

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5104529号  
(P5104529)

(45) 発行日 平成24年12月19日(2012.12.19)

(24) 登録日 平成24年10月12日(2012.10.12)

(51) Int.Cl. F 1  
**G 0 6 F 11/34 (2006.01)** G 0 6 F 11/34 S

請求項の数 6 (全 20 頁)

(21) 出願番号	特願2008-122197 (P2008-122197)	(73) 特許権者	000005223
(22) 出願日	平成20年5月8日(2008.5.8)		富士通株式会社
(65) 公開番号	特開2009-271755 (P2009-271755A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成21年11月19日(2009.11.19)	(74) 代理人	100078330
審査請求日	平成23年1月18日(2011.1.18)		弁理士 笹島 富二雄
		(72) 発明者	大江 和一
			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	熊野 達夫
			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	野口 泰生
			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】 分析支援プログラム、分析支援方法および分析支援装置

(57) 【特許請求の範囲】

【請求項1】

記憶部を有するコンピュータに、  
 要求が送信されてから、送信された要求に対する処理を複数の情報処理装置が連携して実行した結果を示す実行結果情報と前記複数の情報処理装置の稼動状態を示す稼動状態情報とを含む応答が返信されるまでに要した処理時間を算出させ、

返信された前記応答に含まれる各情報処理装置の稼動状態情報をそれぞれ抽出させ、  
 抽出された各情報処理装置の稼動状態情報に、算出された処理時間を対応付けて前記記憶部に記憶させ、

全処理時間範囲を第1の単位時間で区画した処理時間範囲毎に、前記記憶部に記憶された前記複数の情報処理装置の稼動状態情報を、対応する処理時間で分別させ、

全処理時間範囲に対し、分別された各処理時間範囲に属する稼動状態情報に対応する処理時間を積算した合計処理時間の占める割合が、所定の閾値より大きい処理時間範囲に属する稼動状態情報を分析させ、

前記所定の閾値より大きい処理時間範囲に属する稼動状態情報を分析した結果を出力させることを特徴とする分析支援プログラム。

【請求項2】

前記稼動状態情報は、  
 前記複数の情報処理装置における各処理の内訳と、各内訳の処理時間を示す処理時間情報を含み、

前記所定の閾値より大きい処理時間範囲に属する稼動状態情報の分析は、さらに、  
前記コンピュータに、

第2の単位時間で区画された処理時間範囲毎に、前記記憶部に記憶された前記複数の情報処理装置の稼動状態情報を、対応する内訳の処理時間で分別し、

各処理時間範囲に属する内訳の処理時間の積算値又は内訳の数を算出させることを特徴とする請求項1記載の分析支援プログラム。

【請求項3】

前記稼動状態情報は、

前記複数の情報処理装置のカーネル状態の内訳を含んで構成され、

前記所定の閾値より大きい処理時間範囲に属する稼動状態情報の分析は、さらに、

前記コンピュータに、

前記記憶部に記憶された稼動状態情報のカーネル状態の内訳を組み合わせた各グループに属する稼動状態情報の数が、前記所定の閾値より大きい処理時間範囲に属する全ての稼動状態情報の数に対して占める割合を算出させることを特徴とする請求項1記載の分析支援プログラム。

【請求項4】

前記稼動状態情報は、

情報処理装置毎且つ処理毎に、稼動状態情報に含まれる内訳が区別されたことを特徴とする請求項1～請求項3のいずれか1項に記載の分析支援プログラム。

【請求項5】

記憶部を有するコンピュータが、

要求が送信されてから、送信された要求に対する処理を複数の情報処理装置が連携して実行した結果を示す実行結果情報と前記複数の情報処理装置の稼動状態を示す稼動状態情報とを含む応答が返信されるまでに要した処理時間を算出し、

返信された前記応答に含まれる各情報処理装置の稼動状態情報をそれぞれ抽出し、

抽出された各情報処理装置の稼動状態情報に、算出された処理時間を対応付けて前記記憶部に記憶し、

全処理時間範囲を第1の単位時間で区画した処理時間範囲毎に、前記記憶部に記憶された前記複数の情報処理装置の稼動状態情報を、対応する処理時間で分別し、

全処理時間範囲に対し、分別された各処理時間範囲に属する稼動状態情報に対応する処理時間を積算した合計処理時間の占める割合が、所定の閾値より大きい処理時間範囲に属する稼動状態情報を分析し、

前記所定の閾値より大きい処理時間範囲に属する稼動状態情報を分析した結果を出力させることを特徴とする分析支援方法。

【請求項6】

要求が送信されてから、送信された要求に対する処理を複数の情報処理装置が連携して実行した結果を示す実行結果情報と前記複数の情報処理装置の稼動状態を示す稼動状態情報とを含む応答が返信されるまでに要した処理時間を算出する算出部と、

返信された前記応答に含まれる各情報処理装置の稼動状態情報をそれぞれ抽出する抽出部と、

抽出された各情報処理装置の稼動状態情報に、算出された処理時間を対応付けて記憶する記憶部と、

全処理時間範囲を第1の単位時間で区画した処理時間範囲毎に、前記記憶部に記憶された前記複数の情報処理装置の稼動状態情報を、対応する処理時間で分別するとともに、全処理時間範囲に対し、分別された各処理時間範囲に属する稼動状態情報に対応する処理時間を積算した合計処理時間の占める割合が、所定の閾値より大きい処理時間範囲に属する稼動状態情報を分析する分析部と、

前記所定の閾値より大きい処理時間範囲に属する稼動状態情報を分析した結果を出力する出力部とを有することを特徴とする分析支援装置。

【発明の詳細な説明】

10

20

30

40

50

## 【技術分野】

## 【0001】

本発明は、分析を支援する技術に関する。

## 【背景技術】

## 【0002】

近年、コンピュータシステムが社会で担う役割が大きくなってきていると共に、その構成は複雑化し、処理データ量も増加してきている。そして、かかるコンピュータシステムにおいては、さらなる処理速度の向上や信頼性の向上が求められている。このため、このような要求に対処するべく、ある1つの処理要求に対し、複数のコンピュータで連携して処理を行うコンピュータシステムが採用されている。

10

## 【0003】

しかしながら、このように複数のコンピュータで1つの処理要求に対応する処理を連携して実行する場合において、処理遅延などの問題が発生したときには、どのコンピュータが原因になっているのかを特定するのが困難である。そこで、複数のコンピュータで連携して処理が実行されたときの各コンピュータの稼動状態を分析する技術として、次のような技術が提案されている。即ち、1つのコンピュータ(コンピュータ1)から他のコンピュータ(コンピュータ2)に処理を派生(分散)させる構成において、コンピュータ1において、自装置における処理の統計情報を生成して蓄積する。一方、コンピュータ2では、自装置における処理の統計情報を生成し、これに、処理の分散元がコンピュータ1であることを示す情報を付加して蓄積する。そして、これらのコンピュータの稼動状態を分析する段階において、各コンピュータに蓄積された統計情報を関連付ける。

20

【特許文献1】特開平6-28326号公報

## 【発明の開示】

## 【発明が解決しようとする課題】

## 【0004】

しかしながら、かかる技術においては、コンピュータの稼動状態を分析する段階において、すでに各コンピュータごとに統計情報が生成され、蓄積されていることが前提となっている。即ち、処理が繰り返し実行された結果、各コンピュータについて蓄積された統計情報を基に分析が行われる。このため、1つ1つの処理要求に対する処理において、コンピュータ1及びコンピュータ2が夫々どのような稼動状態になっていたかについては、次のように、特定することができない。

30

## 【0005】

例えば、上述の技術において、コンピュータ1及びコンピュータ2により連携して行った処理時間が1msとなっている場合における、各コンピュータの遅延原因を分析することを想定する。この場合、各コンピュータで統計情報が作成された結果、コンピュータ1では、各遅延原因が占める確率として、原因Aが60パーセント、原因Bが40パーセントであることが特定できたとする。一方、コンピュータ2では、原因Cが20パーセント、原因Dが80パーセントとなっていることが特定できたとする。これらの情報からは、各コンピュータにおける遅延原因を、複数の処理についての統計結果として夫々特定することはできる。しかし、例えば、コンピュータ1において原因Aが発生した処理において、その処理を連携して実行したコンピュータ2ではどのような遅延原因が生じたのかについては、予め各コンピュータごとに統計情報が生成されてしまっていては特定することができない。

40

## 【0006】

このように、かかる技術では、各コンピュータの稼動状態を夫々分析することは可能なものの、これらの複数のコンピュータの状態を複合的に分析することは困難であり、分析できる内容に限界があった。

## 【0007】

そこで、本発明は以上のような従来の問題点に鑑み、1つの処理要求に対する処理を複数のコンピュータにおいて連携して実行するときの各コンピュータの稼動状態について、

50

複合的に分析可能にすることを目的とする。

【課題を解決するための手段】

【0008】

このため、本技術の一実施形態では、要求が送信されてから、送信された要求に対する処理を複数の情報処理装置が連携して実行した結果を示す実行結果情報と複数の情報処理装置の稼働状態を示す稼働状態情報とを含む応答が返信されるまでに要した処理時間を算出させる。また、返信された応答に含まれる各情報処理装置の稼働状態情報をそれぞれ抽出させ、抽出された各情報処理装置の稼働状態情報に、算出された処理時間を対応付けて記憶部に記憶させる。次に、全処理時間範囲を第1の単位時間で区画した処理時間範囲毎に、記憶部に記憶された複数の情報処理装置の稼働状態情報を、対応する処理時間で分別させる。それから、全処理時間範囲に対し、分別された各処理時間範囲に属する稼働状態情報に対応する処理時間を積算した合計処理時間の占める割合が、所定の閾値より大きい処理時間範囲に属する稼働状態情報を分析させる。さらに、所定の閾値より大きい処理時間範囲に属する稼働状態情報を分析した結果を出力させる。

10

【発明の効果】

【0009】

本技術によれば、1つの処理要求に対する処理の実行時に各情報処理装置において複合的に生じる稼働状態が特定される。

【発明を実施するための最良の形態】

【0010】

図1は、コンピュータ処理分析システム（分析支援装置）の全体構成図である。コンピュータ処理分析システムは、サーバ10、サーバ20、分析装置30、クライアント40及びスイッチ50を含む。サーバ10、サーバ20、分析装置30及びクライアント40は、少なくともCPU（Central Processing Unit）及び記憶装置を含んだコンピュータであり、スイッチ50を介し、ネットワークで相互に接続されている。また、スイッチ50は、通常のスイッチング機能を備えると共に、ポートミラーリング機能を備える。そして、クライアント40からサーバ10に送信される処理要求を含むリクエストパケット及びサーバ10からクライアント40に返信されるリプライパケットを、分析装置30に対しポートミラーリングする。また、本システムでは、クライアント40からの処理要求に対し、サーバ10及びサーバ20が連携して処理を実行する。このとき、サーバ10からサーバ20に対して処理を分散する。

20

30

【0011】

サーバ10は、他のコンピュータとのパケットの送受信を行うパケット送受信部10Aと、処理要求を含むリクエストパケットを受信したときに、その処理要求に対応する処理を実行し、処理結果を反映させたリプライパケットを生成する処理部10Bと、少なくとも自装置の稼働状態を示す情報をリプライパケットに付加する情報付加部10Cと、を含む。

【0012】

サーバ20もサーバ10と同様の構成であり、サーバ10におけるパケット送受信部10A、処理部10B及び情報付加部10Cに夫々対応する、パケット送受信部20A、処理部20B、及び、情報付加部20Cを含む。

40

【0013】

分析装置30は、スイッチ50からポートミラーリングされたリクエストパケットを保存するリクエストパケット保存ファイル30Aと、同じくスイッチ50からポートミラーリングされたリプライパケットに付加された、サーバ10及びサーバ20の両方の稼働状態を示す情報を含む内部情報並びにリクエストパケット及びリプライパケットの受信時刻の差分から算出された処理時間を蓄積するための内部情報DB30Bと、リクエストパケット及びリプライパケットを受信するパケット受信部30Cと、リプライパケットに付加された内部情報及び処理時間を内部情報DB30Bに蓄積する情報蓄積部30Dと、内部情報DB30Bに蓄積された内部情報及び処理時間を用いてサーバ10及びサーバ20の

50

状態を分析する分析部 30E と、を含む。なお、分析装置 30 の情報蓄積部 30D により、算出ステップ、算出手段、抽出ステップ、抽出手段、蓄積ステップ及び蓄積手段が実現される。さらに、分析部 30E により、分析ステップ、分析手段、出力ステップ及び出力手段が実現される。

#### 【0014】

クライアント 40 は、サーバ 10 に対して処理を要求するリクエストパケットを送信するとともに、サーバ 10 からのリプライパケットを受信する。

スイッチ 50 は、分析装置 30 が接続されたポートをミラーポートとし、クライアント 40 からサーバ 10 に対して送信されたリクエストパケット及びサーバ 10 からクライアント 40 に対して返信されたリプライパケットを、分析装置 30 に対してポートミラーリングする。

10

#### 【0015】

図 2 は、サーバ 10、サーバ 20、分析装置 30 及びクライアント 40 における処理の流れを示す原理図である。

クライアント 40 では、処理要求を含むリクエストパケットをサーバ 10 に対して送信する(1)。そのリクエストパケットは、スイッチ 50 により分析装置 30 にポートミラーリングされる。そして、分析装置 30 でこれを受信し(2)、リクエストパケット保存ファイル 30B に保存する(3)。一方、サーバ 10 でリクエストパケットを受信し(4)、要求された処理を実行すると共に(5)、さらに処理を分散するべく、サーバ 20 に対してリクエストパケットを送信する(6)。サーバ 20 では、リクエストパケットを受信し(7)、要求された処理を実行する(8)。そして、サーバ 20 では、リプライパケットを生成してサーバ 20 の稼働状態を示す情報を付加し(9)、サーバ 10 に対して返信する(10)。サーバ 10 では、サーバ 20 の稼働状態を示す情報が付加されたリプライパケットを受信する(11)。そして、リプライパケットを生成して、サーバ 10 の稼働状態を示す情報及びサーバ 20 から受信したリプライパケットに含まれるサーバ 20 の稼働状態を示す情報を含む内部情報を付加し(12)、さらに、そのリプライパケットをクライアント 40 に対して返信する(13)。返信されたリプライパケットは、スイッチ 50 により分析装置 30 にポートミラーリングされ、分析装置 30 で受信されるとともに(14)、クライアント 40 で受信される(15)。そして、分析装置 30 では、受信したリプライパケット及びリクエストパケット保存ファイル 30A に保存されたリクエストパケットの夫々の受信時刻に基づいて、サーバ 10 及びサーバ 20 で連携して行った処理時間を算出する。さらに、その算出した処理時間をサーバ 10 及びサーバ 20 の内部情報に関連付けて内部情報 DB 30B に蓄積し、その蓄積した情報に基づいて各サーバの稼働状態を分析する(16)。

20

30

#### 【0016】

ここで、サーバ 10 からクライアント 40 に返信されるリプライパケット、即ち、サーバ 10 及びサーバ 20 の内部情報が付加されたリプライパケットの構造について、従来のリプライパケットと対比しつつ説明する。

#### 【0017】

従来のリプライパケットは、図 3 に示すように、ヘッダ及びリプライデータから構成される。なお、このリプライパケットのヘッダには、送信元 IP アドレス、送信元ポート番号、送信先 IP アドレス、送信先ポート番号及び IO サイズなどが含まれる。

40

#### 【0018】

一方、本システムにおいて、サーバ 10 からクライアント 40 に返信されるリプライパケットでは、かかる従来のリクエストパケットのヘッダに対し、さらに、同じく図 3 に示すように、サーバ 10 及びサーバ 20 の稼働状態を示す情報が内部情報として付加される。そして、この内部情報の詳細な構造の一例について示したのが図 4 である。これらのサーバの内部情報は、その先頭に、内部情報を挿入するサーバ数(本システムではサーバ 10 及びサーバ 20 の 2 つ)を含む。そして、各サーバについて、夫々、どのサーバの情報であるかを識別するサーバの情報と、処理により実現される機能を特定するファンクショ

50

ンの情報と、各ファンクションに関連する内部情報の内訳について、その内訳の数を定義する内訳エントリ数と、具体的な内部情報の内訳と、を含む。この各サーバの情報に含めるファンクションの数は、必要に応じて変えることができる。また、1つのファンクションに含める内訳の数も必要に応じて変えることができる。このように、各サーバで実行される処理に含まれるファンクションの数やそのファンクションに関連する内部情報の内訳の数に合わせてデータ構造を変えることを可能にすることで、多様な処理の分析に対応することができる。また、分析対象となるサーバの構成によっては、例えばカーネル状態を示す情報を数多く収集することが必要となり得るが、そのような場合にも必要な情報を全て収集することができる。

**【 0 0 1 9 】**

10

次に、サーバ10、サーバ20及び分析装置30で実行される処理について夫々説明する。

図5は、サーバ10における第1の処理の内容を示す。このサーバ10における第1の処理は、パケット送受信部10Aにおいてクライアント40からのリクエストパケットをサーバ10が受信するごとに行われる。

**【 0 0 2 0 】**

ステップ1（図では「S1」と略記する。以下同様）では、クライアント40からのリクエストパケットを、パケット送受信部10Aにおいて受信する。

ステップ2では、処理部10Bにおいて、受信したリクエストパケットの処理要求に対応する処理を実行する。

20

**【 0 0 2 1 】**

ステップ3では、処理をサーバ20に分散するため、処理部10Bにおいてサーバ20に対するリクエストパケットを生成し、パケット送受信部10Aから、サーバ20に対しリクエストパケットを送信する。

**【 0 0 2 2 】**

図6は、サーバ10における第2の処理の内容を示す。このサーバ10における第2の処理は、サーバ20における内部情報が付加されたリプライパケットをパケット送受信部10Aにおいて受信するごとに行われる。

**【 0 0 2 3 】**

ステップ11では、サーバ20の稼働状態を示す情報が付加されたリプライパケットを、パケット送受信部10Aにおいて受信する。

30

ステップ12では、処理部10Bにおいて、クライアント40へのリプライパケットを生成する。

**【 0 0 2 4 】**

ステップ13では、処理要求に対応する処理時におけるサーバ10の稼働状態を示す情報を、情報付加部10Cにおいて抽出する。

ステップ14では、サーバ20から受信したリプライパケットに付加されているサーバ20の稼働状態を示す情報を、情報付加部10Cにおいて抽出する。

**【 0 0 2 5 】**

ステップ15では、情報付加部10Cにおいて、抽出したサーバ10及びサーバ20の両方の稼働状態を示す情報を含む内部情報を付加する。

40

ステップ16では、パケット送受信部10Aにおいて、内部情報が付加されたリプライパケットをクライアント40に返信する。

**【 0 0 2 6 】**

図7は、サーバ20における処理内容を示す。このサーバ20における処理は、サーバ10からのリクエストパケットをパケット送受信部20Aにおいて受信するごとに行われる。

**【 0 0 2 7 】**

ステップ21では、サーバ10からのリクエストパケットをパケット送受信部20Aにおいて受信する。

50

ステップ 22 では、処理部 20B において、受信したリクエストパケットの処理要求に対応する処理を実行する。

【0028】

ステップ 23 では、処理部 20B において、サーバ 10 へのリプライパケットを生成する。

ステップ 24 では、情報付加部 20C において、処理要求に対応する処理時におけるサーバ 20 の稼働状態を示す情報を抽出する。

【0029】

ステップ 25 では、情報付加部 20C において、抽出した稼働状態を示す情報をリプライパケットに付加する。

ステップ 26 では、パケット送受信部 20A において、サーバ 20 の稼働状態を示す情報が付加されたリプライパケットをサーバ 10 に返信する。

【0030】

かかるサーバ 10 及びサーバ 20 の処理によれば、クライアント 40 からの処理要求に対応した処理ごとにサーバ 10 からクライアント 40 へ返信されるリプライパケットに、サーバ 10 及びサーバ 20 の稼働状態を示す情報を含んだ内部情報が付加される。そして、サーバ 10 からクライアント 40 へ返信されるリプライパケットが、スイッチ 50 により分析装置 30 にポートミラーリングされることで、分析装置 30 では、各処理要求に対応する処理ごとに、この内部情報を収集することができる。

【0031】

図 8 は、分析装置 30 における第 1 の処理の内容を示す。この分析装置 30 における第 1 の処理は、クライアント 40 からサーバ 10 へのリクエストパケットがスイッチ 50 によりポートミラーリングされたものをパケット受信部 30C において受信するごとに行われる。なお、この処理は、パケット受信部 30C において実現される。

【0032】

ステップ 31 では、クライアント 40 からサーバ 10 へのリクエストパケットを受信する。

ステップ 32 では、受信したリクエストパケットに受信時刻を示すタイムスタンプを付し、リクエストパケット保存ファイル 30A に保存する。

【0033】

図 9 は、分析装置 30 における第 2 の処理の内容を示す。この分析装置 30 における第 2 の処理は、サーバ 10 からクライアント 40 へのリプライパケットがスイッチ 50 によりポートミラーリングされたものをパケット受信部 30C において受信するごとに行われる。

【0034】

ステップ 41 では、パケット受信部 30C において、サーバ 10 からクライアント 40 へのリプライパケットを受信する。

ステップ 42 では、受信したリプライパケットに対応するリクエストパケットを、その受信時刻を示すタイムスタンプとともにリクエストパケット保存ファイル 30A から取得する。そして、リプライパケットとリクエストパケットとの受信時刻の差分から、サーバ 10 及びサーバ 20 で連携した行われた処理時間を算出する。

【0035】

ステップ 43 では、受信したリプライパケットから、内部情報を抽出する。

ステップ 44 では、抽出した内部情報と、算出した処理時間とを関連付けて、内部情報 DB 30B に蓄積する。なお、ステップ 42 からステップ 44 は、情報蓄積部 30D において実現される。

【0036】

図 10 は、分析装置 30 における第 3 の処理の内容を示す。この分析装置 30 における第 3 の処理は、スケジューリングされたジョブや管理者によるオペレーションなどにより実行される。なお、この処理は、分析部 30E において実現される。

10

20

30

40

50

## 【 0 0 3 7 】

ステップ 5 1 では、内部情報 D B 3 0 B に蓄積された内部情報及びこれに関連付けられた処理時間を抽出する。

ステップ 5 2 では、抽出した内部情報及び処理時間について所定の統計処理を施し、サーバ 1 0 及びサーバ 2 0 の稼働状況を分析する。

## 【 0 0 3 8 】

ステップ 5 3 では、分析結果を出力する。なお、出力方法としては、例えば、ディスプレイに表示したり、紙媒体等に印刷したりすることができる。

ここで、クライアント 4 0 からの処理要求により、サーバ 1 0 及びサーバ 2 0 の記憶装置に対して 8 K B のデータの書込処理及び読出処理を行う場合を例として、上述の分析装置 3 0 における第 3 の処理における所定の統計処理について、さらに詳細に説明する。ここでは、( 1 ) 各サーバにおける処理要求に対応する処理の内訳及びその内訳の処理時間がリプライパケットに付加される例、( 2 ) 各サーバにおけるカーネル状態の内訳がリプライパケットに付加される例、について夫々説明する。

( 1 ) 各サーバにおける処理要求に対応する処理の内訳及びその内訳の処理時間がリプライパケットに付加される例

この例では、サーバ 1 0 及びサーバ 2 0 において連携して実行される書込処理及び読出処理の夫々について、その処理の内訳及び内訳の処理時間として、ディスク I O ( ディスク入出力 ) 処理、ロック待ち処理、メタデータ更新処理及びプロトコル生成処理の処理時間 ( 単位 : マイクロ秒 (  $\mu s$  ) ) を収集するものとする。

## 【 0 0 3 9 】

このとき、サーバ 1 0 からクライアント 4 0 に対して返信され、分析装置 3 0 にポートミラーリングされるリプライパケットに付加される、書込処理及び読出処理の実行時における夫々の内部情報は、上述したサーバ 1 0 及びサーバ 2 0 における処理を経て、夫々図 1 1 及び図 1 2 に示すような構造となる。

## 【 0 0 4 0 】

図 1 3 は、これらのリプライパケットから抽出され、内部情報 D B 3 0 B に蓄積された内部情報に含まれる処理の内訳及びその内訳の処理時間に基づき、上述した分析装置 3 0 で、書込処理及び読出処理の夫々について実行される統計処理の内容を示す。

## 【 0 0 4 1 】

ステップ 6 1 では、内部情報 D B 3 0 B に蓄積された内部情報を、内部情報に関連付けられた処理時間に応じて、所定規則 ( 例えば 1 0 m s 単位など ) で区画された処理時間範囲ごとに分別する。なお、この所定規則が第 1 の所定規則に相当し、この値は任意に定めることができる。そして、その処理時間範囲ごとに、これに分別された内部情報に関連付けられた処理時間を積算した合計処理時間 ( 積算値 ) を算出する。

## 【 0 0 4 2 】

ステップ 6 2 では、ステップ 6 1 における算出結果に基づき、区画された処理時間範囲のうち、その処理時間範囲における合計処理時間が全処理時間範囲における合計処理時間の予め設定した閾値 ( 例えば 1 0 パーセントなど ) を超える割合を有する処理時間範囲を特定する。そして、この特定した処理時間範囲に属する内部情報を、分析対象とする。なお、この予め設定する閾値は、任意に定めることができる。

## 【 0 0 4 3 】

ステップ 6 3 では、ステップ 6 2 で特定した分析対象となった内部情報に含まれる、各サーバごとの処理の内訳及びその内訳ごとの処理時間を抽出する。さらに、この処理の内訳を、内訳の処理時間に応じて、所定規則 ( 例えば 1 0 m s 単位など ) で区画された処理時間範囲ごとに分別する。そして、その処理時間範囲ごとに、これに分別された内訳の処理時間を積算した合計処理時間 ( 積算値 ) を算出する。なお、この所定規則が第 2 の所定規則に相当し、この値は任意に定めることができる。

## 【 0 0 4 4 】

なお、上記のステップ 6 2 では、各処理時間範囲に属する内部情報に関連付けられた処

10

20

30

40

50



理時間を積算した合計処理時間を算出しているが、各処理時間範囲に属する内部情報の件数を積算した合計件数を算出してもよい。この場合には、各処理時間範囲における合計件数が全処理時間範囲における件数に対して占める割合が閾値より大きい処理時間範囲に属する内部情報が、分析対象となる。

【 0 0 4 5 】

また、上記のステップ 6 3 では、各処理時間範囲に属する処理の内訳の処理時間を積算した合計処理時間を算出しているが、各処理時間範囲に属する処理の内訳の件数を積算した合計件数を算出してもよい。

【 0 0 4 6 】

図 1 4 は、書込処理及び読出処理の夫々について、処理 1 件の処理時間及び処理時間範囲ごとに分別された内部情報に関連付けられた処理時間を積算した合計処理時間の関係を示す、上記ステップ 6 1 の実行結果としてのヒストグラムの一例を示す。ここで、当該ヒストグラムにおいて表示されているパーセンテージは、各処理時間範囲における合計処理時間が、全処理時間範囲における合計処理時間に占める割合を示す。なお、この値は、各処理時間範囲における合計処理時間を、全処理時間範囲における合計処理時間で除算することで算出できる。閾値を 1 0 パーセントで予め設定している場合において、書込処理の分析結果に着目すると、処理時間範囲が 1 0 m s 以下、5 0 m s 及び 1 0 0 m s 以上である場合において、内部情報に関連付けられた合計処理時間が全体に占める割合が 1 0 パーセントを超えていることが判る。そして、図 1 5 は、この結果に基づき、内訳 1 件の処理時間及びこの処理時間範囲ごとに分別された内訳の件数を積算した合計件数の関係を示す、上記のステップ 6 3 の実行結果としてのヒストグラムの一例を示す。なお、このヒストグラムでは、書込処理の処理時間が 5 0 m s の場合を対象としている。このヒストグラムから、サーバ 1 0 のディスク I O 処理及びサーバ 2 0 のディスク I O 処理においてある処理時間を要する件数が、他の処理に比べて多くなっていることが判る（処理時間が 3 0 m s の場合）。このため、書込処理の処理時間が 5 0 m s のときには、サーバ 1 0 のディスク I O 処理及びサーバ 2 0 のディスク I O 処理がボトルネックとなっている可能性があることが推測できる。

【 0 0 4 7 】

( 2 ) 各サーバにおけるカーネル情報がリプライパケットに付加される例

この例では、サーバ 1 0 及びサーバ 2 0 における書込処理及び読出処理の夫々について、空きスレッド数、空きキュー数及び I O 待ち数を、内部情報として収集するものとする。

【 0 0 4 8 】

このとき、サーバ 1 0 からクライアント 4 0 に対して返信され、分析装置 3 0 にポートミラーリングされるリプライパケットに付加される、書込処理及び読出処理の実行時における夫々の内部情報は、上述したサーバ 1 0 及びサーバ 2 0 における処理を経て、夫々図 1 6 及び図 1 7 に示すような構造となる。

【 0 0 4 9 】

図 1 8 は、このリプライパケットに付加された内訳ごとの処理時間の情報に基づき、上述した分析装置 3 0 で実行される統計処理について、さらに詳細に説明したものである。

ステップ 7 1 及びステップ 7 2 は、夫々上述のステップ 6 1 及びステップ 6 2 と同じ処理であるため、説明を省略する。

【 0 0 5 0 】

ステップ 7 3 では、ステップ 7 2 で抽出した内部情報のカーネル状態の内訳、即ち、空きスレッド数、空きキュー数及び I O 待ち数を抽出する。そして、夫々のサーバの空きスレッド数、空きキュー数及び I O 待ち数を組み合わせた各グループに属する内部情報の件数が、分析対象となる全ての内部情報の件数に対して占める割合を算出する。この夫々のサーバの空きスレッド数、空きキュー数及び I O 待ち数の組み合わせは、任意に定めることができる。例えば、サーバ 1 0 の空きスレッド数が 0 であり且つ空きキュー数が 0 である一方で I O 待ち数が 0 でない場合において、サーバ 2 0 においても同様の状態が発生し

10

20

30

40

50

ている場合を1つのグループとしてもよい。また、別のグループとして、サーバ10のI/O待ち数が0である一方でサーバ20のI/O待ち数が0である組み合わせを1つのグループとしてもよい。また、内部情報のカーネル状態によっては、その内訳において生じ得る全ての組み合わせについて、夫々にグループ化してもよい。なお、この組み合わせのグループを定める規則が、所定規則に相当する。

#### 【0051】

図19は、書込処理の処理時間が50msの場合における、サーバ10及びサーバ20の空きスレッド数、空きキュー数及びI/O待ち数の組み合わせの各グループに属する内部情報の件数が、全体の件数に占める割合を算出した表の一例を示す。この表からは、サーバ10のI/O待ち数が1-50であり、且つ、サーバ20の空きスレッド数が1-10であると共に、空きキュー数が1-20の場合の割合が61パーセントであり、全体に占める割合が多いことが判る。このため、書込処理における処理時間が50msの場合には、このような稼働状態がサーバ10及びサーバ20の処理におけるボトルネックとなっていることが推測できる。

10

#### 【0052】

かかる分析装置30の処理によれば、1つの処理要求に対する処理における各サーバの稼働状態を示す情報を含む内部情報が、リプライパケットから抽出される。このため、1つの処理要求に対する処理時に各サーバにおいて複合的に生じる稼働状態を特定することができる。また、リクエストパケット及びリプライパケットのタイムスタンプの差分から、サーバ10及びサーバ20で連携して行われる処理時間が、各処理要求に対応する処理ごとに算出される。そして、この算出された処理時間と、その処理時におけるサーバ10及びサーバ20の内部情報と、が各処理要求に対する処理ごとに関連付けられているため、処理に要した時間ごとに各サーバの稼働状態を特定することが可能となる。

20

#### 【0053】

さらに、これらの内部情報及び処理時間が蓄積され、蓄積された内部情報が、これに関連付けられた処理時間によって、所定の規則で区画された処理時間範囲ごとに分別される。また、その処理時間範囲ごとに、これに属する内部情報に関連付けられた処理時間が積算された合計処理時間又は合計件数が算出される。そして、その合計処理時間又は合計件数が夫々全処理時間範囲における合計処理時間又は合計件数に対して占める割合が、所定の閾値を超える処理時間範囲が特定され、これに属する内部情報が分析対象とされる。このため、蓄積された内部情報に関連付けられた処理時間が、ある範囲内に集中しているときに、その処理時間範囲を特定すると共に、これに属する内部情報を分析対象として特定することができる。なお、このとき、合計処理時間に基づいて割合を算出した場合には、1件の処理時間が長いほど、即ち、遅延が生じているほど、その処理時間範囲に属する内部情報の合計処理時間が全処理時間範囲の合計処理時間に対して占める割合が大きくなりやすい。このため、遅延が生じている処理時間範囲を特定しやすくなる。

30

#### 【0054】

そして、(1)の統計処理では、特定された処理時間範囲に属する分析対象としての内部情報について、その処理の内訳が、処理の内訳の処理時間に応じてさらに処理時間範囲ごとに分別される。そして、処理の内訳ごとに、その内訳の処理時間を積算した合計処理時間又は内訳の合計件数が算出される。このため、分析対象とされた処理時間範囲に属する処理において、各コンピュータでは夫々どのような処理の内訳にどれ位の処理時間を費やしているのかについて、その傾向を特定することができる。

40

#### 【0055】

また、(2)の統計処理においては、特定された処理時間範囲に属する分析対象としての内部情報について、そのカーネル状態の内訳を組み合わせた各グループに属する内部情報の件数が、分析対象となる全ての内部情報の件数に対して占める割合が算出される。このため、処理時間がその特定された処理時間範囲に属する処理において、各サーバでは夫々どのようなカーネル状態になっているのかについて、その傾向を特定することができる。

50

## 【 0 0 5 6 】

このように、これらの統計処理が施された分析結果が出力されることで、複数のコンピュータに連携して1つの処理をさせる場合において、どのような稼働状態が各コンピュータ間で複合的に生じるのかについて、処理時間に応じた傾向が把握可能となる。そして、複数のサーバで連携して行った処理時間がある特定の範囲内に属しているときにおける、各サーバで複合的に生じた稼働状態について、管理者が詳細に把握することができる。このとき、遅延が生じている処理時間範囲に属する内部情報が分析対象として特定されれば、その処理時間を要した処理における各サーバの遅延原因を把握することができる。このため、複数のサーバに連携して処理をさせるときに、各サーバにどのように機能分散や負荷分散をさせればよいかについて、管理者が適切な判断をすることが可能となる。

10

## 【 0 0 5 7 】

なお、本システムでは、分析装置30において、サーバ10からクライアント40に返信されたリプライパケットをポートミラーリングにより受信し、サーバ10及びサーバ20の稼働状態の分析を行っている。しかし、このような構成に限らず、例えば、クライアント40において、サーバ10から受信したリプライパケットに付加された内部情報に基づいて、稼働状況の分析を行ってもよい。そのようにすれば、分析装置として別のコンピュータを導入する必要がなく、システムの構成を単純にすることができる。また、その場合には、スイッチ50においてポートミラーリングをする必要もない。

## 【 0 0 5 8 】

また、本システムでは、2つのサーバで処理を分散し、夫々のサーバでリプライパケットに自装置の稼働状態を示す情報を付加しているが、処理を分散するサーバの数をさらに増やしてもよい。その場合には、処理を分散された各サーバにおいて、自装置に対して処理を分散した分散元に対し、自装置の内部情報を付加したリプライパケットを返信するという、上述のサーバ20の処理と同様の処理が、処理を連携して行う各サーバにおいて行われる。そして、クライアント40にリプライパケットを返信するサーバでは、処理を連携して行った全てのサーバの内部情報を付加したリプライパケットをクライアント40に返信する。ここで、図20は、サーバ0からサーバnまでの稼働状態を示す情報を収集する場合におけるリプライパケットの構造例を示す。このように、サーバの数に関わらず、複数のサーバの稼働状態を示す情報を内部情報として1つのリプライパケットのヘッダに付加することができる。そして、分析装置30では、このリプライパケットを用いて分析を行う。

20

30

## 【 0 0 5 9 】

さらに、各サーバにおける処理要求に対応する処理の内訳及びその内訳の処理時間と、各サーバにおけるカーネル状態を示す情報と、は、1つのリプライパケットに付加してもよい。図4及び図20に示されるリプライパケットの内部情報の構造からも明らかであるように、本システムにおいては、1つのリプライパケットに付加する内部情報の内訳の種類及び数に制限を設けていない。このような構造を採用していることで、多種多様な処理に対する分析を可能にし、また、各サーバで実行したある1つの種類の処理について、多面的な分析を行うことができる。

## 【 0 0 6 0 】

さらに、上述の説明では、クライアントサーバシステムにおいて、サーバの状態を分析するべくサーバの内部情報をリプライパケットに付加する例を示したが、このようなシステム構成に限られない。例えば、複数のノードが備えるストレージにデータを分散して蓄積するマルチノードストレージにおいて、各ノード間で送受信されるリプライパケットに各ノードの内部情報を付加するとともに、そのリプライパケットを分析装置にポートミラーリングしてもよい。そして、分析装置において、その受信したリプライパケットに付加された内部情報に基づいて、各ノードの状態について分析を行ってもよい。

40

## 【 0 0 6 1 】

以上の実施形態に関し、更に以下の付記を開示する。

## 【 0 0 6 2 】

50

(付記1) 処理要求を含むリクエストパッケージが送信されてから、その処理要求に対応する処理を複数のコンピュータが連携して実行した結果に、予め各コンピュータに組み込んだ機能によりその複数のコンピュータの稼動状態を示す情報を含む内部情報が付加されたリプライパッケージが返信されるまでに要した処理時間を算出する算出ステップと、前記リプライパッケージから、前記複数のコンピュータの内部情報を抽出する抽出ステップと、前記抽出ステップにより抽出された内部情報に、前記算出ステップにより算出された処理時間を関連付けて記憶装置に蓄積する蓄積ステップと、前記蓄積ステップにより記憶装置に蓄積された前記内部情報及びその内部情報に関連付けられた処理時間に対して所定の統計処理を施し、前記複数のコンピュータの稼動状態を分析する分析ステップと、前記分析ステップにより分析された結果を出力する出力ステップと、をコンピュータに実現させることを特徴とするコンピュータ稼動状態分析支援プログラム。

10

## 【0063】

(付記2) 前記分析ステップは、第1の所定規則で区画された各処理時間範囲ごとに、前記記憶装置に蓄積された内部情報をその内部情報に関連付けられた処理時間で分別し、各処理時間範囲に属する内部情報に関連付けられた処理時間の積算値が全処理時間範囲の内部情報に関連付けられた処理時間の積算値に対して占める割合が、所定の閾値より大きな処理時間範囲に属する内部情報を、分析対象とすることを特徴とする付記1記載のコンピュータ稼動状態分析支援プログラム。

## 【0064】

(付記3) 前記分析ステップは、第1の所定規則で区画された各処理時間範囲ごとに、前記記憶装置に蓄積された内部情報を、その内部情報に関連付けられた処理時間で分別し、各処理時間範囲に属する内部情報の件数が全処理時間範囲の内部情報の件数に対して占める割合が、所定の閾値より大きな処理時間範囲に属する内部情報を、分析対象とすることを特徴とする付記1記載のコンピュータ稼動状態分析支援プログラム。

20

## 【0065】

(付記4) 前記内部情報は、各コンピュータにおける前記処理の詳細な内訳及びその各内訳の処理時間を示す情報を含んで構成され、前記分析ステップは、さらに、第2の所定規則で区画された各処理時間範囲ごとに、前記分析対象とされた内部情報に含まれる内訳を、その内訳の処理時間で分別し、各処理時間範囲に属する内訳の処理時間の積算値又は内訳の件数を、夫々の内訳について算出することを特徴とする付記2又は付記3に記載のコンピュータ稼動状態分析支援プログラム。

30

## 【0066】

(付記5) 前記内部情報は、前記処理の実行時における前記複数のコンピュータのカーネル状態の内訳を含んで構成され、前記分析ステップは、さらに、前記記憶装置に蓄積された内部情報のカーネル状態の内訳を所定規則で組み合わせた各グループに属する内部情報の件数が、前記分析対象となる全ての内部情報の件数に対して占める割合を算出することを特徴とする付記2又は付記3に記載のコンピュータ稼動状態分析支援プログラム。

## 【0067】

(付記6) 前記内部情報は、各コンピュータごと且つ前記処理によって実現される各機能ごとに、その内部情報に含まれる内訳が区別されたことを特徴とする付記1～付記5のいずれか1つ記載のコンピュータ稼動状態分析支援プログラム。

40

## 【0068】

(付記7) 処理要求を含むリクエストパッケージが送信されてから、その処理要求に対応する処理を複数のコンピュータが連携して実行した結果に、予め各コンピュータに組み込んだ機能によりその複数のコンピュータの稼動状態を示す情報を含む内部情報が付加されたリプライパッケージが返信されるまでに要した処理時間を算出する算出ステップと、前記リプライパッケージから、前記複数のコンピュータの内部情報を抽出する抽出ステップと、前記抽出ステップにより抽出された内部情報に、前記算出ステップにより算出された処理時間を関連付けて記憶装置に蓄積する蓄積ステップと、前記蓄積ステップにより記憶装置に蓄積された前記内部情報及びその内部情報に関連付けられた処理時間に対して所定の統

50

計処理を施し、前記複数のコンピュータの稼動状態を分析する分析ステップと、前記分析ステップにより分析された結果を出力する出力ステップと、をコンピュータが実行することを特徴とするコンピュータ稼動状態分析支援方法。

【 0 0 6 9 】

(付記 8) 処理要求を含むリクエストパケットが送信されてから、その処理要求に対応する処理を複数のコンピュータが連携して実行した結果に、予め各コンピュータに組み込んだ機能によりその複数のコンピュータの稼動状態を示す情報を含む内部情報が付加されたリプライパケットが返信されるまでに要した処理時間を算出する算出手段と、前記リプライパケットから、前記複数のコンピュータの内部情報を抽出する抽出手段と、前記抽出手段により抽出された内部情報に、前記算出手段により算出された処理時間を関連付けて記憶装置に蓄積する蓄積手段と、前記蓄積手段により記憶装置に蓄積された前記内部情報及びその内部情報に関連付けられた処理時間に対して所定の統計処理を施し、前記複数のコンピュータの稼動状態を分析する分析手段と、前記分析手段により分析された結果を出力する出力手段と、を含んで構成されたことを特徴とするコンピュータ稼動状態分析装置

10

【図面の簡単な説明】

【 0 0 7 0 】

【図 1】コンピュータ処理分析システムの概要図

【図 2】各サーバ、分析装置 30 及びクライアント 40 間における処理の流れの原理図

【図 3】リプライパケットの構造を示す説明図

20

【図 4】リプライパケットの構造（内部情報の詳細構造含む）を示す説明図

【図 5】サーバ 10 の第 1 の処理を示すフローチャート

【図 6】サーバ 10 の第 2 の処理を示すフローチャート

【図 7】サーバ 20 の処理を示すフローチャート

【図 8】分析装置 30 の第 1 の処理を示すフローチャート

【図 9】分析装置 30 の第 2 の処理を示すフローチャート

【図 10】分析装置 30 の第 3 の処理を示すフローチャート

【図 11】書込処理における処理の内訳及びその処理時間を収集する例におけるリプライパケットの内部情報の構造を示す説明図

【図 12】読出処理における処理の内訳及びその処理時間を収集する例におけるリプライパケットの内部情報の構造を示す説明図

30

【図 13】処理の内訳及びその処理時間を収集する例における分析装置 30 の第 3 の処理を示すフローチャート

【図 14】処理 1 件の処理時間と合計処理時間との関係を表すヒストグラムを示す説明図

【図 15】書込処理 50 ms の場合における処理の内訳の処理時間と合計件数との関係を表すヒストグラムを示す説明図

【図 16】書込処理におけるカーネル情報を収集する例におけるリプライパケットの内部情報の構造を示す説明図

【図 17】読出処理におけるカーネル情報を収集する例におけるリプライパケットの内部情報の構造を示す説明図

40

【図 18】カーネル情報を収集する例における分析装置 30 の第 3 の処理を示すフローチャート

【図 19】書込処理 50 ms の場合におけるカーネル状態の内訳の組み合わせを示す説明図

【図 20】リプライパケットの構造（内部情報の詳細構造含む）を示す説明図

【符号の説明】

【 0 0 7 1 】

10 サーバ

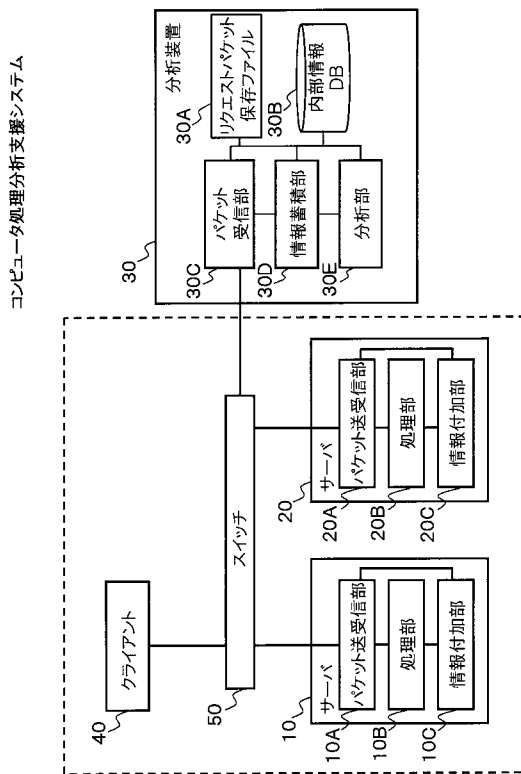
10A パケット送受信部

10B 処理実行部

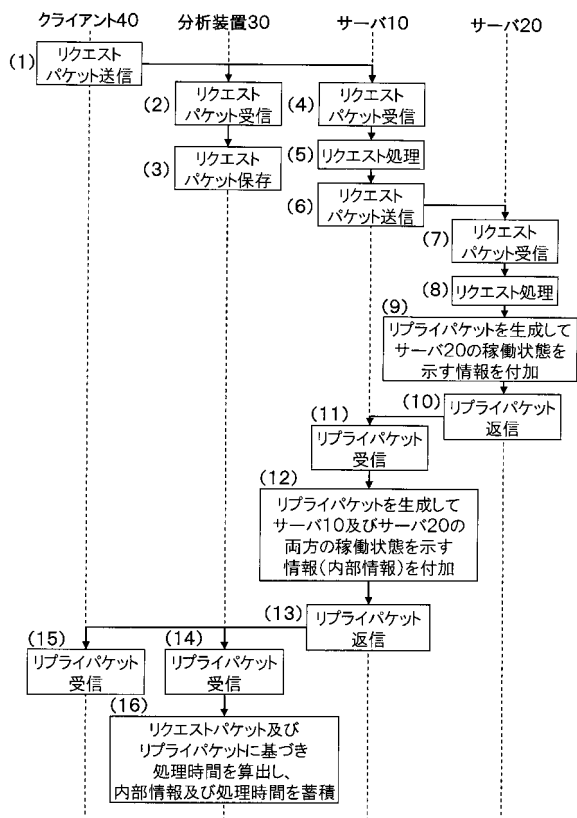
50

- 1 0 C 情報付加部
- 2 0 A パケット送受信部
- 2 0 B 処理実行部
- 2 0 C 情報付加部
- 3 0 A リクエストパケット保存ファイル
- 3 0 B 内部情報 DB
- 3 0 C パケット受信部
- 3 0 D 情報蓄積部
- 3 0 E 分析部

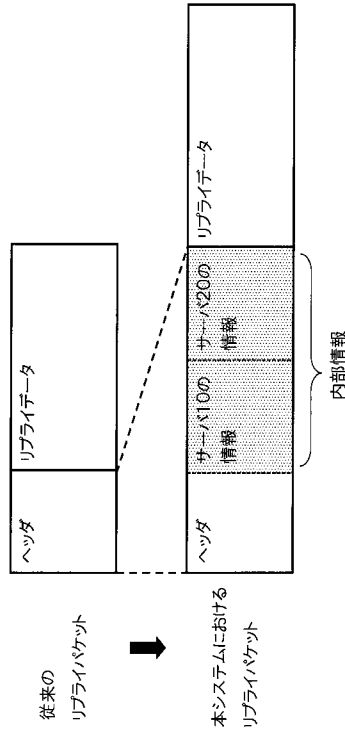
【 図 1 】



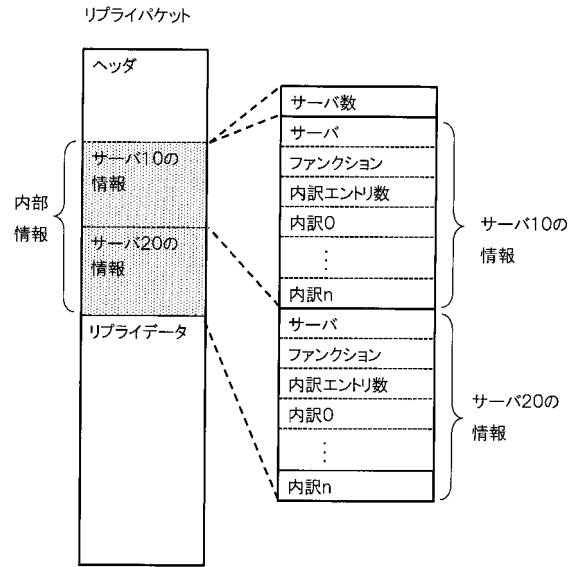
【 図 2 】



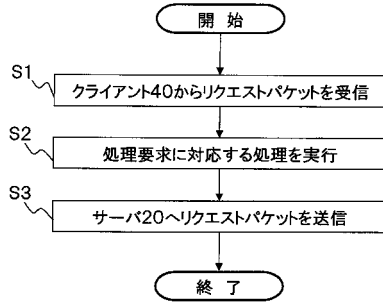
【図3】



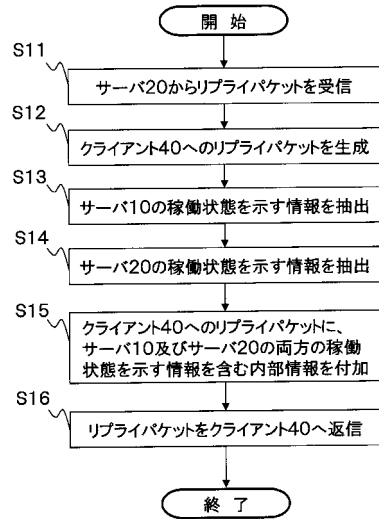
【図4】



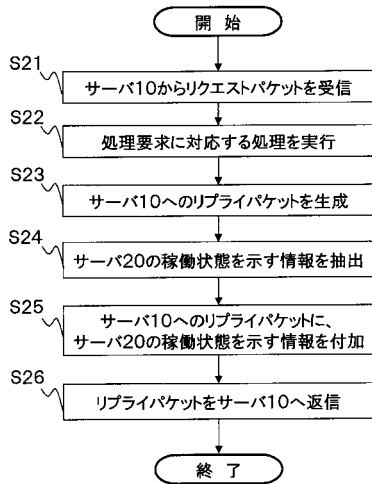
【図5】



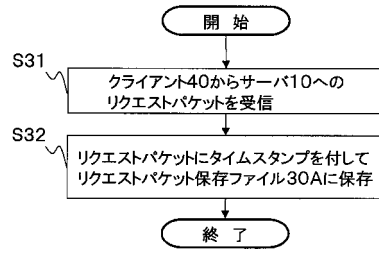
【図6】



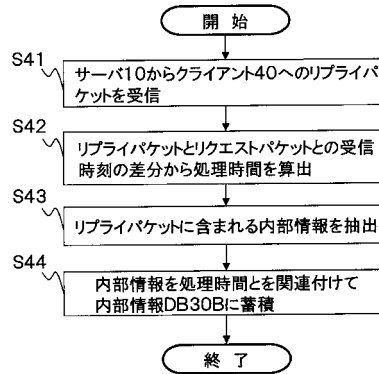
【図7】



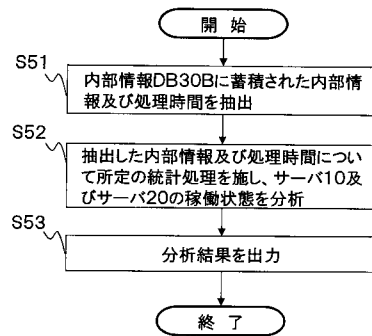
【図8】



【図9】



【図10】



【図11】

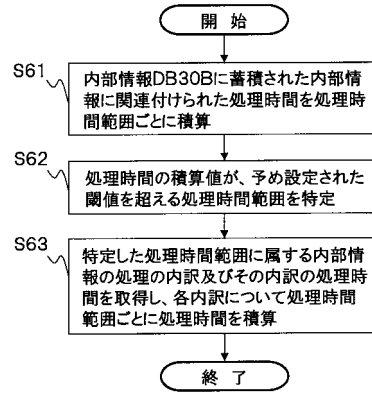
サーバ10の 内部情報	サーバ	サーバ10
	ファンクション	書込処理
	内訳エントリ数	4
	ディスクIO処理時間	AAA us
	ロック待ち処理時間	BBB us
	メタデータ更新処理時間	CCC us
サーバ20の 内部情報	サーバ	サーバ20
	ファンクション	書込処理
	内訳エントリ数	4
	ディスクIO処理時間	JJJ us
	ロック待ち処理時間	KKK us
	メタデータ更新処理時間	LLL us



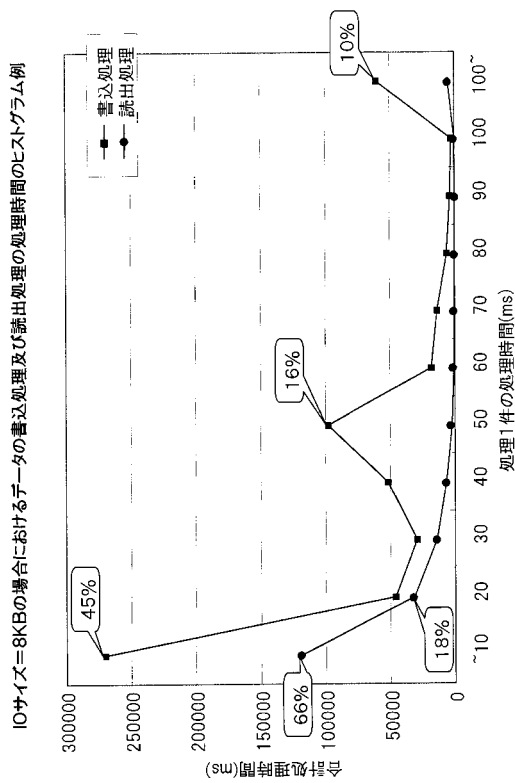
【図12】

サーバ10の 内部情報	サーバ	サーバ10
	ファンクション	読出処理
	内訳エントリー数	4
	ディスクIO処理時間	EEE us
	ロック待ち処理時間	FFF us
	メタデータ更新処理時間	GGG us
	プロトコル生成処理時間	HHH us
サーバ20の 内部情報	サーバ	サーバ20
	ファンクション	読出処理
	内訳エントリー数	4
	ディスクIO処理時間	NNN us
	ロック待ち処理時間	PPP us
	メタデータ更新処理時間	QQQ us
	プロトコル生成処理時間	RRR us

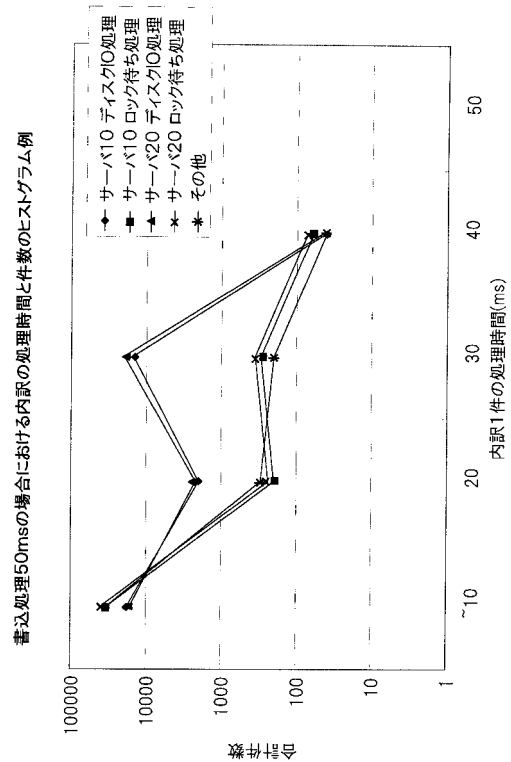
【図13】



【図14】



【図15】



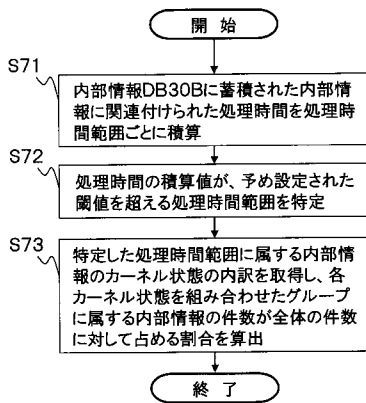
【図16】

サーバ10の 内部情報	サーバ	サーバ10
	ファンクション	書込処理
	内訳エントリー数	3
	空きスレッド数	AAA
	IO待ち数	BBB
サーバ20の 内部情報	サーバ	サーバ20
	ファンクション	書込処理
	内訳エントリー数	3
	空きスレッド数	GGG
	IO待ち数	HHH
	空きキュー数	JJJ

【図17】

サーバ10の 内部情報	サーバ	サーバ10
	ファンクション	読出処理
	内訳エントリー数	3
	空きスレッド数	DDD
	IO待ち数	EEE
サーバ20の 内部情報	サーバ	サーバ20
	ファンクション	読出処理
	内訳エントリー数	3
	空きスレッド数	KKK
	IO待ち数	LLL
	空きキュー数	MMM

【図18】

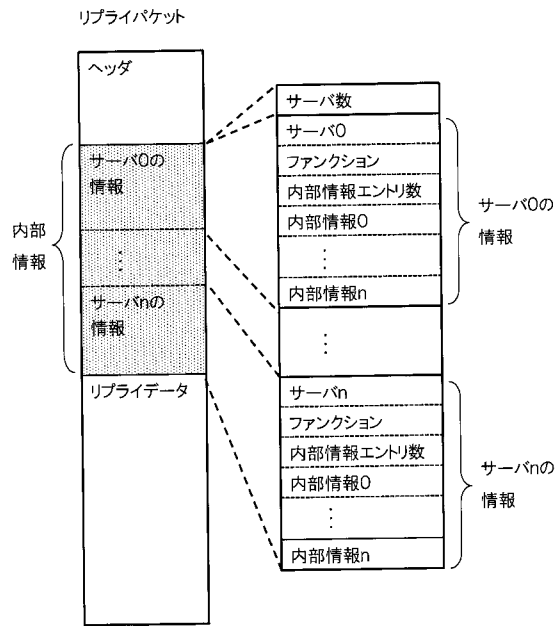


【図19】

書き処理 50msのカーネル状態の内訳の組み合わせ

		サーバ20			割合(%)
サーバ10	空きスレッド数	IO待ち数	空きキュー数	IO待ち数	61
	0	1-50	1-20	0	
	1-10	0	0	1-50	31
	0	1-50	0	1-50	8

【図 20】



## フロントページの続き

- (72)発明者 土屋 芳浩  
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 荻原 一隆  
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 田村 雅寿  
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 丸山 哲太郎  
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 渡辺 高志  
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 鴨志田 稔  
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 宮本 成礼  
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 特開平08-249261(JP,A)  
特開2005-346414(JP,A)  
特開平09-073425(JP,A)

## (58)調査した分野(Int.Cl., DB名)

G06F11/34  
G06F15/00  
G06F 9/46 - 9/54  
G06F15/16 - 15/177