報記憶部421、性能情報記憶部422、構成情報記憶部423を有する。

20

【0024】

閾値情報記憶部421は、クライアント装置500への応答時間の遅延の基準となる閾値を特定する情報を記憶する。

【0025】

性能情報記憶部422は、アプリケーションの実行中に、収集部412が収集したアプリケーションサーバ200、データベース300の性能情報を記憶する。

【0026】

構成情報記憶部423は、アプリケーションサーバ200により実行されるアプリケーションが有する要素および要素の依存関係を特定する情報を記憶する。

30

【0027】

閾値情報記憶部421、性能情報記憶部422、構成情報記憶部423は、例えば、図2に示すHDD403やRAM402を用いて実現されても良い。

〔定義情報〕

図4は、構成情報記憶部423が記憶する情報の一例を示す図である。構成情報記憶部423には、図4に示すような、アプリケーションサーバ200で実行される処理に用いられる、コンポーネントや、アプリケーションの関係を特定する構成情報800が記憶される。例えば、図4では、構成情報800からvirtual-server211、network-listener212、213、thread-pool214、Webアプリケーション215、216、j dbc-resource223、j dbc-connection-pool224、というコンポーネントが含まれることが特定される。また、

40

Webアプリケーションは215、virtual-server211、network-listener212（または213）、thread-pool214を介してクライアント装置500からの要求に応じた処理を実行する。Webアプリケーション215は処理実行時に、必要に応じてデータベース300にアクセスし、j dbc-resource223、j dbc-connection-pool224は、Webアプリケーション215とデータベース300との接続を制御する。

図5および図6は、アプリケーションサーバ200の定義情報の一例を示す図である。図5および図6に示す定義情報900は、アプリケーションサーバ200で実行されるア

50

プリケーション、およびコンポーネントが定義された情報である。尚、図5および図6では定義情報の一部を抜粋して記載している。図4に示したアプリケーションの構成は、例えば、図3の構成管理部423が、定義情報900に基づいて特定することができる。

【0028】

例えば、図5の情報901では、コンポーネントとして、jdbc-resource223、jdbc-connection-pool224、を含むことを定義している。

【0029】

また、例えば、図6の情報904では、http-listener212、213の後段にhttp-thread-pool214が接続されるという、コンポーネントの接続様式を定義している。

10

【0030】

また、例えば、図6の情報906では、virtual-server211の後段に、network-listener212、213が接続されるという、コンポーネントの接続様式を定義している。

図5および図6に示す定義情報900に基づいて、図4のように、アプリケーションサーバ200で実行されるアプリケーション、コンポーネント、およびそれらの接続関係を特定することができる。

【0031】

図7は、本実施例の処理の説明のための、遅延原因を特定する処理対象のアプリケーションの構造を示した図である。図7に示すようなアプリケーションの構造は、例えば、構成情報記憶部423が記憶する情報から特定される、図7に示すように、本実施例においてアプリケーションサーバ200が実行するアプリケーションは、要素A（Webサーバ）1001、要素B（コンポーネントA）1002、要素C（アプリケーション）1003、要素D（コンポーネントB）1004、要素E（外部リソース）1005により実現されるアプリケーションである。また、要素B（コンポーネントA）1002、要素C（アプリケーション）1003、要素D（コンポーネントB）1004は、Java（登録商標）仮想マシン1006（図5では、要素Zと示す）を用いて実現される。尚、要素B（コンポーネントA）1002～要素D（コンポーネントB）1004のそれぞれは、例えば、前述した構成情報800に含まれるアプリケーションやコンポーネントである。

20

【0032】

要素A（Webサーバ）1001は、クライアント装置100からの要求を受信し、要求の内容を要素B（コンポーネントA）1002に送信する。ウェブサーバ1001から要求を受信した要素B（コンポーネントA）1002は、要素C（アプリケーション）1003に要求を実行する。要素C（アプリケーション）1003は要求に基づく処理を実行するため、処理対象のデータを、要素D（コンポーネントB）1004を介して、要素E（外部リソース）1005から抽出する。ここで、外部リソースとは、例えば、アプリケーションの実行に用いられる情報を記憶するデータベースである。処理対象のデータを要素E（外部リソース）1005から抽出後、要素C（アプリケーション）1003は処理対象のデータに処理を実行し、実行結果を、要素B（コンポーネントA）1002および要素A（Webサーバ）1001を介して、クライアント装置に送信する。この一連の処理の開始から終了までの時間を、本明細書においては「応答時間」と記載する。

30

〔性能情報の収集〕

アプリケーションサーバ200が、アプリケーションを実行する際に、管理サーバ400の収集部412は、通信部411を介して、性能情報を取得し、性能情報記憶部422に記憶する。ここで性能情報とは、アプリケーションサーバ200が実行する処理に関する情報、およびアプリケーションの要素それが実行する処理に関する情報である。アプリケーションサーバ200が実行する処理に関する情報は、具体的には、例えば、アプリケーションサーバ200が、クライアント装置100からの要求を受けてから、要求に対する応答をクライアント装置100に送信するまでの時間、つまり前述した応答時間で

40

50

ある。また、アプリケーションの要素それが実行する処理に関する情報は、例えば、要素それが処理に要する時間である。以後、明細書においては、要素それが処理に要する時間を「処理時間」と記載する。

【0033】

図8は、収集部412が収集し、性能情報記憶部422に記憶する情報の一例を示す図である。図8の性能情報1100は、応答時間と、アプリケーション内の各要素の処理時間を記憶した情報である。例えばアプリケーションに、図5のように要素A1001、要素B1002、要素C1003、要素D1004、要素E1005が含まれる場合、性能情報1100は、図8に示すように、取得時刻1101、応答時間1102、要素Aの処理時間1103、要素Bの処理時間1104、要素Cの処理時間1105、要素Dの処理時間1106、要素Eの処理時間1107、要素Zの処理時間1108を含む。
10

【0034】

収集部412は、アプリケーションサーバ200が、クライアント装置からの要求に応じてアプリケーションを実行する際に、応答時間と、各要素の処理時間とを定期的に（例えば、1分に1回）収集し、収集した値を性能情報記憶部422に記憶する。

【0035】

収集部412が実行する処理により、性能情報記憶部422には、図8に示すような応答時間、処理時間が記憶される。

【0036】

また、収集部412は、応答時間や、処理時間以外の性能情報を収集することとしても良い。具体的には、例えば、スレッドプールのキューに格納された作業項目の数（キュー数）や、JDBC（Java（登録商標）Database Connectivity）プールの接続待ちの数（接続待ち数）などが挙げられる。尚、スレッドプールとは、クライアント装置500から受け付けた要求をキューに格納するコンポーネントであり、例えば、図4のhttp-thread-pool214に相当する。また、JDBCプールとは、Java（登録商標）仮想マシンと、データベースとの接続を制御するコンポーネントであり、例えば、図4のjdbcc-connection-pool224に相当する。
20

〔遅延検出の閾値〕

図9は、閾値情報記憶部421が記憶する情報の一例を示す図である。図9に示す閾値情報1200は、検出部413が、処理時間の遅延を検出する基準となる値が記憶されている。例えば、閾値情報1200には、応答時間の閾値が3000ミリ秒であることを示す。したがって、検出部413は、収集部412が取得した応答時間が3000ミリ秒を超えた場合、処理時間の遅延が発生したと判定する。本実施例においては、閾値情報に記憶される閾値の値は固定値であるが、閾値の値は、例えば、収集部412が収集した応答時間の平均値の1.5倍の値といったように、収集部412が収集した応答時間に応じて動的に変更される様であっても良い。
30

〔原因特定処理〕

本実施例において、管理サーバ400が実行する、遅延原因を特定する処理について図10、図11とともに説明する。
40

【0037】

図10は、管理サーバ400が、応答時間の遅延を検出（S01）してから、遅延の原因を特定するために構成情報を参照する（S10）までの手順を示すフローチャートである。

【0038】

まず、検出部413が、性能情報記憶部422に記憶された応答時間1102、および閾値情報記憶部421に記憶された閾値情報1200に基づいて応答時間の遅延を検知する（S01）。具体的には、収集部412が収集し、性能情報記憶部422に記憶された応答時間1102の値が、閾値情報1200に記憶された値（閾値）を超えた場合に、検出部413は応答時間に遅延が発生していることを検知する。例えば、図8の性能情報1
50

110に示すように、採取時刻「2012/2/10 10:11」の応答時間1102の値は、閾値情報1200に示される3000ミリ秒を超える、4957ミリ秒である。したがって、検出部413は、応答時間に遅延が発生していることを検知する。

【0039】

応答時間の遅延の検出においては、応答時間が突発的に大きくなることが考えられる。このため、検出部413が実行する応答時間の遅延の検出は、例えば、閾値情報1200に示される値を超えたことを所定回数（例えば5回）検知した場合に、応答時間に遅延が発生していることとしても良い。

【0040】

応答時間の遅延を検出部413が検知すると、判定部415は、性能情報記憶部422に記憶された応答時間1102を参照する（S02）。また、判定部415は、性能情報記憶部422に記憶された外部リソースの性能情報を参照する（S03）。例えば本実施例の場合は、図7に示すように、要素E1005が外部リソースに該当するので、判定部415は、性能情報記憶部422に記憶された要素Eの処理時間1107を参照する。

10

【0041】

次に、判定部415は、性能情報記憶部422に記憶された、応答時間1102と要素Eの処理時間1107との間に相関関係があるかを判定する（S04）。相関関係があるかの判定の具体的な手法については詳細を後述する。

【0042】

ステップS04において、応答時間1102と、要素Eの処理時間1107との間に相関関係があると判定部415が判定した場合、判定部415は、応答時間の遅延の原因が要素E1005であることを特定する（S05）。つまり、判定部415は、アプリケーションサーバ200のクライアント装置500に対する応答の遅延が、外部リソースである要素E1005の処理の遅延に起因するものであると判定する。応答時間の遅延の原因を特定後、管理サーバ400は図10に示す遅延原因を特定する処理を終了する。

20

【0043】

一方、ステップS04において、応答時間1102と、要素Eの処理時間07との間に相関関係がないと判定部415が判定した場合、判定部415は、外部リソースである要素E1005以外の要素の処理時間を参照する（S06）。本実施例の場合、具体的には、判定部415は、性能情報記憶部422から、要素Aの処理時間1103、要素Bの処理時間1104、要素Cの処理時間1105、要素Dの処理時間1106を参照する。

30

【0044】

判定部415は、ステップS06で参照した要素の処理時間それについて、応答時間1102との間に相関関係があるかを判定し、処理時間が応答時間1102との間に相関関係がある要素を抽出する（S07）。相関関係があるかの判定の具体的な手法については、ステップS04における処理と同様の手法を用いることとし、詳細を後述する。

【0045】

判定部415は、ステップS07の処理の結果、処理時間が応答時間1102との間に相関関係がある要素を抽出したかを判定する（S08）。処理時間が応答時間1102との間に相関関係がある要素を抽出しなかった場合、判定部415は、アプリケーションの応答時間の遅延の原因が特定できないと判定し（S09）、図10に示す処理を終了する。

40

【0046】

一方、ステップS07の処理の結果、判定部415が、応答時間1102との間に相関関係がある要素の処理時間に対応する要素を抽出した場合、判定部415は、構成情報記憶部423に記憶された構成情報800を参照する（S10）。ステップS10の処理を実行後、処理は、図10の端子Aから図11の端子Aに移行する。

【0047】

図11は、管理サーバ400の判定部415が、ステップS07で抽出した要素のすべてに関連する関連要素があるかを判定（S11）してから、応答時間の遅延の原因を特定

50

するまでの手順を示すフローチャートである。

【0048】

判定部415は、ステップS07で抽出した要素のすべてに関連する要素（関連要素）があるかを判定する（S11）。例えば、本実施例の態様では、図7に示すように、要素B1002、要素C1003、要素D1004の関連要素が、要素Z1006である。したがって、例えば、ステップS07で抽出した要素が要素B1002と要素C1003であった場合、判定部415は、関連要素として要素Z1006があると判定する。

【0049】

ステップS11の判定の結果、関連要素があると判定部415が判定した場合、判定部415は、性能情報記憶部422に記憶された性能情報から、関連要素の処理時間を参照する（S12）。そして、判定部415は、応答時間1102と、関連要素の処理時間との間に相関関係があるかを判定する（S13）。相関関係があるかの判定の具体的な手法については、ステップS04、S07における処理と同様の手法を用いることとし、詳細を後述する。

10

【0050】

ステップS13の判定の結果、応答時間1102と、関連要素の処理時間との間に相関関係があると判定部415が判定した場合、判定部415は、応答時間の遅延の原因が関連要素であることを特定する（S14）。つまり、判定部415は、アプリケーションサーバ200のクライアント装置100に対する応答の遅延が、ステップS07で抽出された要素の処理の遅延によるものであり、さらに、それらの要素の処理の遅延は関連要素の処理の遅延に起因するものであると判定する。特定後、判定部415は処理を終了する。

20

【0051】

以上のように、本実施例によれば、アプリケーションの応答に遅延が発生した場合、応答時間と、アプリケーションの各要素の処理時間との相関関係に基づいて、アプリケーションの応答遅延の原因を特定することができる。

〔相関有無の判定の詳細〕

前述したステップS04、S07およびS13における、相関関係の有無を判定する手法について、詳細を説明する。

【0052】

解析部414は、応答時間と、各要素の処理時間の相関係数を算出する。判定部415は、解析部414によって算出された相関係数に基づいて、応答時間と要素の相関があるか否かを判定する。ここで、相関係数とは、比較を行う2つのデータ（群）がどの程度類似しているかを示す指標である。比較する相関関数は、例えば、図12に示す式によって算出する。

30

【0053】

例えば、応答時間1102と、要素Aの処理時間の相関係数を算出する場合、図12に示す式のxに、応答時間1102の値を代入する。また上記の式のyに、xに代入した応答時間1102と同日時に収集した要素Aの応答時間の値を代入する。代入する応答時間および処理時間の範囲は、例えば、図11および図12に示す処理の開始時点を基準として20分前までの応答時間および処理時間のように、性能情報記憶部422に記憶された性能情報から、所定の条件で特定の範囲を選択することとしても良い。尚、選択する範囲は一態様に定められる必要はなく、情報処理システム100の態様に応じて適宜変更されることとして良い。本実施例では、図11および図12に示す処理の開始時点を基準として20分前までの応答時間および処理時間を図12の式に代入する性能情報の対象とする。

40

【0054】

尚、上記の式で算出される相関係数は-1以上1以下の値をとる。算出された相関係数が1に近いほど、正の相関があることを示す。また、算出された相関係数が-1に近いほど、負の相関があることを示す。

【0055】

50

以上の手順で算出された相関係数に基づいて、判定部415は、応答時間と各要素の処理時間との間に相関関係があるか否かを判定する。例えば、判定部415によって算出された相関係数が0.7以上である場合、判定部415は、応答時間と各要素の処理時間との間に正の相関関係があると判定する。また、算出された相関係数が-0.7以下である場合、判定部415は、応答時間と各要素の処理時間との間に負の相関関係があると判定する。また、算出された相関係数が-0.7より大きく、0.7未満である場合、判定部415は、応答時間と各要素の処理時間との間に相関関係がないと判定する。

【0056】

具体例を挙げて説明すると、図8の性能情報に基づいて、応答時間1102と要素Aの処理時間1103との相関係数を算出した場合、相関係数は-0.49（小数第3位以下は四捨五入）である。したがって、判定部415は、応答時間1102と要素Aの処理時間1103との間に相関関係がないと判定する。一方、図8に示される性能情報に基づいて、応答時間1102と要素Bの処理時間1104との相関係数を算出した場合、相関係数は0.84（小数第3位以下は四捨五入）である。したがって、判定部415は、応答時間1102と要素Bの処理時間1104との間に正の相関関係があると判定する。

10

【応用例】

上記の説明においては、ステップS13において、応答時間1102と、関連要素の処理時間との間に相関関係がある場合に、アプリケーションの応答の遅延の原因を特定することができる。一方で、ステップS13において、応答時間1102と、関連要素の処理時間との間に相関関係がない場合であっても、アプリケーションの応答の遅延の原因を特定する手法が考えられる。以降は、この手法について説明する。尚、以降の説明は、図11におけるステップS15～S18の処理に相当する。

20

【0057】

前述したステップS13の判定の結果、判定部415が、応答時間1102と関連要素の処理時間との間に相関関係がないと判定した場合、判定部415は、性能情報記憶部422に記憶された、処理時間以外の性能情報を参照する（S15）。ここで、処理時間以外の性能情報とは、例えば、前述したスレッドプールというコンポーネントのキューに格納された作業項目の数（キュー格納数）や、JDBCプールの接続待ちの数（接続待ち数）などである。

30

【0058】

判定部415は、応答時間1102と、処理時間以外の性能情報との間に相関関係があるかを判定する（S16）。

【0059】

ステップS13の判定の結果、応答時間1102と、処理時間以外の性能情報との間に相関関係があると判定部415が判定した場合、判定部415は、応答時間の遅延の原因が関連要素であることを特定する（S17）。つまり、判定部415は、アプリケーションサーバ200のクライアント装置100に対する応答の遅延が、応答時間の遅延は、ステップS16の判定に用いられた性能情報に示される性能項目に起因するものであると判定する。特定後、判定部415は、図10、図11に示す遅延原因を特定する処理を終了する。

40

【0060】

一方、ステップS13の判定の結果、応答時間1102と、処理時間以外の性能情報との間に相関関係がないと判定部415が判定した場合、判定部415は、応答時間の遅延の原因が関連要素であることを特定する（S18）。つまり、判定部415は、アプリケーションサーバ200のクライアント装置100に対する応答の遅延が、ステップS07で抽出された要素のいずれかに起因するものであると判定する。特定後、判定部415は、図10、図11に示す遅延原因を特定する処理を終了する。

【0061】

応答時間と、処理時間以外の性能情報との相関を判定する場合も、相関係数を算出し、算出された相関係数の値に基づいて相関の有無を判定することができる。ここで、処理時

50

間以外の性能情報とは、例えば、前述したスレッドプールというコンポーネントのキューに格納された作業項目の数（キュー格納数）や、JDBCプールの接続待ちの数（接続待ち数）などである。

【0062】

以降に記載する例においては、他の性能情報の一例として、スレッドプールのキュー格納数を用いる場合について説明する。尚、以降に説明する処理は、図11のステップS16に相当する処理である。

【0063】

図13は、収集部412が収集し、性能情報記憶部422に記憶する、アプリケーション実行時のスレッドプールのキュー格納数の推移を示す情報の一例である。例えば、図13に示す情報の一部1310は、日時「2012/2/10 10:00」において、スレッドプールに格納されたキューの数が3であったことを示している。10

【0064】

応答時間と、キュー格納数との相関があるかを判定する処理について説明する。前述の処理と異なり、応答時間とキュー格納数とでは単位は収集される値の単位が異なる。本実施例では、単位の違いを吸収するため、解析部414が、性能情報記憶部422に記憶された性能情報の正規化を実行する。

【0065】

具体的な正規化の手法は、例えば、解析部414が、相関係数の算出にあたり、応答時間とキュー格納数それぞれについて、計測値の中での最小値と最大値を特定する。より詳細には、正規化後の値 = (正規化前の値 - 正規化前の最小値) ÷ (正規化前の最大値 - 正規化後の最大値) とする。この式を用いて正規化を行うことで、最大値が1、最小値が0となるように、性能情報が正規化され、例えば、時間（ミリ秒）とキュー格納数のように、単位の異なる性能情報同士を比較することが可能となる。解析部414は、応答時間とキュー格納数との相関があるかを判定する場合に、上記の手法を用いてそれぞれの値を正規化する。20

【0066】

図14に、解析部414により正規化が実行された後の、応答時間とキュー格納数を示す。

解析部414は、図14に示す性能情報に基づいて、応答時間（正規化後）1402とキュー格納数（正規化後）1403の相関係数を算出する。そして、判定部415は、算出された相関係数に基づいて、応答時間（正規化後）1402とキュー格納数（正規化後）1403との相関があるかを判定する。ここで、図10の式を用いて相関係数を算出すると、応答時間（正規化後）1402とキュー格納数（正規化後）1403の相関係数は0.87（小数第3位以下を切り捨て）である。したがって、このとき、判定部415は、応答時間（正規化後）1402とキュー格納数（正規化後）1403との間に正の相関関係があると判定する。30

【0067】

以上説明したステップS16の処理のように、本実施例では、性能情報記憶部422に記憶された応答時間の情報とキュー格納数の情報に基づいて、判定部415が応答時間とキュー格納数に相関があるかと判定することもできる。ここで例えば、判定部415が、応答時間とキュー格納数に相関があると判定した場合、応答時間の遅延の原因是、スレッドプールのキュー格納数が増加したことであると特定することができる。40

【0068】

本実施例にて管理サーバ400が実行する処理によって、遅延の原因と特定された要素、つまりコンポーネントや外部リソースの情報は、例えば図1に示す管理端末600に出力される。そして、管理端末600を用いて、システム管理者は、アプリケーションの応答時間の遅延の原因を、容易かつ迅速に把握することができる。

〔総括〕

以上のように、本実施例によれば、アプリケーションサーバ200が実行するアプリケ50

ーションの、クライアント装置 100 に対する応答時間に遅延が発生した場合、遅延の原因である要素を特定することができる。したがって、アプリケーションの応答時間の遅延の原因を、従来技術と比較して迅速に特定することができ、アプリケーションを実行する情報処理システム 100 の運用管理の効率を向上させることが可能である。

【0069】

また、本実施例によれば、アプリケーションの応答時間の推移と処理時間の推移に相関がある要素が複数抽出された場合、抽出された複数の要素が共通して依存する要素（関連要素）を特定することで、遅延の要因となる要素を絞り込むことが可能である。

【0070】

また、本実施例によれば、応答時間と処理時間用いて、応答時間の遅延の原因が特定できない場合であっても、処理時間以外の性能情報を用いて、応答時間の遅延の原因を特定することができる。

【0071】

本実施例は、本発明を実現する一態様であり、本実施例における具体的な処理や、処理の手順は、本発明を適用するシステムの態様を考慮して適宜変更することが可能である。

【0072】

例えば、性能情報を収集するタイミングや、収集する具体的な性能情報の種別は、システムの態様等にあわせて適宜変更することが可能である。また例えば、遅延を検出する閾値は、システムの態様等にあわせて適宜変更することが可能である。また例えば、相関があるか否かの判定に用いられる性能情報の範囲も、適宜変更して良い。

【0073】

また、本実施例では、図 11 のステップ S11において、判定部 415 は、ステップ S07 で抽出した要素のすべてに関連する要素（関連要素）があるかを判定するかを判定することとしたが、ステップ S11 の処理に対しても適宜変更が行われても良い。例えば、ステップ S07 で抽出した要素の内、2 つ以上の要素に関連する要素を関連要素として抽出し、判定部 415 が、応答時間と関連要素の処理時間に相関があると判定した場合には、応答時間の遅延の原因が関連要素の処理の遅延であると特定することとしても良い。

【0074】

上述した事項以外にも、本発明の実施にあたり、実施例における具体的な処理や、処理の手順は適宜変更することが可能である。

【符号の説明】

【0075】

100 情報処理システム

200 アプリケーションサーバ

300 データベース

400 管理サーバ

401 C P U

402 R A M

403 H D D

404 R O M ドライブ

405 ディスク R O M

406 アダプタ

411 通信部

412 収集部

413 検出部

414 解析部

415 判定部

421 閾値情報記憶部

422 性能情報記憶部

423 構成情報記憶部

10

20

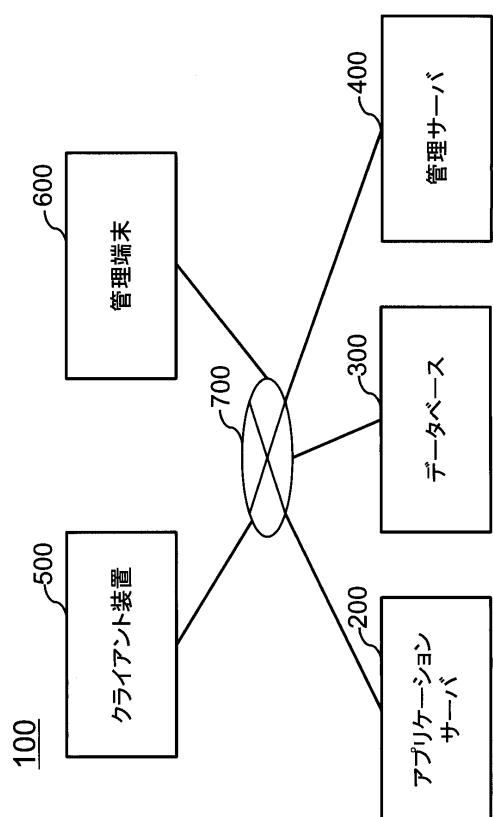
30

40

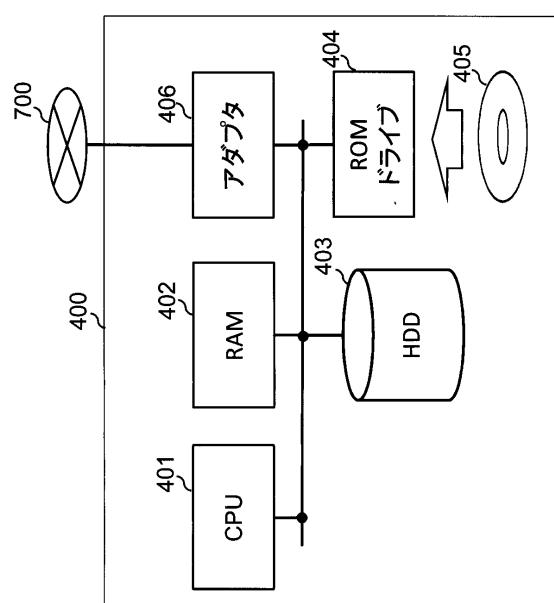
50

500 クライアント装置
 600 管理端末
 700 ネットワーク

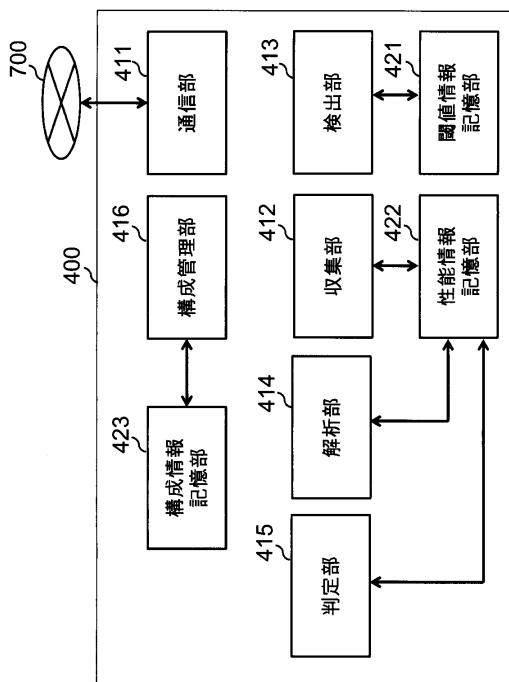
【図1】



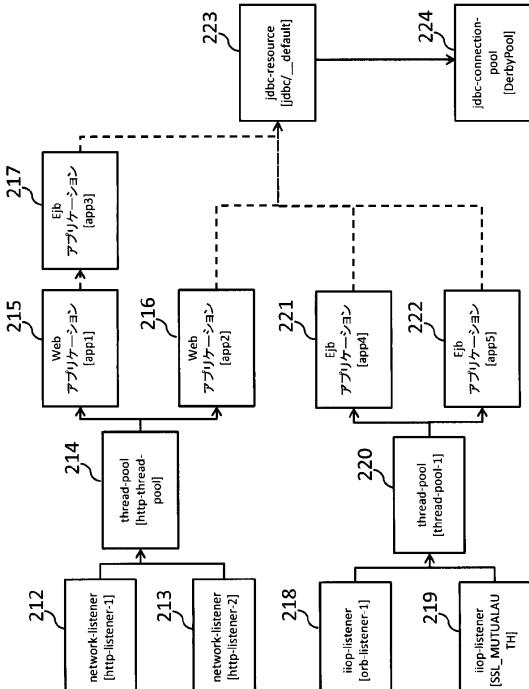
【図2】



【図3】



【 図 4 】



(5)

906

900 図5からの続き

```

<network-listeners>
  <network-listener name="http-listener-1" port="908" />
  <network-listener name="http-listener-2" port="909" />
  <network-listener name="admin-listener" port="910" />
</network-listeners>

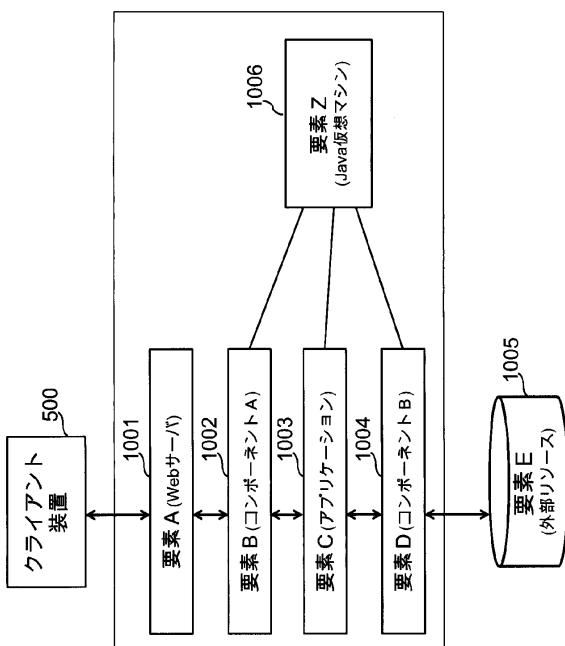
<thread-pools>
  <thread-pool name="admin-thread-pool" max-queue-size="256" max-thread-pool-size="50" />
  <thread-pool name="http-thread-pool" max-queue-size="4096" max-thread-pool-size="200" />
  <thread-pool name="http-thread-pool-1" max-queue-size="3024" max-thread-pool-size="200" />
</thread-pools>

<protocols>
  <protocol name="http-listener-1" server="true">
    <file cache>
      <http>
        <!-- protocol -->
      </http>
    </file cache>
  </protocol>
  <protocol name="http-listener-2" server="true">
    <file cache>
      <https>
        <!-- protocol -->
      </https>
    </file cache>
  </protocol>
</protocols>

```

800

【図7】



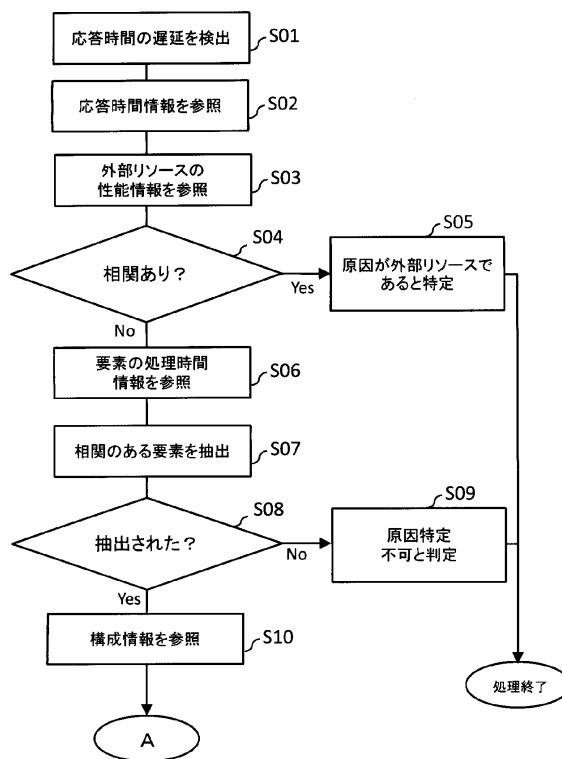
【図8】

検査特別	応答時間	要素Aの処理時間	要素Bの処理時間	要素Cの処理時間	要素Dの処理時間	要素Eの処理時間	要素Zの処理時間
2012/2/10 10:00	1245	184	113	159	194	595	197
2012/2/10 10:01	2999	174	1887	176	195	567	165
2012/2/10 10:02	2111	151	1110	185	184	481	186
2012/2/10 10:03	1929	132	780	185	174	658	188
2012/2/10 10:04	1851	195	284	197	151	1024	128
2012/2/10 10:05	2299	175	808	185	184	947	197
2012/2/10 10:06	1358	104	7	184	174	889	148
2012/2/10 10:07	1696	194	287	174	151	890	198
2012/2/10 10:08	1515	195	133	151	132	904	164
2012/2/10 10:09	1875	184	481	132	104	974	122
2012/2/10 10:10	1655	174	215	174	185	907	165
2012/2/10 10:11	4957	151	1071	1857	1004	874	3279
2012/2/10 10:12	5285	132	1072	1940	1194	947	3657
2012/2/10 10:13	8521	104	3117	2132	2194	974	4665
2012/2/10 10:14	4856	185	1166	1394	1194	917	3211
2012/2/10 10:15	3856	159	874	874	975	974	2804
2012/2/10 10:16	2041	195	554	151	185	956	199
2012/2/10 10:17	1875	175	462	132	159	947	180
2012/2/10 10:18	2515	186	984	195	176	974	183
2012/2/10 10:19	2954	186	1501	175	175	917	151
2012/2/10 10:20	1880	159	432	186	186	917	163

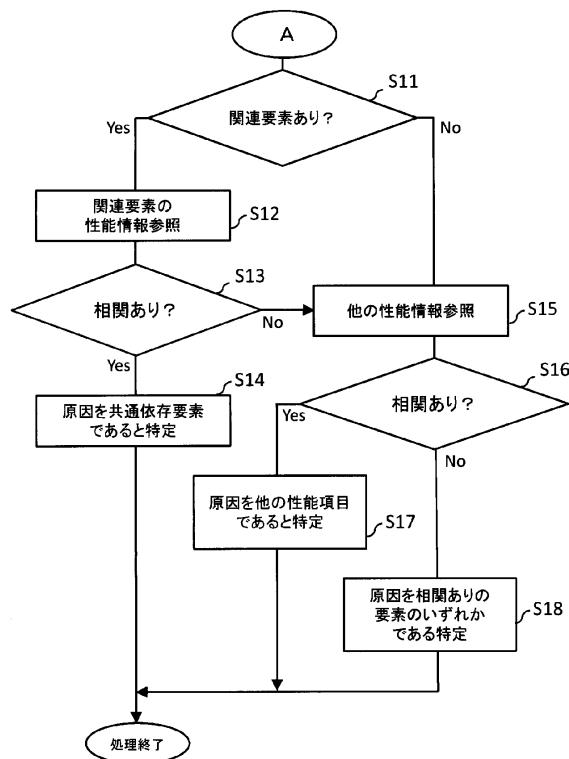
【図9】

1200
閾値(応答時間) [msec]
3000

【図10】



【図11】



【図12】

$$\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

【図13】

採取時刻	キュー格納数
2012/2/10 10:00	3
2012/2/10 10:01	4
2012/2/10 10:02	3
2012/2/10 10:03	2
2012/2/10 10:04	3
2012/2/10 10:05	4
2012/2/10 10:06	6
2012/2/10 10:07	5
2012/2/10 10:08	3
2012/2/10 10:09	4
2012/2/10 10:10	5
2012/2/10 10:11	12
2012/2/10 10:12	15
2012/2/10 10:13	18
2012/2/10 10:14	14
2012/2/10 10:15	13
2012/2/10 10:16	6
2012/2/10 10:17	3
2012/2/10 10:18	1
2012/2/10 10:19	2
2012/2/10 10:20	4

【図14】

採取時刻	応答時間(正規化後)	キュー格納数(正規化後)
2012/2/10 10:00	0	0.117647059
2012/2/10 10:01	0.24106632	0.176470588
2012/2/10 10:02	0.11902144	0.117647059
2012/2/10 10:03	0.094007697	0.058823529
2012/2/10 10:04	0.083287521	0.117647059
2012/2/10 10:05	0.144859813	0.176470588
2012/2/10 10:06	0.015530511	0.294117647
2012/2/10 10:07	0.061984607	0.235294118
2012/2/10 10:08	0.037108301	0.117647059
2012/2/10 10:09	0.086586036	0.176470588
2012/2/10 10:10	0.056349643	0.235294118
2012/2/10 10:11	0.510170423	0.647058824
2012/2/10 10:12	0.555250137	0.823529412
2012/2/10 10:13	1	1
2012/2/10 10:14	0.49628917	0.764705882
2012/2/10 10:15	0.358851017	0.705882353
2012/2/10 10:16	0.109404077	0.294117647
2012/2/10 10:17	0.086586036	0.117647059
2012/2/10 10:18	0.174546454	0
2012/2/10 10:19	0.234881803	0.058823529
2012/2/10 10:20	0.087727327	0.176470588

フロントページの続き

(72)発明者 越後 慎也

神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

審査官 多賀 実

(56)参考文献 特開2006-011902 (JP, A)

特開2005-346414 (JP, A)

特開2010-108445 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G 06 F 11 / 07

G 06 F 11 / 28 - 11 / 36