

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5568776号
(P5568776)

(45) 発行日 平成26年8月13日 (2014. 8. 13)

(24) 登録日 平成26年7月4日 (2014. 7. 4)

(51) Int. Cl.	F I
G 0 6 F 9/50 (2006.01)	G O 6 F 9/46 4 6 2 Z
G 0 6 F 9/46 (2006.01)	G O 6 F 9/46 3 5 O
G 0 6 F 11/30 (2006.01)	G O 6 F 11/30 A
G 0 6 F 13/00 (2006.01)	G O 6 F 13/00 3 5 1 N

請求項の数 10 (全 55 頁)

(21) 出願番号	特願2010-248253 (P2010-248253)	(73) 特許権者	000005108
(22) 出願日	平成22年11月5日 (2010. 11. 5)		株式会社日立製作所
(65) 公開番号	特開2012-99048 (P2012-99048A)		東京都千代田区丸の内一丁目6番6号
(43) 公開日	平成24年5月24日 (2012. 5. 24)	(74) 代理人	100114236
審査請求日	平成25年2月18日 (2013. 2. 18)		弁理士 藤井 正弘
		(74) 代理人	100075513
			弁理士 後藤 政喜
		(74) 代理人	100120260
			弁理士 飯田 雅昭
		(72) 発明者	田中 剛
			東京都国分寺市東恋ヶ窪一丁目280番地
			株式会社日立製作所 中央研究所内
		(72) 発明者	上原 敬太郎
			東京都国分寺市東恋ヶ窪一丁目280番地
			株式会社日立製作所 中央研究所内
			最終頁に続く

(54) 【発明の名称】 計算機のモニタリングシステム及びモニタリング方法

(57) 【特許請求の範囲】

【請求項 1】

1 つ以上の物理計算機と、
 前記物理計算機で実行されて 1 つ以上の仮想計算機を稼働させる仮想化部と、
 前記仮想化部上で稼働する 1 つ以上の仮想計算機と、
 前記物理計算機の構成要素と、前記仮想計算機の構成要素と、を監視するモニタリング部と、
 を備えた計算機のモニタリングシステムであって、
 前記モニタリング部は、
 前記物理計算機の構成要素である物理リソースと、前記仮想化部とを併せたベースリソースと、

10

前記仮想化部上で稼働する仮想計算機の構成要素である仮想リソースと、
 前記ベースリソースの構成要素と前記仮想リソースの構成要素に一意のコンポーネント識別子を付与し、前記ベースリソースの構成要素と前記仮想リソースの構成要素のうち子ノードの構成要素に親ノードのコンポーネント識別子を設定して前記構成要素間の親子関係を管理する割当て済み ID 管理情報と、を有し、

前記ベースリソースの構成要素のうち前記物理計算機を収納するラックを頂点とした前記親ノードと前記子ノードの接続関係を示す第 1 の接続関係と、前記仮想リソースの構成要素のうち前記仮想計算機を頂点とする前記親ノードと前記子ノードの接続関係を示す第 2 の接続関係と、を第 1 のツリーとして生成し、

20

前記ベースリソースを親ノードとし、前記仮想リソースを子ノードとして、前記ベースリソースの構成要素と前記仮想リソースの構成要素の接続関係を第2のツリーとして生成し、

前記第1のツリーに含まれ、かつ、前記第2のツリーに含まれる構成要素について参照することを示す参照情報を設定し、

前記仮想リソースの接続先となる前記ベースリソースを変更した場合には、当該仮想リソースの親ノードのコンポーネント識別子を更新し、前記参照情報を維持することを特徴とする計算機のモニタリングシステム。

【請求項2】

請求項1に記載の計算機のモニタリングシステムであって、

前記ベースリソースと仮想リソースの稼働情報を収集し、前記稼働情報に所定の統計処理を施して統計情報を生成する稼働情報集計部をさらに有し、

前記モニタリング部は、

前記ベースリソースと前記仮想リソース及び前記統計情報に前記コンポーネント識別子を付与し、前記統計情報には前記稼働情報を収集した前記ベースリソースまたは仮想リソースの前記コンポーネント識別子を親のコンポーネント識別子として設定することを特徴とする計算機のモニタリングシステム。

【請求項3】

請求項2に記載の計算機のモニタリングシステムであって、

前記第2のツリーは、

前記ベースリソースの構成要素のうち仮想化または論理化して仮想リソースを生成する上位要素と、前記上位要素を仮想化または論理化して、仮想リソースの複数の構成要素に分割する下位要素とを含むことを特徴とする計算機のモニタリングシステム。

【請求項4】

請求項3に記載の計算機のモニタリングシステムであって、

前記モニタリング部は、

前記上位要素の構成要素と前記下位要素の構成要素と、前記上位要素及び下位要素の前記統計情報をひとつの画面として出力するGUI制御部を有することを特徴とする計算機のモニタリングシステム。

【請求項5】

請求項4に記載の計算機のモニタリングシステムであって、

前記GUI制御部は、

前記上位要素または前記下位要素の構成要素を受け付けて、前記受け付けた構成要素に対応付けられた前記統計情報を出力することを特徴とする計算機のモニタリングシステム。

【請求項6】

1つ以上の物理計算機と、前記物理計算機で実行されて1つ以上の仮想計算機を稼働させる仮想化部と、前記仮想化部上で稼働する1つ以上の仮想計算機と、前記物理計算機の構成要素と、前記仮想計算機の構成要素と、を監視するモニタリング部を備えて、計算機のリソースを監視する計算機のモニタリング方法であって、

前記モニタリング部が、前記物理計算機の構成要素である物理リソースと、前記仮想化部をベースリソースとして管理する第1のステップと、

前記モニタリング部が、前記仮想化部上で稼働する仮想計算機の構成要素を取得して仮想リソースとして管理する第2のステップと、

前記モニタリング部が、前記ベースリソースの構成要素と前記仮想リソースの構成要素に一意のコンポーネント識別子を付与し、前記ベースリソースの構成要素と前記仮想リソースの構成要素のうち子ノードの構成要素に親ノードのコンポーネント識別子を設定して前記構成要素間の親子関係を割当て済みID管理情報で管理する第3のステップと、

前記モニタリング部が、前記ベースリソースの構成要素のうち前記物理計算機を収納するラックを頂点とした前記親ノードと前記子ノードの接続関係を示す第1の接続関係と、

10

20

30

40

50

前記仮想リソースの構成要素のうち前記仮想計算機を頂点とする前記親ノードと前記子ノードの接続関係を示す第2の接続関係と、を第1のツリーとして生成する第4のステップと、

前記モニタリング部が、前記ベースリソースを親ノードとし、前記仮想リソースを子ノードとして、前記ベースリソースの構成要素と前記仮想リソースの構成要素の接続関係を第2のツリーとして生成する第5のステップと、

前記モニタリング部が、前記第1のツリーに含まれ、かつ、前記第2のツリーに含まれる構成要素について参照することを示す参照情報を設定する第6のステップと、

前記モニタリング部が、前記仮想リソースの接続先となる前記ベースリソースを変更した場合には、当該仮想リソースの親ノードのコンポーネント識別子を更新し、前記参照情報を維持する第7のステップと、

を含むことを特徴とする計算機のモニタリング方法。

【請求項7】

請求項6に記載の計算機のモニタリング方法であって、

前記モニタリング部が、前記ベースリソースと仮想リソースの稼働情報を収集し、前記稼働情報に所定の統計処理を施して統計情報を生成する第8のステップと、

前記モニタリング部が、

前記ベースリソースと前記仮想リソース及び前記統計情報に前記コンポーネント識別子を付与し、前記統計情報には前記稼働情報を収集した前記ベースリソースまたは仮想リソースの前記コンポーネント識別子を親のコンポーネント識別子として設定する第9のステップと、

をさらに含むことを特徴とする計算機のモニタリング方法。

【請求項8】

請求項7に記載の計算機のモニタリング方法であって、

前記第2のツリーは、

前記ベースリソースの構成要素のうち仮想化または論理化して仮想リソースを生成する上位要素と、前記上位要素を仮想化または論理化して、仮想リソースの複数の構成要素に分割する下位要素とを含むことを特徴とする計算機のモニタリング方法。

【請求項9】

請求項8に記載の計算機のモニタリング方法であって、

前記モニタリング部が、前記上位要素の構成要素と前記下位要素の構成要素と、前記上位要素及び下位要素の前記統計情報をひとつの画面として出力する第10のステップをさらに含むことを特徴とする計算機のモニタリング方法。

【請求項10】

請求項9に記載の計算機のモニタリング方法であって、

前記第10のステップは、

前記モニタリング部が、上位要素または前記下位要素の構成要素を受け付けて、前記受け付けた構成要素に対応付けられた前記統計情報を出力することを特徴とする計算機のモニタリング方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数の物理リソースと仮想リソースから構成される計算機システムのモニタリングシステムに関する。

【背景技術】

【0002】

＜運用管理における業界の傾向＞

計算機システムを運用するデータセンタなどでは、ネットワーク機器、ストレージ装置、サーバ装置、OS、サーバ仮想化ソフトウェア等の管理情報をXML形式で記述をサポートあるいはサポートを計画しているベンダが増加している。さらに、機器間や運用ソフト

10

20

30

40

50

ト間での管理情報のやり取りをXML形式のファイルで行う傾向も高まっている。

【0003】

これらの運用管理情報の標準化仕様として、DMTF (Distributed Management Task Force) が策定したシステムの管理情報のモデルCIM (Common Information Model) をベースにした管理情報の記述方法が浸透しつつある。CIMはUMLでモデリングするが、装置に関するデータはXML形式で操作される。このような背景のもと、システムの構成情報をXMLで記述することが運用管理に関わる分野では標準的手法となりつつある。

【0004】

＜モニタリングシステムについて＞

企業の情報システムやデータセンタなどでは、業務開発に合わせて構築した多数の計算機システムを統合し、一括運用管理をしていくニーズが高まっている。システム統合をしていく上でサーバ仮想化をはじめとしたシステムの仮想化技術がベースとなり広く普及しつつある。

【0005】

一方、計算機システムの定常的な維持保守作業の中で、各種装置に関する統計情報は、性能モニタリングツールで採取し、シェルスクリプト等の実行をスケジューリングできるツールを使って自動的に運用することで監視対象のサーバ等のシステム装置から定期的に収集されている。

【0006】

計算機システムの構成情報は表計算ソフトウェアのワークシート上で管理していることが多い。構成情報の更新には、構成情報の記録が正しいか否かを、管理者などが現在の構成情報の調査をする必要がある。この構成情報の調査では、SNMP等の業界標準インタフェースを用いて自動的に集められるものや、装置単位で担当者が異なるため、当該装置の担当者にインタビューして確認するなど様々であるため、現実的にはシステム全体（サーバとストレージ、ネットワークなど）の接続関係は、ワークシート上の図面で記録しているケースが多い。図面上での接続関係の調査は、グループ化やオブジェクトの順序等の機能を駆使して調査する。

【0007】

また、データセンタ等では、計算機システムの構成情報に加えて、稼動している計算機や装置の稼動に関する統計情報をソフトウェアなどで自動的に収集し、管理者などが統計情報から稼動状態の分析を行う。稼働状態分析を行うトリガとしては、顧客への稼働状態の定期レポートの作成や、統計情報の閾値監視でアラートが発生したケースがある。稼働状態分析では、調査をしたい時点の計算機システムの構成情報から、仮想サーバが共有している物理リソースを見つけ出す作業を管理者等が試行錯誤しながら実行している。あるいは、計算機システムの性能に関して問題が発生した時点の構成情報と、問題発生前の構成情報とを管理者などが比較して、問題発生箇所のあたりをつけるといったことが日常的に行われている。例えば、管理者等は統計情報から物理リソースのトータルでの使用率を見て、当該個所が性能ボトルネックを起こしているか判断する。そして、管理者等はどの仮想サーバが性能に影響を与えているか統計情報を見て判断する。

【0008】

構成情報上で注目した物理リソースの統計情報が直接採取できていない場合は、構成情報から採取できている他の統計情報から推測できないかを管理者等が検討し、既存のデータから物理リソース上の仮想サーバごとの利用状況を推測する必要がある。しかし、稼働状態監視の為の作業において、（１）構成情報の管理、（２）稼働情報と構成情報の関係調査、（３）仮想化で共有された物理資源の稼働情報の推定を、管理者等が手作業で試行錯誤しながら実行しており、非常に工数がかかっている。

【0009】

一方、データセンタ等においては仮想化技術の導入で、運用すべき仮想化サーバ台数が数万のオーダーになる可能性がある。数万台のオーダーの仮想サーバの運用管理を効率化する

10

20

30

40

50

ためには、上述の稼働状態分析を省力化していく必要がある。

【0010】

特許文献1の技術では、構成情報についてはXMLを使って記述をしているが、稼働統計情報と構成情報を紐づけて表示することは行っていない。

【0011】

特許文献2の技術は、ストレージ装置を中心に仮想サーバとストレージ装置の現在の関係を自動的に取得して構成図として表示し、システム要素の統計情報との関連を表示している。

【先行技術文献】

【特許文献】

10

【0012】

【特許文献1】特表2007-524889号公報

【特許文献2】米国特許出願公開第2008/0164344号明細書

【発明の概要】

【発明が解決しようとする課題】

【0013】

しかし、上記従来例の特許文献1の技術では、統計情報の履歴は記録するが、構成情報の履歴は取得していない。また、特許文献2の技術では、システム要素間にまたがる統計情報の分析、例えば、ストレージのデータからHBA (Host Bus Adaptor) の性能予測は行わない。

20

【0014】

特許文献2の技術では、構成情報の履歴を取得しないので、必要に応じて構成図を作りなおす必要が生じる。構成情報の履歴について、仮に表の履歴をとると仮定する。システム要素数をM、システム要素間の関係（グループ）の数をNとし、操作の回数（手間）をOとすると、以下の考え方でサーバ台数SのときにはO (S²) 回の手間（コスト）がかかる。

【0015】

つまり、仮想サーバを異なる物理サーバ上に移動するケースを想定すると、表の場合、仮想サーバが追加されると新たな行と列を追加することになるので、構成変更のたびに表を作りなおすことになる。システム要素間の関係数とサーバの数とシステム要素数の関係は、

30

$N \propto S$ かつ $M \propto S$ であることから、
 $O(N \times M) = O(S \times S) = O(S^2)$ となる。

【0016】

将来のデータセンタでは仮想サーバ台数は数万台になり、かつ仮想サーバの追加削除が日常的に頻繁に行われることが予想されている。この前提条件のもとでは、表でシステム要素間の関係を管理する場合、仮想サーバの変更が行われる度にO (S²) 回の手間(データ量と表生成のコスト)がかかる、という問題があった。

【0017】

上述の技術的背景や先行技術の問題を解決するために、計算機システムの構成情報の履歴管理と統計情報の取得を行って、構成情報の履歴と統計情報と構成情報の紐づけを行い、仮想サーバで共有している物理リソースの稼働状態を容易に分析可能とすることが必要であり、さらに、仮想サーバの追加削除等の構成変更の際のコストを低減させる、という課題があった。

40

【0018】

そこで、本発明の目的は、計算機システムの構成情報と統計情報の関連付けを表示するモニタリングシステムにおいて、仮想サーバが利用する計算機リソースの関係を漏れなくツリー構造で表現し、かつ物理リソースを共有する仮想リソースの統計情報を自動的に集計することを可能にするものである。

【課題を解決するための手段】

50

【0019】

本発明は、1つ以上の物理計算機と、前記物理計算機で実行されて1つ以上の仮想計算機を稼働させる仮想化部と、前記仮想化部上で稼働する1つ以上の仮想計算機と、前記物理計算機の構成要素と、前記仮想計算機の構成要素と、を監視するモニタリング部と、を備えた計算機のモニタリングシステムであって、前記モニタリング部は、前記物理計算機の構成要素である物理リソースと、前記仮想化部とを併せたベースリソースと、前記仮想化部上で稼働する仮想計算機の構成要素である仮想リソースと、前記ベースリソースの構成要素と前記仮想リソースの構成要素に一意のコンポーネント識別子を付与し、前記ベースリソースの構成要素と前記仮想リソースの構成要素のうち子ノードの構成要素に親ノードのコンポーネント識別子を設定して前記構成要素間の親子関係を管理する割当て済みID管理情報とを有し、前記ベースリソースの構成要素のうち前記物理計算機を収納するラックを頂点とした前記親ノードと前記子ノードの接続関係を示す第1の接続関係と、前記仮想リソースの構成要素のうち前記仮想計算機を頂点とする前記親ノードと前記子ノードの接続関係を示す第2の接続関係と、を第1のツリーとして生成し、前記ベースリソースを親ノードとし、前記仮想リソースを子ノードとして、前記ベースリソースの構成要素と前記仮想リソースの構成要素の接続関係を第2のツリーとして生成し、前記第1のツリーに含まれ、かつ、前記第2のツリーに含まれる構成要素について参照することを示す参照情報を設定し、前記仮想リソースの接続先となる前記ベースリソースを変更した場合には、当該仮想リソースの親ノードのコンポーネント識別子を更新し、前記参照情報を維持する。

10

20

【発明の効果】

【0020】

したがって、本発明によれば、仮想サーバと、仮想サーバが利用するベースリソースの関係を漏れなく、かつ少ない手数で管理することが可能となる。さらに、稼働に関する統計情報を任意の構成要素から、仮想サーバごとの使用状況を分析することが可能となる。また、木構造に従ってユーザが統計情報を分析することで効率的な統計情報の見方をナビゲートすることが可能となる。これらにより、稼働情報から計算機リソースのボトルネックを検出するまでの時間を短縮でき、多数の計算機を備えた環境における運用管理の効率を改善できる。

【0021】

30

上記特許文献1と比較して、本発明は木構造で構成情報を出力するので追加や削除に要するコストは、サーバ台数をSとすると、サーバ台数によらず当該サーバが関係している部分だけの変更にとどまるためO(1)のコストで実行でき、データ量はサーバ台数に比例するのでO(S)となり、上記従来技術と比較すると管理に要するコストとデータ量を低減できる。

【図面の簡単な説明】

【0022】

【図1】本発明の実施形態を示し、本発明を適用する計算機モニタリングシステムを含む計算機システムの全体のブロック図である。

【図2】本発明の実施形態を示し、モニタリングシステム100の内部構成を示したブロック図である。

40

【図3】本発明の実施形態を示し、稼働統計テーブル120の構成例である。

【図4】本発明の実施形態を示し、構成情報管理テーブル130の構成例である。

【図5】本発明の実施形態を示し、構成情報変更履歴管理テーブル140の構成例である。

【図6】本発明の実施形態を示し、コンポーネント履歴管理テーブル150の構成例である。

【図7】本発明の実施形態を示し、構成情報XML500aのツリー表現の一部を抜粋した説明図である。

【図8】本発明の実施形態を示し、構成情報XML500aのツリー表現の一部を抜粋し

50

た説明図である。

【図 9】本発明の実施形態を示し、構成情報XML 500bのツリー表現の一部を抜粋した説明図である。

【図 10】本発明の実施形態を示し、構成情報XML 500cのツリー表現の一部を抜粋した説明図である。

【図 11】本発明の実施形態を示し、構成情報関連管理部230が管理するコンポーネントIDの状態遷移図である。

【図 12】本発明の実施形態を示し、構成情報XML 500aに相当する割当済ID管理テーブル160の構成例である。

【図 13】本発明の実施形態を示し、構成情報XML 500aに相当する割当済ID管理テーブル160の構成例である。 10

【図 14】本発明の実施形態を示し、稼動統計情報表示時に統計情報管理部190で行われる処理のフローチャートを示す。

【図 15】本発明の実施形態を示し、モニタリングシステム100のモニタ端末350に表示される稼動統計モニタ画面の一例を示す画面イメージである。

【図 16】本発明の実施形態を示し、稼働情報収集部300の構成を示すブロック図である。

【図 17】本発明の実施形態を示し、稼働情報収集部300が使用する稼働情報収集パラメータ310の定義の説明図である。

【図 18】本発明の実施形態を示し、稼動情報収集部300で行われる処理のフローチャートを示す。 20

【図 19】本発明の実施形態を示し、稼働統計値登録部340で行われる統計値登録処理1580の詳細なフローチャートを示す。

【図 20】本発明の実施形態を示し、構成管理マネージャ360の詳細を示すブロック図である。

【図 21】本発明の実施形態を示し、構成変更要求生成部364が出力する構成情報生成ポリシー240の一例を示す説明図である。

【図 22】本発明の実施形態を示し、監視対象リソース400の一例を示す仮想計算機システムのブロック図である。

【図 23】本発明の実施形態を示し、構成情報XML 500aのツリー表現の一部を抜粋した図である。 30

【図 24】仮想化を適用した計算機システムの構成図をノード（頂点）とエッジ（辺）を用いて表現した例の説明図である。

【図 25】自動的にサービス提供レイヤーの統計情報からプラットフォームレイヤーの統計情報を算出するためのデータ表現を示した例である。

【図 26】本発明の実施形態を示し、稼働統計分析テーブル115の構成例である。

【図 27】本発明の実施形態を示し、稼働統計分析の演算を指定するポップアップウィンドウ入力画面2015の構成例である。

【図 28】本発明の実施形態を示し、自動的にサービス提供レイヤーの統計情報からプラットフォームレイヤーの統計情報を算出するフローチャートを示した図である。 40

【図 29】本発明の実施形態を示し、自動的にサービス提供レイヤーの統計情報からプラットフォームレイヤーの統計情報を算出した出力画面の構成例を示した画面イメージである。

【図 30】本発明の第1の変形例の実施形態を示し、性能分析ナビゲーション画面の構成例を示した図である。

【図 31】本発明の第1の変形例の実施形態を示し、性能分析ナビゲーションのフローチャートを示した図である。

【図 32A】本発明の実施形態を示し、構成情報XML 500aのXML表現の一部を抜粋した図の前半部である。

【図 32B】本発明の実施形態を示し、構成情報XML 500aのXML表現の一部を抜 50

粹した図の後半部である。

【図 3 3】本発明の実施形態を示し、構成情報XML 5 0 0 aのXML表現の一部を抜粋した図である。

【図 3 4 A】本発明の実施形態を示し、構成情報XML 5 0 0 aから構成情報XML 5 0 0 bに構成変更したときのXML表現の変更部分を示した図の前半部である。

【図 3 4 B】本発明の実施形態を示し、構成情報XML 5 0 0 aから構成情報XML 5 0 0 bに構成変更したときのXML表現の変更部分を示した図の後半部である。

【図 3 5 A】本発明の実施形態を示し、構成情報XML 5 0 0 aから構成情報XML 5 0 0 bに構成変更したときのXML表現の変更部分を示した図の前半部である。

【図 3 5 B】本発明の実施形態を示し、構成情報XML 5 0 0 aから構成情報XML 5 0 0 bに構成変更したときのXML表現の変更部分を示した図の後半部である。

【図 3 6】本発明の実施形態を示し、基準構成テーブルの一例を示す説明図である。

【図 3 7】本発明の実施形態を示し、x p a t h生成テーブルの一例を示す説明図である。

【図 3 8】本発明の実施形態を示し、構成管理マネージャで行われるノード生成処理の一例を示すフローチャートである。

【図 3 9】本発明の実施形態を示し、構成管理マネージャで行われる木構造生成処理の一例を示すフローチャートである。

【図 4 0】本発明の実施形態を示し、構成管理マネージャで行われるx p a t h生成テーブル設定処理の一例を示すフローチャートである。

【図 4 1】本発明の実施形態を示し、構成管理マネージャで行われる構成情報生成処理の一例を示すフローチャートである。

【図 4 2】本発明の実施形態を示し、暫定構成XML 3 8 5の構成例である。

【発明を実施するための形態】

【0 0 2 3】

以下、本発明の一実施形態を添付図面に基づいて説明する。

【0 0 2 4】

<全体ブロック図>

図 1 は、第 1 の実施形態を示し、本発明を適用する計算機モニタリングシステムを含む計算機システムの全体ブロック図である。計算機システムは、監視対象となる計算機リソース（以下、監視対象リソースとする）4 0 0 と、モニタリングシステム（モニタリング装置）1 0 0、稼動情報集計部 3 0 0 と、監視対象リソース 4 0 0 の構成を管理する構成管理マネージャ 3 6 0、およびモニタリングシステム 1 0 0 が収集した情報を参照するモニタ端末 3 5 0 とから構成される。

【0 0 2 5】

監視対象リソース 4 0 0 は、ラックや物理的なサーバ（物理計算機）と、ストレージと、ネットワークおよび物理的なサーバのリソース（物理リソース）を仮想化するソフトウェア（ハイパーバイザまたは仮想マシンモニタ）などを含むベースリソース 4 1 0 と、ベースリソース 4 1 0 の上に実装された仮想的なサーバ（仮想計算機）や、仮想的（または論理的）なストレージ、仮想的なネットワーク、および仮想的なサーバ上で動作するソフトウェア（ゲストOS）などの仮想リソース 4 2 0 を含む。

【0 0 2 6】

仮想リソース 4 2 0 には、ベースリソース 4 1 0 や仮想リソース 4 2 0 のリソース使用状況を示す稼働情報の統計値（統計情報）や、ベースリソース 4 1 0 の消費電力、装置からの排気の温度等の環境情報、および稼働状態を表すログといった情報が含まれる。

【0 0 2 7】

稼動情報集計部 3 0 0 は監視対象リソース 4 0 0 と同一の計算機上で動作する場合も、異なる計算機上で動作する場合もある。本実施形態では稼動情報集計部 3 0 0 は図 2 2 に記載の物理サーバ 4 1 1 - 1 ~ 4 1 1 - n のいずれかひとつで実行するものとする。なお、稼動情報集計部 3 0 0 を実行するサーバは、物理サーバでも良いし、仮想サーバであっ

てもよい。構成管理マネージャ 360 とモニタリングシステム 100 も同様であり、異なる計算機上で実行させても良い。いずれの場合も差異は本発明の本質には影響しない。なお、本実施形態では、構成管理マネージャ 360 とモニタリングシステム 100 も物理サーバ 411-1 ~ 411-n のいずれかひとつで実行するものとする。

【0028】

本実施形態では、監視対象リソース 400 がベースリソースと仮想リソースで構成されているものとする。

【0029】

監視対象リソース 400 のベースリソース 410 は、図 22 に示す複数の物理計算機（物理サーバ） 411-1 ~ 411-n と仮想サーバ 422 を生成するサーバ仮想化部 421 を主体にして構成される。各計算機は物理的なプロセッサとメモリ及びインタフェースを有し、インタフェースを介してストレージ装置 450 とネットワーク 430 に接続される。これらの物理的な計算機リソースを物理リソースとする。そして、ベースリソース 410 には、メモリ 413 にロードされてプロセッサ 412 で実行されるサーバ仮想化部（ハイパーバイザまたは VMM） 421 が含まれる。つまり、ベースリソース 410 は物理リソースにサーバ仮想化部 421（論理リソース）を加えたものである。したがって、以下の説明ではベースリソース 410 からサーバ仮想化部 421 を除いたものを物理リソースとする。

【0030】

そして、本発明では、ベースリソース上で仮想リソースが稼動する、と定義する。

【0031】

仮想リソース 420 は、仮想サーバ 422-1 ~ 422-m と、各仮想サーバ上で実行される OS 423 及び OS 423 がアクセスするストレージ装置 450 の LU（論理ディスク） 465-1 ~ 465-k や、仮想化された I/O デバイス（仮想 NIC を、仮想 HBA）が含まれる。仮想リソース 420 は、サーバ仮想化部 421 が仮想サーバ 422 に対して割り当てる仮想または論理的なリソースで構成される。

【0032】

なお、図示はしないが、サーバ仮想化部 421 上の仮想サーバ 422 でさらにサーバ仮想化部を稼動させる場合では、仮想サーバ 422 上のサーバ仮想化部をベースリソースとして扱うことができる。

【0033】

ベースリソース 410 と仮想リソース 420 を含む監視対象リソース 400 の具体的な例としては、例えば、図 22 のようになる。

【0034】

図 22 は、監視対象リソース 400 の一例を示す仮想計算機システムのブロック図である。ベースリソース 410 は、複数の物理サーバ 411-1 ~ 411-n とネットワーク 430 と SAN（Storage Area Network） 440 及びストレージ装置 450 及びサーバ仮想化部 421 から構成される。物理サーバ 411-1 ~ 411-n は同一の構成であるので、以下では物理サーバ 411-1 のみについて説明する。

【0035】

物理サーバ 411-1 は、プロセッサ 412、メモリ 413、NIC（Network Interface Card） 414、HBA（Host Bus Adapter） 415 から構成される。なお、図示はしないが物理サーバの起動、停止や監視を行う BMC（Baseboard Management Controller）を備えても良い。

【0036】

プロセッサ 412 は、メモリ 413 内に格納された各種プログラムを実行する。HBA 415 は SAN 440 を介してストレージ装置 450 に接続される。NIC 414 はネットワーク 430 に接続される。NIC 414 は、主にメモリ 413 上の各種プログラムからの要求に応じて他のサーバと通信する。ネットワーク 430 には、ベースリソース 410 を仮想リソース 420 へ割り当てる構成管理マネージャ 360 や、ベースリソース 41

10

20

30

40

50

0及び仮想リソース420の構成要素の稼働情報を収集する稼働情報収集部305が接続される。なお、ネットワーク430にはモニタリングシステム100やモニタ端末350が接続されても良い。

【0037】

メモリ413上には、サーバ仮想化部421が稼働することで、複数の仮想サーバ422-1～422-mを構築することができる。サーバ仮想化部421にはハイパーバイザとも呼ばれる仮想化技術が含まれる。なお、仮想サーバ422-1～422-mの総称を仮想サーバ422とする。仮想サーバ422は、それぞれ独立してOS (Operating System) 423を稼働させることができる。プロセッサ412によりサーバ仮想化部421が実行されると、サーバ仮想化部421上で複数の仮想サーバ422を構築することができる。

10

【0038】

サーバ仮想化部421は、構成管理マネージャ360の指令に応じて物理サーバ411-1の物理リソースを仮想化して、仮想サーバ422-1～422-mを主体とする仮想リソース420に割り当てる。また、サーバ仮想化部421は、NIC414やHBA415を仮想化して仮想サーバ422-1～422-mに提供する。仮想化したNIC414やHBA415は、仮想NIC (vNIC) や仮想HBA (vHBA) とする。また、サーバ仮想化部421は、仮想サーバ422-1～422-mに対してプロセッサ412 (物理プロセッサ412) を仮想化した仮想プロセッサや、メモリ413を仮想化した仮想メモリを割り当てる。なお、サーバ仮想化部421がハイパーバイザの場合、プロセッサ412の複数のコアのうち所望のコア数を仮想サーバ422-1～422-mに対して割り当てる。

20

【0039】

仮想サーバ422はストレージ装置450に予め格納されたOSイメージをサーバ仮想化部421が読み込んで実行することで、それぞれ独立した仮想サーバ422が構築される。なお、以下では物理サーバ411-1～411-nの総称を物理サーバ411とする。

【0040】

ストレージ装置450は、コントローラ480とRAID Group (以下、RG) 485-1～485-hとで構成されている。図22ではコントローラ480は1個のみ存在しているが複数個存在することもある。コントローラ480は、ディスクドライブへのデータの読み書きやSAN440と接続するファイバチャネルインターフェースの制御を行っている。RG485-1～485-hは、それぞれハードディスクドライブ(HDD)を1個以上纏めたRAIDで構成される。各RG485-1～485-hは、論理的なディスク装置 (またはストレージデバイス) であるLogical Unit (LU) 465-1～465-kに論理的に分割されている。それぞれのLU465-1～465-kは、個別に仮想サーバ422-1～422-mからアクセス可能になっている。なお、LU465-1～465-kは不揮発性の記憶媒体であればよく、HDDや不揮発性メモリで構成することができる。なお、本実施形態ではRG485-1～485-hをベ

30

40

【0041】

複数の物理サーバ411-1～411-nは、図8に示す複数のサーバシャーシ520にそれぞれ格納され、サーバシャーシ520はサーバラック510に格納されるラックマウントサーバで構成した例を示す。

【0042】

図2は、モニタリングシステム100の内部構成を示したブロック図である。モニタリングシステム100は、各種情報を格納するモニタ情報データベース110、モニタ端末350とのインタフェースを制御する統計情報GUI制御部250、構成情報GUI制御部260、履歴情報GUI制御部270と、統計情報GUI制御部250からの指令に基

50

づきモニタ情報データベース110をアクセスしてモニタ端末350へ出力する情報を決定する統計情報管理部190、構成情報管理部200、履歴情報管理部210、稼働情報を統計処理するための計算式が登録されている稼働統計分析テーブル115と、が含まれる。これらの各機能はプログラムとして実装され、後述の図22に示す記憶媒体としてのストレージ装置450に格納され、物理サーバ411のプロセッサ412によりメモリ413にロードされて実行される。

【0043】

モニタ情報データベース110の中には、稼働統計テーブル120と、構成情報管理テーブル130と、構成情報変更履歴管理テーブル140と、コンポーネント履歴管理テーブル150などが格納される。なお、構成情報変更履歴管理テーブル140およびコンポーネント履歴管理テーブル150は構成情報管理テーブル130から二次的に生成される情報なので、本発明の実施においては必ずしも必要ではなく、表示を高速化するために用意されたテーブルである。また、構成管理マネージャ360から得られた構成情報245および構成情報生成ポリシー240に従って新たな構成情報を生成し、構成情報管理テーブル130へと格納する構成情報関連管理部230と、構成情報に割当てるIDを管理する割当済ID管理テーブル160もモニタリングシステム100に含まれる。

10

【0044】

構成管理マネージャ360から受け付ける構成情報生成ポリシー240の一例としては、例えば、同一の構成情報のリビジョン更新時には、各構成情報にそれぞれ付与されるコンポーネントIDを引き継ぎ、新たな構成情報の追加時には、新たなコンポーネントIDを割り当てる。また、構成情報の削除時にはコンポーネントIDを回収し、コンポーネントIDの使い回しを禁止する等のポリシーを予め設定したものである。

20

【0045】

図20は、構成管理マネージャ360の詳細を示すブロック図である。構成管理マネージャ360は、監視対象リソース400についてベースリソース410と仮想リソース420の構成情報を格納する構成情報DB365と、ベースリソース410と仮想リソース420からプラットフォームツリー600とサービス提供ツリー610を生成するための基準構成テーブル383と、xpath生成テーブル384と、暫定構成XML385を管理する。

【0046】

構成管理マネージャ360は、構成情報DB365が更新される度にモニタリングシステム100へ通知する。あるいは、図1のモニタ端末350から入力されるツリー変換要求373を受けたツリー変換設定381が、構成情報DB365に格納された構成情報を基準構成テーブル383に格納されているシステム要素間の関係を基準にして、xpath生成テーブル384や暫定構成XML385を用いてXML形式の構成情報（構成情報XML386）を生成する。ツリー変換設定381が、この構成情報XML386を受けとった構成変更要求生成部364が、構成情報245でモニタリングシステム100へ通知する。

30

【0047】

モニタリングシステム100は、構成管理マネージャ360からの通知を受信すると、割当済みID管理テーブル160や構成情報管理テーブル130等を更新する。

40

【0048】

基準構成テーブル383と、xpath生成テーブル384は、上述の監視対象リソース400（ベースリソース410と仮想リソース420）の構成要素を、後述する2つのデータ（ツリー）の各構成要素の親子関係の定義とツリーを分類するための定義を格納したものである。基準構成テーブル383に格納する要素は、基本的にモニタ端末350の入力装置等から管理者が設定する。あるいは、取得構成情報375のデータ型式が基準構成テーブル383の形式（図36）で定義されているケースでは、取得構成情報375が基準構成テーブル383に格納される。構成管理マネージャ360は、モニタ端末350からプラットフォームツリー600の要素とサービス提供ツリー610の要素を定義する

50

基準構成入力 382 を受信すると、基準構成テーブル 383 を更新する。

【0049】

なお、基準構成テーブル 383 と、x p a t h 生成テーブル 384 と、暫定構成 XML 385 は、モニタリングシステム 100 から参照または更新することも可能である。また、モニタリングシステム 100 は、基準構成テーブル 383 と、x p a t h 生成テーブル 384 と、暫定構成 XML 385 の複製をモニタ情報データベース 110 に保持しておいてもよい。

【0050】

基準構成テーブル 383 は、プラットフォームツリー 600 の親ノードと子ノードの木構造と、サービス提供ツリー 610 の親ノードと子ノードの木構造をそれぞれ定義するものである。

10

【0051】

x p a t h 生成テーブル 384 と暫定構成 XML 385 は、モニタリングシステム 100 がプラットフォームツリー 600 とサービス提供ツリー 610 を生成するとき利用される。プラットフォームツリー 600 とサービス提供ツリー 610 の生成については後述する。

【0052】

構成管理マネージャ 360 は、サーバ仮想化部 421 に対して、仮想リソース 420 に対する物理リソース割り当てを制御する。つまり、構成管理マネージャ 360 は、仮想サーバ 422 に割り当てる物理サーバ 411 やストレージ装置 450 の監視対象リソース 400 (計算機リソース) をサーバ仮想化部 421 に対して指令する。

20

【0053】

また、構成管理マネージャ 360 は、物理サーバ 411 の稼動状態を制御する。構成管理マネージャ 360 は、仮想サーバ 422 の生成や削除あるいは移動の指令 (構成情報及び構成情報生成ポリシーを含む構成変更指令) を監視対象リソース 400 の仮想化部 421 に指令して、各物理サーバ 411 の仮想サーバ 422 を制御する。また、構成管理マネージャ 360 は、監視対象リソース 400 に指令した構成変更指令の構成情報 245 及び構成情報生成ポリシー 240 をモニタリングシステム 100 に通知する。

【0054】

モニタリングシステム 100 が監視する監視対象リソース 400 が、予め構築されている場合、構成管理マネージャ 360 に対して管理者が構成取得要求 372 を入力し、構成取得要求生成部 371 で構成取得コマンド 374 を生成し、監視対象リソース 400 に発行する。構成取得コマンド 374 の実装形態の一つとしては S N M P や C I M、S M I - S、S M A S H のようなシステムの運用管理の為に標準インタフェースが挙げられる。

30

【0055】

構成取得コマンド 374 の応答として監視対象リソース 400 からの応答である取得構成情報 375 を構成管理マネージャ 360 が受信すると、構成変更要求生成部 364 に入力され、構成情報 DB 365 に格納される。また、取得構成情報 375 のデータ形式が基準構成テーブル 383 (図 36) と同じ形式である場合は、基準構成テーブル 383 にも入力される実装をしてもよい。

40

【0056】

構成取得コマンド 374 に対して監視対象リソース 400 からの応答である取得構成情報 375 は、構成変更要求生成部 364 によって構成情報 DB 365 に格納される。次に、構成変更要求生成部 364 は、ツリー変換設定 381 に構成情報 XML を作成するように要求を出し、ツリー変換設定 381 は、構成情報 DB 365 と基準構成テーブル 383 と x p a t h 生成テーブル 384 と暫定構成 XML 385 とを使って構成情報 XML 386 を作成する。次に、構成変更要求生成部 364 は、構成情報 XML 386 を変更後構成情報 245 y として構成変更情報送信部 362 に送り、構成情報 245 として構成情報関連管理部 230 に転送する。このとき、変更前構成情報 245 x は、“空”のデータが渡される。構成情報関連管理部 230 では、図 2 の構成情報管理部 200 を経て、構成情

50

報管理テーブルに格納される。

【0057】

構成取得コマンド374で自動的に採取できない構成情報、例えば運用管理機構を持たないラックマウントのような情報処理装置を格納する棚のような場合、管理者が、構成入力3720や基準構成入力382を通して、直接的に構成管理マネージャ360に入力する。

【0058】

構成管理マネージャ360は、モニタ端末350の図示しない入出力装置から監視対象リソース400に対する構成変更要求366を受け付け、構成変更要求366が完了すると入出力装置へ構成変更完了通知367を出力する構成変更要求処理部361と、監視対象リソース400の構成情報を格納する構成情報データベース365と、構成変更要求処理部361からの構成変更要求処理部361と構成情報データベース365の構成情報に基づいて、変更前の構成情報245xと変更後の構成情報245yと構成情報の生成ポリシー240を生成する構成変更要求生成部364と、変更前構成情報245yと変更後構成情報245yを構成情報245とし、構成情報生成ポリシー240と構成情報245をモニタリングシステム100の構成情報関連管理部230と構成変更処理指示部363送信する構成変更情報送信部362と、構成情報生成ポリシー240と構成情報245に基づいて監視対象リソース400のサーバ仮想化部421に構成変更要求366を送信する構成変更処理指示部363と、を含んで構成される。

10

【0059】

構成情報データベース365は、後述の図4に示す構成情報管理テーブル130のプラットフォームツリー600とサービス提供ツリー610と同様の構成を備える。ただし、構成情報データベース365は、構成情報管理テーブル130の構成情報500（構成情報XML500a～500cの総称）に含まれる構成要素についてコンポーネントIDを有していない。コンポーネントIDは、モニタリングシステム100の構成情報関連管理部230が、新たな構成情報245を受け付けると新たなコンポーネントIDを付与する。

20

【0060】

構成管理マネージャ360は、モニタ端末350の入出力装置（図示省略）から構成変更要求366を受け付けると、構成変更要求366に含まれるポリシーと、変更前（現在の）構成情報245xと変更後の構成情報245yを受け付けて、構成情報245と構成情報生成ポリシー240をモニタリングシステム100と監視対象リソース400に送信する。

30

【0061】

そして、構成管理マネージャ360は、構成変更要求処理部361の処理が完了すると入出力装置へ構成変更完了通知367を出力し、新たな構成情報245yで構成情報データベース365を更新する。

【0062】

また、構成情報管理テーブル130の構成情報500の構成要素に付与されるコンポーネントIDは、構成情報関連管理部230が付与するモニタリングシステム100内で一意の値である。このコンポーネントIDは、構成情報500を構成する構成要素の木構造に関わりなく、一意の識別子を構成情報の構成要素毎に構成情報関連管理部230が割り当ててものである。コンポーネントIDは、監視対象リソース400の構成要素であるベースリソース410の構成要素と、仮想リソース420の構成要素のそれぞれについて一意の値が割り当てられる。

40

【0063】

すなわち、監視対象リソース400の構成要素は木構造で構成される階層的な情報であるのに対し、コンポーネントIDはモニタリングシステム100内で一意の識別子として機能する。コンポーネントIDは、後述する割当済ID管理テーブル160のように、ベースリソース410の構成要素と仮想リソース420の構成要素及び統計情報に対してそ

50

れぞれ割り当てられる。マイグレーションなどの構成要素の変更を実施するときは、このコンポーネントIDを構成変更が行われた構成要素が引き継ぐことで、構成要素に関連付けられた統計情報を、構成変更の前後で結びつけることが可能となる。

【0064】

図21は、構成管理マネージャ360の構成変更要求生成部364が出力する構成情報生成ポリシー240の一例を示す説明図である。構成変更要求生成部364は、構成変更要求処理部361から受け付けた構成変更要求366から、変更前の構成情報のノード（部分木）と変更後の構成情報のノード情報及びノード間の関係を抽出し、変更前の構成情報のノードを変更前ノード241とし、変更後の構成情報のノードを変更後ノード242とし、ノード間の関係をノード間関連243として構成情報生成ポリシー240を生成する。

10

【0065】

ノード間関連243としては、「重複しない割当」、「同一ノードを継承」、「移動元」、「移動先」、「交替元」、「交替先」などに分類される。

【0066】

「重複しない割当」のノード間関連243は、ひとつの仮想サーバ422を、昼夜で切り替えて使用する場合などで、変更対象のノードが同一で、異なる構成情報ID（162）が割り当てられた構成情報間では、コンポーネントIDが重複しないように割り当てるように、構成管理マネージャ360はモニタリングシステム100に指令する。図示の「重複しない割当」の例では、ノードaで構成要素を切り替えて、変更後のノードaの構成要素に新たなコンポーネントIDを付与することを示す。

20

【0067】

図21の「同一ノードを継承」のノード間関連243は、構成要素が変化しない場合であり、図示の例ではノードbは構成情報変更の前後で同一のコンポーネントIDを引き継ぐことを示す。また、「移動元」および「移動先」のノード間関連243は、ノードが移動する場合であり、図示の例では移動元のノードcがノードdとして移動することを示す。この場合、モニタリングシステム100ではノードcに割り当てられた構成要素のコンポーネントIDを変更後の構成情報ではノードdのコンポーネントIDとして付与する。

【0068】

「交替元」と「交替先」のノード間関連243は、現用系と待機系で系切替を行う場合であり、図示の例ではノードeを交替元として、ノードfに構成要素を移動することを示す。この場合、モニタリングシステム100ではノードeに割り当てられた構成要素のコンポーネントIDを、ノードfの構成要素に割り当てられたコンポーネントIDに書き替える。

30

【0069】

＜構成変更処理の概要＞

図1の計算機システムで行われる処理の一例を以下に示す。

【0070】

構成管理マネージャ360は、監視対象リソース400の構成を変更する度にモニタリングシステム100に構成情報245と構成情報生成ポリシー240及び基準構成テーブル383を通知し、モニタリングシステム100は通知を受けた構成情報245と構成情報生成ポリシー240と基準構成テーブル383から構成情報管理テーブル130のプラットフォームツリー600とサービス提供ツリー610を更新する。

40

【0071】

モニタリングシステム100は、例えば、新たな構成情報245を受け付けると、構成情報管理テーブル130に当該構成情報245の内容（例えば、物理サーバ411や仮想サーバ422あるいはI/Oデバイス）を構成情報134の構成情報XMLに加え、構成情報XMLに含まれる構成要素毎に計算機システム内でユニークなコンポーネントID161を割り当てる。

【0072】

50

一方、モニタ端末350から、監視対象リソース400に対する稼動情報の収集要求が稼動情報集計部300に送信されると、稼動情報集計部300は要求された監視対象リソース400に対する稼動情報の収集及び稼動情報の統計値（統計情報）に関する定義を格納し、統計値に対して統計値IDを付与する。稼動情報集計部300は、稼動情報の統計値の収集対象となった監視対象リソース400の構成情報IDと統計情報（CPU使用率、メモリ使用量など）と統計値IDを対応させてモニタリングシステム100に通知する。

【0073】

モニタリングシステム100は、稼動情報集計部300から構成情報IDと統計値IDを受け付けて、構成情報管理テーブル130のプラットフォームツリー600とサービス提供ツリー610の該当するレコードの構成情報XMLに、統計値IDと統計情報を格納する。これにより、監視対象リソース400内で一意の識別子であるコンポーネントIDと統計値IDが関連づけられる。

10

【0074】

次に、構成管理マネージャ360が仮想サーバ422の移動を実施すると、モニタリングシステム100は構成管理マネージャ360から構成情報245と構成情報生成ポリシー240を受け付けて、構成情報生成ポリシー240の内容が「交替（交替元、交替先）」であれば、構成情報245で指定された移動元のコンポーネントIDとコンポーネントIDに関連付けられた統計値IDを、移動先の構成情報のコンポーネントIDに引き継がせる。これにより、コンポーネントIDに関連付けられた統計値IDも構成情報の移動先に引き継がれる。また、構成情報管理テーブル130では、構成情報の変更の履歴をリビジョンとして記録する。

20

【0075】

稼動情報集計部300は、所定のタイミングで監視対象リソース400の監視を行い、指定された稼動情報を収集し、稼動情報に所定の統計処理を施した統計情報をモニタリングシステム100の稼動統計テーブル120に統計値IDとともに格納する。稼動情報集計部300が監視対象リソース400の監視を行うタイミングは、構成情報収集パラメータ310のモニタ間隔314で定義される。なお、統計処理は、平均値や最大値または最小値、標準偏差など、稼動情報に所定の統計処理を施して統計情報を生成する処理である。

30

【0076】

モニタ端末350が、統計情報をモニタリングシステム100に要求すると、モニタリングシステム100は指定された構成情報のコンポーネントIDを構成情報管理テーブル130から検索し、当該コンポーネントIDに関連づけられた統計値IDを取得する。そして、モニタリングシステム100は、統計値IDに対応する統計情報を稼動統計テーブル120から取得して、モニタ端末350に出力する。

【0077】

仮想サーバ422を物理サーバ411間で移動させても、構成情報生成ポリシー240のノード間関連243が「同一ノードを継承」等であれば、コンポーネントIDと統計値IDはそのまま移動先の構成情報134（構成情報XML）に引き継がれるので、仮想サーバ422の統計値IDに対応する統計情報を、移動の前後を跨いでモニタ端末350で表示することが可能となる。また、モニタ端末350が構成情報の履歴をモニタリングシステム100に要求すると、仮想サーバ422の移動の前後の構成情報の履歴をモニタ端末350に出力することができる。

40

【0078】

以下では、モニタリングシステムの構成要素、及び動作について図面を用いて説明をする。

【0079】

＜モニタ情報データベース内の各種テーブル＞

図3は稼動統計テーブル120の構成例である。統計情報管理部190が管理する統計

50

値（統計情報）123を格納するテーブルである。各レコードはデータ名称129、統計値ID121、タイムスタンプ122および1つ以上の統計値123から構成される。ここでは統計値123として平均値124、最小値125、最大値126、標準偏差127、標本数128を選んだが、一部でも良いし、これ以外の統計情報で表現してもよい。

【0080】

稼動統計テーブル120は、後述するように稼動情報集計部300が収集した監視対象リソース400の稼動情報の統計値を格納する。統計値123は、統計値ID毎に予め設定された単位を有し、例えば、プロセッサの使用率（％）やメモリの使用量（GB）や単位時間当たりのインタフェースのデータ転送量（I/Oトラフィック：GB/sec）などである。

10

【0081】

図4は、モニタリングシステム100の構成情報管理テーブル130の構成例である。構成情報管理テーブル130は、構成管理マネージャ360から受信した構成情報245を蓄積するモニタ情報データベース110のテーブルで、構成管理マネージャ360が管理する構成情報の履歴（リビジョン）を格納する。モニタリングシステム100は、起動すると構成管理マネージャ360から構成情報データベース365を受信して、現在の監視対象リソース400の構成情報245を構成情報管理テーブル130に格納する。

【0082】

構成情報管理テーブル130の各レコードは構成情報ID135、リビジョン131、開始日時132、終了日時133、および構成情報134のフィールドから構成される。ここで、構成情報134を管理する手法は色々考えられるが、たとえばXML形式の構成情報データ500を作成し、当該構成情報データへのリンクやファイル名を構成情報134のフィールドに登録する方法が考えられる。もちろん、構成情報134のフィールドにXML形式の構成情報データを格納することもできる。なお、終了日時133の値が“－”またはNULLであるレコードは、最新の構成情報であることを示している。

20

【0083】

図4の例では、構成情報ID135＝1である構成情報に対して、3つのリビジョン131の構成情報が登録されており、リビジョン131＝1とリビジョン131＝2のレコードには終了日時133が設定されているが、リビジョン131＝3のレコードには終了日時133が設定されておらず（“－”）、リビジョン131＝3の構成情報が現在使用中（最新）のリビジョンであることを示す。各リビジョンに対応する構成情報XML500a、500b、500cをそれぞれ後述する図8、図9、図10に示す。ただし、図8、図9、図10では説明を簡単化するため、稼動情報の統計情報に係る構成情報は省略してある。

30

【0084】

モニタリングシステム100は、構成管理マネージャ360から構成情報245を受け付けて、監視対象リソース400の構成情報を世代（リビジョン）毎に構成情報管理テーブル130へ格納する。構成情報管理テーブル130と構成管理マネージャ360の構成情報データベース365は、上述したように同様の構成である。このため、構成管理マネージャ360は、新たな構成情報245yを生成すると新たな構成情報ID135とリビジョン131＝1を生成する。

40

【0085】

なお、構成管理マネージャ360で監視対象リソース400の構成情報の履歴（世代）を管理しない場合では、モニタリングシステム100の構成情報管理部200がリビジョン131を管理すればよい。この場合、構成情報管理部200は、同一の構成情報ID135に対して新たな構成情報245を受け付けると、現在のリビジョン131をインクリメントした値を、新たな構成情報を格納するレコードのリビジョン131に最新の世代を示す値を設定する。

【0086】

図5は構成情報変更履歴管理テーブル140の構成例である。構成情報変更履歴管理テ

50

ーブル140は、モニタリングシステム100の履歴情報管理部210が管理する構成情報の履歴（リビジョン）を格納する。各レコードは履歴ID141、構成変更のあった時点のタイムスタンプ142、構成情報ID147、変更前リビジョン143、変更後リビジョン144、作業手順830へのリンク145、関連コンポーネントIDリスト146、の各フィールドから構成される。

【0087】

履歴ID141は、構成情報管理部210が設定する値で、一意の識別子が付与される。作業手順145は構成情報変更時にどのような変更があったのかをモニタ端末350へ出力して管理者やユーザが見てわかりやすくするためのテキスト列またはテキスト列へのポインタで、構成情報の差分から自動的に生成する方法や、構成管理マネージャ360での構成変更指示時に管理者にコメントとして入力させる方法などが考えられる。関連コンポーネントIDリスト146は、該当する構成変更の影響を受けるコンポーネントIDのリストである。

10

【0088】

図6はコンポーネント履歴管理テーブル150の構成例である。コンポーネント履歴管理テーブル150は、モニタリングシステム100の履歴情報管理部210が管理するテーブルで、構成変更の影響を受けたコンポーネントID151に関してのみレコードが登録される。

【0089】

各レコードはコンポーネントID151、新たな構成の開始日時152、当該構成の終了日時153、構成情報ID155、リビジョン154の各フィールドから構成される。終了日時153の値が“-”あるいはNULLであるレコードは、そのリビジョンが最新であることを示す。

20

【0090】

図6の例では、コンポーネントID151が「3」であるコンポーネントはリビジョン154が「1」から「2」に変更された時の影響は受けたが、リビジョン154が「2」から「3」へ変更されたときには構成変更の影響は受けていないため、リビジョン154=2が最新である。一方、コンポーネントID151が「9」であるコンポーネントは2回の構成変更の影響を共に受けているため、リビジョン154=3が最新となっている。

【0091】

＜構成情報XMLのツリー表現＞

図7は構成情報管理テーブル130の構成情報134に格納される構成情報XML500aのツリー表現の一部を抜粋した図である。図7の例は、図4に示したリビジョン131=1の構成情報134をツリー（部分木）にて表現した例である。

30

【0092】

構成情報は、プラットフォームツリー600とサービス提供ツリー610の二種類のデータ構造に分割される。

【0093】

プラットフォームツリー600は、物理リソースを主体としたシステム構成や仮想リソースを主体とした構成のように、ある一種類の“プラットフォーム”という界面で閉じた空間に存在するシステム要素の接続関係を表したデータである。換言すれば、プラットフォームツリー600は、計算機リソースのカテゴリー（プラットフォーム）毎に、構成要素の接続関係を解析した木構造のデータである。

40

【0094】

一方、サービス提供ツリー610は、ある一つの計算機リソースを仮想的（または論理的に）に1個以上の要素に分割した接続関係を表すデータである。換言すれば、サービス提供ツリー610は、監視対象リソース400を利用するクライアント計算機（図示省略）が識別可能な仮想リソースを子ノードに持ち、当該子ノードとプラットフォームツリー600の親ノードの関係を示す木構造のデータである。

【0095】

50

そして、プラットフォームツリー600の要素と、サービス提供ツリー610の親ノードの接続関係は、相互参照490（図中破線）で示される。相互参照490は、基準構成テーブル383で定義された親属性（図36中でparentと記載）要素と子属性（図36中でchildと記載）要素の接続関係を示すデータであり、図37のxpath生成テーブル383のプラットフォームツリーxpath396とサービス提供ツリーxpath397のデータで表現される。なお、親属性は上位要素となり、子属性は下位要素となる。

【0096】

なお、ネットワーク430を介して仮想サーバ422を利用するクライアント計算機が存在する場合、クライアント計算機がアクセスする仮想サーバ422、OS423、LU465等の仮想リソース420がサービス提供ツリー610に含まれる。換言すれば、仮想サーバ422のユーザが利用するクライアント計算機から識別可能な計算機リソースをサービス提供ツリー610として扱うことができる。

10

【0097】

監視対象リソース400の全体構成は、プラットフォームツリー600とサービス提供ツリー610を相互参照490のポインタ（または参照情報）で接続することで表現する。このように、計算機リソース（ベースリソース410と仮想リソース420）の構成要素を主旨の異なる別々のツリー構造にする理由を、図24を用いて示す。

【0098】

図24は、仮想化を適用した計算機システム（監視対象リソース400）の構成図をノード（頂点）とエッジ（辺）を用いて表現した例の説明図である。図24のように、仮想化された計算機システムをグラフで表現すると、木構造にはなっておらずXML形式で構成を管理することは困難である。例えば、仮想サーバ α 、 β に接続しているディスクパーティション（図24のLU0、LU1）が、論理的には分離していても物理的には同一のRAID上（図24のRG0）上に構成されている。このため、木構造で計算機システムを表現すると、図24のLU0、LU1のように、どの仮想リソースが同じ物理リソース上に展開されているか判断がつかなくなる。つまり、図24において、ストレージ装置のRaidGroup0には、LU0とLU1が存在するが、LU0にアクセスする仮想HBA（vHBA00）は、物理的なI/OデバイスであるHBA0に接続されたパスと、仮想サーバ α に接続される2つのパスが存在する。このため、木構造のデータとして表現することが難しい。

20

30

【0099】

このような仮想計算機システムで、ある仮想サーバに接続したLUに対して過大な負荷、例えば、データベースで全件検索をかけるようなケースでは、同一のRG上に展開した他のLUを使っている別の仮想サーバの性能が何の前触れもなく性能がスローダウンしてしまうケースも起こりうる。このような事態の原因分析を迅速に行うためには、物理リソースを含むベースリソース410と仮想リソース420の関係を正確に把握できることが必要であり、一つのベースリソース410を1個以上の仮想リソース420として分割して使用している部分を切り出して管理することで原因分析の効率が向上する。一方、実際の仮想サーバのシステム構築のケースにおいては、物理リソースを論理的に分割する操作は必ず一つの作業として定義される。例えば、ハイパーバイザを操作して仮想マシンを構築する操作が挙げられる。このような仮想化の作業を別に切り出すことができるという事実から、構成情報を表すデータ構造上でも仮想化をしている部分を別に切り出して管理することは、実作業とのマッチングが図りやすくモニタリングツールを使った管理作業のイメージが管理者につかみやすいという利点もある。

40

【0100】

図23は、図7から木構造のデータだけを抽出した監視対象リソース400の構成要素を表している。プラットフォームツリー600は、サーバラック510をルートノードにして物理サーバをプラットフォームとして表すツリーと、仮想サーバをプラットフォームとして表すツリーと、ストレージ装置をプラットフォームとして表すツリーが別々に定義

50

されている。

【0101】

プラットフォームツリー600で親ノードと子ノードになる構成要素の定義は、構成管理マネージャ360の基準構成テーブル383で定義される。基準構成テーブル383は、図36で示すように、管理者などがモニタ端末350で設定した監視対象リソース400の親子関係を示す定義情報で構成される。

【0102】

図36において、基準構成テーブル383は、要素名390と、プラットフォームツリー構成要素XML391と、サービス提供ツリー構成要素XML392からひとつのレコードが構成される。

10

【0103】

要素名390には、監視対象リソース400の構成要素の名称（または識別子）が格納される。プラットフォームツリー構成要素XML391には、子ノードとなる構成要素が<child>に設定され、サービス提供ツリー610で親ノードとなる構成要素の場合では<parent>に当該要素名の構成要素が設定される。サービス提供ツリー構成要素XML392には、親ノードとなる構成要素が<parent>に設定され、子ノードとなる構成要素が<child>に設定される。

【0104】

例えば、図36において、要素名390＝「サーバラック」は、プラットフォームツリー構成要素XML391に、「ServerChassis」が<child>に設定されて、子ノードがサーバシャーシ520であることを示す。そして、プラットフォームツリー構成要素XML391の<parent>には空（null）が設定されて、親ノードが存在しないルートノードであることを示す。また、要素名390＝「サーバラック」は、サービス提供ツリー構成要素XML392の各構成要素の全てが空（null）であるため、サービス提供ツリー610には存在しないことを示す。

20

【0105】

一方、要素名390＝「ハイパーバイザ」は、プラットフォームツリー構成要素XML391の<parent>に「PhysicalServer」が設定され、<child>には<xpath>として「VirtualServer」が設定される。これにより、プラットフォームツリー600では、ハイパーバイザの親ノードが物理サーバで、子ノードが仮想サーバであることを示す。また、要素名390＝「ハイパーバイザ」のサービス提供ツリー構成要素392は、<child>に<xpath>として「VirtualServer」が設定される。これにより、サービス提供ツリー610では、ハイパーバイザが親ノードで、子ノードが仮想サーバであることを示す。

30

【0106】

そして、プラットフォームツリー構成要素XML391とサービス提供ツリー構成要素XML392の<xpath>で定義された「value」が、相互参照490を構成してプラットフォームツリー600のノードと、サービス提供ツリー610の親ノードの接続関係を定義する。

【0107】

図7に示すプラットフォームツリー600は、図36の基準構成テーブル383で、サーバラック510、仮想サーバ422、ストレージ装置450をルートノードとして定義した例を示し、サービス提供ツリー610の親ノードとしてサーバ仮想化部4210、HBA4150、RG4850を設定したものである。そして、サービス提供ツリー610の各親ノードとプラットフォームツリー600の子ノードの接続関係は<xpath>によって定義される。

40

【0108】

また、モニタリングシステム100は、後述の基準構成テーブル381（図36）を参照して、モニタ端末350の入力装置から指定された構成情報について、基準構成テーブル383の定義に基づいて親ノードを抽出してサービス提供ツリー610を生成する。親

50

ノードの抽出は、基準構成テーブル381の各要素で、parent属性の要素がnull (<parent></parent>と図36では記載) になっていれば親ノードであることが抽出できる。

【0109】

なお、プラットフォームツリー600とサービス提供ツリー610は、図32A、図32Bに示すXML情報としてモニタリングシステム100の構成情報管理テーブル130に格納される。

【0110】

仮想化環境のプラットフォームとしての物理サーバを表すツリーは、サーバラック510を親ノード（頂点）とし、その下位にサーバラック510に格納されるサーバシャーシ520と、その下位にサーバシャーシ520に格納される物理サーバ411、さらにその下位には、物理サーバ411に接続されたHBA415と、物理サーバ411で実行されるサーバ仮想化部（ハイパーバイザ）412の階層関係を示す。また、物理サーバ411の下位には、物理サーバ411の統計情報560A、560Bが関係づけられている。さらに、サーバ仮想化部421の下位にはハイパーバイザの統計情報570が関連付けられている。なお、サーバラック510、サーバシャーシ520と物理サーバ411の関係等のモニタリングシステム100が識別できない構成については、管理者などがモニタ端末350から設定する。

【0111】

仮想サーバのプラットフォームを表すツリーは、仮想サーバ422を親ノードとし、その下位に仮想HBA（vHBA）460、その下位にLU465、その下位にパーティション470の接続関係を示している。仮想サーバ422には、仮想サーバ422のCPU使用率やメモリ使用量といった統計情報550が関連付けされている。さらに、パーティション470には、ディスク（LU）のスループットやアクセス要求の待ち行列長、ディスクのビジー率といった統計情報580が関連付けされている。

【0112】

ストレージ装置450のプラットフォームを表すツリーは、ストレージ装置450を親ノードとし、その下位にコントローラ480、その下位にRG（RAID Group）485が存在する接続関係を示している。

【0113】

サービス提供ツリー610には、基準構成テーブル383の各要素で定義された親子関係（parent属性とchild属性の関係）から、構成要素のツリーが生成される。例えば、図36のサービス提供ツリー構成要素XML392においては、要素名390の“ハイパーバイザ”の行にある要素（XML記述）がサーバ仮想化部（ハイパーバイザ）4210を親ノードとして表し、そのハイパーバイザの下には要素名390が“仮想サーバ”の行にある要素（XML記述）が仮想サーバ4220をして表現されている。また、それぞれのノードの接続関係は、各要素XMLのparent属性やchild属性で表現されている。後述する図25のように、サービス提供ツリー610の各要素（ツリーを構成する各ノード）にも、プラットフォームツリー600で存在した統計情報が関連付けられている。

【0114】

プラットフォームツリー600とサービス提供ツリー610は、図7に示すような相互参照490（図7の凡例）で関連するノードが相互に接続されている。データ構造上は、図32A、図32B、図33、図36に示すxpathで関連付けをしている。なお、相互参照490には、図36で示す<xpathID>が付与され、この識別子により相互参照490の特定を行うことができる。

【0115】

図7において、プラットフォームツリー600の子ノードであるサーバ仮想化部421が、サービス提供ツリー610の子ノードである仮想サーバ4220に参照される。そして、サービス提供ツリー610の子ノードである仮想サーバ4220が、プラットフォー

10

20

30

40

50

ムツリー600の仮想サーバ422に参照される。また、プラットフォームツリー600の仮想HBA465は、サービス提供ツリー610の子ノードである仮想HBA（vHBA）4650を参照し、vHBA4650の親ノードであるHBA4150がプラットフォームツリー600のHBA415を参照する。また、プラットフォームツリー600の子ノードであるRG485は、サービス提供ツリー610の親ノードであるRG4850に参照され、子ノードであるLU4650は、プラットフォームツリー600のLU465を参照する。

【0116】

このように、本発明のプラットフォームツリー600とサービス提供ツリー610により、仮想サーバ422と、仮想サーバ422が利用するベースリソースの関係を漏れなく、かつ少ない手数で管理することが可能となる。

10

5を参照する。

【0117】

<構成情報の生成>

次に、プラットフォームツリー600とサービス提供ツリー610を提供する図32A～図33の構成情報XML500aの生成について以下に説明する。

【0118】

構成管理マネージャ360は、監視対象リソース400の構成要素と基準構成テーブル383から構成情報500を生成して、構成情報データベース365に格納する。そして、構成管理マネージャ360は構成情報500をモニタリングシステム100に通知する。モニタリングシステム100では、構成情報500にコンポーネントIDを付与した情報を構成情報XML500aとして構成情報管理テーブル130に格納する。構成情報XML500aは、図32A、図32Bに示すプラットフォームツリー600と、図33に示すサービス提供ツリー610に分類される。

20

【0119】

図38～図41は、構成管理マネージャ360で行われる構成情報500の生成処理の一例を示すフローチャートである。

【0120】

構成管理マネージャ360は、モニタ端末350からの指示によって図38のフローチャートを実行し、暫定構成XML385を生成し、プラットフォームツリー600とサービス提供ツリー610のノードを設定する（ノード生成処理）。次に、構成管理マネージャ360は、図39のフローチャートを実行して、暫定構成XML385から各ノードの木構造を生成する。次に、構成管理マネージャ360は、図40のフローチャートを実行して、暫定構成XML385からプラットフォームツリー600とサービス提供ツリー610の相互参照490を示すxpathを抽出し、図37に示すxpath生成テーブル384を生成する。最後に、構成管理マネージャ360は、図41のフローチャートを実行して、暫定構成XML385とxpath生成テーブル384から、暫定構成XML385にxpathを組み込んで、構成情報500を生成し、構成情報245としてモニタリングシステム100に送る。この構成情報500は構成情報管理テーブル130に格納される。また、構成情報500は、構成情報データベース365に登録してもよい。

30

40

【0121】

まず、図38のノード生成処理について説明する。構成管理マネージャ360は、モニタ端末350から構成取得要求372を受信する（ステップ3901）。構成管理マネージャ360では、構成取得要求372を構成取得要求生成部371で監視対象リソース400の構成取得コマンド374に変換し、監視対象リソース400へ構成要素の取得を指令する（ステップ3902）。

【0122】

構成管理マネージャ360は、監視対象リソース400から取得構成情報375を受信すると、構成取得要求生成部371で取得構成情報375を受け付け、構成変更要求生成部364に送り、構成情報DB365に格納する（ステップ3903）。構成取得要求生

50

成部 371 は、受信した取得構成情報 375 を解析して監視対象リソース 400 の構成要素を抽出する（ステップ 3904）。なお、抽出された構成要素は XML で記述され、構成要素の名称が設定されているものとする。この構成要素の名称は監視対象リソース 400 側で付与してもよいし、構成取得要求生成部 371 が構成要素を解析するときに付与してもよい。

【0123】

構成管理マネージャ 360 は、ステップ 3904 で抽出した全ての構成要素についてステップ 3905 からステップ 3908 の処理を行い、親ノードを暫定構成 XML 385 に格納し、親ノードに相互参照 490 を示す `xpathID` を設定する。

【0124】

まず、ステップ 3906 では、ツリー変換設定 381 は、抽出された構成要素のうちの一つを選択して、基準構成テーブル 383 を検索し、図 36 の要素名 390 が一致したプラットフォームツリー構成要素 XML 391 またはサービス提供ツリー構成要素 XML 392 を取得する。この検索は、選択した構成要素の XML に含まれる構成要素の名称が基準構成テーブル 383 の要素名に一致すれば、プラットフォームツリー構成要素 XML 391 またはサービス提供ツリー構成要素 XML 392 を取得する。

【0125】

次に、ステップ 3907 では、ツリー変換設定 381 は、取得したプラットフォームツリー構成要素 XML 391 またはサービス提供ツリー構成要素 XML 392 があれば暫定構成 XML 385 に格納する。このとき、基準構成テーブル 383 のプラットフォームツリー構成要素 XML 391 とサービス提供ツリー構成要素 XML 392 の間に相互参照 490 の関係があれば、図 36 の「ハイパーバイザ」の XML の記述のように `<xpathID>` が含まれる。

【0126】

そして、ステップ 3908 では、ツリー変換設定 381 は、暫定構成 XML 385 に格納したプラットフォームツリー構成要素 XML 391 またはサービス提供ツリー構成要素 XML 392 内に `<xpathID>` の記述があるか否かを判定し、`<xpathID>` の記述があり、かつ `value` の値が `null` であれば構成情報 500 内（または監視対象リソース 400 内）で一意的な `xpathID` の値を `value` に設定し、`xpath` 生成テーブル 384 に登録する。

【0127】

ここで、`xpath` 生成テーブル 384 は、図 37 で示すように、`xpathID` 385 と、プラットフォームツリー `xpath` 396 とサービス提供ツリー `xpath` 397 からひとつのエントリが構成される。

【0128】

上記処理を全ての構成要素について実行することで、構成管理マネージャ 360 のツリー変換設定 381 は暫定構成 XML 385 に、プラットフォームツリー構成要素 XML 391 の親ノード及びサービス提供ツリー 610 の親ノードを格納し、相互参照 490 を示す `xpathID` があればユニークな識別子を付与する。

【0129】

次に、構成管理マネージャ 360 は、図 39 のフローチャートを実行して、図 38 の処理で求めた暫定構成 XML 385 に格納された親ノードの配下の木構造を設定する。ステップ 4001 では、構成取得要求生成部 371 が暫定構成 XML 385 に格納された全ての構成要素 XML 391、392 を抽出する。暫定構成 XML 385 には、親ノード（`parent`）の内容がブランク（空）の構成要素 XML が格納されている。抽出された構成要素 XML は親ノードの構成要素 XML 391、392 であり、以下、親ノードの構成要素 XML とする。

【0130】

次に、構成管理マネージャ 360 のツリー変換設定 381 は、図 38 のステップ 3904 で抽出された全ての構成要素のうち `parent` がブランク以外の子ノードの全てにつ

10

20

30

40

50

いて、ステップ4002、4003の処理を実行する。

【0131】

ステップ4003では、ツリー変換設定381が、parentがブランク以外の子ノードの構成要素XMLをひとつ取得する。そして、取得した子ノードの構成要素XMLが、ステップ4001で抽出した親ノードの構成要素XMLのいずれの配下に属するかを決定する。この決定は、子ノードの構成要素XMLのparentの要素名が、親ノードの構成要素XMLの要素名に一致すれば、当該親ノードを現在着目している子ノードの親とする。

【0132】

そして、ツリー変換設定381は、暫定構成XML385のうち、上記決定した親ノードの構成要素XMLの配下に、子ノードの構成要素XMLを組み込む。

10

【0133】

上記処理を全ての子ノードについて実行することで、ステップ3904で抽出された構成要素は、暫定構成XML385内で親ノードの配下に子ノードが組み込まれて、プラットフォームツリー構成要素XML391とサービス提供ツリー610が構築される。

【0134】

暫定構成XML385の例を図42に示す。基準構成テーブル383の要素が階層的に配置され、各要素に固有な名称（図42の<Server Rack Name>や<Server Chassis Name>や<Server Name>や<Hypervisor Name>や<HBAName>で定義されている名称）とxpathIDが設定されている。

20

【0135】

次に、ツリー変換設定381は、図40のフローチャートを実行してxpath生成テーブル384を更新する。図38、図39の処理では、暫定構成XML385の親ノード同士のxpathは設定されているが、プラットフォームツリー600の子ノードとサービス提供ツリー610の親ノードのxpathが設定されていない。そこで、図40のフローチャートでは、プラットフォームツリー600の子ノードとサービス提供ツリー610の親ノードのxpathを設定する。

【0136】

ツリー変換設定381は、図40のステップ4101～4103の処理をxpath生成テーブル384の全ての要素について実行する。

30

【0137】

ステップ4101では、暫定構成XML384を検索し、全親ノードを見つける。その親ノード全てについて、ステップ4102とステップ4103を実施する。

【0138】

ステップ4102では、ツリー変換設定381が暫定構成XML385を検索して、<xpathID>を検出する。検索は、深さ優先でも幅優先でもかまわない。

【0139】

ステップ4103では、ツリー変換設定381が上記ステップ4102で検出するまでに暫定構成XML385をたどったパスをxpathとして、xpath生成テーブル384のプラットフォームツリーxpath396またはサービス提供ツリーxpath397に登録する。なお、プラットフォームツリーとサービス提供ツリーの判定は、xpathIDを検出した構成要素XMLには、図36の基準構成テーブル383に格納された<platform>または<saervice>の何れかの識別子を取得することでツリー変換設定381が判定することができる。また、xpath生成テーブル384のxpathID395は、ステップ4102で検出したxpathIDを引き継げばよい。

40

【0140】

次に、ツリー変換設定381は、図41のフローチャートを実行してxpath生成テーブル384に登録されたxpathを暫定構成XML385に組み込んで、構成情報500を生成する。

50

【0141】

ツリー変換設定381は、ステップ4201～4203の処理をx p a t h生成テーブル384の要素の全てについて実行する。

【0142】

ステップ4202では、ツリー変換設定381が、x p a t h生成テーブル384の一つのエントリを選択する。そして、選択したx p a t h生成テーブル384のエントリのプラットフォームツリーx p t h 396のx p a t hを、暫定構成XML385でサービス提供ツリーの当該x p a t h I Dが設定されている部分と入れ替える。これにより、暫定構成XML385のサービス提供ツリーの当該x p a t h I Dには、プラットフォームツリーx p t h 396の記述（XML）が組み込まれる。

10

【0143】

次に、ステップ4203では、ツリー変換設定381が、現在選択しているx p a t h生成テーブル384のエントリのサービス提供ツリーx p t h 397のx p a t hを、暫定構成XML385でプラットフォームツリーの当該x p a t h I Dが設定されている部分と入れ替える。これにより、暫定構成XML385のプラットフォームツリーの当該x p a t h I Dには、サービス提供ツリーx p t h 397の記述（XML）が組み込まれる。

【0144】

上記ステップ4202、4203の処理が全てのx p a t h生成テーブル384のエントリについて終了すると、ステップ4204へ進む。ステップ4204では、ツリー変換設定381が、暫定構成XML385の内容を構成情報XML500として構成情報データベース365に格納する。

20

【0145】

以上の処理により、図32A、図32Bに示すプラットフォームツリー600と、図33に示すサービス提供ツリー610の構成情報XML500aが生成される。なお、図32A、図32B、図33の構成情報XML500aは、上述したようにモニタリングシステム100が一意のコンポーネントIDを付与したものである。

【0146】

＜統計情報の関連付け＞

次に、統計情報と構成情報の関連付けについて説明する。

30

【0147】

仮想サーバ422の統計情報としては、例えばOS423が取得したCPU使用率やメモリ使用量といった稼動情報に上述のような統計処理した情報が考えられる。すなわち、サーバ仮想化部421が仮想サーバ422へ提供する仮想CPUの使用率やメモリの使用量（または使用率）、あるいは仮想I/Oデバイス（HBAやNIC等を仮想化したI/Oデバイス）の使用率（転送量）等を仮想サーバ422の性能を示す稼動情報とし、所定の統計処理を行って統計情報とすることができる。

【0148】

一方、物理サーバ411の統計情報としては、サーバ仮想化部421自身のCPU使用率やメモリ413の使用量や、I/O（HBAやNIC）の使用率（転送量）等を仮想化部421の性能を示す稼動情報とし、所定の統計処理を行って統計情報とすることが考えられる。

40

【0149】

そして、本実施形態では、モニタリングシステム100が、構成情報XML500aの各ノードに一意なコンポーネントID（図中CID）を割り当て、このコンポーネントIDに各統計情報に設定した統計値IDを関連付けることで、構成情報の変更の前後での構成の履歴や統計情報の追跡を容易にする。なお、構成情報におけるノードとしては、物理サーバ411や仮想サーバ422に付与したコンポーネントIDをノードとして扱うことができる。同様に統計値IDもノードとして扱うようにしても良い。

【0150】

50

また、コンポーネントIDと統計値IDとを区別する必要は必ずしもなく、お互いに重複しない値をつけて管理する方法もある。以下の例では、各テーブルのフィールド数を省略するためにコンポーネントIDと統計値IDを同一フィールドで示しているが、両者を区別するようなフィールドを追加して管理する方法もある。どちらの場合も本発明の適用に当たっては本質的な差異を生じない。

【0151】

図9は図7に示した構成情報XML500aから仮想サーバαを物理サーバBへ移動させた後の構成情報XML500bのツリー表現の一部を抜粋した図である。この構成情報XML500bは、図4に示したリビジョン131=2の構成情報134をツリーにて表現した例である。リビジョン131=1から2の移動の前後で、移動した仮想サーバαに

10

【0152】

すなわち、図7に示すように、コンポーネントID=3の物理サーバAに割り当てられたコンポーネントID=9の仮想サーバを、物理サーバBへ移動したときには、物理サーバAに関連付けられた仮想サーバαのコンポーネントID=3と、当該仮想サーバの配下の仮想リソースの識別子（統計値ID）を図9で示すように削除し、移動する仮想サーバαのコンポーネントID=9及び当該仮想サーバαの配下の仮想リソースの識別子（統計値ID）と同一のコンポーネントIDと統計値IDを移動先のコンポーネントID=4の

20

【0153】

<割当済ID管理テーブル>

図11は構成情報関連管理部230が管理するコンポーネントIDの状態遷移図である。コンポーネントIDは、ベースリソース410や仮想リソース420の構成要素に構成情報関連管理部230が割り当てたモニタリングシステム100内で一意の識別子である。これらのコンポーネントIDと統計値IDは、構成情報関連管理部230が管理する割当済ID管理テーブル160（図12、図13）のコンポーネント／統計値ID161に格納される。

【0154】

構成情報関連管理部230がベースリソース410や仮想リソース420の構成要素にコンポーネントIDを割り当てると、コンポーネントIDの状態を管理するために割当済ID管理テーブル160へと登録する。

30

【0155】

コンポーネントIDは、まだ割り当てていない状態未割当170（テーブルに未登録のIDを含む）から、割り当てられると使用中171へと遷移する。また、待機用のコンポーネントなどのためにIDを予約した場合は予約済あるいは待機中172へと遷移する。使用中171と待機中172の間は、障害や移動などによる交替があった場合、お互いの間を遷移することがある。コンポーネントの削除や廃棄が行われた場合、そのIDは別の目的で再利用されないように回収済173へと遷移する。新たなIDの値の生成方法として、たとえば単調増加な自然数を使用する方法が考えられる。

40

【0156】

図12、図13は図7の構成情報XML500aの構成要素に付与される割当済ID管理テーブル160の構成例である。各レコードはコンポーネントIDまたは統計値ID161、構成情報ID162、リビジョン163、XMLパス表現164、親コンポーネントID（図中親ID）165、コンポーネントIDの状態166といったフィールドから構成される。これらの値は、モニタリングシステム100が付与するものである。

【0157】

ここでXMLパス表現164は、XMLツリーにおける各ノードを特定するために必要な情報を含む。図12、図13の例ではサーバラック510のIDと、サーバシャーシ5

50

20のIDと、物理サーバ411のIDと、仮想サーバ422のIDを順に指定する方法を示したが、この他にも物理サーバ411や仮想サーバ422にユーザがつけたサーバ名を指定する方法や、稼動するゲストOSのUUIDを指定する方法などが考えられる。いずれの方法でもノード（各コンポーネント）が一意に特定できさえすれば本発明の適用に当たっては差異を生じない。

【0158】

また、親コンポーネントID（図中親ID）165が「一」の構成要素は、プラットフォームツリー600において親ノードとなることを示している。

【0159】

＜構成情報XMLのXML表現＞

10

図32A、図32B、図33は図7に示した物理サーバ＝「A」の仮想サーバαを物理サーバ「B」へ移動する前の構成情報XML500aのXMLによる階層構造の表現である。

【0160】

図32A、図32Bでは、サーバラック510を示すServerRackノードと、サーバシャーシ520を示すServerChassisノードと、物理サーバ530を示すPhysicalServerノードと、サーバ仮想化部421を表すHypervisorノードとHBA415を表すHBAノードとが階層的にツリー構造をしている。

【0161】

また、仮想サーバ540を示すVirtualServerノードと、vHBA460を表すvHBAノード、LU465を表すLUノード、パーティション470を表すPartitionノードとが、図7に示したプラットフォームツリー600と同じく階層的にツリー構造をなしている。

20

【0162】

各ノードには、各ノードの属性としてコンポーネントIDを示すcomponent_idを持つ。またVirtualServerノードの下にはサーバ名を示すServerName、CPU使用率を示すCPUUsage、メモリ使用量を示すMemoryUsageの各ノードがあり、CPUUsageおよびMemoryUsageノードは統計値IDを示すstatistics_id属性を持つ。なお、コンポーネントIDと統計値IDを特に区別せずに管理する場合はstatistics_id属性の代わりにcomponent_id属性を持たせる実装も考えられる。どちらの場合も本質的な差異は生じない。

30

【0163】

一方、図33では、図7に示したサービス提供ツリー610の階層構造が、上記のプラットフォームツリー600と同様にして、XMLにより記述されている。図33の構成情報500aは、図36に示した基準構成テーブル381で定義された各要素390で定義されている親子関係の属性（parentとchild）をツリー上の階層構造で単一のXMLで記述したものである。

【0164】

そして、図32A、図32Bでは図7に示したプラットフォームツリー600の各ノードと図33のサービス提供ツリー610の各ノードとの間の相互参照490として、上述のxpathが設定されている。

40

【0165】

図34A、図34B、図35、図35Bは図9に示した物理サーバ＝「A」の仮想サーバαを物理サーバ「B」へ移動する際の構成情報XML500aと構成情報XML500bを生成する場合のXML表現での変更手順を示した図である。

【0166】

図34A～図35Bでは、構成情報XML500aにおける仮想サーバα配下のノードを、構成情報XML500bでは物理サーバAの配下から物理サーバBの配下へと移動させる。この処理において、仮想サーバαを含む配下のコンポーネントIDおよび統計値I

50

Dは同一のIDを引き継ぐことで、移動前後における稼働の統計情報や変更履歴の追跡を可能とする。また、移動する仮想サーバを含むプラットフォームツリーの記述はpathだけの変更だけでよく、サービス提供ツリー610で新たに仮想サーバを追加するだけでサーバの移行に対して対処可能である。

【0167】

＜稼働統計情報表示＞

図14は稼働に関する統計情報をモニタ端末350へ表示する時にモニタリングシステム100の統計情報管理部190で行われる処理のフローチャートを示す。この処理は、統計情報GUI制御部250がモニタ端末350から統計情報の要求を受け付けたときに実行される。この要求は、統計情報の表示期間の開始日時を変数bに格納し、終了日時を変数eに、コンポーネントの構成情報IDを変数dに、コンポーネントIDを変数cに格納する。

10

【0168】

まずステップ1300で、統計情報管理部190は統計情報GUI制御部250から統計情報表示期間の開始日時bと終了日時eを得る。ステップ1310へ進む。

【0169】

ステップ1310では、構成情報管理部200が構成情報GUI制御部260から、指定されたコンポーネントの構成情報ID135=dとコンポーネントID=cを得る。ステップ1320へ進む。

【0170】

20

ステップ1320では、履歴情報管理部210が、統計情報管理部190および構成情報管理部200から得た変数b、e、d、cを元に、コンポーネント履歴管理テーブル150から<d、c>をキーとして、開始日時152<eかつ終了日時153≥bとなるリビジョン154のリスト(R1、R2、...、Rn)を得る。ここで各リビジョンの開始日時152を(b1、b2、...、bn)、終了日時153を(e1、e2、...、en)とする。ei(ただし、i=1~n)は最新リビジョンの場合NULLの場合もある。ステップ1330へ進む。

【0171】

次に得られた各リビジョンRi(ただし、i=1~n)について、ステップ1340~1390までを繰り返す。

30

【0172】

ステップ1340では、構成情報管理テーブル130から、構成情報ID135=d、リビジョン131=Riであるレコードを選び、構成情報管理テーブル130の構成情報134から構成情報XML500(500a~500c)を取得する。ステップ1350へ進む。なお、以下の説明では構成情報XML500a~500cの総称を構成情報XMLとする。

【0173】

ステップ1350では、取得した構成情報XML500についてコンポーネントIDが変数cの配下の統計値IDのリスト(s1、s2、...、sm)を得る。ただし、統計値IDのリストはsjで表され、j=1~mである。ステップ1360へ進む。

40

【0174】

次に得られた各sjに対してステップ1370~1380を繰り返す。

【0175】

ステップ1370では、稼働統計テーブル120から統計値ID121=sjをキーにして、bi≤タイムスタンプ122<eiとなるレコードのタイムスタンプ122と統計値123の組<t、v>のリストを得る。ステップ1380に進む。

【0176】

ステップ1380では、稼働情報GUI制御部250に<t、v>のリストを送り、このリスト<t、v>とコンポーネントIDによりモニタ端末350の表示装置にグラフを出力する。

50

【0177】

以上が稼動に関する統計情報の表示時に統計情報管理部190で実行される処理のフローチャートである。

【0178】

上記処理により、統計情報管理部190は、同一のコンポーネントIDに関連づけられた統計値ID121について、マイグレーションなどの構成変更があった場合にも、構成変更の時点を跨いで連続的に統計情報を表示させることが可能となる。

【0179】

図15は本発明を適用したモニタリングシステム100のモニタ端末350に表示される稼動統計モニタ画面の一例である。

10

【0180】

稼動統計モニタ画面は、図中左側に表示期間を表示する部分700とラックやシャーシといったプラットフォームツリー600に従って構成管理を行う部分710を備える。また、図中右側には構成要素および稼動に関する統計情報を表示するエリアが設定され、図中上部から物理構成情報（ベースリソース410の構成要素）を表示する物理構成情報表示エリア720、仮想構成情報（仮想リソース420の構成要素）を表示する仮想構成情報表示エリア730、統計情報を表示する稼動統計情報表示エリア740から構成される。

【0181】

物理構成情報表示エリア720と仮想構成情報表示エリア730および稼動統計情報表示エリア740は連動しており、物理構成情報エリア720において指定されたコンポーネント（ここでは物理サーバA）についての仮想構成情報（仮想リソースの構成要素）の詳細が仮想構成情報表示エリア730に表示される。また、この場合、物理サーバAに対応した稼働統計情報が740に表示される。

20

【0182】

ここで仮想構成情報表示エリア730の仮想サーバαをモニタ端末350の入力装置（例えば、マウスなどのポインティングデバイス）で指定した場合、仮想サーバαに関連する統計情報のみが抽出されて稼働統計情報表示エリア740に表示される。図示の例では、稼働統計情報表示エリア740の左側の領域に仮想サーバαのCPU使用率が時系列的なグラフで表示され、右側の領域には仮想サーバαのI/O使用率（リード、ライトの使用率など）が時系列的なグラフで表示される。

30

【0183】

なお、物理構成情報表示エリア720の物理サーバA～Cの何れかをモニタ端末350の入力装置で指定した場合には、選択した物理サーバに関連する統計情報が抽出されて稼働統計情報表示エリア740に表示される。また、図15では、物理構成情報表示エリア720にサーバ仮想化部421を省略した例を示したが、物理サーバ上にサーバ仮想化部が階層的に稼動している場合には、物理サーバとサーバ仮想化部を表示することができる。

【0184】

このように、着目するコンポーネントが物理コンポーネントか仮想コンポーネントかに応じて、稼働統計情報740に表示されるグラフの種類が変わる点が本モニタリングシステム100の特徴の一つである。

40

【0185】

上記図15の稼動統計モニタ画面は、モニタリングシステム100の統計情報GUI制御部250、構成情報GUI制御部260、履歴情報GUI制御部270により一つの画面情報が生成され、モニタリングシステム100は当該画面情報をモニタ端末350へ出力する。そして、モニタ端末350では、受信した画面情報を表示装置に描画する。

【0186】

<稼働情報収集部の動作>

以下図16～図19を用いて稼働情報集計部300で行われる処理の詳細を示す。

50

【0187】

図16は稼働情報集計部300の構成を示したブロック図である。稼働情報集計部300は稼働情報収集パラメータ310を保持し、稼働情報収集パラメータ310の内容はパラメータ設定インタフェース345を介してモニタ端末350の入力装置等から設定される。

【0188】

稼働情報収集パラメータ310の内容は、監視対象リソース400の通常の運用では最初に収集する項目を決めて設定された後は、特に監視対象リソース400に大きな変更が加わらない限り変更されることはない。稼働情報集計部300は、さらに稼働統計情報収集パラメータ310に従って収集コマンドを実行する収集コマンド実行320、および収集コマンドの実行結果である稼働情報を受け取って所定の処理を行う収集コマンド結果取得部330、さらに収集した稼働情報の中で予め設定した条件に合致した結果から統計値を生成し、稼働統計テーブル120へと格納する稼働統計値登録部340、およびタイム341から構成される。

10

【0189】

図17は稼働情報集計部300が使用する稼働情報収集パラメータ310の定義である。図17は稼働情報収集パラメータ310の1つのパラメータの行（またはレコード）を示し、構成情報ID311、パラメータ生成式312、収集コマンド313、モニタ間隔314、およびフィルタ条件315a、315b、統計情報生成式316a、316b、統計値ID検索式317a、317bからなる統計情報生成リスト318a、318bが1つ以上、から構成される。なお、以下では、フィルタ条件315a、315bの総称をフィルタ条件315とし、統計情報生成式316a、316bの総称を統計情報生成式316とし、統計値ID検索式317a、317bの総称を統計値ID検索式317とし、統計情報生成リスト318a、318bの総称を統計情報生成リスト318とする。

20

【0190】

稼働情報収集パラメータ310の内容については、上述のように管理者などが操作するモニタ端末350の入力装置等から設定される。構成情報ID311には、図4の構成情報管理テーブル130の構成情報ID135に対応する値が格納される。収集コマンド313には、対象となる構成情報ID311に対して実行させる稼働情報の収集コマンドを格納する。モニタ間隔314には、モニタ端末350で設定する稼働情報の収集周期が格納される。

30

【0191】

モニタ端末350は、パラメータ設定インタフェースに対して統計値の生成を要求し、構成情報管理テーブル130の構成情報ID135を指定して、上記各パラメータを設定する。設定されたパラメータは新たなレコードとして稼働情報収集パラメータ310に格納される。稼働情報集計部300は、構成情報IDと統計値IDをモニタリングシステム100に通知し、構成情報管理テーブル130の構成情報500に統計値IDが格納される。

【0192】

パラメータ生成式312は、後述するように、構成情報XML500に含まれるサーバ名やIPアドレス等を指定するための式が設定される。

40

【0193】

収集コマンド313の一例としては、後述するように、稼働情報を収集するプロトコルであるSNMPに従ったコマンド等が設定される。

【0194】

フィルタ条件315の一例としては、図17に示す\$FilterElemで示すような条件が設定される。

【0195】

統計情報生成式316の一例としては、図17に示すように、収集した稼働情報の百分率や平均値や合計値を求める式が設定される。

50

【0196】

統計値ID検索式317の一例としては、図17に示すように、構成情報XML500のインスタンスに対応する統計値IDを求める式が設定される。

【0197】

図18に稼働情報集計部300で行われる処理のフローチャートを示す。

【0198】

稼働情報集計部300は、図17のように予め設定された稼働情報収集パラメータ310をパラメータとして読み込んで起動される。ここでは説明を簡易にするため、稼働情報収集パラメータ310の各行ごとに別パラメータを与えて起動するとしているが、一つのプログラムで複数の収集コマンドを発行できるようにしても良い。

10

【0199】

ステップ1500で稼働情報集計部300のプログラムが起動され、ステップ1510へ進む。

【0200】

ステップ1510では、稼働情報集計部300は図17の稼働情報収集パラメータ310を参照し、構成情報ID311をキーにして構成情報管理テーブル130から最新のリビジョン131を持つレコードの構成情報XML500を取得し、ステップ1520へと進む。

【0201】

ステップ1520では、稼働情報集計部300が構成情報XML500から稼働情報収集パラメータ310のパラメータ生成式312に従って、収集コマンド313の起動に必要なパラメータを補完し、ステップ1530へ進む。

20

【0202】

ここでパラメータ生成式312は、構成情報XML500に含まれるインスタンスを特定するのに必要なパラメータ（サーバ名やIPアドレス等）を指定するための式であり、一例としてはXMLパス形式で指定する方法が考えられる。例えば図32で示したXML表現から、仮想サーバのサーバ名を抽出するためのXMLパス形式の一例は"/VirtualServer/ServerName"となる。

【0203】

一般にパラメータ生成式312にマッチするノードは一つの構成情報XML500に複数存在するために、生成されるパラメータは複数のリストとなる。

30

【0204】

ステップ1530では、ステップ1520で生成されたパラメータの数に応じて稼働情報集計部300のインスタンスを生成する。インスタンスの生成は、複製により生成することもできる。インスタンスを複製する方法としては例えばUnix（登録商標）のforkシステムコールを使って複製する方法や、スレッドを分ける方法などが考えられる。なお、稼働情報の収集処理を並行に実行することさえできれば必ずしもプロセスやスレッドを複製する必要は無い。ステップ1540に進む。

【0205】

ステップ1540では、収集コマンド実行部320が上記生成されたパラメータを引数にして収集コマンド313を監視対象リソース400に対して実行する。収集コマンド313としては、稼働情報を収集するプロトコルであるSNMPに従ったコマンドや、予めインストールしたエージェントに対する通信、あるいはリモートシェルなどを使って稼働状態を取得するコマンド（Linux（登録商標）でのvmstatやiostatなど）を発行する方法などが考えられる。ステップ1550へ進む。

40

【0206】

ステップ1550では、収集コマンド結果取得部330はステップ1540で実行した収集コマンド313の結果をパイプ（標準出力）あるいはファイル経由で取得する。ステップ1560へ進む。

【0207】

50

ステップ1560では、パラメータとして与えられたフィルタ条件315と統計情報生成式316と統計値ID検索式317の組からなる稼働情報生成リスト318のそれぞれについて、ステップ1570～ステップ1590までを繰り返し実行する。

【0208】

ステップ1570では、得られた結果がフィルタ条件315に一致しているかを判断する。フィルタ条件315に一致していない場合は集計対象に該当する統計値は含まれていないため、ステップ1590へ進む。フィルタ条件315に一致していた場合は稼働統計値登録部340がステップ1580の統計値の登録処理を行ってからステップ1590へと進む。

【0209】

ステップ1580の統計値の登録処理については後に図19にて詳細に説明する。

【0210】

ステップ1590では、フィルタ条件315と統計情報生成式316と統計値ID検索式317の組の統計情報生成リスト318の残りについて、ステップ1560に戻り繰り返す。全て完了した場合ステップ1600へと進む。

【0211】

ステップ1600では、モニタ間隔314が経過しているか否かをタイマ341をチェックし、経過するまで待つ。経過後ステップ1540へと戻る。

【0212】

以上が稼働情報集計部300の動作を示すフローチャートである。

【0213】

続いて図19に稼働統計値登録部340で行われる統計値登録処理1580の詳細なフローチャートを示す。

【0214】

ステップ1610で、稼働統計値登録部340は、稼働情報収集パラメータ310の統計情報生成式316に従い、統計値を算出する。ここで、統計値とは、稼働統計テーブル120に登録する統計値123に対応する統計値であり、収集した稼働情報の平均値や最小値、最大値、標準偏差あるいは標本数、などから構成される。1つの値のみから統計値を算出する場合は、平均値＝最小値＝最大値となり、標準偏差は0、標本数は1とする。稼働情報収集パラメータ310の統計情報生成式316は、ステップ1550で取得した収集コマンド313の出力結果から、統計値を生成する式である。統計情報生成式316は、パーセントの数字の場合に稼働情報を100で割るといった処理や、2つの値の平均や合計を求めるといった処理が考えられる。ステップ1620へ進む。

【0215】

ステップ1620では、構成情報XML500から統計値ID検索式317に従って統計値IDを取得し変数sに格納する。ここで、統計値ID検索式317とは、構成情報XML500のインスタンスから、対応する統計値ID（構成情報XMLノード上でインスタンスノードの配下にあるケースが多い）を求める式であり、一例としてはXMLパス形式を使うことが考えられる。たとえば、図32、図33の構成情報XML500aの場合、ハイパーバイザのインスタンスを示す式が"/ServerRack/ServerChassis/PhysicalServer/Hypervisor"であり、その仮想サーバのCPU使用率の統計値IDを示すXMLパス式は". /CPUUsage@statistics__id"となる。ステップ1630に進む。

【0216】

ステップ1630では、タイマ341からタイムスタンプtを取得し、ステップ1640へ進む。

【0217】

ステップ1640では、稼働統計テーブル120に、統計値ID121=s、タイムスタンプ122=t、統計値123=(v1、v2、...、vn)を登録する。

【0218】

以上で統計値の登録処理 1 5 8 0 が完了する。

【0 2 1 9】

＜ベースリソースの構成要素の自動的な統計情報分析＞

図 2 9 を用いて監視対象リソース 4 0 0 の仮想リソース 4 2 0 で共有しているベースリソース 4 1 0 の構成要素について行う自動的な統計情報分析の概要を説明する。

【0 2 2 0】

図 2 9 は、図 1 のモニタリングシステム 1 0 0 のモニタ端末 3 5 0 で表示する構成情報と統計情報の関連を一画面で表示している例である。

【0 2 2 1】

ここでは、仮想サーバ 4 2 2 で共有している H B A 4 1 5 の稼働状況を調査し、どの仮想サーバ 4 2 2 が最も当該 H B A 4 1 5 を使用しているのかをモニタリングシステム 1 0 0 の利用者が調査をするシナリオを想定している。

【0 2 2 2】

まず、モニタリングシステム 1 0 0 のユーザがモニタ端末 3 5 0 で、監視したいシステム要素をマウス等の入出力装置でクリックする。図 2 9 では、H B A 0 をクリックしている (2 0 4 0)。そして、モニタリングシステム 1 0 0 は後述の統計情報分析処理を実行した後、H B A 0 上のデータ転送速度 (スループット) を仮想サーバ毎に色 (またはパターン) を変えて積み上げグラフの形式で表示する (2 0 4 1)。このような仮想サーバ毎の統計情報の積み上げグラフを作成するには、直接該当するデータを計測できているケース、例えば、サーバ仮想化部 4 2 1 が仮想サーバ 4 2 2 毎に割り当てているプロセッサ 4 1 2 の C P U 使用率のようなケースでは、計測した数値をそのまま積み上げグラフで表示すればよい。しかし、図 2 9 のように H B A 4 1 5 では仮想サーバ 4 2 2 毎の統計情報を得るためには、特別なハードウェアを搭載していなければ計測はできない。そこで、H B A 0 を経由するディスクアクセスのスループットを H B A 0 のデータスループットとみなして、仮想サーバ 4 2 2 毎のスループット値の積み上げグラフを生成することで H B A 0 のスループットが分析できる。本発明では、このような分析処理を自動化すること目的としている。

【0 2 2 3】

＜データ構造＞

図 2 5 を用いて統計情報のデータ構造について説明する。基本的には図 7、図 3 2、図 3 3 を用いて説明したプラットフォームツリー 6 0 0 とサービス提供ツリー 6 1 0 を相互参照で結合したものである。プラットフォームツリー 6 0 0 では、物理サーバ 4 1 1 の下に H B A 4 1 5 が存在している。H B A 4 1 5 に関連付けられている統計情報 5 8 2 は空集合になっている。一方、仮想サーバ 4 2 2 の下には v H B A 4 6 0 が存在し、その下には L U 4 6 5 が存在し、さらにその下にはパーティション 4 7 0 が存在する。v H B A 4 6 0 に関連付けられている統計情報 5 8 1 は空集合となっている。さらに、パーティション 4 7 0 に紐づけられた統計値 (ディスクの統計情報) 5 8 0 がある。

【0 2 2 4】

サービス提供ツリー 6 1 0 に属する統計情報のデータについて説明する。サービス提供ツリー 6 1 0 では、H B A 4 1 5 0 を親ノードとし、その下に v H B A 4 6 5 0 が存在する。H B A 4 1 5 0 は統計情報 4 5 8 0 と関連付けられている。そして、H B A 4 1 5 と H B A 4 1 5 0 の間と、v H B A 4 6 0 と v H B A 4 6 5 0 の間には相互参照が張られている (図 2 5 の破線)。

【0 2 2 5】

サービス提供ツリー 6 1 0 上の統計情報 4 5 8 0 (統計値 I D = 4 0 0 1 ~ 4 0 0 2)、4 5 8 2 (統計値 I D = 4 1 0 1 ~ 4 1 0 4) は、図 2 6 に示す稼働統計分析テーブル 1 1 5 で統計情報が定義されている。図 2 6 の稼働統計分析テーブル 1 1 5 は、図 2 のモニタリングシステム 1 0 0 の構成要素の一つである。稼働統計分析テーブル 1 1 5 の構成は、統計値 I D 2 0 0 1、データ名 2 0 0 2、表示 2 0 0 3、演算 2 0 0 4 から一つのレコードが構成される。

【0226】

表示2003はモニタ端末350に表示するグラフの形式を指定するものである。表示2003では子ノードに相当するデータは表示対象ではないことを示す“－”が設定される。親ノードに対しては演算2004の演算内容で集計したデータを表示2003の形式でモニタ端末350の表示装置に画面表示される。表示2003に設定される“合計”は子ノードの数値をすべて加算した数値で、“積み上げ”は子ノードの数値を、それぞれのノードの数値が占める割合が分かるように積み上げたグラフを意味している。

【0227】

演算2004の列には、統計値の演算方法が指定されている。演算名が“集合{ }”は、“{ }”で指定された集合をそのまま集合として返す。演算名が“合計値[]”は、“[]”で指定された集合の数値をすべて加算したものが返り値となる。図では記載されていないが、演算としては、平均値、最大値、最小値など他の集合演算も指定できる。

10

【0228】

稼働統計分析テーブル115の設定は、図27の稼働時計分析の演算を指定する入力画面2015から、管理者がモニタ端末350の入力装置から入力して指定する。この入力画面2015は、管理者が図2のモニタ端末350から構成情報GUI制御部260に対してモニタ端末350で表示されるメニュー画面から呼び出す。そして、構成情報管理部200が構成情報管理テーブル130から、サービス提供ツリー610の一覧を検索し、検索結果を構成情報GUI制御部280がモニタ端末350上にウィンドウ（画面2015）を新規に開いて表示する。

20

【0229】

このとき同時に統計情報の種類（使用率やスループットなど）も同時に検索する。入力画面2015は、サービス提供ツリー610の構成要素の一覧を場変換2011の列に表示し、モニタリングシステム100のユーザは演算2012の列に表示されるプルダウンメニュー2014から所望の演算を選択する。さらに統計値の種類2013もプルダウンメニュー（図示なし）で選択する。なお、統計値の種類2013は、統計値ID2001が指し示す稼働統計分析テーブル115のデータ名2002に対応し、統計情報の単位（Read/secやwrite/sec等）を設定する。また、統計情報の単位のRead/secやwrite/secは、データ転送速度＝データ量（MB）／secや転送速度＝トランザクション数／secなど予め設定した単位を意味する。

30

【0230】

<分析のフローチャート>

図28を用いてモニタリングシステム100が実行する図29の分析処理2035の流れを説明する。

【0231】

モニタ端末350で、モニタリングシステム100のユーザが注目するシステム要素（仮想化されている物理リソース＝ベースリソース410の構成要素）を図29の画面に表示したシステム構成図上から選択操作（クリック等）を受け付ける（ステップ2020）とステップ2021に進む。

【0232】

40

ステップ2021では、選択操作で選択されたシステム要素（図25のHBA415）に紐づけられた統計情報をモニタリングシステム100が構成情報管理テーブル130を検索する。統計情報が登録されている場合、ステップ2030に遷移し、統計情報を画面上に表示して処理は終了。統計情報が登録されていない場合は、ステップ2023に移る（ステップ2022）。

【0233】

ステップ2023では、モニタリングシステム100がプラットフォームツリー600から相互参照のあるサービス提供ツリー610をたどり、サービス提供ツリー610の子ノード（図25のvHBA4650）を検索する。そして、モニタリングシステム100は子ノード（vHBA4650）に紐付けられたサービス提供ツリー610上のHBAの

50

統計情報として、統計値 I D = 4 0 0 1 と統計値 I D = 4 0 0 2 を取得する。次に、モニタリングシステム 1 0 0 は稼働統計分析テーブル 1 1 5 を参照して検索した子ノードの統計値 I D のレコードで演算 2 0 0 4 で指定されたデータを参照する（ステップ 2 0 2 4）。

【0 2 3 4】

図 2 6 では、統計値 I D 4 0 0 1 または 4 0 0 2 の演算 2 0 0 4 列で指定されているリンク先要素の統計値を検索する。ここでいうリンク先とは、x p a t h（図 3 2 A、図 3 2 B）で指定されているプラットフォームツリー 6 0 0 上のノードを指している。ステップ 2 0 2 5、2 0 2 6、2 0 2 7 では、サービス提供ツリー 6 1 0 上で注目している（クリックした）親ノードに接続しているすべての子ノードで統計値を検索し、発見した統計値を返す。上記処理の終了後、ステップ 2 0 2 8 に進む。

10

【0 2 3 5】

ステップ 2 0 2 8 では発見した統計情報を、稼働統計分析テーブル 1 1 5 で指定した親ノードの表示形式に統計情報管理部 1 9 0 で変換し、ステップ 2 0 2 9 に進む。

【0 2 3 6】

ステップ 2 0 2 9 では、サービス提供ツリーの親ノードと相互参照のあるプラットフォームツリーのノードの統計値として数値を統計情報 G U I 制御部 2 6 0 がモニタ端末 3 5 0 の画面上に表示する（ステップ 2 0 3 0）。

【0 2 3 7】

上記処理によりモニタ端末 3 5 0 の表示装置には、図 2 9 の統計情報（2 0 4 1）が所定のグラフで表示される。

20

【0 2 3 8】

<変形例>

本発明の変形例として、性能分析ナビゲーションシステムを説明する。システムの概略を、図 3 0 を用いて説明する。構成情報と統計情報の関連を一つの画面上に表示することは先に述べた実施形態と同様である。モニタ端末 3 5 0 の表示装置の画面上の分析ナビゲーション開始ボタン 2 1 0 2 を入力装置等により選択する（クリック）と、物理リソースを含むベースリソース 4 1 0 を共有している仮想リソース（仮想リソース 4 2 0）の利用状況の積み上げグラフが表示される（第一ステップ：共有リソースの状況 2 1 0 1）。

【0 2 3 9】

30

例えば、物理サーバ 4 1 1 の物理 C P U（プロセッサ 4 1 2）を各仮想サーバ（0～2）でどのくらい利用しているかを判別するためのグラフ（C P U 使用率）や、H B A を各仮想サーバ毎にどのくらい使用しているのかを判別するための各仮想サーバの I O スループットの積み上げグラフ（H B A のデータ転送速度）等が挙げられる。

【0 2 4 0】

管理者は、このグラフを見ることで、どの仮想サーバ 0～2 が物理リソースを多く使い、どの仮想サーバ 0～2 の動作に影響を及ぼしているかを判定することが可能である。更に、管理者がモニタ端末 3 5 0 の入力装置で監視対象切換えボタン 2 1 0 3 を操作することで、各仮想サーバ 0～2 で計測された統計情報（例えば、C P U 使用率や I O スループットなど）のグラフが表示され、それぞれの稼働状態を検証できる。このようにして、仮想化された計算機システムでの性能分析を支援することが可能である。

40

【0 2 4 1】

性能分析ナビゲーションシステムの動作を図 3 1 のフローチャートを用いて説明する。初めにナビゲーションの初期画面をモニタリングシステム 1 0 0 がモニタ端末 3 5 0 に表示する（ナビゲート 2 1 0 0 開始）。システムの管理者は、モニタ端末 3 5 0 の入力装置等から表示されている画面上のナビゲート開始ボタン 2 1 0 2 をクリックし、モニタリングシステム 1 0 0 はモニタ端末 3 5 0 の入力装置からの信号を受け付ける（ステップ 2 1 1 1）と、ステップ 2 1 1 2 に進む。

【0 2 4 2】

ステップ 2 1 1 2 では、モニタリングシステム 1 0 0 では、図 2 の統計情報 G U I 制御

50

部は統計情報管理部 190 にナビゲート開始を通知する。統計情報管理部 190 は構成情報管理部 200 にサービス提供ツリー 610 の親ノードをリストアップするように指示し、ステップ 2113 に進む。

【0243】

ステップ 2113 では、構成情報管理部 200 でリストアップされた親ノードに対して、図 28 の分析処理 2035 を実行する（ステップ 2114）。ただし、ステップ 2502 でユーザがクリックして注目するノードを指定する代わりに、リストアップされたサービス提供ツリー 610 の親のノードが関連付けられているプラットフォームツリー 600 のノードが入力される。これは例えば図 25 のサービス提供ツリー 610 の HBA4150 に相互参照がつけられているプラットフォームツリー 600 の HBA415 のことである。一つのノードについて処理が終了したら次のノードに対して同じ処理を実行する（ステップ 2115）。

10

【0244】

全ての親ノードの処理が終了したらモニタ端末 350 に積み上げグラフを表示する。次にステップ 2117、2118、2119、2120 で全ての仮想サーバの統計情報についてグラフを作成する処理をする。グラフの表示に必要な処理は既に説明した図 14 の手順で実行されるので、ここでは省略する。

【0245】

全ての仮想サーバ 0～2 の情報を一度にモニタ端末 350 の表示装置の画面に表示しても、管理者はどこを見るべきか判断が難しくなるため、ステップ 2118 ではモニタリングシステム 100 が画面上の監視対象切替ボタン 2103 の操作を受け付けると、表示するデータを切替える。ステップ 2119 では、モニタリングシステム 100 が、各仮想サーバ 0～2 に関連付けられた統計情報をモニタ端末 350 の表示装置へ出力する。上記処理を全ての仮想サーバ 0～2 について繰り返した後（ステップ 2120）、処理を終了する。

20

【0246】

上記処理によって、監視対象リソース 400 を構成するベースリソース 410 の物理的な構成と仮想リソース 420 の仮想化された構成について、モニタ端末 350 の表示装置へ表示を行い、ベースリソース 410 の構成要素である物理サーバ 411 を共有する全ての仮想サーバのリソースの利用状況（2101）と仮想サーバ毎のリソースの利用状況をそれぞれグラフとして表示することができる。

30

【0247】

以上のように、本発明の実施形態及び変形例によれば、監視対象リソース 400 について、物理リソースを含むベースリソース 410 と、ベースリソース 410 のリソースを使用する仮想リソースを含む仮想リソース 420 に分ける。そして、ベースリソース 410 と仮想リソース 420 の構成要素の関係を漏れなく、かつ少ない手数で構成情報と統計情報の管理を行うことが可能となる。ベースリソース 410 または仮想リソース 420 の統計情報は、モニタリングシステム 100 によって、任意の構成要素から、物理リソース（ベースリソース 410 のリソース）を共有する仮想リソースごとの統計情報を容易に分析することが可能となる。また、構成情報のデータ構造に従って統計情報を分析することで、モニタリングシステム 100 を利用する管理者等に効率的な統計情報の管理をナビゲートすることが可能となる。これらにより、稼動情報から計算機リソースのボトルネックを検出するまでの時間を短縮でき、多数の計算機、特に多数の仮想サーバを備えた計算機システムにおける運用や管理に要する管理者の効率を改善できる。

40

【0248】

上記従来技術の特許文献 1 と比較して、本発明では木構造のプラットフォームツリー 600 とサービス提供ツリー 610 に分類し、木構造の相互参照を定義しておくことで構成要素の追加や変更を容易に管理することが可能となる。

【0249】

監視対象リソース 400 の構成要素の追加や削除のコストは、サーバ台数を S とすると

50

、サーバ台数によらず当該サーバが関係している部分だけの変更にとどまるためO(1)回のコストで実行でき、データ量はサーバ台数に比例するのでO(S)であるといえる。したがって、従来技術と比較するとコスト面で本特許の方式でコストを低減できる。

【0250】

また、上記実施形態では、モニタリングシステム100と構成管理マネージャ360と稼動情報集計部300が異なる計算機で稼動する例を示したが、これらが同一の計算機で稼動することもでき、この場合にはモニタリング部と構成管理部と稼動情報集計部をソフトウェアとして実装することができる。

【産業上の利用可能性】

【0251】

10

以上のように、物理サーバと仮想サーバを備えて構成情報または統計情報の履歴を管理する計算機システムに適用することができる。

【符号の説明】

【0252】

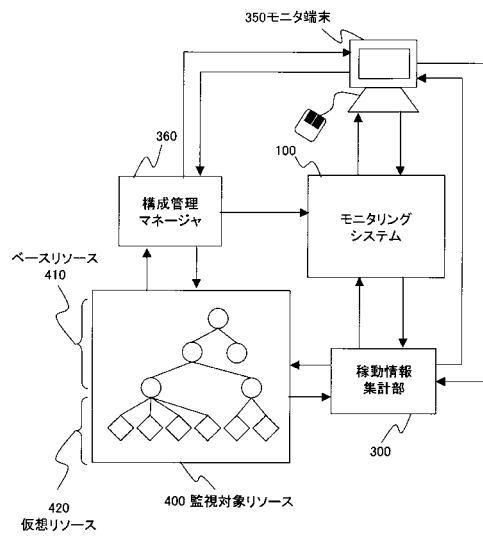
- 100 モニタリングシステム
- 110 モニタ情報データベース
- 115 稼働統計分析テーブル
- 120 稼動統計テーブル
- 130 構成情報管理テーブル
- 150 コンポーネント履歴管理テーブル
- 160 割当済ID管理テーブル
- 180 構成関連管理テーブル
- 190 統計情報管理部
- 200 構成情報管理部
- 210 履歴情報管理部
- 250 統計情報GUI制御部
- 260 構成情報GUI制御部
- 270 履歴情報GUI制御部
- 300 稼動情報収集部
- 350 モニタ端末
- 360 構成管理マネージャ
- 381 ツリー変換設定
- 383 基準構成テーブル
- 384 xpath生成テーブル
- 400 監視対象リソース
- 410 ベースリソース
- 420 仮想リソース
- 490 相互参照
- 600 プラットフォームツリー
- 610 サービス提供ツリー

20

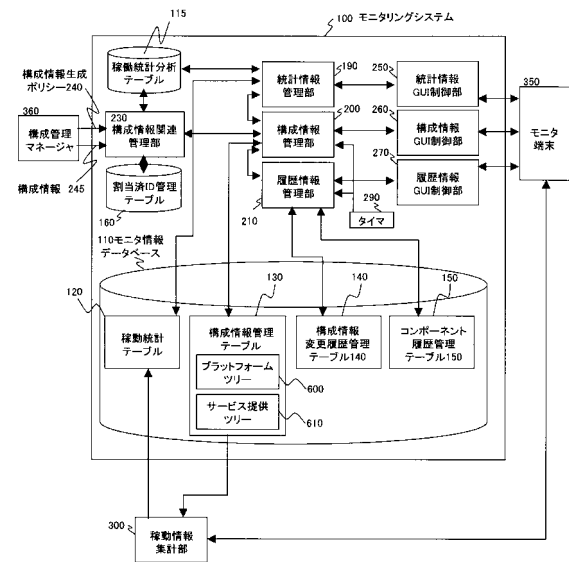
30

40

【図 1】



【図 2】



【図 3】

120 稼働統計テーブル

			統計値123				
統計値ID	データ名称	タイムスタンプ	平均値	最小値	最大値	標準偏差	標本数
406	CPU使用率	'09/11/11 16:59:45	20%	10%	30%	0.1	5
406	CPU使用率	'09/11/11 17:04:02	40%	30%	45%	0.15	3

【図 5】

140 構成情報変更履歴管理テーブル

履歴ID	タイムスタンプ	変更前 リビジョン	変更後 リビジョン	作業手順	関連コンポーネントID/リスト
1	'09/11/11 17:00:00	1	2	830a	6,7,9,10,21,22,23,27
2	'09/12/01 12:00:00	1	2	830b	9,11,12,13,14,19,21,23,25,26

830a作業手順
仮想サーバー(CID=9)を物理サーバー(CID=3)からへ移動

830b作業手順
仮想サーバー(CID=9)を仮想サーバー(CID=9)を削除

【図 4】

構成情報ID	リビジョン	開始日時	終了日時	構成情報
1	1	'09/01/01 00:00:00	'09/11/11 17:00:00	500a
1	2	'09/11/11 17:00:00	'09/12/01 12:00:00	500b
1	3	'09/12/01 12:00:00	-	500c

130 構成情報管理テーブル

【図 6】

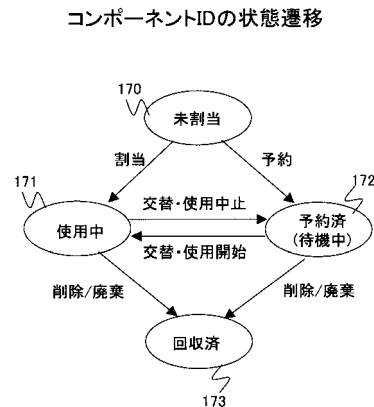
150 コンポーネント履歴管理テーブル

コンポーネント ID	開始日時	終了日時	構成情報ID	リビジョン
3	'09/01/01 00:00:00	'09/11/11 17:00:00	1	1
3	'09/11/11 17:00:00	-	1	2
9	'09/01/01 00:00:00	'09/11/11 17:00:00	1	1
9	'09/11/11 17:00:00	'09/12/01 12:00:00	1	2
9	'09/12/01 12:00:00	-	1	3

物理サーバα(CID=3)は
レビジョン3では
変更の影響無し

仮想サーバα(CID=9)は
レビジョン1,2,3で
変更の影響有り

【図 11】



【図 12】

160 割当済ID管理テーブル (図13に続く)

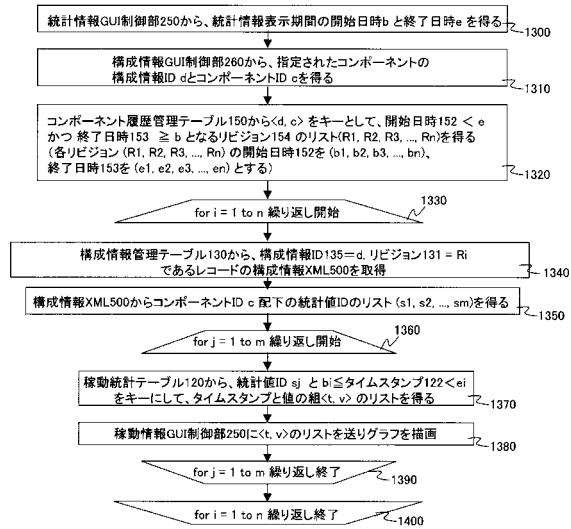
コンポーネント/統計値ID	構成情報ID	リビジョン	XMLパス表現	親ID	状態
1	1	1	/Rack[ID=1]	-	使用中
2	1	1	/Rack[ID=1]/Chassis[ID=1]	1	使用中
3	1	1	/Rack[ID=1]/Chassis[ID=1]/PhysicalServer[ID=A]	2	使用中
4	1	1	/Rack[ID=1]/Chassis[ID=1]/PhysicalServer[ID=B]	2	使用中
5	1	1	/Rack[ID=1]/Chassis[ID=1]/PhysicalServer[ID=A]/Hypervisor[ID=0]	3	使用中
6	1	1	/Rack[ID=1]/Chassis[ID=1]/PhysicalServer[ID=A]/HBA[ID=0]	3	使用中
7	1	1	/Rack[ID=1]/Chassis[ID=1]/PhysicalServer[ID=B]/HBA[ID=1]	4	使用中
8	1	1	/Rack[ID=1]/Chassis[ID=1]/PhysicalServer[ID=B]/Hypervisor[ID=1]	4	使用中
9	1	1	/VirtualServer[ID=α]	-	使用中
10	1	1	/VirtualServer[ID=α]/vHBA[ID=0]	9	使用中
11	1	1	/VirtualServer[ID=α]/vHBA[ID=0]/LU[ID=sda]	10	使用中
12	1	1	/VirtualServer[ID=α]/vHBA[ID=0]/LU[ID=sdb]	10	使用中
13	1	1	/VirtualServer[ID=α]/vHBA[ID=0]/LU[ID=sda]/Partition[ID=sda1]	12	使用中
14	1	1	/VirtualServer[ID=α]/vHBA[ID=0]/LU[ID=sdb]/Partition[ID=sdb1]	13	使用中
15	1	1	/Storage[ID=0]	-	使用中
16	1	1	/Storage[ID=0]/Controller[ID=0]	15	使用中
17	1	1	/Storage[ID=0]/Controller[ID=0]/RG[ID=0]	16	使用中
18	1	1	/Storage[ID=0]/Controller[ID=0]/RG[ID=1]	16	使用中
19	1	1	/Cnv_Hypervisor[ID=0]	-	使用中
20	1	1	/Cnv_Hypervisor[ID=0]/Cnv_VirtualServer[ID=0]	19	使用中
21	1	1	/Cnv_Hypervisor[ID=0]	-	使用中
22	1	1	/Cnv_HBA[ID=0]	-	使用中
23	1	1	/Cnv_HBA[ID=0]/Cnv_vHBA[ID=0]	22	使用中
24	1	1	/Cnv_RG[ID=0]	-	使用中
25	1	1	/Cnv_RG[ID=0]/Cnv_LU[ID=0]	24	使用中
26	1	1	/Cnv_RG[ID=0]/Cnv_LU[ID=1]	25	使用中

【図 13】

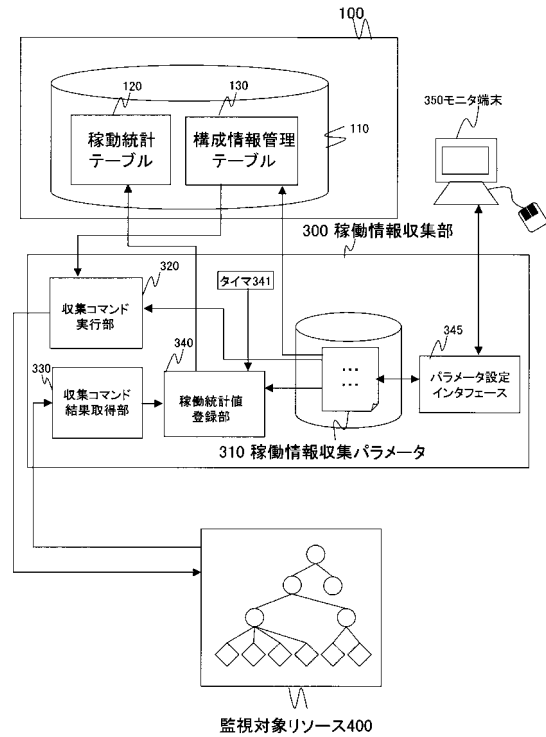
160 割当済ID管理テーブル (図12の続き)

コンポーネント/統計値ID	構成情報ID	リビジョン	XMLパス表現	親ID	状態
400	1	1	/VirtualServer[ID=α]/CPUUsage	9	使用中
401	1	1	/VirtualServer[ID=α]/MemoryUsage	9	使用中
408	1	1	/Rack[ID=1]/Chassis[ID=1]/PhysicalServer[ID=A]/HypervisorCPUUsage	8	使用中
409	1	1	/Rack[ID=1]/Chassis[ID=1]/PhysicalServer[ID=A]/HypervisorMemoryUsage	8	使用中
1001	1	1	/VirtualServer[ID=α]/vHBA[ID=0]/LU[ID=sda]/Partition[ID=sda1]/Read_per_sec	13	使用中
1002	1	1	/VirtualServer[ID=α]/vHBA[ID=0]/LU[ID=sda]/Partition[ID=sda1]/Write_per_sec	13	使用中
1001	1	1	/VirtualServer[ID=α]/vHBA[ID=0]/LU[ID=sdb]/Partition[ID=sdb1]/Read_per_sec	14	使用中
1002	1	1	/VirtualServer[ID=α]/vHBA[ID=0]/LU[ID=sdb]/Partition[ID=sdb1]/Write_per_sec	14	使用中
4001	1	1	/Cnv_HBA[ID=0]/vHBA[ID=0]/Read_per_sec	23	使用中
4002	1	1	/Cnv_HBA[ID=0]/vHBA[ID=0]/Write_per_sec	23	使用中
4101	1	1	/Cnv_HBA[ID=0]/Sum_Read_per_sec	22	使用中
4102	1	1	/Cnv_HBA[ID=0]/Stucked_Read_per_sec	22	使用中
4103	1	1	/Cnv_HBA[ID=0]/Sum_Write_per_sec	22	使用中
4103	1	1	/Cnv_HBA[ID=0]/Stucked_Write_per_sec	22	使用中

【図 14】



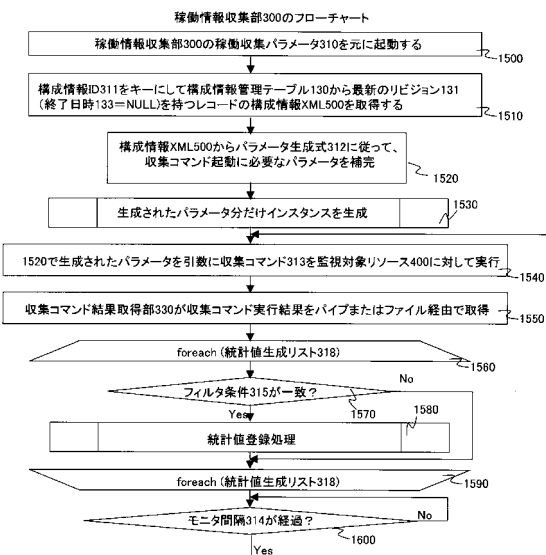
【图 16】



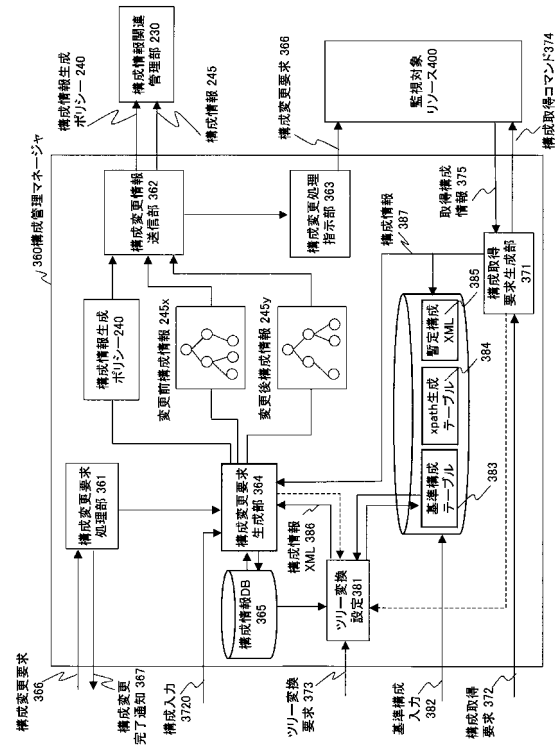
【図 17】

311	構成情報ID	1	...
312	パラメータ生成式	\$MyParam{InstanceXPath} = "/VimgoRoot/ServerRoot/RackInfoA rray/RackInfo/ChassisInfoArray/Ch assisInfo/BladeInfoArray/BladeInfo/ HvMnInfo/LparInfoArray/LparInfo"	
313	収集コマンド	\$RemoteShell \$RemoteHostIP /usr/bin/vmstat 1 2	
314	モニタ間隔	30秒	
統計値生成リスト318a	315a	フィルタ条件	\$FilterElem{Pattern} = "\$%s[%d.%+%s+%d.%+%s+% d.] +%s+%d.%+%s+%d.%+%s+ +%d.%+%s+%d.%+%s+%d.%+ +%s+(%d.%)+%s+(%d.%)+%s+ (%d.%)+%s+(%d.%)+"
	316a	統計情報生成式	\$VRDelem{Expression} = '\$user'
	317a	統計値ID検索式	\$VRDelem{SUID} = \$CPUStat/CPUUser
統計値生成リスト318b	315b	フィルタ条件	\$FilterElem{Pattern} = "\$%s[%d.%+%s+%d.%+%s+% d.] +%s+%d.%+%s+%d.%+%s+ +%d.%+%s+%d.%+%s+%d.%+ +%s+(%d.%)+%s+(%d.%)+%s+ (%d.%)+%s+(%d.%)+"
	316b	統計情報生成式	\$VRDelem{Expression} = '\$sys';
	317b	...	

【図 18】



【図 20】



【图 2 2】

Figure 1 is a block diagram illustrating a system architecture. The diagram shows multiple physical servers (物理サーバ) connected to a central network (ネットワーク) and a storage area network (SAN).

Each physical server (e.g., 物理サーバ 411-1, 物理サーバ 411-n) contains:

- メモリ (Memory) 422-m
- 仮想サーバ (Virtual Server) 423
- OS (Operating System) 421
- サーバ仮想化部 (Server Virtualization Department) 412

The virtual servers (423) are connected to the OS (421), which is connected to the virtualization department (412). The virtualization department (412) is connected to the network (430).

The network (430) is connected to a storage area network (SAN) 440. The SAN 440 is connected to a storage device (ストレージ装置) 450. The storage device 450 contains:

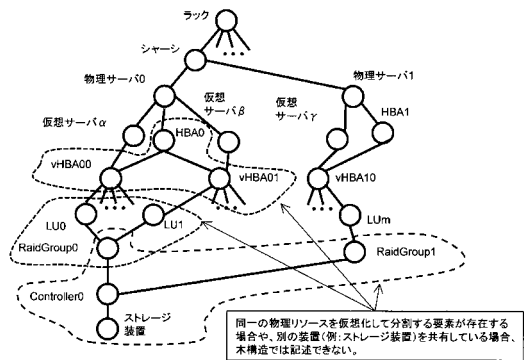
- RAID groups (RG) 485-1, 485-h, 485-k
- Logical Units (LU) 465-1, 465-h, 465-k

The network (430) is also connected to a management system (管理システム) 350. The management system 350 includes:

- モニタリングシステム (Monitoring System) 100
- 稼働情報集計部 (Operation Information Collection Department) 305
- 構成管理マネージャ (Configuration Management Manager) 360

The monitoring system (100) is connected to the operation information collection department (305), which is connected to the configuration management manager (360). The configuration management manager (360) is connected to the network (430).

【図 2 4】

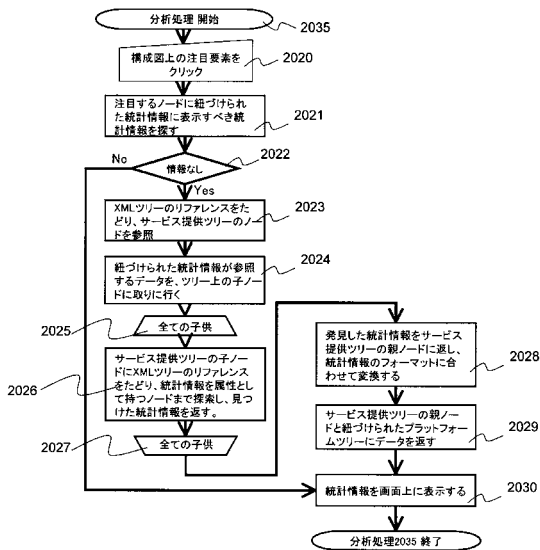


【図 2 6】

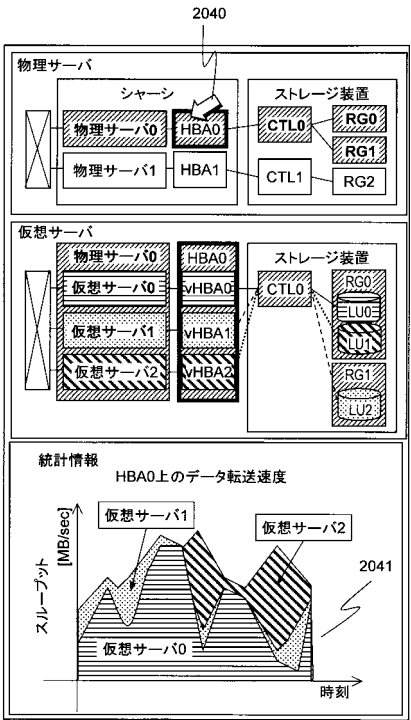
稼動統計分析テーブル 115

統計値ID	データ名	表示	演算
4001	Read/sec	—	集合(リンク先要素の統計値)
4002	Write/sec	—	集合(リンク先要素の統計値)
4101	Read/sec	合計	合計値(子ノード全体)
4102	Read/sec	積み上げ	集合(子ノード全体)
4103	Write/sec	合計	合計値(子ノード全体)
4104	Write/sec	積み上げ	集合(子ノード全体)

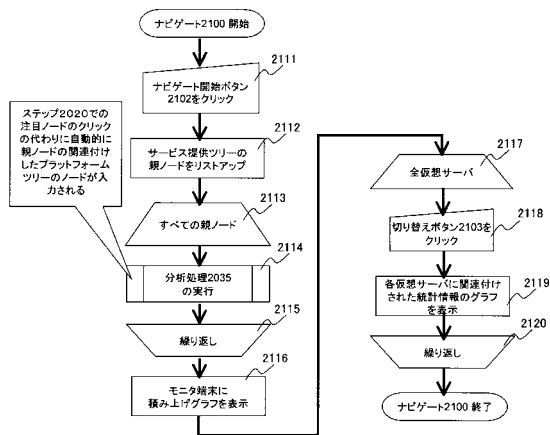
【図 2 8】



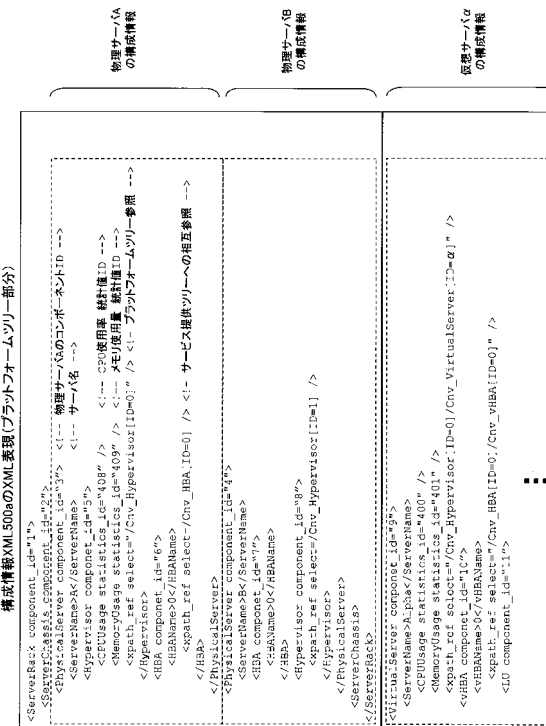
【図 2 9】



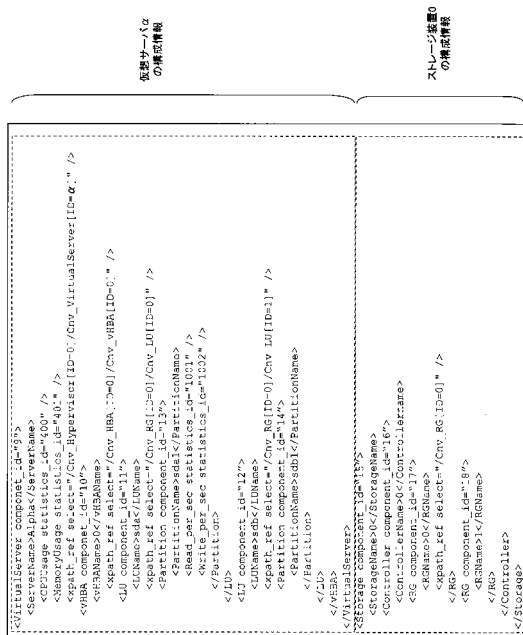
【図 3 1】



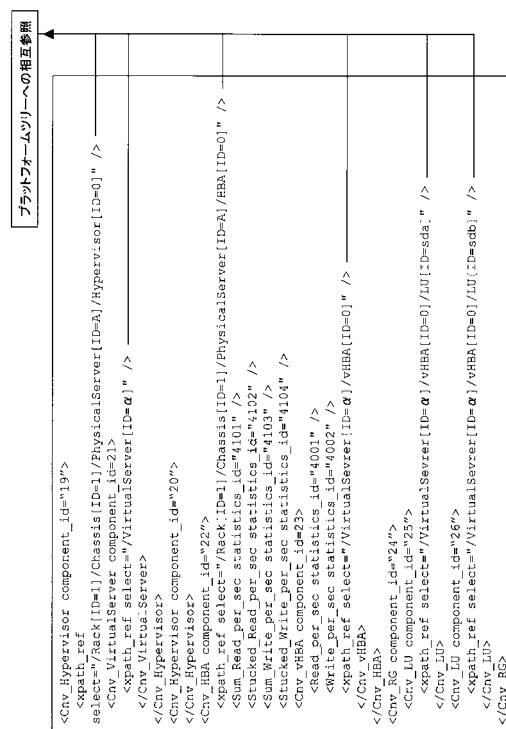
【図 3 2 A】



【図 3 2 B】



【図 3 3】



【図 3 4 A】

```

<ServerPack component_id="1">
  <ServerChassis component_id="2">
    <PhysicalServer component_id="3">
      <ServerName>S/ServerName</ServerName>
      <Hypervisor component_id="5">
        <CPUMUsage statistics_id="408" />
        <MemoryUsage statistics_id="409" />
        <Hypervisor>
          <xpath_ref select="//Cnv_Hypervisor[ID=0]" />
        </Hypervisor>
      </PhysicalServer>
      <HBA component_id="6">
        <HBAName>O/C/HBName</HBAName>
        <xpath_ref select="//Cnv_HBA[ID=0]" />
      </HBA>
    </PhysicalServer>
  </ServerChassis>
  <PhysicalServer component_id="4">
    <ServerName>S/ServerName</ServerName>
    <Hypervisor>
      <xpath_ref select="//Cnv_Hypervisor[ID=0]" />
    </Hypervisor>
    <HBA>
      <HBAName>O/C/HBName</HBAName>
    </HBA>
    <Hypervisor component_id="8">
      <xpath_ref select="//Cnv_Hypervisor[ID=1]" />
    </Hypervisor>
    <PhysicalServer>
      <ServerChassis>
        <ServerPack component_id="9">
          <ServerName>AlphaX/ServerName</ServerName>
          <CPUMUsage statistics_id="407" />
          <MemoryUsage statistics_id="401" />
          <xpath_ref select="//Cnv_Hypervisor[ID=0]" />
          <xpath_ref select="//Cnv_Hypervisor[ID=1]" />
        </ServerPack>
      </ServerChassis>
    </PhysicalServer>
  </ServerPack>

```

【图 3 4 B】

```

<vHBA component_id="10">
  <vHBAName></vHBAName>
  <xpath_ref select="/Cnv_HBA[ID=0]/Cnv_vHBA[ID=0]" />
  <LU component_id="11">
    <LUName></LUName>
    <xpath_ref select="/Cnv_LU[ID=0]/Cnv_LU[ID=0]" />
    <Partition component_id="13">
      <PartitionName></PartitionName>
      <xpath_ref select="/Cnv_Partition[ID=0]/Cnv_Partition[ID=0]" />
      <Read per sec statistics_id="1001" />
      <Write per sec statistics_id="1002" />
    </Partition>
  </LU>
</LU>
<LU component_id="12">
  <LUName></LUName>
  <xpath_ref select="/Cnv_RS[ID=0]/Cnv_LU[ID=1]" />
  <Partition component_id="14">
    <PartitionName></PartitionName>
  </Partition>
</LU>
</VHBA>
</VirtualServer>
<Storage component_id="15">
  <StorageName></StorageName>
  <Controller component_id="16">
    <ControllerName></ControllerName>
  </Controller>
  <RG component_id="17">
    <RGName></RGName>
  </RG>
  <xpath_ref select="/Cnv_RS[ID=0]" />
  </RG>
  <RG component_id="18">
    <RGName></RGName>
  </RG>
  <Controller>
    </Controller>
  </Storage>

```

【図 3 5 A】

```

<Cnv_Hypervisor component_id="19">
  <xpath_ref select="//Rack[ID=1]/Chassis[ID=1]/PhysicalServer[ID=A]/Hypervisor[ID=0]" />
  <Cnv_VirtualServer component_id="21">
    <xpath_ref select="//VirtualServer[ID=α]" />
  </Cnv_VirtualServer>
</Cnv_Hypervisor>

<Cnv_VirtualServer component_id="20">
  <Cnv_VirtualServer component_id="21">
    <xpath_ref select="//VirtualServer[ID=α]" />
  </Cnv_VirtualServer>
</Cnv_VirtualServer>

<Cnv_Hypervisor>
  <Cnv_Hypervisor component_id="22">
    <xpath_ref select="//Rack[ID=1]/Chassis[ID=1]/PhysicalServer[ID=A]/HBA[ID=0]" />
    <xpath_ref select="//Storage[id="101]" />
    <Stucked_Read_per_statistics id="4102" />
    <Sum_Write_per_sec_statistics id="4103" />
    <Stucked_Write_per_sec_statistics id="4104" />
  </Cnv_Hypervisor>
</Cnv_Hypervisor>

```

【図 3 5 B】

```

<Cnv VHBA component_id="23">
  <Read per sec statistics_id="4002"> />
  <Write per sec statistics_id="4105"> />
  <xpath ref select="//VirtualServer[ID=α]/VHBA[ID=0]" />
</Cnv VHBA>
</Cnv VHBA>
<Cnv FBK component_id="27">
  <xpath ref select="//Back[ID=1]/Chassis[ID=1]/PhysicalServer[ID=B]/HBA[ID=1]" />
  <Sum Read per sec statistics_id="4105"> />
  <Stucked Read per sec statistics_id="4106"> />
  <Sum Write per sec statistics_id="4107"> />
  <Stucked Write per sec statistics_id="4108"> />
</Cnv FBK component_id="23">
  <Read per sec statistics_id="4001"> />
  <Write per sec statistics_id="4002"> />
  <xpath ref select="//VirtualServer[ID=α]/VHBA[ID=0]" />
</Cnv VHBA>
</Cnv FBK>
<Cnv RG component_id="24">
  <Cnv IU component_id="25">
    <xpath ref select="//VirtualServer[ID=α]/VHBA[ID=0]/LU[ID=sdα]" />
    </Cnv IU>
  </Cnv IU component_id="26">
    <xpath ref select="//VirtualServer[ID=α]/VHBA[ID=0]/LU[ID=sdβ]" />
    </Cnv IU>
  </Cnv RG>

```

【図 36】

基準構成テーブル381

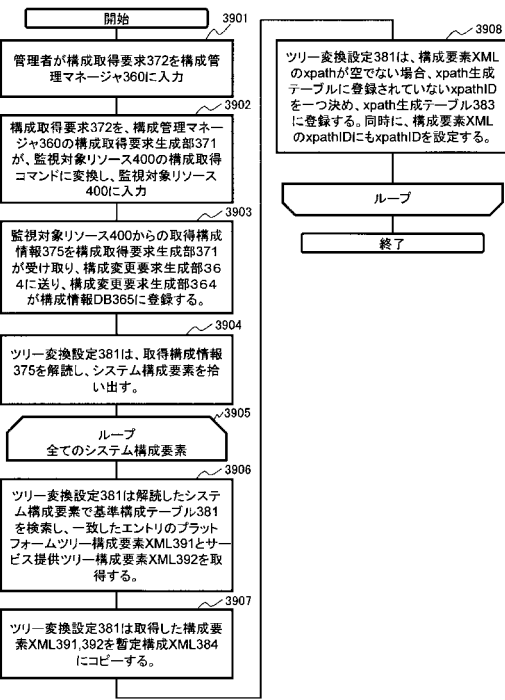
要素名	プラットフォーム ツリー 構成要素XML	サービス提供 ツリー 構成要素XML
サーバラック	<pre><platform> <name> ServerRack </name> <parent> </parent> <child> ServerChassis </child> <xpathID> </xpathID> </platform></pre>	<pre><service> <name> </name> <parent> </parent> <child> </child> <xpathID> </xpathID> </service></pre>
...	...	...
ハイパーバイザ	<pre><platform> <name> Hypervisor </name> <parent> PhysicalServer </parent> <child> VirtualServer </child> <xpathID>value</xpathID> </platform></pre>	<pre><service> <name> Hypervisor </name> <parent> </parent> <child> VirtualServer </child> <xpathID>value</xpathID> </service></pre>
仮想サーバ	<pre><platform> <name> VirtualServer </name> <parent> </parent> <child> vHBA </child> ... <xpathID> value</xpathID> </platform></pre>	<pre><service> <name> VirtualServer </name> <parent> Hypervisor </parent> <child> </child> <xpathID>value</xpathID> </service></pre>
...	...	...

【図 37】

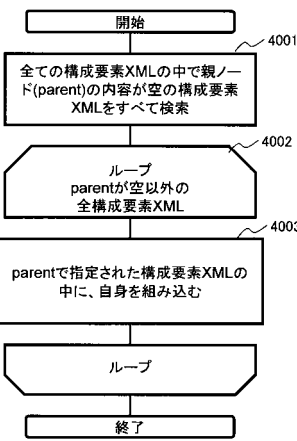
xpath生成テーブル383

xpath ID	プラットフォーム ツリー xpath	サービス提供 ツリー xpath
001	<pre><xpath_ref select="/Cnv_Hypervisor[T D=0]" /></pre>	<pre><xpath_ref select="/Rack[ID=1]/Chassis [ID=1]/PhysicalServer[ID=A] /Hypervisor[ID=0]" /></pre>
...	...	...
...	...	...

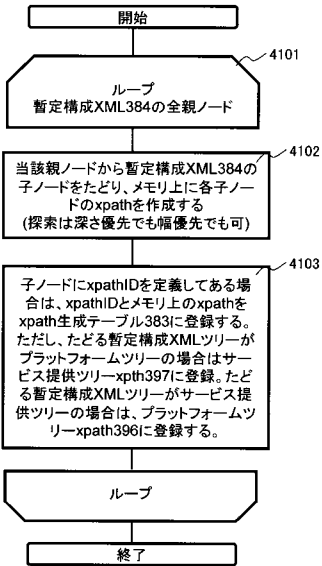
【図 38】



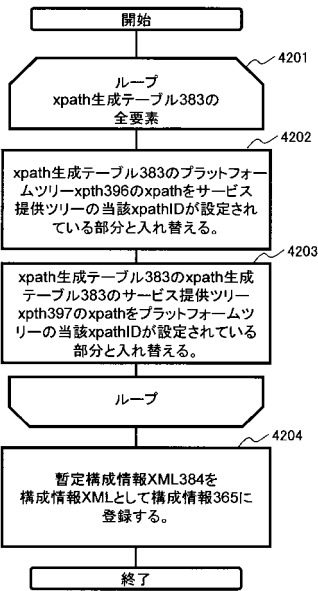
【図 39】



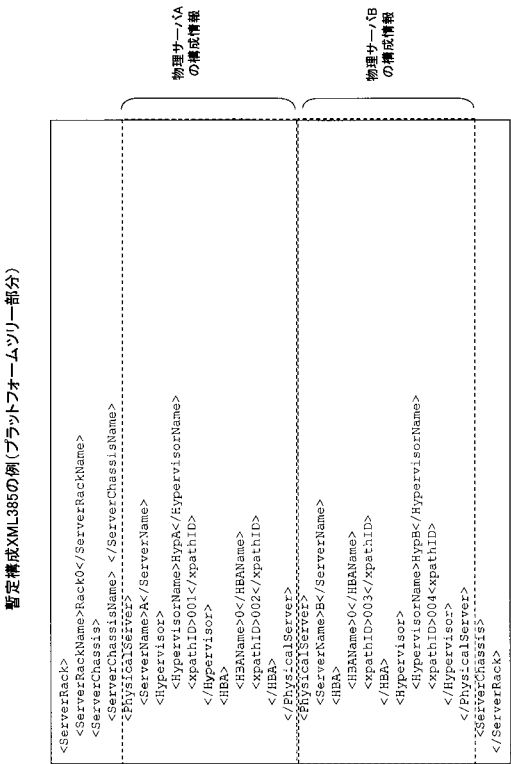
【図 4 0】



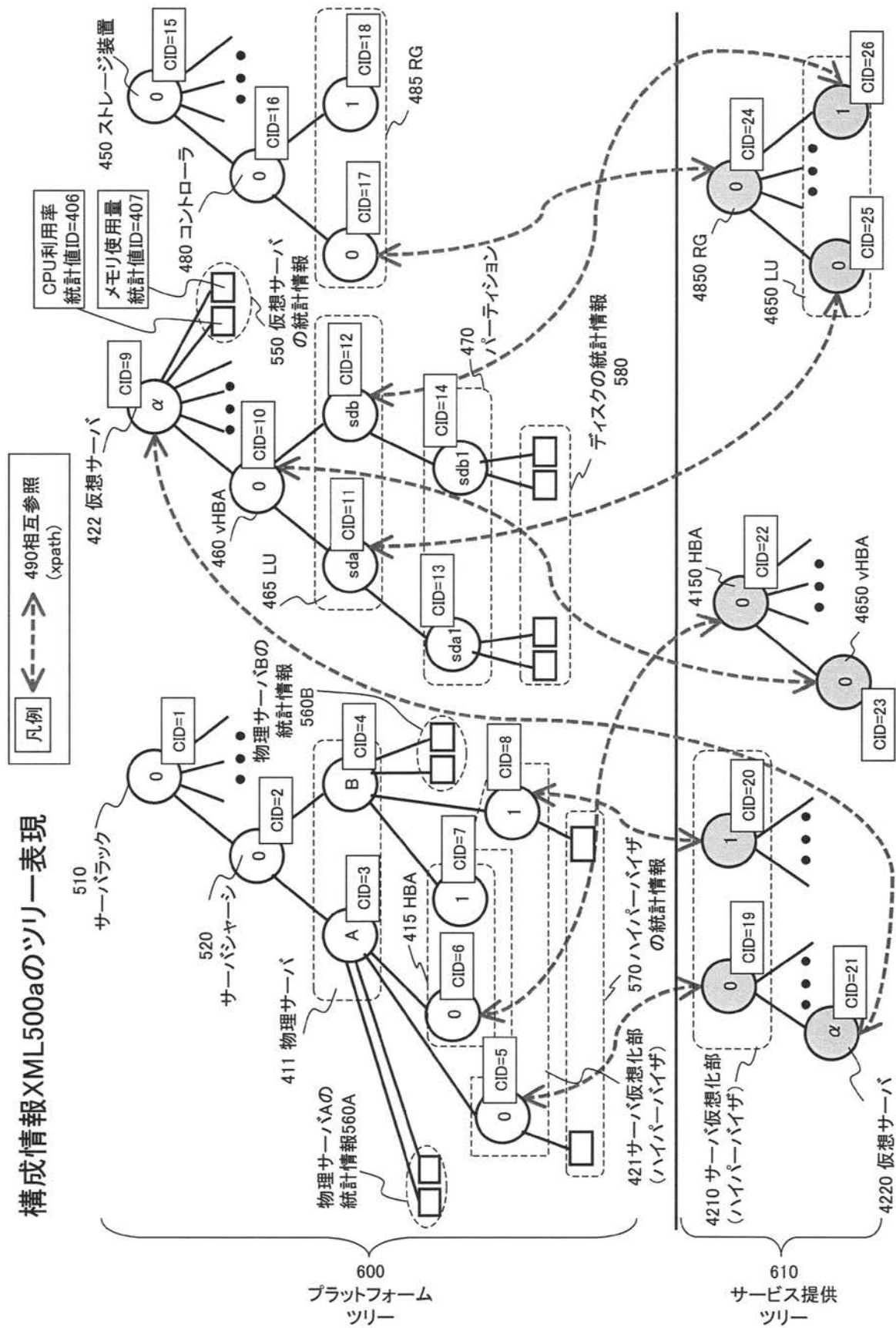
【図 4 1】



【図 4 2】

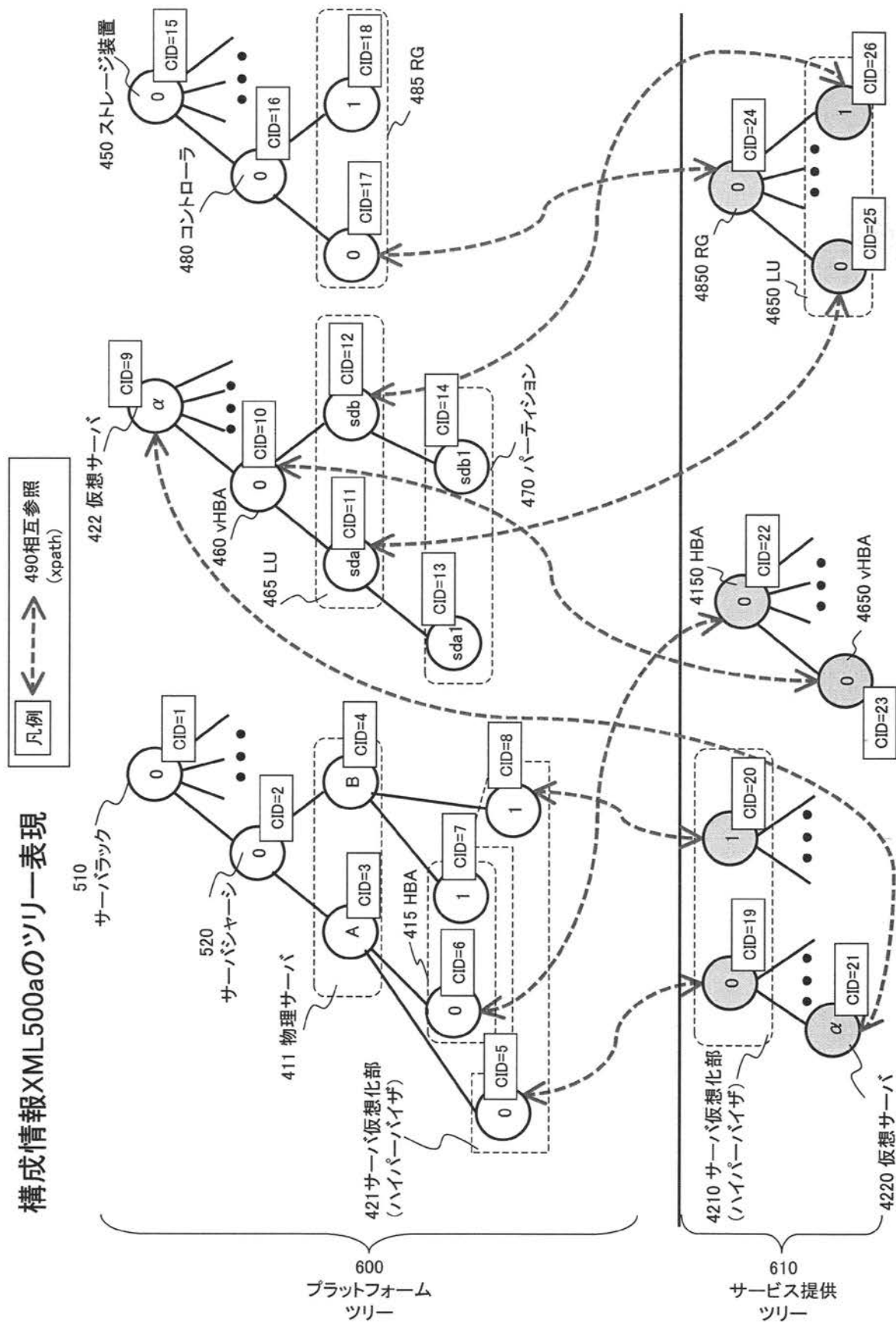


構成情報XML500aのツリー表現

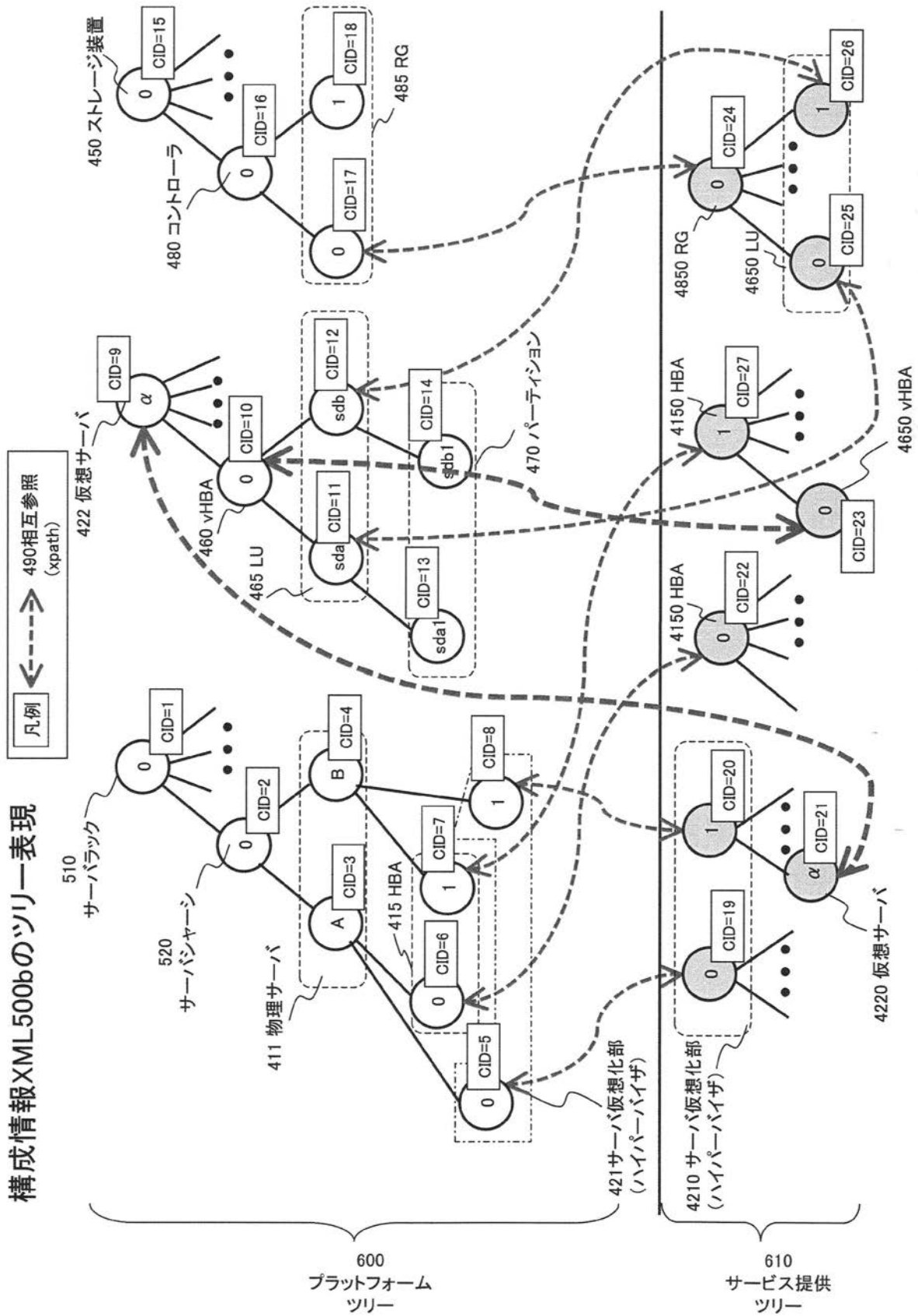


【図 8】

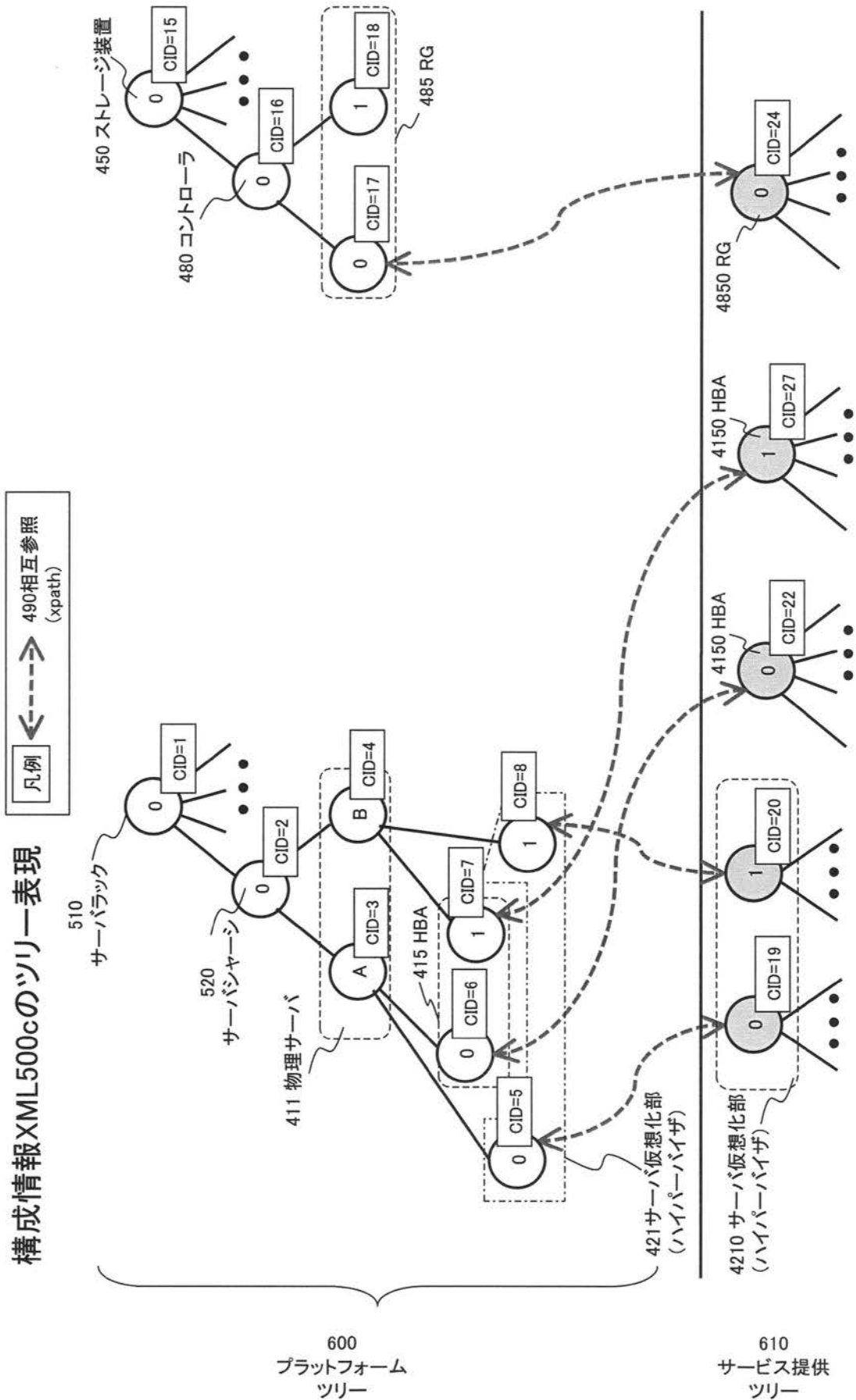
構成情報XML500aのツリー表現



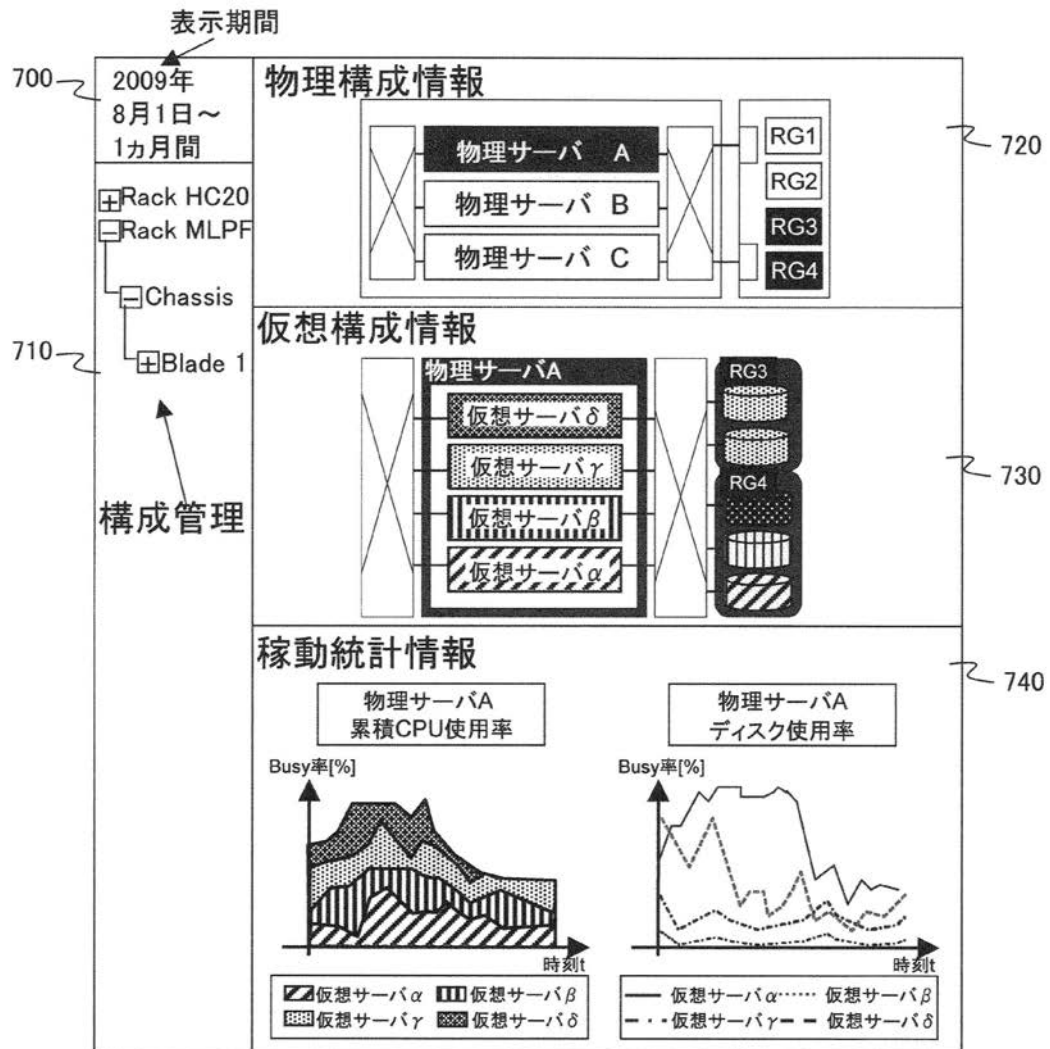
構成情報XML500bのツリー表現



【図10】

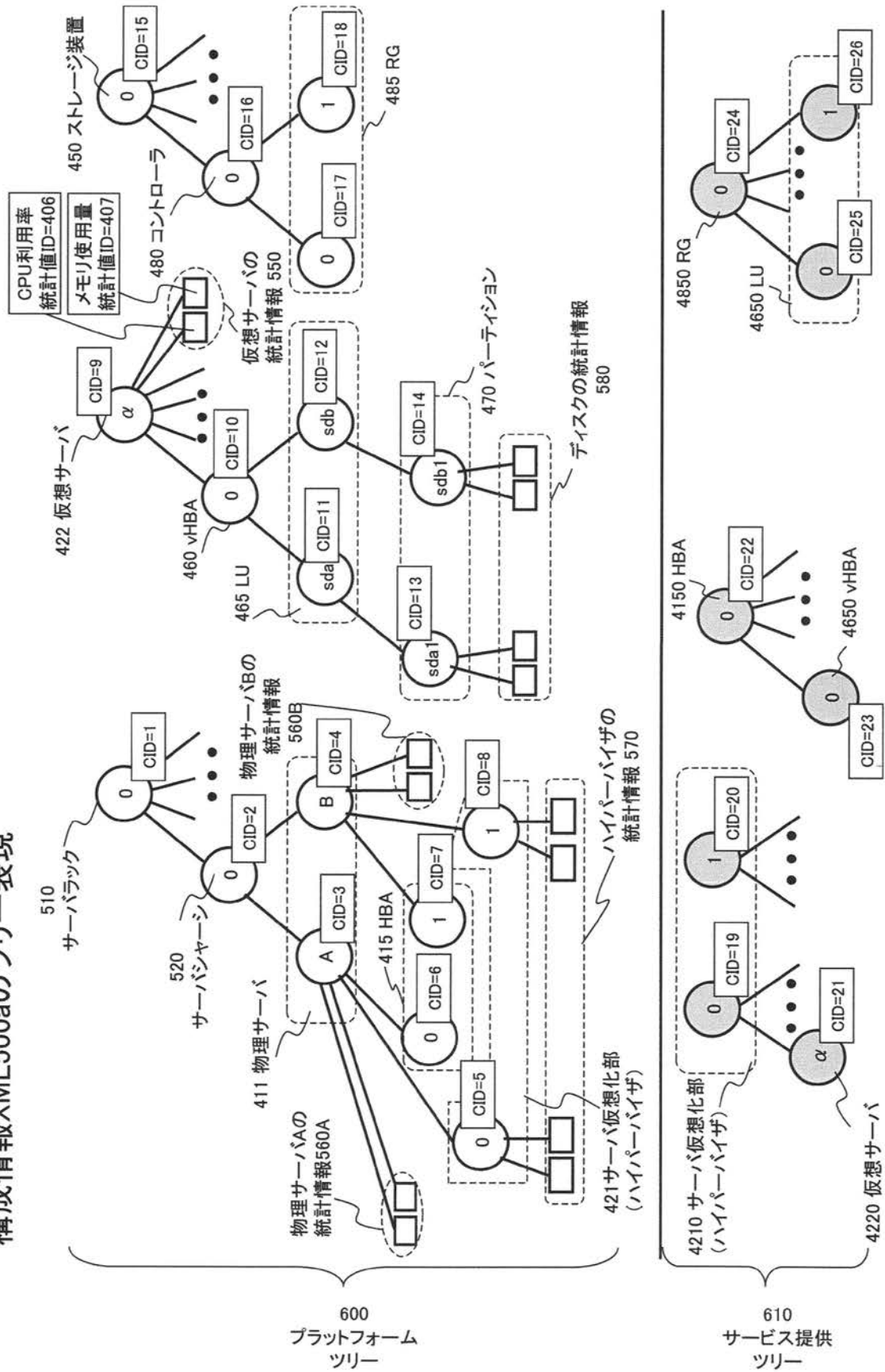


【図 15】

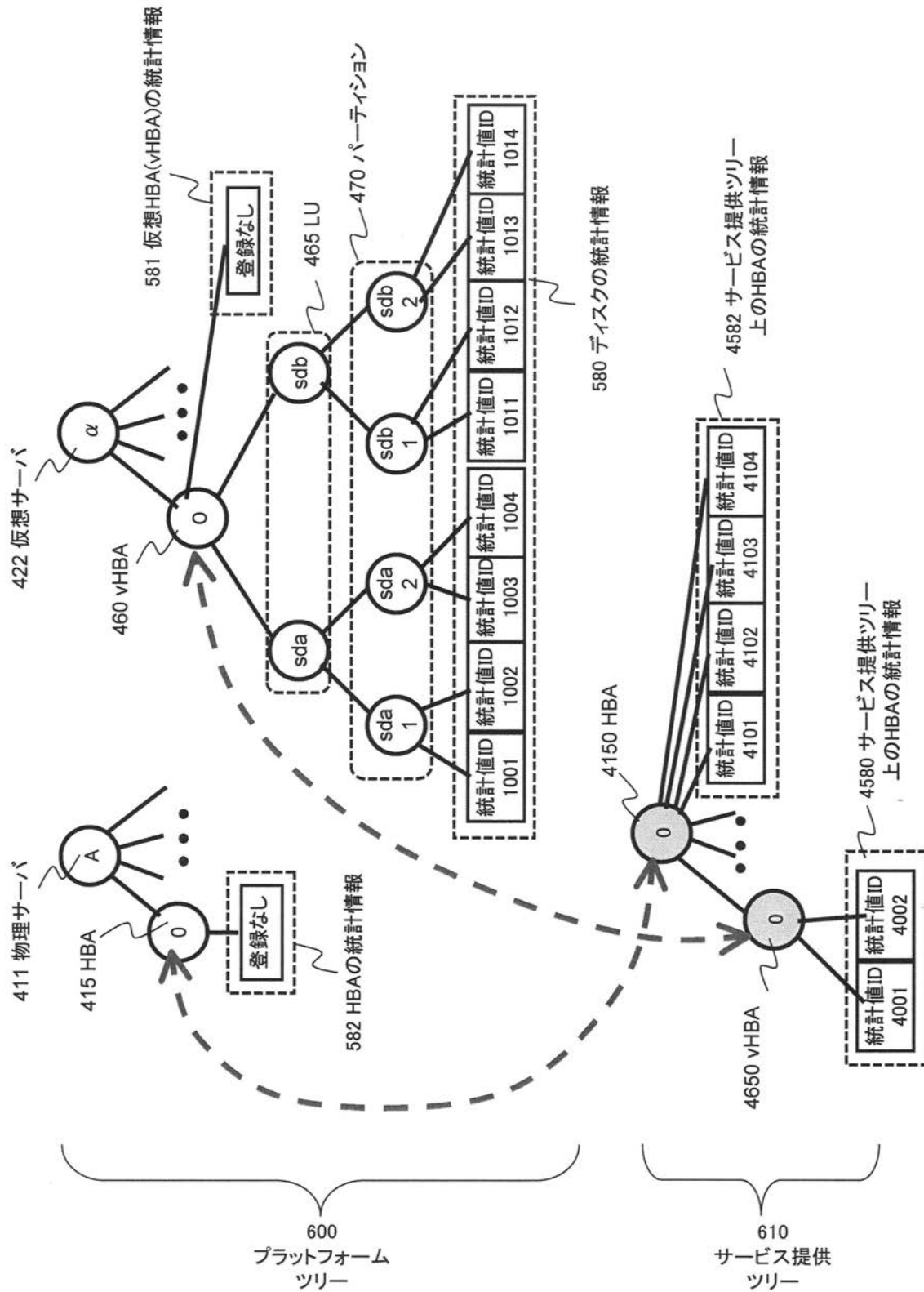


【図 23】

構成情報XML500aのツリー表現



【图 2 5】



【図 27】

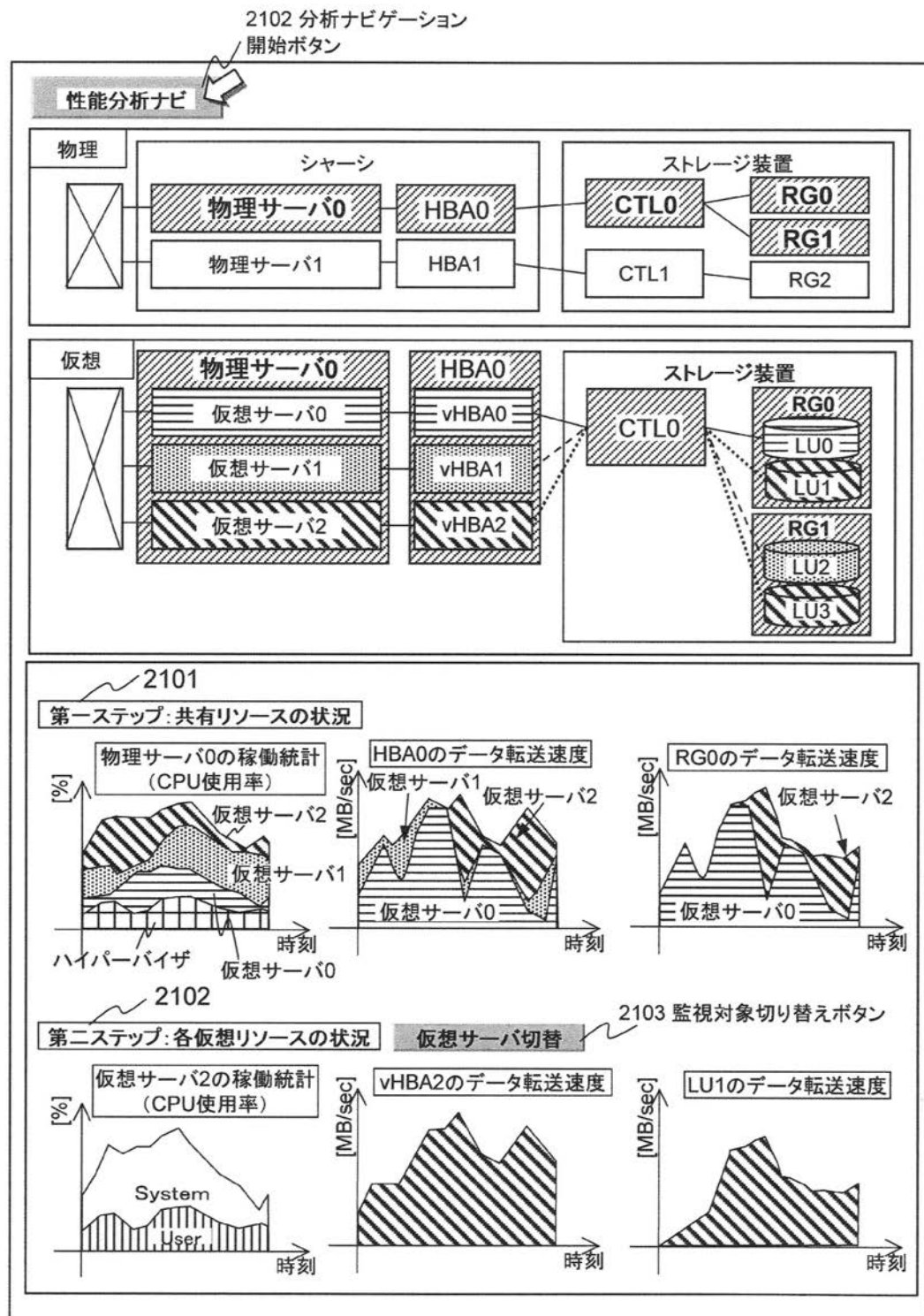
2015 稼動統計分析の演算を指定する入力画面

分析演算入力	2011	2012	2013
場変換	演算	統計値の種類	
HBA ⇄ vHBA	合計	Read/sec	
HBA ⇄ vHBA	積み上げ	Read/sec	
HBA ⇄ vHBA	合計		
	積み上げ		
	平均		
	ユーザ指定の演算		

OK Cancel

2014

【図 30】



フロントページの続き

(72)発明者 川本 真一

東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内

審査官 田中 幸雄

(56)参考文献 米国特許出願公開第2007/0094367 (US, A1)

米国特許第07383327 (US, B1)

特開2005-062941 (JP, A)

特開2011-191807 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/50

G06F 9/46

G06F 11/30

G06F 13/00