(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2019-12477
(P2019-12477A)
(43)公開日　平成31年1月24日(2019.1.24)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 11/07	(2006.01)	G06F	11/07	190	5B042
G06F 9/46	(2006.01)	G06F	9/46	350	
G06F 9/50	(2006.01)	G06F	9/46	462Z	
G06F 11/34	(2006.01)	G06F	11/34	104	
		G06F	11/34	133	

審査請求 未請求　請求項の数 10　ＯＬ　（全 32 頁）　最終頁に続く

(21)出願番号　特願2017-129803 (P2017-129803)
(22)出願日　平成29年6月30日 (2017.6.30)

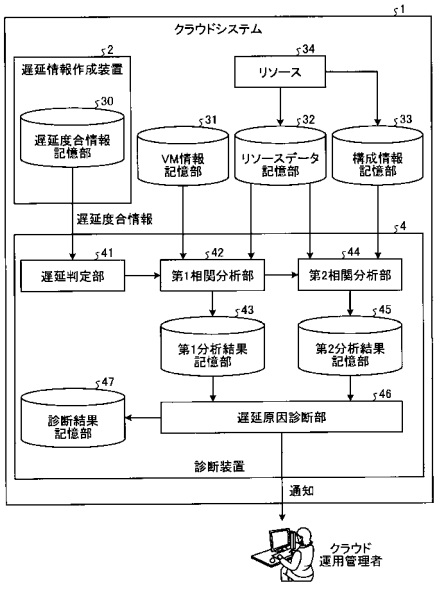
(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　110002147
特許業務法人酒井国際特許事務所
(72)発明者　鈴木　脩司
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　金政　泰彦
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

Ｆターム(参考)　5B042 GA12 GA22 JJ20 JJ29 KK15 MA14 MC22 MC29 MC33

(54)【発明の名称】診断プログラム、診断方法及び診断装置

(57)【要約】
【課題】アプリの性能低下がクラウド基盤に原因があるのかアプリに原因があるのかを特定すること。
【解決手段】遅延判定部４１が遅延度合情報を遅延情報作成装置２から取得してアプリに遅延が発生したか否かを判定し、第１相関分析部４２と第２相関分析部４４が遅延が発生したアプリに関係するリソースデータと遅延度合との相関分析を行う。そして、遅延原因診断部４６が、第１相関分析部４２と第２相関分析部４４の相関分析結果に基づいて遅延原因を特定する。
【選択図】図２

実施例1に係るクラウドシステムの構成を示す図



【特許請求の範囲】
【請求項１】
コンピュータに、
アプリケーション毎に実行の遅延を示す遅延度合を取得して前記遅延度合が所定の閾値以上であるか否かを判定し、
前記遅延度合が所定の閾値以上であると判定した場合に、前記アプリケーションに関係するリソースに関する情報と前記遅延度合との相関関係に基づいて前記遅延の原因を特定する
診断プログラム。
【請求項２】
前記特定する処理は、前記アプリケーションが実行される仮想マシンが利用するリソースに関する情報と前記遅延度合との相関関係である第１相関関係、及び、前記リソースを共有している全仮想マシンのリソース負荷量を表すデータと前記遅延度合との相関関係である第２相関関係に基づいて前記遅延の原因を特定することを特徴とする請求項１に記載の診断プログラム。
【請求項３】
前記第１相関関係は、前記仮想マシンのリソース負荷量を表すデータと前記遅延度合との相関関係、及び、前記仮想マシンのリソース性能低下の影響を受けるデータと前記遅延度合との相関関係であることを特徴とする請求項２に記載の診断プログラム。
【請求項４】
前記第１相関関係は、前記仮想マシンのＣＰＵ使用率と前記遅延度合との相関関係、及び、前記仮想マシンのＣＰＵ割り当て待ち時間の割合と前記遅延度合との相関関係であることを特徴とする請求項３に記載の診断プログラム。
【請求項５】
前記特定する処理は、前記仮想マシンのＣＰＵ割り当て待ち時間の割合と前記遅延度合との間に相関関係があり、かつ、前記仮想マシンが利用するＣＰＵを共有している他の仮想マシンのＣＰＵ使用率と前記遅延度合との相関関係がある場合に、前記遅延の原因としてクラウド基盤を特定することを特徴とする請求項４に記載の診断プログラム。
【請求項６】
前記遅延の原因を特定する処理は、前記遅延度合が所定の閾値以上のときの前記リソースに関する情報と前記遅延度合が所定の閾値以上でないときの前記リソースに関する情報との間に差がある場合に、前記相関関係があると判定することを特徴とする請求項１～５のいずれか１つに記載の診断プログラム。
【請求項７】
前記遅延度合は、リクエストに対する応答時間の一定時間毎の平均値を正規化した正規化平均応答時間であり、
複数の正規化平均応答時間のうち、平均値の算出に用いられたリクエストの数が少ない場合の正規化平均応答時間を不安定な正規化平均応答時間として除く処理をさらに前記コンピュータに実行させ、
前記特定する処理は、前記不安定な正規化平均応答時間が除かれた正規化応答時間のうち第１閾値以上の個数が一定期間に第２閾値以上であるとき、又は、前記不安定な正規化平均応答時間を含む正規化応答時間のうち第１閾値以上の個数が一定期間に第２閾値以上であるときに、前記遅延の原因を特定することを特徴とする請求項１～６のいずれか１つに記載の診断プログラム。
【請求項８】
前記特定する処理は、前記遅延度合との相関係数を計算し、無相関検定を行ってｐ－ｖａｌｕｅを計算し、計算したｐ－ｖａｌｕｅを多重検定補正によって補正し、補正したｐ－ｖａｌｕｅに基づいて前記遅延度合と相関があるか否かを判定して前記第１相関関係を分析することを特徴とする請求項２～５のいずれか１つに記載の診断プログラム。
【請求項９】

コンピュータが、
アプリケーション毎に実行の遅延を示す遅延度合を取得して前記遅延度合が所定の閾値以上であるか否かを判定し、
前記遅延度合が所定の閾値以上であると判定した場合に、前記アプリケーションに関係するリソースに関する情報と前記遅延度合との相関関係に基づいて前記遅延の原因を特定する
処理を実行することを特徴とする診断方法。
【請求項１０】
アプリケーション毎に実行の遅延を示す遅延度合を取得して前記遅延度合が所定の閾値以上であるか否かを判定する遅延判定部と、
前記遅延判定部により遅延度合が所定の閾値以上であると判定された場合に、前記アプリケーションに関係するリソースに関する情報と前記遅延度合との相関関係に基づいて前記遅延の原因を特定する特定部と
を有することを特徴とする診断装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、診断プログラム、診断方法及び診断装置に関する。
【背景技術】
【０００２】
顧客にリソース（Resource）を提供するシステムでは、提供するリソースの状況を監視してリソースに問題がないかを確認することが重要となる。例えば、仮想マシンを提供するクラウドシステムでは、仮想マシン上で動作するアプリケーションの応答時間や負荷を監視してアプリケーションの性能に問題がないかを確認することが重要である。
【０００３】
ここで、仮想マシンとは、物理マシン（コンピュータ）上で動作する仮想的なコンピュータである。また、クラウドシステムとは、ネットワークを介して利用者にコンピュータのハードウェアやソフトウェアを提供するシステムである。
【０００４】
アプリケーションの性能に関するデータの収集にはエージェントが用いられる。図３８は、エージェントによる監視を説明するための図である。図３８に示すように、物理マシン９では仮想マシン９ａが動作し、仮想マシン９ａによりアプリとエージェントが実行される。ここで、アプリは、アプリケーションである。エージェントは、アプリから性能に関するデータを収集してアプリの性能を監視する。
【０００５】
なお、論理的サーバ構成である複数のインスタンスに割り当てられたリソースの性能劣化を検出するとともに、性能劣化を呈するリソースを共有するインスタンスを抽出し、抽出したインスタンスの性能傾向と、性能パターンを比較する技術がある。ここで、性能パターンは、ボトルネック要因と関連付けてリソースの性能情報から抽出した特徴量を示す。この技術によれば、インスタンスの性能傾向との類似度が最大となる性能パターンからボトルネック要因を推定することができる。
【０００６】
また、連携してサービスを提供する複数のリソース各々に関する複数の情報を異なる管理装置から取得し、取得した複数の情報間の相関関係と所定の相関関係との差に基づいて、異常を発生させる原因となった異常原因リソースを特定する技術がある。この技術は、複数の情報間の相関関係と所定の相関関係との差が許容値よりも大きくなったリソースを異常リソース候補として抽出し、サービスにおける複数のリソース構成を示す構成情報に基づいて、異常リソース候補の中から異常原因リソースを特定する。この技術によれば、異常原因リソースを適切に特定することができる。
【先行技術文献】

【特許文献】
【０００７】
【特許文献１】国際公開第２０１５／１４５６６４号
【特許文献２】特開２０１３－１６１３０５号公報
【発明の概要】
【発明が解決しようとする課題】
【０００８】
図３８に示した性能監視には、各アプリケーションの性能を監視することはできるが、アプリケーションの性能低下がクラウド基盤に原因があるのかアプリケーションに原因があるのかを特定することができないという問題がある。ここで、クラウド基盤とは、サーバ、ネットワーク、ストレージ等のＩＣＴ（Information and Communication Technology）インフラを仮想化技術を利用して提供する基盤である。クラウド基盤は、仮想マシン管理、ストレージ管理、ネットワーク管理等の機能を備える。
【０００９】
本発明は、一つの側面では、アプリケーションの性能低下がクラウド基盤に原因があるのかアプリケーションに原因があるのかを特定することを目的とする。
【課題を解決するための手段】
【００１０】
１つの態様では、診断プログラムは、コンピュータに、アプリケーション毎に実行の遅延を示す遅延度合を取得して遅延度合が所定の閾値以上であるか否かを判定する処理を実行させる。そして、診断プログラムは、コンピュータに、遅延度合が所定の閾値以上であると判定した場合に、アプリケーションに関係するリソースに関する情報と遅延度合との相関関係に基づいて遅延の原因を特定する処理を実行させる。
【発明の効果】
【００１１】
１つの側面では、本発明は、アプリケーションの性能低下がクラウド基盤に原因があるのかアプリケーションに原因があるのかを特定することができる。
【図面の簡単な説明】
【００１２】
【図１】図１は、実施例１に係る診断装置による遅延原因診断を説明するための図である。
【図２】図２は、実施例１に係るクラウドシステムの構成を示す図である。
【図３】図３は、遅延情報作成装置の構成を示す図である。
【図４】図４は、種別判定用データ記憶部の一例を示す図である。
【図５】図５は、種別情報記憶部の一例を示す図である。
【図６】図６は、応答時間の算出方法を説明するための図である。
【図７】図７は、応答時間情報記憶部の一例を示す図である。
【図８】図８は、平均応答時間の正規化を説明するための図である。
【図９】図９は、代表情報記憶部の一例を示す図である。
【図１０】図１０は、ｅｘ－Ｇａｕｓｓｉａｎ分布を説明するための図である。
【図１１】図１１は、外れ値を説明するための図である。
【図１２】図１２は、遅延度合情報記憶部の一例を示す図である。
【図１３】図１３は、アプリの遅延判定を説明するための図である。
【図１４】図１４は、ＶＭ情報記憶部の一例を示す図である。
【図１５】図１５は、リソースデータ記憶部の一例を示す図である。
【図１６】図１６は、リソースデータと遅延度合の関係の例を示す図である。
【図１７】図１７は、遅延しているときと遅延していないときのリソースデータの傾向に差があるか否かの判定による相関分析を説明するための図である。
【図１８Ａ】図１８Ａは、相関が弱い場合と強い場合を示す図である。
【図１８Ｂ】図１８Ｂは、遅延度合とよく相関のあるリソースデータの抽出を説明するた

めの図である。
【図１９】図１９は、第１分析結果記憶部の一例を示す図である。
【図２０】図２０は、構成情報記憶部の一例を示す図である。
【図２１】図２１は、第２相関分析部による相関分析を説明するための図である。
【図２２】図２２は、第２分析結果記憶部の一例を示す図である。
【図２３】図２３は、診断結果記憶部の一例を示す図である。
【図２４】図２４は、クラウド運用管理者への通知例を示す図である。
【図２５】図２５は、診断装置による処理のフローを示すフローチャートである。
【図２６】図２６は、第１相関分析処理のフローを示すフローチャートである。
【図２７】図２７は、リソースデータ抽出処理のフローを示すフローチャートである。
【図２８】図２８は、第２相関分析処理のフローを示すフローチャートである。
【図２９】図２９は、遅延原因診断処理のフローを示すフローチャートである。
【図３０】図３０は、二群比較検定を用いる第１相関分析処理のフローを示すフローチャートである。
【図３１】図３１は、二群比較検定を用いる第２相関分析処理のフローを示すフローチャートである。
【図３２】図３２は、遅延度合が不安定なデータの除去を説明するための図である。
【図３３】図３３は、実施例２に係るクラウドシステムの構成を示す図である。
【図３４】図３４は、遅延度合が不安定なデータの除去方法を説明するための図である。
【図３５】図３５は、診断装置による処理のフローを示すフローチャートである。
【図３６】図３６は、閾値計算処理のフローを示すフローチャートである。
【図３７】図３７は、実施例１及び２に係る診断プログラムを実行するコンピュータのハードウェア構成を示す図である。
【図３８】図３８は、エージェントによる監視を説明するための図である。
【発明を実施するための形態】
【００１３】
以下に、本願の開示する診断プログラム、診断方法及び診断装置の実施例を図面に基づいて詳細に説明する。なお、実施例は開示の技術を限定するものではない。
【実施例１】
【００１４】
まず、実施例１に係る診断装置による遅延原因診断について説明する。図１は、実施例１に係る診断装置による遅延原因診断を説明するための図である。図１に示すように、実施例１に係る診断装置は、アプリに関係するＶＭ（Virtual Machine：仮想マシン）のリソースデータと遅延度合との相関分析を行う（ｔ１）。
【００１５】
ここで、アプリに関係するＶＭは、アプリが利用するＶＭであり、複数ある場合がある。例えば、Ｗｅｂサーバ、アプリサーバ、ＤＢ（Database）サーバから成る３階層システムでは、１つのアプリに関係するＶＭは３つある。遅延度合は、アプリの遅延の度合である。
【００１６】
リソースデータには、ＶＭのリソース毎にリソース負荷量を表すデータとリソース性能低下の影響を受けるデータの２種類のデータがある。例えば、リソースがＣＰＵ（Central Processing Unit）の場合には、リソース負荷量を表すデータにはＣＰＵ使用率があり、リソース性能低下の影響を受けるデータにはＣＰＵの割り当て待ち時間の割合がある。図１は、リソースがＣＰＵの場合を示す。
【００１７】
また、リソースがディスク装置の場合には、リソース負荷量を表すデータにはＲｅａｄ発行数、読み込み量、Ｗｒｉｔｅ発行数、書き込み量等があり、リソース性能低下の影響を受けるデータには平均Ｒｅａｄ時間、平均Ｗｒｉｔｅ時間等がある。また、リソースがネットワークの場合には、リソース負荷量を表すデータにはパケット数、通信量等があり

、リソース性能低下の影響を受けるデータにはラウンドトリップ時間、パケットのドロップ率等がある。

【００１８】

実施例１に係る診断装置は、アプリに関係するＶＭのリソース毎に、リソース負荷量を表すデータと遅延度合との相関分析、リソース性能低下の影響を受けるデータと遅延度合との相関分析を行う。

【００１９】

そして、遅延度合が、あるＶＭのＣＰＵのリソースデータと相関がない場合には、実施例１に係る診断装置は、遅延とこのＶＭのＣＰＵとは関係ないと診断する（ｔ２）。また、遅延度合がＣＰＵ使用率と相関がある場合には、実施例１に係る診断装置は、アプリ要因による遅延と診断する（ｔ３）。

【００２０】

また、遅延度合がＣＰＵの割り当て待ち時間の割合と相関がある場合には、実施例１に係る診断装置は、遅延度合がＣＰＵの割り当て待ち時間の割合と相関があったＶＭとＣＰＵを共有している全ＶＭについてＣＰＵ使用率と遅延度合との相関分析を行う（ｔ４）。ここで、遅延度合がＣＰＵの割り当て待ち時間の割合と相関があったＶＭとＣＰＵを共有している全ＶＭには、遅延度合がＣＰＵの割り当て待ち時間の割合と相関のあったＶＭも含まれる。

【００２１】

そして、遅延度合が他ＶＭのＣＰＵ使用率と相関があった場合には、他ＶＭがＣＰＵを大量に使うことによりＣＰＵ負荷が上昇し、遅延が発生しているため、実施例１に係る診断装置は、ＣＰＵ要因すなわちクラウド基盤要因による遅延と診断する（ｔ５）。また、遅延度合がいずれのＶＭのＣＰＵ使用率とも相関がない場合には、実施例１に係る診断装置は、ＣＰＵ負荷上昇の原因不明と診断する（ｔ６）。また、遅延度合がＣＰＵ負荷の影響を受けているＶＭ自体のＣＰＵ使用率と相関がある場合には、遅延しているアプリのＶＭ自身がＣＰＵの負荷を上昇させ遅延が発生してるため、実施例１に係る診断装置は、アプリ要因による遅延と診断する（ｔ３）。

【００２２】

このように、実施例１に係る診断装置は、アプリに関係するＶＭのリソース毎に、リソース負荷量を表すデータと遅延度合との相関分析、リソース性能低下の影響を受けるデータと遅延度合との相関分析を行う。そして、遅延度合がリソース性能低下の影響を受けるデータとの相関がある場合には、実施例１に係る診断装置は、遅延度合がリソース性能低下の影響を受けるデータと相関があったＶＭとリソースを共有している全ＶＭについてリソース負荷量を表すデータとの相関分析を行う。そして、実施例１に係る診断装置は、これらの相関分析の結果に基づいて、アプリの遅延原因を診断する。したがって、実施例１に係る診断装置は、アプリの性能低下がクラウド基盤に原因があるのかアプリに原因があるのかを特定することができる。

【００２３】

次に、実施例１に係るクラウドシステムの構成について説明する。図２は、実施例１に係るクラウドシステムの構成を示す図である。図２に示すように、実施例１に係るクラウドシステム１は、遅延情報作成装置２と、診断装置４と、ＶＭ情報記憶部３１と、リソースデータ記憶部３２と、構成情報記憶部３３と、リソース３４とを有する。リソース３４には、ＣＰＵ、ディスク装置、ネットワークスイッチが含まれる。

【００２４】

遅延情報作成装置２は、アプリの遅延度合情報を作成し、作成した遅延度合情報を遅延度合情報記憶部３０に記憶する。図３は、遅延情報作成装置２の構成を示す図である。図３に示すように、遅延情報作成装置２は、キャプチャ部２１と、パケット情報記憶部２２と、種別判定用データ記憶部２３と、種別判定部２４と、種別情報記憶部２５と、応答時間算出部２６と、応答時間情報記憶部２７とを有する。また、遅延情報作成装置２は、正規化部２８と、代表情報記憶部２９と、遅延度合情報記憶部３０とを有する。

【００２５】
キャプチャ部２１は、ネットワークスイッチ３ｃを通過する通信パケットをポートミラーリングによりキャプチャし、キャプチャした通信パケットの情報をパケット情報記憶部２２に格納する。パケット情報記憶部２２は、ネットワークスイッチ３ｃを通過する通信パケットの情報を記憶する。
【００２６】
種別判定用データ記憶部２３は、アプリの種別を判定するためのデータを記憶する。アプリの種別には、応答時間が性能面で重要となるアプリとその他のアプリとがある。遅延情報作成装置２は、応答時間が性能面で重要となるアプリを遅延度合情報作成の対象とする。
【００２７】
種別判定部２４は、種別判定用データ記憶部２３が記憶するデータを用いて通信コネクション毎にアプリの種別を判定する。図４は、種別判定用データ記憶部２３の一例を示す図である。図４（ａ）は、アプリの種別を判定するためのデータとしてポート番号のリストであるポートリストを記憶する場合を示す。図４（ａ）において、種別判定用データ記憶部２３が記憶するポート番号は、応答時間が性能面で重要となるアプリが使用するポート番号である。例えば、種別判定用データ記憶部２３は、応答時間が性能面で重要となるアプリが使用するポート番号として、「８０」、「４４３」等を記憶する。
【００２８】
種別判定部２４は、パケット情報記憶部２２に記憶された通信パケットの情報を解析し、サーバ側のポート番号を抽出する。ここで、サーバとは、仮想マシンである。そして、種別判定部２４は、抽出したポート番号が種別判定用データ記憶部２３が記憶するポートリストに含まれる場合に、解析した通信パケットに送信又は受信するアプリを応答時間が性能面で重要となるアプリとして判定する。そして、種別判定部２４は、判定結果を種別情報記憶部２５に格納する。
【００２９】
また、種別判定部２４は、ポート番号からアプリの種別が判定できないアプリについては、通信パターンを入力として機械学習によってアプリの種別を判定する。
【００３０】
具体的には、種別判定部２４は、あらかじめ、応答時間が性能面で重要となるアプリ及びその他のアプリについて、通信パケットを収集しておく。そして、種別判定部２４は、収集した通信パケットを解析して一定時間間隔（例えば１分）の平均応答時間、サーバの平均通信量、サーバの平均通信回数、クライアント装置の平均通信量、及びクライアント装置の平均通信回数を計算する。
【００３１】
そして、種別判定部２４は、計算した値を学習データとして、学習器を構築する。学習器としては、ＳＶＭ（Support Vector Machine）、ランダムフォレスト等が利用可能である。図４（ｂ）は、種別判定用データ記憶部２３がアプリの種別を判定するためのデータとして学習データを記憶する場合を示す。図４（ｂ）に示すように、種別判定用データ記憶部２３は、アプリの種別、平均応答時間、サーバの平均通信量、サーバの平均通信回数、クライアント装置の平均通信量、及びクライアント装置の平均通信回数を一つの学習データとして記憶する。平均応答時間の単位はマイクロ秒であり、サーバの平均通信量及びクライアント装置の平均通信量の単位はバイトである。
【００３２】
図４（ｂ）では、アプリの種別が「応答時間が性能面で重要となるアプリ」について二つの学習データが示され、アプリの種別が「その他のアプリ」について一つの学習データが示されている。アプリの種別が「応答時間が性能面で重要となるアプリ」の学習データの一つでは、平均応答時間が「６００」であり、サーバの平均通信量は「１００」であり、サーバの平均通信回数は「１」である。また、その学習データでは、クライアント装置の平均通信量は「１００」であり、クライアント装置の平均通信回数は「１」である。

【００３３】
そして、種別判定部２４は、キャプチャされた通信パケットから通信コネクション毎に学習データと同じ時間間隔の平均応答時間、サーバの平均通信量、サーバの平均通信回数、クライアント装置の平均通信量、及びクライアント装置の平均通信回数を計算する。そして、種別判定部２４は、計算した値から、学習器を利用して通信コネクションに対応するアプリの種別を判定する。そして、種別判定部２４は、判定結果を種別情報記憶部２５に格納する。
【００３４】
種別情報記憶部２５は、アプリの種別の判定結果を記憶する。図５は、種別情報記憶部２５の一例を示す図である。図５に示すように、種別情報記憶部２５は、ＩＰアドレス、ポート番号及び種別をアプリ毎に記憶する。ＩＰアドレスは、アプリが稼働しているＶＭのＩＰアドレスである。ポート番号は、アプリが使用するポート番号である。種別は、アプリの種別である。例えば、ＩＰアドレスが「１０．２０．３０．４０」であるＶＭで稼働するアプリは、使用するポート番号は「８０」であり、「応答時間が性能面で重要となるアプリ」である。
【００３５】
応答時間算出部２６は、応答時間が性能面で重要となるアプリに関して、通信パケットを解析して応答時間を算出し、算出した応答時間を応答時間情報記憶部２７に格納する。応答時間算出部２６は、通信パケットが暗号化されていない場合には、プロトコルメッセージを再構築し、リクエストの時刻とレスポンスの時刻より応答時間を算出する。
【００３６】
すなわち、応答時間算出部２６は、通信パケットからプロトコルメッセージを再構成して、どの通信パケットがリクエストメッセージであり、どの通信パケットがレスポンスメッセージなのかを判定する。そして、応答時間算出部２６は、リクエストメッセージが送信されてからレスポンスメッセージが返ってくるまでの時間を応答時間として算出する。
【００３７】
図６は、応答時間の算出方法を説明するための図である。図６に示すように、クライアント装置が送信したリクエストメッセージがクラウドシステム１内のＶＭで稼働するアプリにより処理され、レスポンスメッセージがアプリからクライアント装置に送信される。応答時間算出部２６は、リクエストメッセージがキャプチャされた時刻とレスポンスメッセージがキャプチャされた時刻の間の時間を応答時間とする。
【００３８】
応答時間算出部２６は、通信パケットが暗号化されている場合には、通信パケットの送受のフローを解析することによって、アプリの応答時間を推定する。通信パケットが暗号化されている場合、応答時間算出部２６は、通信パケットの内部が見られないためにプロトコルの解析ができないので、リクエストメッセージやレスポンスメッセージを再構成することができない。そこで、応答時間算出部２６は、アプリ側から一方的にデータをリアルタイムに送信するための技術である「ロングポーリング」を考慮し、クライアント装置とクラウドシステム１上のアプリの通信パケットの時間間隔から応答時間を推定する。
【００３９】
応答時間情報記憶部２７は、応答時間算出部２６によりアプリ毎に算出された応答時間を記憶する。図７は、応答時間情報記憶部２７の一例を示す図である。図７に示すように、応答時間情報記憶部２７は、時刻と、ＩＰアドレスと、ポート番号と、応答時間とを対応付けて記憶する。
【００４０】
時刻は、応答時間が算出された時刻である。ＩＰアドレスは、アプリが稼働しているＶＭのＩＰアドレスである。ポート番号は、アプリが使用するポート番号である。応答時間は、応答時間算出部２６により算出された応答時間である。応答時間の単位はマイクロ秒である。例えば、ＩＰアドレスが「１０．２０．３０．４０」であるＶＭで稼働し、使用するポート番号が「８０」であるアプリの「２０１６／０６／２４　０９：００：００」

の応答時間は、「６００」である。

【００４１】

正規化部２８は、応答時間算出部２６により算出された応答時間を応答時間情報記憶部２７から読み出し、アプリ毎の平均応答時間を算出する。そして、正規化部２８は、代表情報記憶部２９に記憶された情報を用いて平均応答時間の正規化を行い、正規化した平均応答時間を遅延度合情報として遅延度合情報記憶部３０に格納する。

【００４２】

図８は、平均応答時間の正規化を説明するための図である。応答時間は、アプリによって、正常時にとりうる値や、遅延とみなす基準が異なる。このため、アプリの応答時間をそのまま用いただけでは、アプリに遅延があるのか判断することは困難である。そこで、遅延情報作成装置２は、平均応答時間を正規化して、アプリ間で比較可能な尺度に変換する。図８では、正常時にとりうる値が異なるアプリ＃１とアプリ＃２について、平均応答時間を正規化することによって、応答時間の比較が可能となっている。

【００４３】

正規化部２８は、平均応答時間ｔの基本統計量t_rを代表応答時間として、正規化された平均応答時間t_nを$t_n = t / t_r$により計算する。基本統計量には、平均、中央値、最頻値等がある。

【００４４】

代表情報記憶部２９は、代表応答時間をアプリ毎に記憶する。図９は、代表情報記憶部２９の一例を示す図である。図９に示すように、代表情報記憶部２９は、時刻と、ＩＰアドレスと、ポート番号と、応答時間をアプリ毎に記憶する。

【００４５】

時刻は、代表応答時間が計算された時刻である。ＩＰアドレスは、アプリが稼働しているＶＭのＩＰアドレスである。ポート番号は、アプリが使用するポート番号である。応答時間は、代表応答時間である。代表応答時間の単位はマイクロ秒である。例えば、ＩＰアドレスが「１０．２０．３０．４０」であるＶＭで稼働し、使用するポート番号が「８０」であるアプリの「２０１６／０６／２３　００：００：００」に計算された代表応答時間は、「６００」である。

【００４６】

正規化部２８は、アプリ毎に一定時間間隔（例えば１分）毎の平均応答時間ｔを計算し、平均応答時間ｔの基本統計量t_rを計算する。基本統計量を計算するためのデータとしては前日の１日分のデータ等が用いられる。一定時間間隔が１分の場合は、１日分のデータから６０＊２４＝１４４０のサンプルデータが得られる。

【００４７】

また、正規化部２８は、基本統計量t_rの代わりに、平均応答時間の分布をｅｘ－Ｇａｕｓｓｉａｎ分布でフィッティングさせたときのμを代表応答時間としてもよい。図１０は、ｅｘ－Ｇａｕｓｓｉａｎ分布を説明するための図である。ｅｘ－Ｇａｕｓｓｉａｎ分布は、確率分布の一つであり、図１０（ａ）に示すように、ガウス分布（正規分布）と指数分布を畳込み積分したものである。ｅｘ－Ｇａｕｓｓｉａｎ分布は、正規分布の平均μ及び標準偏差σと、指数分布の平均及び標準偏差τの３つのパラメータにより決定される。ｅｘ－Ｇａｕｓｓｉａｎ分布では、μは、図１０（ｂ）に示すように、分布のピークの部分の値になる。

【００４８】

正規化部２８は、アプリ毎に一定時間間隔（例えば１分）毎の平均応答時間ｔを計算し、平均応答時間の分布をｅｘ－Ｇａｕｓｓｉａｎ分布でフィッティングする。ｅｘ－Ｇａｕｓｓｉａｎ分布へのフィッティングのためのデータとしては前日の１日分のデータ等が用いられる。一定時間間隔が１分の場合は、１日分のデータから６０＊２４＝１４４０のサンプルデータが得られる。

【００４９】

そして、正規化部２８は、フィッティングの確からしさを一標本コルモゴロフ－スミル

ノフ検定で判定する。一標本コルモゴロフ－スミルノフ検定の入力は、平均応答時間の分布と、フィッティング結果の分布曲線との二つである。正規化部２８は、有意水準を例えば０．０５で検定し、検定した結果、平均応答時間の分布がｅｘ－Ｇａｕｓｓｉａｎ分布であった場合、ｅｘ－Ｇａｕｓｓｉａｎ分布のμを代表平均応答時間とする。

【００５０】

なお、正規化部２８は、平均応答時間の分布をｅｘ－Ｇａｕｓｓｉａｎ分布でフィッティングする前に外れ値除去を行ってもよい。図１１は、外れ値を説明するための図である。図１１に示すように、外れ値は、他の値から大きく外れた値である。平均応答時間に外れ値がある場合、ｅｘ－Ｇａｕｓｓｉａｎ分布でうまくフィッティングできない場合がある。このため、正規化部２８は、フィッティングの前に外れ値除去を行う。外れ値除去の方法にはＴｕｋｅｙの外れ値除去等がある。

【００５１】

遅延度合情報記憶部３０は、正規化部２８により正規化された平均応答時間を遅延度合情報としてアプリ毎に記憶する。図１２は、遅延度合情報記憶部３０の一例を示す図である。図１２に示すように、遅延度合情報記憶部３０は、時刻と、ＩＰアドレスと、ポート番号と、正規化平均応答時間と、リクエスト数とをアプリ毎に記憶する。

【００５２】

時刻は、応答時間が算出された時刻である。ＩＰアドレスは、アプリが稼働しているＶＭのＩＰアドレスである。ポート番号は、アプリが使用するポート番号である。正規化平均応答時間は、正規化された平均応答時間である。リクエスト数は、正規化された平均応答時間の計算に用いられたリクエストの数である。

【００５３】

例えば、ＩＰアドレスが「１０．２０．３０．４０」であるＶＭで稼働し、使用するポート番号が「８０」のアプリの「２０１６／０６／２４　０９：００：００」に算出された応答時間に関して正規化された平均応答時間は「１．０」である。正規化された平均応答時間の計算に用いられたリクエストの数は「２」である。

【００５４】

図２に戻って、診断装置４は、遅延判定部４１と、第１相関分析部４２と、第１分析結果記憶部４３と、第２相関分析部４４と、第２分析結果記憶部４５と、遅延原因診断部４６と、診断結果記憶部４７とを有する。

【００５５】

遅延判定部４１は、遅延情報作成装置２から正規化応答時間を遅延度合情報として取得し、アプリに遅延が発生しているか否かを判定する。図１３は、アプリの遅延判定を説明するための図である。図１３に示すように、遅延判定部４１は、アプリ毎に、診断対象期間のデータにおいて、遅延度合が閾値T_{rt}以上になった回数を定期的にカウントし、回数が閾値T_{dr}以上の場合に、アプリが遅延していると判定する。

【００５６】

判定を実行する間隔は、例えば１日である。あるいは、遅延判定部４１は、クラウド運用管理者の指示に基づいて遅延判定を行ってもよい。図１３では、T_{rt}＝１０、T_{dr}＝５として、アプリ＃２の遅延度合がT_{rt}以上の回数が「５」であるので、アプリ＃２に遅延が発生していると判定される。

【００５７】

第１相関分析部４２は、遅延判定部４１により遅延が発生していると判定された各アプリについて、アプリに関係するＶＭのリソースデータと遅延度合との相関分析を行うことによって、遅延と関係するリソースデータを絞り込む。第１相関分析部４２は、ＶＭ情報記憶部３１を参照してアプリに関係するＶＭを特定する。

【００５８】

図１４は、ＶＭ情報記憶部３１の一例を示す図である。図１４に示すように、ＶＭ情報記憶部３１は、変更日時、ＶＭのＩＰアドレス、ポート番号、関係するＶＭリストを対応付けてアプリ毎に記憶する。変更日時は、対応する情報が変更された日時である。ＶＭの

ＩＰアドレスは、アプリが動作するＶＭのＩＰアドレスである。ポート番号は、アプリが使用するポートの番号である。ＶＭのＩＰアドレスとポート番号の組合せで１つのアプリが特定される。

【００５９】

関係するＶＭリストは、ＶＭのＩＰアドレスとポート番号で特定されるアプリに関係するＶＭのリストである。例えば、ＩＰアドレス「１０．２０．３０．４０」とポート番号「８０」で特定されるアプリが関係するＶＭは、「ＶＭ＃１」と「ＶＭ＃２」と「ＶＭ＃３」である。

【００６０】

第１相関分析部４２は、リソースデータ記憶部３２からＶＭのリソースデータを取得する。図１５は、リソースデータ記憶部３２の一例を示す図である。図１５に示すように、リソースデータ記憶部３２は、時刻、ＶＭ名、リソースデータ名、値を対応付けてリソースデータ毎に記憶する。

【００６１】

時刻は、値が取得された時刻である。ＶＭ名は、リソースデータがどのＶＭのものであるか示す。リソースデータ名は、リソースデータを識別する名前である。値は、リソースデータの値である。例えば、「２０１６／０６／２４　０９：００：００」において、「ＶＭ＃１」の「ＣＰＵ使用率」は「１０．０」％である。

【００６２】

なお、リソースデータ記憶部３２には、リソース３４から取得されたデータが記憶される。また、第１相関分析部４２は、アプリが関係するＶＭの情報が得られない場合には、アプリのユーザが利用する全ＶＭのリソースデータとの相関分析を行う。

【００６３】

第１相関分析部４２は、相関係数を利用して相関分析を行う。すなわち、第１相関分析部４２は、遅延度合とリソースデータとの相関係数を計算し、遅延度合とリソースデータとの間で正の相関があるかを検定する。図１６は、リソースデータと遅延度合の関係の例を示す図である。図１６では、１つの時刻のリソースデータの値と遅延度合の値の組が点で表される。第１相関分析部４２は、図１６に示す複数の点の値を用いて相関係数を計算する。

【００６４】

第１相関分析部４２は、相関係数として例えばピアソンの相関係数、スピアマンの順位相関係数を用いる。そして、第１相関分析部４２は、無相関検定のｐ－ｖａｌｕｅを計算する。

【００６５】

第１相関分析部４２は、アプリに関係するＶＭのリソースデータ数の回数の検定を行う。このように、複数回の検定が実行されると、本来相関がないにもかかわらず相関があると判定されるケースの数が増加する。例えば、有意水準が０．０５の場合、１回の検定で、相関がないにもかかわらず、相関があると間違えて判定される確率は５％となる。この検定を３回繰り返すと、１度でも間違える可能性は$1-(1-0.05)^3=14\%$にまで増加する。

【００６６】

そこで、第１相関分析部４２は、誤った判定を減らすために、計算したｐ－ｖａｌｕｅを多重検定補正によって補正する。補正方法には、例えば、Benjamini-Hochberg法がある。第１相関分析部４２は、補正したｐ－ｖａｌｕｅが閾値以下であるリソースデータを遅延度合と相関があると判定する。閾値としては、例えば０．０５が用いられる。

【００６７】

なお、第１相関分析部４２は、相関係数を用いる相関分析の代わりに、遅延しているときと遅延していないときのリソースデータの傾向に差があるか否かを判定することによって相関分析を行ってもよい。相関係数を用いる場合には、全てのデータが等価に扱われる。このため、相関しているかどうかは数が多い方のデータすなわち遅延していないデータ

で決まってしまう場合がある。この結果、遅延しているときにリソースデータの値が大きいものを見逃す場合がある。そこで、遅延しているときのリソースデータを分けて扱うことで、第１相関分析部４２は、データ数の影響を抑えて、相関しているか否かを判断することができる。

【００６８】

図１７は、遅延しているときと遅延していないときのリソースデータの傾向に差があるか否かの判定による相関分析を説明するための図である。図１７に示すように、第１相関分析部４２は、遅延度合が閾値T_{rt}以上のとき遅延しているとして、遅延しているときと遅延していないときで、リソースデータを分ける。そして、第１相関分析部４２は、遅延しているときのリソースデータのほうが、遅延していないときよりも大きくなる傾向にあるかの検定（二群比較検定）のｐ－ｖａｌｕｅを計算する。検定方法には、ｔ－検定、Wilcoxonの順位和検定、Mann-WhitneyのＵ検定等がある。

【００６９】

そして、第１相関分析部４２は、計算したｐ－ｖａｌｕｅを多重検定補正によって補正する。補正方法には、例えば、Benjamini-Hochberg法がある。第１相関分析部４２は、補正したｐ－ｖａｌｕｅが閾値以下であるリソースデータを遅延度合と相関があると判定する。閾値としては、例えば０．０５が用いられる。

【００７０】

相関分析により相関があると検定された場合でも、相関の強さはデータ毎に異なる。図１８Ａは、相関が弱い場合と強い場合を示す図である。図１８Ａに示すように、リソースデータ＃１は遅延度合との相関が弱いが、リソースデータ＃２は遅延度合との相関が強い。そこで、第１相関分析部４２は、アプリの遅延原因のリソース候補を減らすために、相関があるリソースデータのうち、遅延度合とよく相関のあるリソースデータだけを抽出する。

【００７１】

図１８Ｂは、遅延度合とよく相関のあるリソースデータの抽出を説明するための図である。図１８Ｂに示すように、第１相関分析部４２は、遅延しているときのデータを利用し、遅延しているときのリソースデータの値を推定する。そして、第１相関分析部４２は、推定値よりもリソースデータが大きいときの遅延度合の分布を２つのリソースデータについて比較し、リソースデータが大きいときによく遅延しているリソースデータを特定する。図１８Ｂでは、リソースデータ＃２が遅延度合とよく相関のあるリソースデータとして抽出される。

【００７２】

具体的には、第１相関分析部４２は、遅延度合と相関のあったリソースデータ毎に以下の手順（１）、（２）を実行する。

（１）第１相関分析部４２は、遅延度合が閾値T_{rt}以上のリソースデータの中央値を計算し、T_rとする（左斜線網掛けのデータを利用）。

（２）そして、第１相関分析部４２は、リソースデータがT_r以上の時の遅延度合のデータを抽出する（右斜線網掛けのデータを抽出）。

【００７３】

そして、第１相関分析部４２は、抽出された遅延度合のデータをSteel Dwass法で検定し、２つのリソースデータの遅延度合の分布に差があると出た回数の最も多いリソースデータを特定する。検定の有意水準には、例えば０．０５が用いられる。そして、第１相関分析部４２は、差があるとされた回数が最も多いリソースデータと、そのリソースデータとの差がないとされるリソースデータを遅延度合とよく相関があるリソースデータとする。

【００７４】

第１相関分析部４２は、相関分析の結果を第１分析結果記憶部４３に格納する。図１９は、第１分析結果記憶部４３の一例を示す図である。図１９に示すように、第１分析結果記憶部４３は、診断日時、開始時間、終了時間、ＶＭのＩＰアドレス、ポート番号、ＶＭ

名、リソース種類、リソースデータ名を対応付けて記憶する。

【００７５】

診断日時は、診断を行った日時である。開始時間は、診断対象期間の開始時間である。終了時間は、診断対象期間の終了時間である。ＶＭのＩＰアドレスは、アプリが動作するＶＭのＩＰアドレスである。ポート番号は、アプリが使用するポートの番号である。ＶＭ名は、リソースデータがどのＶＭのものであるか示す。リソース種類は、遅延に関係のあるリソース３４の種類であり、「ＣＰＵ」、「ディスク装置」、「ネットワーク」等がある。リソースデータ名は、遅延度合と相関のあったリソースデータを識別する名前である。

【００７６】

例えば、「２０１６／０５／２４　００：００：００」～「２０１６／０６／２４　００：００：００」を診断対象期間として「２０１６／０６／２４　００：００：００」に行った診断において、アプリの遅延度合と相関のあるリソースデータがあった。アプリは、ＶＭのＩＰアドレス「１０．２０．３０．４０」とポート番号「８０」で特定され、「ＶＭ＃３」の「ＣＰＵの割り当て待ち時間の割合」とアプリの遅延度合との間に相関があった。

【００７７】

第２相関分析部４４は、リソース性能低下の影響を受けるデータと遅延度合の相関があった場合に、リソース性能低下の影響を受けるデータの相関があったＶＭとリソース３４を共有している全ＶＭについてリソース負荷量を表すデータと遅延度合との相関分析を行う。

【００７８】

第２相関分析部４４は、リソースデータ記憶部３２と、クラウドシステム１の構成情報を記憶する構成情報記憶部３３とを参照して相関分析を行う。図２０は、構成情報記憶部３３の一例を示す図である。図２０に示すように、構成情報記憶部３３は、変更日時、物理マシン、利用ユーザ、ＶＭ名、ＶＭのＩＰアドレス、利用ディスク名を対応付けてＶＭ毎に記憶する。

【００７９】

変更日時は、対応する情報が変更された日時である。物理マシンは、ＶＭが動作する物理マシンである。利用ユーザは、ＶＭを利用するユーザである。ＶＭ名は、ＶＭを識別する名前である。ＶＭのＩＰアドレスは、ＶＭに付与されたＩＰアドレスである。利用ディスク名は、ＶＭが利用するディスク装置の名前である。例えば、「ＶＭ＃１」は「ＰＭ＃１」で動作して「ディスク＃１」を利用し、「ＶＭ＃１」のユーザは「ユーザ＃１」であり、「ＶＭ＃１」のＩＰアドレスは「１０．２０．３０．４０」である。「ＶＭ＃１」の情報は「２０１６／０６／２４　００：００：００」に変更された。

【００８０】

図２１は、第２相関分析部４４による相関分析を説明するための図である。リソース性能低下の影響を受けるデータと遅延度合の相関があった場合、リソース３４の性能低下の影響を受けてアプリが遅延したと考えられる。このため、第２相関分析部４４は、リソース性能低下の影響を受けるデータの相関があったＶＭとリソース３４を共有する全ＶＭ（リソース性能低下の影響を受けるデータの相関があったＶＭ自体を含む）を構成情報記憶部３３を参照して特定する。図２１では、ユーザ＃１のＶＭがリソース性能低下の影響を受けるデータの相関がアプリ＃１の遅延度合とあったＶＭである。また、リソース性能低下の影響を受けるデータの相関があったＶＭとリソース３４を共有する他のＶＭとしてユーザ＃２のＶＭが特定される。

【００８１】

そして、第２相関分析部４４は、特定した各ＶＭのリソース負荷量を表すデータと遅延度合との相関分析をリソースデータ記憶部３２を参照して行うことで、特定したＶＭの中から負荷をかけているＶＭを特定する。リソース負荷量を表すデータと遅延度合との相関があるＶＭが負荷をかけているＶＭである。図２１では、ユーザ＃２のＶＭのＣＰＵ使用

率とアプリ＃１の遅延度合との相関が分析される。

【００８２】

そして、第２相関分析部４４は、相関分析により相関のあったリソースデータのうち、遅延度合とよく相関のあるリソースデータを抽出する。そして、第２相関分析部４４は、抽出したリソースデータについて、相関分析の結果を第２分析結果記憶部４５に格納する。

【００８３】

図２２は、第２分析結果記憶部４５の一例を示す図である。図２２に示すように、第２分析結果記憶部４５は、診断日時、開始時間、終了時間、ＶＭのＩＰアドレス、ポート番号、アプリＶＭ、リソース種類、相関ＶＭ、リソース負荷量データを対応付けて記憶する。

【００８４】

診断日時は、診断を行った日時である。開始時間は、診断対象期間の開始時間である。終了時間は、診断対象期間の終了時間である。ＶＭのＩＰアドレスは、アプリが動作するＶＭのＩＰアドレスである。ポート番号は、アプリが使用するポートの番号である。アプリＶＭは、リソース性能低下の影響を受けているデータと遅延度合の相関があったアプリが動作するＶＭである。リソース種類は、遅延に関係のあるリソース３４の種類である。相関ＶＭは、リソース負荷量を表すデータと遅延度合との相関があったＶＭである。リソース負荷量データは、遅延度合と相関のあったリソース負荷量を表すデータである。

【００８５】

例えば、「２０１６／０５／２４　００：００：００」～「２０１６／０６／２４　００：００：００」を診断対象期間として「２０１６／０６／２４　００：００：００」に診断が行われた。遅延が発生したアプリは、ＶＭのＩＰアドレス「１０．２０．３０．４０」とポート番号「８０」で特定され、「ＶＭ＃３」で動作する。影響を受けているリソース３４の種類は「ＣＰＵ」であり、「ＶＭ＃３」と「ＣＰＵ」を共有する「ＶＭ＃４」の「ＣＰＵ使用率」が遅延度合と相関があった。

【００８６】

遅延原因診断部４６は、第１分析結果記憶部４３と第２分析結果記憶部４５を参照してＶＭのリソースデータの種類毎にアプリの遅延原因を判定する。具体的には、遅延原因診断部４６は、アプリに関係するＶＭのリソース負荷量を表すデータとの相関があった場合は、常に時間のかかるリクエストを処理したことにより、平均応答時間が増加し、遅延しているので、「アプリの要因による遅延」と判定する。

【００８７】

また、遅延原因診断部４６は、アプリに関係するＶＭのリソース性能低下の影響を受けるデータとの相関があった場合は、以下の３つの場合に分けて判定する。第１に、同じＶＭのリソース負荷量を表すデータとも相関がある場合は、遅延しているアプリのＶＭ自身がリソース３４への負荷が上昇させ、遅延が発生しているので、遅延原因診断部４６は、「アプリの要因による遅延」と判定する。第２に、リソース３４を共有している他ＶＭのリソース負荷量を表すデータと相関がある場合は、他ＶＭがリソース３４を大量に使ったことによりリソース負荷が上昇し、遅延が発生しているので、遅延原因診断部４６は、「クラウド基盤要因による遅延」と判定する。第３に、リソース３４を共有しているＶＭのリソース負荷量を表すデータと相関がなかった場合は、遅延原因診断部４６は、「リソース負荷上昇の原因不明」と判定する。

【００８８】

相関のあったリソース３４が複数ある場合は、遅延原因診断部４６は、遅延原因を複数出力する。また、遅延原因診断部４６は、全てのＶＭで相関のあったリソースデータがない場合、もしくは逆に多い場合（遅延原因の数が閾値T_d以上の場合）、原因を特定できなかったと判定する。

【００８９】

そして、遅延原因診断部４６は、判定結果を診断結果記憶部４７に格納する。図２３は

、診断結果記憶部４７の一例を示す図である。図２３に示すように、診断結果記憶部４７は、診断日時、開始時間、終了時間、ＶＭのＩＰアドレス、ポート番号、遅延割合、アプリＶＭ、リソース種類、リソースデータ、負荷ＶＭ、リソース負荷量データ、診断結果を対応付けて記憶する。

【００９０】

診断日時は、診断を行った日時である。開始時間は、診断対象期間の開始時間である。終了時間は、診断対象期間の終了時間である。ＶＭのＩＰアドレスは、アプリが動作するＶＭのＩＰアドレスである。ポート番号は、アプリが使用するポートの番号である。遅延割合は、遅延が発生している合計時間の割合である。アプリＶＭは、相関のあったアプリが動作するＶＭである。リソース種類は、遅延に関係のあるリソース３４の種類である。リソースデータは、相関のあったアプリのＶＭのリソースデータである。負荷ＶＭは、リソース３４を共有していて負荷をかけているＶＭである。リソース負荷量データは、リソース３４を共有していて負荷をかけているＶＭの相関のあったリソース負荷量データである。診断結果は、遅延原因診断部４６による判定結果である。

【００９１】

例えば、「２０１６／０５／２４　００：００：００」～「２０１６／０６／２４　００：００：００」を診断対象期間として「２０１６／０６／２４　００：００：００」に診断が行われた。合計時間で「０．１０」の割合で遅延が発生したアプリは、ＶＭのＩＰアドレス「１０．２０．３０．４０」とポート番号「８０」で特定され、「ＶＭ＃３」で動作する。遅延に関係のあるリソース３４の種類は「ＣＰＵ」である。「ＣＰＵの割り当て待ち時間の割合」が遅延度合と相関があり、「ＶＭ＃３」と「ＣＰＵ」を共有する「ＶＭ＃４」の「ＣＰＵ使用率」が遅延度合と相関のあったリソース負荷量データであり、遅延原因は「クラウド基盤要因」である。

【００９２】

また、遅延原因診断部４６は、アプリの遅延要因として「クラウド基盤要因」が含まれている場合、遅延していたアプリ、アプリの遅延要因、遅延と関係するＶＭのリソース３４等をクラウド運用管理者に通知する。図２４は、クラウド運用管理者への通知例を示す図である。図２４では、Ｗｅｂサーバ、アプリサーバ、ＤＢサーバから成る３階層システムのＤＢサーバが他アプリのＶＭのＣＰＵ負荷の影響を受けている場合を示す。

【００９３】

図２４に示すように、アプリケーション、診断対象期間、遅延が発生している合計時間の割合、遅延原因、リソース負荷の影響を受けているアプリケーションのＶＭとリソース３４、リソース負荷を与えているＶＭとリソース３４がクラウド運用管理者に通知される。

【００９４】

次に、診断装置４による処理のフローについて図２５～図３１を用いて説明する。図２５は、診断装置４による処理のフローを示すフローチャートである。図２５に示すように、診断装置４は、アプリ数だけステップＳ１～ステップＳ５の処理を繰り返す。すなわち、診断装置４は、アプリの診断対象期間の遅延度合情報を取得する（ステップＳ１）。

【００９５】

そして、診断装置４は、遅延度合が閾値T_{rt}以上の回数がT_{dr}以上か否かを判定し（ステップＳ２）、T_{dr}以上でない場合には、次のアプリを処理する。一方、T_{dr}以上である場合には、診断装置４は、第１相関分析処理を行う（ステップＳ３）。ここで、第１相関分析処理は、第１相関分析部４２が行う処理である。

【００９６】

そして、診断装置４は、第２相関分析処理を行う（ステップＳ４）。ここで、第２相関分析処理は、第２相関分析部４４が行う処理である。なお、ステップＳ４の処理は、リソース性能低下の影響を受けるデータと遅延度合との相関があった場合に行われる。そして、診断装置４は、遅延原因診断処理を行う（ステップＳ５）。ここで、遅延原因診断処理は、遅延原因診断部４６が行う処理である。

【００９７】
そして、アプリ数だけステップＳ１～ステップＳ５の処理を繰り返すと、診断装置４は、クラウド基盤要因により応答遅延が発生と診断されたアプリがあるか否かを判定し（ステップＳ６）、ある場合には、クラウド運用管理者に通知する（ステップＳ７）。
【００９８】
このように、クラウド基盤要因により応答遅延が発生と診断されたアプリがある場合に診断装置４がクラウド運用管理者に通知することで、クラウド運用管理者はクラウド基盤に関して対策を検討することができる。
【００９９】
図２６は、第１相関分析処理のフローを示すフローチャートである。図２６に示すように、第１相関分析部４２は、ＶＭ情報記憶部３１からアプリに関係するＶＭの情報を取得し（ステップＳ１１）、リソースデータ記憶部３２からアプリに関係する全ＶＭのリソースデータを取得する（ステップＳ１２）。
【０１００】
そして、第１相関分析部４２は、アプリの遅延度合とリソースデータとの無相関検定のｐ－ｖａｌｕｅを計算する（ステップＳ１３）処理をリソースデータ数だけ繰り返す。そして、第１相関分析部４２は、多重検定補正を実行し（ステップＳ１４）、多重検定補正により補正したｐ－ｖａｌｕｅが閾値以下のリソースデータのみを抽出する（ステップＳ１５）。
【０１０１】
そして、第１相関分析部４２は、抽出したリソースデータから、遅延度合とよく相関しているリソースデータを抽出するリソースデータ抽出処理を行う（ステップＳ１６）。そして、第１相関分析部４２は、第１分析結果記憶部４３に相関分析結果を保存する（ステップＳ１７）。
【０１０２】
このように、第１相関分析部４２は、アプリの遅延度合とリソースデータとの無相関検定のｐ－ｖａｌｕｅを計算し、多重検定補正によりｐ－ｖａｌｕｅを補正し、補正したｐ－ｖａｌｕｅが閾値以下のリソースデータのみを抽出する。そして、第１相関分析部４２は、抽出したリソースデータから、さらに、遅延度合とよく相関しているリソースデータを抽出する。したがって、遅延度合と相関のあるリソースデータを正確に抽出することができる。
【０１０３】
図２７は、リソースデータ抽出処理のフローを示すフローチャートである。図２７に示すように、第１相関分析部４２は、相関のあるリソースデータ数だけステップＳ２１～ステップＳ２２の処理を繰り返す。すなわち、第１相関分析部４２は、遅延度合が閾値T_{rt}以上のときのリソースデータの中央値を計算し閾値T_rとし（ステップＳ２１）、リソースデータが閾値T_r以上のときの遅延度合のデータを抽出する（ステップＳ２２）。
【０１０４】
そして、第１相関分析部４２は、リソースデータ毎に抽出された遅延度合を入力として多重比較検定を行う（ステップＳ２３）。そして、第１相関分析部４２は、検定の結果、他のリソースデータの遅延度合の分布よりも大きくなる傾向にあるとなった回数が最も多いリソースデータを特定する（ステップＳ２４）。
【０１０５】
そして、第１相関分析部４２は、抽出された遅延度合の分布において、特定されたリソースデータのものと差がないと判定されたリソースデータと、特定されたリソースデータを遅延度合とよく相関があるリソースデータとして出力する（ステップＳ２５）。
【０１０６】
このように、リソースデータ抽出処理により遅延度合とよく相関があるリソースデータを抽出することで、第１相関分析部４２は、遅延度合と相関が強いリソースデータだけを抽出することができる。なお、リソースデータ抽出処理は、後述するように第２相関分析

部４４により呼び出された場合には、第２相関分析部４４が行う。

【０１０７】

図２８は、第２相関分析処理のフローを示すフローチャートである。図２８に示すように、第２相関分析部４４は、リソース性能低下の影響を受けるデータの相関のあったリソース数だけステップＳ３１～ステップＳ３７の処理を繰り返す。

【０１０８】

すなわち、第２相関分析部４４は、注目リソースを共有しているＶＭの情報を取得する（ステップＳ３１）。ここで、注目リソースとは、ステップＳ３１～ステップＳ３７の１回の処理の対象となるリソース３４である。そして、第２相関分析部４４は、注目リソースを共有している全ＶＭのリソース負荷量を表すデータを取得する（ステップＳ３２）。全ＶＭには、注目リソースのＶＭも含まれる。

【０１０９】

そして、第２相関分析部４４は、アプリの遅延度合とリソース負荷量を表すデータとの無相関検定のｐ－ｖａｌｕｅを計算する処理（ステップＳ３３）をリソース３４を共有している全ＶＭのリソース負荷量を表すデータ数だけ繰り返す。そして、第２相関分析部４４は、多重検定補正を実行し（ステップＳ３４）、多重検定補正により補正したｐ－ｖａｌｕｅが閾値以下のリソースデータのみを抽出する（ステップＳ３５）。そして、第２相関分析部４４は、リソースデータ抽出処理を行い（ステップＳ３６）、第２分析結果記憶部４５に相関分析結果を保存する（ステップＳ３７）。

【０１１０】

このように、第２相関分析部４４は、リソース性能低下の影響を受けるデータの相関のあった各リソース３４について、リソース３４を共有する全ＶＭのリソース負荷量を表すデータとの無相関検定のｐ－ｖａｌｕｅを計算する。そして、第２相関分析部４４は、多重検定補正によりｐ－ｖａｌｕｅを補正し、補正したｐ－ｖａｌｕｅが閾値以下のリソースデータのみを抽出する。そして、第２相関分析部４４は、抽出したリソースデータから、さらに、遅延度合とよく相関しているリソースデータを抽出する。したがって、第２相関分析部４４は、リソース性能低下の影響を受けるデータの相関のあった各リソース３４について、リソース３４を共有する全ＶＭのリソース負荷量を表すデータの遅延度合との相関分析を正確に行うことができる。

【０１１１】

図２９は、遅延原因診断処理のフローを示すフローチャートである。図２９に示すように、遅延原因診断部４６は、遅延度合が閾値T_{rt}以上の回数から遅延していた合計時間の割合を計算する（ステップＳ４１）。そして、遅延原因診断部４６は、相関のあるリソースデータを持つアプリのＶＭ数だけステップＳ４２～ステップＳ４９の処理を繰り返す。

【０１１２】

すなわち、遅延原因診断部４６は、注目ＶＭのリソース負荷量を表すデータとの相関があるか否かを判定する（ステップＳ４２）。ここで、注目ＶＭとは、ステップＳ４２～ステップＳ４９の１回の処理の対象となるＶＭである。そして、遅延原因診断部４６は、注目ＶＭのリソース負荷量を表すデータとの相関がある場合には、アプリ要因による遅延と判定し（ステップＳ４３）、次のＶＭの処理に進む。

【０１１３】

一方、注目ＶＭのリソース負荷量を表すデータとの相関がない場合には、遅延原因診断部４６は、注目ＶＭのリソース性能低下の影響を受けるデータとの相関があるか否かを判定し（ステップＳ４４）、否の場合には次のＶＭの処理に進む。一方、注目ＶＭのリソース性能低下の影響を受けるデータとの相関がある場合には、遅延原因診断部４６は、リソース３４を共有している全ＶＭについてリソース負荷量を表すデータとの相関があるか否かを判定する（ステップＳ４５）。そして、遅延原因診断部４６は、リソース負荷量を表すデータと遅延度合との相関があるＶＭが１つもない場合には、リソース負荷上昇の原因不明と判定し（ステップＳ４６）、次のＶＭの処理に進む。

【０１１４】

また、遅延原因診断部４６は、リソース負荷量を表すデータと遅延度合との相関がある各ＶＭについて、ステップＳ４７～ステップＳ４９の処理を行う。すなわち、遅延原因診断部４６は、相関のあるリソース性能低下の影響を受けるデータとリソース負荷量を表すデータが同じＶＭのものか否かを判定する（ステップＳ４７）。

【０１１５】

そして、相関のあるリソース性能低下の影響を受けるデータとリソース負荷量を表すデータが同じＶＭのものである場合には、遅延原因診断部４６は、アプリ要因による遅延と判定し（ステップＳ４８）、次のＶＭの処理に進む。一方、相関のあるリソース性能低下の影響を受けるデータとリソース負荷量を表すデータが同じＶＭのものでない場合には、遅延原因診断部４６は、クラウド基盤要因による遅延と判定し（ステップＳ４９）、次のＶＭの処理に進む。

【０１１６】

そして、ステップＳ４２～ステップＳ４９の繰り返し処理を完了すると、遅延原因診断部４６は、判定結果の数が１以上かつ閾値T_d以下か否かを判定する（ステップＳ５０）。そして、判定結果の数が１以上かつ閾値T_d以下である場合には、遅延原因診断部４６は、遅延と関係するリソース３４を特定できたとして判定結果を出力する（ステップＳ５１）。一方、判定結果の数が１以上かつ閾値T_d以下ではない場合には、遅延原因診断部４６は、遅延と関係するリソース３４を特定できなかったとする（ステップＳ５２）。

【０１１７】

そして、遅延原因診断部４６は、遅延診断結果を保存する（ステップＳ５３）。このように、遅延原因診断部４６が、第１相関分析部４２及び第２相関分析部４４による分析結果に基づいて遅延の原因を特定することで、診断装置４は、クラウド基盤に問題がある場合に、問題のあるリソース３４を特定することができる。

【０１１８】

次に、二群比較検定を用いて相関分析を行う場合の第１相関分析処理及び第２相関分析処理のフローについて図３０及び図３１を用いて説明する。図３０は、二群比較検定を用いる第１相関分析処理のフローを示すフローチャートである。

【０１１９】

図３０に示すように、第１相関分析部４２は、ＶＭ情報記憶部３１からアプリに関係するＶＭの情報を取得し（ステップＳ６１）、リソースデータ記憶部３２からアプリに関係する全ＶＭのリソースデータを取得する（ステップＳ６２）。

【０１２０】

そして、第１相関分析部４２は、遅延度合が閾値T_{rt}未満のときのリソースデータと閾値T_{rt}以上のリソースデータに分け（ステップＳ６３）、二群比較検定によりｐ－ｖａｌｕｅを計算する（ステップＳ６４）処理をリソースデータ数だけ繰り返す。そして、第１相関分析部４２は、多重検定補正を実行し（ステップＳ６５）、多重検定補正により補正したｐ－ｖａｌｕｅが閾値以下のリソースデータのみを抽出する（ステップＳ６６）。

【０１２１】

そして、第１相関分析部４２は、リソースデータ抽出処理を行い（ステップＳ６７）、第１分析結果記憶部４３に相関分析結果を保存する（ステップＳ６８）。

【０１２２】

図３１は、二群比較検定を用いる第２相関分析処理のフローを示すフローチャートである。図３１に示すように、第２相関分析部４４は、リソース性能低下の影響を受けるデータの相関のあったリソース数だけステップＳ７１～ステップＳ７８の処理を繰り返す。

【０１２３】

すなわち、第２相関分析部４４は、注目リソースを共有しているＶＭの情報を取得し（ステップＳ７１）、注目リソースを共有している全ＶＭのリソース負荷量を表すデータを取得する（ステップＳ７２）。

【０１２４】

そして、第２相関分析部４４は、リソース３４を共有している全ＶＭのリソース負荷量

を表すデータ数だけステップＳ７３とステップＳ７４の処理を繰り返す。すなわち、第２相関分析部４４は、遅延度合が閾値$Ｔ_{rt}$未満のときのリソースデータと閾値$Ｔ_{rt}$以上のリソースデータに分け（ステップＳ７３）、二群比較検定によりｐ－ｖａｌｕｅを計算する（ステップＳ７４）。

【０１２５】

そして、第２相関分析部４４は、多重検定補正を実行し（ステップＳ７５）、多重検定補正により補正したｐ－ｖａｌｕｅが閾値以下のリソースデータのみを抽出する（ステップＳ７６）。そして、第２相関分析部４４は、リソースデータ抽出処理を行い（ステップＳ７７）、第２分析結果記憶部４５に相関分析結果を保存する（ステップＳ７８）。

【０１２６】

このように、第１相関分析部４２及び第２相関分析部４４は、遅延度合が閾値$Ｔ_{rt}$未満のときのリソースデータと閾値$Ｔ_{rt}$以上のリソースデータに分け、二群比較検定によりｐ－ｖａｌｕｅを計算することで、遅延していないデータの影響を抑えて相関分析を行うことができる。

【０１２７】

上述してきたように、実施例１では、遅延判定部４１が遅延度合情報を遅延情報作成装置２から取得してアプリに遅延が発生したか否かを判定し、第１相関分析部４２が遅延が発生したアプリに関係するＶＭのリソースデータと遅延度合との相関分析を行う。そして、リソース性能低下の影響を受けるデータと遅延度合との相関がある場合に、第２相関分析部４４がリソース３４を共有している全ＶＭについて、リソース負荷量を表すデータと遅延度合との相関関係を分析する。そして、遅延原因診断部４６が、第１相関分析部４２と第２相関分析部４４の相関分析結果に基づいて遅延原因を特定する。

【０１２８】

したがって、診断装置４は、アプリの性能低下がクラウド基盤に原因があるのかアプリに原因があるのかを特定することができる。また、診断装置４は、クラウド基盤に原因がある場合に、どのリソース３４に原因があるかを特定することができる。

【０１２９】

また、実施例では、第１相関分析部４２は、遅延しているときと遅延していないときのリソースデータの傾向に差があるか否かを判定することで相関分析を行う場合には、遅延していないデータの影響を抑えて相関分析を行うことができる。

【０１３０】

また、実施例では、第１相関分析部４２は、遅延度合との相関係数を計算し、無相関検定を行ってｐ－ｖａｌｕｅを計算し、計算したｐ－ｖａｌｕｅを多重検定補正によって補正し、補正したｐ－ｖａｌｕｅに基づいて遅延度合と相関があるか否かを判定する。したがって、第１相関分析部４２は、遅延度合とリソースデータの相関分析を正確に行うことができる。

【実施例２】

【０１３１】

ところで、リクエスト数が少ない場合には、平均応答時間が不安定になり、遅延度合が不安定になる。遅延度合が不安定な部分があると、遅延と関係するリソース３４を見つけられない場合がある。そこで、実施例２では、遅延度合が不安定な部分のデータを取り除いた場合と全データを使った場合の両方で遅延原因診断を行う診断装置について説明する。

【０１３２】

図３２は、遅延度合が不安定なデータの除去を説明するための図である。図３２の横軸はリクエスト数を示し、縦軸は遅延度合を示す。図３２に示すように、リクエスト数が少ない場合には、遅延度合が不安定になる。このため、実施例２に係る診断装置は、実線の四角で囲まれた全データを使った場合と、破線の四角で囲まれたデータだけを使った場合の両方で遅延原因診断を行う。

【０１３３】

図３３は、実施例２に係るクラウドシステムの構成を示す図である。なお、ここでは説明の便宜上、図２に示した各部と同様の役割を果たす機能部については同一符号を付すこととしてその詳細な説明を省略する。

【０１３４】

図３３に示すように、実施例２に係るクラウドシステム１ａは、図２に示したクラウドシステム１と比較して、診断装置４の代わりに診断装置４ａを有する。診断装置４ａは、診断装置４にはないデータ除去部４０を有し、遅延判定部４１の代わりに遅延判定部４１ａを有する。

【０１３５】

データ除去部４０は、遅延度合が不安定なデータを取り除く。遅延判定部４１ａは、データ除去部４０により遅延度合が不安定な部分のデータが取り除かれたデータと、遅延情報作成装置２が作成した全データと両方でアプリに遅延が発生しているか否かを判定する。

【０１３６】

遅延度合を計算するために用いられる平均応答時間は、リクエスト数が増加すると安定するため、リクエスト数が大きくなると遅延度合の分散が小さくなる。そこで、データ除去部４０は、リクエスト数を基準にデータを分割し、遅延度合の分散が減少から増加に変わるリクエスト数を閾値とし、閾値以下のデータを遅延度合が不安定なデータとして取り除く。

【０１３７】

図３４は、遅延度合が不安定なデータの除去方法を説明するための図である。図３４（ａ）は、データの分割を示す。横軸はリクエスト数であり、縦軸は遅延度合である。図３４（ａ）に示すように、データはリクエスト数に応じて分割される。分割されたデータには、リクエスト数が少ない方から順に０、１、２、３のＩＤが付加される。

【０１３８】

図３４（ｂ）は、分割されたデータの遅延度合の分散を示す。横軸はデータのＩＤであり、縦軸は遅延度合の分散である。図３４（ｂ）に示すように、ＩＤが０であるデータについての遅延度合の分散は約３．３であり、ＩＤが１であるデータについての遅延度合の分散は約２．０であり、ＩＤが２であるデータについての遅延度の分散は約４．９である。すなわち、遅延度合の分散は、ＩＤ＝１までは減少し、ＩＤ＝２で増加する。

【０１３９】

したがって、データ除去部４０は、ＩＤ＝１とＩＤ＝２の境界のリクエスト数を閾値として、リクエスト数が閾値以下のデータを遅延度合が不安定なデータとして取り除く。図３４（ｂ）では、閾値は約１６００であり、ＩＤ＝０とＩＤ＝１のデータが取り除かれる。

【０１４０】

具体的には、データ除去部４０は、以下の手順でデータを取り除く。

（１）データ除去部４０は、遅延度合が閾値T_{rt}以上のデータのうち、最小と最大のリクエスト数をそれぞれC_{min}、C_{max}とする。

（２）そして、データ除去部４０は、リクエスト数がC_{min}以上、C_{max}以下のデータ数をｎとし、分割数ｋをSturgesの公式を利用して$k = 1 + \log_2 n$により計算する。

（３）そして、データ除去部４０は、リクエスト数がC_{min}以上、C_{max}以下のデータを、リクエスト数順にｋ個に分割し、分割されたそれぞれのデータで遅延度合の分散を計算する。

（４）そして、データ除去部４０は、リクエスト数が低いデータから順に分散を見ていき、分散が減少から増加に変わる分割の最小のリクエスト数をT_cとする。

（５）そして、データ除去部４０は、リクエスト数が閾値T_c以下のデータを取り除く。

【０１４１】

図３５は、診断装置４ａによる処理のフローを示すフローチャートである。図３５に示すように、診断装置４ａは、アプリ数だけステップＳ８１～ステップＳ９１の処理を繰り

返す。すなわち、診断装置４ａは、アプリの診断対象期間の遅延度合情報を取得する（ステップＳ８１）。そして、診断装置４ａは、リクエスト数の閾値T_cを計算する閾値計算処理を行う（ステップＳ８２）。

【０１４２】

そして、診断装置４ａは、リクエスト数が閾値T_cより大きくかつ遅延度合が閾値T_{rt}以上の回数がT_{dr}以上か否かを判定し（ステップＳ８３）、ステップＳ８３の判定結果がＮｏの場合には、ステップＳ８８へ進む。一方、ステップＳ８３の判定結果がＹｅｓである場合には、診断装置４ａは、リクエスト数が閾値T_cより大きいデータを抽出し（ステップＳ８４）、第１相関分析処理を行う（ステップＳ８５）。

【０１４３】

そして、診断装置４ａは、第２相関分析処理を行う（ステップＳ８６）、なお、ステップＳ８６の処理は、リソース性能低下の影響を受けるデータの相関があった場合に行われる。そして、診断装置４ａは、遅延原因診断処理を行う（ステップＳ８７）。

【０１４４】

そして、診断装置４ａは、遅延度合が閾値T_{rt}以上の回数がT_{dr}以上か否かを判定し（ステップＳ８８）、遅延度合が閾値T_{rt}以上の回数がT_{dr}以上でない場合には、次のアプリの処理を行う。一方、遅延度合が閾値T_{rt}以上の回数がT_{dr}以上である場合には、診断装置４ａは、第１相関分析処理を行う（ステップＳ８９）。

【０１４５】

そして、診断装置４ａは、第２相関分析処理を行う（ステップＳ９０）、なお、ステップＳ９０の処理は、リソース性能低下の影響を受けるデータの相関があった場合に行われる。そして、診断装置４ａは、遅延原因診断処理を行う（ステップＳ９１）。

【０１４６】

そして、アプリ数だけステップＳ８１～ステップＳ９１の処理を繰り返すと、診断装置４ａは、クラウド基盤要因により応答遅延が発生と診断されたアプリがあるかを判定し（ステップＳ９２）、ある場合には、クラウド運用管理者に通知する（ステップＳ９３）。

【０１４７】

図３６は、閾値計算処理のフローを示すフローチャートである。図３６に示すように、データ除去部４０は、遅延度合が閾値T_{rt}以上のデータのうち、最小のリクエスト数と最大のリクエスト数をそれぞれC_{min}、C_{max}とする（ステップＳ１０１）。そして、データ除去部４０は、リクエスト数がC_{min}以上、C_{max}以下のデータ数をｎとし、分割数ｋを計算する（ステップＳ１０２）。

【０１４８】

そして、データ除去部４０は、リクエスト数がC_{min}以上、C_{max}以下のデータを、リクエスト数順に並び替え、ｋ個に分割する（ステップＳ１０３）。そして、データ除去部４０は、分割されたデータの遅延度合の分散を計算する（ステップＳ１０４）処理を分割数分だけ繰り返す。そして、データ除去部４０は、分散が減少から増加に変わる分割データの最小のリクエスト数をT_cとする（ステップＳ１０５）。

【０１４９】

上述してきたように、実施例２では、診断装置４ａは、リクエスト数がT_c以下のデータが取り除かれたデータと、全データとの両方でアプリに遅延が発生しているか否かを判定してアプリ遅延原因の診断を行う。したがって、診断装置４ａは、遅延度合が不安定なデータに影響されることなくアプリ遅延原因の診断を行うことができる。

【０１５０】

なお、実施例１及び２では、診断装置４及び４ａについて説明したが、診断装置４及び４ａが有する構成をソフトウェアによって実現することで、同様の機能を有する診断プログラムを得ることができる。そこで、診断プログラムを実行するコンピュータについて説明する。

【０１５１】

図３７は、実施例１及び２に係る診断プログラムを実行するコンピュータのハードウェ

ア構成を示す図である。図３７に示すように、コンピュータ５０は、メインメモリ５１と、ＣＰＵ５２と、ＬＡＮ（Local Area Network）インタフェース５３と、ＨＤＤ（Hard Disk Drive）５４とを有する。また、コンピュータ５０は、スーパーＩＯ（Input Output）５５と、ＤＶＩ（Digital Visual Interface）５６と、ＯＤＤ（Optical Disk Drive）５７とを有する。

【０１５２】

メインメモリ５１は、プログラムやプログラムの実行途中結果などを記憶するメモリである。ＣＰＵ５２は、メインメモリ５１からプログラムを読み出して実行する中央処理装置である。ＣＰＵ５２は、メモリコントローラを有するチップセットを含む。

【０１５３】

ＬＡＮインタフェース５３は、コンピュータ５０をＬＡＮ経由で他のコンピュータに接続するためのインタフェースである。ＨＤＤ５４は、プログラムやデータを格納するディスク装置であり、スーパーＩＯ５５は、マウスやキーボードなどの入力装置を接続するためのインタフェースである。ＤＶＩ５６は、液晶表示装置を接続するインタフェースであり、ＯＤＤ５７は、ＤＶＤの読み書きを行う装置である。

【０１５４】

ＬＡＮインタフェース５３は、ＰＣＩエクスプレス（ＰＣＩｅ）によりＣＰＵ５２に接続され、ＨＤＤ５４及びＯＤＤ５７は、ＳＡＴＡ（Serial Advanced Technology Attachment）によりＣＰＵ５２に接続される。スーパーＩＯ５５は、ＬＰＣ（Low Pin Count）によりＣＰＵ５２に接続される。

【０１５５】

そして、コンピュータ５０において実行される診断プログラムは、コンピュータ５０により読み出し可能な記録媒体の一例であるＤＶＤに記憶され、ＯＤＤ５７によってＤＶＤから読み出されてコンピュータ５０にインストールされる。あるいは、診断プログラムは、ＬＡＮインタフェース５３を介して接続された他のコンピュータシステムのデータベースなどに記憶され、これらのデータベースから読み出されてコンピュータ５０にインストールされる。そして、インストールされた診断プログラムは、ＨＤＤ５４に記憶され、メインメモリ５１に読み出されてＣＰＵ５２によって実行される。

【０１５６】

また、実施例１及び２では、正規化平均応答時間を遅延度合として用いる場合について説明したが、本発明はこれに限定されるものではなく、他の値を遅延度合として用いる場合にも同様に適用することができる。

【０１５７】

また、実施例１及び２では、遅延情報作成装置２が、診断装置４及び４ａと異なる装置である場合について説明したが、本発明はこれに限定されるものではなく、遅延情報作成装置２の機能が、診断装置４及び４ａに含まれる場合にも同様に適用することができる。

【符号の説明】

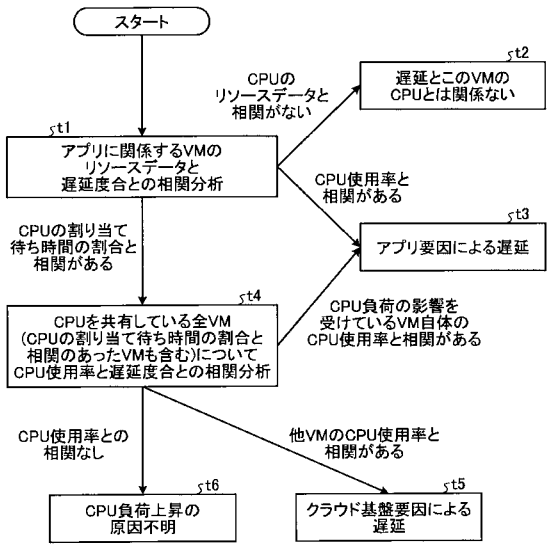
【０１５８】

１，１ａ　クラウドシステム
２　遅延情報作成装置
３ｃ　ネットワークスイッチ
４，４ａ　診断装置
９　物理マシン
９ａ　仮想マシン
２１　キャプチャ部
２２　パケット情報記憶部
２３　種別判定用データ記憶部
２４　種別判定部
２５　種別情報記憶部
２６　応答時間算出部

２７　　応答時間情報記憶部
２８　　正規化部
２９　　代表情報記憶部
３０　　遅延度合情報記憶部
３１　　ＶＭ情報記憶部
３２　　リソースデータ記憶部
３３　　構成情報記憶部
４０　　データ除去部
４１，４１ａ　　遅延判定部
４２　　第１相関分析部
４３　　第１分析結果記憶部
４４　　第２相関分析部
４５　　第２分析結果記憶部
４６　　遅延原因診断部
４７　　診断結果記憶部
５０　　コンピュータ
５１　　メインメモリ
５２　　ＣＰＵ
５３　　ＬＡＮインタフェース
５４　　ＨＤＤ
５５　　スーパーＩＯ
５６　　ＤＶＩ
５７　　ＯＤＤ

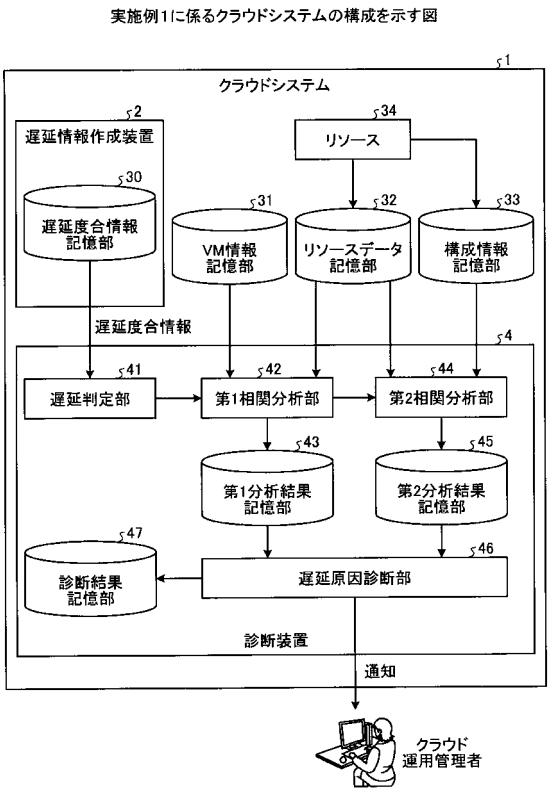
【図１】

実施例1に係る診断装置による遅延原因診断を説明するための図



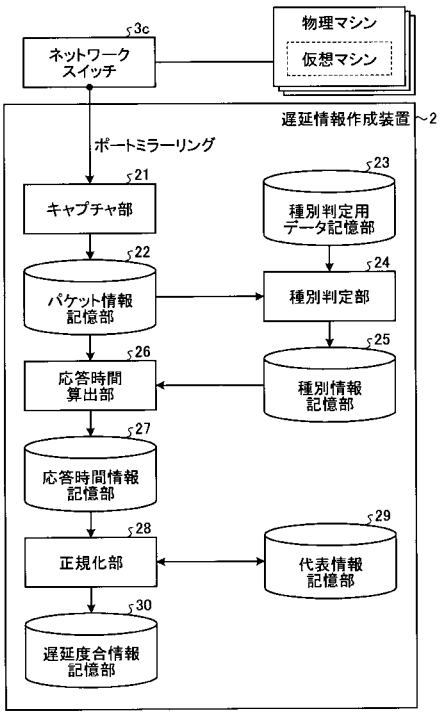
【図２】

実施例1に係るクラウドシステムの構成を示す図



【図３】

遅延情報作成装置の構成を示す図



【図４】

種別判定用データ記憶部の一例を示す図

(a)ポートリスト

ポート番号
80
443
…

(b)学習データ

アプリの種別	平均応答時間（マイクロ秒）	サーバの平均通信量（バイト）	サーバの平均通信回数	クライアント装置の平均通信量（バイト）	クライアント装置の平均通信回数
応答時間が性能面で重要となるアプリ	600	100	1	100	1
応答時間が性能面で重要となるアプリ	1000	100	1	150	1
その他のアプリ	5000	1000	10	100	1
…	…	…	…	…	…

【図５】

種別情報記憶部の一例を示す図

IPアドレス	ポート番号	種別
10.20.30.40	80	応答時間が性能面で重要となるアプリ
20.30.40.50	443	応答時間が性能面で重要となるアプリ
30.40.50.60	20	その他のアプリ
…	…	…

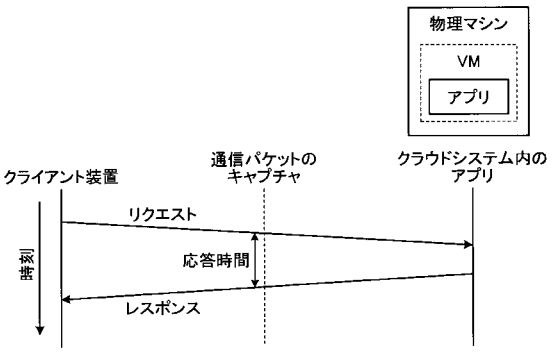
【図７】

応答時間情報記憶部の一例を示す図

時刻	IPアドレス	ポート番号	応答時間（マイクロ秒）
2016/06/24 09:00:00	10.20.30.40	80	600
2016/06/24 09:00:10	10.20.30.40	80	600
2016/06/24 09:01:00	20.30.40.50	443	1000
2016/06/24 09:10:00	10.20.30.40	80	6000
…	…	…	…

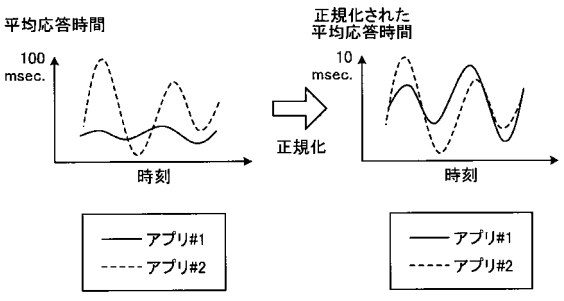
【図６】

応答時間の算出方法を説明するための図



【図８】

平均応答時間の正規化を説明するための図



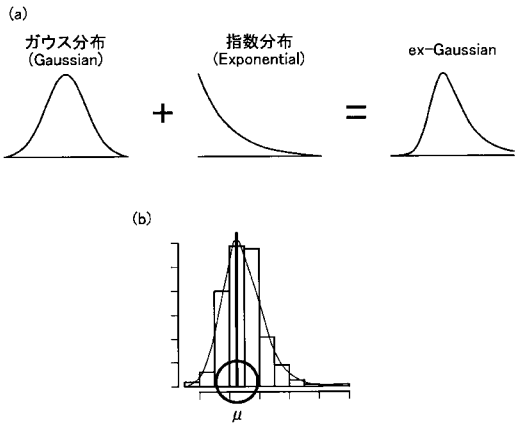
【図９】

代表情報記憶部の一例を示す図

時刻	IPアドレス	ポート番号	応答時間（マイクロ秒）
2016/06/23 00:00:00	10.20.30.40	80	600
2016/06/23 00:00:00	20.30.40.50	443	1000
…	…	…	…

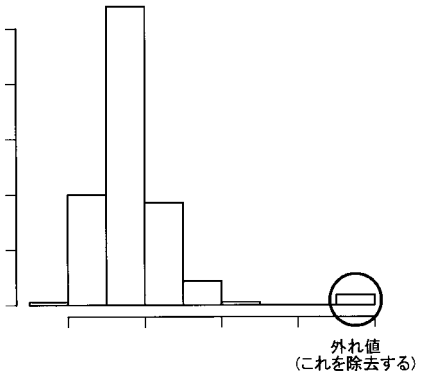
【図１０】

ex-Gaussian分布を説明するための図



【図１１】

外れ値を説明するための図



【図１２】

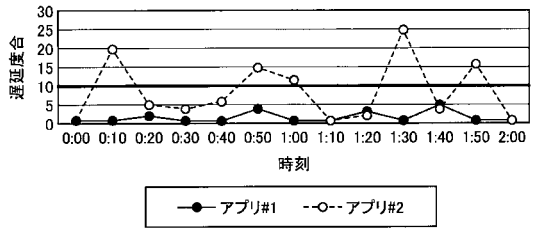
遅延度合情報記憶部の一例を示す図

時刻	IPアドレス	ポート番号	正規化平均応答時間	リクエスト数
2016/06/24 09:00:00	10.20.30.40	80	1.0	2
2016/06/24 09:01:00	20.30.40.50	443	1.0	1
2016/06/24 09:10:00	10.20.30.40	80	10.0	100
…	…	…	…	…

【図１３】

アプリの遅延判定を説明するための図

T_{rt}=10,T_{dr}=5の場合



	遅延度合がT_{rt}以上の回数
アプリ#1	0
アプリ#2	5

→アプリ#2は遅延が発生している

【図１４】

VM情報記憶部の一例を示す図

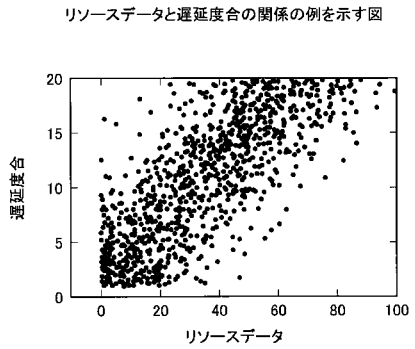
変更日時	VMのIPアドレス	ポート番号	関係するVMリスト
2016/06/24 09:00:00	10.20.30.40	80	VM#1,VM#2,VM#3
2016/06/24 09:00:00	30.40.50.70	443	VM#4
2016/06/24 09:00:00	30.40.50.80	443	VM#5
…	…	…	…

【図１５】

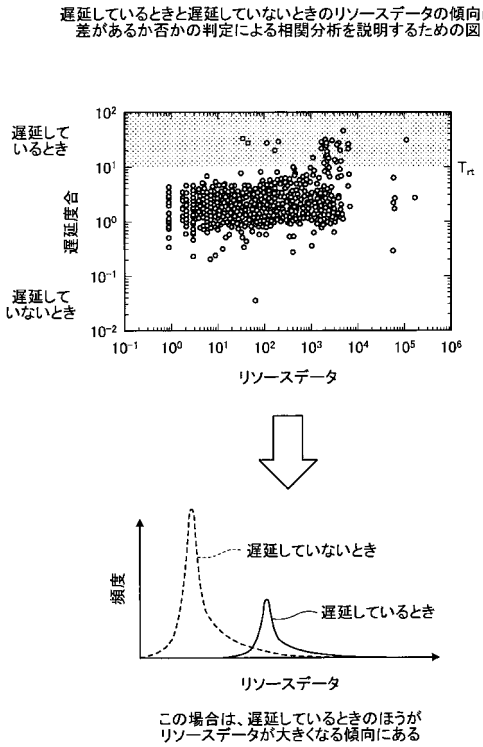
リソースデータ記憶部の一例を示す図

時刻	VM名	リソースデータ名	値
2016/06/24 09:00:00	VM#1	CPU使用率	10.0
2016/06/24 09:00:00	VM#1	CPUの割り当て待ち時間の割合	1.0
2016/06/24 09:00:00	VM#4	CPU使用率	100.0
…	…	…	…

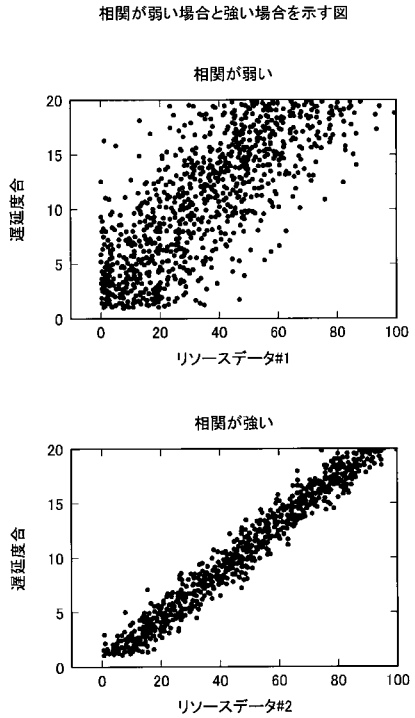
【図１６】



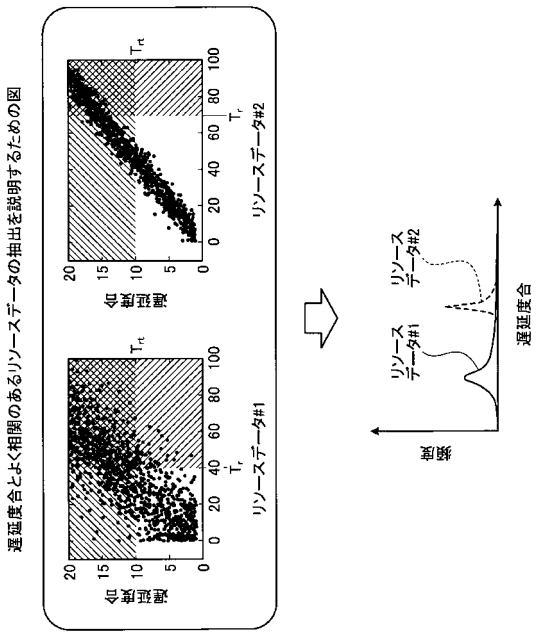
【図１７】



【図１８Ａ】



【図１８Ｂ】



【図１９】

第1分析結果記憶部の一例を示す図

診断日時	開始時間	終了時間	VMのIPアドレス	ポート番号	VM名	リソース種類	リソースデータ名
2016/06/24 00:00:00	2016/05/24 00:00:00	2016/06/24 00:00:00	10.20.30.40	80	VM#3	CPU	CPUの割り当て待ち時間の割合
2016/06/24 00:00:00	2016/05/24 00:00:00	2016/06/24 00:00:00	30.40.50.70	443	VM#4	CPU	CPU使用率
…	…	…	…	…	…	…	…

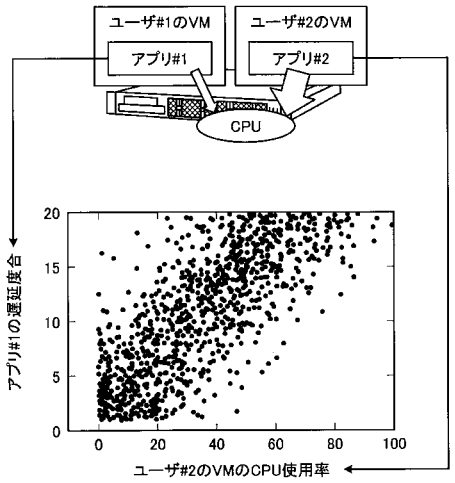
【図２０】

構成情報記憶部の一例を示す図

変更日時	物理マシン	利用ユーザ	VM名	VMのIPアドレス	利用ディスク名
2016/06/24 00:00:00	PM#1	ユーザ#1	VM#1	10.20.30.40	ディスク#1
2016/06/24 00:00:00	PM#2	ユーザ#1	VM#2	20.30.40.50	ディスク#2
2016/06/24 00:00:00	PM#3	ユーザ#1	VM#3	30.40.50.60	ディスク#3
2016/06/24 00:00:00	PM#3	ユーザ#2	VM#4	30.40.50.70	ディスク#3
…	…	…	…	…	…

【図２１】

第2相関分析部による相関分析を説明するための図



【図２２】

第2分析結果記憶部の一例を示す図

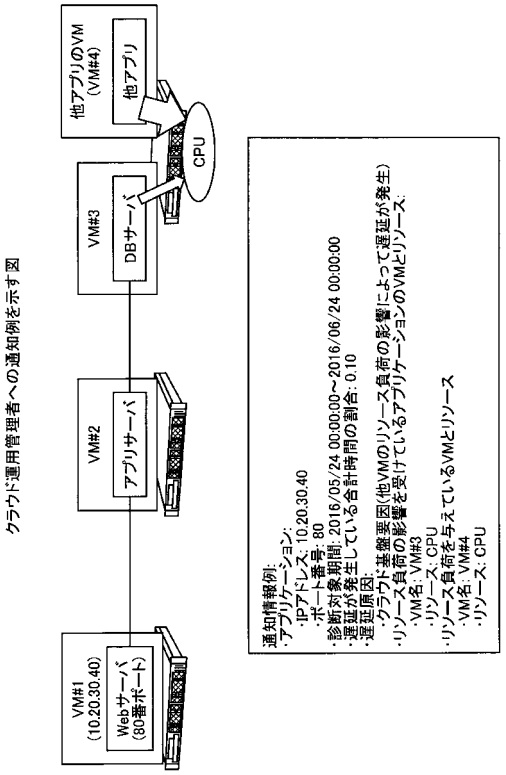
診断日時	開始時間	終了時間	VMのIPアドレス	ポート番号	アプリVM	リソース種類	相関VM	リソース負荷量データ
2016/06/24 00:00:00	2016/05/24 00:00:00	2016/06/24 00:00:00	10.20.30.40	80	VM#3	CPU	VM#4	CPU使用率
2016/06/24 00:00:00	2016/05/24 00:00:00	2016/06/24 00:00:00	30.40.50.80	443	VM#5	CPU	VM#6	CPU使用率
2016/06/24 00:00:00	2016/05/24 00:00:00	2016/06/24 00:00:00	30.40.50.80	443	VM#5	CPU	VM#7	CPU使用率
…	…	…	…	…	…	…	…	…

【図２３】

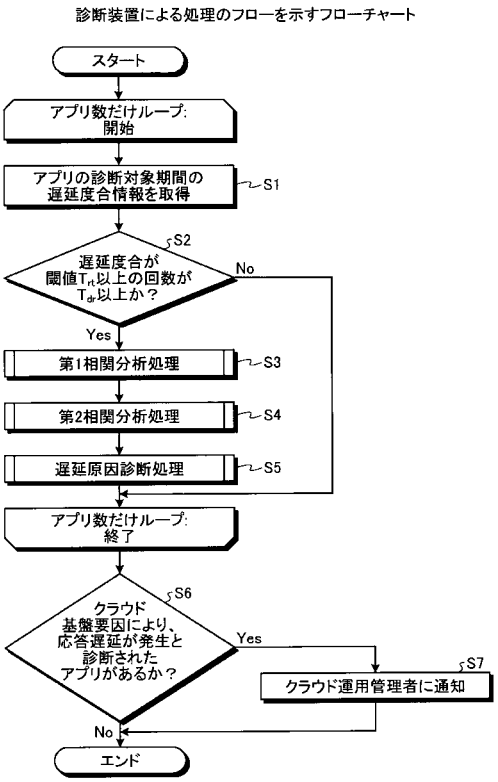
診断結果記憶部の一例を示す図

診断日時	開始時間	終了時間	VMのIPアドレス	ポート番号	遅延割合	アプリVM	リソース種類	リソースデータ	負荷VM	リソース負荷量データ	診断結果
2016/06/24 00:00:00	2016/05/24 00:00:00	2016/06/24 00:00:00	10.20.30.40	80	0.10	VM#3	CPU	CPUの割り当て待ち時間の割合	VM#4	CPU使用率	クラウド基盤要因
2016/06/24 00:00:00	2016/05/24 00:00:00	2016/06/24 00:00:00	30.40.50.70	443	0.05	VM#4	CPU	CPU使用率			アプリ要因
…	…	…	…	…	…	…	…	…	…	…	…

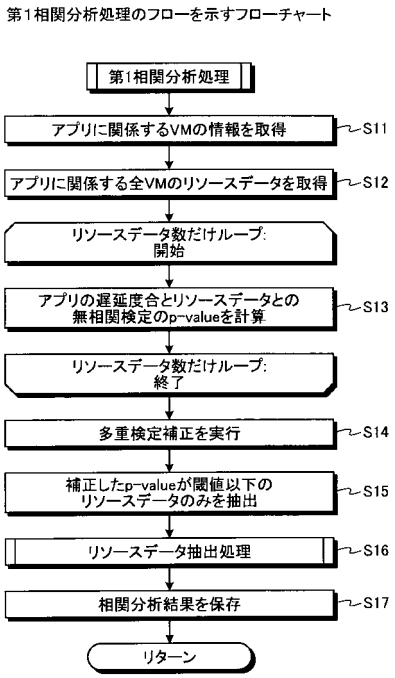
【図２４】



【図２５】

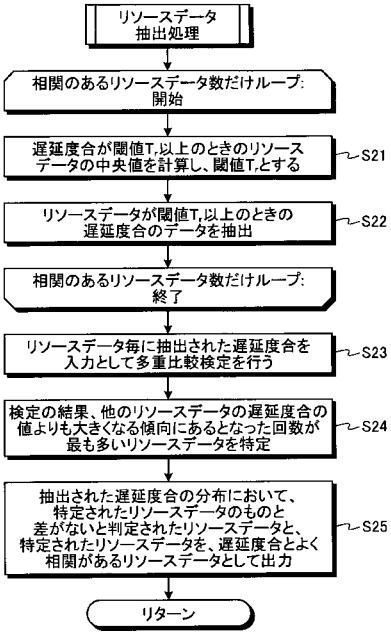


【図２６】



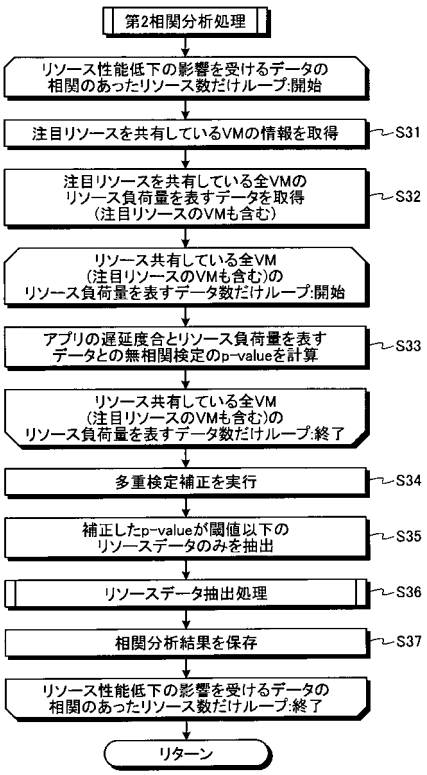
【図２７】

リソースデータ抽出処理のフローを示すフローチャート



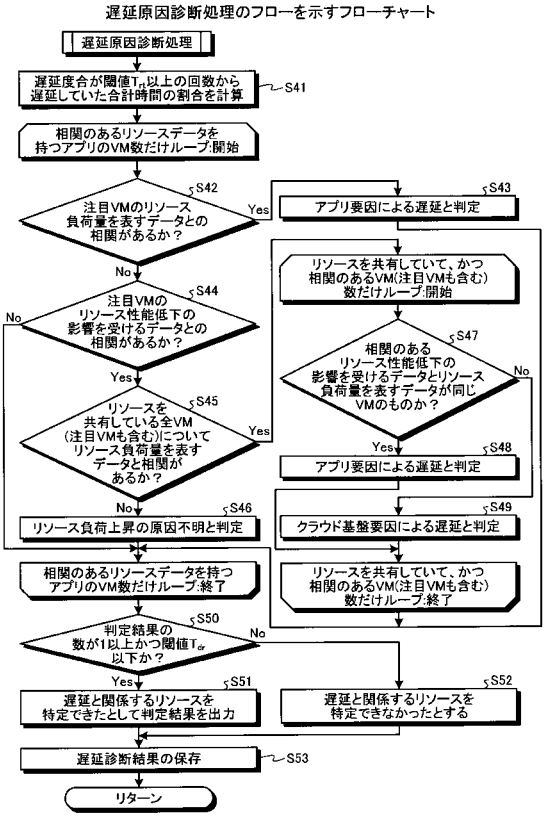
【図２８】

第2相関分析処理のフローを示すフローチャート



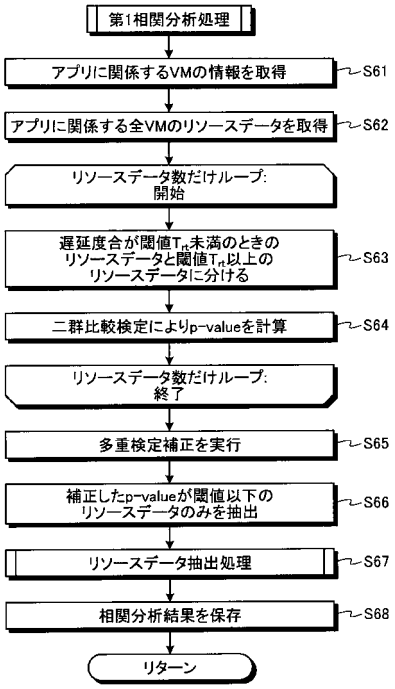
【図２９】

遅延原因診断処理のフローを示すフローチャート



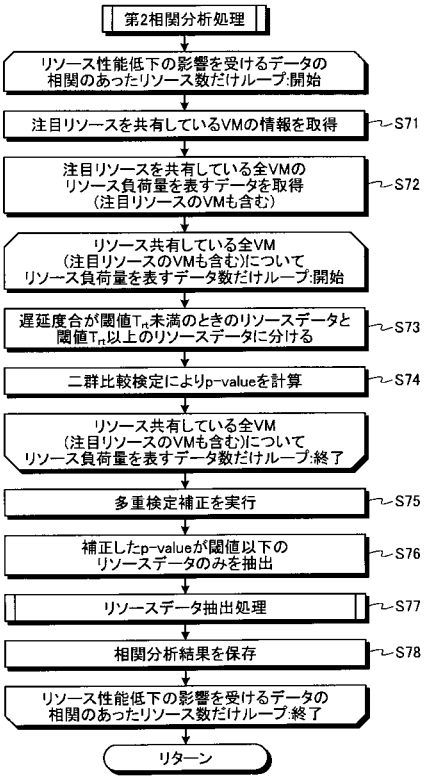
【図３０】

二群比較検定を用いる第1相関分析処理のフローを示すフローチャート



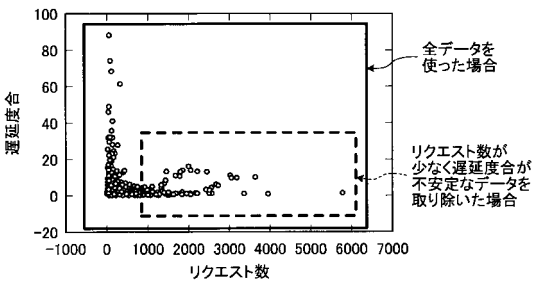
【図３１】

二群比較検定を用いる第2相関分析処理のフローを示すフローチャート



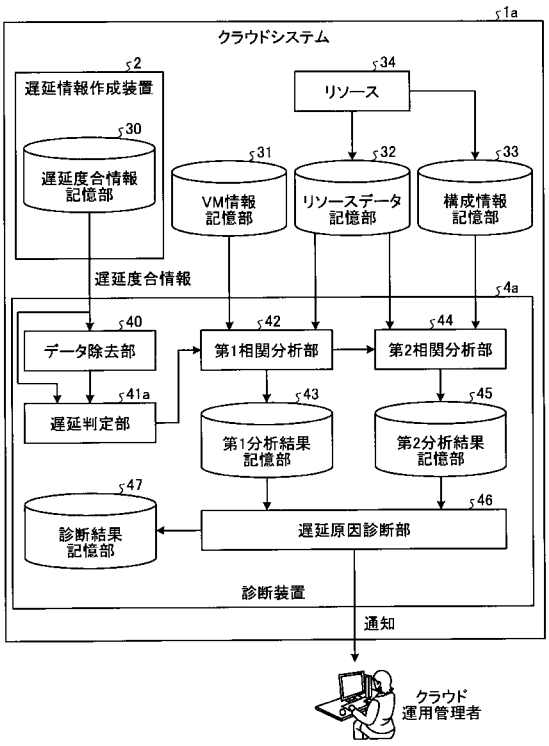
【図３２】

遅延度合が不安定なデータの除去を説明するための図



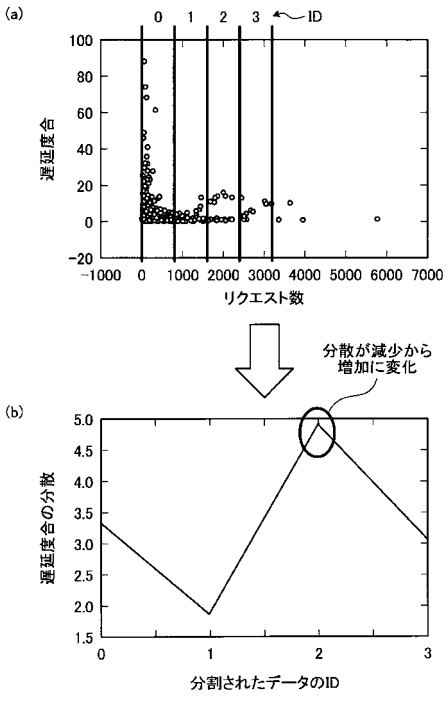
【図３３】

実施例2に係るクラウドシステムの構成を示す図



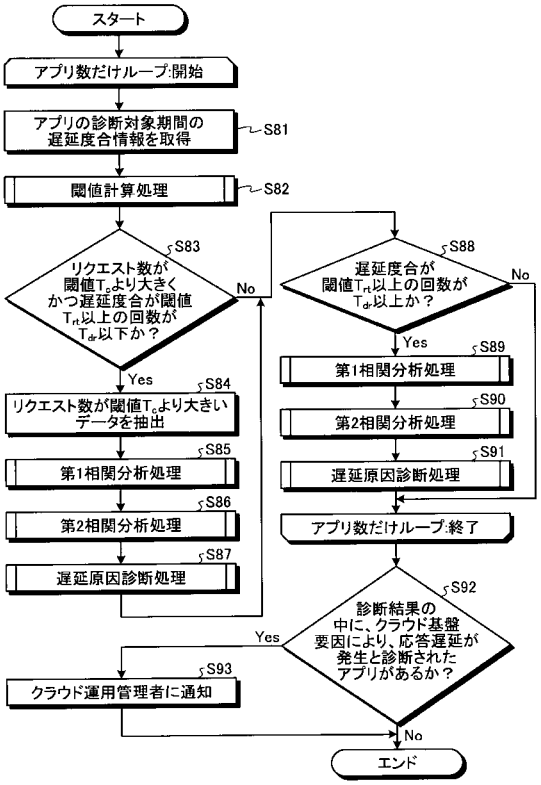
【図３４】

遅延度合が不安定なデータの除去方法を説明するための図



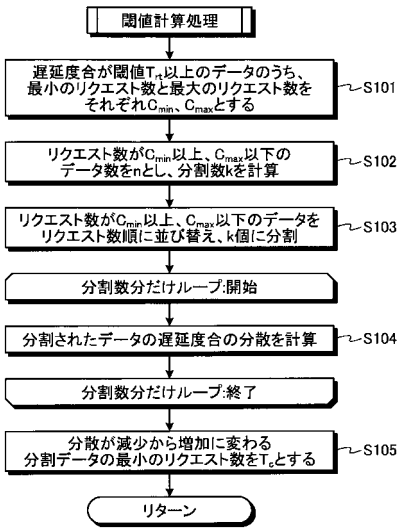
【図３５】

診断装置による処理のフローを示すフローチャート



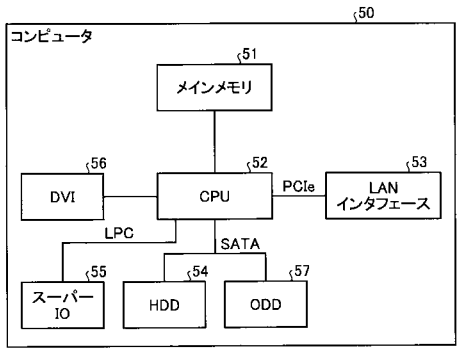
【図３６】

閾値計算処理のフローを示すフローチャート



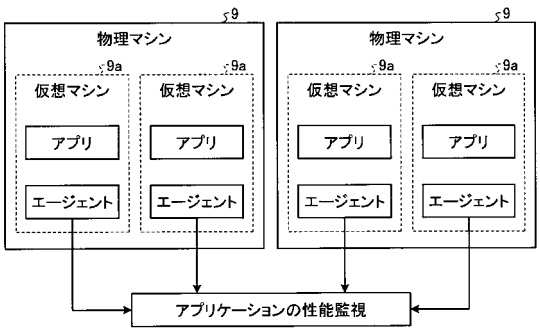
【図３７】

実施例1及び2に係る診断プログラムを実行するコンピュータの構成を示す図



【図３８】

エージェントによる監視を説明するための図



フロントページの続き

(51)Int.Cl.	FI		テーマコード（参考）
	Ｇ０６Ｆ	11/07	１４０Ｃ